IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 6, 2021, accepted December 26, 2021, date of publication January 4, 2022, date of current version January 19, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3140121

Profile Aggregation-Based Group Recommender Systems: Moving From Item Preference Profiles to Deep Profiles

LE NGUYEN HOAI NAM
Faculty of Information Technology, VNUHCM-University of Science, Ho Chi Minh City 72700, Vietnam
Vietnam National University Ho Chi Minh City, Ho Chi Minh City 72700, Vietnam

e-mail: lnhnam@fit.hcmus.edu.vn

This research is partially supported by the research funding from the Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam.

ABSTRACT To meet the increasing demand for group activities, single-user recommender systems need to be scaled up to provide recommendations to groups of users. This issue is solved by aggregating item preference profiles of individual group members into a single item preference profile, thereby allowing recommendations to be created for this item preference profile. In this paper, we introduce the concept of deep profiles of users, and we propose group recommendation methods based on the aggregation of group members' deep profiles, instead of item preference profiles as in previous studies. The term *deep profile* refers to the users' profiles that lie deep within the recommendation algorithms. Experiments have shown that group recommendations based on deep profiles give higher efficiency in terms of F1-score and nDCG than those based on item preference profiles.

INDEX TERMS Collaborative filtering, group recommender systems, recommender systems.

I. INTRODUCTION

Today, people are confronted by a large number of group activities [1]–[4]. At work, they form groups to be able to accomplish projects. During leisure time, people participate in group activities, such as listening to music together [5], watching movies together [6], traveling together [7], etc. This creates a challenge for information systems in transitioning from serving a single user to serving a group of users.

Providing recommendations is an integral part of modern information systems. They work on the item preferences observed by the users to predict what items will be suitable for them in the future. With such a role, recommender systems have greatly contributed to overcoming information overload [8], [9]. Single-user recommender systems are being increasingly used by companies such as Amazon, Netflix, and eBay [10], [11]. However, to be able to meet the demand for group activities today, they need to change to be able to make recommendations to a group of users [12], [13]. Thanks to group recommender systems, a company can assign the most suitable projects to its engineering teams; a class can choose a restaurant to hold the year-end party; a family can find a movie to watch together.

The heart of a single-user recommender system is its ability to predict a user's unknown rating for an item. This is effectively performed by collaborative filtering. In collaborative filtering, a user's preferences can be revealed through users who have similar interests to him/her in the past [14]. This principle can be implemented in two ways, memory-based and model-based. Memory-based collaborative filtering finds users with similar interests to the active user, called neighbors [15], [16]. Then, the active user's predicted rating for an item will come from the aggregation of neighbors' ratings for the item. With such processing, it is easy to interpret a recommendation from the system to the active user [17]–[19]. However, calculating the preference similarity between each pair of users has made it very difficult to implement a memory-based collaborative filtering recommender system in large-scale settings [20]–[22]. In this context, model-based collaborative filtering emerges as the first choice [11], [23], [24]. It learns patterns from observed ratings of users. These compact patterns help make predicting the ratings easier. The latent factor model is a state of the art of model-based collaborative

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

filtering. It is aimed at the representation of users and items on latent factors. The match between a user and an item defined on latent factors determines the user's rating for the item [25]–[27].

Contrary to single-user recommender systems, group recommender systems aim to predict the rating of the whole group for an item after all group members have experienced the item together. The challenge for a group recommender system comes from the fact that each group member's preference for an item can be greatly impacted when he/she experiences the item with other group members. Conflicts may even arise among group members in the process of deciding whether the group likes the item or not [28], [29]. The group recommendation issue is often solved by a profile aggregation-based method [12], [28]–[31]. Specifically, the item preference profiles of individual group members will be aggregated into a single item preference profile of the whole group. Then, it is possible to use single-user recommendation algorithms to make recommendations to the group's item preference profile.

However, the item preference profiles have always been very sparse. Therefore, it is difficult for the aggregation of group members' item preference profiles to create a single item preference profile that fully reflects the preferences of the whole group. Furthermore, the system only recognizes the group's item preference profile after the group is input into the system. Therefore, compared to recommendations for a regular user, recommendations for a group incur more computational costs. This paper aims to solve the above two problems in order to further improve the profile aggregation-based group recommendation. Specifically, our contributions are as follows:

- We introduce the concept of *deep profiles*. They refer to users' profiles deep within the recommendation algorithms. Their advantage over item preference profiles is that they have fewer dimensions and are completely specified.
- We propose a group recommendation based on the aggregation of group members' deep profiles instead of item preference profiles as in previous studies.

The structure of this paper is as follows. In section II, it focuses on presenting related works. In section III, we state the motivation for this paper based on the review of these works. Our proposals are presented in section IV. Our proposed experimental methods are in section V. Finally, section VI is the conclusions and future works. The symbols in Table 1 are used in this paper.

II. LITERATURE REVIEW

Our study is related to using collaborative filtering for group recommendations. Therefore, after the problem definition in subsection II.A, subsection II.B focuses on presenting recent achievements of collaborative filtering for single-user recommendations. Existing studies on extending collaborative filtering to group recommendations are summarized in subsection II.C.

TABLE 1. The symbols.

Symbol	Meaning
$\mathbf{R}$	User-item rating matrix
$\mathbf{R}'$	User-item implicit feedback matrix
$\mathbf{C}$	User-item sentiment matrix
$\mathbf{H}$	User-factor matrix
$\mathbf{Q}$	Item-factor matrix
$\mathbf{I}$	Identity matrix
$\mathbf{H}_{u,:}$	The u^{th} row matrix of the matrix $\mathbf{H}$
$\mathbf{Q}_{i,:}$	The i^{th} row matrix of the matrix $\mathbf{Q}$
$R_{u,i}$	Observed rating of user u for item i
$R'_{u,i}$	Implicit feedback of user u for item i
$C_{u,i}$	Sentiment score of user u for item i
$\widehat{R}_{u,i}$	Predicted rating of user u for item i
k	The number of selected neighbors
m	The number of users
n	The number of items
s	The number of latent factors
μ_u	The average of the ratings observed by user u
μ_i	The average of observed ratings for item i
μ	The average of all observed ratings
o_u	The bias of user u
p_i	The bias of item i
w_u	The weight of user u
$sim_{u,q}$	The preference similarity between user u and user q
λ	The regularization weight
$\mathbb{R}$	The set of user-item pairs whose ratings are observed
$\mathbb{R}'$	The set of user-item pairs whose implicit feedback are observed
$\mathbb{G}$	A group
$\mathbb{I}_u$	The set of items rated by user u
$\mathbb{U}_i$	The set of users that rated item i
$\mathbb{DP}_{\mathbb{G}}$	The deep profile of group $\mathbb{G}$
$\lvert * \rvert$	The number of elements in a set
$\lVert * \rVert$	The Frobenius norm of a matrix

A. PROBLEM DEFINITION

The implementation of a recommender system consists of two phases, the online phase and the offline phase. The most basic data that can be input into the offline phase is a user-item rating matrix $\mathbf{R} = [R_{u,i}]$ where $u = 1 \ldots m$, $i = 1 \ldots n$, m is the number of users, and n is the number of items representing the item preferences that are observed by the users [8], [32]. Recommendation algorithms work on specific entries in this matrix, which are observed ratings ($R_{u,i} \neq *$), to model user preferences by formulas, rules, etc. Thus, in the online phase of single-user recommender systems, the ratings of an active user for each item that the user has not experienced ($R_{u,i} = *$)

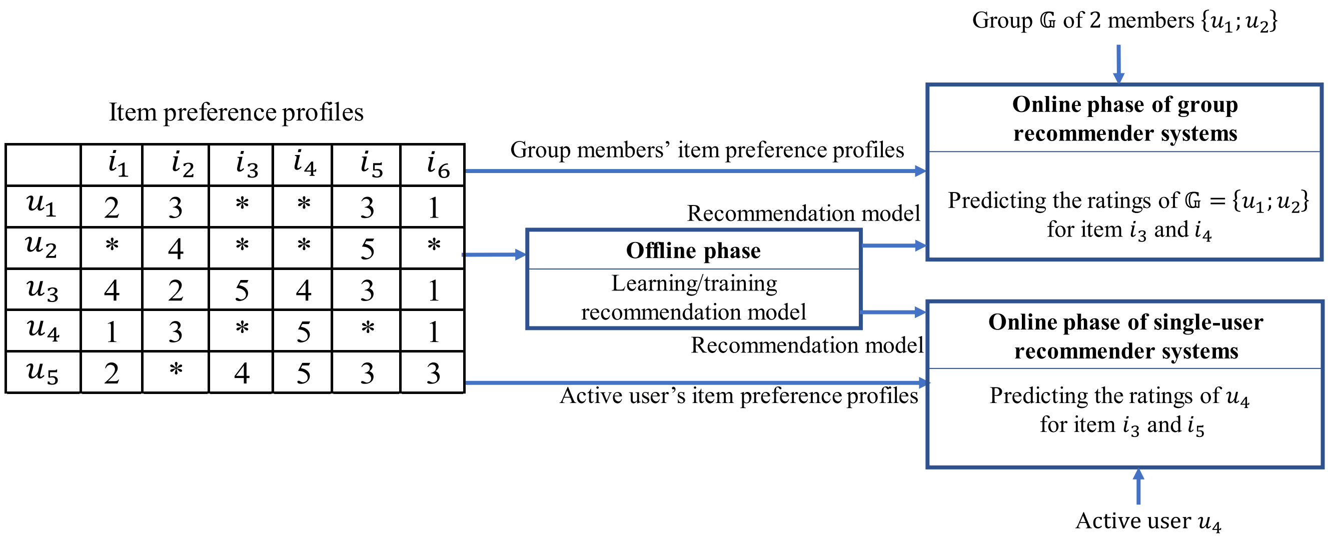


FIGURE 1. The differences between a single-user recommender system and a group recommender system.

will be predicted. In contrast to single-user recommender systems, the online phase of group recommender systems serves a set of active users, who are members of a group. For group recommender systems, the ratings of the whole group for each target item that all group members have not experienced need to be predicted. Finally, the top items with the highest predicted ratings will be selected to recommend an active user in the case of single-user recommender systems and a group of active users in the case of group recommender systems.

As shown in Fig. 1, with the active user u_4, the task of the online phase of a single-user recommender system is to predict the ratings of u_4 for item i_3 and i_5 because $R_{u_4,i_3} = *$ and $R_{u_4,i_5} = *$. For a group recommender system, the ratings of a group $\mathbb{G} = \{u_1; u_2\}$ for item i_3 and i_4 need to be predicted because $R_{u_1,i_3} = *$, $R_{u_2,i_3} = *$, $R_{u_1,i_4} = *$, and $R_{u_2,i_4} = *$. The differences between the two online processes of predicting ratings given by a single user and given by a group of users create the distinct nature of a single-user recommender system and a group recommender system.

B. SINGLE-USER RECOMMENDER SYSTEMS

Collaborative filtering is a class of algorithms for rating prediction in single-user recommender systems. It is divided into two types, memory-based collaborative filtering and model-based collaborative filtering.

1) SINGLE-USER RECOMMENDER SYSTEMS USING MEMORY-BASED COLLABORATIVE FILTERING

Memory-based collaborative filtering is also known as neighbor-based collaborative filtering. It is based on a principle that a user's preferences can be inferred from those with similar interests to him/her, called his/her neighbors [14], [17], [18]. Therefore, in the offline phase of memory-based collaborative filtering, a formula is used to measure the preference similarity of each pair of users, denoted by $sim_{u,q}$ where $u = 1 \ldots m$ and $q = 1 \ldots m$. For an active user u in the online phase, a set of k neighbor users that are similar to u and rated a target item i, denoted by $\mathbb{N}_u^{(k)}$, is determined. The observed ratings of the neighbor users for the target item i, i.e., $R_{q',i}$ where $q' \in \mathbb{N}_u^{(k)}$, are aggregated to produce the rating of u for i [33], denoted by $\hat{R}_{u,i}$, as follows:

$$\hat{R}_{u,i} = \mu_u + \frac{\sum_{q' \in \mathbb{N}_u^{(k)}} sim_{u,q'}.(R_{q',i} - \mu_{q'})}{\sum_{q' \in \mathbb{N}_u^{(k)}} sim_{u,q'}} \quad (1)$$

In Eq. (1), the ratings of neighbor users $q' \in \mathbb{N}_u^{(k)}$ are subtracted from their average ($\mu_{q'}$) to eliminate bias. Therefore, to obtain the correct rating of u for i, reconstruction needs to be performed by adding to the prediction an amount equal to the average of u (μ_u).

It can be seen that the chosen preference similarity measure determines the accuracy of memory-based collaborative filtering. Some recent preference similarity measures for high accuracy are OS [14] and LM [34]. OS [14] combines Percentage of Non-Common Ratings (PNCR) and Absolute Percentage of Non-Common Ratings (PNCR) in a similarity measure as follows:

$$\begin{aligned} sim_{u,q}^{(OS)} &= sim_{u,q}^{(PNCR)}.sim_{u,q}^{(ADF)} \\ sim_{u,q}^{(PNCR)} &= exp(-\frac{n - \left|\mathbb{I}_u \cap \mathbb{I}_q\right|}{n}) \\ sim_{u,q}^{(ADF)} &= \frac{\sum_{i \in \mathbb{I}_u \cap \mathbb{I}_q} exp\left(-\frac{|R_{u,i} - R_{q,i}|}{max\{R_{u,i}, R_{q,i}\}}\right)}{\left|R_{u,i} - R_{q,i}\right|} \end{aligned} \quad (2)$$

where $\mathbb{I}_u$ and $\mathbb{I}_q$ are the item that user u and user q have rated, respectively. LM [34] defines a landmark set to represent users instead of the item set. Landmarks are the users with the most observed ratings. In the space of s landmarks, denoted by $g = 1 \ldots s$, the similarity of two users is computed as follows:

$$sim_{u,q}^{(Landmarks)} = \frac{\sum_{g=1}^{s} l_{u,g}.l_{q,g}}{\sqrt{\sum_{g=1}^{s} l_{u,g}^2}\sqrt{\sum_{g=1}^{s} l_{v,g}^2}}$$
$$\forall g = 1 \ldots s: \quad l_{u,g} = cos_{u,g} \tag{3}$$

2) SINGLE-USER RECOMMENDER SYSTEMS USING MODEL-BASED COLLABORATIVE FILTERING

In this subsection, we focus on the latent factor model, a state-of-the-art model-based collaborative filtering. The methodology across the latent factor models is to learn s latent factors for the representation of user preferences, which are the s-dimensional user latent factor vectors $\mathbf{H}_{u,:}u = 1 \ldots m$ in the user-factor matrix $\mathbf{H}_{mxs}$, and the item properties, which are the s-dimensional item latent factor vectors $\mathbf{Q}_{i,:}i = 1 \ldots n$ in the item-factor matrix $\mathbf{Q}_{nxs}$. Then an unknown rating of a user u for an item i is predicted by matching the two corresponding latent factor vectors, i.e., $\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}$ [35], [36]. Next, we summarize the process of learning latent factors and the process of predicting ratings in some typical latent factor models.

a: BIAS-SVD

The process of learning the user latent factor vectors and the item latent factor vectors determines the effectiveness of latent factor models. It is associated with an objective function to optimize latent factors with the collected data. The most basic data that systems can access are ratings observed by users. With this type of data, a traditional latent factor model named Bias-SVD [37] optimizes the distances between the observed ratings ($R_{u,i} \neq *$) and their predicted ratings ($\hat{R}_{u,i}$). A unique feature of Bias-SVD is the integration of user biases, denoted by $o_u\, u = 1 \ldots m$, and item biases, denoted by $p_i\, i = 1 \ldots n$, into the predicted ratings. Specifically, the processes for learning latent factors and predicting ratings in Bias-SVD are as follows:

$$\hat{R}_{u,i} \approx o_u + p_i + \mu + \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}} \tag{4}$$

$$\underset{\substack{\mathbf{H},\mathbf{Q},\\ o_u\, u=1\ldots m\\ p_i\, i=1\ldots n}}{minimize}\ J^{(BiasSVD)}$$
$$J^{(BiasSVD)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}} \left(R_{u,i} - \hat{R}_{u,i}\right)^2$$
$$\Leftrightarrow$$
$$J^{(BiasSVD)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}} \begin{pmatrix} R_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^T \end{pmatrix}^2 + \frac{\lambda}{2}\begin{pmatrix} ||\mathbf{H}||^2 + ||\mathbf{Q}||^2 \\ + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2 \end{pmatrix} \tag{5}$$

where $\mathbb{R} = \{(u, i)|u = 1 \ldots m \wedge i = 1 \ldots n \wedge R_{u,i} \neq *\}$ are the user-item combinations at which the rating is observed; the last component is as a regularization to avoid overfitting with the weight λ; μ is the average of the ratings observed by all users.

b: BIAS-SVD++

Recently, with the development of the internet, the collected data for recommender systems is getting richer and richer. Combining them with rating data greatly improves the quality of latent factors. Bias-SVD++ [38] is a model that uses implicit feedback data to support rating data in latent factor models. Specifically, the s-dimensional latent factor vector of each user in the Bias-SVD, i.e, $\mathbf{H}_{u,:}u = 1 \ldots m$, will be supplemented with the s-dimensional latent factor vectors of the item set the user has interacted with, denoted by $\mathbb{I}_u\, u = 1 \ldots m$, as follows:

$$\hat{R}_{u,i} \approx o_u + p_i + \mu + \left(\mathbf{H}_{u,:} + |\mathbb{I}_u|^{-\frac{1}{2}}.\sum_{j\in\mathbb{I}_u}\mathbf{T}_{j,:}\right).\mathbf{Q}_{i,:}^{\mathbf{T}} \tag{6}$$

$$\underset{\substack{\mathbf{H},\mathbf{Q},\mathbf{T}\\ o_u\, u=1\ldots m\\ p_i\, i=1\ldots n}}{minimize}\ J^{(BiasSVD++)}$$
$$J^{(BiasSVD++)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}} \left(R_{u,i} - \hat{R}_{u,i}\right)^2$$
$$\Leftrightarrow$$
$$J^{(BiasSVD++)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}} \begin{pmatrix} R_{u,i} - o_u - p_i - \mu \\ -\left(\mathbf{H}_{u,:} + |\mathbb{I}_u|^{-\frac{1}{2}}.\sum_{j\in\mathbb{I}_u}\mathbf{T}_{j,:}\right).\mathbf{Q}_{i,:}^{\mathbf{T}} \end{pmatrix}^2 + \frac{\lambda}{2}\left(||\mathbf{H}||^2 + ||\mathbf{Q}||^2 + ||\mathbf{T}||^2 + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2\right) \tag{7}$$

c: PROBABILISTIC BIAS-SVD

Also with implicit feedback data, recently, two models have been proposed with good results, Probabilistic Bias-SVD [39] and Multi Bias-SVD [26]. With the assumption that the latent factors in the model follow a normal distribution, Probabilistic Bias-SVD [39] is built on the Maximum A Posteriori, as follows:

$$\hat{R}_{u,i} \approx g\left(o_u + p_i + \mu + \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\right) \tag{8}$$

$$\underset{\substack{\mathbf{H},\mathbf{Q},\\ o_u\, u=1\ldots m\\ p_i\, i=1\ldots n}}{minimize}\ J^{(ProbBiasSVD)}$$
$$J^{(ProbBiasSVD)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}} \begin{pmatrix} \left(R_{u,i} - g\left(o_u + p_i + \mu + \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\right)\right)^2 \\ +R'_{u,i}.\ln g\left(o_u + p_i + \mu + \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\right) \end{pmatrix} + \frac{\lambda}{2}\left(||\mathbf{H}||^2 + ||\mathbf{Q}||^2 + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2\right) \tag{9}$$

where $R'_{u,i}$ indicates whether a user u interacts with an item i or not; g is a logistic function.

d: MULTI BIAS-SVD

Contrary to Bias-SVD++ and Probabilistic Bias-SVD, Multi Bias-SVD [26] learns latent factor vectors of users and items through a multi-step decision-making process. The result of one step is enriched by the result of the previous step. Specifically, the first is that a user u decides whether to interact with an item i or not. It depends on the matching of the s-dimensional initial latent factor vector of user u, denoted by $\mathbf{H}^{(0)}_{u,:}$, and the s-dimensional initial latent factor vector of item i, denoted by $\mathbf{Q}^{(0)}_{i,:}$. Based on implicit feedback data $R'_{u,i}$, they are learned as follows:

$$\underset{\mathbf{H}^{(0)},\mathbf{Q}^{(0)}}{minimize}\ J^{(MultiBiasSVD\_0)}$$
$$J^{(MultiBiasSVD\_0)} = \frac{1}{2}\underset{\mathbf{H}^{(0)},\mathbf{Q}^{(0)}}{minimize}.\sum\nolimits_{(u,i)\in\mathbb{R}'}\left(R'_{u,i} - \mathbf{H}^{(0)}_{u,:}.\mathbf{Q}^{(0)\mathbf{T}}_{i,:}\right)^2$$
$$\text{Subject to } \mathbf{H}^{(0)} \geq 0 \text{ and } \mathbf{Q}^{(0)} \geq 0 \tag{10}$$

where $\mathbb{R}' = \{(u,i)|u = 1\ldots m \wedge i = 1\ldots n \wedge R'_{u,i} \neq *\}$ are the user-item combinations at which implicit feedback is observed.

During the item experience, the s-dimensional initial latent factor vectors, i.e., $\mathbf{H}^{(0)}_{u,:}$ and $\mathbf{Q}^{(0)}_{i,:}$, will evolve into the s-dimensional comprehensive latent factor vectors, i.e., $\mathbf{H}_{u,:}$ and $\mathbf{Q}_{i,:}$. The authors model this process as follows:

$$\mathbf{H}_{u,:} = \mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:} \tag{11}$$
$$\mathbf{Q}_{i,:} = \mathbf{Y}_{i,:} + \mathbf{Q}^{(0)}_{i,:} \tag{12}$$

where $\mathbf{D}_{u,:}$ and $\mathbf{Y}_{i,:}$ are the additions to user and item representations after transforming from the initial latent factor space to the comprehensive latent factor space, respectively.

In the final step, the match between the user comprehensive latent factor vector $\mathbf{H}_{u,:}$ and the item comprehensive latent factor vector $\mathbf{Q}_{i,:}$ determines the rating of the user for the item. The matrices $\mathbf{Y}$ and $\mathbf{D}$ are estimated so that the observed ratings are closest to their predicted ratings, as follows:

$$\hat{R}_{u,i} \approx o_u + p_i + \mu + \mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:}$$
$$= o_u + p_i + \mu + \left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right).\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(0)}_{i,:}\right)^{\mathbf{T}} \tag{13}$$

$$\underset{\substack{\mathbf{Y},\mathbf{D},\\ o_u\ u=1\ldots m\\ p_i\ i=1\ldots n}}{minimize}\ J^{(MultiBiasSVD\_1)}$$
$$J^{(MultiBiasSVD\_1)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}}\left(R_{u,i} - \hat{R}_{u,i}\right)^2$$
$$\Leftrightarrow$$
$$J^{(MultiBiasSVD\_1)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}}\begin{pmatrix} R_{u,i} - o_u - p_i - \mu \\ -\left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right).\left(.\mathbf{Y}_{i,:} + \mathbf{Q}^{(0)}_{i,:}\right)^{\mathbf{T}}\end{pmatrix}^2 + \frac{\lambda}{2}\left(||\mathbf{D}||^2 + ||\mathbf{Y}||^2 + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2\right) \tag{14}$$

e: TOPIC-MF

After experiencing an item, besides a rating, users can write a review for the item. Therefore, there are two types of descriptive data for each observed user-item combination: rating and review. In this context, latent factor models need to learn the user latent factor vectors and the item latent factor vectors so that they are appropriate for both types of data. With such a goal in mind, Topic-MF [40] uses a topic modeling Non-negative Matrix Factorization (NMF) to learn latent topics in the review set. This process is approximating the review-word matrix $\mathbf{X}$ into a review-topic matrix $\mathbf{F}$ and a word-topic matrix $\mathbf{M}$, i.e., the minimization of $||\mathbf{X} - \mathbf{F}.\mathbf{M}^{\mathbf{T}}||^2$. This approximation is integrated into the objective function as follows:

$$\hat{R}_{u,i} \approx o_u + p_i + \mu + \mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:} \tag{15}$$

$$\underset{\substack{\mathbf{H},\mathbf{Q},\mathbf{F},\mathbf{M}\\ o_u\ u=1\ldots m\\ p_i\ i=1\ldots n}}{minimize}\ J^{(TopicMF)}$$
$$J^{(TopicMF)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}}\begin{pmatrix} R_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}^2 + ||\mathbf{X} - \mathbf{F}.\mathbf{M}^{\mathbf{T}}||^2 + \frac{\lambda}{2}\begin{pmatrix} ||\mathbf{F}||^2 + ||\mathbf{M}||^2 + ||\mathbf{H}||^2 \\ +||\mathbf{Q}||^2 + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2\end{pmatrix}$$
$$\text{Subject to } \mathbf{F} \geq \mathbf{0} \text{ and } \mathbf{M} \geq \mathbf{0} \tag{16}$$

In the objective function Eq. (16), the authors also define a connection function between latent factors and latent topics as follows:

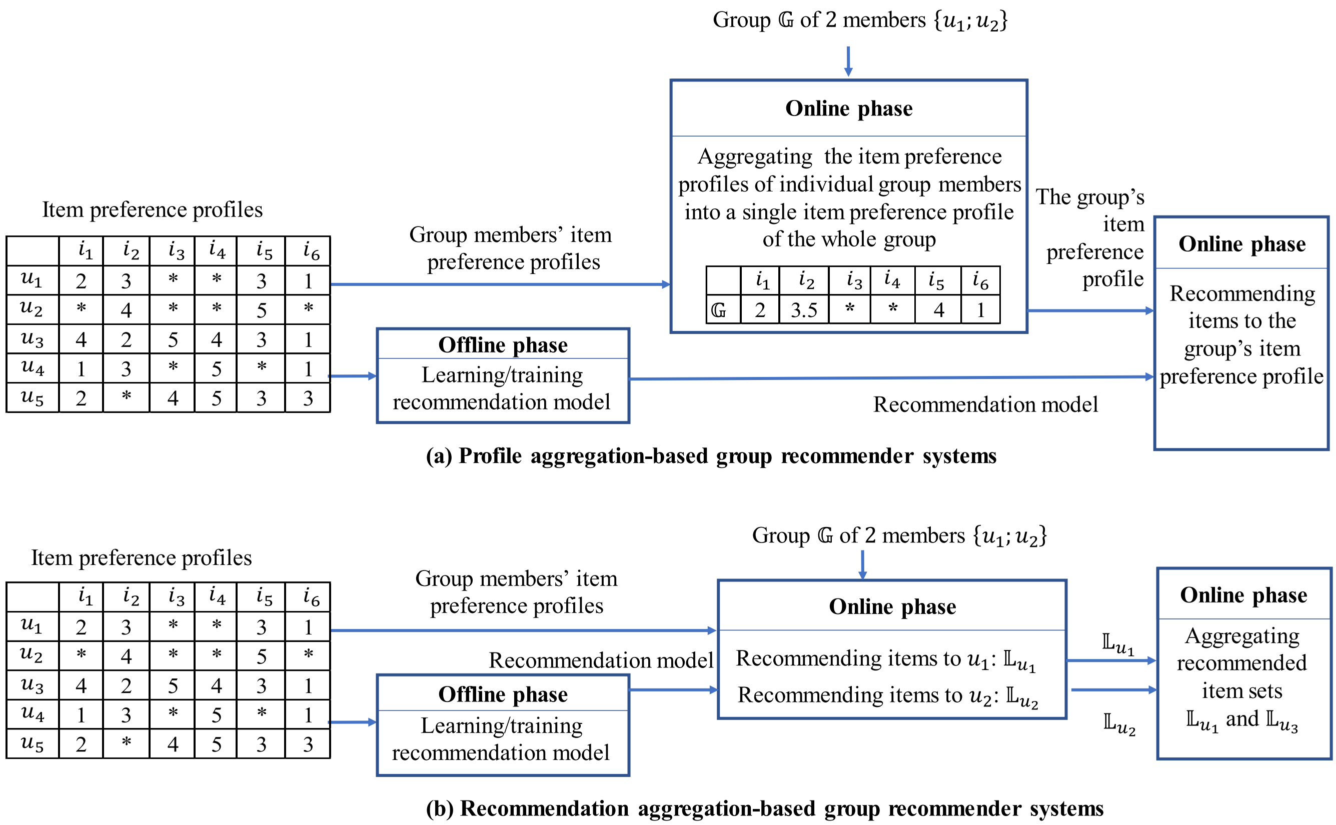
$$F_{u,i,j} = \frac{exp(k_1.\left|H_{u,j}\right| + k_2.\left|Q_{i,j}\right|)}{\sum_{j'=1\ldots s} exp(k_1.\left|H_{u,j'}\right| + k_2.\left|Q_{i,j'}\right|)} \tag{17}$$

f: SBMF

The authors in SBMF [27] have built a dictionary to extract users' sentiments for the items on their textual reviews. The authors then express these sentiments with scores. Therefore, latent factor vectors are learned to fit the observed rating scores $R_{u,i}$ and observed sentiment scores $C_{u,i}$, as follows:

$$\hat{R}_{u,i} \approx o_u + p_i + \mu + \mathbf{H}_{u,:}.\mathbf{Q}^{T}_{i,:} \tag{18}$$

$$\underset{\substack{\mathbf{H},\mathbf{Q}\\ o_u\ u=1\ldots m\\ p_i\ i=1\ldots n}}{minimize}\ J^{(SBMF)}$$
$$J^{(SBMF)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}}\left(R_{u,i} - \hat{R}_{u,i}\right)^2$$
$$\Leftrightarrow$$
$$J^{(SBMF)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}}\begin{pmatrix}\begin{pmatrix} R_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}^2 \\ +\begin{pmatrix} C_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}^2\end{pmatrix} + \frac{\lambda}{2}\left(||\mathbf{H}||^2 + ||\mathbf{Q}||^2 + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2\right) \tag{19}$$



	i_1	i_2	i_3	i_4	i_5	i_6
u_1	2	3	*	*	3	1
u_2	*	4	*	*	5	*
u_3	4	2	5	4	3	1
u_4	1	3	*	5	*	1
u_5	2	*	4	5	3	3

	i_1	i_2	i_3	i_4	i_5	i_6
𝔾	2	3.5	*	*	4	1

FIGURE 2. Two group recommendation methods: profile aggregation and recommendation aggregation.

C. GROUP RECOMMENDER SYSTEMS

The biggest challenge of a group recommender system is to resolve conflicts of interest among group members in order to predict the most accurate rating of a group for a target item. This is done by two methods: profile aggregation and recommendation aggregation. Fig. 2 shows the differences between these two methods.

Specifically, in the profile aggregation, the whole group is represented by a single item preference profile thanks to the aggregation of the item preference profiles of the individual group members, which are the observed ratings. A group's rating prediction for a target item now becomes its item preference profile's rating prediction, which is made by single-user recommendation algorithms [28]–[31]. Completely different from the profile aggregation, the recommendation aggregation recommends the items for each group member independently. The aggregation of the group members' recommended item sets produce the group's recommended item set [12], [41]. The aggregation functions are classified into three categories as follows:

- Majority-based: The opinion of group members in the minority will be ignored in aggregate functions [30];
- Consensus-based: This is a kind of aggregation that is closest to human beings. It means that to make decisions, groups always consult with all members with the desire to reach an agreement that pleases the whole group [42];
- Borderline: The functions of this type are only concerned with the opinions of key members [42]. For example, a group of family members is sometimes completely controlled by the parents; the members of a group tend to cede decision-making power to reputable members.

The profile aggregation proved to be more effective than the recommendation aggregation in terms of accuracy of the rating prediction [3], [29], [43]–[45]. For the profile aggregation, the sooner the aggregation is done, the fewer conflicts arise between the members [12], [29]. For the above reason, in this paper, we focus on profile aggregation-based group recommender systems. Some studies on the use of collaborative filtering for profile aggregation-based group recommender systems are presented below.

1) PROFILE AGGREGATION-BASED GROUP RECOMMENDER SYSTEMS USING MEMORY-BASED COLLABORATIVE FILTERING

The general structure of the implementation of memory-based collaborative filtering for a profile aggregation-based group recommender system is shown in Fig. 3.

As presented in subsection II.B, in the offline phase, memory-based collaborative filtering provides the preference similarity of each pair of users. In the online phase, the authors in [46] performed the aggregation of the group members' item preference profiles to create a single item

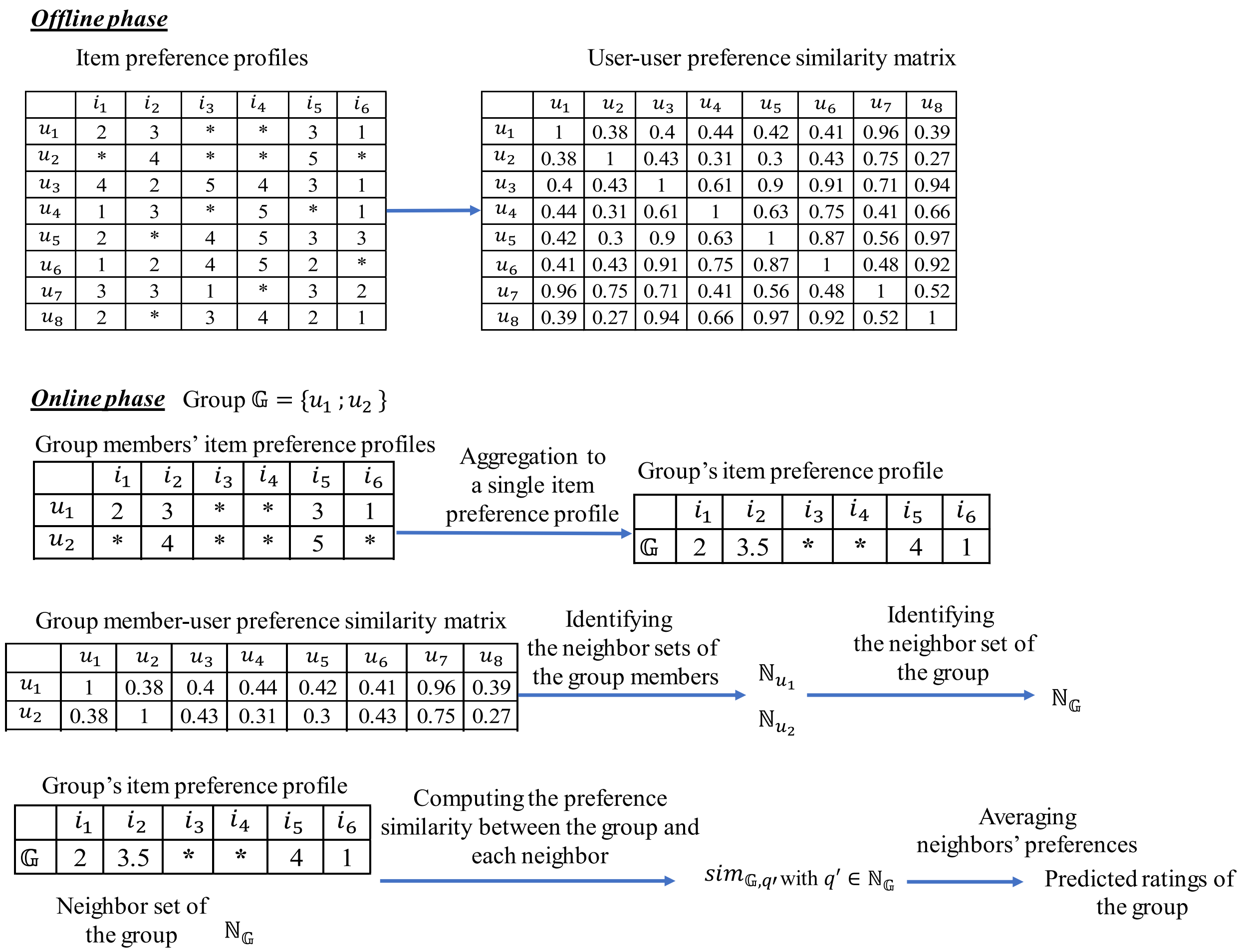


FIGURE 3. The general structure of a profile aggregation-based group recommender systems using memory-based collaborative filtering.

preference profile that reflects the preferences of the whole group. They proposed a definition of the neighbor set of the group's item preference profile using the neighbor sets of the group members. The preference similarity between the group's item preference profile and each neighbor is computed by Pearson measure. The average of neighbors' ratings with weights being similarities between neighbors and the group's item preference profile will help predict the unobserved rating of the group, similar to Eq. (1).

The studies in [45] and [46] are quite similar. The main difference between them lies in the fact that the latter builds an objective function based on the group members' item preference profiles to estimate the group member's weights. As a result, the average function of the group members' item preference profiles is integrated with the group members' weights.

Also, according to the above structure, the authors in [28] proposed a formula to calculate the similarity between the group's item preference profile and each neighbor user. It is the aggregation of the SVD-based similarities of the items that both of them have rated.

2) PROFILE AGGREGATION-BASED GROUP RECOMMENDER SYSTEMS USING LATENT FACTOR MODELS

In [47], the latent factor model was first applied to recommend a group's item preference profile in a profile aggregation-based group recommender system. Based on it, many studies have proposed to further improve the performance of profile aggregation-based group recommendations. Some typical studies are [29], [48], [49]. The general structure of these studies is shown in Fig. 4. Their details are as follows:

- *Learning user-factor and item-factor matrices based on observed ratings through an objective function*: The study [47] uses the objective function without bias of users and bias of items. In [29], [48], and [49], a more advanced objective function is used, Bias-SVD as presented Eq. (4-5).
- *Aggregating group members' item preference profiles to create a single item preference profile of the whole group*: The authors in 47] and [49] aggregated group members' item preference profiles using a weighted average function. In this function, each weight represents the influence of a group member. To ensure the

IEEE Access

Offline phase

Item preference profiles

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	2	3	*	*	3	1
u_2	*	4	*	*	5	*
u_3	4	2	5	4	3	1
u_4	1	3	*	5	*	1
u_5	2	*	4	5	3	3

Learning latent factor matrices →

User-factor matrix

	f_1	f_2	*user bias* (o)
u_1	-0.62	0.83	0.16
u_2	0.44	0.02	-0.41
u_3	0.39	0.27	-0.52
u_4	-0.16	0.13	0.79
u_5	0.94	-0.04	0.24

Item-factor matrix

	f_1	f_2	*item bias* (p)
i_1	0.61	0.30	0.80
i_2	0.96	-0.95	0.69
i_3	0.90	0.19	0.33
i_4	-0.23	0.87	0.26
i_5	0.39	-0.10	0.57
i_6	0.66	0.52	0.97

Online phase Group $\mathbb{G} = \{u_1 ; u_2\}$, target item i_3

Group members' item preference profiles

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	2	3	*	*	3	1
u_2	*	4	*	*	5	*

Aggregation to a single item preference profile →

Group's item preference profile

	i_1	i_2	i_3	i_4	i_5	i_6
$\mathbb{G}$	2	3.5	*	*	4	1

Group's item preference profile

	i_1	i_2	i_3	i_4	i_5	i_6
$\mathbb{G}$	2	3.5	*	*	4	1

Projecting the group's profile into the latent factor space →

Group-factor vector

	f_1	f_2	*bias* (o)
$\mathbb{G}$	0.34	-0.01	0.73

Group-factor vector

	f_1	f_2	*bias* (o)
$\mathbb{G}$	0.34	-0.01	0.73

Target item-factor vector

	f_1	f_2	*item bias* (p)
i_3	0.90	0.19	0.33

Dot product of the group-factor vector and the item-factor vector →

$\widehat{R}_{\mathbb{G},i_3} = 4.2$

FIGURE 4. The general structure of a profile aggregation-based group recommender systems using latent factor models.

high flexibility of the system, it can be calculated with the number of observed ratings of the corresponding group member.

The authors in [48] have stated that the profile aggregation should not be considered independent of the selected recommendation algorithm. Thus, they leverage the results of the selected recommendation algorithm, which is done offline, to enrich the profile aggregation, which is done online. Specifically, for a particular item, they proposed profile aggregation that would work not only on its ratings observed by group members but also from its filled ratings of the remaining members. The latent factor matrices obtained in the recommendation algorithm make it easy to fill in the unobserved ratings of a group member for an item at this phase. With this idea, the profile aggregation achieves a higher consensus by relying on all members instead of just a few members with observed ratings for the item.

In order to further improve profile aggregation, the study [29] has proposed the idea of adding each member's neighbor users to the group. The goal is to leverage neighbor members to further clarify the preferences of the original members in the group's profile.

- *Projecting the group's profile into the latent factor space*: After obtaining a single item preference profile

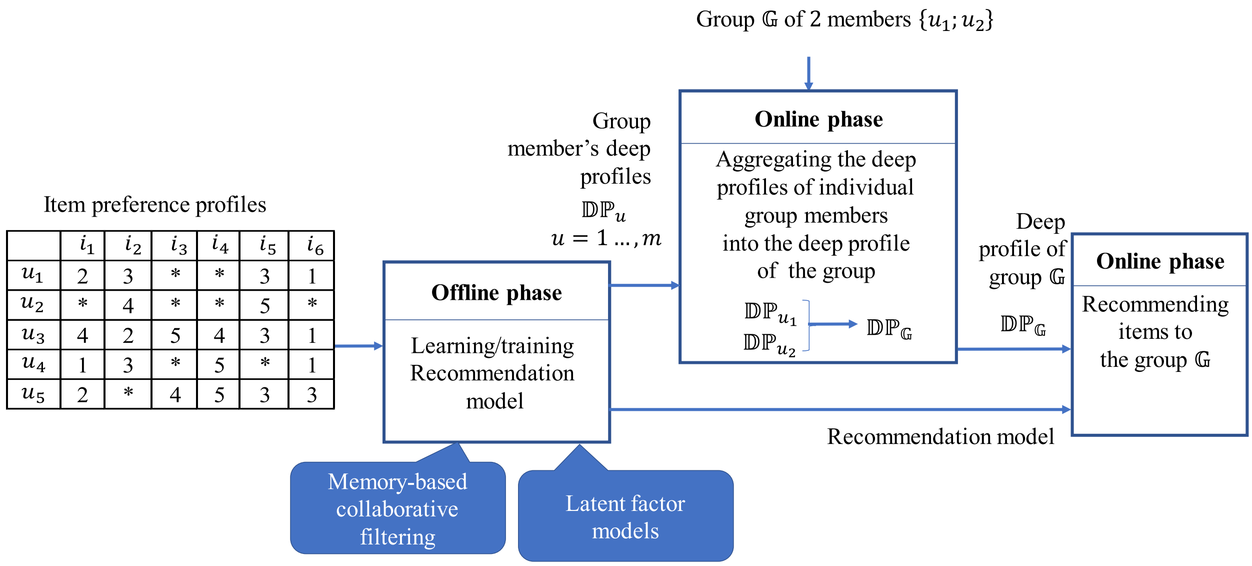


FIGURE 5. Deep profile aggregation-based group recommender systems.

Offline phase

Item preference profiles

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	2	3	*	*	3	1
u_2	*	4	*	*	5	*
u_3	4	2	5	4	3	1
u_4	1	3	*	5	*	1
u_5	2	*	4	5	3	3
u_6	1	2	4	5	2	*
u_7	3	3	1	*	3	2
u_8	2	*	3	4	2	1

User-user preference similarity matrix

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
u_1	1	0.38	0.4	0.44	0.42	0.41	0.96	0.39
u_2	0.38	1	0.43	0.31	0.3	0.43	0.75	0.27
u_3	0.4	0.43	1	0.61	0.9	0.91	0.71	0.94
u_4	0.44	0.31	0.61	1	0.63	0.75	0.41	0.66
u_5	0.42	0.3	0.9	0.63	1	0.87	0.56	0.97
u_6	0.41	0.43	0.91	0.75	0.87	1	0.48	0.92
u_7	0.96	0.75	0.71	0.41	0.56	0.48	1	0.52
u_8	0.39	0.27	0.94	0.66	0.97	0.92	0.52	1

Weight

	$w^{(MCF)}$
u_1	0.49
u_2	0.41
u_3	0.70
u_4	0.54
u_5	0.66
u_6	0.68
u_7	0.63
u_8	0.67

Deep Profile: $\mathbb{DP}_u^{(MCF)}\ u = 1 \dots m$

FIGURE 6. Deep profiles in the memory-based collaborative filtering.

of the whole group, a projection operation of this profile into the latent factor space is performed to learn both the latent factor vector and the bias of the group. This process is done by optimizing the distance between the ratings that are aggregated in the group's profile and their predicted ratings. In [29] and [47]–[49], Bridge Regression is used to solve this optimization.

- *Predicting unknown ratings of the group for items that all group members have not experienced:* Similar to regular users, the group's unknown rating for an item is predicted by the dot product of the group latent factor vector and the item latent factor vector plus bias of the group and bias of the item [29], [47]–[49].

III. MOTIVATION

It can be seen from the literature review in section II that the biggest disadvantage of a profile aggregation-based group recommender system is computing similarities between the group's item preference profile and every neighbor user if memory-based collaborative filtering is selected, and projecting the group' item preference profile into the latent factor space if latent factor model is selected. Remember that all of this processing is done in the online phase, so incurring large computational costs will seriously affect the user experience. This is the main reason why many recent advanced latent factor models and preference similarity measures have not been used for group recommendations even though they produce very promising results in single-user recommender systems.

In addition, working on users' item preference profiles, which are very sparse, makes it difficult to reach consensus among group members on the recommendation results. As presented in subsection II.C, some recent studies like [48], which uses filled ratings to support observed ratings, and [29], which uses neighbor members besides the original members, have successfully solved this problem, thereby improving the

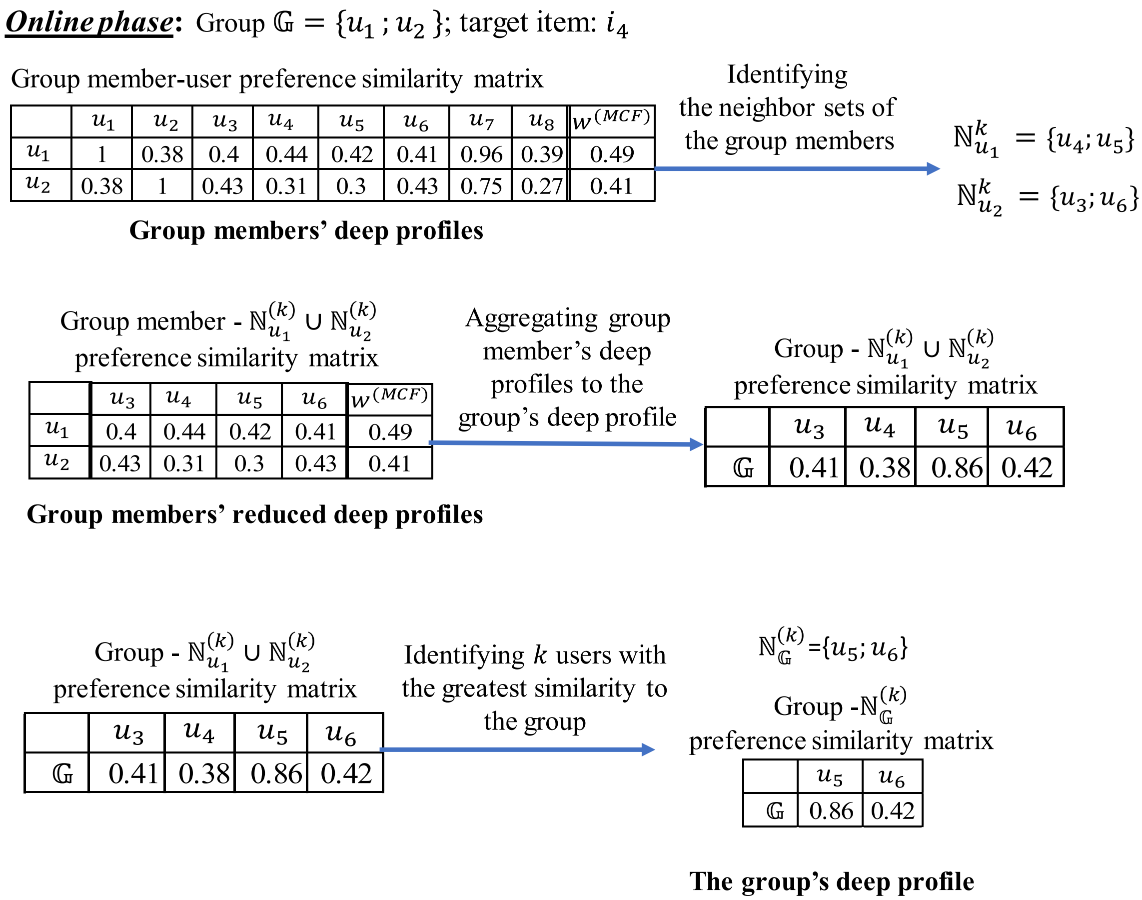


FIGURE 7. Aggregating members' deep profiles to the group's deep profile in the memory-based collaborative filtering.

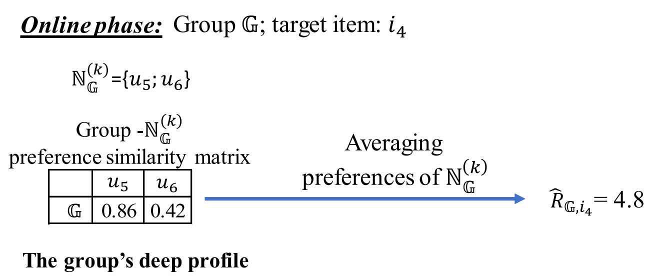


FIGURE 8. Predicting rating of a group for an item in the memory-based collaborative filtering.

performance of group recommendations. However, they incur additional computational costs for the online phase.

In this paper, we aim to propose a more radical method to solve the above problems of profile aggregation-based group recommender systems. Our idea is to perform the aggregation of the so-called deep profiles instead of the item preference profiles as in previous studies. The term *deep profile* refers to the users' profiles that lie deep within the selected recommendation algorithm performed in the offline phase. This study interprets users' deep profiles and aggregates them according to each type of collaborative filtering. Fig. 5 illustrates the difference of our proposal compared with previous studies. Details will be presented in section IV.

IV. DEEP PROFILE AGGREGATION-BASED GROUP RECOMMENDER SYSTEMS

A. DEEP PROFILE AGGREGATION-BASED GROUP RECOMMENDER SYSTEMS USING MEMORY-BASED COLLABORATIVE FILTERING

In this subsection, we present our proposed **D**eep **P**rofile aggregation-based group recommendation method using **M**emory-based **C**ollaborative **F**iltering named **DP_MCF**. In case the chosen recommendation algorithm is memory-based collaborative filtering, we will in turn define the corresponding deep profiles, aggregate them into the deep profile of the group, and finally, predict the rating based on the group's deep profile.

1) DEEP PROFILE DEFINITION

As shown above, in the offline phase, memory-based collaborative filtering algorithms provide the similarity between each pair of users: $sim_{u,q}$ where $u = 1 \ldots m$ and $q = 1 \ldots m$. These are **D**eep **P**rofile of each user in the **M**emory-based **C**ollaborative **F**iltering, denoted by $\mathbb{DP}_u^{(MCF)} u = 1 \ldots m$, as follows:

$$\mathbb{DP}_u^{(MCF)} = \{sim_{u,q} | q = 1 \ldots m\} \tag{20}$$

For a weighted aggregation on such deep profiles, we define the users' weights in the deep profiles, denoted by

Offline phase

Item preference profiles

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	1	3	*	*	5	1
u_2	*	4	*	*	2	*
u_3	1	1	3	5	3	4
u_4	2	2	*	3	*	4
u_5	1	*	3	4	5	1

Learning latent factor matrices

Item-factor matrix

	f_1	f_2	$item\ bias\ (p)$
i_1	-0.56	0.49	0.50
i_2	0.11	0.13	0.43
i_3	0.44	0.80	-0.56
i_4	0.49	-0.83	0.28
i_5	-0.57	0.45	0.38
i_6	-0.05	0.59	-0.64

User-factor matrix

	f_1	f_2	$user\ bias\ (o)$	$w^{(LFM)}$
u_1	0.86	0.26	0.07	0.60
u_2	-0.81	0.61	-0.88	0.04
u_3	0.18	0.47	0.11	0.03
u_4	0.31	-0.67	-0.07	0.19
u_5	0.09	-0.30	0.42	0.15

Deep profile: $\mathbb{DP}_u^{(LFM)}\ u = 1 \ldots m$

FIGURE 9. **Deep profiles in the latent factor models.**

$w_u^{(MCF)} u = 1 \ldots m$. For this work, we interpret that a user is influential when he/she contributes a lot to the prediction of the preferences of other users. Therefore, a user's weight can be the average of his/her similarities with other users, as follows:

$$w_u^{(MCF)} = \frac{\sum_{y \in \mathbb{DP}_u^{(MCF)}} y}{|\mathbb{DP}_u^{(MCF)}|} = \frac{\sum_{\{q|q=1\ldots m\}} sim_{u,q}}{m} \tag{21}$$

A larger value means that this user is more likely to appear in the neighbor set of other users and thereby participate in the process of determining their unknown ratings by memory-based collaborative filtering. Fig. 6 shows an example of building users' deep profiles in memory-based collaborative filtering.

2) AGGREGATING GROUP MEMBERS' DEEP PROFILES TO THE GROUP'S DEEP PROFILE

As presented about memory-based collaborative filtering, to predict the rating of a group $\mathbb{G}$ for a target item, the deep profile of the group $\mathbb{G}$ must contain the similarity between $\mathbb{G}$ and every one of its neighbors, as follows:

$$\mathbb{DP}_{\mathbb{G}}^{(MCF)} = \{sim_{\mathbb{G},q} | q \in \mathbb{N}_{\mathbb{G}}^{(k)}\} \tag{22}$$

where $\mathbb{N}_{\mathbb{G}}^{(k)}$ is the set of k neighbor users of $\mathbb{G}$. In this section, we present the weighted aggregation of deep profiles of group members, i.e., $\mathbb{DP}_u^{(MCF)} u \in \mathbb{G}$, into the deep profile of the whole group, i.e., $\mathbb{DP}_{\mathbb{G}}^{(MCF)}$. Regarding the profile aggregation function used for group recommendation, the weighted average function has always proved to be highly effective in previous studies [3], [12], [45], [49], [50]. Based on this observation, we apply it to the aggregation of group members' deep profiles in this study.

First, each group member $u \in \mathbb{G}$ determines the set of k neighbors most similar to him/her that have rated the target item, i.e., $\mathbb{N}_u^{(k)}$. These are the users who are most likely to support the group's rating prediction. As in previous studies, we consider them as neighbor candidates of the group $\mathbb{G}$, i.e, $\bigcup_{u \in \mathbb{G}} \mathbb{N}_u^{(k)}$. Next, in these candidates, k users with the greatest similarities to the group $\mathbb{G}$ will form the set $\mathbb{N}_{\mathbb{G}}^{(k)}$ in $\mathbb{DP}_{\mathbb{G}}^{(MCF)}$. The similarity between a user q and the group $\mathbb{G}$, i.e., $\mathbb{DP}_{\mathbb{G}}^{(MCF)}$, will come from a weighted aggregation of the similarities between q and the group members, i.e., $\mathbb{DP}_{\mathbb{G}}^{(MCF)} u \in \mathbb{G}$, as follows:

$$\mathbb{DP}_{\mathbb{G}}^{(MCF)} = \frac{\sum_{u \in \mathbb{G}} \left(w_u^{(MCF)} . \mathbb{DP}_u^{(MCF)} \right)}{\sum_{u \in \mathbb{G}} w_u^{(MCF)}}$$

$$\Leftrightarrow$$

$$sim_{p,\mathbb{G}} = \frac{\sum_{u \in \mathbb{G}} \left(w_u^{(MCF)} . sim_{u,p} \right)}{\sum_{u \in \mathbb{G}} w_u^{(MCF)}} \tag{23}$$

Our goal of Eq. (23) is that the more similar a user is to influential group members, the more similar he/she is to the whole group. Obviously, with Eq. (23), the computation of $sim_{p,\mathbb{G}}$ only needs to browse the deep profiles of all group members instead of browsing all items as in previous studies. This greatly reduces the processing cost for the online phase because the number of members of a group is always much smaller than the number of items in the system.

The above processing is illustrated in Fig. 7 in which the group consists of two members, u_1 and u_2. For a target item i_4, $\mathbb{N}_{u_1}^{(k)}$ and $\mathbb{N}_{u_2}^{(k)}$ will be u_4; u_5 and u_3; u_6 respectively where $k = 2$. To find $\mathbb{N}_{\mathbb{G}}^{(k)}$ in $\mathbb{DP}_{\mathbb{G}}^{(MCF)}$, we just need to compute the similarities between $\mathbb{G}$ and each user in the set $\mathbb{N}_{u_1}^{(k)} \cup \mathbb{N}_{u_2}^{(k)} = \{u_3; u_4; u_5; u_6\}$. For example, according to Eq. (23), the similarity between $\mathbb{G}$ and u_3 is calculated as follows:

$$sim_{u_3,\mathbb{G}} = \frac{w_{u_1}.sim_{u_3,u_1} + w_{u_2}.sim_{u_3,u_2}}{w_{u_1} + w_{u_2}} = \frac{0,49.0,4 + 0,41.0,43}{0,49 + 0,41} = 0,41 \quad (24)$$

3) RATING PREDICTION

Based on the deep profile of group $\mathbb{G}$, the rating of $\mathbb{G}$ for the target item i can be predicted similarly to Eq. (1), as follows:

$$\hat{R}_{\mathbb{G},i} = \mu_{\mathbb{G}} + \frac{\sum_{q\in\mathbb{N}_{\mathbb{G}}^{(k)}} \left(sim_{q,\mathbb{G}}.(R_{q,i} - \mu_p)\right)}{\sum_{q\in\mathbb{N}_{\mathbb{G}}^{(k)}} sim_{q,\mathbb{G}}} \quad (25)$$

where $\mu_{\mathbb{G}}$ is the average of observed ratings given by group members. Continuing with the example in Fig. 7, Fig. 8 shows the prediction of the rating of $\mathbb{G}$ for the target item i_4 according to Eq. (25), as follows:

$$\hat{R}_{\mathbb{G},i_4} = \mu_{\mathbb{G}} + \frac{sim_{u_5,\mathbb{G}}.\left(R_{u_5,i_4} - \mu_{u_5}\right) + sim_{u_6,\mathbb{G}}.\left(R_{u_6,i_4} - \mu_{u_6}\right)}{sim_{u_5,\mathbb{G}} + sim_{u_6,\mathbb{G}}} = 3 + \frac{0,86.(5-3,4) + 0,42.(5-2,8)}{0,86+0,42} = 4,8 \quad (26)$$

B. DEEP PROFILE AGGREGATION-BASED GROUP RECOMMENDER SYSTEMS USING LATENT FACTOR MODEL

In this subsection, we propose **D**eep **P**rofile aggregation-based group recommendations in which the used latent factor models are **Bias-SVD** [37] (see Eq. (4-5)), **SBMF** [27] (see Eq. (18-19)), and **Multi Bias-SVD** [26] (see Eq. (10-14)). These are called **DP_BiasSVD**, **DP_SBMF**, and **DP_MultiBiasSVD**. Bias-SVD is a traditional latent factor model based solely on ratings, while Multi Bias-SVD and SBMF are two latent factor models that are integrated with implicit feedback and reviews, respectively.

1) DEEP PROFILE DEFINITION

As presented in subsection II.C, in the offline phase, the objective functions of the latent factor models help identify user latent factor vectors, i.e., $\mathbf{H}_{u,:}u = 1\ldots m$, item latent factor vectors, i.e., $\mathbf{Q}_{i,:}i = 1\ldots n$, user biases, i.e, $o_u\, u = 1\ldots m$, and item biases, i.e, $p_i\, i = 1\ldots n$. As illustrated in Fig. 9, the learned user latent factor vectors and user biases form the user **D**eep **P**rofiles in the **L**atent **F**actor **M**odels, denoted by $\mathbb{DP}_u^{(LFM)} u = 1\ldots m$:

$$\mathbb{DP}_u^{(LFM)} = \{\mathbf{H}_{u,:}; o_u\} \quad (27)$$

However, for a weighted aggregation of deep profiles in group recommendations, the objective functions of the latent factor models must be modified to learn the users' weights, denoted by $w_u^{(LFM)} u = 1\ldots m$. For a user's weight, we interpret it as the contribution of his/her observed ratings to the objective function. In addition, we normalize users' weights according to parameter β as follows:

$$\forall u = 1\ldots m : w_u^{(LFM)} \geq 0 \text{ and } \sum_{i=1}^{m} w_u^{(LFM)^\beta} = 1 \quad (28)$$

Specifically, the objective functions of Bias-SVD, Multi Bias-SVD, and SBMF presented in Eq. (5), Eq. (14), and Eq. (19) of subsection II.C will become as follows:

$$\underset{\substack{\mathbf{H},\mathbf{Q}\\ o_u\, u=1\ldots m\\ p_i\, i=1\ldots n\\ w_u^{(LFM)} u=1\ldots m}}{minimize} J^{(BiasSVD\_Weight)}$$

$$J^{(BiasSVD\_Weight)} = \frac{1}{2}.\sum_{(u,i)\in\mathbb{R}}\left(w_u^{(LFM)}.\begin{pmatrix} R_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right) + \frac{\lambda}{2}.\left(||\mathbf{H}||^2 + ||\mathbf{Q}||^2 + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2\right)$$

$$\text{subject to } \sum_{i=1}^{m} w_u^{(LFM)^\beta} = 1 \text{ and } \forall u = 1\ldots m : w_u^{(LFM)} \geq 0 \quad (29)$$

$$\underset{\substack{\mathbf{H},\mathbf{Q}\\ o_u\, u=1\ldots m\\ p_i\, i=1\ldots n\\ w_u^{(LFM)} u=1\ldots m}}{minimize} J^{(SBMF\_Weight)}$$

$$J^{(SBMF\_Weight)} = \frac{1}{2}.\sum_{(u,i)\in\mathbb{R}}\left(w_u^{(LFM)}.\left(\begin{pmatrix} R_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2 + \begin{pmatrix} C_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)\right) + \frac{\lambda}{2}.\left(||\mathbf{H}||^2 + ||\mathbf{Q}||^2 + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2\right)$$

$$\text{subject to } \sum_{i=1}^{m} w_u^{(LFM)^\beta} = 1 \text{ and } \forall u = 1\ldots m : w_u^{(LFM)} \geq 0 \quad (30)$$

$$\underset{\substack{\mathbf{D},\mathbf{Y}\\ o_u\, u=1\ldots m\\ p_i\, i=1\ldots n\\ w_u^{(LFM)} u=1\ldots m}}{minimize} J^{(MultiBiasSVD\_Weight)}$$

$$J^{(MultiBiasSVD\_Weight)} = \frac{1}{2}.\sum_{(u,i)\in\mathbb{R}}\left(w_u^{(LFM)} .\begin{pmatrix} R_{u,i} - o_u - p_i - \mu \\ -\left(\mathbf{D}_{u,:} + \mathbf{H}_{u,:}^{(0)}\right).\left(\mathbf{Y}_{i,:} + \mathbf{Q}_{i,:}^{(0)}\right)^{\mathbf{T}}\end{pmatrix}^2\right) + \frac{\lambda}{2}\left(||\mathbf{D}||^2 + ||\mathbf{Y}||^2 + \sum_{u=1}^{m} o_u^2 + \sum_{i=1}^{n} p_i^2\right)$$

$$\text{subject to } \sum_{i=1}^{m} w_u^{(LFM)^\beta} = 1 \text{ and } \forall u = 1\ldots m : w_u^{(LFM)} \geq 0 \quad (31)$$

For Eq. (29-31), we optimize them according to one variable while the other variables are treated as constant values. This process is repeated until convergence is reached. The details will be presented below.

a: THE OPTIMIZATION OF $J^{(BiasSVD\_Weight)}$

Keeping $\mathbf{H}$, $\mathbf{Q}$, $o_u\ u = 1\ldots m$, and $p_i\ i = 1\ldots n$ fixed, we perform the optimization of $J^{(BiasSVD\_Weight)}$ concerning the variables $w_u^{(LFM)} u = 1\ldots m$. A Lagrangian function is formed by pushing the equality constraint $\sum_{i=1}^{m} w_u^{(LFM)^\beta} = 1$ into $J^{(BiasSVD\_Weight)}$ with the Lagrange multiplier γ, as follows:

$$\underset{\substack{w_u\ u=1\ldots m\\ \gamma}}{minimize}\ J^{(BiasSVD\_Weight\_Lagrange)}$$

$$J^{(BiasSVD\_Weight\_Lagrange)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}}\left(w_u^{(LFM)}.\begin{pmatrix} R_{u,i}-o_u-p_i\\ -\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right) + \gamma.\left(\sum_{u=1}^{m} w_u^{(LFM)^\beta}-1\right) + \frac{\lambda}{2}.\left(||\mathbf{H}||^2+||\mathbf{Q}||^2+\sum_{u=1}^{m} o_u^2+p_i^2\right) \quad (32)$$

Therefore, the partial derivative of the Lagrangian function $J^{(BiasSVD\_Weight\_Lagrange)}$ with respect to variables $w_u^{(LFM)} u = 1\ldots m$ and the Lagrange multiplier γ is 0, as follows:

$$\begin{cases}\forall u = 1\ldots m : \dfrac{\partial J^{(BiasSVD\_Weight\_Lagrange)}}{\partial w_u^{(LFM)}} = 0\\ \dfrac{\partial J^{(BiasSVD\_Weight\_Lagrange)}}{\partial \gamma} = 0\end{cases}$$

$$\Leftrightarrow$$

$$\forall u = 1\ldots m : w_u^{(LFM)} = \frac{\left(\sum_{i\in\mathbb{I}_u}\begin{pmatrix}R_{u,i}-o_u-p_i\\ -\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)^{1/(\beta-1)}}{\left(\sum_{u'=1}^{m}\left(\sum_{i\in\mathbb{I}_{u'}}\begin{pmatrix}R_{u',i}-o_{u'}-p_i\\ -\mu-\mathbf{H}_{u',:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)^{\beta/(\beta-1)}\right)^{1/\beta}}$$

See Appendix 1 (33)

where $\mathbb{I}_u$ are the item that user u has rated.

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{Q}$, $o_u\ u = 1\ldots m$, and $p_i\ i = 1\ldots n$ fixed, we perform the optimization of $J^{(BiasSVD\_Weight)}$ concerning the variable $\mathbf{H}$. The following equation: his is equivalent to solving t

$$\forall u = 1\ldots m : \nabla_{\mathbf{H}_{u,:}} J^{(BiasSVD\_Weight)} = 0$$

$$\Leftrightarrow$$

$$\mathbf{H}_{u,:} = w_u^{(LFM)}.\sum\nolimits_{i\in\mathbb{I}_u}\left(\mathbf{Q}_{i,:}.\left(R_{u,i}-o_u-p_i-\mu\right)\right).\left(\lambda.\mathbf{I}+w_u^{(LFM)}.\sum\nolimits_{i\in\mathbb{I}_u}\left(\mathbf{Q}_{i,:}^{\mathbf{T}}.\mathbf{Q}_{i,:}\right)\right)^{-1}$$

See Appendix 2 (34)

where $\mathbf{I}$ is the identity matrix; $\mathbb{I}_u$ are the items rated by user u.

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{H}$, $o_u\ u = 1\ldots m$, and $p_i\ i = 1\ldots n$ fixed, we perform the optimization of $J^{(BiasSVD\_Weight)}$ concerning the variable $\mathbf{Q}$ as follows:

$$\forall i = 1\ldots m : \nabla_{\mathbf{Q}_{i,:}} J^{(BiasSVD\_Weight)} = 0$$

$$\Leftrightarrow$$

$$\mathbf{Q}_{i,:} = \sum\nolimits_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}_{u,:}.\left(R_{u,i}-o_u-p_i-\mu\right)\right).\left(\lambda.\mathbf{I}+\sum\nolimits_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}_{u,:}^{\mathbf{T}}.\mathbf{H}_{u,:}\right)\right)^{-1}$$

See Appendix 3 (35)

where $\mathbb{U}_i$ are the users who have rated item i.

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{H}$, $\mathbf{Q}$, and $p_i\ i = 1\ldots n$ fixed, we perform the optimization of $J^{(BiasSVD\_Weight)}$ concerning the variables $o_u\ u = 1\ldots m$ as follows:

$$\forall u = 1\ldots m : \frac{J^{(BiasSVD\_Weight)}}{\partial o_u} = 0$$

$$\Leftrightarrow$$

$$o_u = \frac{w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(R_{u,i}-p_i-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\right)}{w_u^{(LFM)}.|\mathbb{I}_u|+\lambda}$$

See Appendix 4 (36)

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{H}$, $\mathbf{Q}$, and $o_u\ u = 1\ldots m$ fixed, we perform the optimization of $J^{(BiasSVD\_Weight)}$ concerning the variables $p_i\ i = 1\ldots n$ as follows:

$$\forall i = 1\ldots n : \frac{J^{(BiasSVD\_Weight)}}{\partial p_i} = 0$$

$$\Leftrightarrow$$

$$p_i = \frac{\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\left(R_{u,i}-o_u-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\right)\right)}{\sum_{u\in\mathbb{U}_i} w_u^{(LFM)}+\lambda}$$

See Appendix 5 (37)

b: THE OPTIMIZATION OF $J^{(SBMF\_Weight)}$

Keeping $\mathbf{H}$, $\mathbf{Q}$, $o_u\ u = 1\ldots m$, and $p_i\ i = 1\ldots n$ fixed, similar to the optimization of $J^{(BiasSVD\_Weight)}$ concerning $w_u^{(LFM)} u = 1\ldots m$, that of $J^{(SBMF\_Weight)}$ is as follows:

$$\underset{\substack{w_u\ u=1\ldots m\\ \gamma}}{minimize}\ J^{(SBMF\_Weight\_Lagrange)}$$

$$J^{(SBMF\_Weight\_Lagrange)} = \frac{1}{2}.\sum\nolimits_{(u,i)\in\mathbb{R}\}}\left(w_u^{(LFM)}.\left(\begin{pmatrix}R_{u,i}-o_u-p_i\\ -\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2+\begin{pmatrix}C_{u,i}-o_u-p_i\\ -\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)\right) - \gamma.\left(\sum_{u=1}^{m} w_u^{(LFM)^\beta}-1\right) + \frac{\lambda}{2}.\left(||\mathbf{H}||^2+||\mathbf{Q}||^2+\sum_{u=1}^{m} o_u^2+\sum_{i=1}^{n} p_i^2\right)$$

$$\Leftrightarrow$$

$$\forall u = 1\ldots m : w_u^{(LFM)}$$

$$= \frac{\left(\sum_{i\in\mathbb{I}_u}\left(\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2+\begin{pmatrix}C_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)\right)^{1/(\beta-1)}}{\left(\sum_{u'=1}^{m}\left(\sum_{i\in\mathbb{I}_{u'}}\left(\begin{pmatrix}R_{u',i}-o_{u'}-p_i\\-\mu-\mathbf{H}_{u',:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2+\begin{pmatrix}C_{u',i}-o_{u'}-p_i\\-\mu-\mathbf{H}_{u',:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)\right)^{\beta/(\beta-1)}\right)^{1/\beta}} \tag{38}$$

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{Q}$, $o_u\, u = 1\ldots m$, and $p_i\, i = 1\ldots n$ fixed, the optimization of $J^{(SBMF\_Weight)}$ concerning the variable $\mathbf{H}$ is as follows:

$$\forall u = 1\ldots m: \nabla_{\mathbf{H}_{u,:}} J^{(SBMF\_Weight)} = 0$$

$$\Leftrightarrow$$

$$\mathbf{H}_{u,:} = w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\mathbf{Q}_{i,:}.\begin{pmatrix}R_{u,i}+C_{u,i}-2.o_u\\-2.p_i-2.\mu\end{pmatrix}\right).\left(\lambda.\mathbf{I}+\mathbf{2}.w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\mathbf{Q}_{i,:}^{\mathbf{T}}.\mathbf{Q}_{i,:}\right)\right)^{-\mathbf{1}} \tag{39}$$

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{H}$, $o_u\, u = 1\ldots m$, and $p_i\, i = 1\ldots n$ fixed, the optimization of $J^{(SBMF\_Weight)}$ concerning the variable $\mathbf{Q}$ is as follows:

$$\forall i = 1\ldots m: \nabla_{\mathbf{Q}_{i,:}} J^{(SBMF\_Weight)} = 0$$

$$\Leftrightarrow$$

$$\mathbf{Q}_{i,:} = \sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}_{u,:}.\begin{pmatrix}R_{u,i}+C_{u,i}-2.o_u\\-2.p_i-2.\mu\end{pmatrix}\right).\left(\lambda.\mathbf{I}+\mathbf{2}.\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}_{u,:}^{\mathbf{T}}.\mathbf{H}_{u,:}\right)\right)^{-\mathbf{1}} \tag{40}$$

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{H}$, $\mathbf{Q}$, and $p_i\, i = 1\ldots n$ fixed, the optimization of $J^{(SBMF\_Weight)}$ concerning the variables $o_u\, u = 1\ldots m$ is as follows:

$$\forall u = 1\ldots m: \frac{\partial J^{(SBMF\_Weight)}}{\partial o_u} = 0$$

$$\Leftrightarrow$$

$$o_u = \frac{w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\begin{pmatrix}R_{u,i}+C_{u,i}-2.p_i\\-2.\mu-2.\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}}{2.w_u^{(LFM)}.|\mathbb{I}_u|+\lambda} \tag{41}$$

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{H}$, $\mathbf{Q}$, and $o_u\, u = 1\ldots m$ fixed, the optimization of $J^{(SBMF\_Weight)}$ concerning the variable s $p_i\, i = 1\ldots n$ is as follows:

$$\forall i = 1\ldots n: \frac{\partial J^{(SBMF\_Weight)}}{\partial p_i} = 0$$

$$\Leftrightarrow$$

$$p_i = \frac{\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}\begin{pmatrix}R_{u,i}+C_{u,i}-2.o_u\\-2.\mu-2.\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}\right)}{2.\sum_{u\in\mathbb{U}_i} w_u^{(LFM)}+\lambda} \tag{42}$$

c: THE OPTIMIZATION OF $J^{(MultiBiasSVD\_Weight)}$

Keeping $\mathbf{D}$, $\mathbf{Y}$, $o_u\, u = 1\ldots m$, and $p_i\, i = 1\ldots n$ fixed, the optimization of $J^{(MultiSVD\_Weight)}$ concerning $w_u^{(LFM)} u = 1\ldots m$ is as follows:

$$\underset{\substack{w_u\, u=1\ldots m\\ \gamma}}{minimize}\; J^{(MultiBiasSVD\_Weight\_Lagrange)}$$

$$J^{(MultiBiasSVD\_Weight\_Lagrange)} = \frac{1}{2}.\sum_{(u,i)\in\mathbb{R}} \times\left(w_u^{(LFM)}.\begin{pmatrix}R_{u,i}-o_u-p_i-\mu\\-\left(\mathbf{D}_{u,:}+\mathbf{H}_{u,:}^{(0)}\right).\left(\mathbf{Y}_{i,:}+\mathbf{Q}_{i,:}^{(0)}\right)^{\mathbf{T}}\end{pmatrix}^2\right) - \gamma.\left(\sum_{u=1}^{m} w_u^{\beta}-1\right) + \frac{\lambda}{2}\left(||\mathbf{D}||^2+||\mathbf{Y}||^2+\sum_{u=1}^{m}o_u^2+\sum_{i=1}^{n}p_i^2\right)$$

$$\Leftrightarrow$$

$$\forall u = 1\ldots m: w_u^{(LFM)} = \frac{\left(\sum_{i\in\mathbb{I}_u}\begin{pmatrix}R_{u,i}-o_u-p_i-\mu\\-\left(\mathbf{D}_{u,:}+\mathbf{H}_{u,:}^{(0)}\right).\left(\mathbf{Y}_{i,:}+\mathbf{Q}_{i,:}^{(0)}\right)^{\mathbf{T}}\end{pmatrix}^2\right)^{1/(\beta-1)}}{\left(\sum_{u'=1}^{m}\left(\sum_{i\in\mathbb{I}_{u'}}\begin{pmatrix}R_{u',i}-o_{u'}-p_i-\mu\\-\left(\mathbf{D}_{u',:}+\mathbf{H}_{u',:}^{(0)}\right)\\.\left(\mathbf{Y}_{i,:}+\mathbf{Q}_{i,:}^{(0)}\right)^{\mathbf{T}}\end{pmatrix}^2\right)^{\beta/(\beta-1)}\right)^{1/\beta}} \tag{43}$$

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{Y}$, $o_u\, u = 1\ldots m$, and $p_i\, i = 1\ldots n$ fixed, the optimization concerning the variable $\mathbf{D}$ is as follows:

$$\forall u = 1\ldots m: \nabla_{\mathbf{D}_{u,:}} J^{(MultiBiasSVD\_\_Weight)} = 0$$

$$\Leftrightarrow$$

$$\mathbf{D}_{u,:} = w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\begin{pmatrix}R_{u,i}-o_u-p_i-\mu\\-\mathbf{H}_{u,:}^{(0)}.\left(\mathbf{Y}_{i,:}+\mathbf{Q}_{i,:}^{(0)}\right)^{\mathbf{T}}\end{pmatrix}.\left(\mathbf{Y}_{i,:}+\mathbf{Q}_{i,:}^{(\mathbf{0})}\right)\right).\left(\lambda.\mathbf{I}+w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\left(\mathbf{Y}_{i,:}+\mathbf{Q}_{i,:}^{(\mathbf{0})}\right)^{\mathrm{T}}.\left(\mathbf{Y}_{i,:}+\mathbf{Q}_{i,:}^{(0)}\right)\right)\right)^{-1} \tag{44}$$

Keeping $w_u^{(LFM)} u = 1\ldots m$, $\mathbf{D}$, $o_u\, u = 1\ldots m$, and $p_i\, i = 1\ldots n$ fixed, the optimization concerning the variable $\mathbf{Y}$ is as follows:

$$\forall i = 1\ldots m: \nabla_{\mathbf{Y}_{i,:}} J^{(MultiBiasSVD\_Weight)} = 0$$

$$\Leftrightarrow$$

$$\mathbf{Y}_{i,:} = \sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\begin{pmatrix}R_{u,i}-o_u-p_i-\mu\\-\mathbf{Q}_{u,:}^{(0)}.\left(\mathbf{D}_{u,:}+\mathbf{H}_{u,:}^{(0)}\right)^{\mathbf{T}}\end{pmatrix}.\left(\mathbf{D}_{u,:}+\mathbf{H}_{u,:}^{(0)}\right)\right).\left(\lambda.\mathbf{I}+\sum_{u\in\mathbb{U}_i} w_u^{(LFM)}.\left(\mathbf{D}_{u,:}+\mathbf{H}_{u,:}^{(0)}\right)^{\mathbf{T}}.\left(\mathbf{D}_{u,:}+\mathbf{H}_{u,:}^{(0)}\right)\right)^{-1} \tag{45}$$

Keep $w_u^{(LFM)} u = 1 \ldots m$, $\mathbf{D}$, $\mathbf{Y}$, and $p_i \; i = 1 \ldots n$ fixed, the optimization concerning the variable $o_u \; u = 1 \ldots m$ is as follows:

$$\forall u = 1 \ldots m : \frac{J^{(MultiBiasSVD\_Weight)}(o_u)}{\partial o_u} = 0$$

$$\Leftrightarrow$$

$$o_u = \frac{w_u^{(LFM)} \cdot \sum_{i \in \mathbb{I}_u} \left(\begin{array}{c} R_{u,i} - p_i - \mu \\ - \left(\mathbf{D}_{u,:} + \mathbf{H}_{u,:}^{(0)}\right) \cdot \left(\mathbf{Y}_{i,:} + \mathbf{Q}_{i,:}^{(0)}\right)^{\mathbf{T}} \end{array} \right)}{w_u^{(LFM)} \cdot |\mathbb{I}_u| + \lambda} \tag{46}$$

Keep $w_u^{(LFM)} u = 1 \ldots m$, $\mathbf{H}$, $\mathbf{Q}$, and $o_u \; u = 1 \ldots m$ fixed, the optimization with respect to the variable $p_i \; i = 1 \ldots n$ is as follows:

$$\forall i = 1 \ldots n : \frac{J^{(MultiBiasSVD\_Weight)}(p_i)}{\partial p_i} = 0$$

$$\Leftrightarrow$$

$$p_i = \frac{\sum_{u \in \mathbb{U}_i} \left(w_u^{(LFM)} \cdot \left(\begin{array}{c} R_{u,i} - o_u - \mu \\ - \left(\mathbf{D}_{u,:} + \mathbf{H}_{u,:}^{(0)}\right) \cdot \left(\mathbf{Y}_{i,:} + \mathbf{Q}_{i,:}^{(0)}\right)^{\mathbf{T}} \end{array} \right) \right)}{\sum_{u \in \mathbb{U}_i} w_u^{(LFM)} + \lambda} \tag{47}$$

The details of the optimizations of $J^{(BiasSVD\_Weight)}$, $J^{(SBMF\_Weight)}$, and $J^{(MultiBiasSVD\_Weight)}$ are presented in Algorithms I, II, and III.

2) AGGREGATING GROUP MEMBERS' DEEP PROFILES TO THE GROUP'S DEEP PROFILE

After learning the user deep profiles, including user latent factor vectors, user biases, and user weights, we perform a weighted aggregation of group members' deep profiles to learn the group's deep profile, as follows:

$$\mathbb{DP}_{\mathbb{G}}^{(LFM)} = \{\mathbf{H}_{\mathbb{G},:}; o_{\mathbb{G}}\}$$

$$\mathbb{DP}_{\mathbb{G}}^{(LFM)} = \frac{\sum_{u \in \mathbb{G}} \left(w_u^{(LFM)} . \mathbb{DP}_u^{(LFM)} \right)}{\sum_{u \in \mathbb{G}} w_u^{(LFM)}} \tag{48}$$

$$\Leftrightarrow$$

$$\mathbf{H}_{\mathbb{G},:} = \frac{\sum_{u \in \mathbb{G}} \left(w_u^{(LFM)} . \mathbf{H}_{u,:} \right)}{\sum_{u \in \mathbb{G}} w_u^{(LFM)}}$$

$$o_{\mathbb{G}} = \frac{\sum_{u \in \mathbb{G}} \left(w_u^{(LFM)} . o_u \right)}{\sum_{u \in \mathbb{G}} w_u^{(LFM)}} \tag{49}$$

For example, with a group $\mathbb{G}$ consisting of two members, u_1 and u_2, the deep profiles of these group members will be extracted from the deep profiles of users built in the offline phase shown in Fig. 9. Fig. 10 illustrates the weighted aggregation of deep profiles of these two members using Eq. (49).

3) RATING PREDICTION

Based on the latent factor vector and bias of the group $\mathbb{G}$, we can predict the rating of the group $\mathbb{G}$ for a target item i, as follows:

Algorithm 1 The Optimization of $J^{(BiasSVD\_Weight)}$

Input. $\mathbf{R}_{mxn}$: user-item rating matrix**;** λ : regularization weight; s : the number of latent factors; β: constraint of user weights.

Output. $\mathbf{H}_{mxs}$: user-factor matrix; $\mathbf{Q}_{nxs}$:item-factor matrix; $o_u \; u = 1 \ldots m$: user biases; $p_i \; i = 1 \ldots n$: item biases; $w_u^{(LFM)} u = 1 \ldots m$: user weights.

Step 1: Initialize randomly all variables.
Step 2: Repeat until convergence:
Step 3:

$$\forall u = 1 \ldots m : w_u^{(LFM)} = \frac{\left(\sum_{i \in \mathbb{I}_u} \left(\begin{array}{c} R_{u,i} - o_u - p_i \\ - \mu - \mathbf{H}_{u,:} . \mathbf{Q}_{i,:}^{\mathbf{T}} \end{array} \right)^2 \right)^{1/(\beta - 1)}}{\left(\sum_{u'=1}^{m} \left(\sum_{i \in \mathbb{I}_{u'}} \left(\begin{array}{c} R_{u',i} - o_{u'} - p_i \\ - \mu - \mathbf{H}_{u',:} . \mathbf{Q}_{i,:}^{\mathbf{T}} \end{array} \right)^2 \right)^{\beta/(\beta - 1)} \right)^{1/\beta}}$$

Step 4:

$$\forall u = 1 \ldots m : \mathbf{H}_{u,:} = w_u^{(LFM)} \cdot \sum_{i \in \mathbb{I}_u} \left(\mathbf{Q}_{i,:} . (R_{u,i} - o_u - p_i - \mu) \right) . \left(\lambda . \mathbf{I} + w_u^{(LFM)} . \sum_{i \in \mathbb{I}_u} \left(\mathbf{Q}_{i,:}^{\mathbf{T}} . \mathbf{Q}_{i,:} \right) \right)^{-1}$$

Step 5:

$$\forall i = 1 \ldots n : \mathbf{Q}_{i,:} = \sum_{u \in \mathbb{U}_i} \left(w_u^{(LFM)} . \mathbf{H}_{u,:} . (R_{u,i} - o_u - p_i - \mu) \right) . \left(\lambda . \mathbf{I} + \sum_{u \in \mathbb{U}_i} \left(w_u^{(LFM)} . \mathbf{H}_{u,:}^{\mathbf{T}} . \mathbf{H}_{u,:} \right) \right)^{-\mathbf{1}}$$

Step 6:

$$\forall u = 1 \ldots m : o_u = \frac{w_u^{(LFM)} \cdot \sum_{i \in \mathbb{I}_u} \left(\begin{array}{c} R_{u,i} - p_i \\ - \mu - \mathbf{H}_{u,:} . \mathbf{Q}_{i,:}^{\mathbf{T}} \end{array} \right)}{w_u^{(LFM)} . |\mathbb{I}_u| + \lambda}$$

Step 7:

$$\forall i = 1 \ldots n : p_i = \frac{\sum_{u \in \mathbb{U}_i} \left(w_u^{(LFM)} \cdot \left(\begin{array}{c} R_{u,i} - o_u \\ - \mu - \mathbf{H}_{u,:} . \mathbf{Q}_{i,:}^{\mathbf{T}} \end{array} \right) \right)}{\sum_{u \in \mathbb{U}_i} w_u^{(LFM)} + \lambda}$$

Step 8: End Repeat

$$\hat{R}_{\mathbb{G},i} = o_{\mathbb{G}} + p_i + \mu + \mathbf{H}_{\mathbb{G},:} . \mathbf{Q}_{i,:}^{\mathbf{T}} \tag{50}$$

Fig. 11 shows the rating prediction of $\mathbb{G}$ for the target item i_3. Specifically, based on the deep profile of the group shown in Fig. 10 and the latent factor vector of i_3 shown in Fig. 9, the predicted rating of $\mathbb{G}$ for i_3 is calculated as follows:

$$\hat{R}_{\mathbb{G},i} = 0,01 - 0,56 + 2,8 + 0,76.0,44 + 0,80.0,29 = 2,7 \tag{51}$$

V. EXPERIMENTS

A. EXPERIMENTAL SETUP

In this section, we compare our proposed methods with related methods presented in section II. They are summarized in Table 2.

We randomly generate groups of sizes 2, 3, 4, 5 respectively. With 250 groups for each size, we get 1000 groups for evaluating the group recommendation methods.

We establish the convergence of the optimizations of the objective functions in the latent factor models according to

Algorithm 2 The Optimization of $J^{(SBMF\_Weight)}$

Input. $\mathbf{R}_{mxn}$: user-item rating matrix**;** $\mathbf{C}_{mxn}$: user-item sentiment matrix**;** λ : regularization weight; s : the number of latent factors; β: constraint of user weights.

Output. $\mathbf{H}_{mxs}$: user-factor matrix; $\mathbf{Q}_{nxs}$**:**item-factor matrix; $o_u\ u = 1 \ldots m$: user biases; $p_i\ i = 1 \ldots n$: item biases; $w_u^{(LFM)}\ u = 1 \ldots m$: user weights.

Step 1: Initialize randomly all variables.
Step 2: Repeat until convergence:
Step 3:

$$\forall u = 1 \ldots m : w_u^{(LFM)} = \frac{\left(\sum_{i\in\mathbb{I}_u}\left(\begin{pmatrix} R_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}^2 + \begin{pmatrix} C_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}^2\right)\right)^{1/(\beta-1)}}{\left(\sum_{u'=1}^{m}\left(\sum_{i\in\mathbb{I}_{u'}}\left(\begin{pmatrix} R_{u',i} - o_{u'} - p_i \\ -\mu - \mathbf{H}_{u',:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}^2 + \begin{pmatrix} C_{u',i} - o_{u'} - p_i \\ -\mu - \mathbf{H}_{u',:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}^2\right)\right)^{\beta/(\beta-1)}\right)^{1/\beta}}$$

Step 4:

$$\forall u = 1 \ldots m : \mathbf{H}_{u,:} = w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\mathbf{Q}_{i,:}.\begin{pmatrix} R_{u,i} + C_{u,i} - 2.o_u \\ -2.p_i - 2.\mu\end{pmatrix}\right) .\left(\lambda.\mathbf{I} + \mathbf{2}.w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\mathbf{Q}^{\mathbf{T}}_{i,:}.\mathbf{Q}_{i,:}\right)\right)^{-\mathbf{1}}$$

Step 5:

$$\forall i = 1 \ldots n : \mathbf{Q}_{i,:} = \sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}_{u,:}.\begin{pmatrix} R_{u,i} + C_{u,i} - 2.o_u \\ -2.p_i - 2.\mu\end{pmatrix}\right) .\left(\lambda.\mathbf{I} + \mathbf{2}.\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}^{\mathbf{T}}_{u,:}.\mathbf{H}_{u,:}\right)\right)^{-\mathbf{1}}$$

Step 6:

$$\forall u = 1 \ldots m : o_u = \frac{w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\begin{pmatrix} R_{u,i} + C_{u,i} - 2.p_i \\ -2.\mu - 2.\mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}}{2.w_u^{(LFM)}.|\mathbb{I}_u| + \lambda}$$

Step 7:

$$\forall i = 1 \ldots n : p_i = \frac{\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}\begin{pmatrix} R_{u,i} + C_{u,i} - 2.o_u \\ -2.\mu - 2.\mathbf{H}_{u,:}.\mathbf{Q}^{\mathbf{T}}_{i,:}\end{pmatrix}\right)}{2.\sum_{u\in\mathbb{U}_i} w_u^{(LFM)} + \lambda}$$

Step 8: End Repeat

Algorithm 3 The Optimization of $J^{(MultiBiasSVD\_Weight)}$

Input. $\mathbf{R}_{mxn}$: user-item rating matrix**;** λ : regularization weight; s : the number of latent factors; β: constraint of user weights;
$\mathbf{H}^{(0)}_{mxs}$: initial user-factor matrix; $\mathbf{Q}^{(0)}_{nxs}$: initial item-factor matrix.
(see Eq. (10))

Output. $\mathbf{H}_{mxs}$: user-factor matrix; $\mathbf{Q}_{nxs}$**:**item-factor matrix; $o_u\ u = 1 \ldots m$: user biases; $p_i\ i = 1 \ldots n$: item biases; $w_u^{(LFM)}\ u = 1 \ldots m$: user weights.

Step 1: Initialize randomly all variables.
Step 2: Repeat until convergence:
Step 3:

$$\forall u = 1 \ldots m : w_u^{(LFM)} = \frac{\left(\sum_{i\in\mathbb{I}_u}\begin{pmatrix} R_{u,i} - o_u - p_i - \mu \\ -\left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right).\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(0)}_{i,:}\right)^{\mathbf{T}}\end{pmatrix}^2\right)^{1/(\beta-1)}}{\left(\sum_{u'=1}^{m}\left(\sum_{i\in\mathbb{I}_{u'}}\begin{pmatrix} R_{u',i} - o_{u'} - p_i - \mu \\ -\left(\mathbf{D}_{u',:} + \mathbf{H}^{(0)}_{u',:}\right) \\ .\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(0)}_{i,:}\right)^{\mathbf{T}}\end{pmatrix}^2\right)^{\beta/(\beta-1)}\right)^{1/\beta}}$$

Step 4:

$$\forall u = 1 \ldots m : \mathbf{D}_{u,:} = w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\begin{pmatrix} R_{u,i} - o_u - p_i - \mu \\ -\mathbf{H}^{(0)}_{u,:}.\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(0)}_{i,:}\right)^{\mathbf{T}}\end{pmatrix} .\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(\mathbf{0})}_{i,:}\right)\right) .\left(\lambda.\mathbf{I} + w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(\mathbf{0})}_{i,:}\right)^{\mathbf{T}}.\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(\mathbf{0})}_{i,:}\right)\right)\right)^{-1}$$

Step 5:

$$\forall i = 1 \ldots n : \mathbf{Y}_{i,:} = \sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\begin{pmatrix} R_{u,i} - o_u - p_i - \mu \\ -\mathbf{Q}^{(0)}_{u,:}.\left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right)^{\mathbf{T}}\end{pmatrix} .\left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right)\right) .\left(\lambda.\mathbf{I} + \sum_{u\in\mathbb{U}_i} w_u^{(LFM)}.\left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right)^{\mathbf{T}}.\left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right)\right)^{-1}$$

Step 6:

$$\forall u = 1 \ldots m : o_u = \frac{w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\begin{pmatrix} R_{u,i} - p_i - \mu \\ -\left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right).\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(0)}_{i,:}\right)^{\mathbf{T}}\end{pmatrix}}{w_u^{(LFM)}.|\mathbb{I}_u| + \lambda}$$

Step 7:

$$\forall i = 1 \ldots n : p_i = \frac{\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\begin{pmatrix} R_{u,i} - o_u - \mu \\ -\left(\mathbf{D}_{u,:} + \mathbf{H}^{(0)}_{u,:}\right).\left(\mathbf{Y}_{i,:} + \mathbf{Q}^{(0)}_{i,:}\right)^{\mathbf{T}}\end{pmatrix}\right)}{\sum_{u\in\mathbb{U}_i} w_u^{(LFM)} + \lambda}$$

Step 8: End Repeat
Step 9: $\mathbf{H} = \mathbf{D} + \mathbf{H}^{(0)}$
Step 10: $\mathbf{Q} = \mathbf{Y} + \mathbf{Q}^{(0)}$

the following two conditions: the change of the objective function between two consecutive iterations is less than 10^{-6} or the maximum number of iterations is 800 [51]. The parameters in the experimental methods are tuned on the validation set selected from 20% of the training set. Thanks to grid search, we find parameter sets that are optimal for each experimental method on each experimental dataset. Details of parameters after tuning are presented in Table 3.

For each experiment, we run five different times which corresponds to five different random splits of training and testing. The average of these five runs is used to conclude.

B. MEASURE

To evaluate the group recommendation methods, we use two common measures, F1-score and nDCG. F1-score is concerned with whether a recommended item set satisfies the group while nDCG focuses on the accuracy of ranking items in the recommended set.

For the F1-score, it is essential to simulate the item set expected by the group. Specifically, from the test ratings, an item that all group members like, i.e. their test ratings

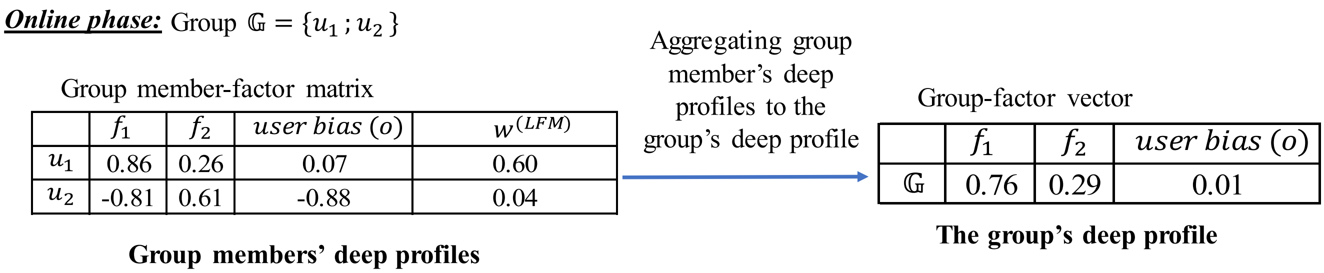


FIGURE 10. Aggregating group members' deep profiles to the group's deep profile in the latent factor models.

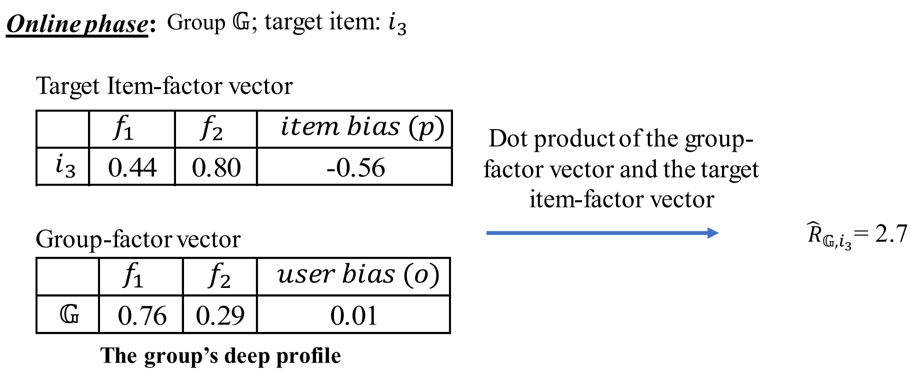


FIGURE 11. Predicting a rating of a group for an item in the latent factor models.

for the item are greater than the like threshold, is selected for the group's expected set [29], [45], [48], [52]. The comparison between the expected set $\mathbb{T}$ and the recommended set $\mathbb{C}$ is performed in two different aspects, recall and precision. Precision is the ratio between the number of correctly recommended items, i.e., $|\mathbb{T} \cap \mathbb{C}|$, and the number of recommended items, i.e., $|\mathbb{C}|$. In contrast, with recall, the number of correctly recommended items, i.e., $|\mathbb{T} \cap \mathbb{C}|$, is considered in relation to the number of expected items, i.e., $|\mathbb{C}|$. The combination of precision and recall produces the F1-score, as follows:

$$precision = \frac{|\mathbb{T} \cap \mathbb{C}|}{|\mathbb{C}|}; \quad recall = \frac{|\mathbb{T} \cap \mathbb{C}|}{|\mathbb{T}|}$$
$$f1 - score = \frac{2.precision.recall}{precision + recall} \tag{52}$$

For the nDCG, it would be appreciated if the items that were ranked first in the recommended set are the items the group liked the most. To calculate the nDCG, the perfect ranking of the items in the recommended set is determined based on the testing ratings. Then, the nDCG will be the ratio between the ranking of items in the recommended set according to the predicted ratings and that according to the testing ratings, as follows:

$$DCG_{\mathbb{C}} = \sum_{i=1}^{|\mathbb{C}|} \frac{2^{r_{\mathbb{G},i}} - 1}{log_2(i+1)};$$
$$IDCG_{\mathbb{C}} = \sum_{i=1}^{|\mathbb{C}\_reorder|} \frac{2^{r_{\mathbb{G},i}} - 1}{log_2(i+1)}$$
$$nDCG = \frac{DCG_{\mathbb{C}}}{IDCG_{\mathbb{C}}} \tag{53}$$

C. DATASET

Since DP_SBMF uses textual review data for the recommendation, we have chosen experimental datasets that contain both rating data and review data. Specifically, we conducted experiments on 4 datasets with a 5-star rating system corresponding to 4 types of items at Amazon (http://jmcauley.ucsd.edu/data/amazon/), as follows:

- Baby
- Tools-Home Improvement
- Beauty
- Clothing-Accessories

Their details are shown in Table 4. For DP_SBMF, to convert textual reviews into sentiment scores, we use a sentiment dictionary proposed in [27]. Regarding the user-item implicit feedback matrix $\mathbf{R}'$ used in Eq. (10) of Multi Bias-SVD, according to the authors in [26], [35], and [39], it can be simulated from the user-item rating matrix $\mathbf{R}$. Specifically, if user u's rating for an item i is observed, then u's implicit feedback for i will be 1. Otherwise, u's implicit feedback for i will be 0.

All experimental datasets are randomly split 70% for training and 30% for testing.

D. EXPERIMENT RESULTS AND DISCUSSIONS

Tables 5-6 show the F1-score and nDCG results of the experimental methods when the recommended sets are the top 8 items with the highest predicted ratings. It can be seen that deep profile-based methods give better performance than item preference profile-based methods in terms of both memory-based collaborative filtering and latent factor models. In the deep profile-based methods, memory-based collaborative filtering produces lower results than latent

TABLE 2. The methods implemented for experiments.

Profile	Recommendation algorithm	Summary	Notation
Item preference **P**rofile (IP)	Latent factor model: **Bias-SVD** [37] (see Eq. (4-5))	-Aggregating group members' **O**bserved **R**atings to the group's profile. -Recommending the group's profile using Bias-SVD.	**IP_BiasSVD_OR** [49]
Item preference**P**rofile (IP)	Latent factor model: **Bias-SVD** [37] (see Eq. (4-5))	-Aggregating group members' **O**bserved **R**atings and **F**illed **R**atings to the group's profile. -Recommending the group's profile using Bias-SVD.	**IP_BiasSVD_OFR** [48]
Item preference **P**rofile (IP)	Latent factor model: **Bias-SVD** [37] (see Eq. (4-5))	-Aggregating **O**bserved **R**atings and **F**illed **R**atings given not only by group members but also by their neighbors the group's profile. -Recommending the group's profile using Bias-SVD.	**IP_BiasSVD_OFNR** [29]
Item preference **P**rofile (IP)	**M**emory-based **C**ollaborative **F**iltering: **Pearson** similarity [33]	-Aggregating group members' observed ratings to the group's profile. -Recommending the group's profile using memory-based collaborative filtering with Pearson similarity measure	**IP_MCF_ Pearson** [46]
Item preference **P**rofile (IP)	**M**emory-based **C**ollaborative **F**iltering: **Pearson** similarity [33]	-Aggregating group members' observed ratings to the group's profile. -The weights in the profile aggregation are estimated based on the group member-item rating matrix. -Recommending the group's profile using memory-based collaborative filtering with Pearson similarity measure	**IP_MCF_ Pearson_Weight [45]**
Item preference **P**rofile (IP)	**M**emory-based **C**ollaborative **F**iltering: **SVM**-based similarity [28]	-Aggregating group members' observed ratings to the group's profile. -Recommending the group's profile using memory-based collaborative filtering with SVM-based similarity measure	**IP_MCF_SVM** [28]
Deep **P**rofile (DP)	Latent factor model: **Bias-SVD** [37] (see Eq. (4-5))	-Learning the group members' deep profiles with Bias-SVD. -Aggregating the group members' deep profiles to the group's deep profile. -Using Bias-SVD to recommend the group with learned deep profile.	**DP_BiasSVD** (proposed in subsection IV.B)
Deep **P**rofile (DP)	Latent factor model: **SBMF** [27] (see Eq. (19-20))	-Learning the group members' deep profiles with SBMF. -Aggregating the group members' deep profiles to the group's deep profile -Using SBMF to recommend the group with learned deep profile	**DP_ SBMF** (proposed in subsection IV.B)
Deep **P**rofile (DP)	Latent factor model: **Multi Bias-SVD** [26] (see Eq. (10-14))	-Learning the group members' deep profiles with Multi Bias-SVD. -Aggregating the group members' deep profiles to the group's deep profile -Using Multi Bias-SVD to recommend the group with learned deep profile	**DP_ MultiBiasSVD** (proposed in subsection IV.B)
Deep **P**rofile (DP)	**M**emory-based **C**ollaborative **F**iltering: **OS** similarity [14] (see Eq. (2))	-Learning the group members' deep profiles with OS similarity -Aggregating the group members' deep profiles to the group's deep profile -Using memory-based collaborative filtering to recommend the group with learned deep profile	**DP_MCF_OS** (proposed in subsection IV.A)
Deep **P**rofile (DP)	**M**emory-based **C**ollaborative **F**iltering: LM similarity [34] (see Eq. (3))	-Learning the group members' deep profiles with LM similarity -Aggregating the group members' deep profiles to the group's deep profile -Using memory-based collaborative filtering to recommend the group with learned deep profile	**DP_MCF_ LM** (proposed in subsection IV.A)

TABLE 3. The parameters in the experiments.

Experiment methods	The size of neighbor set (k) searched from {10, 15, 20, 25, 30}	The number of latent factors (s) searched from {50,60,70,80,90}	The regularization weight (λ) searched from {0.01, 0.02, 0.03, 0.04, 0.05}	The constraint of user weights (β) searched from {0.5, 0.6, 0.7, 0.8, 0.9}
		Baby dataset		
IP_BiasSVD_OR	X	90	0.03	X
IP_BiasSVD_OFR	X	50	0.02	X
IP_BiasSVD_OFNR	X	60	0.03	X
IP_MCF_ Pearson	20	X	X	X
IP_MCF_ Pearson_Weight	20	X	X	X
IP_MCF_SVM	30	X	X	X
DP_BiasSVD	X	90	0.03	0.8
DP_ SBMF	X	50	0.01	0.5
DP_ MultiBiasSVD	X	60	0.05	0.9
DP_MCF_OS	25	X	X	X
DP_MCF_ LM	25	X	X	X
		Tools-Home Improvement dataset		
IP_BiasSVD_OR	X	50	0.05	X
IP_BiasSVD_OFR	X	60	0.03	X
IP_BiasSVD_OFNR	X	90	0.02	X
IP_MCF_ Pearson	25	X	X	X
IP_MCF_ Pearson_Weight	15	X	X	X
IP_MCF_SVM	25	X	X	X
DP_BiasSVD	X	90	0.05	0.6
DP_ SBMF	X	70	0.03	0.8
DP_ MultiBiasSVD	X	80	0.03	0.5
DP_MCF_OS	30	X	X	X
DP_MCF_ LM	20	X	X	X
		Beauty dataset		
IP_BiasSVD_OR	X	50	0.04	X
IP_BiasSVD_OFR	X	60	0.05	X
IP_BiasSVD_OFNR	X	50	0.01	X
IP_MCF_ Pearson	20	X	X	X
IP_MCF_ Pearson_Weight	20	X	X	X
IP_MCF_SVM	30	X	X	X
DP_BiasSVD	X	50	0.01	0.7
DP_ SBMF	X	80	0.02	0.5
DP_ MultiBiasSVD	X	80	0.01	0.7
DP_MCF_OS	30	X	X	X
DP_MCF_ LM	25	X	X	X
		Clothing-Accessories dataset		
IP_BiasSVD_OR	X	90	0.02	X
IP_BiasSVD_OFR	X	60	0.01	X
IP_BiasSVD_OFNR	X	70	0.01	X
IP_MCF_ Pearson	20	X	X	X
IP_MCF_ Pearson_Weight	20	X	X	X
IP_MCF_SVM	30	X	X	X
DP_BiasSVD	X	60	0.03	0.6
DP_ SBMF	X	50	0.05	0.5
DP_ MultiBiasSVD	X	60	0.01	0.9
DP_MCF_OS	30	X	X	X
DP_MCF_ LM	25	X	X	X

TABLE 4. Experimental datasets.

Dataset	#Users	#Items	#Reviews and Ratings
Baby	19,445	7,050	160,792
Tools-Home Improvement	19,856	10,217	134,476
Beauty	22,365	12,101	198,502
Clothing-Accessories	39,387	23,033	278,677

factor models. However, the strength of memory-based filtering lies in the high interpretation of the recommended item set.

Focusing on the experimental results of latent factor models in Table 5, DP_BiasSVD gives F1-score and nDCG a 11.01% and 12.18% increase compared to IP_BiasSVD in the Baby dataset, which is the least sparse dataset. These numbers are 14.79% and 16.60% in the Tools-Home Improvement dataset, which is the sparsest dataset. The sparser the dataset, the better are the group recommendation based on the deep profiles than the one based on item preference profiles. In addition, the experimental results also show that the deep profiles that are learned from side data-integrated latent factor models, i.e., SBMF using textual review data and Multi BiasSVD using implicit feedback data, supports better group recommendation than

TABLE 5. Experimental results of group recommendation methods using latent factor models. The top 2 methods with highest results are highlighted in blue.

F1-SCORE	IP_BiasSVD	IP_BiasSVD_OFR	IP_BiasSVD_OFNR	DP_BiasSVD	DP_SBMF	DP_MultiBiasSVD
Baby	65.911	71.934	73.229	73.169	79.007	75.669
Tools-Home Improvement	65.519	70.737	74.655	75.211	80.066	78.838
Clothing-Accessories	63.939	76.345	76.252	79.623	83.603	84.280
Beauty	67.128	68.418	70.251	74.409	78.646	80.340

NDCG	IP_BiasSVD	IP_BiasSVD_OFR	IP_BiasSVD_OFNR	DP_BiasSVD	DP_SBMF	DP_MultiBiasSVD
Baby	71.996	74.125	73.441	80.763	82.381	83.083
Tools-Home Improvement	70.659	76.142	79.032	82.389	84.990	80.468
Clothing-Accessories	65.774	66.847	69.644	71.029	76.271	73.202
Beauty	68.450	71.069	72.889	76.343	80.146	80.695

TABLE 6. Experimental results of group recommendation methods using memory-based collaborative filtering. The top 2 methods with highest results are highlighted in blue.

F1-SCORE	IP_MCF_Pearson	IP_MCF_Pearson_Weight	IP_MCF_SVM	DP_MCF_OS	DP_MCF_LM
Baby	62.408	64.527	67.134	71.222	73.766
Tools-Home Improvement	61.265	65.781	70.533	73.939	72.367
Clothing-Accessories	63.549	64.931	67.436	75.269	77.689
Beauty	65.211	65.581	63.932	66.524	69.540

NDCG	IP_MCF_Pearson	IP_MCF_Pearson_Weight	IP_MCF_SVM	DP_MCF_OS	DP_MCF_LM
Baby	68.154	69.296	73.964	75.929	78.235
Tools-Home Improvement	65.023	64.666	69.840	72.871	74.517
Clothing-Accessories	66.712	71.558	68.636	79.567	77.491
Beauty	68.221	69.699	67.239	74.557	78.297

the traditional latent factor model based only on rating data, i.e., BiasSVD. For example, in the Beauty dataset, the F1-score results of DP_SBMF and DP_MultiBiasSVD are 1.06 times and 1.08 times the F1-score result of DP_BiasSVD, respectively. These results reinforce previous statements about the effectiveness of side data sources like text reviews and implicit feedback in modern recommender systems.

Next, we analyze the experimental results according to each group size in Fig. 12-13. The larger the group size is, the more our proposed methods demonstrate superiority over previous methods. The reason for this phenomenon is explained: When the group has more members, it is more difficult to find an item in which all members provide interests. Therefore, the aggregation of the item preferences obtained from the members will be less representative of the interests of the whole group. This difficulty does not exist when performing aggregation of group members' deep profiles, which are completely specified.

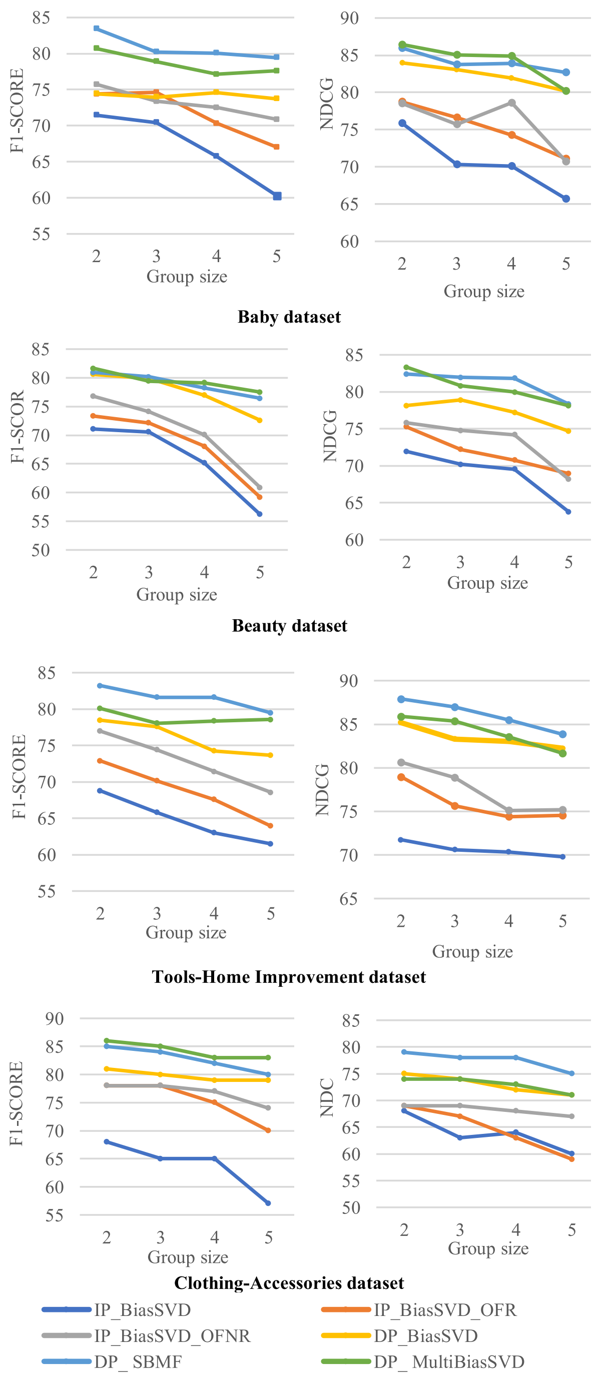


FIGURE 12. Experimental results of group recommendation methods using latent factor models according to the group sizes.

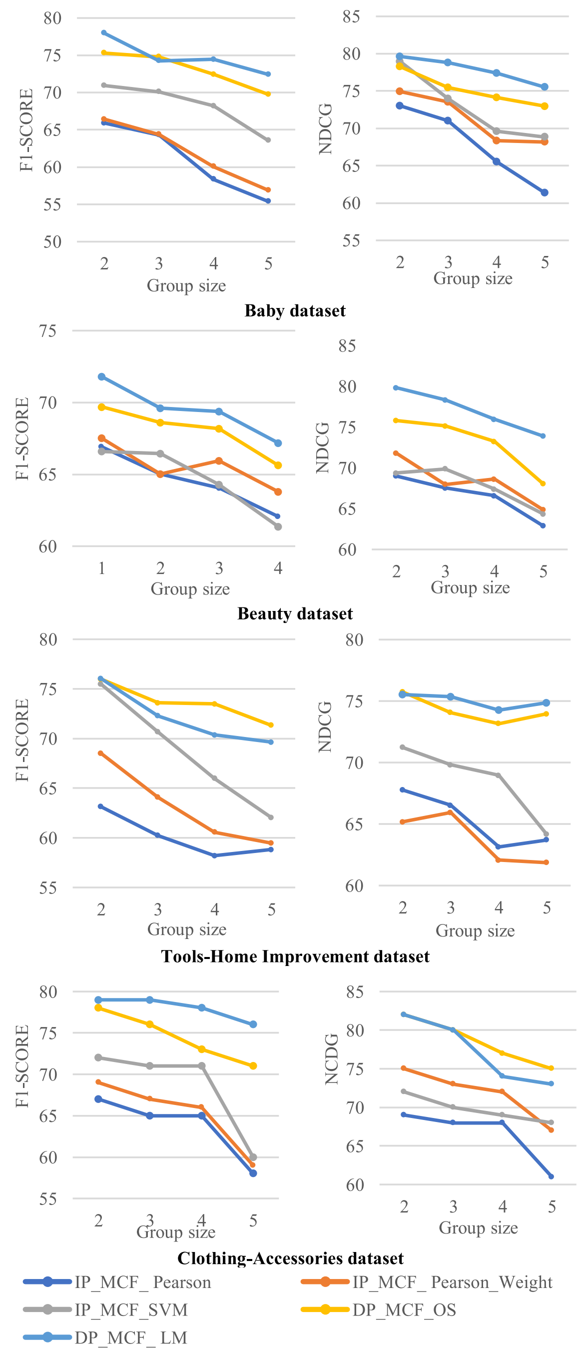


FIGURE 13. Experimental results of group recommendation methods using memory-based collaborative filtering according to the group sizes.

The formation of a group is divided into active and passive based on the group members' perception of other group members [53]. Users with similar interests tend to actively form a group. The greater the similarity of interests among group members, the higher the active in group formation and vice versa [45], [49]. Therefore, now we classify the experimental groups according to the average interest similarity among group members. Fig. 14-15 shows the group recommendation performance in this aspect. It can be remarked that the lower the average similarity of interest in the group, i.e. passive groups, the larger the difference in experimental results between our methods and other methods. Passive groups are more likely to have conflicts of interest than active groups. Therefore, the group's item preference

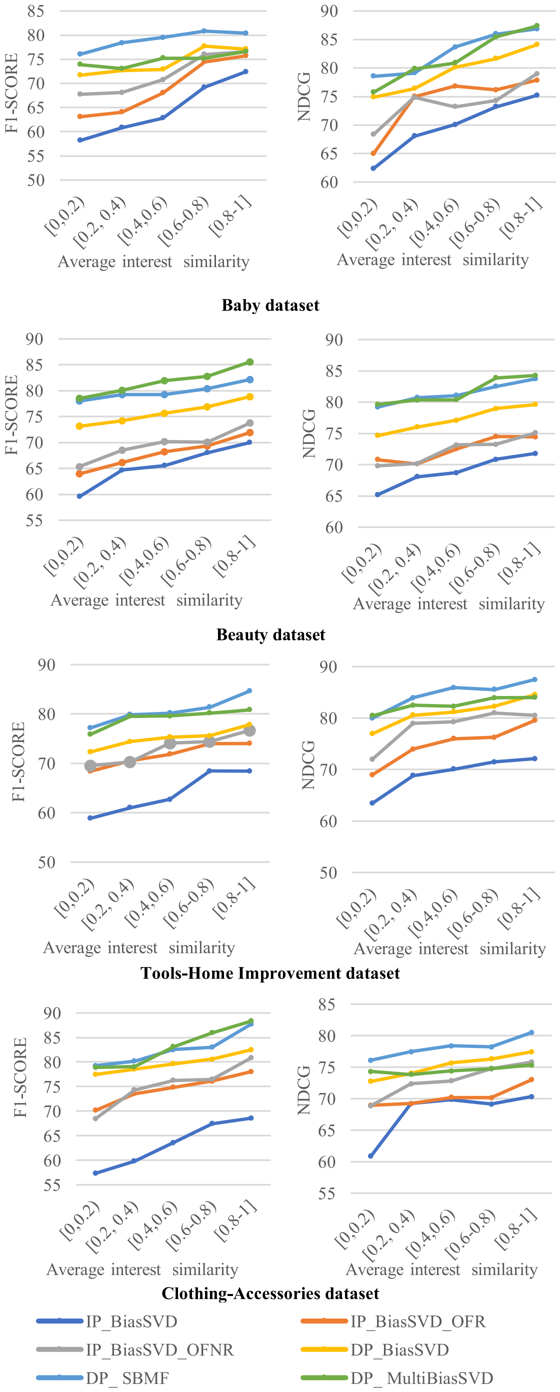


FIGURE 14. Experimental results of group recommendation methods using latent factor models according to average interest similarity among group members.

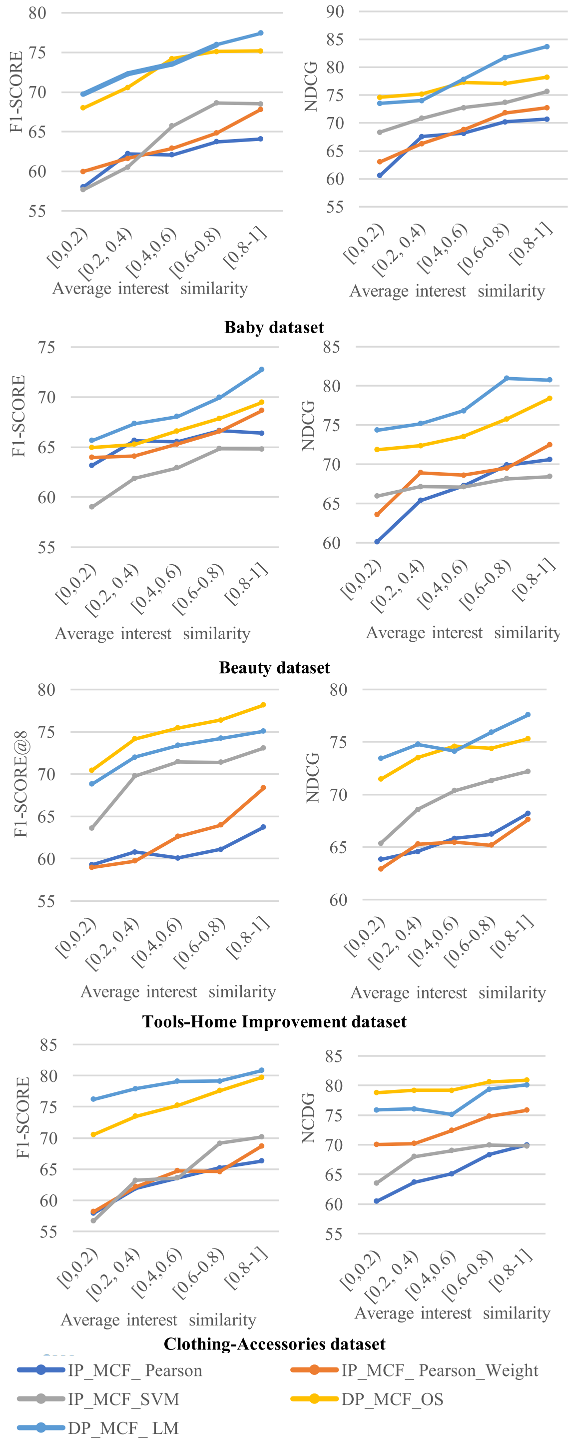


FIGURE 15. Experimental results of group recommendation methods using memory-based collaborative filtering according to average interest similarity among group members.

profile obtained by aggregating members' item preference profiles is often less accurate. This means that the quality of projecting the group's item preference profile into the latent factor space will be bad. Therefore, item preference profile aggregation-based methods always faced many difficulties in recommending passive groups. As discussed in section II, a significant benefit of moving from item preference profile aggregation to deep profile aggregation is the elimination of the above projection operation in the recommendation process. This is why our methods work so well for passive groups.

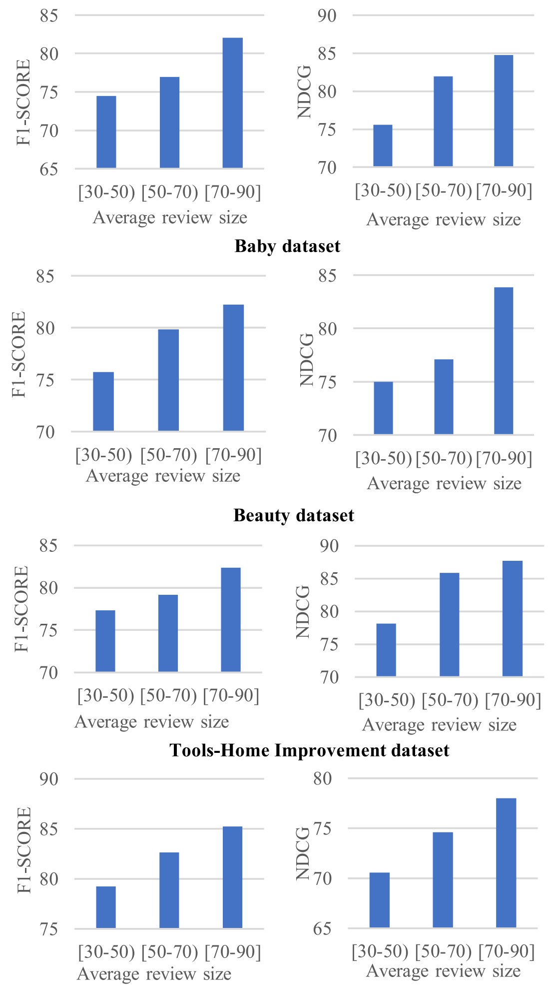


FIGURE 16. Experimental results of the DP_SBMF according to the average size of the text reviews that the group members wrote.

TABLE 7. The results of the t-test statistical comparison. a ≫ b means that a is statistically superior to B.

DP_BiasSVD vs. IP_BiasSVD	
T-value	2.7716
P-value	0.0051
Average F1-score result	75.4126 vs. 65.2589
Statistical conclusion	>>
DP_BiasSVD vs. IP_BiasSVD_OFR	
T-value	2.6491
P-value	0.0081
Average F1-score result	75.4126 vs. 72.7285
Statistical conclusion	>>
DP_BiasSVD vs. IP_BiasSVD_OFNR	
T-value	2.5961
P-value	0.0094
Average F1-score result	75.4126 vs. 74.2985
Statistical conclusion	>>
DP_MultiBiasSVD vs. DP_BiasSVD	
T-value	2.3389
P-value	0.0193
Average F1-score result	79.4992 vs 75.4126
Statistical conclusion	>>
DP_SBMF vs. DP_MultiBiasSVD	
T-value	1.9975
P-value	0.0458
Average F1-score result	80.2566 vs 79.4992
Statistical conclusion	>>
DP_MCF_OS vs. IP_MCF_Peason	
T-value	2.3927
P-value	0.0168
Average F1-score result	71.4873 vs. 62.3641
Statistical conclusion	>>
DP_MCF_OS vs. IP_MCF_Peason_Weight	
T-value	2.2544
P-value	0.0242
Average F1-score result	71.4873 vs. 64.5590
Statistical conclusion	>>
DP_MCF_OS vs. IP_MCF_SVM	
T-value	2.1052
P-value	0.0353
Average F1-score result	71.4873 vs. 68.0429
Statistical conclusion	>>
DP_MCF_LM vs. IP_MCF_OS	
T-value	2.0745
P-value	0.0383
Average F1-score result	74.328 vs 71.4873
Statistical conclusion	>>

The above experimental results have shown that the textual reviews in DP_SBMF best support group recommendations. This motivates us to further analyze the experimental results of the DP_SBMF according to the average size of the text reviews that the group members wrote. As shown in Fig. 16, the more detailed the reviews the group members write, the more accurate the DP_SBMF recommends the group. Therefore, in addition to collecting ratings, group recommender systems need to pay more attention to collecting reviews as well as issuing policies to encourage users to write detailed reviews.

The above experiments were conducted when the group's satisfaction for items is simulated at the strictest level. Specifically, the group is satisfied with an item only if all group members like the item. However, in some contexts, the group becomes more lenient in rating items. As long as 2/3, even 1/2, of the group members like the item, it can be said that the group is satisfied with the item. Fig. 17-18 show the results of the methods in these such contexts. When the process of rating items became less rigorous, all the experimental methods got better results. However, our methods still maintain the lead.

Fig. 19-20 shows the processing time for a group recommendation method to predict a group's rating for an item. Because this time belongs to the online phase, it directly affects the user experience in the system. As can be seen from Fig. 19-20, the rating prediction time of the item preference-based methods depends on the number of items in the dataset. The reason is that both aggregating group members' item preferences and projecting the group's item preferences into the latent factor space require scanning all items. Therefore, the larger the number of items in the dataset, the longer the item preference-based methods take to predict a rating. This is a major disadvantage when implementing item

TABLE 8. The results of the t-test statistical comparison. a ≫ b means that a is statistically superior to B.

DP_BiasSVD vs. IP_BiasSVD	
Z-value	3.1907
P-value	0.0014
Average F1-score result	75.4126 vs. 65.2589
Statistical conclusion	>>
DP_BiasSVD vs. IP_BiasSVD_OFR	
Z-value	3.2685
P-value	0.0011
Average F1-score result	75.4126 vs. 72.7285
Statistical conclusion	>>
DP_BiasSVD vs. IP_BiasSVD_OFNR	
Z-value	2.8660
P-value	0.0041
Average F1-score result	75.4126 vs. 74.2985
Statistical conclusion	>>
DP_MultiBiasSVD vs. DP_BiasSVD	
Z-value	2.5131
P-value	0.0119
Average F1-score result	79.4992 vs 75.4126
Statistical conclusion	>>
DP_SBMF vs. DP_MultiBiasSVD	
Z-value	2.2389
P-value	0.0252
Average F1-score result	80.2566 vs 79.4992
Statistical conclusion	>>
DP_MCF_OS vs. IP_MCF_Peason	
Z-value	2.2075
P-value	0.0273
Average F1-score result	71.4873 vs. 62.3641
Statistical conclusion	>>
DP_MCF_OS vs. IP_MCF_Peason_Weight	
Z-value	2.1339
P-value	0.0329
Average F1-score result	71.4873 vs. 64.5590
Statistical conclusion	>>
DP_MCF_OS vs. IP_MCF_SVM	
Z-value	2.2206
P-value	0.0264
Average F1-score result	71.4873 vs. 64.5590
Statistical conclusion	>>
DP_MCF_LM vs. IP_MCF_OS	
Z-value	2.2037
P-value	0.0275
Average F1-score result	71.4873 vs. 68.0429
Statistical conclusion	>>
DP_MCF_LM vs. IP_MCF_OS	
Z-value	2.1411
P-value	0.0323
Average F1-score result	74.328 vs 71.4873
Statistical conclusion	>>

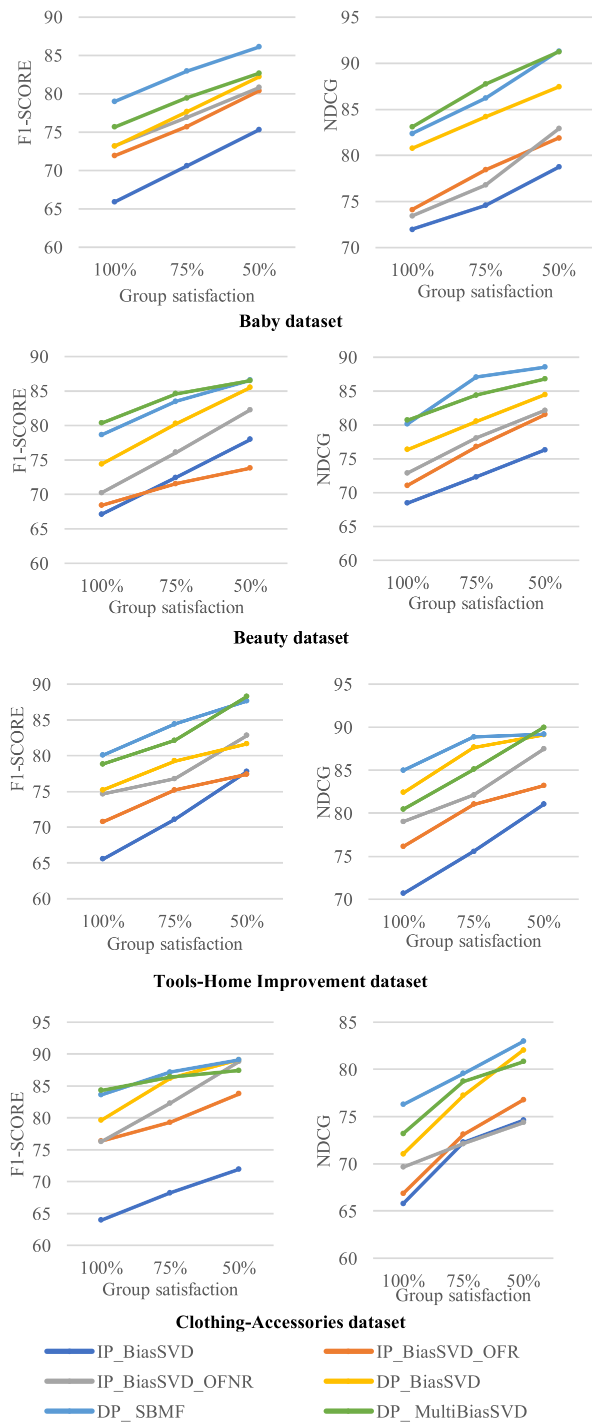


FIGURE 17. Experimental results of group recommendation methods using latent factor models according to the group satisfaction.

preference-based methods in large-scale settings. In contrast, in deep profile-based methods, the processing time for predicting a rating depends only on the number of latent factors. Therefore, when the number of items increases with each experimental dataset, the rating prediction time of deep profile-based methods does not change. Experiments of previous studies on latent factor models have shown that the optimal number of latent factors is always much smaller than the number of items. This provides a significant competitive advantage in terms of rating prediction time of deep profile-based methods over that of item preference profile-based methods. In addition, compared with the original latent factor model, i.e., DP_BiasSVD, the latent factor models that are integrated with side data such as DP_SBMF and DP_MultiBiasSVD only incur additional computational costs in the process of learning latent factors.

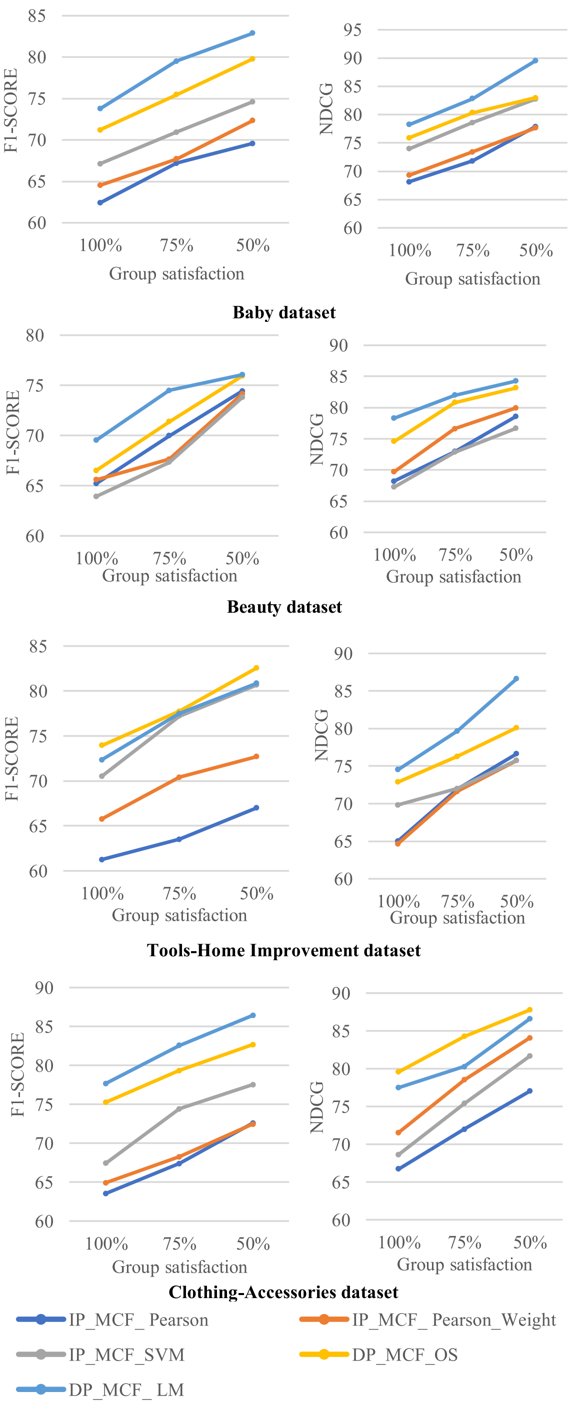


FIGURE 18. Experimental results of group recommendation methods using memory-based collaborative filtering according to the group satisfaction.

The rating prediction functions of all of them are the same. Therefore, the rating prediction time of DP_BiasSVD, DP_SBMF, and DP_MultiBiasSD is equivalent. It can be concluded that the deep profile-based approach is very suitable for deploying advanced latent factor models for group recommendation.

Finally, we use the F1-score samples of all experimental groups for statistical comparisons. Thereby, we aim to draw the most accurate conclusions about the comparison between methods. The statistical comparison includes parametric and non-parametric. In this study, we used both t-test, a parametric statistical comparison, and Wilcoxon signed-ranks test, a non-parametric statistical comparison, as in [43], [54], and [55]. T-test depends on t-value while Wilcoxon signed-ranks test depends on z-value. T-value and z-value are used to calculate the p-value. if the p-value is less than 0.05 then the two methods involved in statistical comparison will be significantly different. As shown in Table 7-8, our proposed methods are statistically better than other methods because the obtained p-values are all less than 0.05. Among our proposed methods, DP_SBMF and DP_MCF_LM are statistically the best in the case of latent factor model and memory-based collaborative filtering, respectively. These give stronger support to our above conclusions.

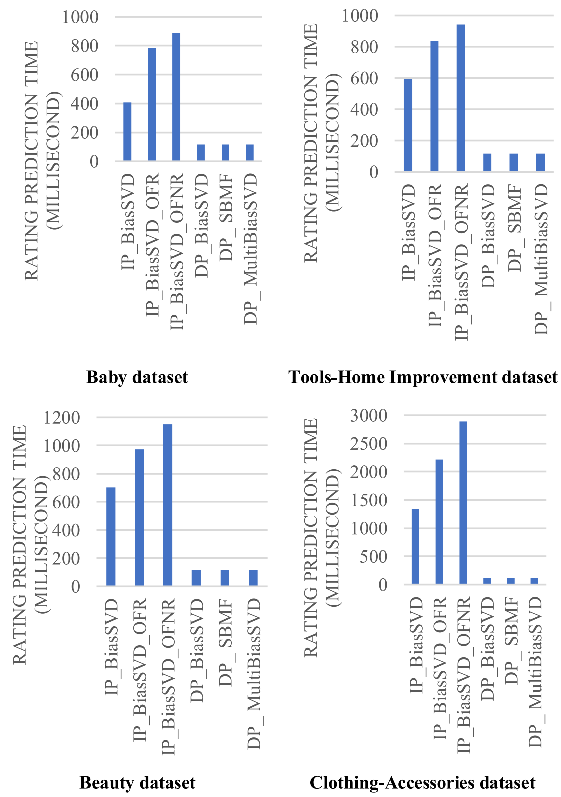


FIGURE 19. The rating prediction time of group recommendation methods using latent factor models.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we propose group recommendation methods based on deep profile aggregation. Deep profiles refer to the users' profiles deep within the recommendation algorithms. Compared with item preferences profiles, deep profiles have the advantage of low dimensionality and are fully specified. In addition, deep profile aggregation-based group recommendation helps eliminate the projection operation, which is a costly step in the online phase.

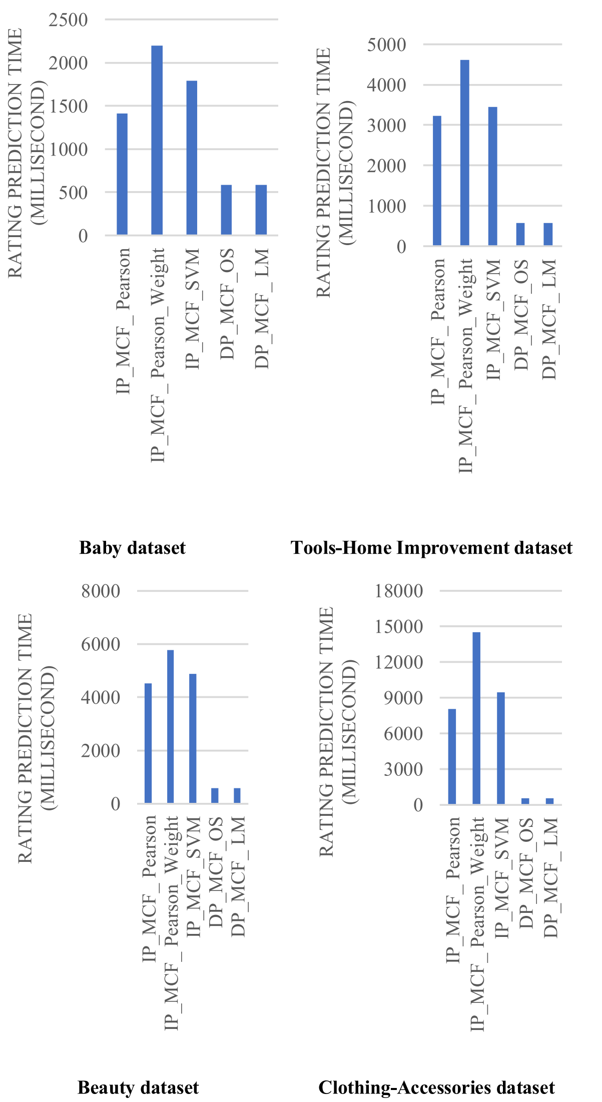


FIGURE 20. The rating prediction time of group recommendation methods using memory-based collaborative filtering.

This has facilitated the implementation of progressive latent factor models for group recommendation. Experiments have demonstrated a clear effect of deep profiles compared with item preference profiles in group recommendations when the recommendation algorithms are used respectively as memory-based collaborative filtering and latent factor model. Especially in the context of sparse data, group recommendations based on the deep profiles show even more outstanding performance. In addition, recent advanced latent factor models such as SBMF and Multi Bias-SVD have also proved very suitable for learning deep profiles. The experimental results were analyzed in many different aspects such as group size, review size, and group formation. Finally, we used statistical comparisons to confirm our conclusions.

The weakness of the methods proposed in this study is the invariance in the user weights. Maybe for one entry, a user has a high weight but for another entry his/her weight is not so high anymore. Therefore, to estimate the user weights that vary with each entry in the deep profile, the objective function of the latent factor models needs to be modified. However, the challenge is that the more variables in the objective function are, the more overfitting they present. Therefore, in the future, we aim to combine user weights defined on item preference profiles and user weights defined on deep profiles to achieve more perfect user weights.

The process of learning users' deep profiles is directly related to the selected recommendation algorithm. In other words, it is sensitive to the parameters used in the recommendation algorithm. Therefore, the methods proposed in this paper will fail if these parameters are not well defined. As shown above, to achieve good experimental results, we have performed parameter tuning very carefully. The definition of deep profiles that are both less dependent on the parameters of the recommendation algorithm and fully represent the users needs to be discussed more in future studies.

This study paves the way for the easy implementation of a single-user recommendation algorithm for the group recommendation. All we need to do is to define the appropriate deep profiles in the selected recommendation algorithm. Currently, hybrid recommendation algorithms have been confirmed to be more efficient than traditional recommendation algorithms. As a result, what will their deep profiles be? And how to deploy them for group recommendation? If their deep profiles are defined to be too complex, of course, there will be more computational costs for the online stage. In future works, we are supposed to address these questions to further refine deep profile aggregation-based group recommendations.

APPENDIX 1

$$
\begin{cases}
\forall u = 1 \ldots m : \dfrac{\partial J^{(BiasSVD\_Weight\_Lagrange)}}{\partial w_u^{(LFM)}} = 0 \\
\dfrac{\partial J^{(BiasSVD\_Weight\_Lagrange)}}{\partial \gamma} = 0
\end{cases}
$$
$$
\Leftrightarrow
\begin{cases}
\sum_{(u,i)\in\mathbb{R}} \begin{pmatrix} R_{u,i} - o_u - p_i \\ -\mu - \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}} \end{pmatrix}^2 + \gamma.\beta.w_u^{(LFM)^{\beta-1}} = 0(*) \\
\sum_{u=1}^{m} w_u^{(LFM)^{\beta}} - 1 = 0(**)
\end{cases}
$$
$$
(*) \Leftrightarrow w_u^{(LFM)} = \frac{\left(\sum_{(u,i)\in\mathbb{R}} \left(R_{u,i} - o_u - p_i - \mu - \mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\right)^2\right)^{1/(\beta-1)}}{(\gamma.\beta)^{1/(\beta-1)}} (***) \tag{54}
$$

Putting $(***)$ into $(**)$ is obtained by the equation as shown at the top of the next page.

$$\frac{1}{(\gamma.\beta)^{\beta/(\beta-1)}}\sum_{u=1}^{m}\left(\sum_{(u,i)\in\mathbb{R}}\begin{pmatrix}R_{u,i}-o_u\\-p_i-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)^{\beta/(\beta-1)}=1$$

$$\Leftrightarrow\quad\frac{1}{(\gamma.\beta)^{\beta/(\beta-1)}}=\frac{1}{\sum_{u=1}^{m}\left(\sum_{(u,i)\in\mathbb{R}}\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)^{\beta/(\beta-1)}}$$

$$\Leftrightarrow\quad\frac{1}{(\gamma.\beta)^{1/(\beta-1)}}=\frac{1}{\left(\sum_{u=1}^{m}\left(\sum_{(u,i)\in\mathbb{R}}\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)^{\beta/(\beta-1)}\right)^{1/\beta}}$$

$w_u^{(LFM)}$ can be rewritten as follows:

$$w_u^{(LFM)}=\frac{\left(\sum_{(u,i)\in\mathbb{R}}\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)^{1/(\beta-1)}}{\left(\sum_{u=1}^{m}\left(\sum_{(u,i)\in\mathbb{R}}\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}^2\right)^{\beta/(\beta-1)}\right)^{1/\beta}}$$

APPENDIX 2

$$\forall u=1\ldots m:\nabla_{\mathbf{H}_{u,:}}J^{(BiasSVD\_Weight)}=0$$

$$\Leftrightarrow$$

$$-w_u^{(LFM)}\sum_{i\in\mathbb{I}_u}\left(\mathbf{Q}_{i,:}.\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}\right)+\lambda.\mathbf{H}_{u,:}=0$$

$$\Leftrightarrow$$

$$\mathbf{H}_{u,:}=w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\mathbf{Q}_{i,:}.\left(R_{u,i}-o_u-p_i-\mu\right)\right).\left(\lambda.\mathbf{I}+w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(\mathbf{Q}_{i,:}^{\mathbf{T}}.\mathbf{Q}_{i,:}\right)\right)^{-1}$$

APPENDIX 3

$$\forall i=1\ldots m:\nabla_{\mathbf{Q}_{i,:}}J^{(BiasSVD\_Weight)}=0$$

$$\Leftrightarrow$$

$$-\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}_{u,:}.\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}\right)+\lambda.\mathbf{Q}_{i,:}=0$$

$$\Leftrightarrow$$

$$\mathbf{Q}_{i,:}=\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}_{u,:}.\left(R_{u,i}-o_u-p_i-\mu\right)\right).\left(\lambda.\mathbf{I}+\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\mathbf{H}_{u,:}^{\mathbf{T}}.\mathbf{H}_{u,:}\right)\right)^{-\mathbf{1}}$$

APPENDIX 4

$$\forall u=1\ldots m:\frac{J^{(BiasSVD\_Weight)}}{\partial o_u}=0$$

$$\Leftrightarrow$$

$$-w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}+\lambda.o_u=0$$

$$\Leftrightarrow$$

$$o_u=\frac{w_u^{(LFM)}.\sum_{i\in\mathbb{I}_u}\left(R_{u,i}-p_i-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\right)}{w_u^{(LFM)}.|\mathbb{I}_u|+\lambda}$$

APPENDIX 5

$$\forall i=1\ldots n:\frac{J^{(BiasSVD\_Weight)}}{\partial p_i}=0$$

$$\Leftrightarrow$$

$$-\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\begin{pmatrix}R_{u,i}-o_u-p_i\\-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\end{pmatrix}\right)+\lambda.p_i=0$$

$$\Leftrightarrow$$

$$p_i=\frac{\sum_{u\in\mathbb{U}_i}\left(w_u^{(LFM)}.\left(R_{u,i}-o_u-\mu-\mathbf{H}_{u,:}.\mathbf{Q}_{i,:}^{\mathbf{T}}\right)\right)}{\sum_{u\in\mathbb{U}_i}w_u^{(LFM)}+\lambda}$$

REFERENCES

[1] Q. Li, N. Liang, and E. Y. Li, "Does friendship quality matter in social commerce? An experimental study of its effect on purchase intention," *Electron. Commerce Res.*, vol. 18, no. 4, pp. 693–717, Dec. 2018.

[2] J. Masthoff, "Group recommender systems: Aggregation, satisfaction and group attributes," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 743–776.

[3] T. N. Nguyen and F. Ricci, "A chat-based group recommender system for tourism," *Inf. Technol. Tourism*, vol. 18, nos. 1–4, pp. 5–28, Apr. 2018.

[4] Y.-D. Seo, Y.-G. Kim, E. Lee, and H. Kim, "Group recommender system based on genre preference focusing on reducing the clustering cost," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115396.

[5] Y. Chen and P. Pu, "Cofeel: Using emotions to enhance social interaction in group recommender systems," in *Proc. Alpine Rendez-Vous (ARV) Workshop Tools Technol. Emotion-Awareness Comput. Mediated Collaboration Learn.*, 2013, pp. 48–55.

[6] M. O'connor, D. Cosley, J. A. Konstan, and J. Riedl, "PolyLens: A recommender system for groups of users," in *Proc. ECSCW*. Dordrecht, The Netherlands: Springer, 2001, pp. 199–218.

[7] T. D. Pessemier, J. Dhondt, and L. Martens, "Hybrid group recommendations for a travel service," *Multimedia Tools Appl.*, vol. 76, no. 2, pp. 2787–2811, Jan. 2017.

[8] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, Jun. 2015.

[9] C. Villavicencio, S. Schiaffino, J. A. Diaz-Pace, and A. Monteserin, "Group recommender systems: A multi-agent solution," *Knowl.-Based Syst.*, vol. 164, pp. 436–458, Jan. 2019.

[10] B. Smith and G. Linden, "Two decades of recommender systems at Amazon.com," *IEEE Internet Comput.*, vol. 21, no. 3, pp. 12–18, May/Jun. 2017.

[11] M. Srifi, A. Oussous, A. Ait Lahcen, and S. Mouline, "Recommender systems based on collaborative filtering using review texts—A survey," *Information*, vol. 11, no. 6, p. 317, Jun. 2020.

[12] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič, *Group Recommender Systems: An Introduction*. Cham, Switzerland: Springer, 2018.

[13] Y. Xiao, Q. Pei, L. Yao, S. Yu, L. Bai, and X. Wang, "An enhanced probabilistic fairness-aware group recommendation by incorporating social activeness," *J. Netw. Comput. Appl.*, vol. 156, Apr. 2020, Art. no. 102579.

[14] A. Gazdar and L. Hidri, "A new similarity measure for collaborative filtering based recommender systems," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105058.

[15] N. Mirbakhsh and C. X. Ling, "Leveraging clustering to improve collaborative filtering," *Inf. Syst. Frontiers*, vol. 20, no. 1, pp. 111–124, Feb. 2018.

[16] M. Nilashi, O. Ibrahi, and K. Bagherifard, "A recommender system based on collaborative filtering using Ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018.

[17] W. Cai, W. Pan, J. Liu, Z. Chen, and Z. Ming, "K-reciprocal nearest neighbors algorithm for one-class collaborative filtering," *Neurocomputing*, vol. 381, pp. 207–216, Mar. 2020.

[18] D. Valcarce, A. Landin, J. Parapar, and Á. Barreiro, "Collaborative filtering embeddings for memory-based recommender systems," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 347–356, Oct. 2019.

[19] J. Wang and Q. Tang, "A probabilistic view of neighborhood-based recommendation methods," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 14–20.

[20] D. K. Chae, S. C. Lee, S. Y. Lee, and S. W. Kim, "On identifying k-nearest neighbors in neighborhood models for efficient and effective collaborative filtering," *Neurocomputing*, vol. 278, pp. 134–143, Feb. 2018.

[21] X. Ma, H. Lu, Z. Gan, and J. Zeng, "An explicit trust and distrust clustering based collaborative filtering recommendation approach," *Electron. Commerce Res. Appl.*, vol. 25, pp. 29–39, Sep./Oct. 2017.

[22] X. Ning, C. Desrosiers, and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender Systems Handbook*. USA: Springer, 2015, pp. 37–76, doi: 10.1007/978-1-4899-7637-6_2.

[23] J. Chambua and Z. Niu, "Review text based rating prediction approaches: Preference knowledge learning, representation and utilization," *Artif. Intell. Rev.*, vol. 54, no. 2, pp. 1171–1200, Feb. 2021.

[24] H. Wu, Q. Zhou, R. Nie, and J. Cao, "Effective metric learning with co-occurrence embedding for collaborative recommendations," *Neural Netw.*, vol. 124, pp. 308–318, Apr. 2020.

[25] Z. Khan, N. Iltaf, H. Afzal, and H. Abbas, "Enriching non-negative matrix factorization with contextual embeddings for recommender systems," *Neurocomputing*, vol. 380, pp. 246–258, Mar. 2020.

[26] L. N. H. Nam, "Latent factor recommendation models for integrating explicit and implicit preferences in a multi-step decision-making process," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114772.

[27] R.-P. Shen, H.-R. Zhang, H. Yu, and F. Min, "Sentiment based matrix factorization with reliability for recommendation," *Expert Syst. Appl.*, vol. 135, pp. 249–258, Nov. 2019.

[28] S. Ghazarian and M. A. Nematbakhsh, "Enhancing memory-based collaborative filtering for group recommender systems," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3801–3812, May 2015.

[29] L. N. H. Nam, "Towards comprehensive profile aggregation methods for group recommendation based on the latent factor model," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115585.

[30] J. Masthoff, "Group recommender systems: Combining individual models," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 677–702.

[31] X. Wang, Y. Liu, J. Lu, F. Xiong, and G. Zhang, "TruGRC: Trust-aware group recommendation with virtual coordinators," *Future Gener. Comput. Syst.*, vol. 94, pp. 224–236, May 2019.

[32] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[33] C. C. Aggarwal, "Neighborhood-based collaborative filtering," in *Recommender Systems*. Cham, Switzerland: Springer, 2016, pp. 29–70.

[34] G. R. Lima, C. E. Mello, A. Lyra, and G. Zimbrao, "Applying landmarks to enhance memory-based collaborative filtering," *Inf. Sci.*, vol. 513, pp. 412–428, Mar. 2020.

[35] C. C. Aggarwal, "Model-based collaborative filtering," in *Recommender Systems*. Cham, Switzerland: Springer, 2016, pp. 71–138.

[36] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[37] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDD Cup Workshop*, 2007, pp. 5–8.

[38] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, pp. 426–434.

[39] S. Chen and Y. Peng, "Matrix factorization for recommendation with explicit and implicit feedback," *Knowl.-Based Syst.*, vol. 158, pp. 109–117, Oct. 2018.

[40] Y. Bao, H. Fang, and J. Zhang, "TopicMF: Simultaneously exploiting ratings and reviews for recommendation," in *Proc. 28th AAAI Conf. Artif. Intell.*, Jun. 2014, pp. 2–8.

[41] E. Khazaei and A. Alimohammadi, "Context-aware group-oriented location recommendation in location-based social networks," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 9, p. 406, Sep. 2019.

[42] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, and A. Aghasaryan, "Evaluation of group profiling strategies," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Jun. 2011, pp. 2728–2733.

[43] L. Boratto and S. Carta, "The rating prediction task in a group recommender system that automatically detects groups: Architectures, algorithms, and performance evaluation," *J. Intell. Inf. Syst.*, vol. 45, no. 2, pp. 221–245, Oct. 2015.

[44] J. C. Quan and S. B. Cho, "A hybrid recommender system based on AHP that awares contexts with Bayesian networks for smart TV," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, Cham, Switzerland: Springer, Jun. 2014, pp. 527–536.

[45] W. Wang, G. Zhang, and J. Lu, "Member contribution-based group recommender system," *Decision Support Syst.*, vol. 87, pp. 80–93, Jul. 2016.

[46] H. S. Ahmad, D. Nurjanah, and R. Rismala, "A combination of individual model on memory-based group recommender system to the books domain," in *Proc. 5th Int. Conf. Inf. Commun. Technol. (ICoIC7)*, May 2017, pp. 1–6.

[47] X. Hu, X. Meng, and L. Wang, "SVD-based group recommendation approaches: An experimental study of moviepilot," in *Proc. 2nd Challenge Context-Aware Movie Recommendation (CAMRa)*, 2011, pp. 23–28.

[48] L. N. H. Nam, H. T. H. Vy, L. H. My, L. T. T. Mai, H. T. Gia, and H. L. T. K. Nhung, "An approach to improving group recommendation systems based on latent factor matrices," in *Proc. 10th Int. Symp. Inf. Commun. Technol. (SoICT)*, 2019, pp. 98–105.

[49] F. Ortega, A. Hernando, J. Bobadilla, and J. H. Kang, "Recommending items to group of users using matrix factorization based collaborative filtering," *Inf. Sci.*, vol. 345, pp. 313–324, Jun. 2016.

[50] E. Yalcin and A. Bilge, "Novel automatic group identification approaches for group recommendation," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114709.

[51] K. Falk, *Practical Recommender Systems*. New York, NY, USA: Simon and Schuster, 2019.

[52] O. Kaššák, M. Kompan, and M. Bieliková, "Personalized hybrid recommendation for group of users: Top-N multimedia recommender," *Inf. Process. Manage.*, vol. 52, pp. 459–477, May 2016.

[53] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 119–126.

[54] L. N. H. Nam and H. B. Quoc, "The hybrid filter feature selection methods for improving high-dimensional text categorization," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 25, no. 2, pp. 235–265, Apr. 2017.

[55] A. F. Tulai and F. Oppacher, "Multiple species weighted voting—A genetics-based machine learning system," in *Proc. Genetic Evol. Comput. Conf.*, Berlin, Germany: Springer, Jun. 2004, pp. 1263–1274.

LE NGUYEN HOAI NAM received the Ph.D. degree in information systems from the VNUHCM-University of Science, Vietnam. He is currently a Researcher with the Department of Information Systems, VNUHCM-University of Science. His research interest includes recommender systems.