A similarity-enhanced hybrid group recommendation approach in cloud manufacturing systems

Computers & Industrial Engineering

Liu, Jian; Chen, Youling; Liu, Qingzhi; Tekinerdogan, Bedir

https://doi.org/10.1016/j.cie.2023.109128

Computers & Industrial Engineering 178 (2023) 109128

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A similarity-enhanced hybrid group recommendation approach in cloud manufacturing systems

Jian Liu[a,b], Youling Chen[c,*], Qingzhi Liu[d], Bedir Tekinerdogan[d]

[a] School of Management Science and Engineering, Anhui University of Technology, Maanshan 243002, China
[b] The Key Laboratory of Multidisciplinary Management and Control of Complex Systems, Department of Education Anhui Province, Maanshan 243002, China
[c] College of Mechanical and Vehicle Engineering, Chongqing University, No.174, Shazheng Street, Shapingba district, Chongqing 400044, China
[d] Wageningen University & Research, Information Technology Group, NL-6708 PB Wageningen, Netherlands

ARTICLE INFO

Keywords:
Cloud manufacturing
Group recommender systems
User similarity
Clustering
Ranking aggregation

ABSTRACT

With the development of cloud manufacturing (CMfg), a huge amount of services appears on the Internet, which makes recommender systems in CMfg service a promising research field. To this end, recent studies mainly focus on solving individual recommendation to meet the requirements of every user. However, due to the time complexity problem, 'Many-to-Many' recommendation mode is increasing in real applications. To implement such a group recommendation is very challenging, because the system not only needs to achieve high recommendation quality but also satisfies user clusters in an even way. Therefore, we propose a similarity-enhanced hybrid group recommendation approach named HGRA for cloud manufacturing. Specifically, we implement the HGRA system by three main components. Firstly, an enhanced user similarity measuring approach is designed to identify a similar user group based on the non-linear model Proximity-Significance-Singularity (PSS) and Kullback-Leibler (KL) distance algorithms. Secondly, a set of user subgroups are further clustered using K-medoids algorithm, in which additional information similarity is calculated by making full use of functional information about the users. Thirdly, a weighted ranking aggregation model is established to generate a recommendation list according to the representative user of each subgroup. The performance of our system is tested by two data sets from real-world cloud manufacturing systems. The experimental results demonstrate the feasible and effectiveness of our approach, compared with some state-of-the-art benchmark solutions, especially in CMfg systems.

1. Introduction

In recent years, cloud manufacturing (CMfg) (Li et al., 2010; Zhang et al., 2014) becomes a new paradigm due to the development of Internet and the emerging information technologies, such as Big Data (Li, Tao, Cheng, & Zhao, 2015), Cloud Computing (Xu, 2012), Internet of Things (Lu & Cecil, 2016). For manufacturing industry, global business model is gradually transformed from traditional off-line mode to platform-based and service-oriented mode in the Industry 4.0 manufacturing systems (Mourtzis, Fotia, Boli, & Vlachou, 2019; Papakostas & Ramasubramanian, 2022). The main purpose of CMfg is to collect a huge amount of distributed manufacturing resources and capabilities in cloud pool, and provide services based on these cloud data to fulfill the requirements of users. With the increasing number of services published in the CMfg systems, it is urgent to take an effective strategy to solve the large-scale service issue.

Under this circumstance, cloud service recommendation is becoming a promising research field. Recommender Systems (RSs) has been employed in various studies within some research field, such as tourism (Baker & Yuan, 2021), news (Bach, Do Hai, & Phuong, 2016), e-commerce (Mao, Lu, Han, & Zhang, 2019), etc. Inspired by the application of Web service, in CMfg environment, RSs (Azadjalal, Moradi, Abdollahpouri, & Jalili, 2017; Gohari, Aliee, & Haghighi, 2018) is mainly used to provide appropriate recommendation by discovering the subset of CMfg services among all the alternatives that are able to satisfy the active user. By doing so, users don't need to take extra time to look for specific services and make a hard decision when facing massive services on a cloud platform.

* Corresponding author.
E-mail address: youlingchencqu@126.com (Y. Chen).

https://doi.org/10.1016/j.cie.2023.109128
Received 17 August 2021; Received in revised form 23 October 2022; Accepted 23 February 2023
Available online 7 March 2023
0360-8352/

Table 1
Research position of HGRA.

Recommender systems	Web service	CMfg service
Traditional systems	Baker & Yuan, 2021 Bach et al., 2016 Mao et al., 2019 Su & Khoshgoftaar, 2009 Desrosiers & Karypis, 2011 Breese et al., 2013 Wu et al., 2012 Yu et al., 2016 Koren et al., 2009 Luo et al., 2012	Azadjalal et al., 2017 Gohari et al., 2018
Group systems	Castro et al., 2018 Garcia et al., 2011 Garcia et al., 2012 Castro et al., 2017 Seo et al., 2021 Abolghasemi et al., 2022 Nam, 2021 Lee & Kim, 2022	HGRA

Collaborative Filtering (CF) algorithm (Breese, Heckerman, & Kadie, 2013; Desrosiers & Karypis, 2011; Su & Khoshgoftaar, 2009), as a common tool and technique in the traditional RSs, has been widely applied to provide accurate recommendation for the active user. There are two popular CF algorithms: neighbor-based and model-based algorithm.

Neighbor-based CF (Wu et al., 2012), also called K-nearest neighbors (KNN) algorithm, is based on a basic idea that users who have similar historical information are likely to share similar preference. Thus, some traditional similarity measuring approaches, like Pearson Correlation Coefficient (PCC) (Yu, Wang, Zhang, & Niu, 2016) and their variants, have been widely used in neighbor-based CF to calculate user similarity. Accordingly, the effectiveness of neighbor-based CF is highly impacted by the similarity computation. However, these similarity measures mainly focus on the co-invoked information and most part of them is described by a linear way, which are not suitable for cloud recommendation in CMfg systems. So, neighbor-based CF has relatively low recommendation quality, especially in the sparsity condition.

Model-based CF is another well-known approach for cloud service recommendation, which is also called matrix factorization (MF) (Koren, Bell, & Volinsky, 2009; Luo, Xia, & Zhu, 2012). The main idea of MF-based algorithm is to train a model using the invoked information in the user-service matrix, and then learn the latent factors which are able to make a further recommendation. However, most of the MF-based algorithm assumed that each user is independent of others. It is obvious that the MF-based algorithm is a kind of objective method, and cannot achieve a good performance without considering the similarity computation, since users usually exchange their information in dynamic environment. In addition, it is often time-consuming and costly to retrain a new net-based model for different needs in CMfg systems.

Although, 'Many-to-One' traditional recommender systems are ubiquitous and do have achieved success in many applications, it is limited by the similarity computation among the users and its inner complexity. Apart from that, a large majority of traditional recommender systems usually cannot get satisfactory feedback from individual user, when the group needs and preferences are not considered. All these lead to the apply difficulty of RSs in CMfg systems.

Indeed, 'Many-to-Many' Group Recommender Systems (GRSs) (Castro, Yera, & Martinez, 2018; Garcia, Sebastia, & Onaindia, 2011) has emerged as an effective solution to help group of users to find some suitable service according to their similar needs and preferences, and thus GRSs play an important role in some specific domains, such as tourist attractions (Garcia, Pajares, Sebastia, & Onaindia, 2012). This is because GRSs not only use the similar group needs and preferences to discover the most reliable service, but also make the recommendation list diverse as the differences always exist. So far, many CF-based techniques have been used in GRSs to improve group recommendation quality by integrating user individual information (Castro, Yera, & Martínez, 2017), and try to satisfy group users to the greatest extent. As a result, more and more researchers focus on the preferences-based model (Abolghasemi, Engelstad, Herrera-Viedma, & Yazidi, 2022; Seo, Kim, Lee, & Kim, 2021) from both user and service side, or extract a latent interest of user group in profile aggregation way (Nam, 2021). Furthermore, deep neural network (Lee & Kim, 2022), as a new learning tool, has also been involved in the process of GRSs.

However, at the background of CMfg platform, the characteristics of group users are various, like inconsistent preferences, information interaction, complex content of request with field terms. So, it is very challenging to achieve effective group recommendation of high satisfactory without considering them. That is one of the research gaps of GRSs in CMfg systems. The research position of this paper is presented in Table 1.

The disadvantages of both the traditional recommender systems and group recommender systems greatly inspire the design of our system. To implement the group recommendation of CMfg service, we focus on the development of a new group recommender systems, called similarity-enhanced hybrid group recommendations approach (HGRA). In HGRA, a new user similarity measure is proposed to identify similar group of users. After that, another information similarity measure is proposed to make a clustering of user subgroup based on the K-medoids algorithm. Then, a weighted ranking aggregation model based on the Collaborative Filtering is employed to generate the group recommendation list. The main contributions of this paper are summarized as follows.

(1) To overcome the low accuracy and stability of traditional similarity computation, we introduce a hybrid similarity model for discovering similar group in different condition, and apply this model into our HGRA approach to help subsequent group recommendation.
(2) To improve recommendation quality and averaged satisfaction, we also propose a K-medoids clustering algorithm and weighted ranking aggregation model as off-line group recommendation modules for studying the characteristics of CMfg service in 'Many-to-Many' mode. We also integrate these three modules into a group recommender system, and seems that the idea of our HGRA could alleviate the recommendation problem in CMfg systems.
(3) We conduct a series of experiments on two real-world data sets to verify the effectiveness of the proposed approach, especially in CMfg systems.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 introduces the problem statement. Section 4 presents the details about HGRA. Section 5 presents the experiments and results based on the proposed approach. Finally, we conclude our work in Section 6.

2. Related work

Group Recommender Systems (GRSs), as an important tool and technique in Cloud Manufacturing domain, have aroused a great deal of interests in both academia and industry. To the best of our knowledge, GRSs consists of three main steps. The first step is to calculate the user similarity which is used to identify similar group of users. The second step is that clustering the similar users into several subgroup according to the distance measurement. The last step is to generate ranked recommendation list for a group of users by utilizing the aggregation technique. In this section, to place our work in a state-of-art context, we give a brief review of the related work on the main steps in GRSs.

2.1. Similarity-based approaches

Similarity measure is a popular tool which is usually used to identify a set of similar objects in recent studies. In both user-service side, the basic idea of the similarity measure is that ones who have a high similarity, then they may share same preference or function. For example, Ekstrand, Riedl, and Konstan (2011) integrated PCC into CF-based approach to calculate the linear correlation between a pair of services. Then, Feng and Huang (2020) presented an extended PCC similarity model to make quality-of-service (QoS) prediction in CMfg systems. Similar to PCC, cosine similarity is another widely used similarity measure. Ahn (2008) presented an adjusted cosine similarity measure and a new similarity called PIP (Proximity-Impact-Popularity) for personalized recommendation.

To improve the performance of similarity measure, many researchers have designed some novel similarity measures recently. For instance, Liu, Hu, Mian, Tian, and Zhu (2014) designed a non-linear similarity model called PSS (Proximity-Significance-Singularity) to overcome the linear-relationship problem, which not only considers the local context information of user ratings, but also the global preference of user behavior. Although the experimental results demonstrated the superiority of PSS model, it still cannot be used independently without enough co-invoked ratings. To alleviate the problem, Wang, Deng, Gao, and Zhang (2017) first consider the vectors in a row-column way, and employ Kullback-Leibler (KL) distance to correct the output in recommender systems. By doing this, it seems to be more effective than the other similarity measures. But this novel idea hasn't been proved in CMfg systems yet.

2.2. Clustering-based algorithms

Clustering-based algorithm, as a well-known unsupervised classification method, whose aim is to reduce the search space and discover similar group of patterns (users, services, etc.) (Ghazanfar & Prügel-Bennett, 2014). In general, there are three main clustering-based algorithms being applied in recommender systems: K-means algorithm, K-medoids algorithm (Guo, Zhang, & Yorke-Smith, 2015) and Hypergraph partition algorithm (Selvitopi, Turk, & Aykanat, 2012).

The basic idea of K-means clustering algorithm is that finding a center of pattern by averaging all the values of each attribute. For example, Ghazanfar and Prügel-Bennett (2014) presented a classic K-means clustering algorithm which is used to decrease the recommendation error by identifying the similar users. Different from K-means, K-medoids clustering algorithm selects a real user as the center, which makes the similarity measure can be directly used to calculate the distance. It seems to be more understandable and easier to implement. Thus, Guo et al. (2015) proposed K-medoids clustering algorithm to make a multi-view clustering, in which the personalized influence of

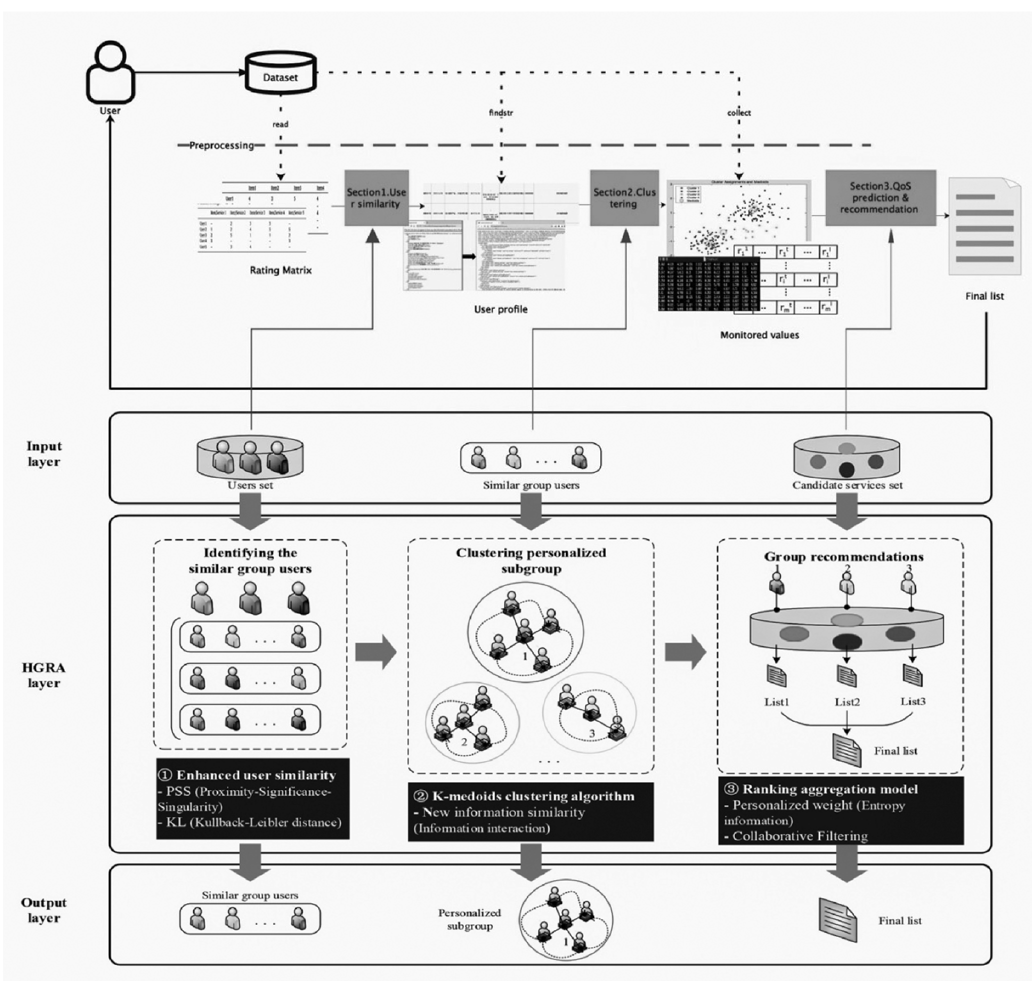

Fig. 1. Overview of the main steps of HGRA.

group users can be preserved to some extent. Hypergraph partition algorithm is a new clustering method in current RSs and has been widely applied to many large-scale applications (Wu, He, & Han, 2014). For example, Yang, Wang, Bhuiyan, and Choo (2017) designed a novel vertex hypergraph partitioning method, called EQHyperpart, to achieve high quality clustering results by integrating all the entropy information. Although demonstrated to be more effective and scalable to large-scale datasets, it is very time-consuming to make a clustering with high complexity.

2.3. Aggregation-based model

Apart from similarity-aware model and clustering-based algorithm, aggregation-based model is also a critical part to complete the whole process in CMfg systems. In general, aggregation-based models in GRSs are either based on the generation of an integrated group profile or on the integration of recommendations built for every user independently (Baltrunas, Makcinskas, & Ricci, 2010). In other words, there are two typical aggregation-based models: Preference aggregation model (Wang, Zhang, & Lu, 2017) and Ranking aggregation model (Baltrunas et al., 2010). In recent studies, Yu, Zhou, Hao, and Gu (2006) took advantage of the first model by merging the individual preferences to obtain a group profile for group recommendation. However, the most important decision-making process regarding the group users with different characteristics are not clear. Baltrunas et al. (2010) employed ranking aggregation model to make a group recommendation, in which all the individual recommendation are integrated into a final list. In addition, Seo, Kim, Lee, Seol, and Baik (2018) also presented an enhanced aggregation method, called upward leveling (UL), to help the group recommendation. During the aggregation process, there are some strategies in the ranking model, such as Averaged, Least Misery, etc. However, the biggest limitation is that recommendation generated with ranking aggregation model seldom satisfy the group users in an even way, namely, the personal influence of users in the group has been ignored. As far as we are concerned, an effective group recommendation approach that not only matches the group's preferences, but also takes the averaged satisfaction of group users into consideration.

Based on the above analysis, in this paper, we develop a hybrid group recommendation approach to improve the recommendation quality and averaged satisfaction. To identify a similar user group, an enhanced user similarity is designed by combining the Proximity-Significance-Singularity (PSS) and Kullback-Leibler (KL) distance. Then, we integrate a new information similarity into K-medoids clustering algorithm to search for the subgroup. After that, we also employ a weighted ranking aggregation model to generate the recommendation lists considering the entropy information among the representative users in each subgroup. Finally, we conduct a series of experiments on two real-world data sets to verify the effectiveness of HGRA.

3. Problem statement

Generally, a GRSs correlates to the historical interactions between users and services in CMfg systems. The main purpose of GRSs is not only about higher recommendation quality, but also about more friendly to user group. So, in this work, we aim to cope with the following two questions: (1) how to improve the recommendation quality in GRSs; (2) how to achieve a higher averaged satisfaction based on the group recommendation list.

In our recommendation scenario, GRSs include a lot of users and services, in which users have various needs and preferences, while services contain different functional and nonfunctional information. Thus, recommendation quality mainly depends on the accuracy of similarity computation and specific recommendation strategy. Different from traditional systems, averaged satisfaction in GRSs is greatly affected by the quality of user subgroup, which is further clustered based on the similar user group. Also, aggregation mode is another impact factor according to the personalized needs and preferences in a cluster. Therefore, our overall framework can be separated into 3 sections: (1) user similarity; (2) clustering; (3) QoS prediction & recommendation.

For simplicity, we should comply with the following assumptions:

(1) Each user has a historical recommendation list;
(2) Each user has a specific need and preference;
(3) Each user can be satisfied by a service when there is an interaction between them;
(4) Each service interacted with at least one user before.

4. Hybrid group recommendation approach

In this section, main steps of the proposed similarity-enhanced hybrid group recommendation approach (HGRA) are presented, including: (1) Identifying the similar group users, (2) Clustering personalized subgroup based on the K-medoids algorithm, (3) Weighted ranking aggregation model for group recommendation. The overview of the proposed approach and the overall recommendation process of GRSs in CMfg systems are illustrated in Fig. 1, and the details of the main steps are described in the following sections.

4.1. User similarity

To identify a similar user group, we develop a new user similarity by combing the non-linear model Proximity-Significance-Singularity (PSS) and Kullback-Leibler (KL) distance, in which all the information in the user-service matrix can be explored as much as possible. And the details of these two concepts are described in the following subsections.

4.1.1. PSS model

Firstly, to better calculate the non-linear relationship between users, we employ the Proximity-Significance-Singularity (PSS) model which not only computes the similarity based on the co-invoked services, but also generates some specific variables in other dimensions. Here the PSS model is defined as follows:

$$S_1(r_{us}, r_{vs}) = \text{Proximity}(r_{us}, r_{vs}) \times \text{Significance}(r_{us}, r_{vs}) \times \text{Singularity}(r_{us}, r_{vs})$$

$$\begin{cases} \text{Proximity}(r_{us}, r_{vs}) = 1 - \dfrac{1}{1 + exp(-|r_{us} - r_{vs}|)} \\ \text{Significance}(r_{us}, r_{vs}) = \dfrac{1}{1 + exp(-|r_{us} - \overline{r_u}|\cdot|r_{vs} - \overline{r_v}|)} \\ \text{Singularity}(r_{us}, r_{vs}) = 1 - \dfrac{1}{1 + exp\left(-\left|\dfrac{r_{us} + r_{vs}}{2} - \overline{r_s}\right|\right)} \end{cases} \tag{1}$$

where $S_1(r_{us}, r_{vs})$ is the similarity between user u and user v computed based on the co-invoked service s, r_{us} is the rating value, $\overline{r_u}$ is the averaged rating value rated by u, and $\overline{r_s}$ is the averaged rating value of s.

Here, function Proximity is computed according to the absolute difference between r_{us} and r_{vs}, while function Significance aims to measure the impact of the rating pair to the similarity value. And function Singularity means the difference between one rating pair to other ratings.

Then, the similarity between user u and user v based on the PSS model is defined as below:

$$S_1(u, v) = \frac{\sum_{s \in S} S_1(r_{us}, r_{vs})}{|S|} \tag{2}$$

where $S = S_u \cap S_v$ represents the set of services that are co-invoked by both u and v. If they are similar, then the value of $S_1(u,v)$ are larger than an expected value.

Although the non-linear PSS model seems to be an important tool to calculate the user similarity, yet it still cannot be employed independently. The reason is that the PSS model mainly focuses on the co-invoked part, however, it is impossible that there are always exist co-invoked services in real applications.

4.1.2. KL distance

Kullback-Leibler (KL) distance is another similarity measure that considers the difference between two sequences from the perspective of probability distribution (Kullback & Leibler, 1951). Thus, it can alleviate the co-invoked problem to some extent, and fully utilize all the ratings between two users. Considering the fact that KL distance is asymmetric, the KL distance between user u and user v is calculated as follows:

$$\begin{cases} D(u,v) = \dfrac{D(u||v) + D(v||u)}{2} \\ D(u||v) = D(p_u||p_v) = \displaystyle\sum_{i=1}^{r} p_{ui} log_2 \frac{p_{ui}}{p_{vi}} \end{cases} \tag{3}$$

where$D(u||v) \neq D(v||u)$, r is the maximum value in rating scale, and $p_{ui} = \frac{|i|}{|u|}$ represents the probability of rating value i on user u, in which $|u|$ denotes the number of all services rated by user u and $|i|$ is the number of rating value i.

It is obvious that the larger the KL distance is, the less similar the two users are, and vice versa. Thus, the similarity measure based on the KL distance is defined as follows:

$$S_2(u,v) = \frac{1}{1 + D(u,v)} \tag{4}$$

Based on the PSS model and the KL distance, we can obtain a new user similarity measure, and the formula is defined as below:

$$S(u,v) = \lambda \times S_1(u,v) + (1-\lambda) \times S_2(u,v) \tag{5}$$

where $S(u,v)$ represents the user similarity between user u and user v. $\lambda (0 < \lambda < 1)$ is a parameter to determine how much the user similarity relies on $S_1(u,v)$ and $S_2(u,v)$, and the sensitivity of λ will be studied in the experiment section.

4.2. Clustering personalized subgroup

After the similar user group has been identified, in this section, we aim to make a clustering based on K-medoids algorithm to obtain several personalized subgroups. And the key process of clustering is described in the following subsections.

4.2.1. Information similarity

Before introducing the K-medoids clustering algorithm, we first propose another similarity measure, called information similarity, to identify the personalized subgroups. Since there are multiple functional information (regular requirements and personalized needs) involved in the users' profile, it is very challenging to discover same-class user subgroup from the similar users, especially in CMfg systems. In this subsection, the information similarity is enhanced to further calculate the similarity between users by integrating the information interactions. And the information similarity between user u and user v is defined as follows:

$$\mathrm{Sim}(u,v) = 1 - \sqrt{\sum_{j=1}^{f} w_j(u,v) \cdot D\left(FuncInfor_j(u) - FuncInfor_j(v)\right)^2} \tag{6}$$

where f is the dimension of functional information, $w_j(u,v)$ is the weight distribution vector corresponding to the functional information.

In this paper, we use the frequency of information interactions to measure the weights. And, a log function (Seo, Kim, Lee, & Baik, 2017; Vosecky, Leung, & Ng, 2014) is also employed to measure the frequency, and the formula is as follows:

$$F_j(u,v) = \begin{cases} log_{10}\left(1 + q_j(u,v)\right), & if \quad q_j(u,v) < 10 \\ 1, & if \quad q_j(u,v) \geq 10 \end{cases} \tag{7}$$

where $F_j(u,v)$ is the frequency of jth information between user u and user v, and $q_j(u,v)$ denotes the number of information interactions.

Based on the frequency of information interactions, the weights are shown as follows:

$$w_j(u,v) = \frac{F_j(u,v)}{\sum_{j=1}^{f} F_j(u,v)} \tag{8}$$

$D\left(FuncInfor_j(u) - FuncInfor_j(v)\right)$ is the attribute quantization distance (Xiang, Jiang, Xu, & Wang, 2016) of functional information, in which $FuncInfor_j(u)$ represents the jth functional information of u. And the formula is as follows:

$$D\left(FuncInfor_j(u) - FuncInfor_j(v)\right) = \frac{|FuncInfor_j(u) - FuncInfor_j(v)|}{|FuncInfor_{\max} - FuncInfor_{\min}|} \tag{9}$$

where $|FuncInfor_{\max} - FuncInfor_{\min}|$ is the charge range of the threshold. A noteworthy point is that the calculation of attribute quantization distance must be executed in same type of items, otherwise, the distance is maximum.

4.2.2. K-medoids clustering algorithm

In this subsection, we integrate the information similarity into K-medoids clustering algorithm to identify the personalized subgroup. Different from the existing clustering algorithm, K-medoids clustering algorithm considers a real user as the center and thus preserve the group characteristics to some extent (Guo et al., 2015). Thus, K-medoids is very suitable for grouping users who have complex features in CMfg systems. And the objective function is presented as follows:

$$\begin{cases} J = min \displaystyle\sum_{c \in C} \sum_{u,v \in c} d(u,v) \\ d(u,v) = 1 - Sim(u,v) \end{cases} \tag{10}$$

where C is a set of user subgroup, user u and user v belong to the subgroup $c \in C$, and $d(u,v)$ denotes the distance between u and v.

In order to discover the personalized subgroup, information similarity is used as the distance metric to cluster the similar users. In other words, the higher information similarity the users have, the closer the users are located. The pseudocode of K-medoids clustering algorithm is presented in Algorithm 1.

Algorithm 1. K-medoids clustering algorithm

Input: distance matrix D_t; cluster number K

Output: user subgroups C

$p \leftarrow 0$;

randomly select K medoids m_t from users, $\Theta_t^0 \leftarrow m_t$; /* Initialize medoids */

$C_t^0 \leftarrow u$, given $\min(d(u, m_t))$; /* Calculate the distance by using Eq. (10) */

while medoids changed **and** $< maxiterations$ **do**

  $p \leftarrow p+1$;

  $\Theta_t^p \leftarrow \Theta_t^{p-1}$;

  $swap(m_t, v)$, $v \in C_t^{p-1}$;

  calculate $sum_t(v) = \sum_u d(u, v)$, $u \in C_t^{p-1}$;

  if $sum_t(v) < sum_t(m_t)$ **then** /* Update medoids */

    $m_t \leftarrow v$;

    $\Theta_t^p \leftarrow m_t$;

  $C_t^p \leftarrow u$, for $\forall u$, find m_t $s.t.$ $min(d(u, m_t))$; /* Cluster assignment */

return $C \leftarrow C_t^p$;

4.3. Ranking aggregation for group recommendation

After the process of K-medoids clustering, we can obtain K representative users in the subgroups. In this section, we first utilize Collaborative Filtering to rate all the candidate service from the perspective of QoS prediction. Then, we establish a weighted ranking aggregation model to make a Top-N group recommendation.

All the candidate services can be predicted with these K representative users, and the rating value is calculated as follows:

$$R_{u_k}(CS_i) = \begin{cases} R'_{u_k}(CS_i), & \text{if } CS_i \text{ has been rated by } u_k \\ \overline{R_k}(CS_i) + \dfrac{\sum\limits_{u_a \in c(u_k)} Sim(u_k, u_a) \times (R_{u_a}(CS_i) - \overline{R_a})}{\sum\limits_{u_a \in c(u_k)} Sim(u_k, u_a)}, & others \end{cases} \tag{11}$$

where $R_{u_k}(CS_i)$ denotes the rating value of ith candidate service rated by kth representative user (u_k), while $R'_{u_k}(CS_i)$ denotes the authentic rating value if CS_i has been invoked by u_k. $\overline{R_k}(CS_i)$ denotes the averaged rating value among the users in kth subgroup, like u_a.

To improve the recommendation accuracy and averaged satisfaction, we design a new ranking aggregation model by taking the personalized influence of these representative users into consideration, which called entropy information (Feng & Huang, 2020). And the entropy of u_k is defined as:

$$E(u_k) = -\frac{1}{ln(|S^*|)} \sum_{s \in S^*} p_{u_k s} \times ln(p_{u_k s}) \tag{12}$$

where S^* is the set of service that has been invoked by u_k, and $p_{u_k s} = \frac{R_{u_k}(s)}{\sum_{s \in S^*} R_{u_k}(s)}$.

Based on the entropy information, we can compute the weight of K representative users as the personalized influence, and the formula is as

Table 2
The complexity of HGRA.

(1) Similarity computation	(2) Clustering algorithm	(3) Ranking and group recommendation	(4) HGRA
$O(m^2n)$	$O(mn)$	$O(mn)$	$O(m^2n)$

Table 3
Experimental parameters.

Parameters	Web service	CMfg service
f: the dimensional number of functional information	20	50
n: the number of candidate services	500	500
λ: a parameter determines the user similarity	0.5	0.5
K: maximum cluster number	6	8
N: the number of group recommendation	30	30

follows:

$$W_k = \frac{1 - E(u_k)}{K - \sum_{k=1}^{K} E(u_k)} \tag{13}$$

Then, our weighted ranking aggregation model can be established to satisfy the maximum possible number in user clusters according to the rating value from K representative users. And the formula is presented as below:

$$R(CS_i) = \sum_{k=1}^{K} W_k \times R_{u_k}(CS_i) \tag{14}$$

where $R(CS_i)$ represents the rating value of CS_i based on the weighted ranking aggregation model.

To sum up, in this study, we first identify the similar users group using a new user similarity measure (See in Section 4.1). Then, we integrate an enhanced information similarity into K-medoids clustering algorithm to discover the personalized subgroup (See in Section 4.2). After that, we also utilize Collaborative Filtering and weighted ranking aggregation model to obtain a final ranking list of all candidate services (See in Section 4.3). Finally, Top-N group recommendation can be generated according to the final rating results.

4.4. Time complexity

The computational complexity is another critical metric to show the performance of HGRA, in this paper, the total complexity can be accordingly divided into three phases: (i) Similarity computation, (ii) Clustering algorithm, (iii) Ranking and group recommendation.

There are two kind of similarity measure during the process of similarity computation, namely, user similarity and information similarity. Firstly, we assume that there are m users and n services, and the computation time of user similarity is around $O(mn(m-1))$. Then, we need to further identify the subgroup using the enhanced information similarity in Section 4.2.1, and the computation time is around $O(mn(f-1))$. Considering the fact that f is the constant number refers to dimension of functional information, and the total complexity of similarity computation is around $O(mn(m-1)) + O(mn(f-1)) \approx O(m^2n)$.

For K-medoids clustering algorithm, the computation time is $O(pKmn)$, where p is the maximum number of iterations, and K is the number of user subgroup. Since p and K are constant, the total complexity is around $O(mn)$.

As for the final ranking process, we first calculate the entropy information of K representative users, and the computation time is around $O(Kmn)$. Then, Top-N group recommendation is listed based on the ranking aggregation model, and the complexity is about $O(Nmn)$. Because of K and N are also constant value, the computation time is around $O(Kmn) + O(Nmn) \approx O(mn)$.

Based on the above analysis, the total computational complexity of HGRA can be concluded in Table 2. As we can see, our time complexity is still in a reasonable range compare to others. However, in a long-term view, HGRA may shows a lower computational cost due to GRSs mode.

5. Experiment

In this section, we present an empirical study of our approach HGRA on two real-world data sets. The experimental environment and data sets are introduced in Section 5.1.1 and Section 5.1.2. And the metrics employed to evaluate the performance of HGRA are shown in Section 5.1.3. The performance comparison from the proposal and baseline methods and discussion are presented in Section 5.2. The sensitivity analysis is also studied in Section 5.2.1.

5.1. Experiment setup

5.1.1. Environment

Note that the experimental environment in this paper is constructed in an off-line way. To carry out the verification, all the experiments are run on a computer with Quad-Core Intel Core i5 2GHz CPU processors and 16GB RAM. As for the software of HGRA development, we mainly apply MATLAB R2019b in this part to conduct relevant experiments.

5.1.2. Data sets

In our experiments, we use two different real-world data sets which are employed in individual recommendation scenario. One is classic Web service data set, called WSDream, which is collected from 339 users on 5825 services by Zheng, Ma, Lyu, and King (2010). Inspired by Xiang et al. (2016), another is private CMfg service data set collected from the relative machinery industry platform, which contains about 463 users and 7548 services. It is note that all these data sets have updated some additional information to meet the experimental requirements, especially CMfg services, such as personalized needs of user and nonfunctional information of service.

For more realistic, different baseline methods are also employed to demonstrate the effectiveness of our approach. Although different characteristics of Web service and CMfg service, HGRA can be applied to both scenarios without any modification. The experimental parameters are summarized in Table 3.

5.1.3. Metrics

To evaluate our approach HGRA based on a list of recommendations, in this subsection, we use both normalized discounted cumulative gain (nDCG) (Seo et al., 2018) and group satisfaction (GS) (Villavicencio, Schiaffino, Diaz-Pace, & Monteserin, 2019) as the metrics in two data sets. Recently, nDCG has been widely used to measure the effectiveness of group recommendation approaches. The main reason is that it not only considers the prediction accuracy but also takes the recommendation quality into account. More specifically, nDCG is easily measured based on a key rule: more high-ranked services in the list.

The nDCG metric is defined as:

$$\begin{cases} DCG_{u,N} = R_{u,l_1} + \sum_{i=2}^{N} \frac{R_{u,l_i}}{log_2(i)} \\ nDCG_{u,N} = \frac{DCG_{u,N}}{IDCG_{u,N}} \end{cases} \tag{15}$$

where $l_1 \cdots l_N$ denote the recommendation lists, and $DCG_{u,N}$ measures the accuracy of a list of recommendation that is ordered by ratings. And $IDCG_{u,N}$ is the optimal possible gain value for users where the lists are reordered by ratings.

Different from nDCG, group satisfaction (GS) is measured to show the averaged satisfaction degree of the group users when a list of services being recommended to them. And the GS metric is defined as:

Table 4
nDCG of Web service.

Methods	Group Size					
	5	10	15	20	25	30
EPCC	0.470	0.493	0.520	0.554	0.596	0.648
PSS	0.503	0.533	0.568	0.609	0.657	0.713
H-US	**0.507**	**0.538**	**0.572**	**0.614**	**0.660**	**0.722**
Our model	**0.518**	**0.548**	**0.579**	**0.629**	**0.682**	**0.741**
vs H-US (%)	**2%**	**2%**	**1%**	**2%**	**3%**	**3%**

Table 5
nDCG of CMfg service.

Methods	Group Size					
	5	10	15	20	25	30
EPCC	0.492	0.514	0.543	0.575	0.611	0.652
PSS	0.530	0.566	0.607	0.654	0.708	0.771
H-US	**0.536**	**0.574**	**0.618**	**0.669**	**0.722**	**0.788**
Our model	**0.543**	**0.582**	**0.621**	**0.676**	**0.727**	**0.793**
vs H-US (%)	**1%**	**1%**	**0.5%**	**1%**	**1%**	**1%**

Table 6
nDCG of Web service.

Methods	Group Size					
	5	10	15	20	25	30
K-Mean	0.458	0.479	0.502	0.535	0.566	0.607
K-Mediods	**0.496**	**0.525**	**0.554**	**0.589**	**0.641**	**0.698**
EQHyperpart	0.472	0.496	0.523	0.561	0.602	0.654
RACF	0.437	0.456	0.479	0.503	0.530	0.567
UL	**0.505**	**0.534**	**0.563**	**0.602**	**0.646**	**0.708**
HGRA	**0.518**	**0.548**	**0.579**	**0.629**	**0.682**	**0.741**
vs K-Mediods (%)	**4%**	**4%**	**5%**	**7%**	**6%**	**6%**
vs UL (%)	**3%**	**3%**	**3%**	**5%**	**6%**	**5%**

Table 7
nDCG of CMfg service.

Methods	Group Size					
	5	10	15	20	25	30
K-Mean	0.475	0.496	0.523	0.554	0.586	0.619
K-Mediods	**0.525**	**0.559**	**0.601**	**0.648**	**0.702**	**0.765**
EQHyperpart	0.493	0.516	0.545	0.578	0.614	0.656
RACF	0.453	0.478	0.502	0.528	0.565	0.612
UL	**0.532**	**0.569**	**0.611**	**0.658**	**0.712**	**0.771**
HGRA	**0.543**	**0.582**	**0.621**	**0.676**	**0.727**	**0.793**
vs K-Mediods (%)	**3%**	**4%**	**3%**	**5%**	**4%**	**4%**
vs UL (%)	**2%**	**2%**	**2%**	**3%**	**2%**	**3%**

$$\begin{cases} GS(l_i) = \dfrac{\sum\limits_{GroupSize} S(l_i)}{GroupSize} \\[2ex] GS = \dfrac{\sum\limits_{i=1}^{N} GS(l_i)}{N} \end{cases} \tag{16}$$

where $GroupSize$ is the number of users in a group, $S(l_i)$ is the satisfaction degree of l_i measured by the interactions of each user in a group, and $GS(l_i)$ is the group satisfaction degree of l_i. We assumed that a service would satisfy the user if there is a interaction between them.

Without loss of generality, we repeat the experiment for ten times to avoid possible inconclusiveness caused by randomness. And the evaluation metrics nDCG and GS are employed to show the performance of HGRA, where higher nDCG and GS values mean better group recommendation performance.

5.2. Performance comparison

To make a better performance comparison, we compare HGRA with other baseline methods, including the following:

EPCC: EPCC (Feng & Huang, 2020) is an enhanced PCC similarity model to make QoS prediction in CMfg systems, in which neighborhood information has been extracted.

PSS: PSS (Proximity-Significance-Singularity) (Liu et al., 2014) is a non-linear similarity model to overcome the linear correlation problem, which not only considers the local context information of user ratings, but also the global preference of user behavior.

H-US: H-US (Wang, Deng, et al., 2017) is a hybrid user similarity model to improve prediction accuracy and recommendation quality, in which KL distance has been first integrated.

Our model: We propose an enhanced user similarity model in this study (See in Section 4.1).

K-Mean: K-Mean clustering algorithm (Ghazanfar & Prügel-Bennett, 2014) is a classic method which is designed to identify the similar users and could decrease the recommendation error.

K-medoids: K-medoids clustering algorithm (Guo et al., 2015) is also presented to make a recommendation, in which the PCC similarity model is integrated to measure the inter-distance among the users.

EQHyperpart: EQHyperpart (Yang et al., 2017) is a novel vertex hypergraph partitioning method that is able to achieve high quality clustering result based on information entropy modularity.

RACF: RACF (Baltrunas et al., 2010) is a common group recommendation algorithm based on the rank aggregation model and Collaborative Filtering.

UL: UL (Seo et al., 2018) is an enhanced aggregation method, called upward leveling, which considers deviations for group recommendation.

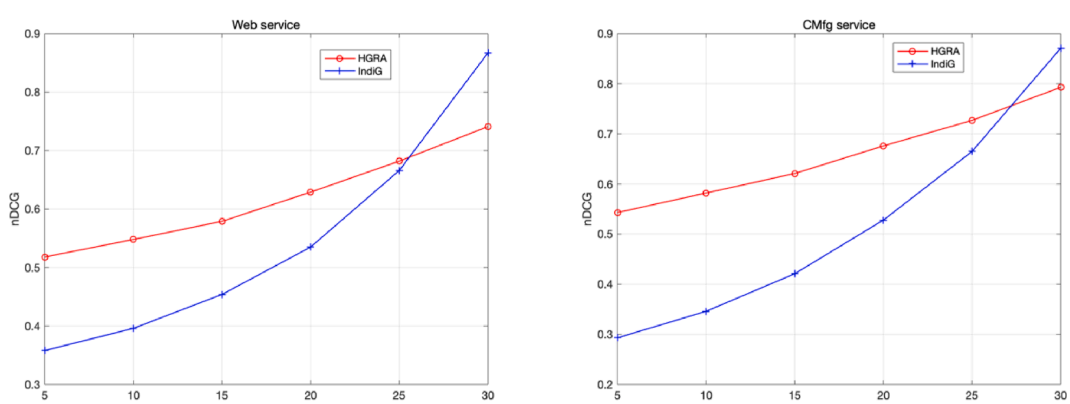

Fig. 2. Performance comparison between HGRA and IndiG.

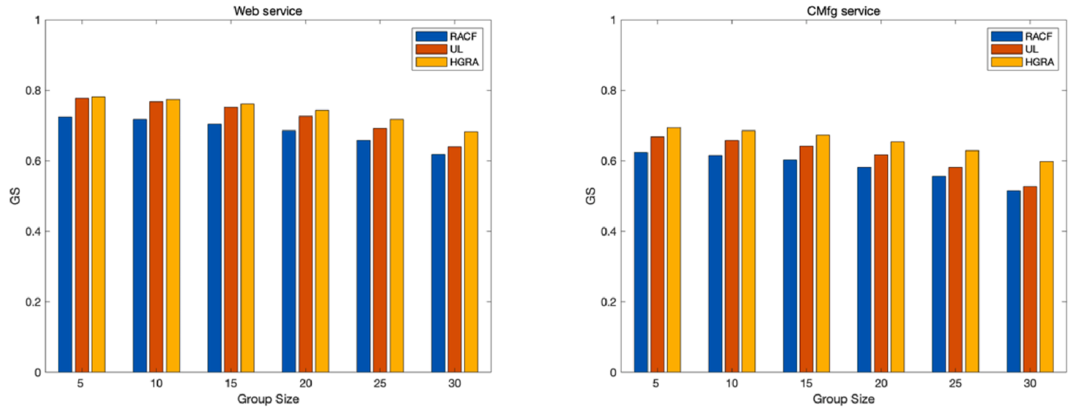

Fig. 3. Performance comparison of group satisfaction.

Table 8
Qualitative analysis of different GRSs.

Method	Domain	Datasets	Scalability
RACF	Movie	Large [MovieLens100k (943*1682)]	
UL	Movie	Large [MovieLens100k (943*1682)]	
HGRA	Manufacturing industry	Large [WSDream(339*5825) / CMfg(463*7548)]	√

IndiG: IndiG (Lee & Kim, 2022) is an inductive learning method, which combines both Bayesian modeling and deep neural networks in group recommendation.

HGRA: HGRA is a similarity-enhanced hybrid group recommendation approach proposed in this paper, including an enhanced user similarity model, new information similarity measure, K-medoids clustering algorithm and weighted rank aggregation model (See in Section 4).

5.2.1. nDCG

In this part, we vary the group size from 5 to 30 with a step value of 5 to study the group prediction accuracy and recommendation quality of our approach in terms of nDCG. Table 4 and Table 5 show the results of different similarity-based methods in two data sets, respectively. In addition, the nDCG results of different group recommendation methods are also illustrated in Table 6, Table 7 and Fig. 2 for a performance comparison.

We employ different user similarity measures in our HGRA, such as EPCC and PSS, and the group recommendation results can be found in Table 4 and Table 5, in which the nDCG value of all methods increase as the group size goes up. It is obvious that our enhanced user similarity has a better performance than other baseline methods. Although the H-US shows a great performance in both data sets, our user similarity model still has a relative higher nDCG value in different condition of group size. This is because our model takes the advantage of both PSS and KL distance in a more reasonable way, and could get more accurate similarity with less co-invoked entry.

The experimental results of Table 6, Table 7 and Fig. 2 show that:

(1) In these two Tables, all the group users are identified by our user similarity model. For the clustering-based approaches, we can find that our HGRA has a slight advantage compared with other baseline methods, like K-Mean and Hypergraph-based algorithm. K-Mediods clustering algorithm does have a great performance, but information similarity as an effective distance measure is totally ignored by Guo et al. (2015).
(2) It is obvious that the nDCG value of HGRA is relative higher than two classic aggregation-based models RACF and UL. This is because HGRA make a ranking aggregation from perspective of K representative users, instead of a single user's preference or all the users' profile. In addition, in HGRA, we also consider the personal influence of K representative users, namely, entropy information. By doing this, the weighted ranking aggregation model can make a contribution to the group recommendation to some extent.
(3) In Fig. 2, we can observe that the nDCG value of HGRA is relative higher than IndiG in most condition, due to the implement of understandable model. But as group size goes up, the improvement of IndiG is faster than HGRA, especially when group size =

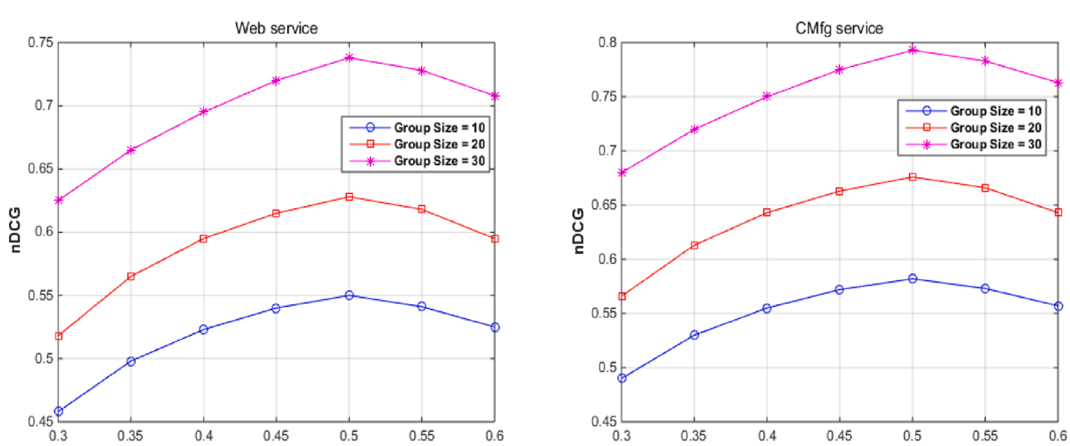

Fig. 4. Impact of λ.

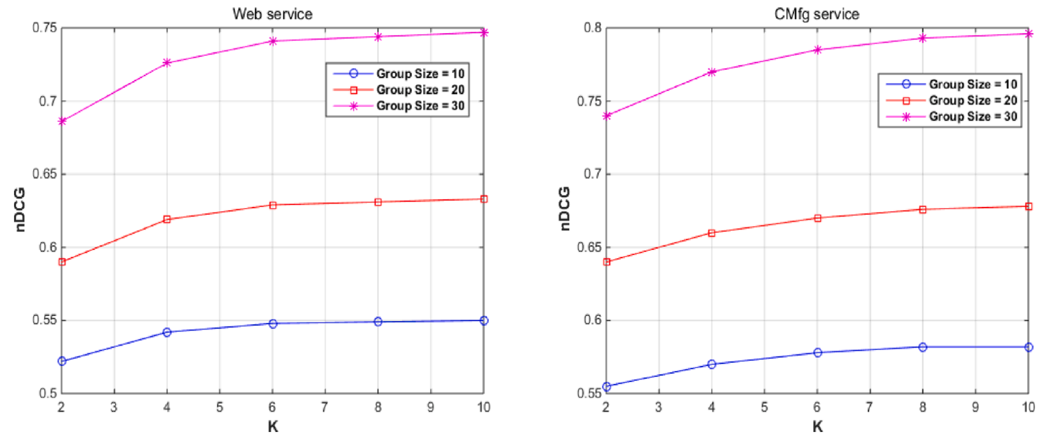

Fig. 5. Impact of K.

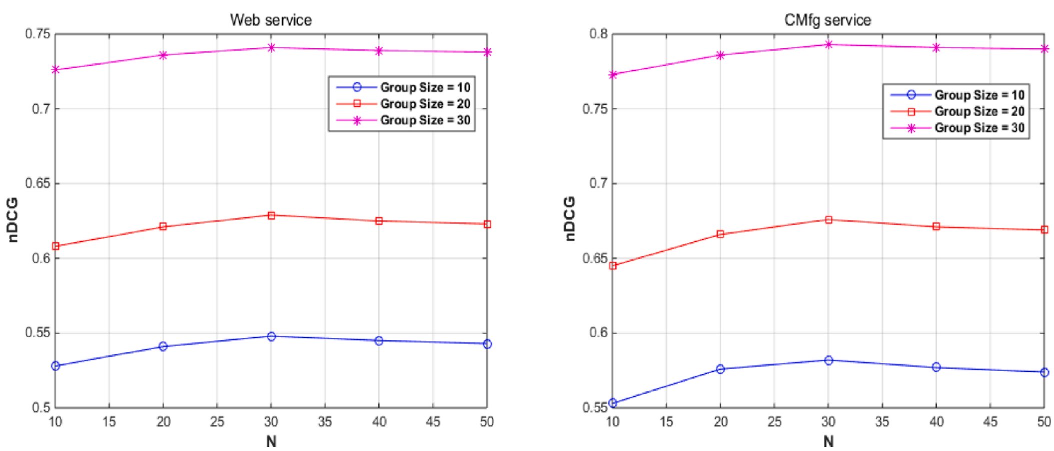

Fig. 6. Impact of N.

30. The reason is that Bayesian inductive learning method can extract more latent information from a larger group, and thus get a better inductive train model. However, deep neural network-based methods, like IndiG, urge more and more information with high dimension, and also take a large amount of time to complete the train process. In contrast, HGRA achieves a stable performance and is more suitable to fine-grained CMfg systems.

(4) To sum up, the HGRA approach significantly outperforms the clustering-based algorithms and the ranking aggregation-based methods for group recommendation of both web service and CMfg service under different group size. This is due to that, our approach is like a systematic method which considers every part of group recommendation in CMfg systems, and takes the advantage of an enhanced user similarity model, new information similarity measure, K-medoids clustering algorithm and weighted rank aggregation model to achieve a superior performance than each single one.

5.2.2. GS

In addition to nDCG, we employ another important metric to evaluate the performance of group recommendation, namely, group satisfaction (GS). In the Fig. 3, we can see that HGRA shows a highest GS value with the weighted ranking aggregation model. As the group size increases, we can also find that the higher the group size is, the GS value becomes lower, which indicates that it is hard to satisfy the larger group users, especially in CMfg systems.

In order to make a classification, Table 8 represents the qualitative analysis of these three GRSs in terms of scalability. It seems to be that our HGRA performances more friendly and could be applied to other domains since others haven't prove it yet. This is because our three-sections structure could be composed to work in most scenarios.

5.3. Sensitivity analysis

In this section, we aim to explore the effects of different parameters on the performance of our approach HGRA. In general, we vary the value of a specific parameter while holding the others consistent in Table 3, then study the nDCG results under different group size. And the detail analysis process is presented below.

5.3.1. Impact of λ

In the HGRA, λ is a critical parameter that determines the accuracy of user similarity and then affects the quality of group recommendation. To study the impact of λ, we carry on a set of experiments under three different group size of 10, 20 and 30, respectively. And the other parameters are same as Table 3.

As shown in Fig. 4, our HGRA achieves the highest nDCG value of both Web service and CMfg service when $\lambda = 0.5$, indicating that the optimal point of λ is not influenced by the group size. Considering the fact that $\lambda (0 < \lambda < 1)$ is a parameter to determine how much the user similarity relies on the non-linear PSS model and KL-distance. Since when $\lambda < 0.5$, the effect of the non-linear PSS model is less to be considered, while when $\lambda > 0.5$, the effect of KL-distance is also decreased. Thus, according to the experimental results, we set $\lambda = 0.5$ as the default value in our experiments.

5.3.2. Impact of K

In HGRA, *K* is another critical parameter that determines the number of subgroup and then affects the quality of group recommendation. To study the impact of *K* on nDCG, we investigate the impact of *K*under three group size of 10, 20 and 30, respectively. And the other parameters are same as Table 3.

In Fig. 5, the experimental results show that the value of *K* significantly impacts the quality of group recommendation (nDCG), and a proper value of *K* in HGRA will present better result in different data sets. This is because too small value of *K* cannot obtain the personalized subgroup. Although large value of *K* will keep the better performance, it is very time-consuming and costly to do that. Thus, we set $K = 6$ in the data set of Web service, and $K = 8$ in the data set of CMfg service.

5.3.3. Impact of N

N is always a critical parameter in the group recommendation systems, and it determines the number of Top-N recommendation list in HGRA. To study the impact of *N* on group recommendation quality, we investigate the impact of *N* in the value range of 10 to 50 with a step value of 10, while holding other experimental parameters unchanged. Fig. 6 shows the nDCG value of Web service and CMfg service under three group size conditions of 10, 20 and 30, respectively.

As shown in Fig. 6, our approach HGRA achieves the highest group recommendation quality when $N = 30$, indicating that the optimal value of *N* is almost not affected by the group size. Another observation is that, an appropriate value of *N* will come out better recommendation in two given data sets. In general, too large number of Top-N recommendation list would decreases the individual recommendation quality, but not in the group recommendation systems. Thus, we set $N = 30$ as the default value in our experiments. However, the value of *N* can be adjustable according to real scenario.

6. Conclusion

In this paper, we proposed a hybrid group recommendation approach (HGRA) as an alternative to the traditional approaches. To address the group recommendation problem in CMfg systems, we first split up the whole process into three successive components, including enhanced user similarity model, K-medoids clustering algorithm and weighted rank aggregation model. In the enhanced user similarity model, we improved the user similarity measure which aims to identify the set of group users by incorporating non-linear PSS model and KL distance model. Based on the similar user group, we then integrated a new information similarity into K-medoids clustering algorithm to discover the personalized subgroup, in which the information interactions have been taken into account. Considering the personal influence of representative users in the subgroup, we also established a weighted rank aggregation model by exploring the entropy information among them, so as to obtain the final Top-N list in a group recommendation way. The results of the experiments showed that our HGRA not only can greatly improve the quality of group recommendation, but also increase the level of group satisfaction to some extent.

In our future work, we plan to further extend our HGRA system from the following two perspectives. Firstly, exploiting more context factors in CMfg systems to make a multi-objective group recommendation, such as trust in collaborative network, location information from both user and service side. Also, we think there still have some latent factors hidden in the user profile that need to be further expressed and represented, especially in the manufacturing industry. Secondly, we would like to optimize the group recommendation systems in terms of time complexity. For example, more refined and understandable model should be established to improve the similarity, and try to design a high-dimension clustering algorithm with some advanced operations as well.

CRediT authorship contribution statement

Jian Liu: Writing – original draft, Data curation, Software. **Youling Chen:** Conceptualization, Methodology. **Qingzhi Liu:** Writing – review & editing. **Bedir Tekinerdogan:** Supervision.

Declaration of Competing Interest

None.

Data availability

The authors do not have permission to share data.

Acknowledgments

The author would like to thank the editor and anonymous reviewers for their constructive and helpful comments which helped to improve the presentation of the paper. And, this work was supported by National Key R&D Program of China under grant No. 2018YFB1703002, Natural Science Key Research Project of Anhui Province Education Department under grant No. 2022AH050324, the Fundamental Research Funds for the Central Universities of China under grant No. 2019CDCGJX222, the Fundamental Research Funds for the Central Universities of China under grant No. 2018CDXYJX0019. Also, I would like to thank the financial support from the program of China Scholarships Council (No.202006050069).

References

Abolghasemi, R., Engelstad, P., Herrera-Viedma, E., & Yazidi, A. (2022). A personality-aware group recommendation system based on pairwise preferences. *Information Sciences, 595*, 1–17.

Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences, 178*(1), 37–51.

Azadjalal, M. M., Moradi, P., Abdollahpouri, A., & Jalili, M. (2017). A trust-aware recommendation method based on Pareto dominance and confidence concepts. *Knowledge-Based Systems, 116*, 130–143.

Bach, N. X., Do Hai, N., & Phuong, T. M. (2016). Personalized recommendation of stories for commenting in forum-based social media. *Information Sciences, 352*, 48–60.

Baker, O., & Yuan, Q. (2021, November). Machine learning: Factorization machines and normalized discounted cumulative gain for tourism recommender system optimisation. In *In 2021 IEEE international conference on computing (ICOCO)* (pp. 31–36). IEEE.

Baltrunas, L., Makcinskas, T., & Ricci, F. (2010, September). Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on recommender systems* (pp. 119–126).

Breese, J. S., Heckerman, D., & Kadie, C. (2013). *Empirical analysis of predictive algorithms for collaborative filtering.* arXiv preprint arXiv:1301.7363.

Castro, J., Yera, R., & Martínez, L. (2017). An empirical study of natural noise management in group recommendation systems. *Decision Support Systems, 94*, 1–11.

Castro, J., Yera, R., & Martinez, L. (2018). A fuzzy approach for natural noise management in group recommender systems. *Expert Systems with Applications, 94*, 237–249.

Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. *Recommender Systems Handbook*, 107–144.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction, 4* (2), 81–173.

Feng, Y., & Huang, B. (2020). Cloud manufacturing service QoS prediction based on neighbourhood enhanced matrix factorization. *Journal of Intelligent Manufacturing, 31*(7), 1649–1660.

Garcia, I., Pajares, S., Sebastia, L., & Onaindia, E. (2012). Preference elicitation techniques for group recommender systems. *Information Sciences, 189*, 155–175.

Garcia, I., Sebastia, L., & Onaindia, E. (2011). On the design of individual and group recommender systems for tourism. *Expert Systems with Applications, 38*(6), 7683–7692.

Ghazanfar, M. A., & Prügel-Bennett, A. (2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications, 41*(7), 3261–3275.

Gohari, F. S., Aliee, F. S., & Haghighi, H. (2018). A new confidence-based recommendation approach: Combining trust and certainty. *Information Sciences, 422*, 21–50.

Guo, G., Zhang, J., & Yorke-Smith, N. (2015). Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems, 74*, 14–27.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30–37.
Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics, 22*(1), 79–86.
Lee, K. H., & Kim, M. H. (2022). Bayesian inductive learning in group recommendations for seen and unseen groups. *Information Sciences, 610*, 725–745.
Li, B. H., Zhang, L., Wang, S. L., Tao, F., Cao, J. W., Jiang, X. D., … Chai, X. D. (2010). Cloud manufacturing: A new service-oriented networked manufacturing model. *Computer Integrated Manufacturing Systems, 16*(1), 1–7.
Li, J., Tao, F., Cheng, Y., & Zhao, L. (2015). Big data in product lifecycle management. *The International Journal of Advanced Manufacturing Technology, 81*(1), 667–684.
Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems, 56*, 156–166.
Lu, Y., & Cecil, J. (2016). An internet of things (IoT)-based collaborative framework for advanced manufacturing. *The International Journal of Advanced Manufacturing Technology, 84*(5), 1141–1152.
Luo, X., Xia, Y., & Zhu, Q. (2012). Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems, 27*, 271–280.
Mao, M., Lu, J., Han, J., & Zhang, G. (2019). Multiobjective e-commerce recommendations based on hypergraph ranking. *Information Sciences, 471*, 269–287.
Mourtzis, D., Fotia, S., Boli, N., & Vlachou, E. (2019). Modelling and quantification of industry 4.0 manufacturing complexity based on information theory: A robotics case study. *International Journal of Production Research, 57*(22), 6908–6921.
Nam, L. N. H. (2021). *Towards comprehensive profile aggregation methods for group recommendation based on the latent factor model*.
Papakostas, N., & Ramasubramanian, A. K. (2022). Digital technologies as a solution to complexity caused by mass personalization. In *Design and operation of production networks for mass personalization in the era of cloud technology* (pp. 153–180). Elsevier.
Selvitopi, R. O., Turk, A., & Aykanat, C. (2012). Replicated partitioning for undirected hypergraphs. *Journal of Parallel and Distributed Computing, 72*(4), 547–563.
Seo, Y. D., Kim, Y. G., Lee, E., & Baik, D. K. (2017). Personalized recommender system based on friendship strength in social network services. *Expert Systems with Applications, 69*, 135–148.
Seo, Y. D., Kim, Y. G., Lee, E., & Kim, H. (2021). Group recommender system based on genre preference focusing on reducing the clustering cost. *Expert Systems with Applications, 183*, Article 115396.
Seo, Y. D., Kim, Y. G., Lee, E., Seol, K. S., & Baik, D. K. (2018). An enhanced aggregation method considering deviations for a group recommendation. *Expert Systems with Applications, 93*, 299–312.
Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence, 2009*.
Villavicencio, C., Schiaffino, S., Diaz-Pace, J. A., & Monteserin, A. (2019). Group recommender systems: A multi-agent solution. *Knowledge-Based Systems, 164*, 436–458.
Vosecky, J., Leung, K. W. T., & Ng, W. (2014, July). Collaborative personalized twitter search with topic-language models. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 53–62).
Wang, W., Zhang, G., & Lu, J. (2017). Hierarchy visualization for group recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49*(6), 1152–1163.
Wang, Y., Deng, J., Gao, J., & Zhang, P. (2017). A hybrid user similarity model for collaborative filtering. *Information Sciences, 418*, 102–118.
Wu, J., Chen, L., Feng, Y., Zheng, Z., Zhou, M. C., & Wu, Z. (2012). Predicting quality of service for selection by neighborhood-based collaborative filtering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43*(2), 428–439.
Wu, L., He, M., & Han, Y. (2014, May). Hypergraph clustering-based cloud manufacturing service management method. In *Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design (CSCWD)* (pp. 220–225). IEEE.
Xiang, F., Jiang, G., Xu, L., & Wang, N. (2016). The case-library method for service composition and optimal selection of big manufacturing data in cloud manufacturing system. *The International Journal of Advanced Manufacturing Technology, 84*(1), 59–70.
Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing, 28*(1), 75–86.
Yang, W., Wang, G., Bhuiyan, M. Z. A., & Choo, K. K. R. (2017). Hypergraph partitioning for social networks based on information entropy modularity. *Journal of Network and Computer Applications, 86*, 59–71.
Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2006). TV program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction, 16*(1), 63–82.
Yu, Z. Y., Wang, J. D., Zhang, H. W., & Niu, K. (2016). Services recommended trust algorithm based on cloud model attributes weighted clustering. *Automatic Control and Computer Sciences, 50*(4), 260–270.
Zhang, L., Luo, Y., Tao, F., Li, B. H., Ren, L., Zhang, X., … Liu, Y. (2014). Cloud manufacturing: A new manufacturing paradigm. *Enterprise Information Systems, 8*(2), 167–187.
Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2010). QoS-aware web service recommendation by collaborative filtering. *IEEE Transactions on Services Computing, 4*(2), 140–152.