

RESEARCH

Open Access



A method to recommend cloud manufacturing service based on the spectral clustering and improved Slope one algorithm

Qiang Hu^{1,2}, Haoquan Qi¹, Wen Huang¹ and Minghua Liu^{1*}

Abstract

The booming growth of cloud manufacturing services provides users with more choices. However, cloud manufacturing service recommendation remains a challenging issue due to numerous similar candidate services and diverse user preferences. The purpose of this paper is to provide an efficient and accurate cloud manufacturing service recommendation method. A spectral clustering algorithm is first designed to cluster the cloud manufacturing services. Then the candidate rating service set is constructed based on the service clusters by service function comparison and parameter matching. Finally, an improved Slope one algorithm, which integrates user similarity and service similarity, is proposed to rate the cloud manufacturing services. The top-*k* services with the highest scores are recommended to the users. Experiments show that the proposed method can provide more accurate service rating with less time consumption. The service recommendation performance of this method is also proved to be superior to other methods in terms of precision, recall, and F-score.

Keywords Cloud manufacturing service, Service similarity, Service rating, Slope one

Introduction

Cloud manufacturing is a typical implementation mode of intelligent manufacturing [1]. In the cloud manufacturing environment, service providers encapsulate their processing equipment resources or manufacturing operations as cloud services. Cloud manufacturing service is a kind of application program interface of manufacturing service. A user can rent cloud manufacturing services from the cloud manufacturing platforms. By integrating different cloud manufacturing services, users can quickly implement manufacturing operations beyond their own business capabilities [2].

There are many cloud manufacturing services with similar functions on the cloud manufacturing platforms. Compared with the common cloud services, cloud manufacturing services have more manufacturing attribute parameters [3]. Moreover, the manufacturing attribute parameters vary greatly between different types of cloud manufacturing services. The value types of these parameters are also diverse. It is difficult for users to select suitable services from a large number of similar cloud manufacturing services. Therefore, cloud manufacturing service recommendation is greatly concerned by researchers [4].

Service recommendation finds a set of services that fit the user's needs. In recent years, researchers introduce some methods of recommendation system into the field of service computing. It mainly includes content-based service recommendation, service quality oriented recommendation, and hybrid service recommendation methods [5]. These methods have improved the accuracy of recommendation. However, cloud manufacturing services

*Correspondence:

Minghua Liu
qustlmh@163.com

¹ College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China

² Yunnan Key Laboratory of Service Computing, Kunming 650214, China

differ from ordinary commodities and face the following two prominent problems in service recommendation.

One problem is that there are many manufacturing attribute parameters in cloud manufacturing services. The values of these manufacturing attribute parameters have different dimensions and orders of magnitude, so it is much more difficult to determine the similarity between cloud manufacturing services than the common cloud services [6]. Moreover, a large number of cloud manufacturing services are published on the cloud manufacturing service platforms. Cloud manufacturing service recommendation suffers a huge service search space.

Another problem is that users' historical service preferences greatly influence the recommendation of cloud manufacturing services [7]. Users prefer to choose the cloud manufacturing services published by the service providers with cooperation experience. Alternatively, they are more likely to adopt the cloud manufacturing services from the providers that have cooperated with the users similar to them. The invoking time of historical services, the difference in service scores, and service preferences can also help improve the recommendation accuracy and rationality. These factors need to be considered in the cloud manufacturing service recommendation.

To enhance the quality of service recommendation, a cloud manufacturing service recommendation method based on spectral clustering and the improved Slope one algorithm is proposed. The main contributions of this paper are as follows:

- (1) A spectral clustering algorithm for cloud manufacturing services is devised. The search space of rating services is reduced by introducing service clustering. It can improve the efficiency of obtaining services to be rated.
- (2) An improved Slope one algorithm, which integrates user similarity and service similarity, is proposed to rate the cloud manufacturing services more reasonably and accurately.
- (3) A cloud manufacturing service data set with user ratings is established. Experiments carried out on this data set show that the proposed method is superior to the current popular methods in service rating and recommendation.

The rest of this paper is organized as follows. [Related work](#) section introduces the related works on service recommendation. [Spectral clustering for cloud manufacturing services](#) section presents a spectral clustering method for cloud manufacturing services. The improved Slope one algorithm is elaborated in [The improved Slope](#)

[one method integrating user and service similarity](#) section. [Recommendation of cloud manufacturing services](#) section provides our proposed service recommendation method. [Experiment and comparison](#) section verifies the performance of the proposed method. [Conclusions](#) section concludes this paper and throws light on future work.

Related work

Service recommendation has been concerned by researchers for more than a decade. Early service recommendation focused on Web services described by WSDL [8, 9]. Currently, the research objects of service recommendation mainly include Web services described by natural language text [10], various cloud services (for example, cloud manufacturing services [11], e-health cloud services [12]), and Internet of things services [13, 14].

Researchers have proposed many types of service recommendation methods. Among these methods, collaborative filtering and its variants are the most widely used. For example, Xiao et al. proposes a hybrid collaborative filtering algorithm to recommend the manufacturing service based on the multidimensional information in cloud manufacturing resource, the information entropy and rough set theory [15]. Zhou et al. presents a hybrid collaborative filtering model for consumer service recommendation based on mobile cloud by introducing user preferences. It can effectively reduce the data sparsity and increase the accuracy of the prediction [16]. To improve the accuracy and scalability, Wang et al. integrates the score, social trust information and review into a comprehensive model through collaborative filtering and propose a multi-source information fusion recommendation method [17]. The existing collaborative filtering service recommendation methods usually assist recommendation from one dimension of users or services. Although they can improve the quality of recommendation, it lacks comprehensive consideration of the similarity between users and services, which may affect the improvement of service recommendation accuracy.

Service quality is an important factor in evaluating recommended services [18]. QoS-aware service recommendation has always attracted much attention. Chang et al. designs an integrated-graph to consolidate multi-source information from user-aware context and service-aware context. A Gaussian Mixture Model of QoS value is built to combine local and global information on the integrated-graph and to perform QoS-based Web service prediction [19]. Cao et al. proposes a QoS-aware service recommendation based on relational topic model and factorization machines. It exploits factorization machines to train the latent

topics for predicting the link relationship among Mashup and services to recommend adequate relevant top-k Web APIs [20]. Service trust and location information are frequently used to improve QoS-aware service recommendation. For example, Liu et al. presents a trust-aware collaborative filtering approach to build the trust network of clustered users. A more personalized QoS prediction and reliable cloud service can be recommended for a user by the proposed method [21]. Khavee et al. proposes a probabilistic matrix factorization-based recommendation approach, which considers geographic location information in the derivation of the preference degree underlying a mashup-API interaction. The geographical location information increases the precision of API recommendation for mashup services [22]. Although service quality can improve the recommendation quality, the recommendation effect is affected by the constituent services in the candidate service set. If there is no high-quality candidate recommendation service set, the effectiveness of such methods is very limited.

Service social relationships are also a key concern in the service recommendation. A series of service recommendation methods based on social network are proposed. For example, a Social-powered Graph Hierarchical Attention Network (SGHAN) is designed to capture users' social connections by Wei [23]. By mining users' dynamic preferences through social connections, SGHAN outperforms the state-of-the-art methods in terms of service prediction accuracy for mashup creation. Cao et al. proposes a Web service recommendation method via combining bilinear graph attention representation and xDeepFM (eXtreme Deep Factorization Machine) quality prediction. It adopts the content and structure-oriented service function classification and predicts service invocation based on multi-dimensional quality attributes [24]. Service social relationships can effectively improve the quality of recommendation. However, how to obtain the social relationships between services is a very challenging problem in the cloud manufacturing service environment.

Hybrid service recommendation methods have gradually become the mainstream way of service recommendation. Mezni et al. proposes a method to enable the context-sensitive service recommendation system with great analysis and learning capabilities based on a knowledge graph. The recommendation algorithm is defined to deliver top-rated services according to the target user's context [25]. Jiang et al. proposes a two-stage model for cloud service recommendation. He first uses Hierarchical Dirichlet Processes model to cluster cloud services and then accurately rank and recommend cloud services in each cluster based on

the personalized PageRank algorithm [26]. Ma et al. employs the interval neutrosophic numbers to measure the fuzzy trustworthiness of cloud services by client contexts. A non-compensatory multi-criteria decision-making procedure is used to rank candidate services. It can effectively recommend the trustworthy service for small and medium-sized enterprises [27]. Chao et al. explores the relationship between the content and location information to enhance the accuracy of service recommendation. It alleviates the data scarcity problem in cloud services by introducing similar domain knowledge based on transfer learning [28]. Currently, hybrid recommendation is a popular service recommendation method. Liu et al. proposes a similarity-enhanced hybrid group recommendation approach for cloud manufacturing. A weighted ranking aggregation model is established to generate a recommendation list according to the representative user of each subgroup [29]. Zhang et al. proposes service hyper-network to recommend the raw material suppliers, semi-finished product processors and finished product manufacturers that realize on-demand customization [30]. Further, he proposed an architecture of C3DP (cloud 3D printing) order task methods for complex networks based on the dynamic coupling of nodes. The C3DP model can identify the work breakdown structure of coupling task sets with high accuracy [31]. However, most hybrid recommendation methods have high complexity in practical application or require more service implicit information to assist in improving the recommendation accuracy.

Although the introduction of collaborative filtering, quality of service, and service collaboration into service recommendation can improve the quality of recommendation. However, the single use of the above methods has limited improvement in recommendation accuracy. In this paper, we synthesize the recommendation idea of the above methods and construct a new cloud manufacturing service recommendation method with low complexity and high accuracy.

Spectral clustering for cloud manufacturing services

To facilitate the elaboration of the proposed methods, we provide Table 1 to describe the symbols in this study.

Normally, cloud manufacturing services need to be rated in the service recommendation. To improve recommendation efficiency, we do not rate all cloud manufacturing services. A set of candidate rating services is constructed to store the cloud manufacturing services to be rated. The services in the candidate rating service set can meet the recommendation requirements in terms of functions and manufacturing attribute parameters.

Table 1 Symbols and meanings

Symbols	Meanings
$TS(s_i, s_j)$	Textual attribute similarity between service s_i and s_j
$NS(s_i, s_j)$	Numeric attribute similarity between service s_i and s_j
$srv(s)$	Service representation vector of cloud manufacturing service s
L	Laplacian matrix
D	Degree matrix
SM	Similarity matrix
U	The set of users
S	The set of cloud manufacturing services
r_{ij}	The score of service s_j rated by the user u_i
$R_{m \times n}$	Rating matrix
$B_{m \times n}$	The full value rating matrix
$Ur(s)$	The set of users who have rated the service s
$Sr(u)$	The set of cloud manufacturing services that the user u has rated
$P(s)$	The popularity of service s
tw_{ij}	The time weight of service s_j rated by the user u_i
$Ne(u)$	The neighbor set of user u
$ServSim(s_i, s_j)$	Similarity of service s_i and s_j
$UserSim(u, v)$	Similarity of user u and v
$DevS_{ij}$	Service deviation between service i and service j
pr_{ui}	The predicted score of user u for service i
req_s	Service request

It is costly to find the suitable services for the candidate rating service set from a large number of cloud manufacturing services [32]. Spectral clustering is adopted to reduce the search space in discovering the target services. In spectral clustering, it is necessary to construct the similarity matrix between the objects participating in the clustering. The existing methods are unsuitable for computing the similarity of cloud manufacturing services because of the many types of attribute parameters and the large difference of their values. [Similarity evaluation of cloud manufacturing services](#) section provides a method for computing the similarity of cloud manufacturing services and constructs the similarity matrix. After the similarity matrix of cloud manufacturing services is obtained, the spectral clustering algorithm can be employed for clustering.

Similarity evaluation of cloud manufacturing services

It is a popular way to convert service description into service function vector and determine service similarity by calculating the Cosine angle value or Euclidean distance of service function vectors [33]. Unlike the common cloud services, there are not only service function descriptions but also multi-dimensional manufacturing

attribute parameters in the cloud manufacturing services. The values of these manufacturing attribute parameters have different dimensions and orders of magnitude, so the cloud manufacturing services with similar functions may be grouped into different service clusters due to their difference in manufacturing attribute parameters. Therefore, both service descriptions and manufacturing attribute parameters should be involved in computing the cloud manufacturing service similarity.

Definition 1 (cloud manufacturing service)

Cloud manufacturing service s is defined as a 5-tuple $s = (id, sn, st, sf, TP)$, where id is the service identification, sn is the service name, st is the service category, sf is the description of the service function, and TP is the set of manufacturing attribute parameters.

The set of manufacturing attribute parameters is denoted as $TP = \{tp_i\}$, $tp_i = \{n, c, v, u\}$, where n is the name of the manufacturing attribute parameter, c is the comparison operator, v is the value of the manufacturing attribute parameter, and u is its unit. For a given manufacturing attribute parameter $tp_1 = (die_filletradius, >, 0.5, m)$, it means that the fillet radius formed by this manufacturing service during stamping the die can be greater than 0.5 m. The parameter $tp_2 = (Manufacturing_cycle, \leq, 6, day)$ means that the manufacturing cycle of this manufacturing service does not exceed 6 days.

Table 2 presents an example of cloud manufacturing service. We can see that the textual attributes of cloud manufacturing services mainly include processing object, function description, category, subordinate industry, material, etc. Among these attribute values, there are not only short text at the paragraph level, but also a single word or phrase. GSDMM is used to generate a topic vector for the short text. GSDMM is a probabilistic unsupervised model. It generates documents based on the Dirichlet multinomial mixtures (DMM) model. DMM can model the topic information of documents and divide them into different categories. Gibbs sampling algorithm is employed to extract the topic information for DMM in GSDMM [34].

Word2Vec is employed to build vectors for the single word or phrase in this study [34]. Word2Vec is a natural language processing model based on neural networks. It works by training a shallow neural network to represent words as vectors in a continuous vector space. CBOW and Skip-Gram algorithms are employed to learn distributed representations of words. CBOW predicts the target word from the context, while Skip-Gram predicts the context word from the target word. Semantically similar words are closer in the word vector space of Word2Vec. Therefore, we can calculate word similarity using these word vectors.

Table 2 An example of cloud manufacturing service

Service name	Axle processing for motor vehicles		
Service description	Mechanical processing services can be provided for automobiles, motorcycles, engineering vehicles and other motor vehicles, with camshaft, hydraulic press, stamping die, CAM hydraulic press, half shaft processing and other products production and processing capacity. Personalized customization is also presented according to drawings of various kinds of different materials of motor vehicle axle, transmission rod and other forgings.		
Service category	hammer forging		
Manufacturing attribute parameter	shape	beam barrel	
	material	alloy steel	
	surface treatment	anodic oxidation	
	production pattern	mass customization	
	weight	<	500 kg
	length	>	2 m
	cutting diameter	>	0.5 m
	price	<	10,000 yuan
	manufacturing cycle	<	1 day
	annual processing capacity	>	5000 set
	machining precision	>	99.5 %
	service reliability	>	99 %

Definition 2 (similarity of textual attributes)

$TA = \{ta_1, ta_2, \dots, ta_n\}$ is the set of textual attributes for the manufacturing service s . $VTA = \{vt_1, vt_2, \dots, vt_n\}$ is a set of vectors generated by values of the attributes in TA , that is, $\forall ta_k \in TA$, vt_k is the vector generated for the attribute value of ta_k . The similarity of textual attributes for the cloud manufacturing services s_i and s_j is defined as:

$$TS(s_i, s_j) = \sum_{k=1}^n \frac{s_i \cdot vt_k \cdot s_j \cdot vt_k}{|s_i \cdot vt_k| \times |s_j \cdot vt_k|} / n \quad (1)$$

The numerical attribute values of the cloud manufacturing services are normalized by the maximum-minimum method. The similarity of numerical attributes can be evaluated by the Euclidean distance based on these normalized attribute values. Since the smaller the value of Euclidean distance is, the more similar the sample data is, the power index function is introduced to map the Euclidean distance forward to the numerical attribute similarity between cloud manufacturing services.

Definition 3 (similarity of numerical attributes)

$NA = \{na_1, na_2, \dots, na_m\}$ is the set of numerical attributes for the cloud manufacturing service s and $NNA = \{nt_1, nt_2, \dots, nt_m\}$ is a set of normalized values of the attributes in NA , that is, $\forall nt_k \in NNA$, nt_k is the normalized value for the attribute value of na_k . The similarity of numerical attributes for the cloud manufacturing services s_i and s_j is defined as:

$$NS(s_i, s_j) = e^{-\sqrt{\sum_{k=1}^m (s_i \cdot nt_k - s_j \cdot nt_k)^2}} \quad (2)$$

Definition 4 (similarity of cloud manufacturing services)

Let s_i and s_j be two cloud manufacturing services. $TS(s_i, s_j)$ and $NS(s_i, s_j)$ represent the similarity of textual attributes and the similarity of numerical attributes for s_i and s_j , respectively. The similarity of cloud manufacturing services s_i and s_j is defined as:

$$ServSim(s_i, s_j) = e^{-\left[1 - \left[\frac{(NS(s_i, s_j) + TS(s_i, s_j))}{2}\right]\right]} \quad (3)$$

The cosine similarity of vectors is exploited to calculate the similarity of textual attributes. The reciprocal power of Euclidean distance of attribute values is used to express the similarity of numerical attributes. We employ the reciprocal power of the mean similarity of textual attributes and numerical attributes to represent the similarity of cloud manufacturing services.

Definition 5 (service representation vector)

$VTA = \{vt_1, vt_2, \dots, vt_n\}$ is the set of the textual attribute values for the cloud manufacturing service s after vectorization. $NNA = \{nt_1, nt_2, \dots, nt_m\}$ is the set of numerical attribute values for the cloud manufacturing service s after normalization. The service representation vector of cloud manufacturing service s is represented as $srv(s) = (vt_1, vt_2, \dots, vt_n, nt_1, nt_2, \dots, nt_m)$.

Spectral clustering algorithm for cloud manufacturing services

In view of the fact that cloud manufacturing services have multi-dimensional feature descriptions of functions

and manufacturing attribute parameters, and there are many kinds of data types and significant differences in values of these features, we select the spectral clustering algorithm that can identify arbitrary shape sample space and quickly converge to the global optimal solution as the service clustering method.

The spectral clustering algorithm is evolved from the spectral graph partition theory [35]. It can be summarized into the following three steps: (1) construct the similarity matrix and degree matrix of the data set; (2) build the Laplacian matrix L and obtain the eigenvectors of the first k eigenvalues of the normalized L ; (3) cluster feature vectors by a given clustering method.

The gaussian kernel function is usually used to construct the similarity matrix in the traditional spectral clustering algorithm [36]. This method is unsuited to construct similarity matrix for cloud manufacturing services. In addition, the number of service clusters needs to be set manually, which makes the number of clusters greatly affected by human subjective factors. To achieve cloud manufacturing service clustering more accurately, we adopt the service similarity calculation method in the previous section to construct the similarity matrix of cloud manufacturing services. The eigengap is also introduced to determine the number of clusters, which makes up for the deficiency of artificially determining the number of clusters in the traditional spectral clustering algorithm [37].

Algorithm 1 is the spectral clustering algorithm for cloud manufacturing services. From line (1) to line (4), we first construct a similarity matrix SM for cloud manufacturing services. The element sm_{ij} in SM is the similarity between service s_i and s_j . Then, the degree matrix D is established in line (5). In the degree matrix D , the value of diagonal elements is the sum of the element values of the corresponding row in SM , and the value on non-diagonal elements is 0.

The normalized Laplacian matrix L is obtained based on D by formula (4). Assuming that there are n eigenvalues in the Laplacian matrix L , the above eigenvalues are sorted from large to small, and the sorted eigenvalues are $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ (line (6) and line (7)). The eigengap sequence is defined as $\{g_1, g_2, \dots, g_{n-1} | g_i = \lambda_i - \lambda_{i+1}\}$. The algorithm searches the first maximum value g_i in this sequence and sets the index i corresponding to this value as the number of clusters k from line (8) to line (11). Finally, the eigenvectors with the first k maximum eigenvalues are selected to generate a reduced dimension matrix $L'_{m \times k}$ in line (12). K-means++ algorithm is employed to cluster cloud manufacturing services based on the vectors in $L'_{m \times k}$. The k service clusters are generated and returned as the set CS (line (13) to line (15)).

$$L = D^{-\frac{1}{2}} SMD^{-\frac{1}{2}} = \left(d_i^{-\frac{1}{2}} sm_{ij} d_j^{-\frac{1}{2}} \right)_{i,j=1,\dots,n} \quad (4)$$

Input: the set of cloud manufacturing services S

Output: the set of service clusters CS

```
(1) for each two service  $s_i$  and  $s_j$  in  $S$ ;
(2)   compute  $SerSim(s_i, s_j)$ ;
(3) endfor
(4) generate the similarity matrix  $SM$  based on  $SerSim(s_i, s_j)$ ;
(5) construct degree matrix  $D$  from  $SM$ ;
(6) compute the normalized Laplacian matrix  $L$  by formula(X);
(7) obtain the eigenvalues of  $L$  and sort them by  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ ;
(8) for  $i=1$  to  $n-1$ 
(9)   compute the eigengap sequence  $g_i = \lambda_i - \lambda_{i+1}$ ;
(10) end for
(11)  $k = \text{argmin} \{g_i - g_{i+1} > 0 \& \& g_i - g_{i+1} > 0\}$ ;
(12)  $L_{m \times m} \rightarrow L'_{m \times k}$  by the eigenvectors with the first  $k$  maximum eigenvalues;
(13) cluster the cloud manufacturing services by K-means++ and  $L'_{m \times k}$ ;
(14)  $CS = \{cs_i\}, i=1, 2, \dots, k$ ;
(15) return( $CS$ );
```

Algorithm 1 *Spec_clustering_CMS*

The improved Slope one method integrating user and service similarity

The basic Slope one method

Slope one algorithm is a collaborative filtering method based on item ratings [38]. It uses the linear regression model $w = f(v) + b$ to predict the score, where w represents the score of the target user on item i , $f(v)$ represents the scores of the target user on other items, and b is the average deviation of item i relative to other items.

The set of users in cloud manufacturing platforms is represented as $U = \{u_1, u_2, u_3, \dots, u_m\}$ while the set of cloud manufacturing services is denoted as $S = \{s_1, s_2, s_3, \dots, s_n\}$. The rating matrix is represented by $R_{m \times n}$, where the element r_{ij} in $R_{m \times n}$ is the score of service s_j rated by the user u_i . The score range is between 1 and 5. The higher the score, the better the user evaluates the service. If a user u_i did not rate the service s_j , the r_{ij} was set as 0. We also present the following symbols: $Ur(s)$ is the set of users who have rated the service s . $Sr(u)$ is the set of cloud manufacturing services that the user u has rated.

The average deviation between service s_i and s_j is shown in formula (5) when the traditional Slope one method is used to predict the scores of cloud manufacturing services [39].

$$DevS_{ij} = \frac{\sum_{u \in Ur(s_i) \cap Ur(s_j)} (r_{ui} - r_{uj})}{|Ur(s_i) \cap Ur(s_j)|} \quad (5)$$

According to the score deviation between cloud manufacturing services and the user's historical scores, the predicted score of service s_i by the user u is shown in formula (6).

$$pr_{ui} = \frac{\sum_{s_j \in Sr(u)} (r_{uj} + DevS_{ij})}{\left| \bigcup_{s_j \in Sr(u)} Ur(s_i) \cap Ur(s_j) \right|} \quad (6)$$

The traditional Slope one method faces two problems in rating prediction. One is that it does not differentiate users' scores. Using the scoring data of all users for prediction means that some users with different or even opposite preferences are also involved. When there is a large difference in the service preferences of users participating in the prediction, the accuracy of service rating will be significantly reduced. Second, the service deviation is not differentiated. The rating deviation of services with high similarity to the target service should play a larger role in the rating prediction. On the contrary, the rating deviation of services with low similarity to the target service should be given a smaller role in the rating prediction. However, the deviations between all services and the target service are treated equally in the traditional Slope one algorithm, which further reduces the accuracy of rating prediction.

To solve the above problems, we introduce user similarity and service similarity to improve the quality of service deviation calculation and correct the weight of service deviation corresponding to different services to enhance the accuracy of rating prediction.

User similarity

User similarity is an important factor in service recommendation. In traditional methods, the similarity of users is usually calculated based on the difference of users' scores on services. These methods treat users' historical scores indiscriminately when calculating user similarity. However, users' scores are affected by the service popularity and the change in users' interests.

Most users are willing to participate in rating the popular services, so popular services should contribute less to distinguishing user similarity. On the contrary, if two users have common needs or interests in some unpopular services, it can better reflect the high similarity between the two users. Considering that user

similarity will be affected by popular services, we introduce service popularity to balance the influence of popular services on user similarity.

Service popularity is measured by the frequency of service rating. The more the service is rated, the more popular the service is, and the smaller the contribution of the service to the similarity of users' interests. The calculation method of service popularity can refer to formulas (7) and (8).

$$vr_{ij} = \begin{cases} 1, & r_{ij} \neq 0 \\ 0, & r_{ij} = 0 \end{cases} \quad (7)$$

$$P(s_j) = \frac{\sum_{j=1}^m vr_{ij}}{\sum_{j=1}^n \sum_{i=1}^m vr_{ij}} \quad (8)$$

Users' preferences for services will change over time. The services that users have visited in the recent period of time can more accurately reflect their current needs and interests. So, the weight of recently visited services in evaluating user similarity should be larger. We build a rating time matrix T according to the user's rating time for the cloud manufacturing services. In matrix T , the element t_{ij} is the rating time of user u_i on service s_j , that is, the generation time of r_{ij} .

Recently generated scores have a higher weight in calculating user similarity, while scores with a longer time should have a smaller weight. tw_{ij} is used to represent the time weight of service s_j rated by the user u_i . The calculation method of tw_{ij} is shown in formula (9), where t_{current} is the current time.

$$tw_{ij} = e^{t_{ij} - t_{\text{current}}} \quad (9)$$

In most scenarios, there are personalized differences between user ratings. Some users generally give high scores to rating objects, while others generally give low scores to the rating objects. To eliminate the subjective deviation of users' scores, the adjusted Cosine is used to calculate user similarity. The adjusted Cosine subtracts all the scores from the user's average score to calculate the similarity, which can more objectively express the user's preferences.

As shown in formula (10), we comprehensively consider the three elements of service score, rating time weight and service popularity, and use the adjusted Cosine similarity to calculate the user similarity. Here $Ne(u)$ represents the neighbor of user u . The neighbor of user u is defined as $Ne(u) = \{v \mid UserSim(u, v) > \delta\}$, δ is a threshold of user similarity.

$$UserSim(u, v) = \frac{\sum_{s \in Sr(u) \cap Sr(v)} (r_{us} \times tw_{us} - \overline{r_u} \times \overline{tw_{us}}) \times (r_{vs} \times tw_{vs} - \overline{r_v} \times \overline{tw_{vs}}) \times P(s)}{\sqrt{\sum_{s \in Sr(u)} ((r_{us} \times tw_{us} - \overline{r_u} \times \overline{tw_{us}}))^2} \times \sqrt{\sum_{s \in Sr(v)} ((r_{vs} \times tw_{vs} - \overline{r_v} \times \overline{tw_{vs}}))^2}} \quad (10)$$

The improved Slope one method

There are many cloud manufacturing services in the cloud manufacturing service platforms. Each user has only evaluated a few services, and each service has only been rated by a few users. Therefore, the rating data of users on cloud manufacturing services is sparse. It leads to the deviation between users and inaccurate prediction scores, which reduces the accuracy of recommendation.

To alleviate data sparsity, we first fill the zero value in the rating matrix with a weighted average value before rating the cloud manufacturing services. Formula (11) presents the average filling method before prediction. The scores of cloud manufacturing services rated by u_i and all the scores of s_i rated by other users are employed to average and assign to r_{ij} once r_{ij} is zero.

$$\bar{r}_{ij} = \begin{cases} \frac{\sum_{s_t \in Sr(u_i)} r_{it} + \sum_{u_k \in \{U - \{u_i\}\}} r_{kj}}{\frac{|Sr(u_i) + Ur(s_j)|}{|Sr(u_i) + Ur(s_j)|}}, & |Sr(u_i) + Ur(s_j)| \neq 0 \\ 0, & |Sr(u_i) + Ur(s_j)| = 0 \end{cases} \quad (11)$$

The traditional Slope one algorithm uses the scoring data of all users to rate the services, which means that some users with different or even opposite preferences are also involved. When there are large differences in the service preferences of users participating in the rating prediction, the prediction accuracy will be significantly reduced. It will have a negative influence on the final service recommendation.

To address this problem, we have made two improvements to the traditional deviation calculation. The first one is that the users participating in service deviation are limited to the target user's neighbors. The second one is that the user similarity is introduced to correct the weight of the score difference between different users on the final service deviation. The improved service deviation can refer to formula (12).

$$DevS_{ij} = \frac{\sum_{u \in Ne(d)} |r_{ui} - r_{uj}| \times UserSim(u, d)}{|Ne(d)|} \quad (12)$$

$$pr_{ui} = \frac{\sum_{s_j \in Sr(u) \cap Sr(Ne(u))} (r_{uj} + DevS_{ij}) \times ServSim(s_i, s_j)}{\left| \bigcup_{s_j \in Sr(u) \cap Sr(Ne(u))} Ur(s_i) \cap Ur(s_j) \cap Ne(u) \right|} \quad (13)$$

How to perform service rating in the improved Slope one algorithm is shown in formula (13). Compared with the traditional Slope one algorithm, we have made the following improvements:

- (1) The service deviation calculation method proposed in formula (12) is adopted;
- (2) Service similarity is introduced to correct the contribution degree of service deviation in the rating services;
- (3) The users participating in service rating are set as neighbor users.

Points (1) and (3) improve the rating rationality from the perspective of users. We only choose the neighbor users to participate in the service rating. It is beneficial to enhance the accuracy of rating prediction for the high similarity of service preferences between the target users and their neighbors. Point (2) improves the rating rationality from the perspective of the services. By introducing service similarity, the contribution degrees of service deviations for different services are differentiated, which is conducive to enhancing the rating rationality.

As shown in formula (14), we train the improved Slope one model using the log-cosh loss function. The Adam optimizer is employed to speed up the convergence of the loss function.

$$L = \sum_{u \in U} \sum_{i \in Sr(u)} \log(\cosh(pr_{ui} - r_{ui})) \quad (14)$$

Here, U is the set of users and $Sr(u)$ is set of cloud manufacturing services that the user u has rated. pr_{ui} and r_{ui} are the predicted score and real score, respectively. $\cosh(x) = (e^x + e^{-x})/2$ is the hyperbolic cosine function. The optimization objective of the log-cosh loss function is to minimize the difference between the predicted scores and real scores.

Recommendation of cloud manufacturing services

To reduce the search space during service recommendation, we first cluster cloud manufacturing services as service clusters (part 1 in Fig. 1). Then the candidate rating service set is constructed based on these service clusters by function similarity evaluation and attribute parameter matching (part 2 in Fig. 1). Finally, the cloud

manufacturing services in candidate rating service set are rated by the improved Slope one algorithm. The cloud manufacturing services with the highest top- k scores are recommended to the users (part 3 in Fig. 1). Algorithm 2 presents the recommendation process of cloud manufacturing services.

Input: the set of cloud manufacturing services S , the user's requirement req_s , the user u ;
Output: the top- k recommended cloud manufacturing services;

```

(1)  $SC = Spec\_clustering\_CMS(S)$ 
(2) for each service cluster  $sc$  in  $SC$ 
(3)   if  $(TS(sc.c, req_s) > \delta)$  then  $CSC = CSC \cup \{sc\}$ ;
(4) endfor
(5) for each service  $s$  in  $CSC$ 
(6)   if  $(TS(s, req_s) > \delta \& \& s.TP \rightarrow req_s.TP >)$  then  $CRS = CRS \cup \{s\}$ ;
(7) endfor
(8) construct the  $Ne(u)$  by  $UserSim(u, v)$ 
(9)  $B_{mn} = R_{mn}$ ;
(10) for each  $p_{ij}$  in  $P_{mn}$ 
(11)   if  $p_{ij} = 0$  then set  $p_{ij}$  by formula (11)
(12) endfor
(13) for each service  $s$  in  $CRS$ 
(14)   compute  $DevS_{ij}$  by formula (12) based on  $B_{mn}$ 
(15)   compute  $pr_{ui}$  by formula (13) based on  $B_{mn}$ 
(16) endfor
(17)  $rs = \{ \{s_i\} \mid Top-k\_max(pr_{ui}) \wedge s_i \in CRS \}$ 
(18) return ( $rs$ )

```

Algorithm 2 Recommend_CMS

The service request is symbolically expressed as req_s in this study. The component elements of req_s are consistent with the definition of cloud manufacturing service. The algorithm1 is first used to cluster cloud manufacturing services (line (1)). After vectorization or normalization, the attribute values of cloud manufacturing services are spliced into the service representation vectors of cloud manufacturing services. Cloud manufacturing services with similar functions are grouped into a service cluster. For a service cluster sc , its central point $sc.c$ is used to represent the service cluster. The central point of a service cluster is a virtual cloud manufacturing service. Its attribute values are the mean values of the service representation vectors of the constituent services in the service cluster.

Then the candidate rating service set is constructed (line (2) to line (7)). Cloud manufacturing services that can meet service requests in functions are added to the candidate rating service set. Function similarity calculation and attribute parameter matching are the two steps to build the candidate rating service set. From the definition of cloud manufacturing services, we can see that text attributes describe service functions, while numerical attributes illustrate service quality. So we use the text

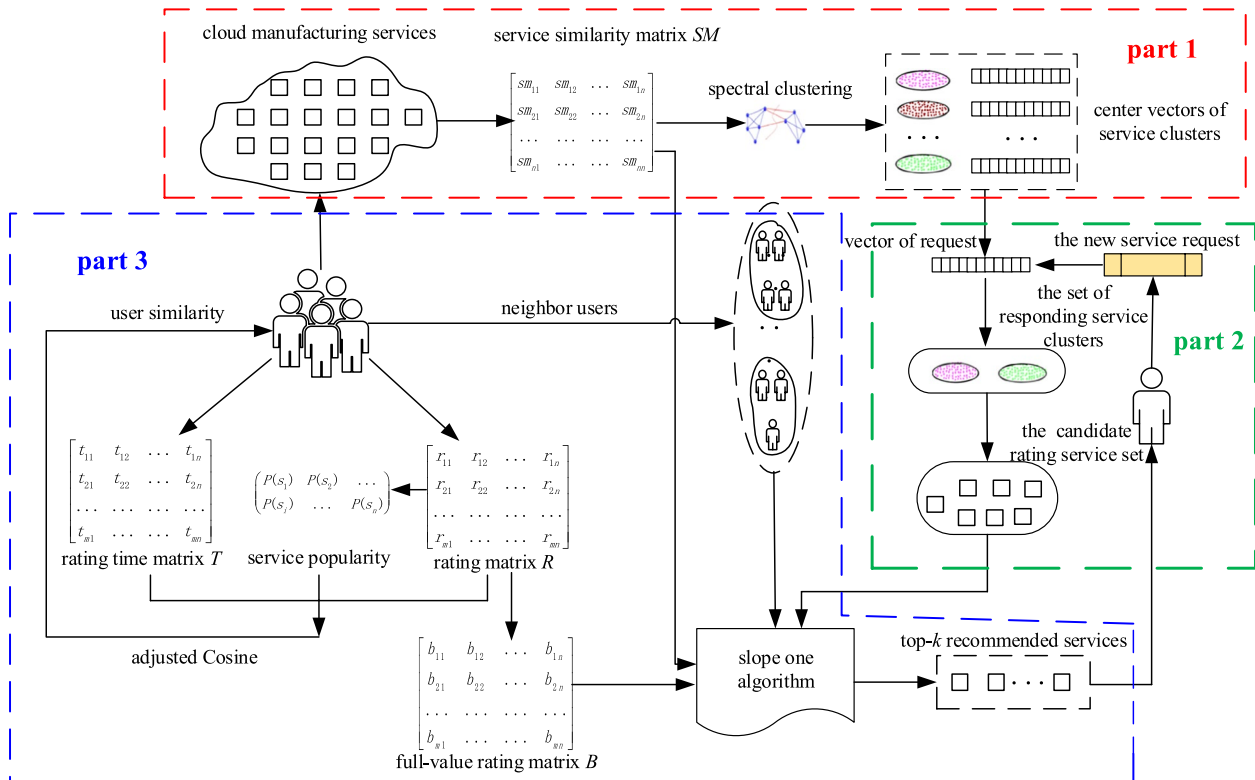


Fig. 1 Cloud manufacturing service recommendation based on spectral clustering and improved Slope one algorithm: schematic view

attribute similarity of cloud manufacturing services to evaluate their functional similarity.

The manufacturing attribute parameters can be divided into positive parameters and negative parameters. Positive parameters, such as grinding precision and cutting precision, are represented by P^+ . Better service quality will be provided if a positive manufacturing attribute parameter is with a larger value. Positive parameters, such as manufacturing cycle and price, are represented by P^- . A smaller value of positive manufacturing attribute parameter means a better service quality.

Let p be a manufacturing attribute parameter of req_s while its requested value is vr . The value provided by the attribute parameter p of the cloud manufacturing service s is vs . s can provide the matching parameter of p for req_s if the formula (15) is workable. Such parameter matching is symbolized as $s \rightarrow req_s < p >$.

$$\begin{cases} vs \geq vr, p \in P^+ \\ vs \leq vr, p \in P^- \end{cases} \quad (15)$$

It should be noticed that parameter matching is only carried out for the manufacturing attribute parameters, which are assigned with the required values in the req_s . If a user does not make a value constraint on a manufacturing attribute parameter, it means he has no requirements for it. Thus, parameter matching is unnecessary for this manufacturing attribute parameter.

Finally, the cloud manufacturing services in candidate rating service set are rated by the improved Slope one algorithm (line (8) to line (16)). We obtain the neighbors of target user u in line (8). To retain the existing service scores, a copy sample B of the rating matrix R is generated in line (9). We also fill the zero value in the rating matrix B with a weighted average value by formula (11). The improved service deviation and rating method are employed to evaluate the scores of cloud manufacturing services in the candidate rating service set CRS . The cloud manufacturing services with the top- k highest scores are obtained and recommended to users in line (17) and line (18).

Experiment and comparison

A total of 1282 cloud manufacturing services were crawled on well-known cloud manufacturing platforms such as casicloud (<https://www.casicloud.com/>) and cosmoplat (<https://www.cosmoplat.com/>). We selected 978 cloud manufacturing services as the data set in the experiments. Textual information about cloud manufacturing services, such as service name, service function description, service category, industry, production mode, processed material, is provided in the dataset. Some numerical manufacturing parameters such as price, machining accuracy, machining diameter, surface roughness, service reliability, etc.,

are also presented. Meanwhile, 96,640 scores of these services from 200 users were randomly generated. Ratings are scored on a scale of 1 to 5. The simulation program is designed by Python3.9. The operating system is Windows 10 and the hardware is as follows: CPU is i7-8750H and memory is 16 GB.

We conducted two experiments. One is service rating prediction, which is used to verify whether the improved Slope one algorithm has high accuracy in rating cloud manufacturing services. The other one is service recommendation prediction, which is used to verify the performance of our proposed service recommendation method.

The method proposed in this study is named as SC_Slope. The comparison methods are divided into two groups. The first group is the service recommendation methods with neighbor users, including CT_CF [21], TrustCF [40], wSlopeone [41] and DR_LT [42]. The second group is the service recommendation methods without neighbor users. They are vsPMF [19], HDP_PageRank [26], TASERM [43] and SC_TD [44].

We will answer the following questions in this section.

- Q1: Whether the accuracy of service rating can be improved by introducing the service similarity and user similarity?
- Q2: Whether the number of neighbor users has an influence on the accuracy of service rating?
- Q3: Does our method outperform other methods in rating accuracy?
- Q4: Does our method provide better service recommendation performance than other methods?
- Q5: Is our method more efficient than other methods in service recommendation?

Service rating

Evaluation metrics

MAE and RMSE are employed to evaluate the rating performance of above methods in the service recommendation.

(1) MAE

MAE is the mean absolute error between the real score and the predicted score. MAE can refer to formula (16). A more accurate prediction is performed if the MAE has a smaller value.

$$MAE = \frac{\sum_{u,i \in T} |r_{ui} - p_{ui}|}{n} \quad (16)$$

(2) RMSE

RMSE is the square root of the average square difference between the actual score and the predicted score. RMSE can refer to formula (17). A more accurate prediction is performed if the RMSE has a smaller value.

$$RMSE = \sqrt{\frac{\sum_{u,i \in T} (r_{ui} - p_{ui})^2}{n}} \quad (17)$$

Compared with MAE, RSME is sensitive to the deviation between the predicted value and the true value. The small RSME value reflects the small fluctuation between the predicted and true values, and the model performs well in fitting the predicted and true values. If the RSME shows a small value in multiple predictions, it can reflect the stability of the prediction model.

Performance comparison

Figures 2 and 3 show the values of MAE and RSME calculated from the service scores rated by different methods in the first group and the real scores, respectively. As can be seen from Fig. 2, the MAE value of our proposed method is lower than that of other methods under any number of neighbors. So the service scores rated by our method is much closer to the real value than other methods on the mean. Similarly, the RSME value of the proposed method is also lower than that of other methods. The lower RMSE values verify that our method shows more stable performance in rating predictions. There is less fluctuation between the predicted service scores and the real scores. Therefore, for Q1, we can draw the following conclusion: The accuracy of service rating can be improved by introducing the service similarity and user similarity.

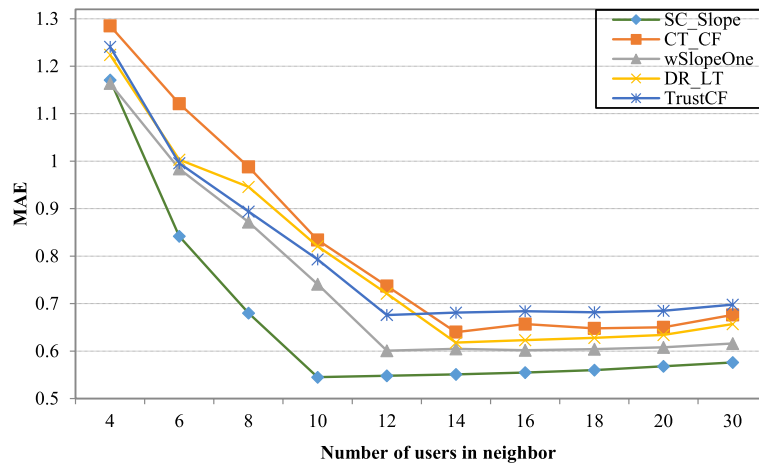


Fig. 2 MAE of service scores for different methods under different number of neighbor users

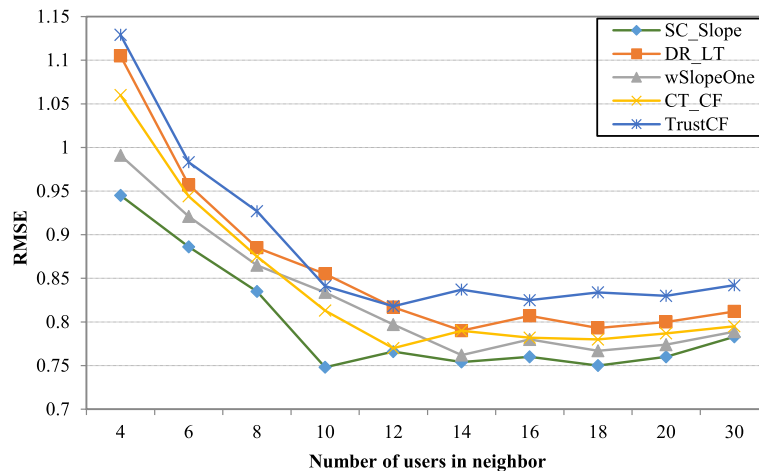


Fig. 3 RMSE of service scores for different methods under different number of neighbor users

From the curve morphology, we can see that the MAE values of all methods show a trend of decreasing first and then increasing as the number of neighbors increases from small to large. It indicates that too few or too many neighbors are not conducive to improve the accuracy of rating prediction. If few neighbor users are participating in the rating, the service preferences of target user cannot be completely extracted. Conversely, if there are too many neighbor users, the service preferences of some users with large differences in preferences may be added to the service scoring process, so that the score may deviate from the real preferences of the target user. Similar curve morphology can be found in the RMSE values. Therefore, for Q2, we can conclude that rating accuracy is affected by the number of neighborhood users.

To further show the improvement of Slope one algorithm by integrating user similarity and service similarity, we carry out experiment to verify the performances of the plain Slope one algorithm, Slope one + user similarity, Slope one + service similarity with the method proposed in this study.

From Tables 3 and 4, we can see that the rating performance of Slope one algorithm is improved after integrating user similarity and service similarity. Among all the methods, the proposed method in this paper achieves the optimal values in MAE and RSME. It shows that the performance of Slope one algorithm is greatly improved after the introduction of user similarity and service similarity.

We can obtain an appropriate number of neighbor users from MAE and RMSE, at which the service rating has the highest accuracy. In general, it is difficult to find a number of neighbor users such that the MAE and RMSE values are the lowest. Thus, we can only choose a number of users as the optimal number

Table 4 RMSE with different neighbors for different Slope one algorithm

Number of neighbors	Slope one	Slope one + us	Slope one + ss	SC_Slope
4	1.010	0.968	0.972	0.945
6	0.997	0.922	0.894	0.886
8	0.878	0.859	0.867	0.835
10	0.850	0.818	0.792	0.748
12	0.815	0.789	0.788	0.766
14	0.784	0.772	0.769	0.754
16	0.796	0.783	0.778	0.760
18	0.797	0.789	0.783	0.750
20	0.791	0.781	0.778	0.760
30	0.803	0.798	0.791	0.783

of neighbor users from the perspectives of MAE and RMSE according to experience. We expect MAE and RMSE to be as low as possible under the optimal number of neighbors.

According to the values and curve shapes of MAE and RMSE in Figs. 2 and 3, we selected 10, 14, 12, 14 and 12 as the optimal number of neighbor users for SC_Slope, DR_LT, wSlopeOne, CT_CF and TrustCF, respectively.

The methods vsPMF and TAESRM can provide service rating prediction in service recommendation. Figures 4 and 5 show the values of MAE and RSME for seven methods in service rating. The methods include vsPMF, TAESRM and all the methods in the first group. The MAE and RMSE values of the first group of methods participate in the comparison are obtained under the optimal number of neighbors. We can see that the MAE and RMSE values of our proposed method are lower than those of all other methods. Therefore, for Q3, we can conclude that our method outperforms the comparison methods in rating accuracy.

Table 3 MAE with different neighbors for different Slope one algorithm

Number of neighbors	Slope one	Slope one + us	Slope one + ss	SC_Slope
4	1.187	1.168	1.169	1.164
6	0.995	0.882	0.868	0.842
8	0.886	0.703	0.716	0.68
10	0.765	0.569	0.584	0.545
12	0.616	0.596	0.593	0.548
14	0.623	0.599	0.589	0.551
16	0.621	0.571	0.585	0.555
18	0.620	0.573	0.584	0.56
20	0.627	0.585	0.591	0.568
30	0.640	0.604	0.598	0.576

Service recommendation

Evaluation metrics

Let TS be the set of real cloud manufacturing services adopted by the users in the test data. RS is the set of the recommended cloud manufacturing services. Precision, recall and F-score are the evaluation metrics used to verify the recommendation performance.

(1) Precision

Precision is the proportion of correctly recommended services in the actual number of recommended services. It can refer to formula (18). The value range of precision is $[0,1]$. A higher the precision means a better recommendation performance.

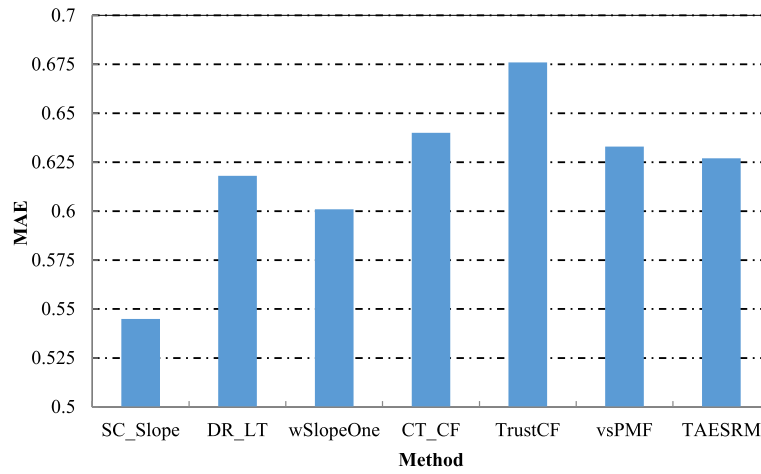


Fig. 4 MAE of service scores for different methods

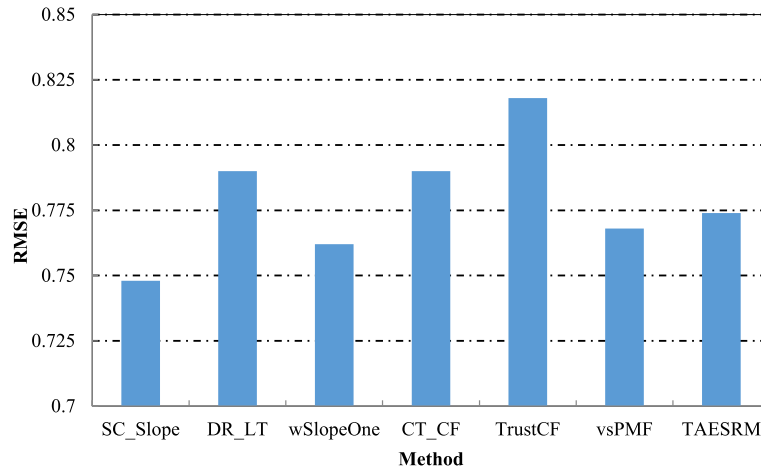


Fig. 5 RMSE of service scores for different methods

$$Precision = \frac{|RS \cap TS|}{RS} \quad (18)$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (20)$$

(2) Recall

Recall is the proportion of the correctly recommended services in the number of services users have adopted of the test set. It can refer to formula (19). Similar to the precision, a higher the recall means a better recommendation performance.

$$Recall = \frac{|N_s \cap T_s|}{T_s} \quad (19)$$

(3) F-score

F-score is the harmonic value of precision and recall. It is often used to reflect the overall performance of the recommendation system. The value range of F-score is [0,1]. Its calculation method is shown in formula (20).

Performance comparison

Since a service request can be responded by multiple cloud manufacturing services, we test the precision and recall rate under the different number of response services, and then calculate the value of *F*-score. In the performance evaluation, we set the number of cloud manufacturing services corresponding to the service requirements as 3 to 5. We calculate the precision, recall and *F*-score by counting the number of correct services in the recommended top-*k* cloud manufacturing services. Here, *k* ranges from 1 to 5. The comparison results for the top-*k* recommended services are shown in Figs. 6, 7 and 8.

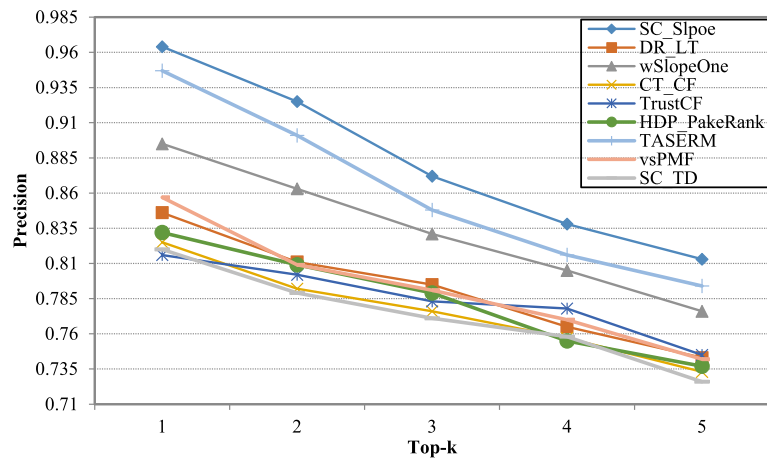


Fig. 6 Precision comparison under top- k service recommendation

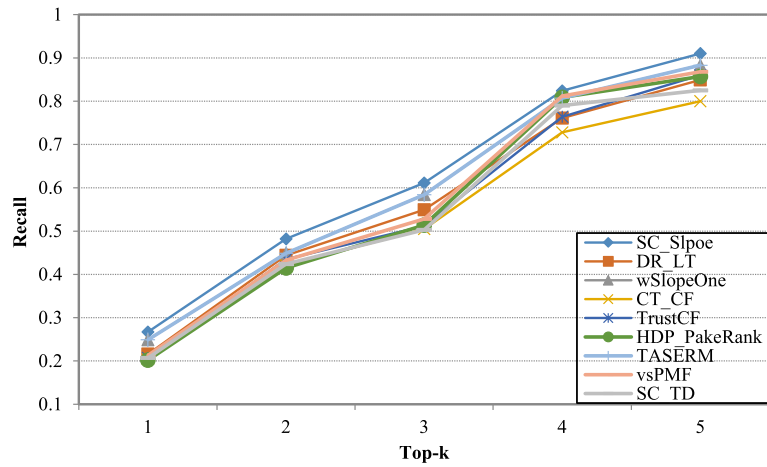


Fig. 7 Recall comparison under top- k service recommendation

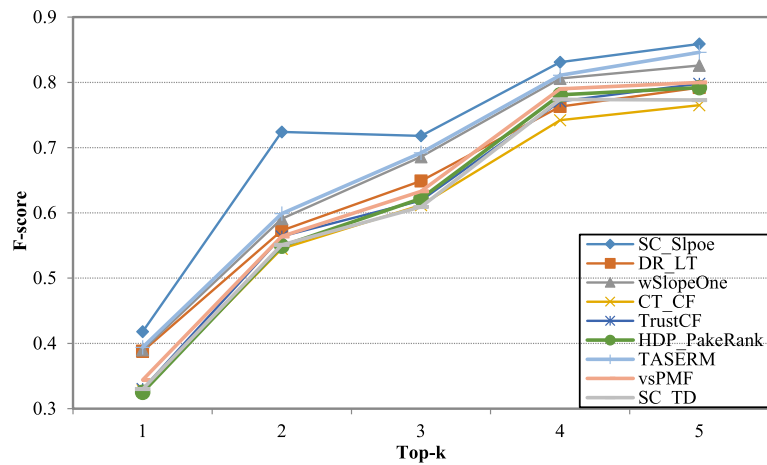


Fig. 8 F-score comparison under top- k service recommendation

We can see that our method has achieved higher values than other methods in terms of precision, recall, and F-score at any monitoring point. Compared with other top- k service recommendations, our method has more significant advantages in terms of metric values corresponding to top-1 and top-2 service recommendations. This shows that our approach is better than others at prioritizing the services that best suit the user's needs.

Compared with the average metric values of the comparison methods, the growth rate of precision, recall, and F-score at different top- k service recommendation can be seen in Table 5. Therefore, for Q4, we can get the conclusion that our method provides better service recommendation performance than other methods.

We count the execution time of different methods for service recommendation on the experimental dataset. As shown in Fig. 9, our method is the least time-consuming of all the methods. The average execution time in multiple rounds experiments is 3.78 s. The second less time-consuming method is wSlopeOne. The execution time for wSlopOne is 4.12 s. The wSlopeOne is also a Slope one variant method. The reason for the high recommendation efficiency of Slope one algorithm is that its recommendation mechanism is relatively simple, and no complex calculation is involved in the recommendation process.

The most time-consuming service recommendation method is SC_TD. The SC_TD method takes more than 8 s on average to perform cloud manufacturing service

recommendation in the experimental dataset. The second most time-consuming recommendation method is TASERM. The average time used in service recommendation of TASERM is 7.63 s. The time consumption of these methods is much higher than Slope one algorithm and its variants. Complex graph operation or matrix operation in service recommendation is the main reason for the high time consumption of these methods.

On the same dataset, the time required by our method for service recommendation is significantly lower than that of other service recommendation methods. Therefore, for Q5, we can draw the following conclusion: Our method is more efficient than other methods in service recommendation.

Conclusions

This paper proposes a method to recommend cloud manufacturing service based on spectral clustering and an improved Slope one algorithm. We design a similarity matrix for cloud manufacturing services containing many textual and numerical attributes, and apply it to spectral clustering to achieve cloud manufacturing services clustering. The introduction of service clustering reduces the service rating space in service recommendation. Additionally, we integrate service similarity and user similarity into Slope one algorithm to improve the accuracy of service rating and top- k service recommendation. The experimental results demonstrate that the proposed method outperforms the comparison method in terms of service rating and recommendation. Moreover, the time consumption of our method is significantly lower than that of other methods.

Future work includes exploring a more reasonable method to supplement the missing values of service scores, so as to better cope with the problem of sparse service score data and cold start.

Table 5 The growth rate of the evaluation metrics under top- k service recommendation

Evaluation metrics	Top1	Top2	Top3	Top4	Top5
Precision	10.8%	10.6%	8%	7.2%	7.4%
Recall	17.5%	9.7%	11.7%	4.9%	5.8%
F-score	14.6%	21.4%	10.2%	6%	6.6%

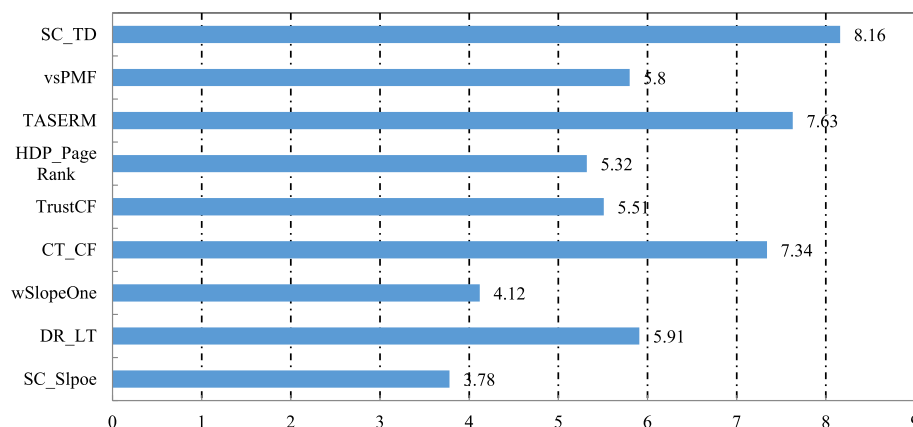


Fig. 9 The execution time of different methods in service recommendation

Acknowledgements

The authors would like to thank Yifan Wang for helping to improve the English expression.

Authors' contributions

Qiang HU designed the recommendation solution and wrote the main manuscript text. Haoquan Qi improved the Slope One algorithm. Wen Huang proposed Spectral clustering algorithm. Minghua Liu and Wen Huang designed and carried out experimental verification.

Authors' information

Qiang Hu, born in 1980, Ph. D., associate professor, e-mail: huqiang200280@163.com, his main research interests include formal modeling by Petri net and service computing.
Haoquan Qi, born in 1998, master degree candidate, e-mail: 993491261@qq.com, his main research interests include service computing.
Wen Huang, born in 1998, master degree candidate, e-mail: 786206332@qq.com, her main research interests include cloud manufacturing and service computing.
Minghua Liu, born in 1980, Ph. D., associate professor, e-mail: qustlmh@163.com, his main research interests include artificial intelligence and machine vision.

Funding

This work is supported by the Natural science foundation of China under Grant 61973180, and the Key Research Program of Shandong Province (Soft Sciences) under grant 2023RKY01009, and the Foundation of Yunnan Key Laboratory of Service Computing under Grant YNSC23116.

Availability of data and materials

Some examples of the data are displayed in "<https://github.com/QHQ5295/Dataset-examples-and-user-service-ratings.git>". Researchers who want to use the data in this paper for scientific purposes may request it from the corresponding author.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Consent has been granted by all authors and there is no conflict.

Competing interests

The authors declare no competing interests.

Received: 7 April 2023 Accepted: 20 July 2023

Published online: 03 August 2023

References

- Aheleroff S, Huang H, Xu X et al (2022) Toward sustainability and resilience with Industry 4.0 and Industry 5.0. *Front Manuf Technol* 26:1–20
- Hu Y, Wu F, Yang Y et al (2022) Tackling temporal-dynamic service composition in cloud manufacturing systems: a tensor factorization-based two-stage approach. *J Manuf Syst* 63:593–608
- Zhang Z, Hu J, Xu X et al (2023) Functional importance evaluation approach for cloud manufacturing services based on complex network and evidential reasoning rule. *Comput Ind Eng* 175:108895
- Helo P, Hao Y, Toshev R et al (2021) Cloud manufacturing ecosystem analysis and design. *Robot Comput Integr Manuf* 67:102050
- Agarwal N, Sikka G, Awasthi LK (2022) A systematic literature review on web service clustering approaches to enhance service discovery, selection and recommendation. *Comput Sci Rev* 45:100498
- Zhang Z, Chen Y, Wang X (2021) A semantic similarity computation method for virtual resources in cloud manufacturing environment based on information content. *J Manuf Syst* 59:646–660
- Li X, Wang Z, Chen CH et al (2021) A data-driven reversible framework for achieving Sustainable Smart product-service systems. *J Clean Prod* 279:123618
- Zheng Z, Ma H, Lyu MR et al (2010) QoS-aware web service recommendation by collaborative filtering. *IEEE Trans Serv Comput* 4(2):140–152
- Chan NN, Gaaloul W, Tata S (2012) A recommender system based on historical usage data for web service discovery. *SOCA* 6(1):51–63
- Rahman MM, Liu X, Cao B (2017) Web API recommendation for mashup development using matrix factorization on integrated content and network-based service clustering. In: 2017 IEEE International Conference on Services Computing (SCC). IEEE, Honolulu, p 225–232
- Zhang WY, Zhang S, Guo SS (2017) A PageRank-based reputation model for personal manufacturing service recommendation. *Enterp Inf Syst* 11(5):672–693
- Deepa N, Pandiaraja P (2020) Hybrid Context Aware Recommendation System for E-Health Care by merkle hash tree from cloud using evolutionary algorithm. *Soft Comput* 24(10):7149–7161
- Gao H, Xu Y, Yin Y et al (2019) Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services. *IEEE Internet Things J* 7(5):4532–4542
- Altulyan M, Yao L, Wang X et al (2022) A survey on recommender systems for Internet of Things: techniques, applications and future directions. *Comput J* 65(8):2098–2132
- Xiao Y, Li C, Song L et al (2021) A multidimensional information fusion-based matching decision method for manufacturing service resource. *IEEE Access* 9:39839–39851
- Zhou Q, Zhuang W, Ren H et al (2022) Hybrid collaborative filtering model for consumer dynamic service recommendation based on mobile cloud information system. *Inf Process Manage* 59(2):102871
- Wang Y, Wen J, Zhou W et al (2022) A cloud service recommendation method based on extended multi-source information fusion. *Concurr Comput Pract Exp* 34(10):e6826
- Ghafoori SH, Hashemi SM, Hung PCK (2022) A survey on Web service QoS prediction methods. *IEEE Trans Serv Comput* 15(4):2439–2454
- Chang Z, Ding D, Xia Y (2021) A graph-based QoS prediction approach for web service recommendation. *Appl Intell* 51(10):6728–6742
- Cao B, Liu J, Wen Y et al (2019) QoS-aware service recommendation based on relational topic model and factorization machines for IoT Mashup applications. *J Parallel Distrib Comput* 132:177–189
- Liu J, Chen Y (2019) A personalized clustering-based and reliable trust-aware QoS prediction approach for cloud service recommendation in cloud manufacturing. *Knowl-Based Syst* 174:43–56
- Botangen KA, Yu J, Sheng QZ et al (2020) Geographic-aware collaborative filtering for web service recommendation. *Expert Syst Appl* 151:113347
- Wei C, Fan Y, Zhang J (2022) Time-aware service recommendation with social-powered graph hierarchical attention network. *IEEE Trans Serv Comput*, (early access). <https://doi.org/10.1109/TSC.2022.3197655>
- Cao B, Zhang L, Peng M, et al (2023) Web service recommendation via combining bilinear graph representation and xDeepFM quality prediction. *IEEE Trans Netw Serv Manag*, (early access). <https://doi.org/10.1109/TNSM.2023.3234067>
- Mezni H, Benslimane D, Bellatreche L (2022) Context-aware service recommendation based on knowledge graph embedding. *IEEE Trans Knowl Data Eng* 34(11):5225–5238
- Jiang Y, Tao D, Liu Y et al (2019) Cloud service recommendation based on unstructured textual information. *Futur Gener Comput Syst* 97:387–396
- Ma H, Huang Z, Zhang X et al (2022) Cloud service recommendation for small and medium-sized enterprises: a context-aware group decision making approach. *J Intell Fuzzy Syst* 42(3):1331–1351
- Lei C, Dai H, Yu Z et al (2020) A service recommendation algorithm with the transfer learning based matrix factorization to improve cloud security. *Inf Sci* 513:98–111
- Liu J, Chen Y, Liu Q et al (2023) A similarity-enhanced hybrid group recommendation approach in cloud manufacturing systems. *Comput Ind Eng* 178:109128
- Zhang C, Zhao F, Wang Z (2019) Modeling of Cloud 3D printing service hyper-network in service-oriented manufacturing systems. *IEEE Access* 8:16225–16235

31. Zhang C, Liu J, Xu B et al (2020) Architecture of cloud 3D printing task modeling for nodes dynamic scheduling and coupling based on complex networks. *IEEE Access* 8:135208–135222
32. Achir M, Abdelli A, Mokdad L et al (2022) Service discovery and selection in IoT: a survey and a taxonomy. *J Netw Comput Appl* 200:103331
33. Balaji BS, Balakrishnan S, Venkatachalam K et al (2021) Automated query classification based web service similarity technique using machine learning. *J Ambient Intell Humaniz Comput* 12(6):6169–6180
34. Shen J, Huang W, Qiang Hu (2022) PICF-LDA: a topic enhanced LDA with probability incremental correction factor for Web API service clustering. *J Cloud Comput* 11(1):1–13
35. Janani R, Vijayarani S (2019) Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Syst Appl* 134:192–200
36. Kang Z, Shi G, Huang S et al (2020) Multi-graph fusion for multi-view spectral clustering. *Knowl-Based Syst* 189:105102
37. Afzalan M, Jazizadeh F (2019) An automated spectral clustering for multi-scale data. *Neurocomputing* 347:94–108
38. Song YT, Wu S (2020) Slope one recommendation algorithm based on user clustering and scoring preferences. *Procedia Comput Sci* 166:539–545
39. Fayala M, Mezni H (2020) Web service recommendation based on time-aware users clustering and multi-valued QoS prediction. *Concurr Comput Pract Exp* 32(9):e5603
40. Noulapeu Ngaffo A, El Ayeb W, Choukair Z (2022) A service recommendation approach based on trusted user profiles and an enhanced similarity measure. *Electron Commer Res* 22:1537–1572
41. Gao M, Ling B, Yang L et al (2019) From similarity perspective: a robust collaborative filtering approach for service recommendations. *Front Comp Sci* 13(2):231–246
42. Wang L, Zhang X, Wang T et al (2020) Diversified and scalable service recommendation with accuracy guarantee. *IEEE Trans Comput Soc Syst* 8(5):1182–1193
43. Ngaffo AN, Ayeb WE, Choukair Z (2022) Service recommendation driven by a matrix factorization model and time series forecasting. *Appl Intell* 52(1):1110–1125
44. Priya AS, Bhuvaneswaran RS (2021) Cloud service recommendation system based on clustering trust measures in multi-cloud environment. *J Ambient Intell Humaniz Comput* 12:7029–7038

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)