IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON HUMAN-CENTERED SMART SYSTEMS AND TECHNOLOGIES

Received April 29, 2018, accepted May 28, 2018, date of publication June 18, 2018, date of current version July 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2847631

P-CSREC: A New Approach for Personalized Cloud Service Recommendation

CHENGWEN ZHANG[1], ZENGCHENG LI[2], TANG LI[2], YUNAN HAN[3,4], CUICUI WEI[2], YONGQIANG CHENG[5], YONGHONG PENG[6]

[1]Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[4]Beijing University of Chemical Technology, Beijing 100029, China
[5]School of Engineering and Computer Science, University of Hull, Hull HU6 7RX, U.K.
[6]Faculty of Computer Science, University of Sunderland, Sunderland SR6 0DD, U.K.

Corresponding author: Yunan Han (hanyn@mail.buct.edu.cn)

This work was supported in part by the Funds for International Cooperation and Exchange of NSFC under Grant 61720106007, in part by the National Key Research and Development Program of China under Grant 2017YFB1400603, in part by the National Natural Science Foundation project of China under Grant 61772479, and in part by the Open Foundation of State Key Laboratory of Networking and Switching Technology of Beijing University of Posts and Telecommunications under Grant SKLNST-2018-1-02.

ABSTRACT It is becoming a challenging issue for users to choose a satisfied service to fit their need due to the rapid growing number of cloud services and the vast amount of service type varieties. This paper proposes an effective cloud service recommendation approach, named personalized cloud service recommendation (P-CSREC), based on the characterization of heterogeneous information network, the use of association rule mining, and the modeling and clustering of user interests. First, a similarity measure is defined to improve the average similarity (AvgSim) measure by the inclusion of the subjective evaluation of users' interests. Based on the improved AvgSim, a new model for measuring the user interest is established. Second, the traditional K-Harmonic Means (KHM) clustering algorithm is improved by means of involving multi meta-paths to avoid the convergence of local optimum. Then, a frequent pattern growth (FP-Growth) association rules algorithm is proposed to address the issue and the limitation of traditional association rule algorithms to offer personalization in recommendation. A new method to define a support value of nodes is developed using the weight of user's score. In addition, a multi-level FP-Tree is defined based on the multi-level association rules theory to extract the relationship in higher level. Finally, a combined user interest with the improved KHM clustering algorithm and the improved FP-Growth algorithm is provided to improve accuracy of cloud services recommendation to target users. The experimental results demonstrated the effectiveness of the proposed approach in improving the computational efficiency and recommendation accuracy.

INDEX TERMS Association rules, clustering, heterogeneous information network, personalized cloud service recommendation, user interest model.

I. INTRODUCTION

Cloud service is a new kind of network service relying on cloud computing platform. Cloud service providers are continuously releasing various configuration in terms of software applications, computing power, storage capacity, network resources et al. in the forms of SAAS(Software As A Service) [1], IAAS (Infrastructure As A Service) [2], PAAS (Platform As A Service) [3] et al. Cloud service libraries have thus been quietly formed. Confronted with a number of services with similar or even identical functions, it is complicated enough for majority of users to determinate an ideal cloud service that can fit their specific needs.

Services in a cloud environment are normally in form of cloud applications which usually exists in the form of Web services. Hence most of the current cloud service recommendation methods like CF (Collaborative Filtering) [4] and association rules [5] share the same methods used for recommendation of Web services. Although service recommendation algorithms for cloud service have been significantly improved to address the overload problem of services on a cloud platform, there are still unsolved issues in current Web service recommendation methods, including data sparsity issues, cold start issues and real-time issues [6], [7]. All the current problems of service recommendation algorithms have

been found to be attributed to two issues, low efficiency and insufficient accuracy.

In this paper we propose a new algorithm for cloud service recommendation. It employs the heterogeneous information of cloud service as key, and uses the associate rule for clustering analysis of user interests and the associated services recommendation. Low calculation efficiency can lead to long running time, and insufficient accuracy can lead to inaccurate recommendation results. The experimental results showed the improvement of the efficiency and accuracy in the cloud service recommendation algorithm.

The contributions of the paper can be summarized as below:

1) A new similarity measure method taking into account subjective evaluation

2) A new user clustering method based on multiple meta-paths

3) Improved FP-Growth association rules for personalised recommendation

The rest of the paper is organized as following: Section II summarizes the related work; Details of the proposed work is described in Section III with discussions in section IV; Section V gives the experiment and Section VI concludes the paper.

II. RELATED WORK

A. HETEROGENEOUS CLOUD SERVICE NETWORK

Assume an information network can be defined as a directed network graph G = (V, E) , where V is the set of all physical nodes and E is the set of all the relevant edges. And there is a mapping function ϕ of the node type: V $\rightarrow$ A and an edge type mapping function $\psi : E \rightarrow R$. For each object v $\in$ V belongs to a special object type ϕ(v) $\in$ A, each link e $\in$ E belongs to a special type of relationship $\psi(e) \in R$. When the type of the object type |A| $>$1 or the type of the relationship type |R| $>$1, this information network is called as a heterogeneous information network; otherwise, it is called as a homogeneous information network [8].

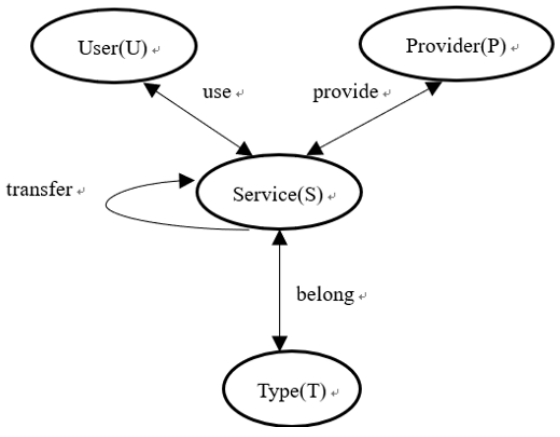


FIGURE 1. Heterogeneous cloud service network.

Compared to a homogeneous information network, a heterogeneous information network consists of different types of nodes and edges.

As shown in Figure 1, the heterogeneous service network has four different object types, users (U), services (S), types (T), and providers (P). A heterogeneous service information network includes a wealth of node types and link relationships, from which it thus can dig out relationships not only between objects of services (S), but also between services (S) and other objects.

Whist in the homogenous information network, we can only analyze the association in a single type of object, such as analyzing only the association between users or the correlation between services. This method of analysis has great limitations and does not dig deep into the objects. Compared to the homogeneous information network, the heterogeneous network reflects the diverse data objects in the entire system and the complex relationships between them.

B. SIMILARITY CALCULATION IN HETEROGENEOUS SERVICE NETWORK

For measuring the similarity between data object nodes in a heterogeneous information network, the differences between different link paths should be taken into consideration. Different from the homogeneous information network, two object nodes in the heterogeneous information network can be connected by different link relationships, which reflect the different meanings of the diverse link relationships.

A meta path of a network is denoted by P $= A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$, which establishes the composition of relations R $= R_1^{\circ} R_2^{\circ} \ldots^{\circ} R_l$ between entity types A_1 and A_{l+1}, where $^{\circ}$ denotes the composition operator of relations.

In Figure 1, U $\rightarrow$ S $\rightarrow$ T is an asymmetric meta path, representing that the user uses the same type of service. U $\rightarrow$ S $\rightarrow$ T $\rightarrow$ S $\rightarrow$ U is a symmetric meta path, representing that two users are using the same type of service.

Some similarity measure algorithms have been applied for the analysis of the heterogeneous information network in literature, e.g. PathSim (Path Similarity) [9], PCRW (Path-Constrained Random Walks) [10], HeteSim (Heterogeneous Similarity) [11], AvgSim (Average Similarity) [12] et al. All those mentioned methods measure the similarity between nodes based on meta paths.

With the measure of PathSim two objects of the same type are not only considered to be closely related, but also share the subtle semantics of peer similarity. The peer relationship is symmetric in the PathSim, so it can only measures the similarities between nodes of the same type [13].

The PCRW, based on meta paths and the random walk model, calculates the similarities based on the forwards and backwards path of the network. However it does not distinguish between them.

The HeteSim, based on meta paths and the bidirectional random walk model, distinguishes the forward and backward path however it has significant computational complexity in

both time and space. The AvgSim, an improved method based on HeteSim, compensates for HeteSim's shortcomings. The AvgSim algorithm does not consider the user ratings natively, so it is not able to effectively capture the users' interests. Wang et al. proposed the concept of weighted meta path [14], which take into account the properties in different meta paths in calculating similarity. Therefore, it is feasible to add the information of user ratings to the calculation of AvgSim based on meta path.

The paper will focus on AvgSim algorithm hence only AvgSim is described in detail here. The principle of the AvgSim algorithm can be described as follows: The similarity between node pairs s and d is the average of the reachability probability from the source node s to the target node d based on the specific element path P and the reachability probability from the target node d to the source node s travelling on the inverse element path P^{-1}.

$$\begin{aligned}&\text{AvgSim}\,(s, d|P)\\&\quad=\frac{1}{2}\left[\text{RW}\,(s, d|P)+\text{RW}\left(d, s|P^{-1}\right)\right]\end{aligned}\tag{1}$$

$$\begin{aligned}&\text{RW}\left(s, d|R_1\circ R_2\circ\ldots\circ R_l\right)\\&\quad=\frac{1}{|O(s|R_1)|}\sum_{i=1}^{|O(s|R_1)|}\text{RW}\left(O_i\,(s|R_1)\,, d|R_2\circ R_3^{\circ}\ldots\circ R_l\right)\end{aligned}\tag{2}$$

The probability that the source node s walks randomly and reaches the target node d via the meta path P is defined as RW(•) in (2), where $O(s|R_1)$ denotes the out-node set of node s in R1. This equation calculates the similarities iteratively among all the out-nodes from s and the target node d and the sum of all the similarities. The calculation keeps processing until reaching d.

If s and d are identical, then the similarity is 1, otherwise 0. When s does not have any out-node along the meta path (i.e. $O\,(s|R_1)=\emptyset$), then s cannot reach d. In this case their similarity is defined as 0.

For the simplest relationship, $A \xrightarrow{R} B$, A and B are correlated with relation R, where A and B denote node sets of the same type. The relationship between A and B can be represented with an adjacency matrix, denoted by M_{AB}. However, the adjacency matrix does not reflect the direction of the path. Therefore, M_{AB} is regulated by row and by column and get two regulated matrices, R_{AB} and C_{AB}, respectively. These are two transition probability matrices representing two directed relationships, $A \xrightarrow{R} B$ and $B \xrightarrow{R^{-1}} A$. Those two matrices have the following properties:

$$\begin{cases}R_{AB}=C'_{BA}\\C_{AB}=R'_{BA}\end{cases}\tag{3}$$

Where (R'_{AB} is the transpose of R_{AB}):

Equation (3) describes a simple relation R. For a given path $P=A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$ of a compositive relation $R=R_1\circ R_2\circ\ldots\circ R_l$, the reachable matrix between A_1 and A_{l+1} on path P is defined as $RW_P=R_{A_1A_2}R_{A_2A_3}\ldots R_{A_lA_{l+1}}$, which represents the probability that A_1 reaches A_{l+1} randomly along the path P.

The AvgSim based on transition probability matrix and reachable matrix with probability is calculated as follows:

$$\begin{aligned}\text{AvgSim}\,(A_1, A_{l+1}|P)=\frac{1}{2}\,[&\text{RW}(A_1, A_{l+1}|P)\\&+\text{RW}(A_{l+1}, A_1|P^{-1})\Big]\end{aligned}\tag{4}$$

Arithmetic average calculation of the two reachable matrices is needed in (4). The dimensions of the two matrices should be equal, so the reachable matrix of the reversed direction needs transposing. Based on the properties of (3), the Equation (4) can be rewritten as the following:

$$\begin{aligned}&\text{AvgSim}\,(A_1, A_{l+1}|P)\\&=\frac{1}{2}\left[R_{A_1A_2}R_{A_2A_3}\ldots R_{A_lA_{l+1}}+\left(R_{A_{l+1}A_l}\ldots R_{A_3A_2}R_{A_2A_1}\right)'\right]\\&=\frac{1}{2}\left[R_{A_1A_2}R_{A_2A_3}\ldots R_{A_lA_{l+1}}+\left(R'_{A_2A_1}R'_{A_3A_2}\ldots R'_{A_{l+1}A_l}\right)\right]\\&=\frac{1}{2}\left[R_{A_1A_2}R_{A_2A_3}\ldots R_{A_lA_{l+1}}+C_{A_1A_2}C_{A_2A_3}\ldots C_{A_lA_{l+1}}\right]\end{aligned}\tag{5}$$

Thus, two random walks' reachable matrices with probability are in the form of chained products of transition probability matrices from A_1 to A_{l+1}.

C. KHM CLUSTERING ALGORITHM

Clustering is to divide a data object set into groups so that the data objects with high similarity will be sitting in the same group, while those having low similarity will sit into different groups.

The K-Means is the most widely used approach for clustering analysis. However, the selection of initial cluster centers in K-Means has a great influence on the final clustering effect, and the result converges easily to a local optimum. To address this issue, we proposed a new KHM (K-Harmonic Means) algorithm in the following article to calculate distances between objects. An improved algorithm called KHM has been proposed in literature based on which the clustering result is not susceptible to the selection of initial cluster centers. However, the KHM algorithm still tends to converge to a local optimum.

The main differences between KHM and K-Means are as follows [15]:

1) KHM uses k-harmonic means of distances between data elements and the cluster centers to replace the minimum distances used in K-Means. The k-harmonic mean of n pieces of data is calculated as follows:

$$\text{HA}=\frac{n}{\sum_{i=1}^{n}\frac{1}{a_i}}\tag{6}$$

where a_i is the i-th number. The k-harmonic means of distance between Xi and cluster centers are thus defined by:

$$\text{HA}\,(X_i)=\frac{k}{\sum_{j=1}^{k}\frac{1}{d^p(X_i,m_j)}}\;(p\geq 2)\tag{7}$$

In (7), X_i is an element in data set X, m_j denotes the centers of clusters, d (X_i, m_j) is the distance measure function, and p denotes the input parameter.

2) The objective function of KHM is defined as a sum of k-harmonic means of distances between all elements and cluster centers as follows:

$$E_{KHM} = \sum_{i=1}^{n} \mathrm{HA}\,(X_i) \tag{8}$$

3) The degree of membership b $(m_j|X_i)$ of each data element X_i to each cluster center m_j is defined by:

$$\mathrm{b}\left(m_j|X_i\right) = \frac{d^{-p-2}\left(X_i, m_j\right)}{\sum_{l=1}^{k} d^{-p-2}\left(X_i, m_l\right)} \tag{9}$$

4) The weight w (X_i) of each data element is calculated by:

$$\mathrm{w}\,(X_i) = \frac{\sum_{j=1}^{k} d^{-p-2}\left(X_i, m_j\right)}{\left(\sum_{j=1}^{k} d^{-p}\left(X_i, m_j\right)\right)^2} \tag{10}$$

5) The iteration to generate a new cluster center in KHM by:

$$m_j = \frac{\sum_{i=1}^{n} b\left(m_j|X_i\right) w\,(X_i)\, X_i}{\sum_{i=1}^{n} b\left(m_j|X_i\right) w\,(X_i)} \tag{11}$$

There have been a variety of improved versions of KHM aimed at tackling the drawback of KHM that tends to converge to a local optimum. However, current studies on KHM are based on single object types, unable to be applied to the heterogeneous information networks of cloud service. In this paper we proposed a new approach to improve KHM algorithm based on more than one meta paths for heterogeneous information networks in cloud service.

In heterogeneous information service networks, due to their multiple object types, relationships between users are reflected by not only using the same type of services, but also the same type of providers, which is exactly the abundant information in different meta paths.

Compared to a single meta path, multiple meta paths involve a wider range and are more global. The clustering analysis the users' profiles based on the multiple meta paths offers deep insight into the correlations between users, which can hopefully overcome the drawback of converging to a local optimum.

D. FP-GROWTH ASSOCIATION RULE RECOMMENDATION ALGORITHM

The association rules are defined as in the implication forms like $\mathrm{X} \rightarrow \mathrm{Y}$, with $X \bigcap Y = \Phi$. Association rules mining are defined based on the support and confidence coefficients. The support is the probability that X and Y appear together in database, i.e. sup $(\mathrm{X} \rightarrow \mathrm{Y}) = \mathrm{P}(\mathrm{X} \bigcup \mathrm{Y})$. Higher probability indicates a stronger correlation. The confidence is the probability of a database that contains Y which also contains X, i.e.conf $(\mathrm{X} \rightarrow \mathrm{Y}) = \mathrm{P}(\mathrm{X}|\mathrm{Y})$, which is a conditional probability of X given Y.

The association mining is to find association rules where support and confidence meet requirements of user-defined minimum support minSup and minimum confidence minConf. If a rule, $\mathrm{X} \rightarrow \mathrm{Y}$, whose support and confidence are greater than the minSup and minConf respectively, then this rule is valid, called the strong association rule. If the support of $\mathrm{X} \rightarrow \mathrm{Y}$ is greater than minSup, then it is called a frequent pattern (FP). An item set X is called a frequent item set when it meets the minimum support threshold.

Association rule generation is usually achieved by two separate steps: (1) to find all frequent item sets; (2) to form the association rules. To find frequent item sets there have been two well-known algorithms like Apriori and FP-Growth (Frequent Pattern -Growth).

Apriori uses a breadth-first search strategy to obtain all the frequent item sets, which faces challenges in terms of time efficiency and spatial scalability. FP-Growth, fundamentally different from Apriori, transfers the data onto a FP-Tree (Frequent Pattern-Tree). It only scans databases twice so that it is significantly faster than Apriori.

FP-Growth is usually split up into two separate steps: (1) to establish an FP-Tree; (2) to mine the FP-Tree recursively [19]. The traditional FP-Growth algorithm can have issue when data set grouping is imbalance. To address this, Zhang proposed an improved FP-growth algorithm [16] which effectively improves the load imbalance and clustering efficiency.

Similar to Zhang's work [16], Chen [17] and Niu [18] improved FP-Growth respectively, for reducing the mining time of frequent item sets and accelerating the generation of association rules. The existing studies of FP-Growth algorithm have been focused on the reduction of mining time of frequent item sets and increasing the speed of rule generation, but less concerning the quality of mined frequent item sets.

In this paper, we improve the FP-Growth when counting the support of each node, the prefix paths involving the user ratings so that the mined frequent item sets meet the user preferences.

The process of mining frequent item sets based on FP-Growth is summarized as follows [19]:

1) Generate all frequent 1-item sets. The algorithm traverses a data item set I and counts the occurrence of items. Frequent 1-item sets refer to the data items that pass the threshold of minimum support.

2) Build frequent item header table. The frequent 1-item sets are sorted in descending in terms of the support count and add the newly built tree node to the header table.

3) Build a FP-Tree based on transaction set. For the first transaction T_1, the data items are sorted in the order of the header table. For the first data item i_1, we search for its child nodes. Assign it to be null if the node is with the same name as i_1. If the node exists, the support count of this node is increased by 1. Otherwise a new node is added to this data item, denoted by node j. Then the node's next node with same name is pointed to i_1.

4) Repeat step 3) until all transactions are added to the FP-Tree.

5) Based on the established FP-Tree, for each item in the header table, the associated ancestor path is searched to construct new item set from nodes existed in the path which is called conditional pattern base. For each item in the header table, conditional FP-Trees are constructed based on all of their conditional pattern bases.

6) For each item in the header table, mine their frequent patterns in conditional FP-Tree. Infrequent data items are removed when they do not pass the minimum support threshold and the left data items in the path are combined with items in the currently searching header table to form the frequent patterns.

7) Repeat steps 5) and 6) until all of frequent item sets are found.

III. THE PROPOSED ALGORITHM: P-CSREC

The proposed P-CSREC algorithm comprises of improved AvgSim algorithms for objects similarity calculation; User interests model to incorporate user ratings; improved KHM algorithm for users clustering; FP-Growth algorithm for FP-Tree building and cloud service recommendation.

A. IMPROVED AVGSIM ALGORITHM

In order to adapt to the application situation of heterogeneous information service network with user ratings, an attribute of user ratings is added to the meta path. The improved AvgSim algorithm, named as HAvgSim (Heterogeneous AvgSim), is performed as follows:

$$\mathrm{HAvgSim}(\mathrm{s},\mathrm{d}|\mathrm{P}) = \frac{1}{2}\left[\mathrm{RW}'(\mathrm{s},\mathrm{d}|\mathrm{P}) + \mathrm{RW}'\left(\mathrm{d},\mathrm{s}|\mathrm{P}^{-1}\right)\right] \tag{12}$$

$$\mathrm{RW}'\left(\mathrm{s},\mathrm{d}|\mathrm{R}_1{}^{\circ}\mathrm{R}_2{}^{\circ}\ldots{}^{\circ}\mathrm{R}_l\right) = \frac{1}{\mathrm{x}}\frac{1}{|O(s|R_1)|}\sum_{i=1}^{|O(s|R_1)|} RW'(O_i(s|R_1),\mathrm{d}|\mathrm{R}_2{}^{\circ}\mathrm{R}_3{}^{\circ}\ldots{}^{\circ}\mathrm{R}_l)\times \mathrm{w_i} \tag{13}$$

In (13), $\mathrm{w_i}$ denotes the user rating of source node s based on relationship of path $\mathrm{R_1}$ to node i and x denotes the full score of rating.

Figure 2 is a simple heterogeneous information service network used to illustrate the process of HAvgSim.

In the simple heterogeneous information service network as shown in Figure 2, there are 3 users, u_1, u_2, u_3, and 4 services, s_1, s_2, s_3, s_4, and 2 types of service, t_1, t_2. User u_1 has made ratings on s_1, s_2, s_3 respectively as 1.5, 1, 5; User u_2 has ratings on s_1, s_4 respectively of 3 and 4; User u_3 has ratings on s_3, s_4 respectively of 1.5 and 5. Among them, the rating range is between 0 and 5. Based on the meta path, UST(User-Service-Type), HAvgSim is applied to measure the similarity between user u_2 and type t_1. Apply the data on the Equations (12) and (13) and we can get following:

$$\mathrm{HAvgSim}(\mathrm{u}_2, t_1|\mathrm{UST}) = \frac{1}{2}\left[\mathrm{RW}'(u_2, t_1|\mathrm{UST}) + \mathrm{RW}'(t_1, u_2|\mathrm{TSU})\right] \tag{14}$$

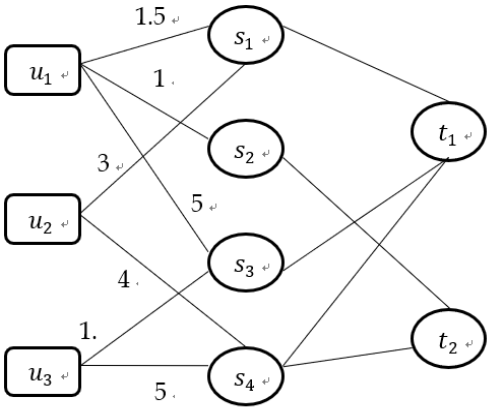


FIGURE 2. A simple heterogeneous service network.

$$\mathrm{RW}'(u_2, t_1|\mathrm{UST}) = \frac{1}{5}\frac{1}{|O(u_2|\mathrm{US})|}\sum_{\mathrm{i}=1}^{|O(u_2|\mathrm{US})|} \mathrm{RW}'(\mathrm{O_i}(u_2|\mathrm{US}), t_1|ST)\times \mathrm{w_i} \tag{15}$$

As shown in Figure 2, we can see that $\mathrm{O}(u_2|\mathrm{US}) = \{s_1, s_4\}$. The probability is first calculated for the path that u_2 reaches t_1 through the node* of user-service (US). s_1 and s_4 both can reach t_1, i.e., $\mathrm{O}(s_1|\mathrm{ST}) = t_1$ and $\mathrm{O}(s_4|\mathrm{ST}) = t_1$, so $\mathrm{RW}'(s_1, t_1|\mathrm{ST}) = 1\times\frac{3}{5}$, $\mathrm{RW}'(s_4, t_1|\mathrm{ST}) = 1\times\frac{4}{5}\times\frac{1}{2}$, $\mathrm{RW}'(u_1, t_1|\mathrm{UST}) = \frac{1}{2}\left(\frac{3}{5}+\frac{2}{5}\right) = 0.5$. Similarly, the inverse weighted probability calculated by TSU (Type-Service-User) is 0.467. Finally, the similarity between $\mathrm{u_2}$ and t_1 is 0.5+0.467/2=0.4835.

B. USER INTEREST MODEL BASED ON THE IMPROVED AVGSIM ALGORITHM

In a heterogeneous information service network, a user's preference to some service can be represented by the similarity or likeness between users and services. A higher similarity or likeness indicates that the user is more interested in this service. Therefore, we build user-service similarity matrices, based on HAvgSim, and then establish user interest models, by which we calculate the user's interest in services or types.

Firstly, the user-service rating matrix is built based on the ratings of services given by users. Then, the similarity between certain service or service type and a user is calculated using the HAvgSim algorithm proposed in this paper. The similarity calculated is the expected value of interest. The interest model is shown in Figure 3:

C. USER CLUSTERING BASED ON IMPROVED KHM ALGORITHM

The KHM clustering algorithm is improved, in this paper, by involving two different meta paths, USU(User-Service-User) and USTSU(User-Service-Type-Service-User). The path USU represents the relationship between users who use the same services. The path USTSU represents the relationship between users who use the same type of services.

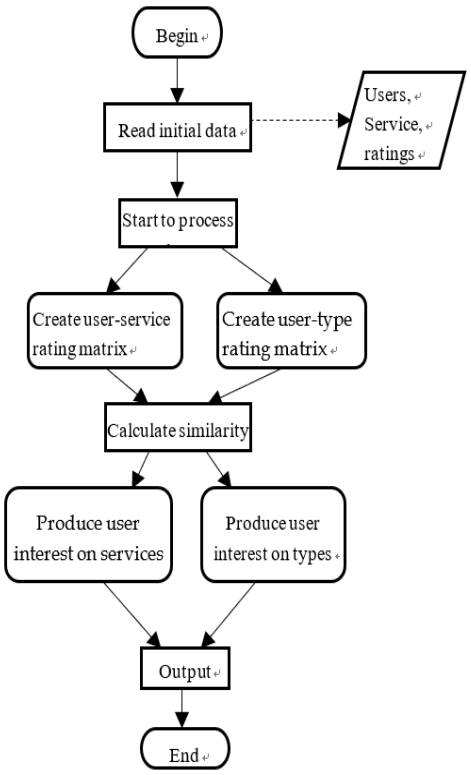


FIGURE 3. The proposed user interest model.

Let $U = \{U_1, U_2, \ldots U_n\}$ be a set of users, n is the number of users, and U_i $(i \in [1 \ldots n])$ is the i-th user. $S = \{S_1, S_2, \ldots S_r\}$ is defined as the set of all services, r is the number of services, and S_a $(a \in [1 \ldots r])$ is the a-th service. $T = \{T_1, T_2, \ldots T_q\}$ defines a set of service types, q is the number of service types, and T_b $(b \in [1 \ldots q])$ is the b-th type. $x_{i0} = \{s_{i1}, s_{i2}, \ldots, s_{ir}\}$ is the interest vector defining the user interest of U_i to the services, where s_{ia} $(a \in [1 \ldots r])$ is the interest degree of user U_i to the a-th service. $x_{i1} = \{t_{i1}, t_{i2}, \ldots, t_{iq}\}$ is the interest vector defining the user interest of U_i to the service types, where t_{ib} $(b \in [1 \ldots q])$ is the interest degree of user U_i to the b-th service type.

A user U_i has two interest vectors x_{i0} and x_{i1}. The similarity between user U_i and user U_j is calculated based on their interest vectors x_{i0}, x_{i1} and x_{j0}, x_{j1} is defined by $s(U_i, U_j)$:

$$s\left(U_i, U_j\right) = \left(\sum_{a=1}^{r}\left(\frac{min\left(s_{ia}, s_{ja}\right)}{max\left(s_{ia}, s_{ja}\right)}\right)^2 + \sum_{b=1}^{q}\left(\frac{min\left(t_{ib}, t_{jb}\right)}{max\left(t_{ib}, t_{jb}\right)}\right)^2\right) \Big/ 2 \quad (16)$$

In (16), $min\left(s_{ia}, s_{ja}\right)$ denotes the smaller value of the interest degree of user U_i and U_j to the same service S_a and $max\left(s_{ia}, s_{ja}\right)$ is the bigger one. $\left(min\left(s_{ia}, s_{ja}\right)/max\left(s_{ia}, s_{ja}\right)\right)^2$ measures the similarity of U_i and U_j based on a single service S_a and $\sum_{a=1}^{r}\left(min\left(s_{ia}, s_{ja}\right)/max\left(s_{ia}, s_{ja}\right)\right)^2$ measures the similarity based on all the services on meta path USU. Similarly, $\sum_{b=1}^{q}\left(min\left(t_{ib}, t_{jb}\right)/max\left(t_{ib}, t_{jb}\right)\right)^2$ represents the similarities of all service types of U_i and U_j based on meta path USTSU. Finally, normalization is performed that the average of the result is the similarities of U_i and U_j based on two meta paths USU and USTSU, as is shown in (16).

Based on the similarity between U_i and U_j, the distance between them can then be defined as follows:

$$d^2\left(U_i, U_j\right) = \frac{1}{s\left(U_i, U_j\right)} \quad (17)$$

In summary the process of the proposed method for the improved KHM is calculated as follows:

1) Set the number of clusters k, where $1 \leq k \leq n$.

2) Generate k random clusters from all the users as initial cluster centers $M = \{m_1, m_2, \ldots m_k\}$.

3) Calculate all the similarities $s\left(U_i, m_j\right)$ between user U_i and cluster center m_j.

4) Calculate the harmonic means of similarities of every users and every cluster centers, and user U_i is the data X_i in (7):

$$HA\left(U_i, m_j\right) = \frac{k}{\sum_{j=1}^{k} s^{p/2}(U_i, m_j)} \quad (18)$$

and calculate the value of target function E_m (m is the number of iteration):

$$E_m = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{k} s^{p/2}\left(U_i, m_j\right)} \quad (19)$$

5) Calculate the membership from every user U_i to every cluster center $m_j b\left(m_j|U_i\right)$:

$$b\left(m_j|U_i\right) = \frac{s^p\left(U_i, m_j\right)}{\sum_{l=1}^{k} s^p\left(U_i, m_l\right)} \quad (20)$$

6) Calculate the weight $w\left(U_i\right)$ of every user:

$$w\left(U_i\right) = \frac{\sum_{j=1}^{k} s^p\left(U_i, m_j\right)}{\left(\sum_{j=1}^{k} s^{p/2}\left(U_i, m_j\right)\right)^2} \quad (21)$$

7) Generate new cluster centers m_j:

$$m_j = \frac{\sum_{i=1}^{n} b\left(U_i, m_j\right) w\left(U_i\right) U_i}{\sum_{i=1}^{n} b\left(U_i, m_j\right) w\left(U_i\right)} \quad (22)$$

8) Calculate the target function E_{m+1} and update the number of iteration to m+1. If $|E_{m+1} - E_m| \leq \varepsilon$ (ε is a certain threshold set before) or the number of iteration meet the upper threshold, then the target function converges and clustering ends. Otherwise, return to Step 4.

9) Distribute U_i, according to the membership $b\left(m_j|U_i\right)$, to a class cluster C_j with maximum membership.

10) Output the cluster result $C = \{C_1, C_2, \ldots C_k\}$, which meet the conditions.

D. IMPROVED FP-GROWTH ALGORITHM

Ratings from users to services reflect preferences of users to services. If FP-Growth employs the user's ratings as the weight to calculate the support count of nodes, it is expected that the extracted frequent item sets reflect better the users' interests.

In addition, a heterogeneous information service network has objects like 'services', 'types', 'providers' et al. If only the layer of 'services' has been analyzed it will lead to many abandonments of services due to failure to meet the support count threshold, where some useful association information between services fails to be mined. If the 'types' or 'providers' that these 'services' belong to could meet the support count threshold to some extent, then finding frequent item sets related to these 'types' or 'providers' could be helpful to mine some useful information from a higher level, which is the concept of multi-level association rules mining [20]. Therefore, we have improved the FP-Growth in terms of support count of nodes and multi-level.

Firstly, referencing to [21], we added two fields, visited and ancestor List, which overcome the problem of repeated scan. Secondly, we modified traditional support count to the sum of all the user ratings weights on the prefix path, so that the accuracy of frequent item set mining was improved. Lastly, we added a pointer to the upper level nodes to every node, built a two-level FP-Tree, and defined the upper level nodes as 'types'. When a support count of a certain 'service' fails to meet the requirement and its 'type' node meet the support threshold, we mine the frequent items of 'type'. Decreasing support thresholds are used in the two levels.

The improved FP-Tree based on Table 1 is shown in Figure 4, where the support counts of nodes are the sum of user ratings weights on the prefix paths. If the full score is out of 5 and a user gives it a 4 as the score, then the rating weight is 4/5=0.8.

TABLE 1. Service transaction set.

TID	Transactions (Service:Rate)
T_1	$\{(S_1:4),(S_2:2),(S_3:3.5)\}$
T_2	$\{(S_2:5),(S_1:2)\}$
T_3	$\{(S_3:5),(S_2:3.5)\}$
T_4	$\{(S_1:1),(S_3:4.5)\}$

Next, a two-level FP-Tree is built. As shown in Figure 5, the minimum of support count for the level of 'type' is set to 20, while the minimum support count of the level of 'service' is 13.

If two relationships such as S_1 and S_2 as shown in Figure 5, occurs in conditional FP-Tree, where S_1 and S_2 neither meet the threshold in the associated levels, then S_1 and S_2 do not appear to be selected as frequent item sets. If both of them are pruned and abandoned, some useful information would

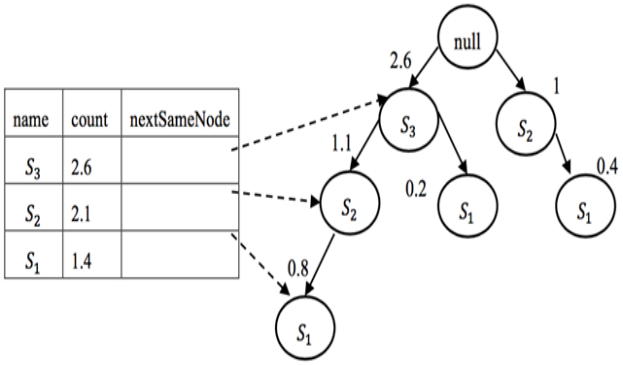


FIGURE 4. FP-Tree based on service transaction set.

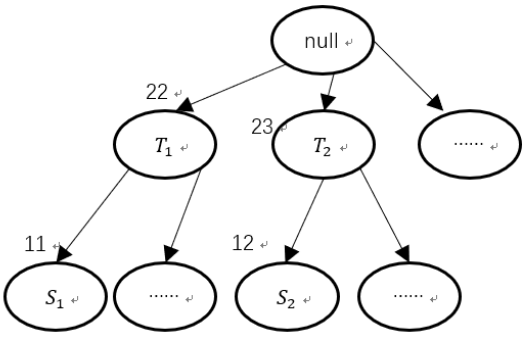


FIGURE 5. FP-Tree with services and types.

be lost. To prevent this, we checked the 'type' nodes of S_1 and S_2, noting that both T_1 and T_2 meet the minimum support thresholds in their levels. Therefore, we replaced S_1 with T_1, the 'type' of S_1, and got frequent item sets $< T_1T_2 >$. This indicates that when type T_1 shows up, the probability that T_2 shows up is high. If one uses more services with T_1 type, then T_2 is recommended to him or her, so that associations in abstract level are extracted.

As mentioned above, when the item sets in the lower level fail to meet the minimum support threshold but those in the higher level make it, the latter will be extracted. We did not extract all the 'types' in the higher level to calculate the frequent item sets. The lower level the frequent item sets are extracted at, the more accurately the recommendations can be.

IV. THE NEW SERVICE RECOMMENDATION APPROACH

When there are too many pending transaction databases, large size of memory is needed to build a FP-Tree, and also the need of time for mining the frequent item-sets iteratively would increase.

Using clustering approach to divide the huge service set and to do FP-Growth on various smaller service sets could reduce significantly the pressure comparing to all the services that are involved in a FP-Tree, so that the time and space complexity would be reduced.

The steps for cloud services recommendation based on FP-Growth association rules and user cluster are as the following,

Firstly, HAvgSim and user interest models are used to calculate uses' interests to all the services and service types.

Next, KHM cluster computing is based on users meta paths and users' interest value to get user clustering results.

Then, FP-Growth is used to mine service association rules on service-sets that all users use in clusters. Users were recommended services according to association rules.

The clustering algorithm divides users into groups. Each user group uses a group of services. Hence the number of services used by each user group is a subset of the overall services which reduces the computational time cost and space complexity. All the users in a cluster have similar interests, so the accuracy of service recommendation in clusters is also improved.

V. EXPERIMENTS

In the heterogeneous information network as shown in Figure 1, there are four different objects, 'user (U)', 'service (S)', 'type (T)', and 'provider (P)'. We can mine from it the relationship between 'service (S)' and 'service (S)' as well as 'service(S)' and other objects.

Similarly, as shown in Figure 6, there are object types such as 'user (U)', 'movie (M)', 'actor (A)', 'type (T)', and 'director (D)' in movie heterogeneous information networks.

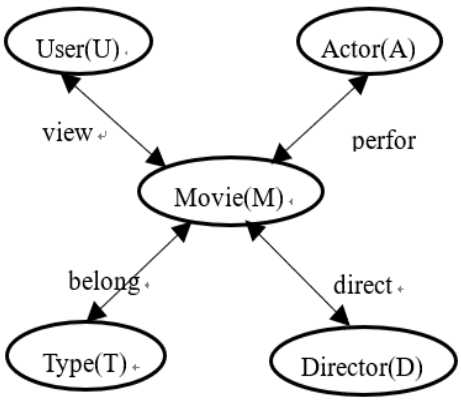


FIGURE 6. Heterogeneous movie network.

It can be seen that there is a similarity between cloud and movie service heterogeneous information network, both reflecting the rating relationships between 'users' and 'cloud services' or between 'users' and 'movies'. Therefore, this experiment used MovieLens dataset to simulate cloud service dataset [22], and used movies to simulate cloud service, to evaluate the effectiveness of the proposed algorithms.

A. DATASET

The MovieLens dataset comprises of viewing information of 671 users, over 20 movie types, 9126 movies, and about 100 thousand valid remarks on these movies from the users. The data are stored in 3 tables, movie information, rating information and tag information.

TABLE 2. Calculation accuracy based on UMT Meta-Path.

User Number	Algorithm	Top 2 Types	Top 3 Types	Top 4 Types	Top 5 Types
100	HAvgSim	0.5000	0.5550	0.6133	0.6200
100	AvgSim	0.4900	0.5450	0.5867	0.6000
200	HAvgSim	0.4600	0.5675	0.6117	0.6288
200	AvgSim	0.4550	0.5350	0.6000	0.6250
300	HAvgSim	0.4633	0.5700	0.6144	0.6333
300	AvgSim	0.4500	0.5583	0.6122	0.6325
400	HAvgSim	0.4650	0.5825	0.6217	0.6369
400	AvgSim	0.4500	0.5675	0.6133	0.6331
500	HAvgSim	0.4920	0.5950	0.6240	0.6400
500	AvgSim	0.4880	0.5830	0.6160	0.6340
600	HAvgSim	0.5017	0.5983	0.6228	0.6421
600	AvgSim	0.4917	0.5917	0.6133	0.6367
671	HAvgSim	0.4978	0.5984	0.6175	0.6420
671	AvgSim	0.4858	0.5931	0.6110	0.6367

B. EXPERIMENTAL METHOD

The algorithms proposed in this paper are implemented by Java in Eclipse 10, an integrated development environment.

The MovieLens dataset are divided into several parts and a comparative experiments are conducted on the data after division.

The assessment criteria of this experiment is the recommendation accuracy (or hit rate).

C. HAVGSIM ALGORITHM

For the movie heterogeneous information network HAvgSim and AvgSim algorithms are applied on meta path UMT(User-Movie-Type) to calculate the similarity between two heterogeneous nodes, 'user' and 'type', so that the advantages and disadvantages of the two algorithms could be compared.

HAvgSim and AvgSim are used to obtain the similarities between 'user' and movie 'type' and the similarities were sorted in descending order. In the data division, 80% of the data is used as the training set and 20% of the data is used as the test set. This can result in better training results and ensure that there are enough test samples.

The bottom 20% data are gathered for verifying data and further processing. If the first k 'types' of the sorted top 80% data appears in the first k of the sorted bottom 20% data, then the algorithm is valid and the hit number is regarded as the accuracy.

On the MovieLens dataset, we select the top 2, 3, 4, 5 types which have high similarity with 'users'. The accuracy for those two methods is shown in Table 2.

We selected the top 5 types with high similarity from Table 2 to simulate the algorithms' accuracy, the experimental result is shown in Figure 7.

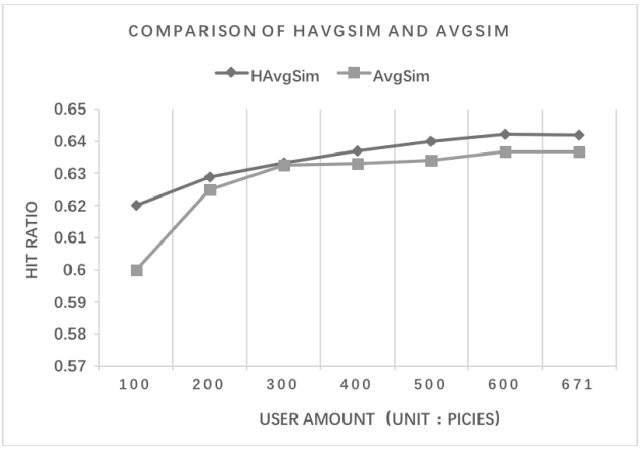


FIGURE 7. Accuracy comparison of HAvgsim and Avgsim.

As one can see from the Table 2, the accuracy of HAvgSim is higher than AvgSim with different user numbers and different numbers of top k types. This is because the improved HAvgSim considers user ratings on meta paths. Adding user ratings to the HAvgSim improve the accuracy in calculating similarity between nodes.

D. IMPROVED KHM ALGORITHM

The improved KHM is applied to cluster users in situations based on the UMU single meta path and the UMU and UMTMU two meta paths respectively.

Traditional clustering algorithms usually take the average of distances between all the elements and the centers as a standard to judge the clustering. The small average means that the elements are closer to each other and the clustering works well. In this paper, the proposed method takes the average of similarities between all the elements and centers as the standard to judge the clustering after the clustering result comes out.

The average of multiple traits of clustering is used as the metric to measure the clustering performance. We did clustering analysis with the improved KHM based on both the single meta path of UMU, and the two meta paths of UMU and UMTMU. The averages of similarities of all clusters we obtained on different user numbers are shown in Table 3.

We simulated more intuitively the data in Table 3 and the result is shown in Figure 8. The higher the average of the similarities, the more alike to each other. As shown in Figure 8, the similarities based on multiple meta paths are higher than those based on a single meta path. This demonstrates that better accuracy of clustering based on multiple meta paths has been achieved with the improved KHM algorithm.

TABLE 3. Clustering Result Based on Single Meta-Path and Multi Meta-Path.

User Number	UMU Single Meta-Path	UMU and UMTMU Multi Meta-Path
100	10.67390928	32.73481872
200	6.968013501	16.62966868
300	4.453264749	13.25071936
400	5.806641878	15.89301354
500	3.832719103	9.592264931
600	2.819353348	7.821367638
671	3.03715002	7.483807579

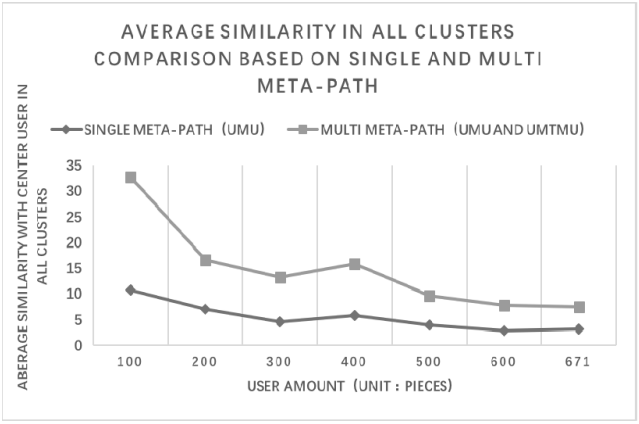


FIGURE 8. Average similarity comparison based on single and multi-meta-path.

E. IMPROVED FP-GROWTH ALGORITHM

After several experiments and analysis, it was found that when the support threshold minSup is 13% and the confidence threshold minConf is 70%, the recommendation result has the highest accuracy. Therefore, the above threshold is used during the experiment.

We divided users of MovieLens into groups and sorted the users within a group in the order of the timestamps. Then we divided the sorted data into two groups, 80% of them as experiment data to extract association rules. The rest 20% of the data were used to verify the accuracy of recommendation according to association rules.

Table 4 shows the comparison of the accuracy between three algorithms, i.e. the traditional single level FP-Growth, the improved single level FP-Growth, and the improved 2-level FP-Growth.

TABLE 4. Comparisons of traditional, one-level and two-level FP-growth algorithms.

User Amount	FP-Growth	One-level FP-Growth	Two-level FP-Growth
100	0.06305	0.10675	0.28346
200	0.08273	0.14594	0.30106
300	0.09871	0.15683	0.30779
400	0.16416	0.11848	0.30879
500	0.08647	0.10941	0.31257
600	0.11931	0.13607	0.31096
671	0.11798	0.12837	0.30842

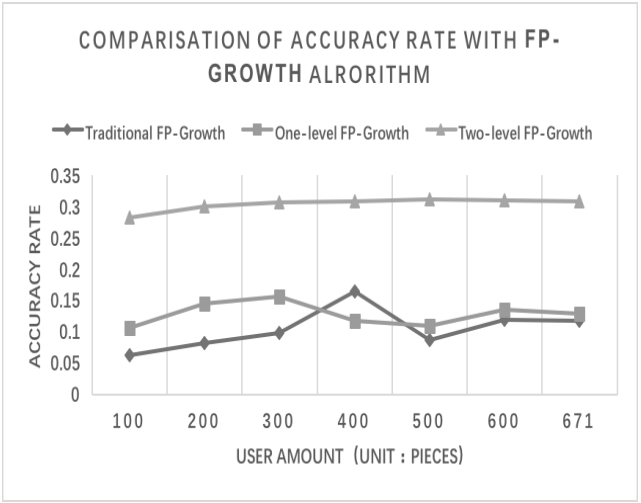


FIGURE 9. Comparison of accuracy rate with FP-Growth algorithm.

As shown in Table 4 and Fig. 9, by improving FP-Growth with additional level, the accuracy has been improved. Moreover, the 2-level FP-Growth algorithm achieves the best in terms of accuracy in the experiments.

The experimental results showed that the accuracy of recommendation of improved single level FP-Growth is all higher than the traditional FP-Growth except for the user number 400, while the improved 2-level FP-Growth performed better consistently than the other two algorithms.

F. CLOUD SERVICE RECOMMENDATION

In order to evaluate the performance of the proposed method, comparison experiments have been performed.

Firstly, we cluster users and use the improved FP-Growth algorithm to mine the association rules of user's viewing information on each cluster, and then recommend the service according to the improved FP-Growth algorithm, and finally calculate the recommendation accuracy rate.

Then, we calculate the accuracy of service recommendation using the 2-level FP-Growth algorithm after clustering users and the accuracy of service recommendation without using user clustering and traditional FP-Growth algorithm. The comparison results are shown in Table 5.

TABLE 5. Comparisons of Recommendation accuracy with Cluster and without cluster.

User Amount	With User Cluster	Without User Cluster
100	0.114682540	0.071428571
200	0.105952381	0.059523810
300	0.057640974	0.055952381
400	0.161169673	0.106269841
500	0.264671202	0.109523810
600	0.224351611	0.133968254
671	0.235657334	0.121086861

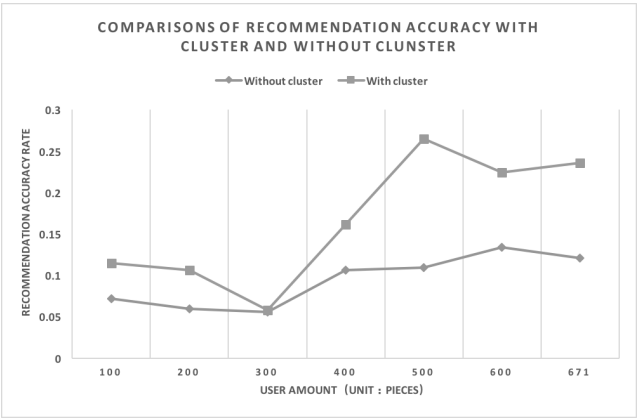


FIGURE 10. Comparison of accuracy rate with cluster and without cluster.

We simulated the recommendation accuracy in Table 5 and obtained the results of accuracy based on FP-Growth under cluster and non-cluster respectively, as shown in Figure 10. With different user numbers, the accuracy based on user clusters and improved FP-Growth are better than those without user clusters, which demonstrate the effectiveness of the recommendation algorithm proposed in this paper.

VI. CONCLUSION

This paper proposed a personalized cloud service recommendation algorithm based on association rules and user interest model aiming to improve the computation efficiency and recommendation accuracy of the cloud service.

Firstly, we improved the similarity measurement algorithm of AvgSim by combining user's rating into calculating of similarities between nodes, and obtained the user's interest in all services.

Then we improved the KHM clustering algorithm. First we calculate the similarity between users and then take its inverse as the square of the distance between users. The user interest value is taken as the attribute feature of the user, and the user is clustered with multiple meta-paths to obtain the user clustering result.

Finally, the traditional FP-Growth algorithm is improved. According to the theory of multi-level association rules, we establish a ''service-type'' two-layer FP-Tree. After that, the FP-Growth algorithm is used to mine the service sets used by all the users in each cluster obtained by clustering, and the service association rules can be obtained.

This paper combines the user interest, the improved KHM clustering algorithm and the improved FP-Growth association rules mining algorithm for cloud service recommendation. Extensive experiments performed on the MovieLens data set have verified the effectiveness and accuracy of the proposed cloud service recommendation algorithm.

REFERENCES

[1] Y. Yahfizham, F. Purwani, K. Rukun, and K. Chaniago, ''A review of cloud learning management system (CLMS) based on software as a service (SaaS),'' in *Proc. ICELTIC*, Banda Aceh, Indonesia, Oct. 2017, pp. 205–210.
[2] O. Ivanchenko, V. Kharchenko, B. Moroz, L. Kabak, and K. Smoktii, ''Semi-Markov availability model considering deliberate malicious impacts on an infrastructure-as-a-service cloud,'' in *Proc. TCSET*, Feb. 2018, pp. 570–573.
[3] T. Yamakami, ''Unconventional service engineering: Toward a new paradigm of service engineering for empowering senior citizens in city platform as a service,'' in *Proc. ICSESS*, Nov. 2017, pp. 11–14.
[4] Y. Hu, Q. Peng, X. Hu, and R. Yang, ''Web service recommendation based on time series forecasting and collaborative filtering,'' in *Proc. IEEE IC*, Jun. 2015, pp. 233–240.
[5] S. Parvatikar and B. Joshi, ''Online book recommendation system by using collaborative filtering and association mining,'' in *Proc. IEEE IC*, Dec. 2015, pp. 1–4.
[6] L. Guo, X. Zheng, C. Ding, D. Mu, and Z. Li, ''Cloud service recommendation: State of the art and research challenges,'' in *Proc. IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 761–764.
[7] A. Fariba and N. J. Navimipour, ''Cloud services recommendation: Reviewing the recent advances and suggesting the future research directions,'' *J. Netw. Comput. Appl.*, vol. 77, pp. 73–86, Jan. 2017.
[8] H. Jaiwei, ''Mining heterogeneous information networks: The next frontier,'' in *Proc. ACM-SIGKDD-ICKDDM*, 2012, pp. 2–3.
[9] Y. Chen, R. Liu, and W. Xu, ''Movie recommendation in heterogeneous information networks,'' in *Proc. IEEE-ITNEACC*, May 2016, pp. 637–640.
[10] L. Rui and L. Jian, ''Study of the clustering result based on user behavior feedback,'' in *Proc. IEEE-ICCCBDA*, Apr. 2017, pp. 371–375.
[11] C. Shi, X. Kong, Y. Huang, P. S. Yu, and B. Wu, ''HeteSim: A general framework for relevance measure in heterogeneous networks,'' *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2479–2492, Oct. 2014.
[12] X. Meng, C. Shi, Y. Li, L. Zhang, and B. Wu, ''Relevance measure in large-scale heterogeneous networks,'' in *Proc. 16th Asia–Pacific Web Conf.*, 2014, pp. 636–643.
[13] T. Mizumoto, K. El-Fakih, and K. Yasumoto, ''PathSim: A tool for finding minimal energy device operation sequence for reaching a target context in a Smart-Home,'' in *Proc. IEEE-ICUIC IEEE-ICATC*, Dec. 2013, pp. 64–71.
[14] C. Shi, Y. T. Li, J. W. Zhang, Y. Z. Sun, and P. S. Yu, ''A survey of heterogeneous information network analysis,'' *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 17–37, Aug. 2016.
[15] A. Ahmad and S. Hashmi, ''K-Harmonic means type clustering algorithm for mixed datasets,'' *Appl. Soft Comput.*, vol. 48, pp. 39–49, Nov. 2016.
[16] F. Zhang, Y. Xiao, and Y. Long, ''Research and improvement of parallelization of FP—Growth algorithm based on spark,'' in *Proc. IEEE-ICEIEC*, Jul. 2017, pp. 145–148.
[17] H. Chen, Q. Luo, Z. Chen, and Y. Chen, ''Distributed pruning optimization oriented FP-Growth method based on PSO algorithm,'' in *Proc. IEEE-ITNEACC*, Dec. 2017, pp. 1244–1248.
[18] X.-Z. Niu, J.-J. Niu, D. Z. Su, and K. She, ''FP-tree-based approach for frequent trajectory pattern mining,'' *J. Univ. Electron. Sci. Technol. China*, vol. 45, pp. 86–90, Jan. 2016.
[19] A. U. Ahmed, C. F. Ahmed, M. Samiullah, N. Adnan, and C. K.-S. Leung, ''Mining interesting patterns from uncertain databases,'' *Inf. Sci.*, vol. 354, pp. 60–85, Aug. 2016.
[20] A. H. H. Ghazala, M. A. Naeem, F. Mirza, and N. Jamil, ''Uncovering useful patterns in shopping cart data,'' in *Proc. IEEE-ICCSCWD*, Apr. 2017, pp. 349–354.
[21] W. Huo, X. Feng, and Z. Zhang, ''An efficient approach for incremental mining fuzzy frequent itemsets with FP-tree,'' *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 24, no. 3, pp. 367–386, 2016.
[22] *Non-Commercial, Personalized Movie Recommendations*. Accessed: Jun. 5, 2018. [Online]. Available: https://movielens.org/

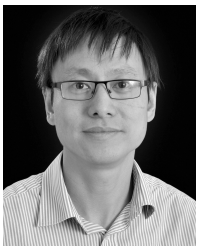
CHENGWEN ZHANG has been with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, since 2007. He is currently an Associate Professor. His research interests include big data, personalized recommendation, and cloud computing.

ZENGCHENG LI is currently a Graduate Student with the School of Computer Science, Beijing University of Posts and Telecommunications. He currently assists Prof. Zhang in the research about personalized recommendation.

TANG LI is a graduate student of School of Computer Science, Beijing University of Posts and Telecommunications, and assists Prof. Zhang in the research about personalized recommendation.

YUNAN HAN is currently an Associate Professor with the School of Information Science and Technology, Beijing University of Chemical Technology. His research interest includes personalized recommendation and service computing.

CUICUI WEI was a Graduate Student with the School of Computer Science, Beijing University of Posts and Telecommunications. He assisted Prof. Zhang in the research about personalized recommendation.



YONGQIANG CHENG is currently a Lecturer with the School of Engineering and Computer Science, University of Hull, U.K. His research interest includes digital healthcare technologies, embedded systems, control theory and applications, AI, and data mining.

YONGHONG PENG is currently a Professor of data science and the Leader of the Data Science Research, University of Sunderland, U.K. His research areas include data science, machine learning, data mining, and artificial intelligence. He is the Chair for the Big Data Task Force, and a member of the Data Mining and Big Data Analytics Technical Committee of the IEEE Computational Intelligence Society. He is also a founding member of the Technical Committee on Big Data of the IEEE Communications and an Advisory Board Member of the IEEE Special Interest Group on Big Data for Cyber Security and Privacy. He is an Associate Editor of the IEEE TRANSACTION ON BIG DATA and an Academic Editor of PeerJ and *PeerJ Computer Science*.