Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 4397061, 12 pages
http://dx.doi.org/10.1155/2016/4397061

Research Article

Time-Aware IoE Service Recommendation on Sparse Data

Lianyong Qi,[1,2] Xiaolong Xu,[3] Wanchun Dou,[2] Jiguo Yu,[1] Zhili Zhou,[3] and Xuyun Zhang[4]

[1] *School of Information Science and Engineering, Qufu Normal University, Qufu, China*
[2] *Department of Computer Science and Technology, State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China*
[3] *School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China*
[4] *Department of Electrical and Computer Engineering, University of Auckland, Auckland, New Zealand*

Correspondence should be addressed to Lianyong Qi; lianyongqi@gmail.com

Received 15 September 2016; Accepted 10 November 2016

Academic Editor: Beniamino Di Martino

With the advent of "Internet of Everything" (IoE) age, an excessive number of IoE services are emerging on the web, which places a heavy burden on the service selection decision of target users. In this situation, various recommendation techniques are introduced to alleviate the burden, for example, Collaborative Filtering- (CF-) based recommendation. Generally, CF-based recommendation approaches utilize similar friends or similar services to achieve the recommendation goal. However, due to the sparsity of user feedback, it is possible that a target user has no similar friends and similar services; in this situation, traditional CF-based approaches fail to produce a satisfying recommendation result. Besides, recommendation accuracy would be decreased if time factor is overlooked, as IoE service quality often varies with time. In view of these challenges, a time-aware service recommendation approach named Ser_Rec$_{\text{time}}$ is proposed in this paper. Concretely, we first calculate the time-aware user similarity; afterwards, indirect friends of the target user are inferred by Social Balance Theory (e.g., "enemy's enemy is a friend" rule); finally, the services preferred by indirect friends of the target user are recommended to the target user. At last, through a set of experiments deployed on dataset WS-DREAM, we validate the feasibility of our proposal.

1. Introduction

With the wide adoption of various smart devices and connection technologies, human society is gradually transforming into an "Internet of Everything" (IoE) one [1–5]. In the age of IoE, the links among users, devices, or other things could be built easily, which significantly improves people's quality of life and also brings many challenging open problems that need to be addressed [6–11].

With the advent of IoE age, an excessive number of IoE services with different functionalities or qualities are emerging on the web, which places a heavy burden on the service selection decision of target users [12–14]. In this situation, various recommendation techniques are put forward to help alleviate the service selection burden, for example, Collaborative Filtering- (CF-) based recommendation [15–19]. Generally, CF-based recommendation approaches (including user-based CF, item-based CF, and hybrid CF) utilize the similar friends or similar services of target users to achieve the recommendation goal. However, in certain situations, the available user-service rating data generated from historical service invocations is really sparse [20]. Therefore, it is probable that a target user cannot find his/her similar friends and similar services of target services (here, target services mean the services preferred by target users). In this situation, traditional CF-based recommendation approaches cannot return a satisfying recommendation result, which brings a great challenge for the robustness of service recommendation approaches. Besides, the quality of an IoE service often varies with time, due to the unstable network environment [21]. For example, the response time of a ticket-order service often becomes larger when Christmas day is approaching. Therefore, recommendation accuracy would be decreased if the time factor is not taken into consideration.

In view of the above two challenges, a novel time-aware service recommendation approach, that is, Ser_Rec$_{\text{time}}$, is put forward in this paper, to make robust and accurate service recommendation for target users when the historical user-service rating data is sparse. Concretely, in Ser_Rec$_{\text{time}}$, we first calculate the time-aware user similarity; afterwards, we look for the enemies (antonym of "friend") of the target user and further determine the "indirect friends" of target user by Social Balance Theory [22] (e.g., "enemy's enemy is a friend" rule); finally, the services preferred by indirect friends of target user are recommended to the target user.

The contributions of our paper are threefold.

(1) Time factor is considered in user similarity calculation to adapt to the dynamic quality variation of IoE services, which makes the subsequent recommendation result more objective and accurate.

(2) Social Balance Theory is introduced for service recommendation on sparse data so that the recommendation robustness is improved.

(3) A wide range of experiments are designed and deployed on a real web service quality set WS-DREAM [23], so as to further validate the feasibility of our proposal.

The remainder of our paper is organized as follows. In Section 2, we first formalize the CF-based service recommendation problem and afterwards demonstrate the motivation of our paper. In Section 3, a novel time-aware service recommendation approach named Ser_Rec$_{\text{time}}$ is brought forth. A set of experiments are deployed in Section 4 and evaluations are presented in Section 5. Finally, in Section 6, we summarize the paper and point out our future research directions.

2. Formalization and Motivation

In this section, we first formalize the CF-based service recommendation problem. And afterwards, an example is presented to demonstrate the motivation of our paper intuitively.

2.1. Formal Specification. Generally, the CF-based service recommendation problem could be specified with a four-tuple CF-Ser-Rec $(U, \text{WS}, \rightarrow, \text{user}_{\text{target}})$, where

(1) $U = \{\text{user}_1, \ldots, \text{user}_m\}$ denotes the user set in user-service invocation network and m is the number of users;

(2) $\text{WS} = \{\text{ws}_1, \ldots, \text{ws}_n\}$ denotes the web service set in user-service invocation network and n is the number of web services;

(3) $\rightarrow = \{(\text{user}_i \xrightarrow[T_{i-j}]{R_{i-j}} \text{ws}_j) \mid 1 \le i \le m,\ 1 \le j \le n\}$ denotes the historical invocation record set, that is, $\text{user}_i \xrightarrow[T_{i-j}]{R_{i-j}} \text{ws}_j$, which means that user_i invoked ws_j in the past and the rating of ws_j by user_i is R_{i-j}. Here, for simplicity, the well-known $\{1^*, 2^*, 3^*, 4^*, 5^*\}$ rating system is adopted to depict R_{i-j}. Also, each invocation record owns a timestamp T_{i-j} that indicates the service invocation time (here, we assume the timestamp format is "yyyy.mm.dd");

(4) $\text{user}_{\text{target}}$ denotes the target user that requires service recommendation and $\text{user}_{\text{target}} \in U$ holds.

With the above formalization, CF-based service recommendation problem could be specified as follows: according to the historical invocation record set "→" between users (in U) and web services (in WS), find out the services that were never invoked but may be preferred by target user $\text{user}_{\text{target}}$ and recommend them to $\text{user}_{\text{target}}$.

2.2. Motivation. In this subsection, the paper motivation is clarified more intuitively with the example shown in Figure 1. In Figure 1, there are three users {Tom, Alice, Bob} (Tom is the target user) in set U and six web services $\{\text{ws}_1, \text{ws}_2, \text{ws}_3, \text{ws}_4, \text{ws}_5, \text{ws}_6\}$ in set WS; historical user-service invocation record set → (including rating data and timestamp) is also presented in Figure 1.

Then according to the Adjusted Cosine Similarity [24] (i.e., $\text{ACS} \in [-1, 1]$; here, the reason that we utilize ACS for similarity calculation is that user-service rating data is often discrete and the rating scales of different users often vary), we can calculate the similarity between target user Tom and other two users (i.e., Alice and Bob). Concretely, $\text{Sim}(\text{Tom}, \text{Alice}) = -0.2747$ and $\text{Sim}(\text{Tom}, \text{Bob}) = \text{Null}$ (as no services were rated by both Tom and Bob). Therefore, we can conclude that target user Tom has no similar friends, so according to the traditional user-based CF recommendation approaches, no qualified services are recommended to Tom.

Likewise, the similarities between target services (i.e., ws_1 and ws_2) and other services (i.e., ws_3, ws_4, ws_5, and ws_6) could also be obtained. Concretely, $\text{Sim}(\text{ws}_1, \text{ws}_3) = \text{Sim}(\text{ws}_1, \text{ws}_4) = \text{Sim}(\text{ws}_2, \text{ws}_3) = \text{Sim}(\text{ws}_2, \text{ws}_4) = -1$ and $\text{Sim}(\text{ws}_1, \text{ws}_5) = \text{Sim}(\text{ws}_1, \text{ws}_6) = \text{Sim}(\text{ws}_2, \text{ws}_5) = \text{Sim}(\text{ws}_2, \text{ws}_6) = \text{Null}$ (as no users invoked any pair of services above simultaneously). Therefore, a conclusion could be drawn that target user Tom's preferred services (i.e., ws_1 and ws_2) have no similar services, so according to the traditional item-based CF recommendation approaches, no qualified services are recommended to Tom.

Therefore, in this situation, traditional CF-based service recommendation approaches (e.g., user-based CF, item-based CF, or hybrid CF) cannot make accurate service recommendation for target user. Besides, as Figure 1 shows, the timestamps of various invocation records are often different, so the service recommendation accuracy and fairness would be decreased if we overlook the time factor in recommendation process. In view of the above two challenges, a novel service recommendation approach named Ser_Rec$_{\text{time}}$ is put forward in Section 3. Ser_Rec$_{\text{time}}$ cannot only find out the indirect friends of a target user so as to improve the recommendation robustness in sparse-data environment, but also consider time factor in recommendation so as to ensure the fairness and accuracy of recommendation results.

3. Time-Aware Service Recommendation Approach: Ser_Rec$_{\text{time}}$

In this section, a novel time-aware recommendation approach, that is, Ser_Rec$_{\text{time}}$, is introduced to deal with the

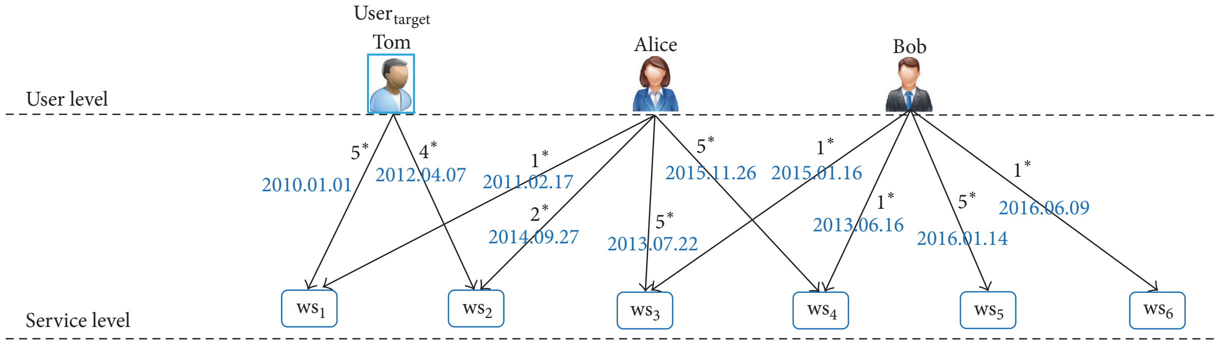


FIGURE 1: Time-aware service recommendation when target user has no similar friends and similar services.

service recommendation problem on sparse data. The main idea of our proposal is as follows: first, we calculate the time-aware similarity between target user and other users based on the historical user-service rating records; second, according to the derived user similarity and Social Balance Theory, we infer the indirect friends of target user; third, the services preferred by indirect friends of target user are recommended to the target user. Concretely, the three steps of time-aware service recommendation approach (Ser_Rec$_{\text{time}}$) are as follows.

Step 1 (time-aware user similarity calculation). Calculate the similarity $\text{Sim}(\text{user}_{\text{target}}, \text{user}_i)$ between $\text{user}_{\text{target}}$ and other users $\text{user}_i \in U$. If $\text{Sim}(\text{user}_{\text{target}}, \text{user}_i) \le P$ (P is similarity threshold), then user_i is considered as an enemy of $\text{user}_{\text{target}}$.

Step 2 (determining indirect friends of target user). According to the enemies (derived in Step 1) of target user and Social Balance Theory, determine target user's indirect friends.

Step 3 (service recommendation). Select the services preferred by the indirect friends (derived in Step 2) of target user and recommend them to the target user.

The three steps are explained as follows:

Step 1 (time-aware user similarity calculation). In this step, we calculate the similarity between target user $\text{user}_{\text{target}}$ and other users user_i ($\text{user}_i \in U$), that is, $\text{Sim}(\text{user}_{\text{target}}, \text{user}_i)$. As user-service rating data is often discrete and rating scores of a service by different users are often varied, Adjusted Cosine Similarity is recruited here for user similarity calculation. Concretely, $\text{Sim}(\text{user}_{\text{target}}, \text{user}_i)$ could be obtained based on

$$\text{Sim}\left(\text{user}_{\text{target}}, \text{user}_i\right) = \frac{\sum_{\text{ws}_j \in I} \left(R_{\text{target}-j} - \overline{R_{\text{target}}}\right) * \left(R_{i-j} - \overline{R_i}\right)}{\sqrt{\sum_{\text{ws}_j \in I_{\text{target}}} \left(R_{\text{target}-j} - \overline{R_{\text{target}}}\right)^2} * \sqrt{\sum_{\text{ws}_j \in I_i} \left(R_{i-j} - \overline{R_i}\right)^2}}. \quad (1)$$

Here, set I denotes the common services that were rated by both $\text{user}_{\text{target}}$ and user_i; sets I_{target} and I_i denote the service set rated by $\text{user}_{\text{target}}$ and user_i, respectively; $R_{\text{target}-j}$ and R_{i-j} denote ratings of service ws_j by $\text{user}_{\text{target}}$ and user_i, respectively; while $\overline{R_{\text{target}}}$ and $\overline{R_i}$ represent $\text{user}_{\text{target}}$'s and user_i's average rating values. Then according to (1), we can calculate the similarity between target user and any other user.

Next, we improve the user similarity formula in (1) by introducing four kinds of weight coefficients.

(i) *Weight for Service Intersection Size*. As formula (1) indicates, I_{target} and I_i denote the service set rated by $\text{user}_{\text{target}}$ and user_i, respectively. Generally, for $\text{user}_{\text{target}}$ and user_i, the larger their service intersection (i.e., set I in formula (1)) is, the more convincing their similarity is. So in order to depict this correlation, weight $W_{\text{ser-intersection}}$ (in formula (2)) is assigned to user similarity $\text{Sim}(\text{user}_{\text{target}}, \text{user}_i)$:

$$W_{\text{ser-intersection}} = \frac{1}{2} * \left(\frac{I_{\text{target}} \cap I_i}{I_{\text{target}}} + \frac{I_{\text{target}} \cap I_i}{I_i}\right). \quad (2)$$

(ii) *Weight for Invocation Time*. Due to the dynamic and unstable network environment, the IoE service quality is often varied with time; therefore, two neighboring service invocations with close invocation time often contribute more to the user similarity. In view of this observation, similar to work [4], weight W_{time} in formula (3) is assigned to user similarity $\text{Sim}(\text{user}_{\text{target}}, \text{user}_i)$. In (3), α ($\alpha \ge 0$) is a parameter, while $T_{\text{target}-j}$ and T_{i-j} represent the invocation time of service ws_j by $\text{user}_{\text{target}}$ and user_i, respectively

$$W_{\text{time}} = e^{-\alpha |T_{\text{target}-j} - T_{i-j}|}. \quad (3)$$

(iii) *Weight for Invocation Load*. Service invocation time cannot reflect all the time-related information in similarity calculation. For example, user_1 and user_2 invoked the same ticket-booking service on 22-12-2015 and 24-12-2015, respectively. Although their service invocation time is close, the user experienced service quality may vary significantly as heavy service load is inevitable when Christmas day is approaching. So in order to depict the effect of service load on user similarity, weight W_{load} in formula (4) is assigned to user similarity $\text{Sim}(\text{user}_{\text{target}}, \text{user}_i)$. In (4), $\text{Load}_{\text{target}-j}$ and Load_{i-j} denote the service loads when $\text{user}_{\text{target}}$ and user_i invoked service ws_j, respectively.

$$W_{\text{load}} = 1 - \frac{\left|\text{Load}_{\text{target}-j} - \text{Load}_{i-j}\right|}{\max\left\{\text{Load}_{\text{target}-j}, \text{Load}_{i-j}\right\}}. \quad (4)$$

(iv) *Weight for Service Version*. In the life cycle of a web service, service provider may publish a series of service versions with updated service qualities. Therefore, if two users invoked an identical web service that belongs to different versions, it would not make much sense to compare their experienced service qualities. In view of this observation, weight W_{version} is suggested as follows:

$$W_{\text{version}} = \begin{cases} 1 & \text{user}_{\text{target}}, \text{user}_i \text{ invoked same service of same version} \\ 0 & \text{else.} \end{cases} \quad (5)$$

With the above analyses, we can update the user similarity $\text{Sim}(\text{user}_{\text{target}}, \text{user}_i)$ in (1) to be time-aware user similarity $\text{Sim}_{\text{time}}(\text{user}_{\text{target}}, \text{user}_i)$ in (6). Then based on (6), we can calculate the time-aware similarity between target user and any other user:

$$\text{Sim}_{\text{time}}\left(\text{user}_{\text{target}}, \text{user}_i\right) = W_{\text{ser-intersection}} * \frac{\sum_{\text{ws}_j \in I} W_{\text{time}} * W_{\text{load}} * W_{\text{version}} * \left(R_{\text{target}-j} - \overline{R_{\text{target}}}\right) * \left(R_{i-j} - \overline{R_i}\right)}{\sqrt{\sum_{\text{ws}_j \in I_{\text{target}}} \left(R_{\text{target}-j} - \overline{R_{\text{target}}}\right)^2} * \sqrt{\sum_{\text{ws}_j \in I_i} \left(R_{i-j} - \overline{R_i}\right)^2}}. \quad (6)$$

Furthermore, according to the derived user similarity, the enemy set of target user, that is, Enemy_set $(\text{user}_{\text{target}})$, could be obtained based on (7). Here, parameter P $(-1 \le P \le -0.5)$ is a predefined user similarity threshold for enemy relationship:

$$\text{user}_i \begin{cases} \in \text{Enemy\_set}\left(\text{user}_{\text{target}}\right) & \text{if } \text{Sim}_{\text{time}}\left(\text{user}_{\text{target}}, \text{user}_i\right) \le P \\ \notin \text{Enemy\_set}\left(\text{user}_{\text{target}}\right) & \text{else.} \end{cases} \quad (7)$$

Step 2 (determining indirect friends of target user). In Step 1, we have obtained the enemy set of target user, that is, Enemy_set $(\text{user}_{\text{target}})$. Next, in this step, we will introduce how to get the indirect friends of target user, that is, Indirect_friend_set$(\text{user}_{\text{target}})$, based on the obtained Enemy_set $(\text{user}_{\text{target}})$ and Social Balance Theory. First of all, we introduce Social Balance Theory briefly.

Social Balance Theory [22] was first put forward by psychologist F. Heider in 1958. The theory investigates the stable social relationships among involved three parties (i.e., P, O, and X in Figure 2). Concretely, according to Figure 2, we introduce the four stable social relationships, respectively, in a more intuitive manner.

(a) *Friend's Friend Is a Friend*. If X is a friend of O and O is a friend of P, then we can infer that X is probably an indirect friend of P (see Figure 2(a)).

(b) *Enemy's Enemy Is a Friend*. If X is an enemy of O and O is an enemy of P, then we can infer that X is probably an indirect friend of P (see Figure 2(b)).

(c) *Friend's Enemy Is an Enemy*. If X is an enemy of O while O is a friend of P, then we can infer that X is probably an indirect enemy of P (see Figure 2(c)).

(d) *Enemy's Friend Is an Enemy*. If X is a friend of O while O is an enemy of P, then we can infer that X is probably an indirect enemy of P (see Figure 2(d)).

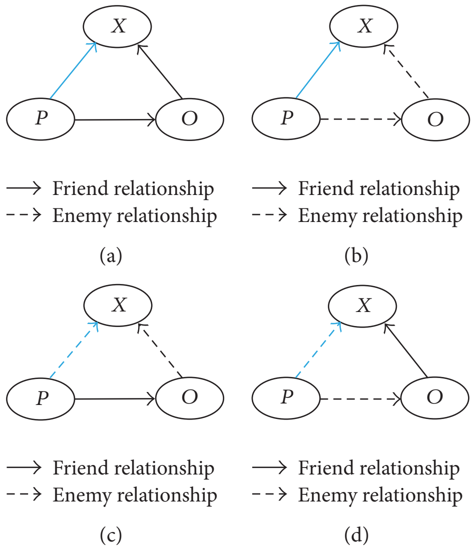
Next, with the above four rules in Social Balance Theory, we introduce how to obtain the indirect friend set of target user, that is, Indirect_friend_set$(\text{user}_{\text{target}})$, based on the derived Enemy_set$(\text{user}_{\text{target}})$ in Step 1.

Concretely, for each $\text{user}_i \in$ Enemy_set $(\text{user}_{\text{target}})$, we first calculate his/her similarities with other users based on (6); afterwards, we determine user_i's enemies user_k (i.e., $\text{user}_k \in$ Enemy_set (user_i)) based on (7) and user_i's friends user_g (i.e., $\text{user}_g \in$ Friend_set (user_i)) based on (8). In (8), parameter $-P$ denotes the user similarity threshold for friend relationship. To ease the understanding of readers, the relationships among $\text{user}_{\text{target}}$, user_i, user_k, and user_g are presented in Figure 3.

$$\text{user}_g \begin{cases} \in \text{Friend\_set}\left(\text{user}_i\right) & \text{if } \text{Sim}_{\text{time}}\left(\text{user}_i, \text{user}_g\right) \ge -P \\ \notin \text{Friend\_set}\left(\text{user}_i\right) & \text{else.} \end{cases} \quad (8)$$

Then according to "enemy's enemy is a friend" rule in Figure 2(b), we can infer that user_k is probably an indirect friend of $\text{user}_{\text{target}}$, and the probability could be calculated based on (9). If $\text{Probability}_{\text{friend}}(\text{user}_{\text{target}}, \text{user}_k) \ge -P$, then user_k is regarded as a qualified indirect friend of target user and put into set Indirect_friend_set$(\text{user}_{\text{target}})$ (see Figure 3(b)). Likewise, according to "enemy's friend is an enemy" rule in Figure 2(d), it can be inferred that user_g is probably an indirect enemy of $\text{user}_{\text{target}}$, and the probability can be obtained based on (10). If $\text{Probability}_{\text{enemy}}(\text{user}_{\text{target}}, \text{user}_g) \ge -P$, then user_g is regarded as a qualified indirect enemy of target user and put into set Enemy_set $(\text{user}_{\text{target}})$ (see Figure 3(d))

$$\begin{aligned} &\text{Probability}_{\text{friend}}\left(\text{user}_{\text{target}}, \text{user}_k\right) \\ &\quad = \text{Sim}_{\text{time}}\left(\text{user}_{\text{target}}, \text{user}_i\right) \\ &\quad * \text{Sim}_{\text{time}}\left(\text{user}_i, \text{user}_k\right), \end{aligned} \quad (9)$$



(a) (b) (c) (d)

Figure 2: Four stable relationships among P, X, and O according to Social Balance Theory (the blue arrows between P and X denote the inferred relationships).

$$\begin{aligned}&\text{Probability}_{\text{enemy}}\left(\text{user}_{\text{target}},\text{user}_{g}\right)\\&\quad=\left|\text{Sim}_{\text{time}}\left(\text{user}_{\text{target}},\text{user}_{i}\right)\right.\\&\quad\left.*\,\text{Sim}_{\text{time}}\left(\text{user}_{i},\text{user}_{g}\right)\right|.\end{aligned}\tag{10}$$

Next, for each user$_x$ ∈ Indirect_friend_set(user$_{\text{target}}$), we calculate his/her similarity with other users (∈ (U-user$_{\text{target}}$-Indirect_friend_set(user$_{\text{target}}$)-Enemy_set(user$_{\text{target}}$))) based on (6) and further determine user$_x$'s enemies user$_y$ based on (7) and user$_x$'s friends user$_z$ based on (8). Then according to "friend's enemy is an enemy" rule in Figure 2(c), user$_y$ is regarded as an indirect enemy of target user and put into Enemy_set (user$_{\text{target}}$) (see Figure 3(c)) if the probability in (11) is larger than $-P$. Likewise, according to "friend's friend is a friend" rule in Figure 2(a), user$_z$ is considered as an indirect friend of target user and put into Indirect_friend_set (user$_{\text{target}}$) (see Figure 3(a)) if the probability in (12) is larger than $-P$.

$$\begin{aligned}&\text{Probability}_{\text{enemy}}\left(\text{user}_{\text{target}},\text{user}_{y}\right)\\&\quad=\left|\text{Probability}_{\text{friend}}\left(\text{user}_{\text{target}},\text{user}_{x}\right)\right.\\&\quad\left.*\,\text{Sim}_{\text{time}}\left(\text{user}_{x},\text{user}_{y}\right)\right|,\end{aligned}\tag{11}$$

$$\begin{aligned}&\text{Probability}_{\text{friend}}\left(\text{user}_{\text{target}},\text{user}_{z}\right)\\&\quad=\text{Probability}_{\text{friend}}\left(\text{user}_{\text{target}},\text{user}_{x}\right)\\&\quad*\,\text{Sim}_{\text{time}}\left(\text{user}_{x},\text{user}_{z}\right).\end{aligned}\tag{12}$$

Repeat (a)–(d) process in Figure 3 until set Indirect_friend_set(user$_{\text{target}}$) stays stable. Then we obtain the indirect friends of target user, that is, Indirect_friend_set(user$_{\text{target}}$).

Step 3 (service recommendation). In Step 2, we have obtained target user's indirect friend set Indirect_friend_set(user$_{\text{target}}$). Next, in this step, we select the services preferred (i.e., with 4^* or 5^* rating) by indirect friends of target user and recommend them to the target user. More formally, for each element user$_x$ ∈ Indirect_friend_set(user$_{\text{target}}$), if his/her rating over web service ws$_j$ ($1 \le j \le n$), that is, $R_{x-j} \in \{4^*, 5^*\}$, holds, then ws$_j$ is put into the recommended service set, that is, Rec_Ser_Set. Finally, all the web services in set Rec_Ser_Set are recommended to user$_{\text{target}}$.

With above Step 1–Step 3 of our proposed Ser_Rec$_{\text{time}}$ approach, a set of IoE services (in set Rec_Ser_Set) are recommended to the target user. More formally, the pseudocode of Ser_Rec$_{\text{time}}$(U, WS, →, user$_{\text{target}}$) is presented in Algorithm 1 (please note that algorithm Ser_Rec$_{\text{time}}$(U, WS, →, user$_{\text{target}}$) is abbreviated as Ser_Rec$_{\text{time}}$ in the whole paper).

4. Experiment

In this section, a set of experiments are designed and tested to validate the feasibility of our proposed Ser_Rec$_{\text{time}}$ approach, in terms of recommendation accuracy, recall, and efficiency.

4.1. Experiment Dataset and Deployment. The experiment is based on a real service quality dataset WS-DREAM [23]. WS-DREAM consists of 4532 IoE services on the web, and 142 distributed users from Planet-Lab are employed for evaluating the real quality (e.g., *response time* and *throughput*) of services in 64 time intervals (time interval = 15 minutes).

Our paper focuses on the user-service rating-based recommendation, while available user-service rating data is really rare on the web; therefore, in the experiment, we need to transform the service quality data in WS-DREAM into corresponding user-service rating data (essentially, objective service quality data and subjective user-service rating data both reflect the service running quality; therefore, we argue that the transformation from former data to latter data makes sense for the service recommendation simulation here). Concretely, the transformation process is as follows: we determine the minimal and maximal quality values (denoted by min and max, resp.) of a service observed by an identical user, and afterwards we divide the range [min, max] into five subranges in an arithmetic progression manner, each corresponding to a rating value. For example, if the minimal and maximal throughput values of a service ws$_j$ observed by user$_i$ are 10 kbps and 60 kbps, respectively, then we can get five subranges after division, that is, [10, 20) kbps, [20, 30) kbps, [30, 40) kbps, [40, 50) kbps, and [50, 60] kbps. Furthermore, if the throughput value of ws$_j$ observed by user$_i$ is 44 kbps in WS-DREAM, then the transformed user-service rating, that is, $R_{i-j} = 4^*$, holds. For each service invoked by a user, we randomly select a time interval from all the 64 intervals and

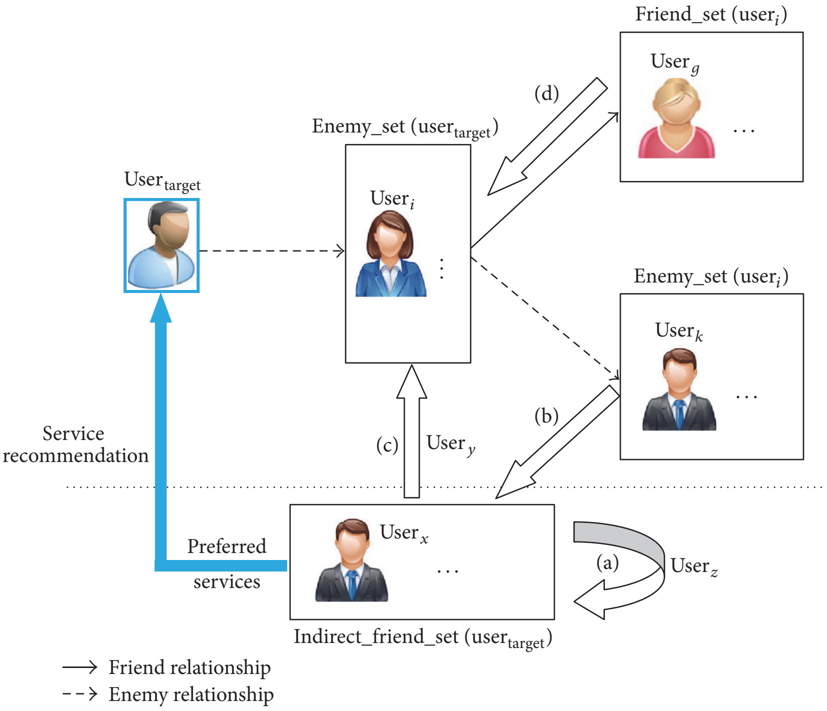


Figure 3: Relationships among user$_{target}$, user$_i$, user$_k$, user$_g$, user$_x$, user$_y$, and user$_z$.

transform the service quality corresponding to the selected time interval into a user-service rating with a timestamp (i.e., the selected time interval). In the experiment, only a quality dimension, that is, *throughput*, is considered.

Besides, our paper only discusses the time-aware service recommendation in sparse-data environment where target user has no similar friends and similar services. Therefore, we select appropriate data from WS-DREAM to simulate the above sparse recommendation situations. Furthermore, for each recruited target user, his/her 80% ratings are randomly selected for training while the remaining 20% ratings for testing.

Also, we compare our proposal with related works, for example, WSRec [25], MCCP [26], and SBT-SR [27], in terms of recommendation accuracy and recall. Concretely, accuracy and recall are measured by Mean Absolute Error (i.e., MAE; the smaller the better) in (13) and Recall (the larger the better) in (14). In (13), Rec_Ser_Set denotes the recommended service set for target user; $|X|$ denotes the size of set X; $R_{\text{target}-x}$ and $R^{\#}_{\text{target}-x}$ represent target user's real and predicted ratings over service ws$_x$, respectively. While in (14), symbols Preferred_Ser_Set and Rec_Ser_Set denote the service set really preferred (with 4* or 5* rating) by target user and the service set recommended to target user, respectively.

$$\text{MAE} = \sum_{\text{ws}_x \in \text{Rec\_Ser\_Set}} \frac{\left| R_{\text{target}-x} - R^{\#}_{\text{target}-x} \right|}{\left|\text{Rec\_Ser\_Set}\right|}, \qquad (13)$$

$$\text{Recall} = \frac{\left|\text{Preferred\_Ser\_Set} \cap \text{Rec\_Ser\_Set}\right|}{\left|\text{Preferred\_Ser\_Set}\right|}. \qquad (14)$$

The experiments were conducted on a Dell laptop with 2.80 GHz processors and 2.0 GB RAM. The software configuration is Windows XP + JAVA 1.5. Each experiment was carried out 10 times and the average results were adopted finally.

4.2. Experiment Results and Analyses. Concretely, four profiles are designed, tested, and compared. Here, m and n denote the number of users and number of services in the historical user-service invocation network.

(Profile 1) Recommendation Accuracy Comparison. In this profile, we compare the recommendation accuracy of Ser_Rec$_{\text{time}}$ with the other three approaches, that is, WSRec, MCCP, and SBT-SR. Concretely, parameters $w_u = w_i = 0.5$ hold in WSRec; $\alpha = 0.8$ holds in MCCP; user similarity threshold $P = -0.5$ holds in both SBT-SR and Ser_Rec$_{\text{time}}$. Besides, in Ser_Rec$_{\text{time}}$, $\alpha = 0.09$ holds in (3) and $W_{\text{load}} = W_{\text{version}} = 1$ holds (in WS-DREAM, a service was invoked by a user 64 times within successive 16 hours; therefore, we assume that the service load and version do not vary much). Finally, m is varied from 20 to 100 in all the four approaches.

The experiment result is presented in Figure 4. As Figure 4 shows, the recommendation accuracy of WSRec is low, as it only adopts the average ratings of target user and target services without considering the valuable user-service relationships hidden in historical service invocation records. The accuracy of MCCP is improved by considering the user-service relationships; however, it is not very suitable for service recommendation when target user has no similar friends and similar services, as MCCP is essentially a kind

```
Inputs:
(1) $U = \{\text{user}_1, \ldots, \text{user}_m\}$: a set of users in user-service invocation & rating network;
(2) $\text{WS} = \{\text{ws}_1, \ldots, \text{ws}_n\}$: a set of web services in user-service invocation & rating network;
(3) $\rightarrow = \{(\text{user}_i \xrightarrow[T_{i-j}]{R_{i-j}} \text{ws}_j) \mid 1 \le i \le m,\ 1 \le j \le n\}$: a set of historical user-service rating records;
(4) $\text{user}_{\text{target}}$: target user who requires service recommendation.
Output: Rec_Ser_Set: service set recommended to $\text{user}_{\text{target}}$
(1) Set user similarity threshold $P$ $(-1 \le P \le -0.5)$
(2) for each $\text{user}_i \in U$ do    // Step 1
(3)   calculate $\text{Sim}_{\text{time}}(\text{user}_{\text{target}}, \text{user}_i)$ based on (1)–(6)
(4)   if $\text{Sim}_{\text{time}}(\text{user}_{\text{target}}, \text{user}_i) \le P$
(5)     then put $\text{user}_i$ into set Enemy_set($\text{user}_{\text{target}}$)
(6)   end if
(7) end for
(8) for each $\text{user}_i \in$ Enemy_set($\text{user}_{\text{target}}$) do    // Step 2
(9)   determine $\text{user}_i$'s friend $\text{user}_g$ based on (6) and (8)
(10)   if probability in (10) is larger than $-P$
(11)   then put $\text{user}_g$ into Enemy_set($\text{user}_{\text{target}}$)
(12)   end if
(13)   determine $\text{user}_i$'s enemy $\text{user}_k$ based on (6) and (7)
(14)   if probability in (9) is larger than $-P$
(15)   then put $\text{user}_k$ into Indirect_friend_set($\text{user}_{\text{target}}$)
(16)   end if
(17) end for
(18) for each $\text{user}_x \in$ Indirect_friend_set($\text{user}_{\text{target}}$) do
(19)   determine $\text{user}_x$'s enemy $\text{user}_y$ based on (6) and (7)
(20)   if probability in (11) is larger than $-P$
(21)   then put $\text{user}_y$ into Enemy_set($\text{user}_{\text{target}}$)
(22)   end if
(23)   determine $\text{user}_x$'s friend $\text{user}_z$ based on (6) and (8)
(24)   if probability in (12) is larger than $-P$
(25)   then put $\text{user}_z$ into Indirect_friend_set($\text{user}_{\text{target}}$)
(26)   end if
(27) end for
(28) repeat Line (8)–Line (27) until Indirect_friend_set($\text{user}_{\text{target}}$) stays stable
(29) Rec_Ser_Set $= \emptyset$    // Step 3
(30) for each $\text{user}_x \in$ Indirect_friend_set($\text{user}_{\text{target}}$) do
(31)   for each $\text{ws}_j \in \text{WS}$ do
(32)     if $R_{x-j} \in \{4^*, 5^*\}$
(33)     then put $\text{ws}_j$ into set Rec_Ser_Set
(34)     end if
(35)   end for
(36) end for
(37) return Rec_Ser_Set
```

Algorithm 1: Ser_Rec$_{\text{time}}$ (U, WS, →, user$_{\text{target}}$).

of similar user-based service recommendation approach. The recommendation accuracy of SBT-SR is high as "enemy's enemy is a friend" rule of Social Balance Theory is recruited to find the "indirect friends" of target user. Furthermore, our proposed Ser_Rec$_{\text{time}}$ outperforms SBT-SR as Ser_Rec$_{\text{time}}$ not only considers Social Balance Theory, but also takes time factor into consideration for looking for the really similar "indirect friends" of target user. Besides, as shown in Figure 4, recommendation accuracy values of MCCP, SBT-SR, and Ser_Rec$_{\text{time}}$ all increase with the growth of m approximately; this is because more valuable user-service relationship information would be discovered and recruited for service recommendation when there are many users as well as their invocation records.

(Profile 2) Recommendation Recall Comparison. In this profile, we compare the recommendation recall values of four approaches. The parameter settings are the same as those in profile 1. Concrete experiment results are shown in Figure 5.

As Figure 5 shows, the recall of WSRec is low as the "average" idea adopted in WSRec often leads to a low recommendation hit rate. The recommendation recall values of the remaining three approaches all increase with the growth of m; this is because when there are many available historical users,

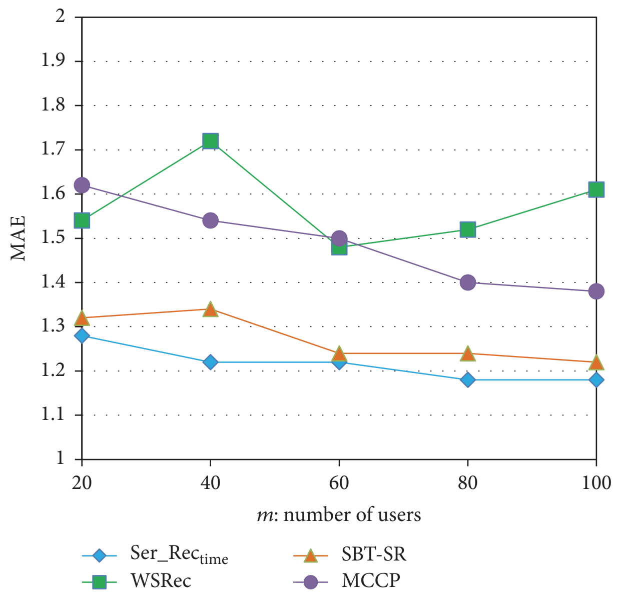


Figure 4: Recommendation accuracy comparison with respect to m.

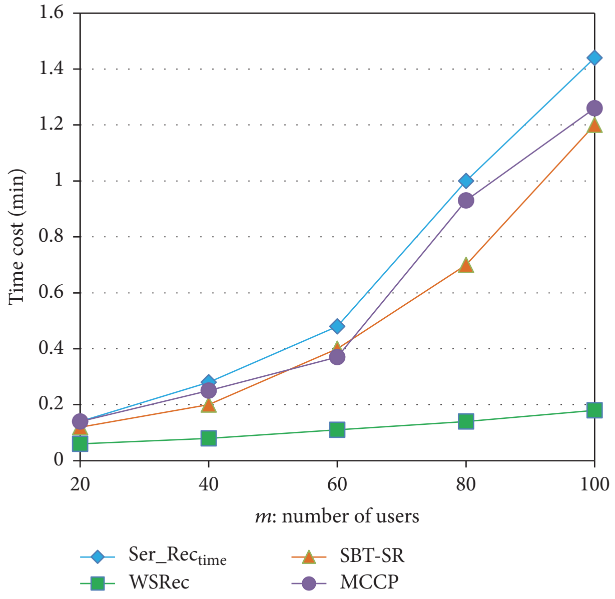


Figure 6: Time cost comparison with respect to m.

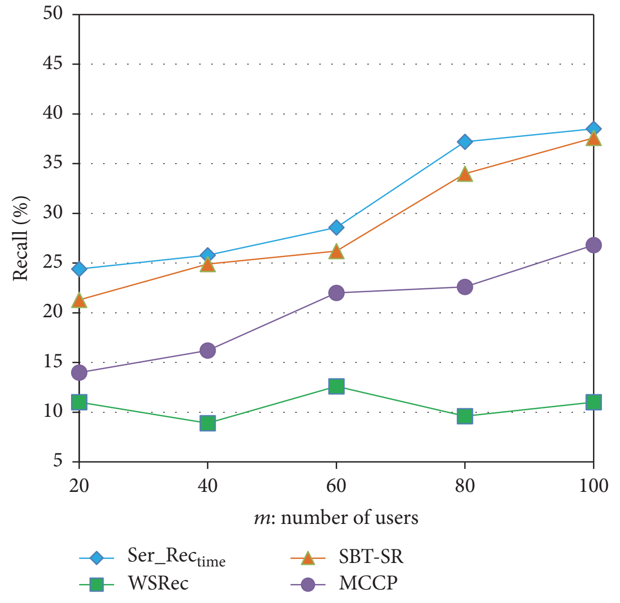


Figure 5: Recommendation recall comparison with respect to m.

more useful user-service relationships could be mined and recruited in service recommendation; hence, more qualified recommendation results are returned finally. However, the recall of MCCP is still not high as few really similar friends of target user could be found in the recommendation situations that we discuss in this paper (i.e., the sparse recommendation situations when target user has no similar friends and similar services).

While both SBT-SR and Ser_Rec$_{\text{time}}$ approaches achieve good performances in recommendation recall, as Social Balance Theory is recruited to find out the indirect friends of target user even if the target user has no similar friends and similar services. Furthermore, our proposed Ser_Rec$_{\text{time}}$ approach often outperforms SBT-SR in recall. This is because only "enemy's enemy is a friend" rule of Social Balance Theory is recruited in SBT-SR, while in Ser_Rec$_{\text{time}}$, all four rules in Figure 2 are considered, which improves the recommendation hit rate to some extent.

(Profile 3) Execution Efficiency Comparison with respect to m. In this profile, we test the execution efficiency of four approaches with respect to the number of users, i.e., m. Here, m is varied from 20 to 100; the number of services, that is, $n = 1000$, holds; besides, $P = -0.5$, $\alpha = 0.09$, and $W_{\text{load}} = W_{\text{version}} = 1$ hold. The experiment result is shown in Figure 6.

As Figure 6 shows, time cost of WSRec is the best, as WSRec only adopts the average ratings of target user and target services, without complex computation. The time costs of the remaining three approaches all increase with the growth of m quickly, as more similarity computation cost is required to determine the similar friends or dissimilar enemies of a user when the number of users increases. Furthermore, Ser_Rec$_{\text{time}}$ often requires more computation time as multiple iteration processes are probable for finding all the indirect friends of target user. However, as Figure 6 indicates, the execution efficiency of Ser_Rec$_{\text{time}}$ is often acceptable (at "minute" level).

(Profile 4) Execution Efficiency Comparison with respect to n. In this profile, we test the execution efficiency of four approaches with respect to the number of services, that is, n. Here, n is varied from 200 to 1000; the number of users, that is, $m = 100$, holds; besides, $P = -0.5$, $\alpha = 0.09$, and $W_{\text{load}} = W_{\text{version}} = 1$ hold. The concrete experiment result is shown in Figure 7.

As Figure 7 shows, similar to profile 3, the time cost of WSRec is the best due to the adopted average idea. Besides,

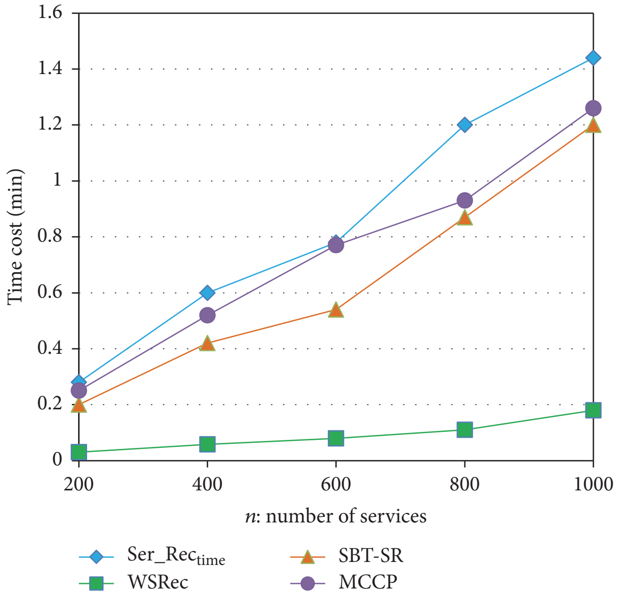


Figure 7: Time cost comparison with respect to n.

the time costs of the remaining three approaches all increase with the growth of m approximately linearly, as each service is considered at most once in each user similarity calculation process. However, as can be seen from Figure 7, the service recommendation process of Ser_Rec$_{\text{time}}$ approach could be generally finished in polynomial time.

5. Evaluations

In this section, we first analyze the time complexity of our proposed Ser_Rec$_{\text{time}}$ approach. Afterwards, related works and comparison analyses are presented to further clarify the advantages and application scope of our proposal. Finally, we point out our future research directions.

5.1. Complexity Analyses. Suppose there are m users and n services in the historical user-service invocation network.

Step 1. According to (1)–(6), the time-aware similarity between target user and any other user could be calculated, whose time complexity is $O(n)$. Afterwards, a user is judged to be an qualified enemy of target user or not based on (7), whose time complexity is $O(1)$. As there are totally m users in set U, the time complexity of this step is $O(m * (n + 1)) = O(m * n)$.

Step 2. In this step, we utilize the four rules (see Figure 3) in Social Balance Theory iteratively, so as to find all the indirect friends of target user. And, in the worst case, the similarity between any two users in set U needs to be calculated. As there are m users in set U, the time complexity of this step is $O(m^2 * n)$.

Step 3. For each derived indirect friend (at most $m-1$ indirect friends) of target user, we select his/her preferred services (at most n services) and recommend them to the target user. As the time complexity of preferred-service-judgment process is $O(1)$, the time complexity of this step is $O(m * n)$.

With the above analyses, we can conclude that the total time complexity of our proposed Ser_Rec$_{\text{time}}$ approach is $O(m^2 * n)$.

5.2. Related Works and Comparison Analyses. With the advent of IoE age, an excessive number of IoE services are emerging on the web, which places a heavy burden on the service selection decision of target users. In this situation, various recommendation techniques, for example, CF-based recommendation [28] and content-based recommendation [29], are put forward to help the target users find their interested IoE services.

A two-phase K-means clustering approach is brought forth in [30] to make service quality prediction and service recommendation. However, this clustering-based approach often requires a dense historical user-service invocation matrix, and hence cannot deal with the service recommendation problem on sparse data very well. In [31], a CF-based recommendation approach is put forward, which realizes service recommendation based on the similar friends of target user. However, when the target user has no similar friends, the recommendation accuracy is decreased significantly. A bidirectional (i.e., user-based CF + item-based CF) service recommendation approach named WSRec is brought forth in [25], for high-quality recommendation results. However, when a target user has no similar friends and similar services, WSRec can only make a rough prediction and recommendation, by considering both the average rating from target user and the average rating of the service that is ready to be predicted. In [26], a MCCP approach is put forward to capture and model the preferences of various users over different services; however, only similar friends of target user are recruited for service quality prediction and recommendation, which drops some valuable user-service relationship information hidden in the historical user-service invocation records.

In order to mine and introduce more user-service relationship information into recommendation, a service recommendation approach SBT-SR is proposed in our previous work [27]. By utilizing "enemy's enemy is a friend" rule in Social Balance Theory, some indirect friends of target user could be found and utilized for further recommendation, which improves the accuracy and recall of recommendation in sparse-data environment. However, SBT-SR has two shortcomings: first, service invocation time is not considered in SBT-SR, which may decrease the recommendation accuracy as service quality is often dynamic and varied with time; second, SBT-SR only employs "enemy's enemy is a friend" rule for service recommendation, while overlooks other valuable rules in Social Balance Theory, for example, "friend's friend is a friend" rule, "friend's enemy is an enemy" rule, and "enemy's friend is an enemy" rule. In view of the above two shortcomings, a novel time-aware service recommendation approach named Ser_Rec$_{\text{time}}$ is put forward in this paper, to deal with the recommendation problem in sparse-data environment. Ser_Rec$_{\text{time}}$ considers not only the service invocation time but also the four rules of Social Balance Theory, so that the recommendation accuracy and recall could be ensured. Finally, through a set of experiments deployed on a real web service quality dataset WS-DREAM, we validate the feasibility of Ser_Rec$_{\text{time}}$ in terms of recommendation accuracy, recall, and efficiency.

5.3. Further Discussions. In this paper, we put forward a time-aware similarity for service recommendation. Generally, the proposed time-aware similarity could also be applied in other similarity-based application domains, such as content searching [32–36], information detection [37–45], and quality optimization [46–50]. However, there are still several shortcomings in our paper, which are discussed as follows:

(1) The time cost of our proposed Ser_Rec$_{\text{time}}$ approach increases fast when the number of users grows. Therefore, the execution efficiency of Ser_Rec$_{\text{time}}$ needs to be improved, especially when a huge number of users are present in the historical user-service invocation records.

(2) In this paper, we have investigated the time-aware user similarity. However, besides service invocation time, many other factors, for example, user-service location information, also play an important role in user similarity calculation. In the future, we will improve our proposal by combining the time and location factors together.

6. Conclusions

In this paper, a novel time-aware service recommendation approach, that is, Ser_Rec$_{\text{time}}$, is put forward, to handle the service recommendation problems in sparse-data environment where target user has no similar friends and similar services. Instead of looking for similar friends in traditional CF-based service recommendation approaches, in Ser_Rec$_{\text{time}}$, we first look for dissimilar enemies of target user based on time-aware user similarity and further determine the indirect friends of target user based on Social Balance Theory. Afterwards, the services preferred by indirect friends of target user are recommended to the target user. Finally, through a set of experiments deployed on a real web service quality dataset WS-DREAM, we validate the feasibility of our proposal in terms of recommendation accuracy, recall, and efficiency.

In the future, we will improve the recommendation effect of our proposal by considering more user-service location information. Moreover, distributed or parallel recommendation approaches will be investigated in the future to improve the recommendation efficiency.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This paper is partially supported by Natural Science Foundation of China (no. 61402258, no. 61602253, no. 61672276, no. 61373027, and no. 61672321) and Open Project of State Key Laboratory for Novel Software Technology (no. KFKT2016B22).

References

[1] Y. Duan, G. Fu, N. Zhou, X. Sun, N. C. Narendra, and B. Hu, “Everything as a service (XaaS) on the cloud: origins, current and future trends,” in *Proceedings of the 8th International Conference on Cloud Computing (Cloud ’15)*, pp. 621–628, New York, NY, USA, July 2015.

[2] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, “Mutual verifiable provable data auditing in public cloud storage,” *Journal of Internet Technology*, vol. 16, no. 2, pp. 317–323, 2015.

[3] J. Shen, H. Tan, J. Wang, J. Wang, and S. Lee, “A novel routing protocol providing good transmission reliability in underwater sensor networks,” *Journal of Internet Technology*, vol. 16, no. 1, pp. 171–178, 2015.

[4] C. Yuan, X. Sun, and L. V. Rui, “Fingerprint liveness detection based on multi-scale LPQ and PCA,” *China Communications*, vol. 13, no. 7, pp. 60–65, 2016.

[5] X. Wen, L. Shao, Y. Xue, and W. Fang, “A rapid learning algorithm for vehicle classification,” *Information Sciences*, vol. 295, no. 1, pp. 395–406, 2015.

[6] Z. Xia, X. Wang, X. Sun, and Q. Wang, “A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[7] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, “Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing,” *IEICE Transactions on Communications*, vol. 98, no. 1, pp. 190–200, 2015.

[8] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, “A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[9] Z. Pan, Y. Zhang, and S. Kwong, “Efficient motion and disparity estimation optimization for low complexity multiview video coding,” *IEEE Transactions on Broadcasting*, vol. 61, no. 2, pp. 166–176, 2015.

[10] S. Xie and Y. Wang, “Construction of tree network with limited delivery latency in homogeneous wireless sensor networks,” *Wireless Personal Communications*, vol. 78, no. 1, pp. 231–246, 2014.

[11] Z. Zhou, Y. Wang, J. Wu, C. N. Yang, and X. Sun, “Effective and efficient global context verification for image copy detection,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 48–63, 2016.

[12] L. Qi, X. Xu, X. Zhang et al., “Structural Balance Theory-based E-commerce recommendation over big rating data,” *IEEE Transactions on Big Data*, 2016.

[13] T. Ma, J. Zhou, M. Tang et al., “Social network and tag sources based augmenting collaborative recommender system,” *IEICE Transactions on Information and Systems*, vol. 98, no. 4, pp. 902–910, 2015.

[14] L. Qi, W. Dou, C. Hu, Y. Zhou, and J. Yu, "A context-aware service evaluation approach over big data for cloud applications," *IEEE Transactions on Cloud Computing*.

[15] X. Fan, Y. Hu, R. Zhang, W. Chen, and P. Brézillon, "Modeling temporal effectiveness for context-aware web services recommendation," in *Proceedings of the IEEE International Conference on Web Services (ICWS '15)*, pp. 225–232, New York, NY, USA, June 2015.

[16] S. Wang, L. Huang, C.-H. Hsu, and F. Yang, "Collaboration reputation for trustworthy Web service selection in social networks," *Journal of Computer and System Sciences*, vol. 82, no. 1, pp. 130–143, 2016.

[17] S. Wang, Y. Ma, B. Cheng, F. Yang, and R. Chang, "Multidimensional QoS prediction for service recommendations," *IEEE Transaction on Services Computing*, 2016.

[18] Y. Ma, S. Wang, P. C. Hung, C. H. Hsu, Q. Sun, and F. Yang, "A highly accurate prediction algorithm for unknown web service QoS value," *IEEE Transactions on Services Computing*, vol. 9, no. 4, pp. 511–523, 2016.

[19] S. Wang, L. Huang, L. Sun, C.-H. Hsu, and F. Yang, "Efficient and reliable service selection for heterogeneous distributed software systems," *Future Generation Computer Systems*, 2016.

[20] Y. Ni, Y. Fan, W. Tan, K. Huang, and J. Bi, "NCSR: negative-connection-aware service recommendation for large sparse service network," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 579–590, 2016.

[21] Y. Zhong, Y. Fan, K. Huang, W. Tan, and J. Zhang, "Time-aware service recommendation for mashup creation," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 356–368, 2015.

[22] D. Cartwright and F. Harary, "Structural balance: a generalization of Heider's theory," *Psychological Review*, vol. 63, no. 5, pp. 277–293, 1956.

[23] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world web services," *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 32–39, 2014.

[24] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the the 10th International Conference on World Wide Web*, pp. 285–295, ACM, Hong Kong, May 2001.

[25] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," *IEEE Transactions on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[26] Y. Rong, X. Wen, and H. Cheng, "A Monte Carlo algorithm for cold start recommendation," in *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*, pp. 327–336, ACM, Seoul, South Korea, April 2014.

[27] L. Qi, X. Zhang, Y. Wen, and Y. Zhou, "A Social Balance Theory-based service recommendation approach," in *Advances in Services Computing: 9th Asia-Pacific Services Computing Conference, APSCC 2015, Bangkok, Thailand, December 7–9, 2015, Proceedings*, vol. 9464 of *Lecture Notes in Computer Science*, pp. 48–60, Springer, Berlin, Germany, 2015.

[28] F. Zhang, T. Gong, V. E. Lee, G. Zhao, C. Rong, and G. Qu, "Fast algorithms to evaluate collaborative filtering recommender systems," *Knowledge-Based Systems*, vol. 96, pp. 96–103, 2016.

[29] L. Yao, Q. Z. Sheng, A. H. H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based web service recommendation," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 453–466, 2015.

[30] C. Wu, W. Qiu, Z. Zheng, X. Wang, and X. Yang, "QoS prediction of web services based on two-phase K-means clustering," in *Proceedings of the IEEE International Conference on Web Services (ICWS '15)*, pp. 161–168, IEEE, New York, NY, USA, July 2015.

[31] S.-Y. Lin, C.-H. Lai, C.-H. Wu, and C.-C. Lo, "A trustworthy QoS-based collaborative filtering approach for web service discovery," *Journal of Systems and Software*, vol. 93, pp. 217–228, 2014.

[32] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Towards efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[33] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[34] Z. Pan, P. Jin, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast reference frame selection based on content similarity for low complexity HEVC encoder," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 516–524, 2016.

[35] Z. Fu, F. Huang, X. Sun, A. V. Vasilakos, and C. Yang, "Enabling semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Transactions on Services Computing*, 2016.

[36] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM '16)*, pp. 1–9, San Francisco, Calif, USA, April 2016.

[37] Z. Xia, X. Wang, X. Sun, and B. Wang, "Steganalysis of least significant bit matching using multi-order differences," *Security and Communication Networks*, vol. 7, no. 8, pp. 1283–1291, 2014.

[38] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[39] Z. Pan, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast motion estimation based on content property for low-complexity H.265/HEVC encoder," *IEEE Transactions on Broadcasting*, vol. 62, no. 3, pp. 675–684, 2016.

[40] Y. Zheng, B. Jeon, D. Xu, Q. M. J. Wu, and H. Zhang, "Image segmentation by generalized hierarchical fuzzy C-means algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 28, no. 2, pp. 961–973, 2015.

[41] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, and J. L. Coatrieux, "Color image analysis by quaternion-type moments," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 124–144, 2015.

[42] Z. Xia, X. Wang, X. Sun, Q. Liu, and N. Xiong, "Steganalysis of LSB matching using differences between nonadjacent pixels," *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 1947–1962, 2016.

[43] Y. Zhang, X. Sun, and B. Wang, "Efficient algorithm for k-barrier coverage based on integer linear programming," *China Communications*, vol. 13, no. 7, pp. 16–23, 2016.

[44] J. Wang, T. Li, Y. Shi, S. Lian, and J. Ye, "Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics," *Multimedia Tools and Applications*, 2016.

[45] Z. Zhou, C. Yang, B. Chen, X. Sun, Q. Liu, and Q. J. Wu, "Effective and efficient image copy detection with resistance

to arbitrary rotation," *IEICE Transactions on Information and Systems D*, vol. 99, no. 6, pp. 1531–1540, 2016.

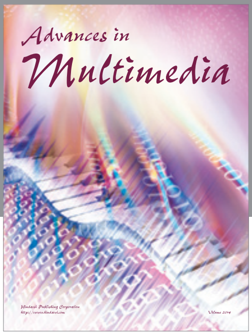
[46] L. Qi, W. Dou, and J. Chen, "Weighted principal component analysis-based service selection method for multimedia services in cloud," *Computing*, vol. 98, no. 1-2, pp. 195–214, 2016.

[47] Y. Chen, C. Hao, W. Wu, and E. Wu, "Robust dense reconstruction by range merging based on confidence estimation," *Science China Information Sciences*, vol. 59, no. 9, Article ID 092103, 11 pages, 2016.

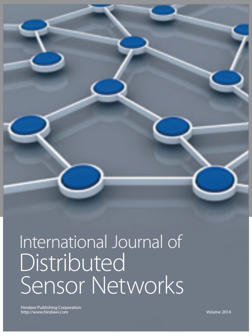
[48] Q. Liu, W. Cai, J. Shen, Z. Fu, X. Liu, and N. Linge, "A speculative approach to spatial–temporal efficiency with multi–objective optimization in a heterogeneous cloud environment," *Security and Communication Networks*, vol. 9, no. 17, pp. 4002–4012, 2016.

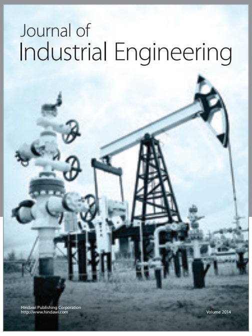
[49] Y. Kong, M. Zhang, and D. Ye, "A belief propagation-based method for task allocation in open and dynamic cloud environments," *Knowledge-Based Systems*, vol. 115, pp. 123–132, 2017.

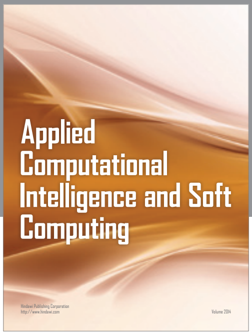
[50] B. Gu, V. S. Sheng, and S. Li, "Bi-parameter space partition for cost-sensitive SVM," in *Proceedings of the 24th International Conference on Artificial Intelligence (ICAI '15)*, pp. 3532–3539, Las Vegas, Nev, USA, July 2015.

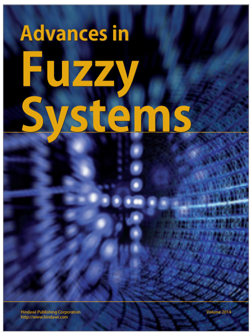

Advances in Multimedia

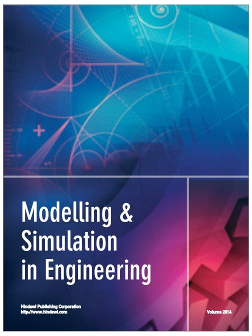
The Scientific World Journal

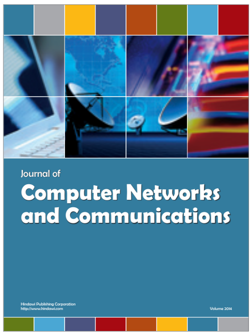

International Journal of Distributed Sensor Networks

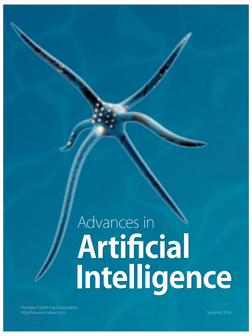

Journal of Industrial Engineering

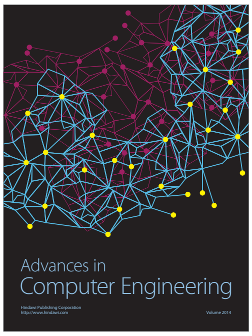

Applied Computational Intelligence and Soft Computing


Advances in Fuzzy Systems

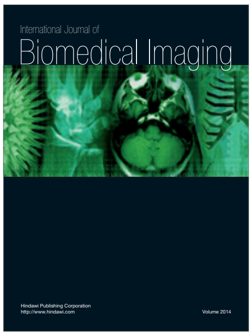

Modelling & Simulation in Engineering

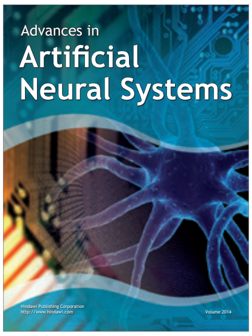

Journal of Computer Networks and Communications

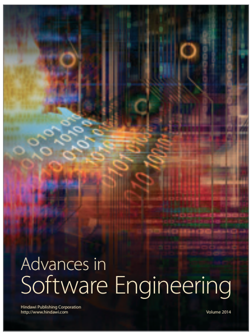

Advances in Artificial Intelligence


Advances in Computer Engineering

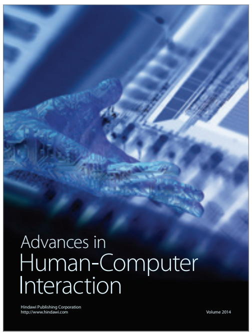
International Journal of Computer Games Technology

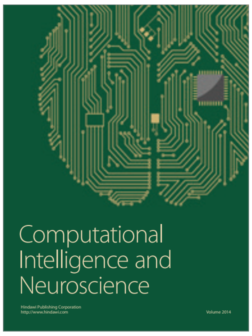

International Journal of Biomedical Imaging


Advances in Artificial Neural Systems


Advances in Software Engineering

Journal of Robotics


Advances in Human-Computer Interaction


Computational Intelligence and Neuroscience

International Journal of Reconfigurable Computing

Journal of Electrical and Computer Engineering