An Inverse Collaborative Filtering Approach for Cold-start Problem in Web Service Recommendation

Lianyong Qi
Qufu Normal University
Rizhao, China, 276826
+86-0633-3981060
lianyongqi@gmail.com

Wanchun Dou
Nanjing University
Nanjing, China, 210023
+86-025-89686399
douwc@nju.edu.cn

Xuyun Zhang
University of Auckland
Auckland, New Zealand, 1010
+64-9-923-8857
xuyun.zhang@auckland.ac.nz

ABSTRACT
Due to the increasing volume and variety of web services in different service communities, users are apt to find their interested web services through various recommendation techniques, e.g., Collaborative Filtering (i.e., CF) recommendation. In CF (e.g., user-based CF, item-based CF or hybrid CF) recommendation, "the similar friends of target user" or "the similar services of target services (i.e., the services preferred by target user)" are determined first; afterwards, "the services preferred by similar friends" or "the similar services of target services" are recommended to the target user. However, due to the inherent data sparsity in service recommendation, cold-start problem is inevitable when the target user has no similar friends and the target services have no similar services. While present CF recommendation approaches cannot deal with this cold-start problem very well. In view of this shortcoming, in this paper, a novel inverse CF approach named *Inverse_CF_Rec* is introduced to help alleviate the cold-start problem in service recommendation. Concretely, in *Inverse_CF_Rec*, we first look for the target user's enemy (i.e., antonym of "friend"), and then determine the target user's "possible friends" based on Social Balance Theory (e.g., "enemy's enemy is a friend" rule). Afterwards, "the services preferred by "possible friends" of target user" or "the services disliked by enemies of target user" are recommended to the target user, so as to alleviate the cold-start problem. Finally, through a set of simulation experiments deployed on well-known MovieLens-1M dataset, we validate the feasibility of our proposal in terms of recommendation accuracy, recall and efficiency.

CCS Concepts
• Theory of computation → Theory and algorithms for application domains • Human-centered computing → Collaborative and social computing.

Keywords
Service recommendation; Cold-start; Friend user; Enemy user; Inverse Collaborative Filtering; Social Balance Theory.[1]

1. INTRODUCTION
With the gradual development and maturity of SOA (Service-Oriented Architecture), an increasing number of individual or enterprise developers are apt to encapsulate their business applications into various smart web services, ranging from lightweight weather forecast to large-scale scientific computing [1-3]. With the continuous growth of registered web services, web service community is becoming an indispensable platform for web service sharing, which significantly benefits the users from different domains to build their complex business applications.

However, with the increase of volume and variety of web services registered in service communities, it becomes an important but difficult task to recommend appropriate web services to the target user [4-5]. In this situation, various recommendation techniques are brought forth to alleviate the heavy burden on service selection decision of target user, e.g., wide spread Collaborative Filtering (i.e., CF) recommendation [6]. In traditional CF (e.g., user-based CF, item-based CF or hybrid CF) recommendation, "similar friends of target user" or "similar services of target services (i.e., the services preferred by target user)" would be determined first; afterwards, "the services preferred by similar friends" or "the similar services of target services" would be recommended to the target user.

Generally, the traditional CF recommendation approaches can work very well when the historical user-service rating matrix is dense (in this paper, we only discuss the service recommendation problem based on the subjective user-service ratings). However, in certain situations, the historical user-service rating matrix is not dense, but sparse [7]. Therefore, it is possible that the target user has no similar friends and the target services (i.e., the services preferred by target user) have no similar services; in this situation, a cold-start recommendation problem is raised. While the traditional CF recommendation approaches cannot deal with this cold-start problem very well; this brings a great challenge for the robustness and accuracy of service recommendation.

In view of this challenge, a novel inverse CF recommendation approach named *Inverse_CF_Rec* is introduced in this paper. Instead of looking for the similar friends of target user in traditional CF recommendation approaches, in *Inverse_CF_Rec*, we first look for the target user's enemy (antonym of "friend"; i.e., the user who holds opposite preferences with target user. This is also the meaning of "inverse" in our recommendation approach) and then further determine the target user's "possible friends" based on Social Balance Thoery (e.g., "enemy's enemy is a friend" rule). And finally, we choose the web services preferred by the "possible friends" of target user or the services disliked by the enemy of target user, and recommend them to the target user.

The rest of paper is organized as follows. In Section 2, we first formalize the CF-based service recommendation problem, and then present an example to further demonstrate the motivation of our paper. In Section 3, a novel inverse CF recommendation approach named *Inverse_CF_Rec* is put forward. In Section 4, a set of simulation experiments deployed on well-known MovieLens-1M dataset are designed and tested, to validate the feasibility of our proposal in terms of cold-start recommendation accuracy, recall and efficiency. Related works and comparison analyses are introduced in Section 5. And finally, in Section 6, we conclude our paper and point out the future research directions.

ACSW '17, January 31-February 03, 2017, Geelong, Australia
© 2017 ACM. ISBN 978-1-4503-4768-6/17/01…$15.00
DOI: http://dx.doi.org/10.1145/3014812.3014860

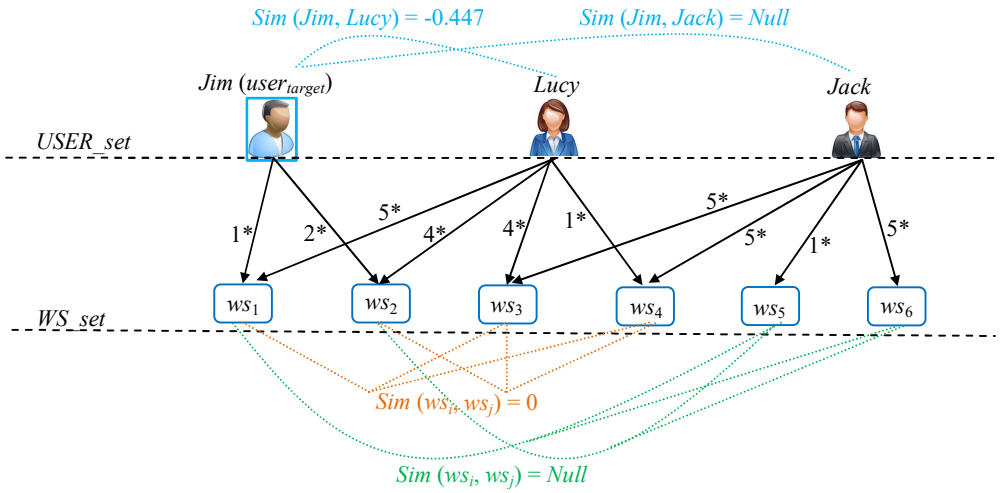

Figure 1. A service recommendation scenario where traditional CF recommendation approaches fail.

2. FORMALIZATION AND MOTIVATION

The CF recommendation problem is a hot research topic and has attracted much attention from many domain researchers. As our previous research work [8] did, in this section, we first formalize the CF recommendation problem for web services, and afterwards, we demonstrate the motivation of our paper with an intuitive example.

2.1 Formalization of Service Recommendation Problem

In this paper, the CF recommendation problem for web services could be formalized with a four-tuple *CF_Ser_Recommendation* (*USER_set, WS_set, RATING_set*, $user_{target}$), where

(1) *USER_set* = {$user_1$, …, $user_m$}: user set in the historical user-service invocation & rating network, where m denotes the number of users.

(2) *WS_set* = {ws_1, …, ws_n}: web service set in the historical user-service invocation & rating network, where n denotes the number of web services.

(3) *RATING_set* = {$r_{i-j} \mid 1 \le i \le m, 1 \le j \le n$ }: rating record set in the historical user-service invocation & rating network, where r_{i-j} denotes user $user_i$'s rating value over web service ws_j after service invocation. For simplicity, the classic 1*~5* rating system is recruited to depict r_{i-j} in this paper.

(4) $user_{target}$: the target user who requires recommended web services. And in this paper, constraint condition $user_{target} \in USER\_set$ holds.

Then with the above four-tuple *CF_Ser_Recommendation* (*USER_set, WS_set, RATING_set*, $user_{target}$), we can specify the CF recommendation problem for web services more formally as below: according to the historical rating record set *RATING_set* of web services (in *WS_set*) rated by users (in *USER_set*), select appropriate web services ws_x ($\in WS\_set$) that may be preferred by target user $user_{target}$ ($\in USER\_set$) and recommend them to $user_{target}$.

2.2 Motivation

Next, we present an example (in Fig.1) to further demonstrate the motivation of our paper. In Fig.1, three users are present, i.e., *USER_set* = {*Jim*, *Lucy*, *Jack*} (*Jim* is target user); besides, six services are present, i.e., *WS_set* = {ws_1, …, ws_6}; the historical user-service rating data (1* ~ 5*) is also shown in Fig.1. Here, please note that the services preferred by target user *Jim*, i.e., ws_1 and ws_2 are called "target services".

Then according to the Adjusted Cosine Similarity [9] (as user-service rating data is often discrete value and the rating scales of different users are varied, Adjusted Cosine Similarity is more suitable here), we can calculate the similarity *Sim* (*Jim*, $user_i$) between target user *Jim* and other users $user_i$ (i.e., *Lucy* and *Jack*). Concretely, as Fig.1 shows, *Sim* (*Jim*, *Lucy*) = - 0.447 while *Sim* (*Jim*, *Jack*) = *Null* (as *Jim* and *Jack* have no commonly rated web services). So according to the user-based CF recommendation theory, target user *Jim* has no similar friends; and therefore, traditional user-based CF recommendation approaches fail to generate a satisfying recommendation result.

Likewise, we can calculate the similarity between target services (i.e., ws_1 and ws_2) and other services (i.e., ws_3, ws_4, ws_5, ws_6). Concretely, as Fig.1 shows, *Sim* (ws_1, ws_3) = *Sim* (ws_1, ws_4) = *Sim* (ws_2, ws_3) = *Sim* (ws_2, ws_4) = 0, while *Sim* (ws_1, ws_5) = *Sim* (ws_1, ws_6) = *Sim* (ws_2, ws_5) = *Sim* (ws_2, ws_6) = *Null* (as no user rated both ws_1 (or ws_2) and ws_5 (or ws_6)). Therefore, according to the item-based CF recommendation theory, the target services (i.e., ws_1 and ws_2) have no similar services; and hence, traditional item-based CF recommendation approaches cannot generate a satisfying recommendation result.

From the above analyses, we can see that the traditional CF recommendation approaches for web services (including user-based CF, item-based CF or hybrid CF) fail to generate a satisfying recommendation result, when the target user has no similar friends and the target services do not have similar services (i.e., cold-start recommendation). This raises a great challenge for target user's accurate service recommendation. In view of this challenge, in the next section, we put forward a novel inverse CF recommendation approach, i.e., *Inverse_CF_Rec* to alleviate the cold-start problem in web service recommendation.

3. INVERSE CF RECOMMENDATION APPROACH FOR WEB SERVICES

In this section, we introduce a novel inverse CF recommendation approach for web services, i.e., *Inverse_CF_Rec*. Concretely, in Subsection 3.1, two hypotheses are first introduced and explained; they are the preconditions that *Inverse_CF_Rec* approach works. Afterwards, the concrete process of *Inverse_CF_Rec* is introduced in Subsection 3.2, step by step. Finally, in Subsection 3.3, the time complexity of our proposal is analyzed.

3.1 Hypotheses

Concretely, the following two hypotheses are important in our proposed *Inverse_CF_Rec* approach.

(1) hypothesis-1: it is probable that the target user has dissimilar "enemy" (antonym of "friend", i.e., the user who holds opposite preferences with target user).

Explanation: With the historical user-service rating data, it is possible to find out some users who have opposite preferences with target user (i.e., target user's enemy), although the probability is not always definitely 100%. For example, let's consider the service recommendation scenario in Fig.1. Target user *Jim*'s rating over ws_1 is 1*, while *Lucy*'s rating over ws_1 is 5*. Therefore, *Lucy* has completely opposite preferences with *Jim* if we only consider web service ws_1, and hence, *Lucy* is a possible enemy of *Jim*.

(2) hypothesis-2: the web services disliked by the enemies of target user may be preferred by target user.

Explanation: Traditional user-based CF recommendation approaches are based on an underlying hypothesis, i.e., the web services preferred by the friends of target user may be preferred by target user. Likewise, in this paper, we assume that if a web service is disliked by the enemies of target user, then the service is probably preferred by target user. For example, let's still consider the scenario in Fig.1. As analyzed in hypothesis-1, *Lucy* is a possible enemy of target user *Jim*, while *Lucy* dislikes web service ws_4 (with 1* rating); therefore, according to the trust propagation theory, it is natural to infer that target user *Jim* probably prefers web service ws_4.

3.2 An Inverse CF Recommendation Approach: *Inverse_CF_Rec*

With the above two hypotheses, a novel inverse CF recommendation approach for web services, i.e., *Inverse_CF_Rec* is brought forth in this subsection. The main idea of *Inverse_CF_Rec* is: first, we look for the target user's direct "enemies" based on user similarity; afterwards, we determine the target user's indirect "enemies" and indirect "friends", based on the derived direct "enemies" of target user and Social Balance Theory (e.g., "enemy's friend is an enemy" rule, "enemy's enemy is a friend" rule); finally, we select the web services that are preferred by the "friends" (i.e., indirect friends) of target user, as well as the services that are disliked by the "enemies" (i.e., direct enemies ∪ indirect enemies) of target user, and recommend them to target user.

Concretely, the proposed *Inverse_CF_Rec* approach consists of the three steps specified in Fig.2. To ease the subsequent understanding of readers, the multiple symbols recruited in Fig.2, as well as their relationships are presented in Fig.3. Besides, the boundaries of three steps of *Inverse_CF_Rec* approach are also shown in Fig.3.

Step1: Looking for the direct enemies of target user. Calculate similarity $Sim(user_{target}, user_i)$ between $user_{target}$ and $user_i$ ($user_i \in USER\_set$ and $user_i \neq user_{target}$), and determine $user_{target}$'s direct enemy set $Direct\_enemy\_set(user_{target})$ based on user similarity and predefined similarity threshold P ($-1 \le P \le -0.5$).

Step2: Determining target user's indirect enemies and indirect friends based on Social Balance Theory. For target user $user_{target}$, determine his/her indirect enemy set $Indirect\_enemy\_set(user_{target})$ and indirect friend set $Indirect\_friend\_set(user_{target})$, based on $Direct\_enemy\_set(user_{target})$ (derived in Step1) and Social Balance Theory.

Step3: Service recommendation. Select the services preferred by users in set $Indirect\_friend\_set(user_{target})$ and the services disliked by users in union set $Direct\_enemy\_set(user_{target}) \cup Indirect\_enemy\_set(user_{target})$, and recommend them to $user_{target}$.

Figure 2. Three steps of recommendation approach *Inverse_CF_Rec*.

(1)Step1: Looking for the direct enemies of target user.

In this step, we first calculate the similarity between target user $user_{target}$ and other users in *USER_set*. Concretely, for any user $user_i \in USER\_set$ ($user_i \neq user_{target}$), his/her similarity with $user_{target}$, i.e., $Sim(user_{target}, user_i)$ could be calculated by (1) based on the Adjusted Cosine Similarity.

Next, we introduce the recruited symbols in (1). set I contains the web services that were rated by both $user_{target}$ and $user_i$; I_{target} and I_i represent the service sets that were rated by $user_{target}$ and $user_i$, respectively; $r_{target\text{-}k}$ and $r_{i\text{-}k}$ represent $user_{target}$'s and $user_i$'s ratings on service ws_k, respectively; $\overline{r_{target}}$ and $\overline{r_i}$ respectively denote $user_{target}$'s and $user_i$'s average rating values. Here, please note that if $user_{target}$ and $user_i$ have no commonly rated web services (i.e., set $I = Null$), then their similarity $Sim(user_{target}, user_i) = Null$ holds. As (1) indicates, $Sim(user_{target}, user_i) \in [-1, 1]$ holds; and the smaller $Sim(user_{target}, user_i)$ is, the more probable that $user_{target}$ and $user_i$ are direct enemies, vice versa.

$$Sim(user_{target}, user_i) = \frac{\sum_{ws_k \in I} (r_{target\text{-}k} - \overline{r_{target}}) * (r_{i\text{-}k} - \overline{r_i})}{\sqrt{\sum_{ws_k \in I_{target}} (r_{target\text{-}k} - \overline{r_{target}})^2} * \sqrt{\sum_{ws_k \in I_i} (r_{i\text{-}k} - \overline{r_i})^2}} \quad (1)$$

With the calculated user similarity $Sim(user_{target}, user_i)$ in (1), we can determine $user_{target}$'s direct enemy set $Direct\_enemy\_set(user_{target})$ (see Fig.3). Concretely, first, we set a similarity threshold for enemy relationship, i.e., P ($-1 \le P \le -0.5$); second, with threshold P and derived $Sim(user_{target}, user_i)$ in (1), set $Direct\_enemy\_set(user_{target})$ could be obtained by (2).

$$user_i \begin{cases} \in Direct\_enemy\_set(user_{target}) & \text{if } Sim(user_{target}, user_i) \le P \\ \notin Direct\_enemy\_set(user_{target}) & \text{otherwise} \end{cases} \quad (2)$$

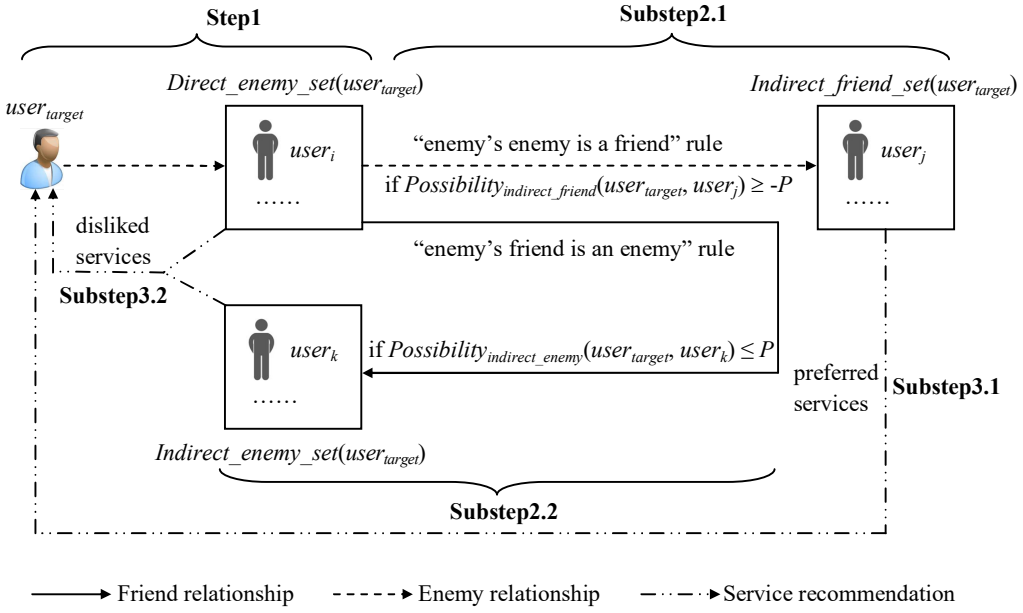

Figure 3. Recruited symbols in Fig.2 and their relationships.

(2)Step2: Determining target user's indirect enemies and indirect friends based on Social Balance Theory.

In Step1, we have obtained target user's direct enemy set *Direct_enemy_set*($user_{target}$) (as introduced in paper motivation, target user has no similar friends; therefore, target user's direct enemy set is an unique basis for subsequent service recommendation). Next, we introduce how to obtain the target user's indirect friend set *Indirect_friend_set*($user_{target}$) and indirect enemy set *Indirect_enemy_set*($user_{target}$) based on Social Balance Theory and *Direct_enemy_set*($user_{target}$) derived in Step1. Concretely, Step2 consists of the following two substeps.

Substep2.1: Determining target user's indirect friend set *Indirect_friend_set*($user_{target}$).

Social Balance Theory [10] investigates the various social relationships among different social partners and reveals the important hidden information that is valuable to cold-start problem in recommendation. Therefore, in this paper, we utilize Social Balance Theory to help discover the target user's indirect friend set *Indirect_friend_set*($user_{target}$). This discovering process is mainly based on "enemy's enemy is a friend" rule in Social Balance Theory.

Concretely, for any $user_i \in$ *Direct_enemy_set*($user_{target}$), we first calculate his/her similarity *Sim*($user_i$, $user_j$) with other users $user_j \in USER\_set$ ($i \neq j$ and $user_j \neq user_{target}$) by (1). Afterwards, we determine $user_i$'s direct enemies $user_j$ by (2) and put them into set *Direct_enemy_set*($user_i$). As $user_i$ is a direct enemy of $user_{target}$ while $user_j$ is a direct enemy of $user_i$; according to "enemy's enemy is a friend" rule in Social Balance Theory, a conclusion could be drawn that $user_j$ is an indirect friend of target user $user_{target}$. Here, please note that $user_j$ is just a "possible" indirect friend of $user_{target}$. Next, in order to measure this possibility, a new measurement criterion *Possibility*$_{indirect\_friend}$($user_{target}$, $user_j$) is put forward, which could be calculated by formula (3). As shown in formula (2), $-1 \le Sim(user_{target}, user_i), Sim(user_i, user_j) \le P$, so *Possibility*$_{indirect\_friend}$($user_{target}$, $user_j$) $\in [P^2, 1]$ holds. Then $user_j$ is regarded as a qualified indirect friend of $user_{target}$ and put in set *Indirect_friend_set*($user_{target}$), iff condition in (4) holds. In (4), $-P$ ($-1 \le P \le -0.5$) denotes the user similarity threshold for friend relationship. Through the above calculation process, we can obtain the indirect friend set of target user, i.e., *Indirect_friend_set*($user_{target}$) (see Fig.3).

$$Possibility_{indirect\_friend}(user_{target}, user_j) = Sim(user_{target}, user_i) * Sim(user_i, user_j) \quad (3)$$

$$Possibility_{indirect\_friend}(user_{target}, user_j) \ge -P \quad (4)$$

Substep2.2: Determining target user's indirect enemy set *Indirect_enemy_set*($user_{target}$).

Likewise, we can discover the indirect enemy set *Indirect_enemy_set*($user_{target}$) of target user, based on the derived direct enemy set *Direct_enemy_set*($user_{target}$) in Step1. This discovering process is mainly based on "enemy's friend is an enemy" rule in Social Balance Theory.

Concretely, for any $user_i \in$ *Direct_enemy_set*($user_{target}$), his/her similarity values *Sim*($user_i$, $user_k$) with other users $user_k \in USER\_set$ ($i \neq k$ and $user_k \neq user_{target}$) have already be calculated by (1) in Substep2.1. Afterwards, we determine $user_i$'s direct friends $user_k$ by (5) and put them into set *Direct_friend_set*($user_i$). As $user_i$ is a direct enemy of $user_{target}$ while $user_k$ is a direct friend of $user_i$; according to "enemy's friend is an enemy" rule in Social Balance Theory, we can infer that $user_k$ is an indirect enemy of $user_{target}$. Here, similar to Substep2.1, $user_k$ is just a "possible" indirect enemy of $user_{target}$. Next, we measure this possibility with a new criterion *Possibility*$_{indirect\_enemy}$($user_{target}$, $user_k$) in (6). And if condition in (7) holds, $user_k$ is considered as a qualified indirect enemy of $user_{target}$ and put into set *Indirect_enemy_set*($user_{target}$) (see Fig.3). Then through the above calculation process, target user's indirect enemy set, i.e., *Indirect_enemy_set*($user_{target}$) is derived.

$$user_k \begin{cases} \in Direct\_friend\_set(user_i) & \text{if } Sim(user_i, user_k) \geq -P \\ \notin Direct\_friend\_set(user_i) & \text{otherwise} \end{cases} \quad \textbf{(5)}$$

$$Possibility_{indirect\_enemy}(user_{target}, user_k) = Sim(user_{target}, user_i) * Sim(user_i, user_k) \quad \textbf{(6)}$$

$$Possibility_{indirect\_enemy}(user_{target}, user_k) \leq P \quad \textbf{(7)}$$

Then through Step2, we can obtain the target user's indirect friend set $Indirect\_friend\_set(user_{target})$ and indirect enemy set $Indirect\_enemy\_set(user_{target})$. As introduced in the motivation part, target user has no direct friends; therefore, as Fig.3 shows, all the friends (i.e., direct friends + indirect friends) of target user are included in set $Indirect\_friend\_set(user_{target})$ (derived in Substep2.1). Likewise, all the enemies (i.e., direct enemies + indirect enemies) of target user are included in sets $Direct\_enemy\_set(user_{target})$ (derived in Step1) and $Indirect\_enemy\_set(user_{target})$ (derived in Substep2.2).

(3)Step3: Service recommendation.

After Step1 and Step2, we have obtained the target user's friend set $Indirect\_friend\_set(user_{target})$ and target user's enemy set ($Direct\_enemy\_set(user_{target}) \cup Indirect\_enemy\_set(user_{target})$). Next, in this step, service recommendation process is performed based on the above friend set and enemy set. Concretely, the service recommendation process consists of the following two substeps.

Substep3.1: Recommending the web services preferred by users in set $Indirect\_friend\_set(user_{target})$. Concretely, for any $user_i \in Indirect\_friend\_set(user_{target})$, we select $user_i$'s preferred services (with 4* and 5*) and put them in the recommended service set *Rec_ser_set*.

Substep3.2: Recommending the web services disliked by users in set ($Direct\_enemy\_set(user_{target}) \cup Indirect\_enemy\_set(user_{target})$). Concretely, for any $user_i$ that belongs to union set ($Direct\_enemy\_set(user_{target}) \cup Indirect\_enemy\_set(user_{target})$), we select $user_i$'s disliked services (with 1* and 2*) and put them in the recommended service set *Rec_ser_set*.

With the above Step1-Step3, a set of web services (i.e., in *Rec_ser_set*), are recommended to the target user, so as to alleviate the service recommendation cold-start problem. More formally, the pseudocode of our proposed *Inverse_CF_Rec* approach is specified as below.

3.3 Time Complexity Analyses

In this subsection, we analyze the time complexity of our proposed *Inverse_CF_Rec* approach. Suppose that m users and n web services are present in the historical user-service invocation & rating network.

Step1: **Looking for the direct enemies of target user.** For each user in *USER_set*, we calculate his/her similarity with target user by (1), whose time complexity is O(n) as n web services are considered at most. As m users are present in set *USER_set*, the time complexity of Step1 is O($m * n$).

Step2: **Determining target user's indirect enemies and indirect friends based on Social Balance Theory.** In Substep2.1, for target user's each direct enemy (m-1 direct enemies are present at most), we should calculate his/her similarities with other users (m-2 users are present at most) in *USER_set*; therefore, the time complexity of this process is O($m^2 * n$). Furthermore, the indirect friend set of target user could be determined by (2)-(4), whose time complexity

---

Algorithm: *Inverse_CF_Rec*

Inputs:

(1) $USER\_set = \{user_1, \ldots, user_m\}$: user set;

(2) $WS\_set = \{ws_1, \ldots, ws_n\}$: web service set;

(3) $RATING\_set = \{r_{i-j} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$: rating set;

(4) $user_{target}$: target user who requires recommended services.

Output:

Rec_ser_set: web services that are recommended to $user_{target}$

1: Set user similarity threshold for enemy relationship, i.e., P
2: **for** each $user_i \in (USER\_set - user_{target})$ **do** // Step1
3: calculate $Sim(user_{target}, user_i)$ by (1)
4: determine $Direct\_enemy\_set(user_{target})$ by (2)
5: **end for**
6: **for** each $user_i \in Direct\_enemy\_set(user_{target})$ **do** // Substep2.1
7: **for** each $user_j \in (USER\_set - user_{target} - user_i)$ **do**
8: calculate $Sim(user_i, user_j)$ by (1)
9: **end for**
10: determine $Direct\_enemy\_set(user_i)$ by (2)
11: **for** each $user_j \in Direct\_enemy\_set(user_i)$ **do**
12: calculate $Possibility_{indirect\_friend}(user_{target}, user_j)$ by (3)
13: **if** $Possibility_{indirect\_friend}(user_{target}, user_j) \geq -P$ // condition (4)
14: **then** put $user_j$ in set $Indirect\_friend\_set(user_{target})$
15: **end if**
16: **end for**
17: **end for**
18: **for** each $user_i \in Direct\_enemy\_set(user_{target})$ **do** // Substep2.2
19: determine $Direct\_friend\_set(user_i)$ by (5)
20: **for** each $user_k \in Direct\_friend\_set(user_i)$ **do**
21: calculate $Possibility_{indirect\_enemy}(user_{target}, user_k)$ by (6)
22: **if** $Possibility_{indirect\_enemy}(user_{target}, user_k) \leq P$ // condition (7)
23: **then** put $user_k$ in set $Indirect\_enemy\_set(user_{target})$
24: **end if**
25: **end for**
26: **end for**
27: **for** each $user_i \in Indirect\_friend\_set(user_{target})$ **do** // Substep3.1
28: **if** $r_{i-x} = \{4*, 5*\}$
29: **then** put service ws_x into set *Rec_ser_set*
30: **end if**
31: **end for**
32: $U = Direct\_enemy\_set(user_{target}) \cup Indirect\_enemy\_set(user_{target})$
33: **for** each $user_i \in U$ **do** // Substep3.2
34: **if** $r_{i-x} = \{1*, 2*\}$
35: **then** put service ws_x into set *Rec_ser_set*
36: **end if**
37: **end for**
38: **return** *Rec_ser_set*

---

is O(m). Therefore, time complexity of Substep2.1 is O(m^2 * n). In Substep2.2, the process of user similarity calculation is unnecessary, as the similarity has already been obtained in Substep2.1; furthermore, the indirect enemy set of target user could be determined by (5)-(7), whose time complexity is O(m). Therefore, the time complexity of Substep2.2 is O(m). With the above analyses, the time complexity of Step2 is O(m^2 * n).

Step3: **Service recommendation.** In Substep3.1, the services preferred by indirect friends of target user are recommended to the target user. As target user has m-2 indirect friends at most and each indirect friend has n recommended services at most, the time complexity of Substep3.1 is O(m * n). Likewise, in Substep3.2, the services disliked by direct and indirect enemies of target user are recommended to the target user, whose time complexity is also O(m * n). Therefore, time complexity of Step3 is O(m * n).

With the above analyses, we can conclude that the total time complexity of our proposed *Inverse_CF_Rec* approach is O(m^2*n), which means that the recommendation process could be finished in polynomial time.

4. EXPERIMENTS

In order to validate the feasibility of our proposed *Inverse_CF_Rec* approach, in this section, a set of simulation experiments are designed, tested and compared.

4.1 Dataset and Deployment

In this paper, we only focus on the user rating-based web service recommendation. However, the present real user-service rating data is really rare, which cannot meet the experiment requirements. Therefore, as work [11] did, the well-known MovieLens-1M dataset [12] is adopted here for simulation purpose. MovieLens-1M contains 1000209 user-movie ratings over 3952 movies by 6040 users.

As introduced in the motivation part, this paper only focuses on the service recommendation cold-start problem when target user has no similar friends and target services have no similar services; therefore, we construct appropriate dataset from MovieLens-1M to simulate the cold-start scenario. Furthermore, for each recruited target user, we divide his/her rating data into two parts: 80% ratings for training and 20% ratings for test.

In order to observe and compare the recommendation effect of different approaches, recommendation accuracy, recall and efficiency are tested in the experiments, respectively.

(1) Accuracy. *Mean Absolute Error* (*MAE*) is adopted here to measure the recommendation accuracy, which could be calculated by (8). In (8), *Rec_ser_set* denotes the recommended service set, and $|X|$ denotes the element number of set X, $r^{predict}_{target-x}$ and $r^{real}_{target-x}$ represent web service ws_x's predicted rating and real rating by $user_{target}$, respectively.

$$MAE = \sum_{ws_x \in Rec\_ser\_set} \frac{| r^{predict}_{target-x} - r^{real}_{target-x} |}{| Rec\_ser\_set |} \quad (8)$$

(2) Recall. *Recall* is recruited here to measure the recommendation hit rate, which could be calculated by (9). In (9), *Rec_ser_set* is the recommended service set, while *Preferred_ser_set* denotes target user's preferred service set (i.e., with 4* or 5* rating) in the 20% real test data; $|X|$ represents the element number of set X.

$$Recall = \frac{| Preferred\_ser\_set \cap Rec\_ser\_set |}{| Rec\_ser\_set |} \quad (9)$$

Besides, we compare our proposed *Inverse_CF_Rec* approach with other three ones, i.e., *WSRec* [13], *CAP* [14] and *SBT-SR* [8]. Concretely, in *WSRec*, both $r_{average}(user_{target})$ (i.e., $user_{target}$'s average rating value over all his/her rated services) and $r_{average}(ws_j)$ (i.e., ws_j's average rating value from all the users who rated ws_j) are considered, and their average value is accepted as the predicted rating value of ws_j by $user_{target}$; while in *CAP* approach, K-means clustering technique is recruited to find the similar neighbors of target user, so as to make further recommendation; in our previous *SBT-SR* approach, only "enemy's enemy is a friend" rule is recruited for service recommendation.

The experiments were conducted on a Dell pc with 2.40 GHz processors and 2.0 GB RAM. The software configuration environment is: Windows XP + JAVA 1.5. Each experiment was repeated 10 times and their average experiment results were adopted finally.

4.2 Experiment Results

In the experiments, the following four profiles are tested and compared respectively. Here, symbols m and n denote the number of users and number of web services in the historical user-service rating network, and P ($-1 \le P \le -0.5$) denotes the predefined user similarity threshold for enemy relationship.

(1) *Profile*1: Recommendation accuracy of different approaches

In this profile, the *MAE* values of different recommendation approaches are tested, to measure their respective recommendation accuracy data. The experiment parameter configuration is as follows: the similarity threshold for enemy relationship, i.e., P = -0.5 holds; the number of users, i.e., m = {200, 400, 600, 800, 1000}; for each recruited user, all his/her rating records are taken into consideration (concretely, the former 80% rating records for training and the rest 20% ones for testing). Concrete experiment results are presented in Fig.4.

As indicated in Fig.4, *WSRec*'s recommendation accuracy is the lowest (i.e., *MAE* is the largest), as *WSRec* only considers target user's average rating and target service (i.e., the service whose rating is going to be predicted by target user)'s average rating, without considering other valuable information hidden in user-service rating network. While the *CAP* approach does not perform very well in recommendation accuracy either, as the K-means clustering results by *CAP* are not very ideal when the user-service rating data is very sparse (the density of MovieLens-1M dataset recruited in our experiments is 4.5%). Besides, as shown in Fig.4, *SBT-SR* and *Inverse_CF_Rec* outperform the other two approaches in terms of recommendation accuracy, as more hidden social relationship information is taken into consideration in both *SBT-SR* and *Inverse_CF_Rec*. And generally, the recommendation accuracy values of *SBT-SR* and *Inverse_CF_Rec* are close, as "enemy's enemy is a friend" rule in Social Balance Theory is considered in both approaches.

(2) *Profile*2: Recommendation recall of different approaches

In this profile, the recall values of different recommendation approaches are tested respectively. The parameter configuration is as follows: user similarity threshold for enemy relationship, i.e., P = -0.5 holds; the number of users, i.e., m is varied from 200 to 1000. Experiment results are shown in Fig.5.

As shown in Fig.5, the recall of *WSRec* is low as the "average" idea is adopted in *WSRec*; while "average" idea often generates a large prediction error and hence leads to a small recall value. Besides, for *CAP* approach, its recommendation recall is very low either; this is because the K-means clustering algorithm recruited in *CAP* cannot generate a satisfying clustering result (towards target user) when the target user has no similar friends. While *SBT-SR* and *Inverse_CF_Rec* outperform the other two approaches as they consider more social structure information hidden in user-service rating network. Furthermore, the recall values of *SBT-SR* and *Inverse_CF_Rec* both increase with the growth of *m* (i.e., the number of users); this is because more valuable social structure information could be found and employed for cold-start service recommendation, when more users are present in the user-service rating network. Finally, as Fig.5 shows, our proposed *Inverse_CF_Rec* outperforms *SBT-SR* in recommendation recall, which is due to the following two reasons: (1) *SBT-SR* only considers "enemy's enemy is a friend" rule in Social Balance Theory, while *Inverse_CF_Rec* considers both "enemy's enemy is a friend" rule and "enemy's friend is an enemy" rule simultaneously; (2) *SBT-SR* only recommends the services preferred by the friends of target user, while in *Inverse_CF_Rec*, both the services preferred by friends of target user and the services disliked by enemies of target user are recommended to the target user.

(3) *Profile*3: Time cost of different approaches w.r.t. *m*

In this profile, we test the time cost of different recommendation approaches. The experiment parameter configuration is as follows: *m* is varied from 200 to 1000; similarity threshold P = -0.5 holds in both *SBT-SR* and *Inverse_CF_Rec*; K = 5 and *Top-K* = 2 hold in *CAP* approach. The experiment results are shown in Fig.6.

As Fig.6 shows, *WSRec*'s time cost stays stable and outperforms the other three approaches, as *WSRec* only adopts the simple average data that is easily calculated. While for the other three approaches, their time costs all increase with the growth of *m*, as more time is needed for clustering or similarity calculation when the number of users increases. Besides, *SBT-SR* and *Inverse_CF_Rec* achieve approximate execution efficiency as Social Balance Theory is considered in both approaches. Furthermore, although *Inverse_CF_Rec* does not outperform other approaches in time cost, its time cost is still polynomial.

(4) *Profile*4: Failure rate of *Inverse_CF_Rec* w.r.t. *P*

As Subsection3.1 shows, *Inverse_CF_Rec* can only work when hypothesis-1 holds, i.e., when target user has dissimilar "enemy". In other words, if the target user has no "enemy", the recommendation process fails. Therefore, it becomes a necessity to test the failure rate of our proposal. As formula (2) indicates, whether target user has an enemy depends on the predefined similarity threshold for enemy relationship, i.e., P ($-1 \le P \le -0.5$). Therefore, P is a key factor that affects the failure rate of *Inverse_CF_Rec*. Next, we test the relationship between failure rate and P. The experiment was repeated 200 times.

The experiment parameter configuration is as follows: P is varied from -1 to -0.5; the number of users, i.e., m = 200 holds. The concrete experiment result is presented in Fig.7. As Fig.7 shows, when P is very small (e.g., P = - 1 or - 0.9), *Inverse_CF_Rec* approach often fails to generate a recommended service, as target user has few qualified direct enemies based on the threshold P. While the failure rate drops significantly with the growth of P. And the failure rate stays approximately stable (around 0.5%) when P = - 0.6 holds, which means that the successful recommendation could be often ensured when P is large.

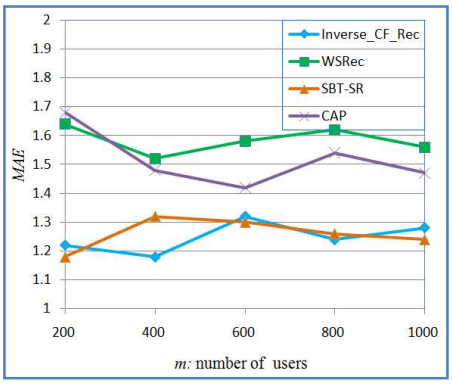

Figure 4. Recommendation accuracy comparison

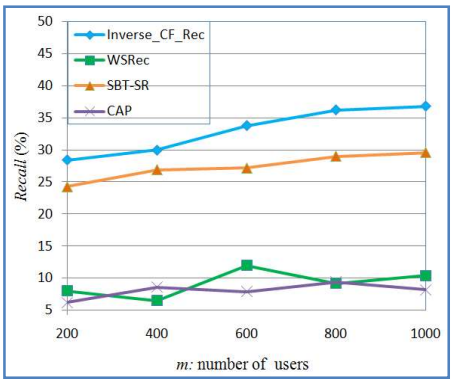

Figure 5. Recommendation recall comparison

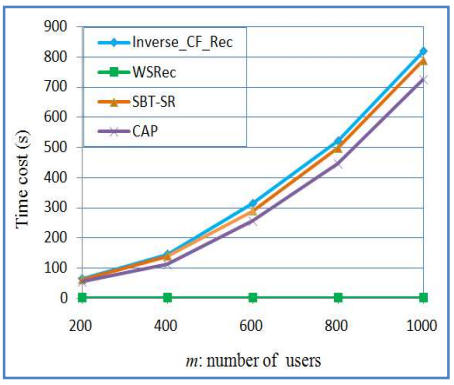

Figure 6. Recommendation efficiency comparison w.r.t. *m*

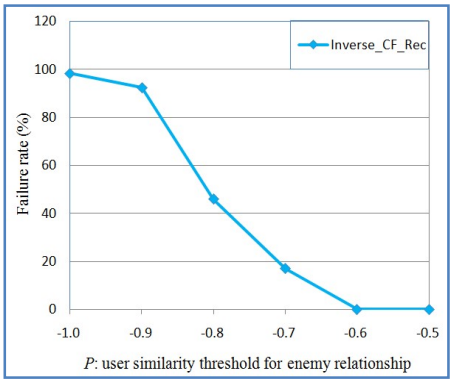

Figure 7. Failure rate of *Inverse_CF_Rec* w.r.t. *P*

5. RELATED WORKS & DISCUSSIONS

In this section, we first introduce the related works and compare them with our proposal; afterwards, we discuss the possible limitations of our paper and point out the future research directions.

5.1 Related Works and Comparison Analyses

With the increasing development of services computing technology, the volume and categories of available web services in various service communities both grow significantly [15-17]. Therefore, from the perspective of target user, it becomes an important but boring task to find his/her interested web services. In this situation, various service recommendation techniques are introduced to aid the target user's service selection decision, e.g., Content-based recommendation [18], Collaborative Filtering (i.e., CF) -based recommendation [13] and Hybrid recommendation [19].

Due to the easy-to-understand principle and easy-to-calculate process, CF is now widely adopted in various recommendation applications including service recommendation. A trustworthy service discovery approach is introduced in [20], which recruits user-based CF for service recommendation. A hybrid service recommendation approach named *SD-HCF* is put forward in [21], by integrating user-based CF recommendation and service-based CF recommendation, for better recommendation results. While above approaches simply employ CF for service recommendation, while omit some important and valuable recommendation information, e.g., user trust and user location. In view of the above shortcoming, in [22], trust relationship between different users is modeled and introduced in recommendation decisions. Namely, only the similar and trustable users are considered in recommendation process. Besides, location information of target user and candidate web services are taken into consideration in [23], where the web services with the "closest" distance from target user are recommended to the target user.

The above recommendation approaches can work very well when (1) the target user has similar friends, or (2) the services preferred by target user have similar services. However, due to the inherent sparsity of user-service rating data, it is possible that neither condition (1) nor condition (2) holds, i.e., a cold-start problem in service recommendation is raised. In view of this cold-start problem, a simple recommendation approach named *WSRec* is put forward in [13], where both $r_{average}(user_{target})$ (i.e., $user_{target}$'s average rating value over all his/her rated services) and $r_{average}(ws_j)$ (i.e., ws_j's average rating value from all the users who rated ws_j) are considered, and their average value is accepted as the predicted rating value of ws_j by $user_{target}$. However, the recommendation effect of *WSRec* is often low as the recruited "average" idea in *WSRec* can only generate an approximate prediction instead of an accuracy prediction. A K-means clustering-based recommendation approach named *CAP* is put forward in [14]. In *CAP*, the users who are often grouped in the same set as target user are considered as the similar users of target user; and furthermore, the *Top-K* similar users are recruited for service recommendation. However, due to the data sparsity, the similarity values between target user and *Top-K* similar users are often not high enough; therefore, the recommendation accuracy and recall of *CAP* are not as good as expected. In order to find and utilize more social relationship information hidden in user-service rating network for service recommendation, in our previous work [8], a Social Balance Theory-based recommendation approach named *SBT-SR* is put forward. In *SBT-SR*, "enemy's enemy is a friend" rule is recruited for finding the possible friends of target user; afterwards, the services preferred by the possible friends of target user are recommended to the target user.

In order to further improve the recommendation effect, on the basis of our previous work [8], a novel inverse CF recommendation approach, i.e., *Inverse_CF_Rec* is brought forth in this paper. Our improvements are two-fold: (1) more social relationship rules (e.g., "enemy's friend is an enemy" rule) are taken into consideration, for further improving the recommendation recall; (2) not only the services preferred by (indirect) friends of target user but also the services disliked by (direct + indirect) enemies of target user are recommended to the target user, so as to obtain more recommended services and alleviate the service recommendation cold-start problem. Finally, through a set of experiments, we validate the feasibility of our proposal in terms of recommendation accuracy, recall and efficiency.

5.2 Further Discussions

(1) As indicated in Profile3 and Fig.6, the time cost of our proposal is a bit large when many users are present in the user-service rating network. In the future, we will investigate distributed or parallel recommendation approach to improve the recommendation efficiency.

(2) As introduced in hypothesis-1 of Subsection3.1, our proposed *Inverse_CF_Rec* approach can work under the basic hypothesis that the enemies of target user exist. While in fact, if the user-service rating data is extremely sparse, our *Inverse_CF_Rec* approach fails to generate a recommendation result as the target user has no enemies. In the future, we will analyze this exceptional recommendation scenario.

6. CONCLUSIONS

In this paper, a novel inverse CF recommendation approach named *Inverse_CF_Rec* is put forward, to deal with the service recommendation cold-start problem where the target user has no similar friends and the services preferred by target user have no similar services. Concretely, in *Inverse_CF_Rec*, we first look for the direct enemies of target user, and then determine the target user's indirect friends and indirect enemies by considering "enemy's enemy is a friend" rule and "enemy's friend is an enemy" rule in Social Balance Theory; and finally, the services preferred by (indirect) friends of target user and the services disliked by (direct + indirect) enemies of target user are recommended to the target user. At last, through a set of experiments deployed on MovieLens-1M dataset, we demonstrate the feasibility of our proposal in terms of recommendation accuracy, recall and efficiency. In the future, we will improve the recommendation efficiency and investigate the recommendation approach for extremely sparse data environment.

7. ACKNOWLEDGMENTS

This paper is partially supported by Natural Science Foundation of China (No. 61402258), China Postdoctoral Science Foundation (No. 2015M571739), Open Project of State Key Laboratory for Novel Software Technology (No. KFKT2016B22).

8. REFERENCES

[1] Lemos, A. L., Daniel, F., and Benatallah, B. 2016. Web service composition: a survey of techniques and tools. *ACM Compt. Surv.* 48, 3 (Feb. 2016), 33. DOI = 10.1145/2831270.

[2] Qi, L., Dou, W., Zhou, Y., Yu, J., and Hu, C. 2015. A context-aware service evaluation approach over big data for cloud applications. *IEEE T. Cloud Comput.* (Dec. 2015). DOI = 10.1109/TCC.2015.2511764.

[3] Xu, J., Zheng, Z., and Lyu, M. R. 2016. Web service personalized quality of service prediction via reputation-based matrix factorization. *IEEE T. Reliab.* 65, 1 (Mar. 2016), 28-37. DOI = 10.1109/TR.2015.2464075.

[4] Wang, X., Zhu, J., Zheng, Z., Song, W., Shen, Y., and Lyu, M. R. 2016. A spatial-temporal QoS prediction approach for time-aware web service recommendation. *ACM Trans. Web* 10, 1 (Jan. 2016), 1-25. DOI: http://dx.doi.org/10.1145/2801164.

[5] Rong, Y., Wen, X., and Cheng, H. 2014. A Monte Carlo algorithm for cold start recommendation. In *Proceedings of International Conference on World Wide Web* (The Seoul, The Korea, April 07 - 11, 2014). ACM, New York, NY, 327-336. DOI = 10.1145/2566486.2567978.

[6] Su X. and Khoshgoftaar, T. M. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* 2009 (Jan. 2009), 1-19. DOI = 10.1155/2009/421425.

[7] Ni, Y., Fan, Y., Tan, W., Huang, K., and Bi, J. 2016. NCSR: negative-connection-aware service recommendation for large sparse service network. *IEEE T. Autom. Sci. Eng.* 13, 2 (Apr. 2016), 579-590. DOI = 10.1109/TASE.2015.2466691.

[8] Qi, L., Zhang, X., Wen, Y., and Zhou, Y. 2015. A Social Balance Theory-based service recommendation approach. *In Proceedings of the Asia-Pacific Services Computing Conference* (The Bangkok, The Thailand, December 07 - 09, Springer, Berlin, BL, 48-60. DOI = 10.1007/978-3-319-26979-5_4.

[9] Sarwar B., Karypis G., Konstan J., et al. 2001. Item-based collaborative filtering recommendation algorithms. *In Proceedings of the 10th international conference on World Wide Web* (The Hongkong, The China, May 01 - 05, 2001). ACM, New York, NY, 285-295. DOI = 10.1145/371920.372071.

[10] Cartwright D., Harary F. 1956. Structural balance: a generalization of Heider's theory. *Psychol. Rev.* 63, 5 (Sep. 1956), 277-293. DOI = http://dx.doi.org/10.1037/h0046049.

[11] Ma, Y., Wang, S., Hung, P. C., Hsu, C. H., Sun, Q., and Yang, F. 2015. A highly accurate prediction algorithm for unknown web service QoS values. *IEEE T. Serv. Comput.* 9, 4 (Feb. 2015), 511-523. DOI = 10.1109/TSC.2015.2407877.

[12] http://www.grouplens.org/datasets/movielens/ (accessed on 2016-02-04).

[13] Zheng, Z., Ma, H., Lyu, M. R., and King, I. 2011. Qos-aware web service recommendation by collaborative filtering. *IEEE T. Serv. Comput.* 4, 2 (May 2011), 140-152. DOI = 10.1109/TSC.2010.52.

[14] Wu, C., Qiu, W., Zheng, Z., Wang, X., and Yang, X. QoS prediction of web services based on two-phase k-means clustering. *In Proceedings of IEEE International Conference on Web Services* (The Seoul, The Korea, June 27 – July 02, 2015). IEEE Press, New York, NY, 161-168. DOI = 10.1109/ICWS.2015.31.

[15] Qi, L., Xu, X., Zhang, X., Dou, W., Hu, C., Zhou, Y., Yu, J. Structural Balance Theory-based E-commerce recommendation over big rating data. *IEEE T. Big Data* (2016). DOI : 10.1109/TBDATA.2016.2602849.

[16] Duan, Y., Fu, G., Zhou, N., Sun, X., Narendra, N. C., and Hu, B. Everything as a Service (XaaS) on the cloud: origins, current and future Trends. *In Proceedings of IEEE International Conference on Cloud Computing* (The New York, The USA, June 27 – July 02, 2015) IEEE Press, New York, NY, 621-628. DOI: 10.1109/CLOUD.2015.88.

[17] Qi, L., Dou, W., and Chen, J. Weighted PCA-based service selection method for multimedia services in cloud environment. *Computing*, 98, 1 (Jan. 2016), 195-214. DOI: 10.1007/s00607-014-0413-x.

[18] Liu, L., Lecue, F., and Mehandjiev, N. 2013. Semantic content-based recommendation of software services using context. *ACM Trans. Web* 7, 3 (Sep. 2016), 17. DOI= 10.1145/2516633.2516639.

[19] Yao, L., Sheng, Q. Z., Ngu, A. H., Yu, J., and Segev, A. 2015. Unified collaborative and content-based web service recommendation. *IEEE T. Serv. Comput.* 8, 3 (Jun. 2015), 453-466. DOI = 10.1109/TSC.2014.2355842.

[20] Lin, S. Y., Lai, C. H., Wu, C. H., and Lo, C. C. 2014. A trustworthy QoS-based collaborative filtering approach for web service discovery. *J. Syst. Software*, 93 (Jul. 2014), 217-228. DOI = http://dx.doi.org/10.1016/j.jss.2014.01.036.

[21] Cao, J., Wu, Z., Wang, Y., and Zhuang, Y. 2013. Hybrid Collaborative Filtering algorithm for bidirectional Web service recommendation. *Knowl. Inf. Syst.* 36, 3 (Sep. 2013), 607-627. DOI = 10.1007/s10115-012-0562-1.

[22] Tang, M., Xu, Y., Liu, J., Zheng, Z., and Liu, X. 2014. Combining global and local trust for service recommendation. In *Proceedings of International Conference on Web Services* (The Alaska, The USA, June 27 - July 02, 2014), IEEE Press, New York, NY, 305-312). DOI = 10.1109/ICWS.2014.52.

[23] Jiang, D., Guo, X., Gao, Y., Liu, J., Li, H., and Cheng, J. 2014. Locations recommendation based on check-in data from location-based social network. In *Proceedings of International Conference on Geoinformatics* (The Kaohsiung, The Taiwan, June 25–27, 2014). IEEE Press, New York, NY, 1-4. DOI = 10.1109/GEOINFORMATICS.2014.6950814.