Context-Aware Cloud Service Selection Model for Mobile Cloud Computing Environments

Xu Wu[1, 2]

[1]School of Computer, Electronics and Information, Guangxi University, Nanning, China

[2]School of Engineering and Technology, Indiana University–Purdue University Indianapolis,
Indianapolis, USA
xrdz2006@163.com

Abstract

Mobile cloud computing (MCC) has attracted extensive attention in recent years. With the prevalence of MCC, how to select trustworthy and high quality mobile cloud services becomes one of the most urgent problems. Therefore, this paper focuses on the trustworthy service selection and recommendation in mobile cloud computing environments. We propose a novel service selection and recommendation model (SSRM), where user similarity is calculated based on user context information and interest. In addition, the relational degree among services is calculated based on PropFlow algorithm and we utilize it to improving the accuracy of ranking results. SSRM supports a personalized and trusted selection of cloud services through taking into account mobile user's trust expectation. Simulation experiments are conducted on ns3 simulator to study the prediction performance of SSRM compared with other two traditional approaches. The experimental results show the effectiveness of SSRM.

Keywords: *mobile cloud computing, Quality of Service, service selection, collaborative filtering*

1. Introduction

Mobile Cloud Computing (MCC) is a new computing model where Cloud Computing (CC) is integrated into mobile computing environments. The new computing model breaks through the resource limitation of mobile terminals by moving data processing and storage from mobile device to cloud service platforms via wireless networks. Rich mobile applications can be easily created and accessed just based on web browser on the mobile devices [1]. The architecture of MCC is shown is in the figure 1. Internet content providers in figure 1 put video information, games and news resources in appropriate data centers in order to provide users more rich and efficient content services. Mobile users use the wireless connections to access the data centers of public clouds over the Internet. Data centers of public clouds are distributed in different locations and provide users with elastic and scalable computing or storage services. In addition, some mobile users with the demand for higher privacy protection, lower network latency and energy consumption can connect to cloudlets via local area networks. A cloudlet located at the edge of the Internet is a mobility-enhanced small-scale cloud datacenter, which extends cloud computing infrastructure [2].

MCC is derived from Cloud Computing, thus it inherits the advantages of Cloud Computing such as dynamic development of mobile applications, resource scalability, multi-user sharing and multi-service integration, et.al. But, quite apart from that, it has also some problems including resource limits, weak battery strength, user mobility and low network coverage. With the prevalence of MCC, more and more internet users from mobile devices use cloud computing services through wireless interface (e.g. GPRS/3G/WiFi). As shown in the figure 1, there exits abundant functionally similar cloud services provided by different cloud service providers. Therefore, how to select appropriate cloud services for mobile users becomes one of the most urgent problems.

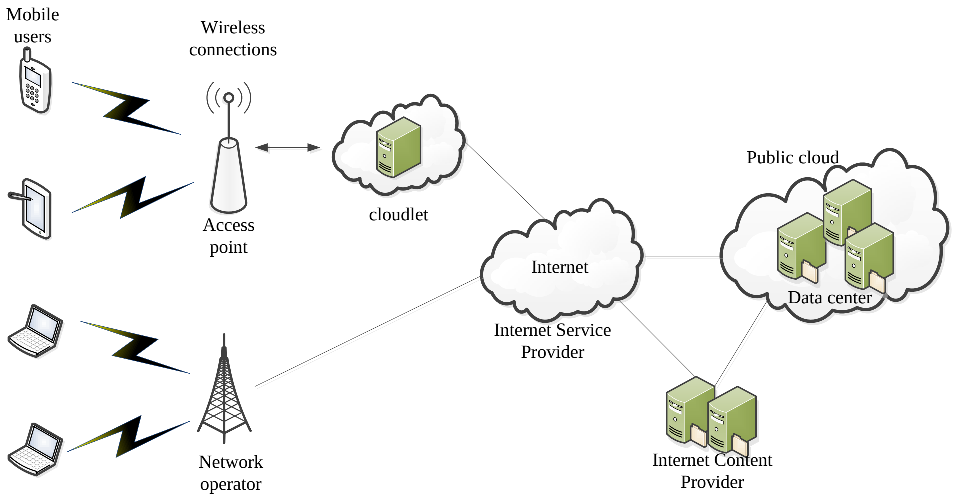


Figure 1 Architecture of Mobile Cloud Computing

To address the problem of cloud service selection and recommendation, researchers have done many notable works in cloud computing literature [3-7]. The most of existing methods mainly depend on ranking Quality-of-Service (QoS) to select an optimal cloud service from a set of functionally equivalent cloud services. Quality-of-Service (QoS) usually describes non-functional performance of cloud services. QoS values of cloud services provide valuable information to assist decision making [3].

The challenge here is to recommend an optimal mobile cloud service based on dynamic QoS properties. In real-world applications, QoS in MCC is referring to dynamic QoS properties (packet loss ratio, end-to-end throughput, delay etc.), which are affected by the user context information [23]. Context information includes location, time, resource ability (processing power, memory or battery capacity), bandwidth, status (online or offline), etc. For example, limited bandwidth and resource ability can make the values of QoS properties degrade. Naturally, consumers hope to select the most trustworthy cloud service among abundant candidates by considering the dynamic context information over multiple time periods. However, the works done for cloud computing rarely focus on investigating the influence of user context information on service selection. User mobility in MCC can lead to the dynamic changes of user context information. The selection results of mobile user may vary according to the changes of user context information. Therefore, user context information plays an important role when designing such selection and recommendation algorithms.

Moreover, the existing methods always assume that cloud services are independent and ignore the fact that the relation among cloud services has important effect on the accuracy of recommendation results. The related services will be probably selected by the same user based on the intuition [8]. For example, in tourism service scenario the users purchasing airplane ticket service probably prefer to purchase car rental service in future.

To attack this critical challenge, the algorithm for MCC is expected not only to satisfy the required service level by user, but also to adapt to dynamic changes of user context information. In the paper, we make use of user-based collaborative filtering method to propose a novel context-aware service selection and recommendation model (SSRM), which combines the user context information, user interest, historical service usage experiences (QoS value or customer rating) of cloud services and the relation among cloud services to rank the cloud services. Collaborative filtering technologies [22] in the field of online recommendation system offer us a strong theoretical foundation to deal with the service selection and recommendation problem. Recently, they have been successfully applied to predict the missing QoS value in cloud computing. Collaborative filtering technologies are categorized as user-based collaborative filtering approaches, item-based collaborative filtering approaches and their fusion approaches. In addition, SSRM model only select trusted mobile cloud service as candidates for mobile users. Different users have different trust requirement, the personalized service selection and recommendation method is thus required by different users [3]. To support personalized selection of cloud services, SSRM takes into account mobile user's trust expectation through using a trust threshold.

Within the SSRM model, there are a few modules. Figure 2 shows the system architecture of SSRM model, which supports personalized selection and recommendation of cloud services.

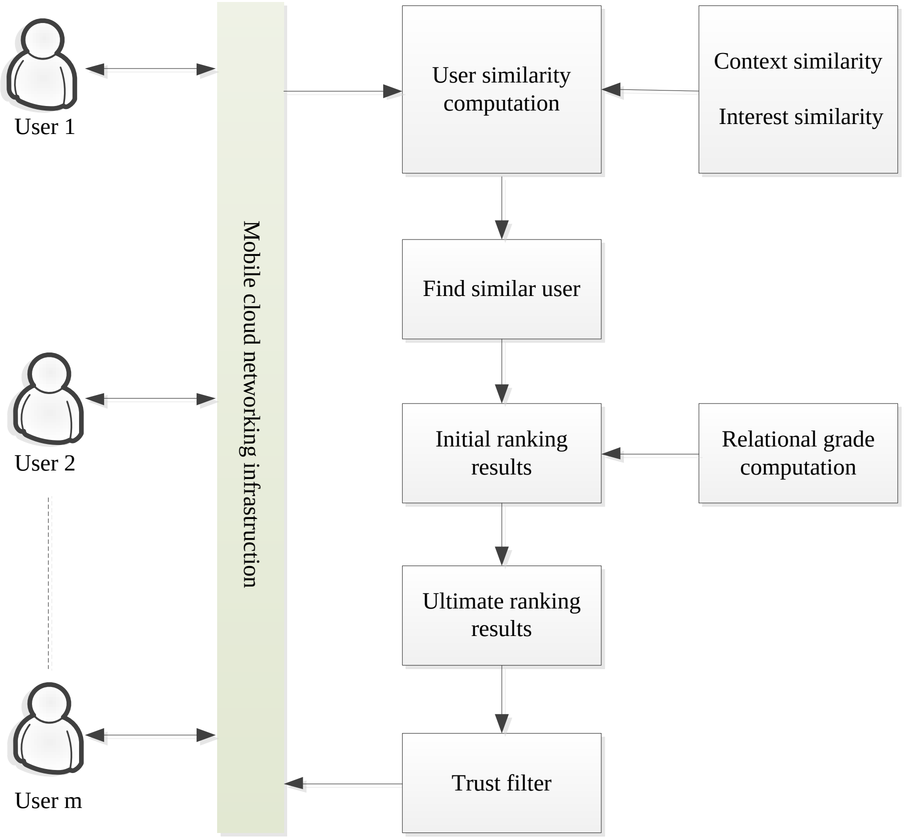


Figure 2 Architecture of SSRM model

A user requesting service selection and recommendation is called target user in the model. A user providing historical service usage experiences is called training user. Our aim is to get the service ranking results from SSRM model by predicting the missing service usage experiences for target user. SSRM involves the following key steps:

Step A: A target user requests service selection and recommendation;

Step B: User similarities between target user and training users are calculated based on context similarity and user interest similarity.

Step C: Based on the similar values, a set of similar users are identified.

Step D: The basic service ranking results are obtained by taking advantages of the past service usage experiences of similar users.

Step E: Relational degree among cloud services is evaluated. Through utilizing relational degree to improving the accuracy of ranking results, we succeed in obtaining the ultimate service ranking results.

Step F: SSRM only selects trusted mobile cloud services as candidates based on trust filter module.

Step G: The ranking prediction results are provided to the active user.

This paper targets the research task of accurately making QoS ranking prediction, with consideration of user context information and the relation among cloud services, according to the requirements of different application scenarios. This paper makes the following contributions:

1) This paper focuses on the trustworthy service selection and recommendation in mobile cloud computing environments. We propose a novel service selection and recommendation model (SSRM), where similarity is calculated based on user context information and interest.
2) SSRM calculate the relational degree among services based on PropFlow algorithm and utilize relational degree to improving the accuracy of ranking results.
3) SSRM supports a personalized and trusted selection of cloud services through taking into account mobile user’s trust expectation.
4) Simulation experiments are conducted on ns3 simulator to study the prediction performance of SSRM compared with other two traditional approaches: item-based CF approach (IBCF) and user-based CF approach (UBCF). The experimental results show the effectiveness of SSRM.

This paper is organized as follows. Section 2 describes related work. In Section 3, the proposed SSRM model is discussed. Section 4 describes the test scenario and simulation results. Finally, we conclude with a summary of our results and directions for new research in Section 5.

2. Related Work

Mobile cloud computing (MCC) has attracted extensive attention in recent years, since it provides real-time access to real-time information through the applications on mobile devices. The rapid advancements of mobile cloud technology have been the prime reason for significant expansions in this market, where exists abundant functionally similar services provided by cloud vendors including Amazon, Google, Inc., Apple, Inc., Microsoft Corporation, etc. Markets and Markets [14] forecast the global mobile cloud market will grow to over $46.90 Billion by 2019. The exponential growth of mobile cloud market makes it hard for users to select the most suitable mobile cloud service. Therefore, how to select trustworthy and high quality mobile cloud services becomes one of the most urgent problems. A number of works have been carried out on cloud service selection issue including rating-oriented collaborative filtering methods [4, 5], ranking-oriented collaborative filtering methods [3] and some other methods.

Rating-oriented or ranking oriented collaborative filtering methods can produce QoS prediction or service ranking using collaborative filtering technology. Rating-oriented collaborative filtering methods rank the cloud services based on the predicted QoS values. QoS presents the non-functional properties of cloud services including security, availability, throughput and response-time properties, etc. Based on the service QoS measures, various approaches are proposed for service selection. Pan et al. [4] propose a trust-enhanced cloud service selection model based on QoS analysis. In the model, trust is utilized to find similar neighbors and predict the missing QoS values. Ding et al. [5] propose a personalized cloud service selection method to make experience usability and value distribution to measure the service similarity. Rating-oriented collaborative filtering approaches first predict the missing QoS values before making QoS ranking. The target of rating-oriented approaches is to predict QoS values as accurate as possible. However, accurate QoS value prediction may not lead to accurate QoS ranking prediction [3].

Compared with rating-oriented methods, ranking-oriented methods predict the QoS rankings directly. Different from the previous ranking-oriented methods, Zheng et al. [3] propose a comprehensive study of how to provide accurate QoS ranking for cloud services. Although ranking-oriented methods can be used to make optimal cloud service selection from a set of functionally equivalent service candidates, these methods ignore the changes of QoS. QoS in MCC might vary largely, even for the same type of mobile cloud services. QoS in MCC are affected by the user context information. The rating-oriented collaborative filtering methods and ranking-oriented collaborative filtering methods both can help user to predict missing QoS value, but they did not consider the dynamic QoS properties in MCC.

Other types of service selection approaches are also been widely examined. The most employed approaches include multicriteria decision analysis-based service selection [7, 18, 19 and 26], reputation-aware service selection [16], adaptive learning mechanism-based service selection [17, 23], economic theoretical model-based service selection [15, 20], service level agreement-based service ranking [21], visualization framework for service selection [24] and trust evaluation middleware for cloud service selection [25]. Though these approaches can efficiently measure service quality, the implementation of some approaches is time-consuming and costly.

Ma et al. [7] propose a time-aware service selection approach by using interval neutrosophic set. In the paper, the strategy for selecting trustworthy services from an abundant field of candidates involves formulating the problem of time-aware service selection with tradeoffs between performance – costs and potential risk as a multi-criterion decision-making (MCDM) problem that creates a ranked services list using interval neutrosophic set (INS) theory. The experimental results demonstrate that the proposed approach can work effectively in both the risk-sensitive service selection mode and the performance-cost-sensitive service selection mode, but the approach ignores the changes of user context information in MCC environments.

Whaiduzzaman et al. [18] identify and synthesize several multicriteria decision analysis (MCDA) techniques and provide a comprehensive analysis of this technology for general readers. In addition, this paper presents a taxonomy derived from a survey of the current literature. The results show that MCDA techniques are indeed effective and can be used for cloud service selection but that different techniques do not select the same service.

This paper [26] presents a cloud service selection methodology that utilizes quality of service history of cloud services over different time periods and performs parallel multi-criteria decision analysis to rank all cloud services in each time period in accordance with user preferences before aggregating the results to determine the overall rank of all the available options for cloud service selection. This methodology assists the cloud service user to select the best possible available service according to the requirements. The main disadvantages are that the framework proposed in this paper deals with service selection in the pre-interaction period only. Work on post-interaction service migration decisions is needed, and several other important factors such as the cost of migration in terms of service disruption and data transfer, etc. also need to be included in the decision making process.

Fan et al. [16] propose a multi-dimensional trust-aware cloud service selection mechanism based on evidential reasoning approach which aggregates multi-dimensional trust feedback ratings to form the reputation values of the cloud service providers. Though the experimental results show that this approach is effective in cloud systems, but it can be only used to recommend an optimal cloud service which satisfies the key QoS requirements for users.

Wang et al [17] propose a dynamic cloud service selection method by using an adaptive learning mechanism, which involves incentive, forgetting and degenerate functions that can realize the self-adaptive regulation for optimizing next service selection according to the status of current service selection. To assist users to efficiently select their preferred cloud services, a cloud service selection model adopting the cloud service brokers is given in the paper. However, in the paper the brokerage scheme on cloud service selection typically assumes that brokers are completely trusted, and do not provide any guarantee over the correctness of the service recommendations. It is then possible for a compromised or dishonest broker to easily take advantage of the limited capabilities of the clients and provide incorrect or incomplete responses.

Do et al. [15] propose a dynamic service selection method which provides a price game in heterogeneous cloud market. The paper studies price competition in a heterogeneous cloud market, where users can identify benefit and cost of cloud service application and choose the best one through analyzing market-relevant factors. But this paper only considers one service. In real-world applications, there are many cloud services in the practical cloud market. Additionally, this paper has not considered service-level agreement issues that are also important for cloud users.

The visualization framework for service selection [24] takes into cognizance the set of cloud services that matches a user’s request and based on QoS attributes, users can interact with the results via bubble graph visualization to compare and contrast the search results to ascertain the best alternative. Although the result from the experiments shows that visualization framework simplifies decision making, the use of bubble graph introduces additional complexity for the user when making a suitable service selection.

The authors [25] discuss the method of enhancing service trust evaluation and propose a trustworthy selection framework for cloud service selection, named TRUSS. Aiming at developing an effective trust evaluation middleware for TRUSS, this paper proposes an integrated trust evaluation method via combining objective trust assessment and subjective trust assessment. Simulation-based experiments validated the performance of the proposed method, but the method assumes that the majority of service users are honest and a dishonest user gives more unfair ratings than fair ratings. This assumption is unrealistic.

Cloud computing and mobile cloud computing are quite different. The mobile cloud pays attention to services available through mobile network operators (MNOs like Verizon and AT&T). While a great number of researchers have focused on the service selection and recommendation in cloud computing, little attention has been devoted to trustworthy service selection in mobile cloud computing. Different from these existing approaches, our work focuses on how to select trustworthy and high quality cloud services for mobile user in MCC, which is an urgently-required research problem. Thus, we propose a context-aware cloud service selection method for mobile cloud computing environments, which considers user context information and interest in order to calculate user similarities and utilize relational degree among cloud services to improve the accuracy of service ranking results.

3. SSRM Model

In this section, we present a detailed discussion of the proposed novel service selection and recommendation model (SSRM). Firstly, user similarities are calculated and the basic personalized service ranking results are obtained in section 3.1. Secondly, relational degree is calculated and used to improve the accuracy of ranking results in section 3.2.

3.1. User Similarity Computation

Given m users and n mobile cloud services, the user-service matrix (USM) for predicting the missing service usage experiences is denoted as

$$\begin{bmatrix} r_{u_1,mcs_1} & \cdots & r_{u_1,mcs_n} \\ \vdots & \ddots & \vdots \\ r_{u_m,mcs_1} & \cdots & r_{u_m,mcs_n} \end{bmatrix} \quad (1)$$

where r_{u_m,mcs_n} expresses historical service usage experiences (QoS value or customer rating) of mobile cloud service mcs_n made by user u_m. "$r_{m,n} = null$" states that u_m didn't invoke mcs_n yet. In order to get the ranking results, we firstly calculate user similarity. The user similarity is measured with the following Equation:

$$sim_U(u_x, u_y) = \lambda * sim_C(u_x, u_y) + (1-\lambda) sim_I(u_x, u_y) \quad (2)$$

where u_x and u_y denote two users. $sim_C(u_x, u_y)$ and $sim_I(u_x, u_y)$ respectively express context similarity and user interest similarity between u_x and u_y. λ is defined to determine how much similarity measure relies on context and interest. λ is in the interval [0,1].

3.1.1 Context Similarity Computation

Cloud service selection of a user in a context (e.g., a laptop with enough battery) may be much different from that of another user in a different context (e.g., a smart phone without enough battery). Hence, cloud service selection can be significantly affected by user context. To evaluate the context similarities between target user and training user, we provide the definition of context. The definition from [9] is referenced by us. Let r_{u_x,mcs_j} be usage experience of cloud service mcs_j made by user u_x.

Definition 1 (Context): Context C_{u_x,mcs_j} is any information that can be used to characterize the situation where usage experience r_{u_x,mcs_j} of cloud service mcs_j is made by user u_x.

Context information has different properties which include location, resource ability, status, etc. Let v be the type of context property of C_{u_x,mcs_j}.

The context C_{u_x,mcs_j} is denoted as:

$$C_{u_x,mcs_j} = (C^{nor}{}_{u_x,mcs_j}(1), C^{nor}{}_{u_x,mcs_j}(2), \ldots, C^{nor}{}_{u_x,mcs_j}(v)) \tag{3}$$

where $C^{nor}{}_{u_x,mcs_j}(v)$ is the normalized property value.

Pearson Correlation Coefficient (PCC) has been successfully employed to obtain the numerical distance between different users for similarity calculation [4-5]. Let u_x and u_y be two users, then PCC is applied to calculate the context similarity between u_x and u_y over service mcs_j by

$$sim_{C_{mcs_j}}(u_x, u_y) = \frac{\sum_{s=1}^{v} (C^{nor}{}_{u_x,mcs_j}(s) - \overline{C}^{nor}_{u_x,mcs_j}(s))(C^{nor}{}_{u_y,mcs_j}(s) - \overline{C}^{nor}_{u_y,mcs_j}(s))}{\sqrt{\sum_{s=1}^{v} (C^{nor}{}_{u_x,mcs_j}(s) - \overline{C}^{nor}_{u_x,mcs_j}(s))^2} \sqrt{\sum_{s=1}^{v} (C^{nor}{}_{u_y,mcs_j}(s) - \overline{C}^{nor}_{u_y,mcs_j}(s))^2}} \tag{4}$$

where $sim_{C_{mcs_j}}(u_x, u_y)$ expresses the context similarity between u_x and u_y and it is in the interval of [-1, 1], and $\overline{C}_{u_x,mcs_j}(s)$ and $\overline{C}_{u_y,mcs_j}(s)$ stand for the average values of context.

3.1.2 Interest Similarity Computation

In mobile cloud computing environments, each user lives in a large network of friends which is called social network. It has been reported that user interests' similarity can be leveraged to support services and products recommendation [10]. A user prefers to choose the items recommended by other user with similar interest in a social network. Exploring those users of a high interest similarity with the existing clients could efficiently enlarge client groups for cloud service providers [10].

Here, we use cosine similarity to compute interest similarity, similar to [10]. Interest similarity between users u_x and u_y is then defined as the cosine distance between their respective cloud service invocation sets:

$$sim_I(u_x, u_y) = \frac{\left\| I_{u_x} \cap I_{u_y} \right\|}{\left\| I_{u_x} \right\| \cdot \left\| I_{u_y} \right\|} \tag{5}$$

where $\left\| I_{u_x} \right\| = \sqrt{l_{u_x}}$ (l_{u_x} is the number of mobile cloud service which have been invoked by u_x) and $\left\| I_{u_x} \cap I_{u_y} \right\|$ is the number of mobile cloud service which been invoked co-invoked by u_x and u_y. If $l_{u_x} = 0$ or $l_{u_y} = 0$, $sim_I(u_x, u_y)$ is undefined.

Cloud service invocation is an important factor to determine the interest of users [5]. For example, user u_x has invoked mobile cloud service mcs_1, mcs_3, mcs_6 and mcs_7, u_y has invoked mcs_6 and mcs_7 and u_z has invoked mcs_1 and mcs_2. Though u_x, u_y and u_z don't know each other in real world, u_x and u_y have a high interest similarity than u_x and u_z, as they both invoked mcs_6 and mcs_7 [4].

3.1.3 Getting Initial Ranking Results

We first predict the missing value r_{u_x,mcs_j} of mobile cloud service mcs_j made by user u_m before making initial ranking results. For every mobile cloud service, only the Top-*k* similar training user is selected to making missing value prediction. Based on the $sim_U(u_x,u_y)$ values, a set of the Top-*k* similar training user $N_{C_{mcs_j}}(u_x)$ over service mcs_j is identified for target user u_x by:

$$N_{C_{mcs_j}}(u_x) = \left\{ u_y \middle| u_y \in TS_{u_x}, sim_U(u_x, u_y) > 0, u_x \neq u_y \right\} \tag{6}$$

where TS_{u_x} is a set of the Top-*k* similar training user to target user u_x and $sim_U(u_x,u_y) > 0$ excludes the dissimilar user with negative similar values. The value of $sim_U(u_x,u_y)$ in Equation (6) is calculated by Equation (2). The missing value r_{u_x,mcs_j} is predicted as follows:

$$r_{u_x,mcs_j} = \bar{r}_{u_x} + \frac{\sum_{u_y \in N_{C_{mcs_j}}(u_x)} (r_{u_y,mcs_j} - \bar{r}_{u_y}) sim_U(u_x, u_y)}{\sum_{u_y \in N_{C_{mcs_j}}(u_x)} sim_U(u_x, u_y)} \tag{7}$$

where $\bar{r}_{u_x}$ and $\bar{r}_{u_y}$ is the average value of service usage experiences (QoS value or customer rating) of mobile cloud service made by user u_x and u_y, respectively . As different QoS properties have different dimensions and range of values, we first ensure predicted missing values in the range of [0, 1]. In [5], QoS properties are classified into two categories: “cost” and “benefit”. For “cost” property (response-time), the lower its value is, the greater possibility that a user would choose it becomes. In SSRM, all QoS properties are considered as “benefit” attribute. Normalized missing value Q_{u_x,mcs_j} is computed as followed:

$$Q_{u_x,mcs_j} = \begin{cases} \dfrac{r_{u_x,mcs_j} - \min(r_{u_x})}{\max(r_{u_x}) - \min(r_{u_x})}, & r_{u_x,mcs_j} \in "benefit" \\ \dfrac{\max(r_{u_x}) - r_{u_x,mcs_j}}{\max(r_{u_x}) - \min(r_{u_x})}, & r_{u_x,mcs_j} \in "cost" \end{cases} \tag{8}$$

where $\min(r_{u_x})$ and $\max(r_{u_x})$ denote the minimum and maximum QoS property value for user u_x and they are subject to the following constrains:

$$\begin{cases} \min(r_{u_x}) = \min(r_{u_x,mcs_j} \mid j = 1,\ldots,n) \\ \max(r_{u_x}) = \max(r_{u_x,mcs_j} \mid j = 1,\ldots,n) \end{cases} \tag{9}$$

Q_{u_x,mcs_j} is in the range of [0,1]. The larger its value is, the more possibility that user u_x would be satisfied with the service becomes. Finally, the cloud services are ranked for u_x in the order of decreasing Q_{u_x,mcs_j} values for cloud service selection, similar to [4]. The basic ranking algorithm is shown in figure 3. The basic ranking algorithm is similar to [3]. Compared with [3] , [4] and [5], in our ranking algorithm the user similarity is measured based on user context information and user interest. In addition, the basic ranking algorithm is enhanced in section 3.2. In the enhanced ranking algorithm, we calculate the relational degree among cloud services and utilize it to improving the accuracy of ranking results.

The basic algorithm ranks the employed mobile cloud service in E based on the observed historical service usage experiences. The ranking results are stored in $R_e(t)$ which is in the interval $[1,|E|]$. A smaller value shows a higher rank. t expresses a mobile cloud service. For every cloud service mcs_j, normalized value Q_{u_x,mcs_j} is calculated. Services are ranked from high to low by picking up the service t that has the maximum Q_{u_x,mcs_j} value.

```
Input:
an employed service set $E$;
a full service set $I$;
an employed similar training user set $N_{C_{mcs_j}}(u_x)$;
observed service usage experience values $r_{u_y,mcs_j}(u_y \in N_{C_{mcs_j}}(u_x))$
Out:
a service ranking $\hat{R}$
$F = E$;
while $F \neq \emptyset$ do
    $t = \text{argmax}_{mcs_j \in F} Q_{u_x,mcs_j}$;
    $R_e(t) = |E| - |F| + 1$;
    $F = F - \{t\}$;
end
foreach $mcs_j \in I$ do
    $Q_{u_x,mcs_j} = \begin{cases} \frac{r_{u_x,mcs_j} - \min(r_{u_x})}{\max(r_{u_x}) - \min(r_{u_x})}, & r_{u_x,mcs_j} \in "benefit" \\ \frac{\max(r_{u_x}) - r_{u_x,mcs_j}}{\max(r_{u_x}) - \min(r_{u_x})}, & r_{u_x,mcs_j} \in "\cos t" \end{cases}$
end
    $n = |I|$;
    while $I \neq \emptyset$ do
        $t = \arg\max_{mcs_j \in I} Q_{u_x,mcs_j}$
        $\hat{R}(t) = n - |I| + 1$;
        $I = I - \{t\}$;
    end
end
```

Figure 3 Basic ranking algorithm

3.2. Improving Accuracy of Ranking Results

Through calculating the similar relation among users, the missing value r_{u_x,mcs_j} is successfully predicted in the above section 3.1. However, the relation among cloud services is no doubt an important evaluation factor of missing value prediction. The related services will be probably selected by the same user based on the intuition [8]. Therefore, we calculate the relational degree and utilize it to improving the accuracy of missing value r_{u_x,mcs_j} computation in the section. The relational degree between service mcs_j and service mcs_i is denoted as d_{ji} . In the paper, PropFlow [11] is used to calculate d_{ji} . PropFlow algorithm computes the information flow between services, where a larger value indicates tighter relation. In order to calculating d_{ji} , we firstly describe the formal definition of service relation graph.

Definition 2 (Service Relation Graph). Given a set of service MCS and a totally ordered domain of weights W, a service relation graph is a weighted undirected graph $G = (MCS, E)$. The edge (*mcs*$_i$; *mcs*$_j$; *w*$_{ij}$) in the set $E \in U \times U \times W$ encodes the link weight w_{ij} ($w_{ij} \geq 0$) between service *mcs*$_i$ and service *mcs*$_j$. An edge depicts the direct link relationship between *mcs*$_i$ and *mcs*$_j$. As G is undirected, $w_{ij} = w_{ji}$.

Equation (10) shows how to calculate d_{ji} .

$$d_{ji} = Input_j * \frac{w_{ji}}{\sum_{k \in N(j)} w_{jk}} \tag{10}$$

where initial input is regarded as 1, $N(j)$ is the set of neighbor of *mcs*$_j$.

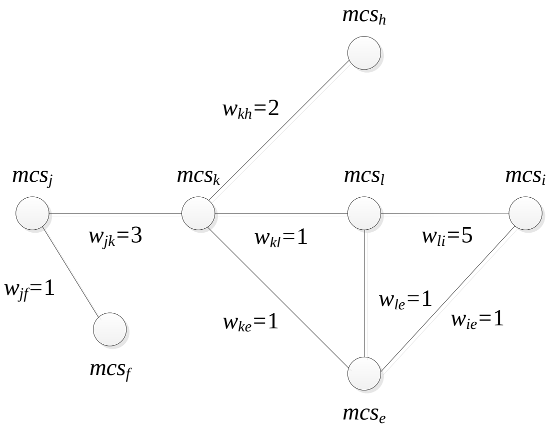


Figure 4 An example of service relation graph

Figure 4 shows an example of relation degree computation, where there are six kinds of mobile cloud services. *mcs*$_i$ and *mcs*$_j$ are indirectly linked. We assume *mcs*$_j$ is starting node.

Four paths can reach *mcs*$_i$ from *mcs*$_j$, but only two paths are considered for computation based on PropFlow algorithm. They are $mcs_j \rightarrow mcs_k \rightarrow mcs_l \rightarrow mcs_i$ and $mcs_j \rightarrow mcs_k \rightarrow mcs_e \rightarrow mcs_i$.

First, we compute d_{jk} . The sum of link weights of *mcs*$_j$ and its neighbor is 4. d_{jk} is computed as;

$$d_{jk} = 1 * \frac{3}{(1+3)} = 1 * \frac{3}{4} = \frac{3}{4} \tag{11}$$

d_{kl} is computed as;

$$d_{kl} = d_{jk} * \frac{1}{(1+1+2)} = \frac{3}{4} * \frac{1}{4} = \frac{3}{16} \tag{12}$$

With the same method, d_{ji} is computed as;

$$d_{ji} = d_{li} + d_{ei} = d_{kl} * \frac{5}{5} + d_{ke} * \frac{1}{1} = \frac{3}{8} \tag{13}$$

The more details of calculating d_{ji} are shown in [12].

Next, we assume service *mcs*$_i$ is invoked recently by user u_x. d_{ji} is incorporated into the Equation (7), and then, the missing value r_{u_x,mcs_j} is computed with Equation (14) in basic ranking algorithm. Finally, the ultimate ranking results are got.

$$r_{u_x,mcs_j} = (\overline{r}_{u_x} + \frac{\sum_{u_y \in N_{C_{mcs_j}}(u_x)} (r_{u_y,mcs_j} - \overline{r}_{u_y}) sim_U(u_x, u_y)}{\sum_{u_y \in N_{C_{mcs_j}}(u_x)} sim_U(u_x, u_y)})(1 + d_{ji}) \tag{14}$$

In SSRM model, we only select trusted mobile cloud service as candidates. Therefore, a set of trusted candidates are identified for u_x by:

$$MCS^x = \left\{ mcs_j \middle| trust_{mcs_j} > \mu_x, \ j = 1, \ldots, N \right\} \tag{15}$$

where $trust_{mcs_j}$ denotes the trustworthiness of mobile cloud service *mcs*$_j$, μ_x expresses the trust threshold decided by user u_x. Many existing methods [4-6] can be applied to compute $trust_{mcs_j}$. Here, how to compute $trust_{mcs_j}$ is not be discussed. We omit the details for brevity.

4. Experimental Study

In this section, in order to evaluate the effectiveness of SSRM model, a series of test scenarios are developed. To study the prediction performance, we compare SSRM with other two traditional approaches: item-based CF approach (IBCF) and user-based CF approach (UBCF). SSRM, IBCF and UBCF are both rating-oriented methods, which rank the cloud services based on the predicted QoS values. Since there is no suitable real data supporting mobile cloud computing simulation, we use ns3 simulator to generate the experimental dataset. Firstly, we generated cloud service invocation codes by Axis2 [27], a Java-based open source package for cloud services. Then we simulates mobile cloud service on ns3 [13] based on the invocation codes. On ns3, the creation of servers, mobile users and service model was easier than with other network simulators. In the simulating process of dataset, the real-world web service QoS dataset from WS-DREAM team [3] is referenced by us. The detailed real-world QoS values are publicly released online [28], which makes our experimental evaluations reproducible. We simulate 100 mobile users, 25 servers and 300 services. The value of link weight between two services is randomly generated, which is in the range of [1, 5]. In order to conduct our experiments realistically, the changes of user context information are simulated. Three types of context are considered including bandwidth, memory and battery capacity. The bandwidth value of a mobile user is in the range of 4M-8M, The values of memory and battery capacity will declines over time. These values are in the range of 100000000-2100000000MB and 0.5-2KVA.

We normalize the context information by:

$$C^{nor}_{u_x,mcs_j}(v) = \frac{a\tan(C_{u_x,mcs_j}(v)) * 2}{\pi} \tag{16}$$

where $C^{nor}_{u_x,mcs_j}(v)$ denotes the normalized context value of user u_x over cloud service mcs_j. Inverse tangent function is applied to normalize the raw data $C_{u_x,mcs_j}(v)$. Then, these normalized property value are used to calculate context similarity with Equation (4). Once the context simulation is finished, the target user u_x will obtain context similarity results for each cloud service.

Similar to [3], QoS properties include throughput and response-time that are in the range of 0–1000 kbps and 0–20 second, respectively. Most of the throughput and response-time values fall between 5–40 kbps and 0.1–0.8 seconds. In order to simply the experimental process, 200 service QoS records are randomly selected and two 100 × 200 user-service matrices are constructed. The two matrices include throughput and response-time, respectively. User-service matrices are separated into two parts: training set (80% historical usage experiences in the matrix) and test set (the remaining 20% usage experiences). Each entry in the matrix is the QoS value (e.g., response-time or throughput) of a mobile cloud service observed by a user. The experiments employ the QoS values of response-time and throughput to rank the services independently. Table 1 shows descriptions of the obtained the practical cloud service QoS values and context information.

Table 1. Mobile Cloud Service QoS Dataset and Context Information Descriptions

Statistics	Value
Num. of mobile cloud service invocations	180,000
Num. of service users	100
Num. of mobile cloud services	200
Minimum response-time value	0.004s
Maximum response-time value	20s
Mean of response-time	1.10s
Standard deviation of response-time	2.26s
Minimum throughput value	0.1kbps
Maximum throughput value	1000kbps
Mean of throughput	25.46kbps
Standard deviation of throughput	48.03kbps
Bandwidth	4M-8M
Memory	100000000MB-2100000000MB
Battery capacity	0.5KVA-2KVA

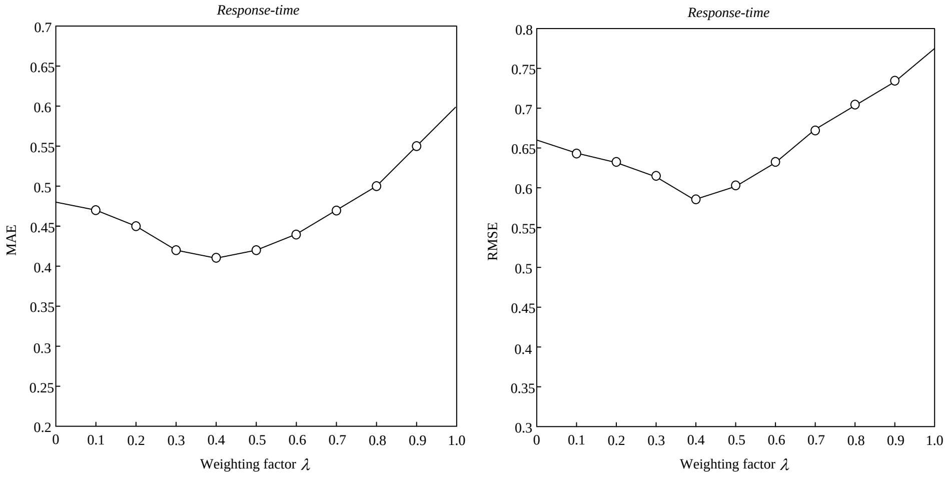
We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as the metric to evaluate prediction performance of the proposed approach in comparison with other approaches. MAE and RMSE are defined as:

$$\begin{cases} MAE = \dfrac{\sum_{u_x,mcs_j} \left| r^{act}_{u_x,mcs_j} - r^{pre}_{u_x,mcs_j} \right|}{l_{pre}} \\ RMSE = \sqrt{\dfrac{\sum_{u_x,mcs_j} (r^{act}_{u_x,mcs_j} - r^{pre}_{u_x,mcs_j})^2}{l_{pre}}} \end{cases} \tag{17}$$

where $r^{act}_{u_x,mcs_j}$ and $r^{pre}_{u_x,mcs_j}$ denote the actual QoS value and predicted value, respectively. l_{pre} is the number of predicted QoS values. Smaller values of MAE and RMSE indicate better results.

4.1. Impact of λ

λ is weighting factor in Equation (2). λ determines how much similarity measure relies on context and interest. In the experiment, we vary the weighting factor λ from 0 to 1 in increment of 0.1. The Top-*k* is set to 5. Matrix density is set to 10%. Matrix density means the percentage of selected QoS entries that are used to predict the missing QoS value. The experimental results are shown in figure 5. As the value of λ increases, MAE firstly decreases and then quickly increases. When λ =0.4, our results have the best value. As shown in figure 5(b), RMSE presents similar trend.



(a) MAE values (b) RMSE values

Figure 5. Impact of weighting factor λ

4.2. Performance Comparison of SSRM, IBCF and UBCF

To compare the performance of SSRM, IBCF and UBCF similarity computation methods, we implement IBCF and UBCF methods.

Item-based CF approach (IBCF): We apply PCC to calculate similarities between users and predicted QoS values based on similar users. The user similarity is calculated by:

$$sim(u_x, u_y) = \frac{\sum_{mcs_j \in mcs_{u_x,u_y}} (r_{u_x,mcs_j} - \overline{r}_{u_x})(r_{u_y,mcs_j} - \overline{r}_{u_y})}{\sqrt{\sum_{mcs_j \in mcs_{u_x,u_y}} (r_{u_x,mcs_j} - \overline{r}_{u_x})^2}\sqrt{\sum_{mcs_j \in mcs_{u_x,u_y}} (r_{u_y,mcs_j} - \overline{r}_{u_y})^2}} \tag{18}$$

where mcs_{u_x,u_y} is the set of mobile cloud services that have been co-invoked by u_x and u_y . $\overline{r}_{u_x}$ and $\overline{r}_{u_y}$ are average QoS values of cloud service invoked by u_x and u_y ,respectively.

User-based CF approach (UBCF). We apply PCC to calculate similarities between services and predicted QoS values based on similar services. The service similarity is calculated by:

$$sim(mcs_j, mcs_i) = \frac{\sum_{u_x \in u_{mcs_j,mcs_i}} (r_{u_x,mcs_j} - \overline{r}_{mcs_j})(r_{u_x,mcs_i} - \overline{r}_{mcs_i})}{\sqrt{\sum_{u_x \in u_{mcs_j,mcs_i}} (r_{u_x,mcs_j} - \overline{r}_{mcs_j})^2}\sqrt{\sum_{u_x \in u_{mcs_j,mcs_i}} (r_{u_x,mcs_i} - \overline{r}_{mcs_i})^2}} \tag{19}$$

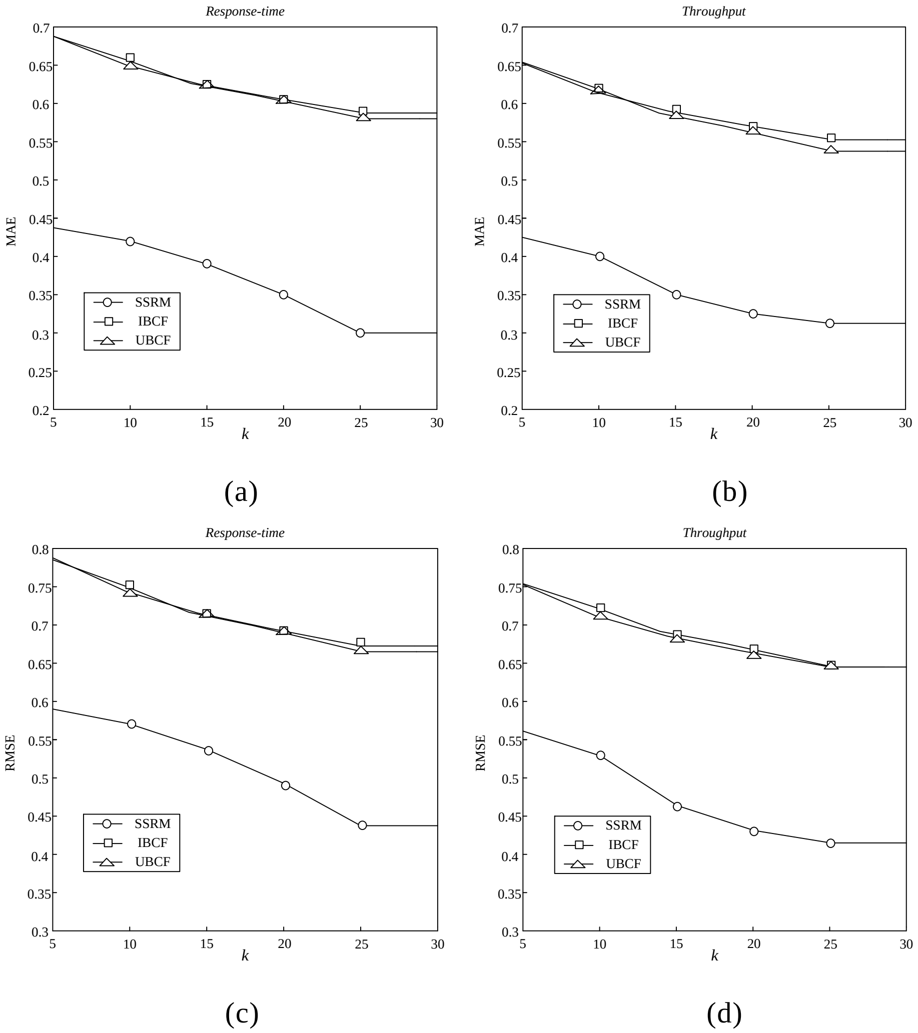
where u_{mcs_j,mcs_i} is the set of users who have both invoked the cloud service mcs_j and mcs_i . $\overline{r}_{mcs_j}$ and $\overline{r}_{mcs_i}$ are average QoS values of cloud service mcs_j and mcs_i made by u_x .

In the experiments, trustworthiness of cloud service has no influence on similarity computation and service decision in the experiments. In order to simplify the experimental process, we consider all cloud services are trustworthy. Weighting factor λ is set to 0.4. We firstly study the effect of neighbor size *k*. Matrix density is set to 10%. We change *k* from 5 to 30 in increment of 5. Figure 6 shows the experimental results for response time and throughput.

As shown in figure 6, the prediction performance of SSRM outperforms other two approaches. The performance of IBCF is similar to UBCF. As the values of *k* increases, better accuracy can be achieved. This indicates that more similar neighbor records can provides more information for missing value prediction. However, when *k* is larger than 25, MAE and RSME fail to drop with the value of *k* increasing. The main reason is the limited number of similar neighbors. The observations also show that RMSE has the similar trend, but with larger fluctuations.

Next, we investigate the impact of matrix density. The matrix density varies from 10% to 50% in increment of 10%. Similar user size *k* is set to 5 in this experiment. Figure 7 shows the experimental results for response time and throughput.

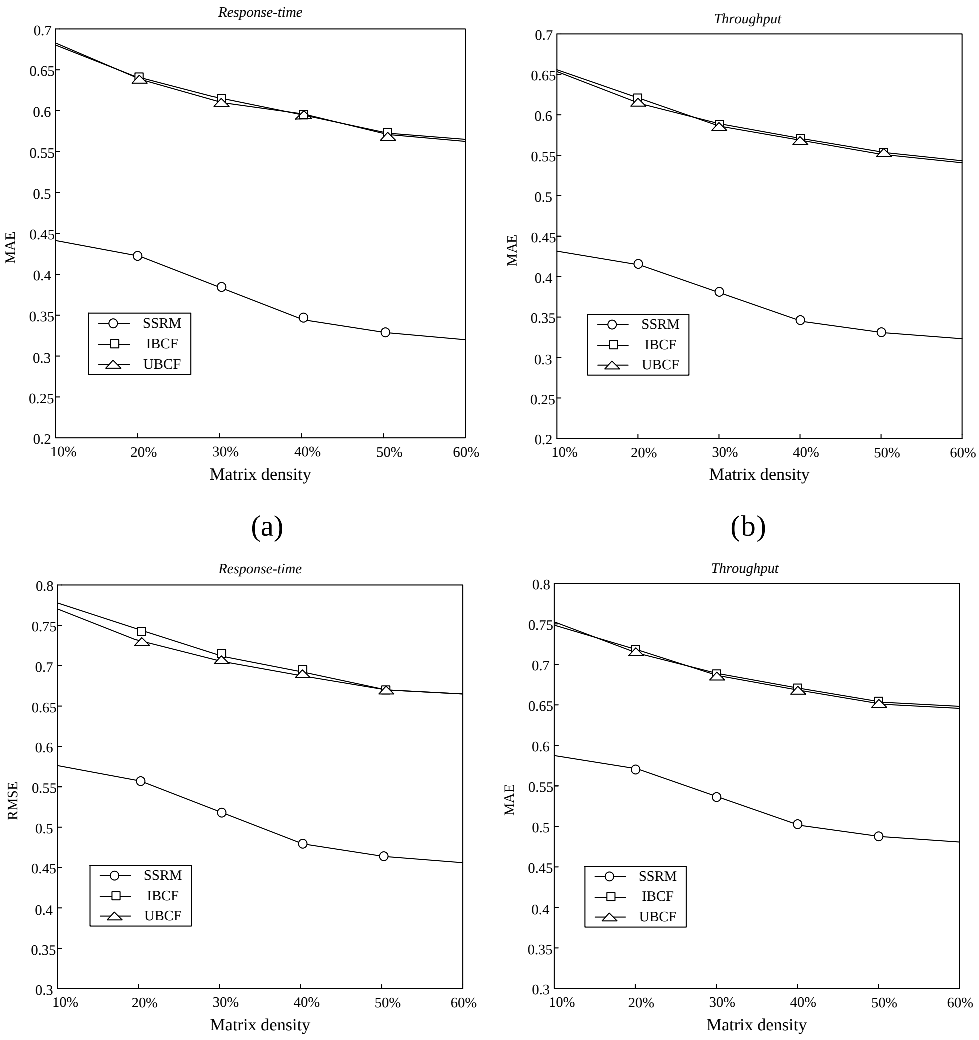
As presented in figure 7, the prediction performance is also enhanced with the value of matrix density increasing. The main reason is that denser user-service matrix provides more information for missing value prediction. NARM has the better performance than LBCF and UBCF under all experimental setting consistently, since NARM considers context similarity and the relational degree among cloud services. The experimental results of LBCF and UBCF are similar in this experiment, since these two similarity computation methods are similar with each other and are both rating-oriented.



(a) (b)

(c) (d)

Figure 6. Impact of neighbor size *k* comparison with SSRM, IBCF and UBCF



(a) (b)

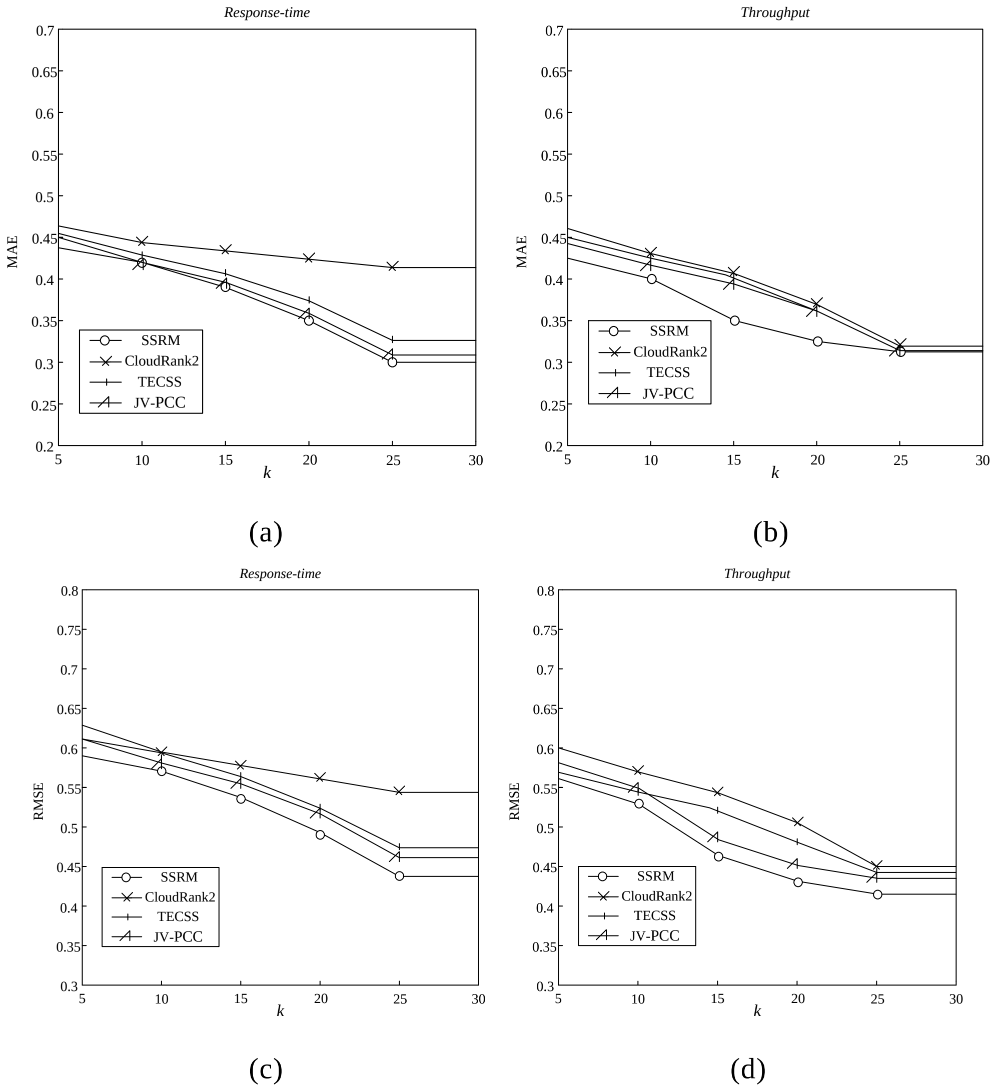
(c) (d)

Figure 7. Impact of matrix density comparison with SSRM, IBCF and UBCF

4.2. Performance Comparison with Other Popular Approaches

To study the prediction performance of SSRM, we compare SSRM with three existing QoS properties prediction approaches: CloudRank2 [3], TECSS [4] and JV-PCC [5].

The size of top-k similar service is an important factor in CF approach, which determines how many neighbors' historical records are employed to generate predictions [5]. Therefore, the impact of matrix density is not considered in the experiments. We set the density to 10%. We vary k from 5 to 30 in increment of 5. In order to simplify the experimental process, we consider all cloud services are trustworthy. Figure 8 shows the experimental results for response time and throughput. Under the same simulation condition, SSRM significantly outperform CloudRank2, TECSS and JV-PCC. The observations also suggest that better accuracy can be achieved by our model when more historical records are available in the service selection study. The main reason is that computing context similarity can improve the performance of prediction evaluation in MCC environments.



(a) (b)

(c) (d)

Figure 8. Impact of neighbor size *k* comparison with Other Popular Approaches

Figure (a) and (b) depict the MAE fractions of SSRM, CloudRank2, TECSS and JV-PCC for response-time and throughput, while Figure (c) and (d) depict the RMSE fractions. It can be observed that SSRM achieves smaller MAE and RMSE consistently than other approaches for both response-time and throughput.

5. Conclusions and Future

In the paper, we propose a novel service selection and recommendation model (SSRM) for mobile cloud computing environments. When calculating user similarity, context information and interest are considered. Through utilizing relational degree to improving the accuracy of ranking result, our service selection and recommendation method succeed in obtaining the ultimate service ranking results. In addition, SSRM only select trusted mobile cloud service as candidates. Therefore, a set of trusted candidates are identified for target user. The experimental results show that our approach significantly improves the prediction performance as compared with other two traditional approaches: item-based CF approach (IBCF) and user-based CF approach (UBCF).

There are some disadvantages in our SSRM model. Firstly, we exploit node distance to compute context similarity. There is an underlying assumption in this exploitation: each two adjacent nodes have equal semantic distance or granularity of nodes in each level is identical. This underlying assumption is not true in some scenarios and it deteriorates the performance of similarity evaluation. Secondly, the trust threshold is used to select trusted mobile cloud service as candidates. However, it cannot detect and exclude malicious QoS values provided by users. Thirdly, in our experiments there are only three types of context information are considered.

Therefore, in the future we will study the methods to improve context similarity measurement. We will study how to select the most trustworthy cloud service of certain type for the active users based on multi-dimensional trust evidence. We also will conduct more experimental investigations deal with the impact of different context changes on service selection. Moreover, we will improve the ranking accuracy of our approaches by exploiting additional techniques (e.g., combining rating-oriented approaches and ranking-oriented approaches, matrix factorization, intelligent optimization algorithms, etc.).

Acknowledgments

The work in this paper has been supported by National Natural Science Foundation of China (Program No. 71501156) and China Postdoctoral Science Foundation (Program No.2014M560796) and Shanxi Provincial Education Department (Program No. 15JK1679).

References

[1] White Paper, “Mobile Cloud Computing Solution Brief,” AEPONA, November 2010.

[2] https://en.wikipedia.org/wiki/Cloudlet.

[3] Z.B. Zheng, X.M. Wu, Y.L. Zhang, M.R. Lyu and J.M. Wang, “QoS Ranking Prediction for Cloud Services,” IEEE Transactions on Parallel and Distributed Systems, Vol.24, No.6, 2012, pp.1213-1222.

[4] Pan Y, Ding S, Fan W, Li J, Yang S, “Trust-Enhanced Cloud Service Selection Model Based on QoS Analysis,” PLoS ONE, Vol.10, No.11, 2015: e0143448. doi:10.1371/journal.pone.0143448.

[5] Ding S, Xia C-Y, Zhou K-L, Yang S-L, Shang JS, “Decision Support for Personalized Cloud Service Selection through Multi-Attribute Trustworthiness Evaluation,” PLoS ONE, Vol.9, No.6, 2014: e97762. doi:10.1371/journal.pone.0097762.

[6] S.K. Garg, S. Versteeg, R.Buyyaa, “A framework for ranking of cloud computing services,” Future Generation Computer Systems, Vol.29, 2013, pp.1012-1023.

[7] H. Ma, Z.G. Hu, K.Q. Li, H.Y. Zhang, “Toward trustworthy cloud service selection: A time-aware approach using interval neutrosophic set,” Journal of Parallel and Distributed Computing, Vol.96, 2016, pp.75-94.

[8] G. Lei, M. Jun, Z.M. Chen, H.R. Jiang, “Incorporating item relations for social recommendation,” Chinese Journal of Computers, Vol.37, No.1, 2014, pp.219-228.

[9] A. K. Dey, “Understanding and using context,” Personal and ubiquitous computing, Vol. 5, No. 1, 2001, pp. 4-7.

[10] X. Han, L.Y. Wang, S. Park, A.C.Rumin, N.Crespi, “Alike people, alike interests? A large-scale study on interest similarity in social networks,” Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2014, pp.491-496.

[11] Ryan N. L., Jake T. L., Nitesh V. C., “New perspectives and methods in link prediction,” Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010, pp.243-252.

[12] L. Munasinghe, R. Ichise, “Link Prediction in Social Networks Using Information Flow via Active Links,” IEICE Transactions on Information and Systems, Vol. E96.D, No. 7, 2013, pp. 1495-1502.

[13] ns3. https://www.nsnam.org/

[14] Markets and Markets. http://www.marketsandmarkets.com/PressReleases/mobile-cloud.asp

[15] C.T. Do, N.H.Tran, E.N. Huh, C.S. Hong, D. Niyato, Z. Han, “Dynamics of service selection and provider pricing game in heterogeneous cloud market,” Journal of Network and Computer Applications, Vol.69, 2016, pp.152–165.

[16] W.J. Fan, S.L. Yang, H.Perros, J. Pei, “A multi-dimensional trust-aware cloud service selection mechanism based on evidential reasoning approach,” International Journal of Automation and Computing, Vol. 12, No. 2, 2015, pp.208-219.

[17] X.G Wang, J. Cao, Y. Xiang, “Dynamic cloud service selection using an adaptive learning mechanism in multi-cloud computing,” Journal of Systems and Software, Vol. 100, 2015, pp.195-210.

[18] M. Whaiduzzaman, A. Gani, N.B. Anuar, M.Shiraz, M.N.Haque, and I.T. Haque, “Cloud service selection using multicriteria decision analysis,” • The Scientific World Journal, 2014, Article ID 459375: 1-10.

[19] Tserpes K, Aisopos F, Kyriazis D, Varvarigou T, “A recommender mechanism for service selection in service-oriented environments,” Future Generation Computer Systems, Vol.28, No.8, 2012, pp.1285–1294.

[20] Pal R., Hui P., “Economic models for cloud service markets: Pricing and capacity planning,” Theoretical Computer Science, Vol.496, No.22, 2013, pp.113–124.

[21] Grag S.K., Versteeg S., Buyya R., “A framework for ranking of cloud computing services,” Future Generation Computer Systems, Vol.29, No.4, 2013, pp.1012–1023.

[22] Z. Yang, B. Wu, K. Zheng, X.B. Wang, L. Lei, “A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications,” IEEE Access, Vol.4, 2016, pp.3273-3287.

[23] A. Ahmed, A.S. Sabyasachi, “Mobile cloud service selection using back propagation neural network,” International Journal of Computer Applications, Vo.129, No.14, 2015, pp. 1-5.

[24] A. Ezenwoke, O. Daramola, M. Adigun, “Towards a Visualization Framework for Service Selection in Cloud e-Marketplaces,” Proceedings of the IEEE International Conference on Web Services (ICWS), 2017, pp. 828-835.

[25] M.D. Wang, X.D. Dai, J.X. Liu, J.J. Chen, “Towards a trust evaluation middleware for cloud service selection,” Future Generation Computer Systems, Vol.74, 2017, pp.302-312.

[26] Z.U. Rehman, O.K.Hussain, F.K.Hussain, “Parallel Cloud Service Selection and Ranking Based on QoS History,” International Journal of Parallel Programming, Vol.42, No.5, 2014, pp.820-852.

[27] Axis: http://axis.apache.org/axis2

[28] Dataset: http://www.zibinzheng.com/tpds2012