A Web Service Recommendation Approach Based on QoS Prediction Using Fuzzy Clustering

Meng Zhang, Xudong Liu, Richong Zhang, Hailong Sun
School of Computer Science and Engineering
Beihang University
Beijing, China
Email: {zhangm, liuxd, zhangrc, sunhl}@act.buaa.edu.cn

***Abstract*—Web services, as loosely-coupled software systems, are increasingly being published to the web and there are a large number of services with similar functions. Therefore, service users compare the non-functional properties of services, e.g., Quality of Service (QoS), when they make service selection. This paper aims at generating a more comprehensive web service recommendation to users with a novel approach to fulfill more accurate prediction of unknown services QoS values. We accomplish the QoS prediction by using fuzzy clustering method with calculating the users similarity. Our approach improves the prediction accuracy and this is confirmed by comparing experiments with other methods. In addition, the quality of web services is considered as a multi-dimensional object, and each dimension is one aspect of the web services non-functional properties. We also provide an application example to demonstrate how to utilize our approach to rank services by a score function and map multi-dimensional QoS properties into a single dimensional value.**

***Keywords*-Web service; QoS prediction; Fuzzy clustering; Recommender system**

I. INTRODUCTION

The number of web services published in the Internet is increasing rapidly. For example, both Seekda[1] and ServiceXchange[2] have collected over 20,000 Web services. When choosing web services, most users focus on comparing the differences of non-functional properties between web services with similar functionalities. Therefore, Quality of Service (QoS), which represents the non-functional properties of services, is used as a basis of ranking in the web service recommender systems.

Since many web services have the similar functions, consumers have to make service selecting decisions without prior knowledge about service candidates. The recommender systems aim at helping consumers in service selecting. Collaborative Filtering (CF) is a commonly used approach to build e-Commerce recommender systems. Several approaches are proposed based on CF in order to make QoS prediction, and many attempts are made to improve the accuracy of the predicted values. Existing quality of web service predicting approaches [1], [2] or web service recommendation methods [3], [4] merely focus on modeling the response time or the throughput of web services. Therefore, these quality attributes cannot comprehensively characterize the performance of a specific service. There is a need of providing web service users with an efficient and effective approach to generate multi-dimensional QoS recommendations and to establish a score function to represent the overall performance of services.

Our approach makes use of the fuzzy clustering in QoS prediction. In fuzzy clustering approach, each pattern does not belong to an exact cluster. This is more close to the QoS prediction problem where services or users have different properties and it is difficult to find the accurate cluster. In this paper we take web service users as the clustering objects and optimize the traditional FCM for the web service recommendation.

Moreover, we take web services as multi-dimensional objects, and each dimension is the one aspect of the non-functional properties. Response time, for example, belongs to performance dimension properties. Similarly there are accuracy, scalability and strength, etc. We advocate that all of these factors should be taken into account when ranking the services. We propose a function to map the multi-dimensional QoS properties into a single dimensional value to improve the rationality of normalization and give a comprehensive and reasonable recommendation. In summary, the major contribution of this study includes:

- We use an improved fuzzy clustering method (Fuzzy c-means [5]) during the ranking process by incorporating the advantages of collaborative filtering to increase the accuracy of prediction in the case of data sparsity.
- We propose a method to normalize multiple aspects of QoS data, and to rank web services based on this overall score, which comprehensively evaluates multiple aspects of web services and improve the quality of recommendation and users experience.
- We provide experiments to compare our approach with other methods in prediction accuracy and give an application example to demonstrate how to utilize our approach to rank web services.

[1]Seekda can be access at http://www.seekda.com
[2]ServiceXchange can be access at http://www.servicexchange.cn

978-0-7695-4753-4/12 $26.00 © 2012 IEEE
DOI 10.1109/SCC.2012.24

The rest of the paper is organized as follows. Section II briefly introduces the background information of web service recommendation and fuzzy clustering. Section III introduces the architecture of our recommender system. Section IV describes the approach in detail, including theory and algorithm. Section V provides the experimental process and results, gives the experimental comparison with other QoS prediction methods and an application example about the ranking methods. Section VI comes to the conclusion and discussion of the future work.

II. Background

A. Web Services Recommendation and QoS Prediction

In the existing web services recommender system, such as collaborative filtering approaches, it is a commonly used approach to calculate the similarity between service users to make prediction for the missing QoS data. Traditional collaborative filtering method mainly includes memory-based and model-based approaches.

The main idea of the memory-based collaborative filtering method is using the user-item rating matrix to calculate the similarity between users or services, and making prediction by a certain algorithm. Shao et al. [6] propose a method to find the users similarity by collaborative filtering and predict the QoS data based on the similar users service invocation histories. This is a typical user-based collaborative filtering method.

There are many existing approaches to improve the memory-based collaborative filtering method. Zhang [7] presents a method of Web service ranking by collaborative filtering using invocation history especially the QoS query information. The approach asks for the name of the QoS property that the user most cares about when looks up a service.

As a hybrid collaborative filtering method, the approach [8] proposed by Zheng et al. uses both user-based and item-based methods, and solves the Web service recommendation problem based on these two approaches. They design an hybrid CF algorithm for QoS prediction. User-based results and item-based results are combined by confidence weight in their study. Finally the recommendation is given based on the ranking of QoS values.

Other attempts in QoS prediction using model-based method are not an uncommon approach. The ideas of pattern recognition, data mining and machine learning are often used. Ge et al. [9] propose a QoS prediction method based on pattern recognition to predict the QoS of the web services, and consider the impact of the differences of users feeling for the different network environments and platforms.

However, there are still exist limitations in existing recommender systems. In this paper, we focus on reducing the error of prediction in the case of data sparsity, and improving the quality of recommendation by providing a comprehensive QoS properties ranking result.

B. Fuzzy Clustering

The function of clustering is to organize the data elements and make data elements reach a certain level of similarities in the same cluster. Elements between clusters should be as different as possible. But in fuzzy clustering, data elements are allowed to belong to more than one cluster, and the method defines the strength of association between each element and cluster with membership function. The theoretical basis is the fuzzy sets theory introduced by Zadeh et al. in 1974 [10]. It was R. Yager [11] who firstly used fuzzy logic in recommender systems in 2003. Recent research works are proposed using hybrid recommendation methods based on fuzzy logic to recommend movies [12] and books [13] to users. In [14] proposed a Fuzzy Clustering based Recommender System for generating possible interested movies for users and results show improvements compared to other existing approaches. In this paper, we proposed a Fuzzy Clustering based QoS prediction approach by combining both collaborative filtering and Fuzzy Clustering approaches.

C. Fuzzy C-Means(FCM)

FCM is a data clustering method which had been widely used in pattern recognition. It is introduced by Bezdek [5] and it aims at clustering several patterns into clusters while minimizing the objective function:

$$J = \sum_{j=1}^{N} \sum_{i=1}^{c} \mu_{ij}^{m} \left(d_{ij}\right)^{2} \tag{1}$$

In Equation (1), as described in [5], N stands for the number of patterns; c represents the number of clusters;m is the weighting exponent; μ_{ij} is the membership value that shows the membership degree of that the j^{th} pattern to the i^{th} cluster center; and d_{ij} is a distance measure function to measure the distance between the j^{th} pattern and the i^{th} cluster center.

The result of FCM contains two parts. The former is the cluster center matrix, and the latter is the membership values matrix. At the end of iteration, each cluster has a center, so these centers form the cluster center matrix , and each pattern has a membership value for each cluster center, and the membership values matrix $M = [\mu_{ij}]_{N\times c}$.

R. Mohana et al. [15] use fuzzy clustering in web service discovery. The authors take web services as patterns and the dimensions are aspects of QoS properties. But the limitation of this method is that not all the QoS data can be collected simply from the invocation record, and the experiment requires an ideal experimental data. Birtolo et al.'s algorithm [16] improves the accuracy of prediction by two different fuzzy clustering strategy named BestCluster and AllClusters, and authors give several experiments to compare the prediction accuracy.

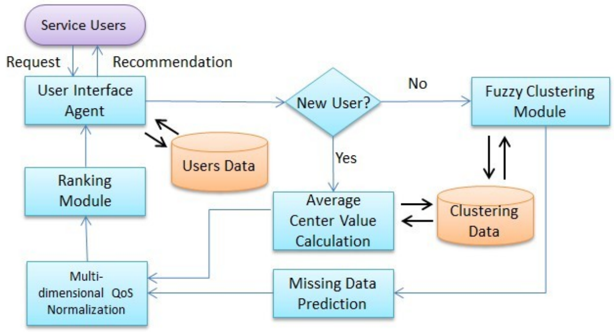


Figure 1. Recommender System Architecture

III. System Architecture

In this paper, we propose a web service recommender system. The schematic of the proposed architecture is shown in Figure 1. The functionality of the user interface agent is to receive the users requests and recommend services according to the results from the ranking module. When service users send a query request for a web service to the user interface agent, the agent should distinguish if the user is an existing active user or a new one. In other words, the agent needs to interact with the users history so as to determine whether the user is an active user (already has an invocation record in database).

In the situation where a user is an active user, our system will use the fuzzy clustering module. The functionality of this module is to retrieve the cluster related data. The parameters of clustering and similarity are the input for this module; the membership values matrix and the cluster centers matrix are the output of this module. These two matrixes are the input of next module, missing data prediction which predicts missing values of QoS properties. The QoS values will be normalized in the step of multi-dimensional QoS normalization in different methods, and then mapped into a single dimensional value on which the ranking module works.

If the user is not an active user, which means she/he do not have any invocation experience, then we look up all the cluster centers, calculate an average values for each service and come to the normalization module and ranking module to complete the recommendation which is consistent with the active user situation.

IV. Recommendation Approach

Many recommendation methods based on calculating similarity between users or items have been successfully applied to make prediction. In our approach, we combined the advantages of similarity calculation and fuzzy clustering, and thus increase the accuracy of prediction. In this study, we calculate the Pearson Correlation Coefficient (PCC) [8] between users. For active users, who have had some web services invocation experiences, we make use of a fuzzy clustering algorithm to predict the performance of web services they are going to invoke. In this section, we introduce the algorithms used in our recommender system.

A. Problem Definition

$U = \{u_1, u_2, u_3, \cdots, u_v\}$ is the set of users, where $u_i \, (1 \leq i \leq v)$ denotes i^{th} user.

$S = \{s_1, s_2, s_3, \cdots, s_w\}$ is the set of web services which have similar functionalities, where $s_j \, (1 \leq j \leq w)$ denotes j^{th} web service.

$q_{ij} \, (1 \leq i \leq v, 1 \leq j \leq w)$ is a vector of a specific QoS value that the user u_i provided as a feedback after invoking the service s_j.

$Q_i = \{q_{i1}, q_{i2}, q_{i3}, \cdots, q_{iw}\}$ is a vector describing the QoS data of services provided by user u_i and the user-service matrix is defined as follows:

$Q = \{Q_1, Q_2, Q_3, \cdots, Q_v\}'$

It can also be written as: $Q = \begin{pmatrix} q_{11} & \cdots & q_{1w} \\ \vdots & \ddots & \vdots \\ q_{v1} & \cdots & q_{vw} \end{pmatrix}$

The quality of web service predicting problem of the interest of this paper can then be rephrased as determining how good is an arbitrary web service s_j would be, given S and U.

B. Similarity Calculation

In our approach, we use Pearson Correlation Coefficient (PCC) to measure the degree of similarity between users based on QoS data of services that have been invoked. For example, we use the following equation to calculate the similarity between user a and b:

$$sim'(a,b) = \frac{\sum_{i \in I(a) \bigcap I(b)} (q_{ai} - \overline{q_a}) \cdot (q_{bi} - \overline{q_b})}{\sqrt{\sum_{i \in I(a)} (q_{ai} - \overline{q_a})^2} \cdot \sqrt{\sum_{i \in I(b)} (q_{bi} - \overline{q_b})^2}} \quad (2)$$

where $\overline{q_a}$ and $\overline{q_b}$ represent the average QoS values that the user a and b observed, and $I(a)$ and $I(b)$ represent the services sets the users have invoked respectively.

We note that when there are only a small number of services that both users have invoked, or one popular service has been invoked by the majority of users, the degree of similarity calculated only with the equation (2) will be overestimated. In order to address this problem, we employ a parameter to reduce this impact, and the new similarity can be calculated as follows:

$$sim(a,b) = \frac{I(a) \bigcap I(b)}{I(a) \bigcup I(b)} sim'(a,b) \quad (3)$$

C. Fuzzy Clustering Algorithm-PFCM

QoS data relates to a variety of factors. It is difficult to divide all QoS data observed by different users into groups on some exact constraints. Therefore, traditional clustering approaches, for example k-means which cannot describe the membership accurately. In our approach, we improve the clustering process based on the FCM algorithm by combining PCC calculation. We use the value of similarity between users data and cluster centers instead of using Euclidean distance. In other words, we make use of the PCC as the distance measure function to calculate the membership function. The specific formula will be given latter in this section.

According to the problem definition, we have $U = \{u_1, u_2, u_3, \cdots, u_v\}$ as the set of v users that have provided a set of services invocation data, and u_i provides $Q_i = \{q_{i1}, q_{i2}, q_{i3}, \cdots, q_{iw}\}$, here we take as the i^{th} pattern. So $Q = \{Q_1, Q_2, Q_3, \cdots, Q_v\}'$ is a set of v patterns correspond to v users, and each pattern is a w-dimensional vector corresponding to w services. The main idea of fuzzy clustering is to form k cluster centers, and each pattern has a membership value to each cluster center. So our purpose of this part is to determine the membership degree of each user for each cluster center.

The membership function in FCM should satisfy the following conditions [5]:

$$\begin{cases} \sum_{i=1}^{k} \mu_{ij} = 1, \forall j = 1, \cdots, v \\ 0 < \sum_{j=1}^{n} \mu_{ij} < n, \forall i \\ \mu_{ij} \in [0, 1] \end{cases} \tag{4}$$

In equation (4), μ_{ij} represents the membership value of the j^{th} pattern for the i^{th} cluster.

The fuzzy clustering is an iterative process, it iterates until the objective function reaching a minimum, and the objective function is calculated by employing the following equation in the original algorithm as mentioned in Section II:

$$J' = \sum_{j=1}^{N} \sum_{i=1}^{c} \mu_{ij}^{m} \left(d_{ij}\right)^2 \tag{5}$$

where $\left(d_{ij}\right)^2 = \left\|C_i - Q_j\right\|^2$ represents the Euclidean distance between i^{th} cluster and j^{th} pattern in the original article [13].

As PCC describes the distance between users more accurate than the Euclidean distance when pattern is the quality of services observed by users in the real world, in this study, we use similarity calculated by improved PCC, as shown in equation (3), instead of Euclidean distance. In the Section V we will prove this fact by our experiments. In addition, due to the special requirements of the distance measure function during the calculation, we use the square of the similaritys reciprocal. In this case, the higher similarity represents a shorter distance and vice versa. To incorporate the similarity of patterns, we enhance the objective function of PFCM (PCC combined with FCM) with the following improvement based on equation (5):

$$J = \sum_{j=1}^{N} \sum_{i=1}^{c} \mu_{ij}^{m} \left[sim\left(C_i, Q_j\right)^{-1}\right]^2 \tag{6}$$

The essence of the algorithm is to obtain the membership values matrix M and the cluster center matrix C which make the objective function, equation (6), reach a minimum. After mathematical derivation, it can be concluded that the following equations meet the conditions of M with elements μ_{ij} and C with elements C_i:

$$\mu_{ij} = \frac{sim\left(C_i, Q_j\right)^{\frac{2}{m-1}}}{\sum_{s=1}^{k} \left[sim\left(C_s, Q_j\right)\right]^{\frac{2}{m-1}}} \tag{7}$$

$$C_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{m} Q_j}{\sum_{j=1}^{n} \mu_{ij}^{m}} \tag{8}$$

Equation (8) is the function to calculate the center for each cluster. Obviously, C_i and Q_j have equal number of dimensions. The following part shows the detailed PFCM algorithm.

Algorithm 1: PFCM

Input : threshold ε, max iteration number L, user-service matrix Q

Output: membership values matrix M, center matrix C

1 $l \leftarrow 0$
2 Randomly select points in M as the initial cluster centers $C^{(0)}$
3 Initialize the membership values matrix $M^{(0)}$ with $C^{(0)}$ and Q by equation(7)
4 Calculate $C^{(1)}$ with equation (8) based on $M^{(0)}$
5 Update $M^{(1)}$ with equation(3)and (7) based on $C^{(1)}$
6 Select the appropriate matrix norm to compare $C^{(1)}$ and $C^{(0)}$ (as $C^{(l+1)}$ and $C^{(l)}$)
7 if $\left\|C^{(l+1)} - C^{(l)}\right\| \leq \varepsilon$ or $l = L$, end the iteration, and return $M^{(l)}$ and $C^{(l)}$
8 else $l \leftarrow l + 1$ and turn to step 4

D. Prediction Method

Equation (9) defines the formulation of making prediction on the QoS data of web services. It consists of two parts: the membership values multiplied by the center values as the first part of the prediction, and the data from the similar users as the second part to correct the accuracy of the former part. For example, if user a has never invoked service s_w,

Table I
MULTI-DIMENSIONAL QOS DEFINITION

Dimension name	Measurable indicators
Performance	Response time
Strength	Throughput
Scalability	Number of operations
Reliability	Rate of failures in a certain time period
Accuracy	Rate of errors over a time interval

the prediction of s_w's QoS value will be defined as a vector s_w in Q_a:

$$\hat{Q}_a(s_w) = \sum_{i=1}^{k} \mu_{ia} C_i(s_w) + \frac{\sum_{j=1}^{v} sim(Q_j, Q_a) \cdot \left(Q_j(s_w) - \overline{Q_j}\right)}{\sum_{j=1}^{v} sim(Q_j, Q_a)} \quad (9)$$

In equation (9), s_w represents the w^{th} vector in C_i. It can be seen that this formulation takes into account users similarities to the cluster center and other users in the same cluster.

E. Normalization and Ranking

According to the W3C's definition[3], QoS contains many aspects. In this paper, we regard some non-functionalities as the multi-dimensional QoS. The Table I lists the QoS properties used in this study, including the name and the measurable indicators. With these indicators, we can calculate the value of each dimension.

Because the values of QoS properties can be different types, for example, data types and ratio types, the normalization is necessary. Existing approaches normalizing the properties with linear normalization method could not distinguish the differences between properties.

For the performance dimension, we have adopted Gaussian Normalization approach [17] and 3-Sigma rules to convert it into the range of [0, 1]. As services, which have shorter response time, will possibly achieve better performances, we use equation (10) to calculate the score of performance dimension.

$$score_{perf} = 1 - \left(\left(q_{ij} - \overline{q_i}\right)/(3 * \sigma_i) + 1\right)/2 \quad (10)$$

where σ_i represent the standard deviation.

For strength and scalability dimensions, the similar function using Gaussian Normalization approach can be written as:

$$score_{stre/scal} = \left(\left(q_{ij} - \overline{q_i}\right)/(3 * \sigma_i) + 1\right)/2 \quad (11)$$

The last two dimensions of properties are rates and the scores are defined as:

[3]http://www.w3c.or.kr/kr-office/TR/2003/ws-qos

$$score_{relia} = 1 - N_{failures}/N_{invokes} \quad (12)$$

$$score_{accu} = 1 - N_{errors}/N_{invokes} \quad (13)$$

With these equations, all the dimensions of QoS properties are converted into the range of [0, 1]. The final ranking result based on scores in the form of summarizing on all predicted QoS properties as shown in equation (14).

$$score_{final} = \sum_{d \in D} score_d \quad (14)$$

where D is the set of all QoS properties dimensions.

In Equation (12) and Equation (13), $N_{failures}$ represents the number of times the feedback presents failure information in a certain period; N_{errors} means the same when information contains error description, and $N_{invokes}$ represents the total number of invocations in a certain period. As we calculate equation (14), parameters (weights) can be introduced for each QoS dimension to enhance this final score. These parameters can be inferred or learned from users history as long as the users behaviors are collected.

It is obvious that the significance of the score is not to represent a specific value, but to describe the position of the real value in the dimension.

We note that the scalability can be directly obtained from the WSDL file of the web service or the history of users invocation experiences, so that recommendations would be made based on these extended dimensions of web service quality properties.

V. EXPERIMENTS AND APPLICATION EXAMPLE

In this section, we introduce the metric and the prediction results in our experiments. Also, we implement an application example on the normalization and ranking for recommendation to show our proposed approach.

A. Data Preparing and Evaluation Metric

The dataset in the experiments is a set of QoS data based on the service repository ServiceXchange, which is collected by the ACT Lab of Beihang University. Inspired by Zheng's work in [8], we collected data in the following way. The dataset is collected by 10 nodes from Planet-Lab[4] , which contains 4.7 million web service invocation records of more than 17000 web services in ServiceXchange. We randomly select 100 services and obtain the WSDL files to record the operations number, and for each node we select 300 records which is a file of one invocation of the 100 web services, represent 300 users. As a result, we have 3000 users, and the user-service matrix is generated from these data.

We change the number of services in each training set and testing set to compare the prediction effect in different data density and different number of clusters.

[4]http://www.planet-lab.org

To simulate the real situation, we randomly remove some QoS data of the training matrix and the testing matrix. This makes the sparse matrixes with data density of 10%, 20% and 30%.

To evaluate the performance of our algorithm, we make use of Mean Absolute Error (MAE) to compare with other commonly used QoS prediction approaches. MAE is a statistical accuracy metric which is widely used to measure the prediction quality in collaborative filtering methods. It is defined as the average absolute deviation of the predictions and the truth values in equation (15):

$$MAE = \frac{\sum_{ij} \left| Q_i\left(s_j\right) - \hat{Q}_i\left(s_j\right) \right|}{N} \tag{15}$$

where $Q_i\left(s_j\right)$ denotes the actual data of web services s_j given by user u_i, $\hat{Q}_i\left(s_j\right)$ denotes the predicted one ,and N denotes the number of tested data. Smaller MAE means better prediction accuracy.

B. Result Analysis

1) FCM distance measure functions Comparison: In this part, we compare the algorithm performances by using both Euclidean distance and PCC as the distance measure function in FCM.

It should be noted that when we use PFCM algorithm in the fuzzy clustering process, we used m=2 as the weighting exponent like most fuzzy clustering methods; the number of clustering is 6, and this number do not have very great impact on the final result which will be proved in later experiment; the maximum number of iterations is 500, and the threshold is 0.01.

In Figure 2, the abscissa axis means the density of the training matrix, and the vertical axis represents the MAE of the prediction. Figure 2 shows that even the given services number changes from 10 to 30 (marked as Given =10, 20 and 30), the density of the training matrix ranges from 0.1 to 0.9, the MAE when using PCC as the distance functions is always smaller than the prediction using Euclidean distance. This proves that the distance measure function we used can get the desired results.

2) Impact of Clustering Number: In this part we change the number of clusters during the fuzzy clustering process. In Figure 3(a), we have a training matrix employing 20% density and change the number of services given in the training matrix. The figure shows that the MAE is getting smaller when the number of clusters becomes larger than 2 until reaching 10. When the number of clusters comes to 12 and 14, the MAE is stop getting smaller. The similar trend is shown in the Figure 3(b). In Figure 3(b), the changing indicator is density, and the service number of the training matrix is 20. We can also see a rise of the MAE when the number of clusters comes to 14, and decrease latter. Our analysis result shows that the clustering results by using

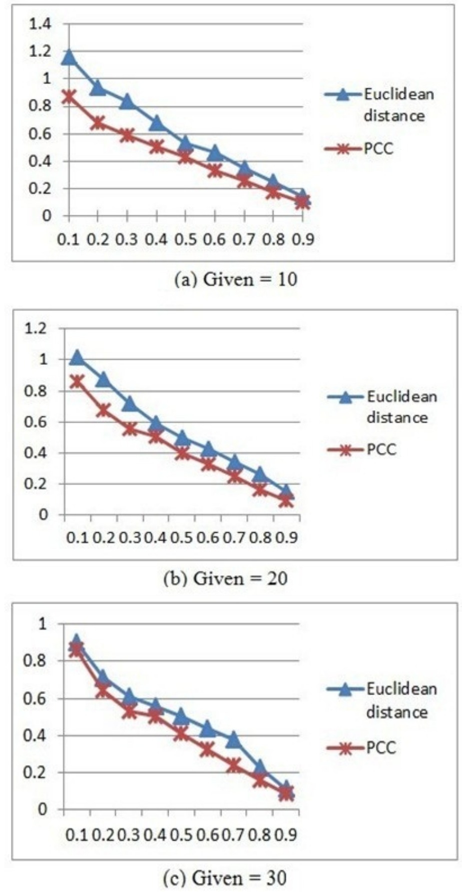


Figure 2. FCM distance measure function comparison

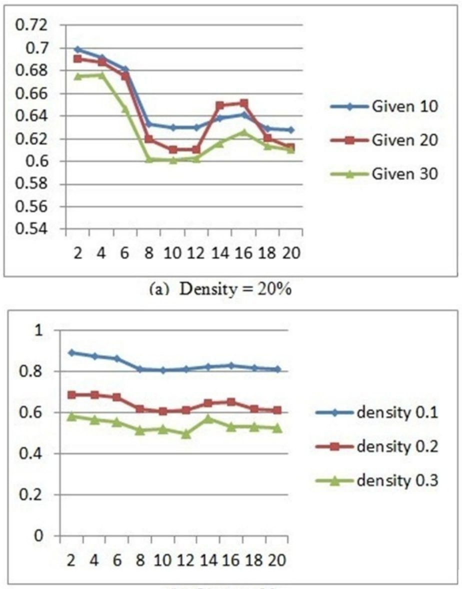
different numbers of clusters are related to the patterns number. So a reasonable choice of number of clusters can be obtained in the experiments when using our method in different environment. Based on these experiments, we find that the smallest MAE appears when the number of clusters is 12 which is used as the cluster number in next experiments.

3) MAE Performance Comparison: In order to show the performance of our approach, we compare the prediction performances with other prediction methods: UMEAN, which makes prediction for a user by calculating average values from the QoS data that this specific user obtained as they invoking other web services; IMEAN, which predicts the missing QoS values in the user-service matrix by calculating the average values of this specific service obtained by other users. We also compare our proposed algorithm, PFCM, with other commonly-used recommendation approaches, such as user-based collaborative filtering method UPCC [18] and item-based collaborative filtering method IPCC [19].

The experimental results in Table II show that UMEAN and UPCC do not performance well under the condition of

Table II
MAE PERFORMANCE COMPARISON

	Matrix Density=10%			Matrix Density=20%			Matrix Density=30%		
Given	10	20	30	10	20	30	10	20	30
UMEAN	1.422	1.265	1.197	1.336	1.283	1.154	1.145	1.007	1.056
IMEAN	0.926	0.913	0.854	0.917	0.897	0.837	0.818	0.809	0.787
UPCC	1.250	1.103	0.929	0.842	0.829	0.812	0.756	0.729	0.705
IPCC	0.901	0.881	0.867	0.806	0.770	0.733	0.690	0.678	0.659
PFCM	0.875	0.864	0.860	0.681	0.675	0.647	0.593	0.556	0.533



(a) Density = 20%

(b) Given = 20

Figure 3. Impact of clustering number

data sparsity. Our approach can achieve smaller MAE in the most situations.

C. Application Example

We choose 10 services from serviceXchange and all of them provide weather forecast service. After calling these services from 10 nodes distributed on Planet-Lab, we randomly select 100 invocations record of each service. We choose the user whose ID is 2913 as a potential weather forecast service user. After the similarity calculation and fuzzy clustering, we make prediction for each QoS property dimension. The results are as follows in Table III. In Table III, we can see the predicted QoS values of 10 web services. With the normalization functions proposed in Section IV, we get a normalized value for each QoS property dimension. This result is shown in Table IV. Table V shows the final result by mapping the multi-dimensional QoS value into a single dimensional score. With this overall quality score of web services, one may easily build a web service recommender system.

In order to make the ranking results easily understood by potential users, we can show the predicted results for multiple quality dimensions. It can be observed from this application example that our approach could assist services users as they make service selection decisions.

Table III
QOS PROPERTIES OF SELECTED WEB SERVICE

WS name	Performance	Strength	Scalability	Reliability	Accuracy
ShowTheWeather	5.435	7.923	5	0.076	0.021
BerreWeather	0.326	20.032	2	0.122	0.019
GlobalWeather	0.258	9.784	6	0.178	0.101
LivedoorWeather	3.587	0.924	2	0.210	0.038
GlobalWeatherService	0.295	5.822	14	0.053	0.076
Weather	0.052	53.489	3	0.032	0.022
WeatherStationService	1.776	5.211	1	0.131	0.201
WeatherService	0.137	4.932	2	0.244	0.100
WeatherForecastService	0.231	23.193	2	0.089	0.049
DASWorldWeather	0.849	10.622	6	0.057	0.105

Table IV
NORMALIZED QOS PROPERTIES FOR RANKING

WS name	Performance	Strength	Scalability	Reliability	Accuracy
ShowTheWeather	0.269	0.495	0.508	0.924	0.979
BerreWeather	0.554	0.505	0.471	0.878	0.981
GlobalWeather	0.558	0.497	0.521	0.812	0.899
LivedoorWeather	0.372	0.490	0.471	0.790	0.962
GlobalWeatherService	0.556	0.493	0.621	0.947	0.924
Weather	0.569	0.531	0.484	0.968	0.978
WeatherStationService	0.473	0.493	0.459	0.869	0.799
WeatherService	0.565	0.492	0.471	0.756	0.900
WeatherForecastService	0.559	0.507	0.471	0.911	0.951
DASWorldWeather	0.523	0.497	0.521	0.943	0.895

VI. CONCLUSION AND FUTURE WORK

In this paper we propose a web service recommender system. One of the main contributions is to improve fuzzy clustering algorithm by using the similarity calculation as a distance measure function to make prediction for the missing QoS values more accurately in the case of data sparsity. The

Table V
RANKING RESULT

WS name	Score	Ranking
Weather	3.530	1
WeatherForecastService	3.399	2
BerreWeather	3.389	3
DASWorldWeather	3.379	4
GlobalWeatherService	3.354	5
GlobalWeather	3.287	6
WeatherService	3.184	7
ShowTheWeather	3.175	8
WeatherStationService	3.093	9
LivedoorWeather	3.085	10

other contribution is to take services QoS properties as a multi-dimensional object. We provide a mapping function to normalize the value of each dimension to a score, which is the basis of building service recommender systems.

We combine two commonly-used recommendation approaches, collaborative filtering and fuzzy clustering, during the process of making prediction. The experimental result confirms an increase of the accuracy of prediction in the case of data sparsity.

In this study, we merely consider 5 QoS properties. In fact, there are still many aspects of QoS properties can be taken into account in the future, such as the security, network environment and etc.

The future work also includes taking into account users preferences, such as asking users to select the dimension of QoS properties that she/he cares more. Such, we can improve the weight of dimensions in the overall quality score calculation and provide more satisfactory results to service users.

ACKNOWLEDGMENT

This work was supported partly by National Natural Science Foundation of China (No. 61103031), partly by China 863 program (No. 2012AA011203), partly by the State Key Lab for Software Development Environment (No. SKLSDE-2010-ZX-03), and partly by the Fundamental Research Funds for the Central Universities (No. YWF-12-RHRS-016).

REFERENCES

[1] L. Li, M. Rong, and G. Zhang, "A web service qos prediction approach based on multi-dimension qos," in *Computer Science & Education (ICCSE), 2011 6th International Conference on*. IEEE, 2011, pp. 1319–1322.

[2] L. Chen, Y. Feng, J. Wu, and Z. Zheng, "An enhanced qos prediction approach for service selection," in *Services Computing (SCC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 727–728.

[3] Y. Jiang, J. Liu, M. Tang, and X. Liu, "An effective web service recommendation method based on personalized collaborative filtering," in *Web Services (ICWS), 2011 IEEE International Conference on*. IEEE, 2011, pp. 211–218.

[4] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *Web Services (ICWS), 2010 IEEE International Conference on*. IEEE, 2010, pp. 9–16.

[5] J. Bezdek, R. Ehrlich *et al.*, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[6] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction forweb services via collaborative filtering," in *Web Services, 2007. ICWS 2007. IEEE International Conference on*. IEEE, 2007, pp. 439–446.

[7] Q. Zhang, C. Ding, and C. Chi, "Collaborative filtering based service ranking using invocation histories," in *Web Services (ICWS), 2011 IEEE International Conference on*. IEEE, 2011, pp. 195–202.

[8] Z. Zheng, H. Ma, M. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*. IEEE, 2009, pp. 437–444.

[9] J. Ge, Z. Chen, J. Peng, T. Li, and L. Zhang, "Web service recommendation based on qos prediction method," in *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*. IEEE, 2010, pp. 109–112.

[10] L. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[11] R. Yager, "Fuzzy logic methods in recommender systems," *Fuzzy Sets and Systems*, vol. 136, no. 2, pp. 133–149, 2003.

[12] T. Jeon, J. Cho, S. Lee, G. Baek, and S. Kim, "A movie rating prediction system of user propensity analysis based on collaborative filtering and fuzzy system," in *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*. IEEE, 2009, pp. 507–511.

[13] M. Maatallah and H. Seridi, "A fuzzy hybrid recommender system," in *Machine and Web Intelligence (ICMWI), 2010 International Conference on*. IEEE, 2010, pp. 258–263.

[14] J. Wang, N. Zhang, and J. Yin, "Collaborative filtering recommendation based on fuzzy clustering of user preferences," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, vol. 4. IEEE, 2010, pp. 1946–1950.

[15] R. Mohana and D. Dahiya, "Optimized service discovery using qos based ranking: A fuzzy clustering and particle swarm optimization approach," in *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*. IEEE, 2011, pp. 452–457.

[16] C. Birtolo, D. Ronca, and R. Armenise, "Improving accuracy of recommendation system by means of item-based fuzzy clustering collaborative filtering," in *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*. IEEE, 2011, pp. 100–106.

[17] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. Huang, "Supporting similarity queries in mars," in *Proceedings of the fifth ACM international conference on Multimedia*. ACM, 1997, pp. 403–413.

[18] J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.

[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 1994, pp. 175–186.