ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 22, April 2016

Web Service Proposal and Visualization using Value Prediction Method

Mr. R. Govindan, M.E., Mr. C. Kumar, M.E.,
Assistant Professor, Idhaya Engineering College for Women, Chinnasalem

Abstract—With the proliferation of web services, effective QoS-based approach to service recommendation is becoming more and more important. Although service recommendation has been studied in the recent literature, the performance of existing ones is not satisfactory, since 1) previous approaches fail to consider the QoS variance according to users' locations; and 2) previous recommender systems are all black boxes providing limited information on the performance of the service candidates. In this paper, we propose a novel collaborative filtering algorithm designed for large-scale web service recommendation. Different from previous work, our approach employs the characteristic of QoS and achieves considerable improvement on the recommendation accuracy. To help service users better understand the rationale of the recommendation and remove some of the mystery, we use a recommendation visualization technique to show how a recommendation is grouped with other choices. Comprehensive experiments are conducted using more than 1.5 million QoS records of real-world web service invocations. The experimental results show the efficiency and effectiveness of our approach.

Key Terms—Service recommendation, QoS, collaborative filtering, self-organizing map, visualization

1 INTRODUCTION

EB services are software components designed to support interoperable machine-to-machine interaction over a network. The adoption of web services as a delivery mode in business has fostered a new paradigm shift from the development of monolithic applications to the dynamic setup of business process. In recent years, web services have attracted wide attentions from both industry and academia, and the number of public web services is steadily increasing.

When implementing service-oriented applications, ser- vice engineers (also called service users) usually get a list of web services from service brokers or search engines that meet the specific functional requirements. They need to identify the optimal one from the functionally equivalent candidates. However, it is difficult to select the best performing one, since service users usually have limited knowledge of their performance. Effective approaches to service selection and recommendation are urgently needed. Quality-of-Service (QoS) is widely employed to represent the nonfunctional performance of web services and has been considered as the key factor in service selection. QoS is defined as a set of user-perceived

properties including response time, availability, reputation, etc. Currently, it's not practical for users to acquire QoS information by evaluating all the service candidates, since conducting real-world web service invocations is time- consuming and resource-consuming. Moreover, some QoS properties (e.g., reputation and reliability) are difficult to be evaluated, since long-duration observation and a number of invocations are required. Besides client-side evaluation, it's impractical to acquire QoS information from service providers or third-party communities, because service QoS performance is susceptible to the uncertain Internet envir- onment and user context (e.g., user location, user network condition, etc.). Therefore, different users may observe quite different QoS performance of the same web service, and QoS values evaluated by one user cannot be used directly by another in service selection and recommendation.

The objective of this paper is to make personalized QoS- based web service recommendations for different users and thus help them select the optimal one among the functional equivalents. Several previous work has applied collaborative filtering (CF) to

web service recom- mendation. These CF-based web service recommender systems work by collecting user observed QoS records of different web services and matching together users who share the same information needs or same tastes. Users of a CF system share their judgments and opinions on web services, and in return, the system provides useful personalized recommendations. However, three unsolved problems of the previous work affect the performance of current service recommender systems. The first problem is that the existing approaches fail to recognize the QoS variation with users' physical locations. After the analysis of a real-world web service data set,which contains 1.5 million service invocation results of 100 public services evaluated by users from more than 20 countries we

highly relate to the users' physical locations. For example, the response time of a service observed by users who are closely located with each other usually fluctuates mildly around a certain value, while it sometimes varies significantly between users far away from each other.

The second problem is the online time complexity of memory-based CF recommender systems . The increasing number of web services and users will pose a great challenge to current systems. With O(*mn*) time complexity where *m* is the number of services and *n* the number of users, existing systems cannot generate recom- mendations for tens of thousands users in real time.

The last problem is that current web service recommen- der systems are all black boxes, providing a list of ranked web services with no transparency into the reasoning behind the recommendation results. It is less likely for users to trust a recommendation when they have no knowledge of the underlying rationale. The opaque recommendation approaches prevent the accep- tance of the recommended services. Herlocker et al. [5] mention that explanation capabilities is an important way of building trust in recommender systems, since users are more likely to trust a recommendation when they know the reason behind it.

To address the first two problems, we propose an innovative CF algorithm for QoS-based web service recom- mendation. To address the third problem and enable an improved understanding of the web service recommenda- tion rationale, we provide a personalized map for browsing the recommendation results. The map explicitly shows the QoS relationships of the recommended web services as well as the underlying structure of the QoS space by using map metaphor such as dots, areas, and spatial arrangement.

The main contributions of this work are threefold:

- First, we combine the model-based and memory- based CF algorithms for web service recommenda- tion, which significantly improves the recommenda- tion accuracy and time complexity compared with previous service recommendation algorithms.
- Second, we design a visually rich interface to browse the recommended web services, which enables a better understanding of the service performance.
- Finally, we conduct comprehensive experiments to evaluate our approach by employing real-world web
- service QoS data set. More than 1.5 millions real- world web service QoS records from more than 20 countries are used in our experiments.

The remainder of this paper is organized as follows: Section 2 describes the recommendation approach in detail. Section 3 presents the method for recommendation visua- lization. Sections 4 and 5 show the experiments of the recommendation and visualization, respectively. Section 6 discusses the related work, and Section 7 concludes the paper with a summary and a description of future work.

2 THE RECOMMENDATION APPROACH

A Motivating Scenario

In this section, we present an online service searching scenario to show the research problem of this paper. As Fig. 1

depicts, Alice is a software engineer working in India. She needs an

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 22, April 2016

ascending order of the service average response time. Alice tries the first two services provided by a Canadian company and finds that the response time is much higher than her expectation. She then realizes that the service ranking is based on the evaluation conducted by the registry in US, and the response time of the same service may vary greatly due to the different user context, such as user location, user network conditions, etc. Alice then turns to her colleagues in India for suggestion. They suggest her try service k provided by a local company though ranked lower in the previous recommendation list. After trying it, Alice thinks that service k has a good performance and meets her requirements.

The problem that Alice faces is to find a service that meets both functional and nonfunctional requirements. The current way of finding a suitable web service is rather inefficient, since Alice needs to try the recommended services one by one. To address this challenge, we propose a more accurate approach to service recommendation with consideration of the region factor. Moreover, we try to provide a more informative and user-friendly interface for browsing the recommendation results rather than a ranked list. By this way, users are able to know more about the overall performance of the recommended services, and thus trust the recommendations.

The basic idea of our approach is that users closely located with each other are more likely to have similar service experience than those who live far away from each other. Inspired by the success of Web 2.0 websites that emphasize information sharing, collaboration, and interac- tion, we employ the idea of user-collaboration in our web service recommender system. Different from sharing in- formation or knowledge on blogs or wikis, users are encouraged to share their observed web service QoS performance with others in our recommender system. The more QoS information the user contributes, the more accurate service recommendations the user can obtain, since more user characteristics can be analyzed from the user contributed information.

Based on the collected QoS records, our recommendation approach is designed as a two-phase process. In the first phase, we divide the users into different regions based on their physical locations and historical QoS experience on web services. In the second phase, we find similar users for
the current user and make QoS prediction for the unused services. Services with the best predicted QoS will be recommended to the current user. These two phases are presented in Sections 2.2 and 2.3, respectively, and the time complexity analysis of the proposed algorithm is presented in Section 2.4.

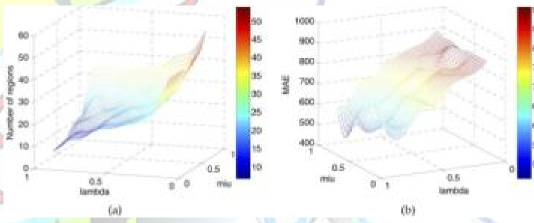
Phase 1: Region Creation

In web service recommender system, users usually provide QoS values on a small number of web services. Traditional memory-based CF algorithms suffer from the sparse user- contributed data set, since it's hard to find similar users without enough knowledge of their service experience. Different from existing methods, we employ the correlation between users' physical locations and QoS properties to solve this problem. In this paper, we focus on the QoS properties that are prone to change and can be easily obtained and objectively measured by individual users, such as response time and availability. To simplify the description of our approach, we use response time (also called round-trip time (RTT)) to describe our approach.

We assume that there are n users and m services. The relationship between users and services is denoted by an $n \times m$ matrix R. Each entry $R_{i;j}$ of the matrix represents the RTT of service j observed by user i and ? is the symbol of no RTT value. Each user i (i 2 f1; 2; ... ; ng) is associated with a row vector R_i representing his/her observed RTT values on different web services. The user aða 2 f1; 2; ... ; ngÞ is called the active user or current user if he/she has provided some RTT records and needs service recommendations.

We define a *region* as a group of users who are closely located with each other and likely to have similar QoS profiles. Each user is a member of exactly one region. Regions need to be internally coherent, but clearly different from each other. The region creation phase is designed as a three-step process. In the first step, we put users with similar IP addresses into a small region and extract region features. In the second step, we calculate the similarity between different regions.

```
Algorithm 1. Region Aggregation
      in: regions r1; .. . ; rl
      out: result list A
  1:  for n    1 to l — 1
  2:        for i   n þ 1 to N
  4:            C½n]½i]:index    i
  5:      end for
  6:      I½n]:sensitivity    ISSENSITIVEðrn) 7:
          if I½n]:sensitivity ¼ 0
  8:          then I½n]:aggregate    1
  9:      else I½n]:aggregate    0
 10:     P½n]   priority queue for C[n] sorted on sim
 11:  end for
 12:  calculate the sensitivity and aggregate of I½l]
 13:  A    []
 14:  while true
 15:          k1     argmaxfk:I½k]:aggregate¼1gP½k]:MAXðÞ:sim
 16:          if k1 ¼ null or sim<μ
 17:              then return A
 18:          k2     P½k1]:MAXðÞ:index
 19:          A.APPEND(<k1; k2>) and comput k1 center 20:
              I½k2].aggregate    0
 21:          P½k1]    ½]
 22:          I½k1]:sensitivity    ISSENSITIVEðk1Þ
 23:          if I½k1]:sensitivity ¼ 1
 24:              then I½k1]:aggregate    0
 25:                  for each i with I½i]:aggregate ¼ 1
 26:                      P½i]:DELETEðC½i]½k1]Þ
 27:                      P½i]:DELETEðC½i]½k2]Þ
 28:                  end for
 29:          else
 30:                for each i with i½i]:aggregate ¼ 1Ki ¼ k1
 31:                    P½i]:DELETEðC½i]½k1]Þ
 32:                    P½i]:DELETEðC½i]½k2]Þ
 33:                    C½i]½k1]:sim    SIMði; k1Þ
 34:                    P½i]:INSERTðC½i]½k1]Þ
 35:                    C½k1]½i]:sim    SIMði; k1Þ
 36:                    P½k1]:INSERTðC½k1]½i]Þ
 37:                end for
 38:  end while
```



Phase 2: QoS Value Prediction

After the phase of region aggregation, thousands of users are clustered into a certain number of regions based on their physical locations and historical QoS similarities. The service experience of users in a region is represented by the region center. With the compressed QoS data, searching neighbors and making predictions for an active user can be computed quickly. Traditionally, the QoS prediction meth- ods need to search the entire data set, which is rather inefficient. In our approach, similarity between the active user and users of a region is computed by the similarity between the active user and the region center. Moreover, it is more reasonable to predict the QoS value for active users based on their regions, for users in the same region are more likely to have similar QoS experience on the same web service, especially on those region-sensitive ones. To predict the RTT value for the active user a on an unused service s, we take the following steps:
of RTT varies largely from service to service. The average RTT of all services provided by user a cannot reveal the performance of a specific web service. Instead, we turn to the RTT profile of the region center and use its RTT of service s to predict the missing value.

Time Complexity Analysis

We discuss the worst-case time complexity of the proposed algorithm. Since there are two phases in our algorithm: the offline phase for region creation and the online phase for the QoS value prediction, we analyze their time complexity separately. We assume the input is a full matrix with n users and m services.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 22, April 2016

In Section 2.2.1, the time complexity of calculating the median and MAD of each service is Oð*nlogn*Þ. For *m* services, the time complexity is Oð*mnlogn*Þ. With MAD and median, we identify the region-sensitive services from the service perspective. Since there are at most *n* records for each service, the time complexity of each service is Oð*n*Þ using definition 1. Therefore, the total time complexity of region-sensitive service identification is Oð*mnlogn* þ *mn*Þ¼ Oð*mnlogn*Þ.
time complexity of the region similarity is Oð*m*Þ using 5), and the complexity for computing similarity matrix C is

where $R_{c_j;s}$ is the RTT of service *s* provided by center c_j, and $\bar{R}_{c_j}$ is the average RTT of center c_j. The prediction is
composed of two parts. One is the RTT value of the region center of the active user $R_{center;s}$, which denotes the average QoS observed by this region users. The other part is the normalized weighted sum of the deviations of the service *s* RTT from the average RTT observed by the *k* most similar neighbors.

- Otherwise, we use the service *s* RTTs observed by the *k* neighbors to compute the prediction as (8) shows. The more similar the active user *a* and the neighbor c_j are, the more weighting the RTT of c_j will carry in the prediction.

Online Time Complexity

Let l_1 be the number of regions after the phase of region creation. To predict the QoS value for an active user, Oðl_1Þ similarity calculations between the active user and region centers are needed, each of which takes Oð*m*Þ time.

However, it is not applicable in our context, since this equation is based on the idea that each user's rating range is subjective and comparatively fixed (e.g., critical users always rate items with lower ratings), whereas the range

3 RECOMMENDATION VISUALIZATION

Conventionally, CF-based web service recommender sys- tems employ the predicted QoS mainly in two ways. 1) When users query a service with specific functionality, the one with the best predicted QoS is recommended to them. 2) Top-*k* best-performing services are recommended to help users discover potential services. While this kind of recommenda- tion is useful, it is not obvious to users why certain services are recommended. More than a service list ranked by predicted QoS as recommendation, we need to develop an exploratory recommendation tool that provides valuable insight into the QoS space and enables an improved understanding of the overall performance of web services.

The QoS space visualization of all web services on a map will reveal the rationale behind QoS-based service recom- mendations. QoS space visualization is more than a picture or method of computing. It transforms the information of high- dimensional QoS data into a visual form enabling service users to observe, browse, and understand the information.

We draw the QoS map by two steps: dimension

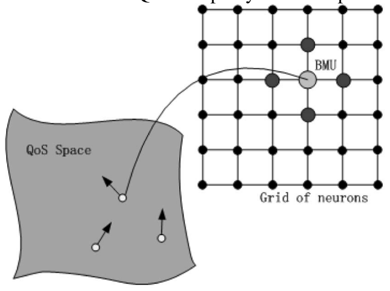


Fig. 2. Mapping QoS space to the 2D output space of SOM. Each QoS vector is mapped to the BMU with closest euclidean distance.

respective QoS values provided by all users (columns). However, since the data set is rather sparse and the number of QoS values varies from

from region centers is employed to train the SOM.

Let *l* denote the dimension of the input space (QoS data) and *q* ¼ ½*q*1; *q*2; . . . ; *ql*]^T denote an input pattern (QoS vector of a service). The prototype of neuron *j* is denoted by

reduction step and map creation step. In the first step, we create a 2D representation of the high-dimensional QoS space by using self-organizing map (SOM), and each web service is mapped to a unique 2D coordinates. In the second step, we create a geographic-like QoS map based on the SOM training results. We detail the two drawing steps in Sections 3.1 and 3.2, respectively.

SOM Training

The SOM [8] is a popular unsupervised artificial neural network that has been successfully applied to a broad range of areas, such as medical engineering, document organiza- tion, and speech recognition. When SOM is used for information visualization, it can be viewed as a mapping of a high-dimensional input space to a lower dimensional output space (usually one or two dimensions).

The output space of SOM is a network of neurons located on a regular, usually 2D grid. Each neuron is equipped with a prototype vector which has the same dimension of the input space. The neurons are connected to adjacent ones by a neighborhood relation indicating the structure of the map such as a rectangular or hexagonal lattice. After the training phase, data points close to each other in the input space are mapped onto the nearby neurons. With this topology

inherent structure of high-dimensional complex data.

The principal goal of using SOM in our context is to transform the QoS data employed by CF into a 2D discrete map in a topologically ordered fashion. In this context, the QoS map is to show the similarity of RTT variance of different web services. Intuitively, the input of the SOM is the QoS matrix containing all the services (rows) and their

initial neighborhood radius and learning rate. After the training of SOM, services with similar QoS are mapped onto the same neuron or nearby neurons. The mapping result of the QoS data reflects the QoS similarities between services.

Map Creation

The direct approach to web service QoS map is to assign each web service a distinct portion of the 2D display area, and put services with similar QoS performance next to each other. With the training result, we first assign each service unique coordinates by randomly distributing them within the cell boundary of the corresponding neuron. Then the Voronoi diagram is used to form a base map in which each service corresponds to a unique polygon.

When applied to a large set of services, base map alone is insufficient and will quickly become too complex to reveal the underlying data relationships. A generalized map explicitly telling the cluster information is needed.

We put web service recommendations on the map by using the predicted QoS values. For those functionally equivalent services, the one with the best predicted QoS will be marked on the map. We also highlight the top-*k* best performing services to help users find potential ones. Section 5 provides the detail of the map implementation.

4 EXPERIMENTS

In this section, we give a comprehensive study on the QoS prediction performance of our proposed algorithm.

Experimental Setup

We adopt a real-world web service QoS performance data set[2] for the experiment. The data set contains about 1.5 million web service invocation records of 100 web services from more than 20 countries. The RTT records are collected by 150 computer nodes from the Planet-Lab,[3] which are distributed over 20 countries. For each computer node, there are 100 RTT profiles, and each profile contains the RTT records of 100 services. We randomly extract 20 profiles from each node, and obtain 3,000 users with RTTs ranging from 2 to 31,407 milliseconds.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 22, April 2016

In this experiment, we randomly remove 90 and 80 percent RTTs of the initial training matrix to generate two sparse matrices with density 0.1 and 0.2, respectively. We vary the number of RTT values given by active users from 10, 20 to 30, and name them given 10, 20, and 30, respectively. The removed records of active users are used to study the prediction accuracy. In this experiment, we set μ ¼ 0:3, Z ¼ 0:8, *top* — *k* ¼ 10. To get a reliable error estimate, we use 10 times 10-fold cross-validation [31] to evaluate the prediction performance and report the average MAE value. Table 1 shows the prediction performance of different methods employing the 0.1 and 0.2 density training matrix. We observe that our method significantly improves the prediction accuracy, and outperforms others consistently. The performance of UPCC, WSRec, and our approach enhances significantly with the increase of matrix density as well as the number of QoS values provided by active users (given number). On the contrary, there is only a slight improvement of IPCC. The original idea of IPCC is to match items with similar user ratings and combine them to recommendations. Apparently, it is not appropriate to apply this idea to our context, because even services provided by the same company are hardly to have similar response times to different users.

The two thresholds Z and μ in the phase of region creation play a very important role in determining the number of regions and thus impact the final performance of our approach. As shown in Algorithm 1, only those regions with similarity higher than μ and sensitivity less than Z are able to be aggregated. In this experiment, we study the impact of Z and μ on a sparse matrix with 2,700 training users and 300 active users. We set density ¼ 0:2, given ¼ 10 and employ all the neighbors with positive PCC for QoS prediction. We vary the two thresholds Z and μ both from 0.1 to 0.9 with a step of 0.1. Fig. 3 shows how Z and μ affect the number of regions and the final performance. It shows that lower μ and higher Z result in fewer regions, but fewer regions does not necessarily mean better prediction accuracy. For this data set, better prediction accuracy is achieved with higher Z and μ.

Note that the optimal value of Z is related to the sensitivity of the original regions at the outset. Fig. 4 shows the distribution of the region sensitivity before aggregation. It shows that the sensitivity of most regions (81.3 percent) is less than 0.1, while the sensitivity of a few regions (4.67 percent) is around 0.8. Higher Z and μ allow very similar regions with high sensitivity to be aggregated and achieve good performance in this experiment.

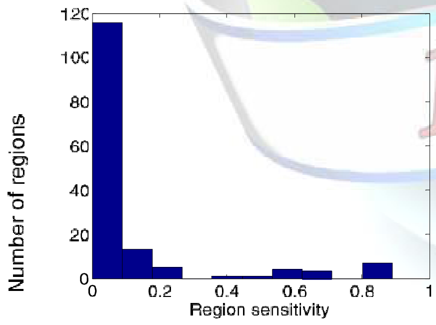


Fig. 5a shows the relation between μ and prediction accuracy with training matrix density 0.2, 0.5, and 1. We employ all the neighbors with positive PCC values for QoS prediction and set Z ¼ 1, so that we do not consider the factor of sensitivity in region aggregation. Similarity becomes the single factor. Obviously, for denser matrix, with higher μ we obtain a set of coherent regions, and better prediction is obtained.

Impact of *Top-k*

Top-k determines how many neighbors are employed in the phase of QoS prediction which relates to the prediction accuracy. Christo Ananth et al. [10] discussed about Reconstruction of Objects with VSN. By this object reconstruction with feature distribution scheme, efficient processing has to be done on the images received from nodes to reconstruct the image and respond to user query. Object matching methods form the foundation of many state- of-the-art algorithms. Therefore, this feature distribution scheme can be directly applied to several state-of- the-art matching methods with little or no adaptation. The future challenge lies in mapping state-of-the-art matching and reconstruction methods to such a distributed framework. The reconstructed scenes can be converted into a video file work can be brought into real time by implementing the code on the server side/mobile phone and communicate with several nodes to collect images/objects. This work can be tested in real time with user query results. We first study the impact of training matrix density. We vary the density of the training matrix from 0.1 to 0.5 with a step of 0.1, and set given ¼ 10. Fig. 5c shows the experimental results. It shows that: 1) with the increase of the training matrix density, the performance of IPCC, UPCC, and our method enhances indicating that better prediction is achieved with more QoS data. WSRec is not sensitive to the data sparseness, and it stays around a certain value. 2) Our method outperforms others consistently.

To study the impact of given number on the prediction results, we employ the training matrix with density 0.3 and vary the given number from 10 to 50 with a step of 10. Significance Weighting

Significance weighting factor is added to devalue similarity weights that are based on a small number of coinvoked web

services. To study the impact of this factor, we implement two versions of the algorithm, one with the significance weighting (5) and the other without (4). Since the over- estimation of the similarity occurs when the active user and training users have few coinvoked services, we study the impact by varying the number of QoS provided by both the training users (training matrix density) and active users (given number) with two experiments.

In the first experiment, we employ 2,700 training users, 300 active users, set given ¼ 5, Z ¼ 0:2, μ ¼ 0:3, top-k ¼ 20. We vary the density of the training matrix from 0.1 to 0.5

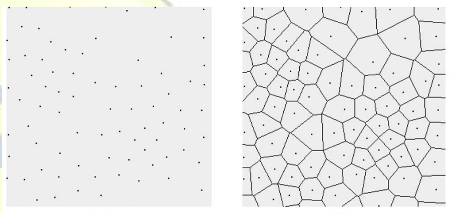


Fig. 6. Map creation. (a) 2D locations of services derived from SOM training. (b) Voronoi diagram of services based on the 2D locations.

with a step of 0.1. As Fig. 5e shows, applying the significance weighting increases the accuracy of the pre- diction algorithm in most cases.

The other experiment is carried out with the same number of training users and active users. We set the density of the training matrix 0.05, Z ¼ 0:2, μ ¼ 0:3, top-*k* ¼ 10. We vary the given number from 10 to 50 by a step of 10. Fig. 5f shows that the algorithm with the significance weighting consistently increases the accuracy of the prediction by a relatively large amount. As the given number increases, the gap between the two becomes more obvious. This is because with a sparse training matrix

obtained 42 clusters. We simplified the base map by merging the neighboring polygons if they are in the same cluster. By this way, we form a generalized map high- lighting the underlying structure of the QoS space.

5 A MAP DISPLAY FOR RECOMMENDATION

In this section, we demonstrate how to create a map showing the similarity of RTT variance of web services, and how to put personalized web service recommendations on the map for an active user. We use 2,700 training users and set given ¼ 10, density ¼ 0:5. After the region aggregation phase, 17 regions are formed. The input of SOM is a 100 × 17 RTT matrix containing 100 services (rows) and their respective performance on 17 regions (columns). Each web service's QoS (row) is an input vector. We train an SOM with neurons arranged in a 60 × 80 hexagonal lattice. The prototypes of SOM are randomly initialized, and Gaussian function is adopted as the neighborhood function (see (13)). We train the SOM in two phases: a rough training phase with initial neighborhood width 15 and learning rate 0.05; a fine tuning phase with initial neighborhood width 2 and learning rate 0.01. The training lengths of the two phases are 500 and 3,000 epochs, respectively. The learning rate decreases linearly to zero during the training.

When the training process is completed, each service is mapped on to (see Fig. 6a). To create a geographic map, each point is assigned to a distinct portion of the map display by forming a Voronoi diagram (see Fig. 6b). After that, we adopt the hierarchical clustering to the services based on their QoS similarities and

With the normalized values, each property will have equal weights in the SOM training. The map creation process is the same, and we can obtain a map reflecting the web service QoS similarity of a specific region.

6 CONCLUSION AND FUTURE WORK

In this paper, we have presented an innovative approach to web service recommendation and visualization. Different from previous work, our algorithm employs the character- istic of QoS by clustering users into different regions. Based on the region feature, a refined nearest-neighbor algorithm is proposed to generate QoS prediction. The final service recommendations are put on a map to reveal the under- lying structure of QoS space and help users accept the recommendations. Experimental results show that our approach significantly improves the prediction accuracy than the existing methods regardless of the sparseness of the training matrix. We also demonstrate that the online time complexity of our approach is better than the traditional CF algorithms.

In this paper, our recommendation approach considered the correlation between QoS records and users' physical locations by using IP addresses, which has achieved good prediction performance. In some cases, however, users in the same physical locations may observe different QoS performance of the same web service. Besides the user physical location, we will investigate more contextual information that influences the client-side QoS perfor- mance, such as the workload of the servers, network conditions, and the activities that users carry out with web services (e.g., web services are used alone or in composi- tion). More investigations on the distribution of RTT and the correlation between different QoS properties such as RTT and availability will be conducted in our web service search engine project ServiceXchange.[4]

For the visualization of the recommendation results, we plan to add more user interactions such as searching web services on the QoS map, zooming in and zooming out. Graphic map like google map will be combined to help users navigate their similar users and web service providers on the map.

User acceptance rate of the recommendation is a key indicator of the effectiveness of the recommender system. We will collect more user feedbacks of the recommendation to help improve the prediction accuracy of our web service recommendation algorithm.

ACKNOWLEDGMENTS
This research work is partly supported by China 863 program MOST (Ministry of Science and Technology) under grant No. 2007AA010301 and the State Key Lab for Software Development Environment under grant SKLSDE- 2010-ZX-03.

REFERENCES

[1] M.B. Blake and M.F. Nowlan, "A Web Service Recommender System Using Enhanced Syntactical Matching," *Proc. Int'l Conf. Web Services*, pp. 575-582, 2007.
[2] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI '98)*, pp. 43-52, 1998.
[3] Y.H. Chen and E.I. George, "A Bayesian Model for Collaborative Filtering," *Proc. Seventh Int'l Workshop Artificial Intelligence and Statistics*, http://www.stat.wharton.upenn.edu/~edgeorge/ Research_papers/Bcollab.pdf, 1999.
[4] S. Haykin, *Neural Networks: A Comprehensive Foundation*, second ed. Prentice-Hall, 1999.
[5] J.L. Herlocker, J.A. Konstan, and J. Riedl, "Explaining Collabora- tive Filtering Recommendations," *Proc. ACM Conf. Computer Supported Cooperative Work*, pp. 241-250, 2000.
[6] J. Himberg, "A SOM Based Cluster Visualization and Its Application for False Coloring," *Proc. IEEE-INNS-ENNS Int'l Joint Conf. Neural Networks*, pp. 587-592, 2000, vol. 3, doi:10.1109/ IJCNN.2000.861379.
[7] Hsu, and S.K. Halgamuge, "Class Structure Visualization with Semi-Supervised Growing Self-Organizing Maps," *Neurocomput- ing,*

pp. 1464-1480, Sept. 1990.
[9] S. Kaski, J. Venna, and T. Kohonen, "Coloring that Reveals High-Dimensional Structures in Data," *Proc. Sixth Int'l Conf. Neural Information Processing*, vol. 2, pp. 729-734, 1999.
[10] Christo Ananth, M.Priscilla, B.Nandhini, S.Manju, S.Shafiqa Shalaysha, "Reconstruction of Objects with VSN", International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Vol. 1, Issue 1, April 2015, pp:17-20
[11] G. Linden, B. Smith, and J. York, "Amazon.com Recommenda- tions: Item-to-Item Collaborative Filtering," *IEEE Internet Comput- ing*, vol. 7, no. 1, pp. 76-80, Jan./Feb. 2003.
[12] Z. Maamar, S.K. Mostefaoui, and Q.H. Mahmoud, "Context for Personalized Web Services," *Proc. 38th Ann. Hawaii Int'l Conf.*, pp. 166b-166b, 2005.
[13] M.R. McLaughlin and J.L. Herlocker, "A Collaborative Filtering Algorithm and Evaluation Metric That Accurately Model the User Experience," *Proc. Ann. Int'l ACM SIGIR Conf.*, pp. 329-336, 2004.
[14] B. Mehta, C. Niederee, A. Stewart, C. Muscogiuri, and E.J. Neuhold, "An Architecture for Recommendation Based Service Mediation," *Semantics of a Networked World*, vol. 3226, pp. 250-262, 2004.
[15] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl, "MovieLens Unplugged: Experiences with an Occasionally Con- nected Recommender System," *Proc. ACM Int'l Conf. Intelligent User Interfaces*, pp. 263-266, 2003.
[16] C.D. Mining, P. Raghavan, and H. Schü tze, *An Introduction to Information Retrieval.* Cambridge Univ., 2009.
[17] C. Zhao, C. Ma, J. Zhang, J. Zhang, L. Yi, and X. Mao, "HyperService: Linking and Exploring Services on the Web," *Proc. Int'l Conf. Web Services*, pp. 17-24, 2010.
[18] E. Rich, "User Modeling via Stereotypes," *Cognitive Science*, vol. 3, no. 4, pp. 329-354, 1979.
[19] W. Rong, K. Liu, and L. Liang, "Personalized Web Service Ranking via User Group Combining Association Rule," *Proc. Int'l Conf. Web Services*, pp. 445-452, 2009.
[20] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS Prediction for Web Services via Collaborative Filtering," *Proc. Int'l Conf. Web Services*, pp. 439-446, 2007.
[21] A. Skupin, "A Cartographic Approach to Visualizing Conference Abstracts," *Computer Graphics and Applications*, vol. 22, no. 1, pp. 50-58, 2002.
[22] R.M. Sreenath and M.P. Singh, "Agent-Based Service Selection," *J. Web Semantics.* vol. 1, no. 3, pp. 261-279, 2003, doi:10.1016/ j.websem.2003.11.006.
[23] K. Tasdemir and E. Merényi, "Exploiting Data Topology in Visualization and Clustering of Self-Organizing Maps," *IEEE Trans. Neural Networks*, vol. 20, no. 4, pp. 549-562, Apr. 2009.
[24] A. Ultsch, "U*-Matrix: A Tool to Visualize Clusters in High Dimensional Data," Technical Report 36, CS Department, Phi- lipps-Univ., 2004.
[25] A. Ultsch and H.P. Siemon, "Kohonen's Self-Organizing Feature Maps for Exploratory Data Analysis," *Proc. Int'l Neural Networks Conf.*, pp. 305-308, 1990.
[26] L.H. Ungar and D.P. Foster, "Clustering Methods for Collabora- tive Filtering," *Proc. AAAI Workshop Recommendation Systems*, 1998.
[27] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 586-600, May 2000.
[28] J. Zhang, H. Shi, Y. Zhang, "Self-Organizing Map Methodology and Google Maps Services for Geographical Epidemiology Mapping," *Proc. Digital Image Computing: Techniques and Applica- tions*, pp. 229-235, 2009, doi:10.1109/DICTA.2009.46.
[29] Z. Zheng, H. Ma, M.R. Lyu, and I. King, "WSRec: A Collaborative Filtering Based Web Service Recommendation System," *Proc. Int'l Conf. Web Services*, pp. 437-444, 2009.
[30] F. Aurenhammer, "Voronoi Diagrams—A Survey of a Funda-

[31] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques,* second ed. Elsevier, 2005.

[32] P.J. Rousseeuw and C. Croux, "Alternatives to the Median Absolute Deviation," *J. Am. Statistical. Assoc.,* vol. 88, no. 424, pp. 1273-1283, 1993.

[33] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing.* Springer and Tsinghua Univ., 2007.

[34] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints," *ACM Trans. Web,* vol. 1, no. 1, pp. 1-26, 2007.

[35] S. Rosario, A. Benveniste, S. Haar, and C. Jard, "Probabilistic QoS and Soft Contracts for Transaction-Based Web Services Orchestra- tions," *IEEE Trans. Services Computing,* vol. 1, no. 4, pp. 187-200,
Oct. 2008.

[36] X. Chen, X. Liu, Z. huang, and H. Sun, "RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation," *Proc. Int'l Conf. Web Services,* pp. 9-16, 2010.

[37] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, "Similarity Search for Web Services," *Proc. 30th Int'l Conf. Very Large Data Bases,* pp. 372-383, 2004.

[38] X. Liu, G. Huang, and H. Mei, "Discovering Homogeneous Web Service Community in the User-Centric Web Environment," *IEEE Trans. Services Computing,* vol. 2, no. 2, pp. 167-181, Apr.-June 2009.

[39] E.M. Maximilien and M.P. Singh, "A Framework and Ontology for Dynamic Web Services Selection," *IEEE Internet Computing,*
vol. 8, no. 5, pp. 84-93, Sept. 2004.

[40] 68-95-99.7 Rule, http://en.wikipedia.org/wiki/68-95-99.7_rule,
2012.