International Journal of and Research (IJSR)
ISSN (Online): 2319-7064

An Effective Method to Predict the Accuracy of QoS by Using User Acceptance Rate

Saradhambal .G[1], Priya Radhika Devi .T[2]

[1, 2] Mailam Engineering College, Department of Computer Science and Engineering, TamilNadu, India

Abstract: ***In web services the effective QoS-based approach to service recommendation is becoming more and more important. The Performance in existing is not satisfactory even when the service recommendation is studied in recent literature 1) previous recommender systems are all black boxes providing limited information on the performance of the service candidates 2) previous approaches fail to consider the QoS variance according to users' locations. In this project, for large scale web service recommendation a collaborative filtering is designed. This approach employs the characteristic of QoS and achieves considerable improvement on the recommendation accuracy compared with previous approach. To help service users better understand the rationale of the recommendation and remove some of the mystery, 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.***

Keywords: Service recommendation, QoS, collaborative filtering, self-organizing map, visualization

1. Introduction

Web services are software components designed to support Interoperable machine-to-machine interaction over a network. Web service is a software function provided at the network address over the web or in the cloud. Delivery mode in business has fostered a new paradigm shift from the development of monolithic applications to the dynamic setup of business process. Web service is the communication of two electronic components over the World Wide Web. 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, service engineers (also called service users) usually get a list of web services from service brokers or search engines that meet the specific functional requirements. 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 [1], [2], [3]. QoS is important to the service providers. Because of the dynamic and unpredictable characteristics of web service it is not easy to provide the quality of service for web service users.

The objective of this project 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 [4], [5], [6], [7] has applied collaborative filtering (CF) to web service recommendation. The first problem is that the existing approaches fail to recognize the QoS variation with users' physical locations.

The second problem is the online time complexity of memory-based CF recommender systems [5], [9]. 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 recommendations for tens of thousands users in real time.

The last problem is that current web service recommender systems are all black boxes, providing a list of ranked web services with no transparency into the reasoning behind the recommendation results [4], [5], [6], [7], [8]. 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 acceptance of the recommended services. To address the first two problems, an innovative CF algorithm for QoS-based web service recommendation. To address the third problem and enable an improved understanding of the web service recommendation rationale, 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:
1. First, combine the model-based and memory-based CF algorithms for web service recommendation, which significantly improves the recommendation accuracy and time complexity compared with previous service recommendation algorithms.
2. Second, design a visually rich interface to browse the recommended web services, which enables a better understanding of the service performance.
3. Finally, 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.

2. The Recommendation Approach

2.1 A Motivating Scenario

In this section, an online service searching scenario is illustrated to show the research problem of this paper.

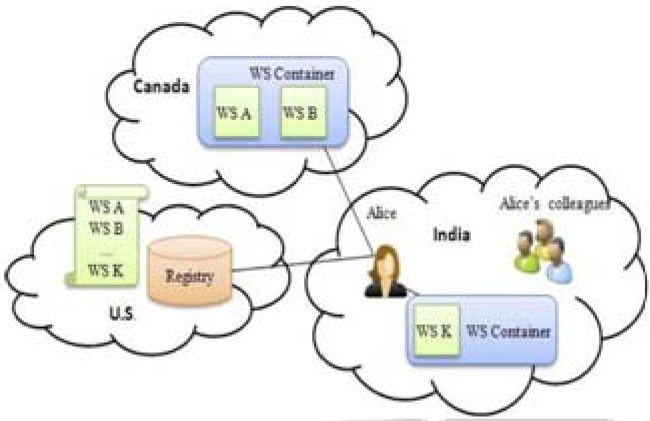

Figure 1: Alice's situational problem.

As Fig. 1 depicts, Alice is a software engineer working in India. She needs an email validation service to filter emails. After searching a service registry located in US, she gets a list of recommended services in 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.

To address this challenge, 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.

2.2 Phase 1: Region Creation

In web service recommender system, users usually provide QoS values on a small number of web services. To simplify the description of our approach, use response time (also called round-trip time (RTT)) to describe the approach. Then 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.

2.2.1Region Feature Extraction

MAD is defined as the median of the absolute deviations from the sample's median

$$\mathrm{MAD} = median_i(|R_{i,s} - median_j(R_{j,s})|) \quad (1)$$
$$i = 1, \ldots, k, j = 1, \ldots, k.$$

$$\hat{\mu} = median_i(R_{i,S}) \; i = 1, \ldots, k, \quad (2)$$

$$\hat{\sigma} = MAD_i(R_{i,S}) \; i = 1, \ldots, k. \quad (3)$$

2.2.2Computation For Region Similarity

Pearson Correlation Coefficient (PCC) is widely used in recommender systems to calculate the similarity of two users [9]. PCC values ranges from -1 and 1. Positive PCC value indicates that the two users have similar preferences, while negative PCC value means that the two user preferences are opposite.

2.2.3 Region Aggregation

Each region formed by users' physical locations at the outset always has a very sparse QoS data set, since users only use a small number of web services and provide limited QoS records. In this case, it is difficult to find similar users and predict the QoS values of the unused web services for the active user. To solve this problem, a region aggregation method based on the region aggregation.

2.3 Phase 2: QoS Value Prediction

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 methods need to search the entire data set [5], [7] which is rather inefficient. The average RTT of all services provided by user can't 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 [10].

2.4 Time Complexity Analysis

2.4.1 Offline Time Complexity

The time complexity of calculating the median and MAD of each service is O (nlogn). With MAD and median, identify the region-sensitive services from the service perspective. Therefore, the total time complexity of region-sensitive service identification is O (mnlogn + mn) = O (mnlogn).

2.4.2 Online Time Complexity

To predict the QoS value for an active user, O (l_1) similarity calculations between the active user and region centers are needed, each of which takes O (m) time.

3. Recommendation Visualization

The QoS space visualization of all web services on a map will reveal the rationale behind QoS-based service recommendations. 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. The QoS map can be designed by two steps: dimension reduction step and map creation step.

3.1 SOM Training

The SOM [11] is a popular unsupervised artificial neural network that has been successfully applied to a broad range of areas, such as medical engineering, document organization, 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).

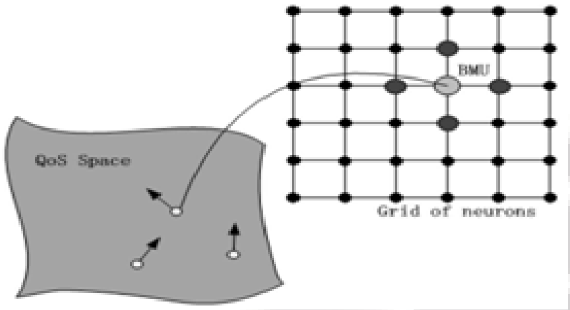

Figure 2: Mapping QoS space to the 2D output space of SOM

3.2 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 [12], [13] is used to form a base map in which each service corresponds to a unique polygon. 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. Then also highlight the top-k best performing services to help users find potential ones.

4. Experiments

4.1 Experimental Setup

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, 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. Then randomly extract 20 profiles from each node, and obtain 3,000 users with RTTs ranging from 2 to 31,407 milliseconds. To evaluate the prediction performance, compare our approach with user-based CF algorithm using PCC (UPCC) [5], item-based CF algorithm using PCC (IPCC) [14], and WSRec [7] which combines UPCC and IPCC.

4.2 Prediction Evaluation

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.

4.3 Data Sparseness

This experiment investigates the impact of data sparseness on the prediction accuracy. We examine the impact from two aspects: the density of training matrix and the number of QoS values given by active users (given number). We divide the experiment into two parts and use 10 times 10-fold cross-validation to assess the prediction results and report the average MAE. 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.

4.4 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 and the other without. 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.

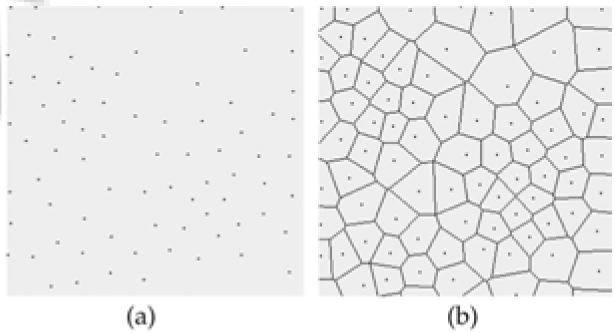

Figure 3: Map creation. (a) 2D locations of services derived from SOM training. (b) Voronoi diagram of services based on the 2D locations.

5. A Map Display Recommendation

When the training process is completed, each service is mapped on to a neuron. Assign unique coordinates to each service by randomly distributing them within the boundaries of the corresponding neuron cell (see Fig. 3a). To create a geographic map, each point is assigned to a distinct portion of the map display by forming a Voronoi diagram (see Fig.3b). After that, adopt the hierarchical clustering to the services based on their QoS similarities and obtained 42 clusters. Then simplified the base map by merging the neighboring polygons if they are in the same cluster. By this way, form a generalized map high-lighting the underlying structure of the QoS space.

Labeling individual service is an integral part of the map creation. The goal is to help users identify the potential services with optimal QoS values. Use different label styles to mark services showing how strongly we recommend them. For example, the top 10 best performing services are labeled with 12-point boldface; services with good predicted QoS are labeled with 8-point nonboldface; while dots indicate those services with poor predicted QoS values. Fig. 4 shows the final map for web service recommendations.

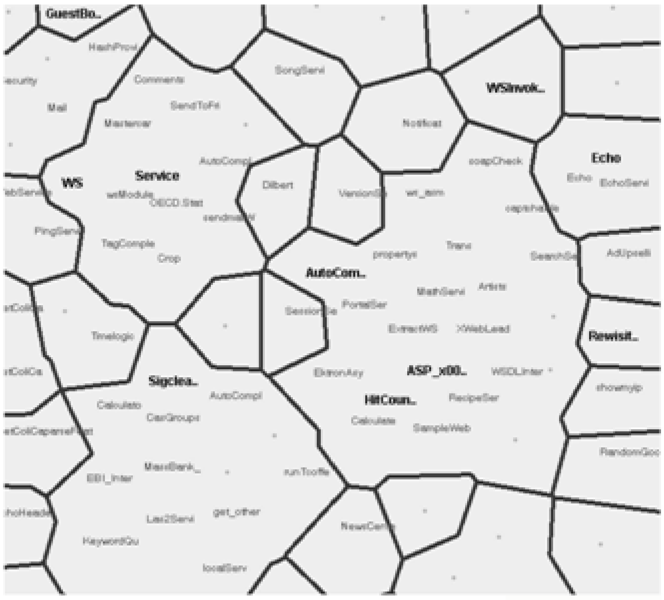

Figure 4: Final map with service recommendations

The map can also be created for one region to show the similarity of web services based on a set of QoS properties. For example, we can employ three QoS properties: RTT, cost, and reputation. RTT and cost are quantitative properties, while reputation is qualitative (usually 1-5 starts). Each region center is a matrix composed of three column vector (RTT, Cost, Reputation) as Fig. 5 shows. Since different properties have different ranges, we first normalize each of them to [0, 1] by the following steps: 1) find the minimum R_{min} and maximum R_{max} value of the property; 2) for each original value R submitted by the user, the normalized value R'.

	RTT	Cost	Reputation
WS_1	1200	10	1
WS_2	1000	20	2
...	...	...	...
WS_{100}	800	50	5

Figure 5: Region center matrix.

6. Related Work

6.1 Collaborative Filtering

Collaborative Filtering is first proposed by Rich [15] and widely used in commercial recommender systems, such as Amazon.com [16]. The basic idea of CF is to predict and recommend the potential favorite items for a particular user by leveraging rating data collected from similar users. Essentially, CF is based on processing the user-item matrix. Breese et al. [17] divided the CF algorithms into two broad classes: memory-based algorithms and model-based algorithms. Memory-based algorithms such as user-based KNN [18], [19] use the entire user-item matrix when computing recommendations. These algorithms are easy to implement, require little or no training cost, and can easily take new users' ratings into account. However, memory-based algorithms cannot cope well with large number of users and items, since their online performance is often slow.

Alternatively, model-based algorithms, such as K-means clustering [20], Bayesian model [21], etc., learn the model from the data set using statistical and machine learning techniques. These model-based algorithms can quickly generate recommendations and achieve good online performance. However, the model must be performed anew when new users or items are added to the matrix.

6.2 Web Service Recommendation

Web service discovery is a hot topic which plays a crucial role in the area of services computing [1]. Some syntactic and semantic-based web service search engines have been proposed in the recent literature. Dong et al. [22] found that the traditional key word-based web service search was insufficient, and they provided a similarity search algorithm for web services underlying the Woogle search engine. Liu et al. [23] investigated the similarity measurement of web services and designed a graph-based search model to find web services with similar operations. Recommendation techniques have been used in recent research projects to enhance web service discovery. Mehta et al. [24] found that semantics and syntax were inadequate to discover a service that meets user requirements. They added two more dimensions of service description: quality and usage pattern. Based on this service description, they propose the service mediation architecture. Blake and Nowlan [25] computed a web service recommendation score by matching strings collected from the user's operational sessions and the description of the web services. Based on this score, they judged whether a user is interested in the service. Based on the input keywords, users can get a set of recommendations with linkages to the query. Previous work mainly focused on providing a mechanism to formalize users' preference, resource, and the description of web services, and recommendations are generated based on the predefined semantic models. Different from these methods, our recommendations are generated by mining the QoS records that are automatically collected from interactions between users and services.

Limited work has been done to apply CF to web service recommendation. Shao et al. [5] proposed a user-based CF algorithm to predict QoS values. Zheng et al. [7] combined the user-based and item-based CF algorithm to recommend web services. However, since neither of the two approaches recognized the different characteristic between web service QoS and user ratings, the prediction accuracy of these methods was unsatisfactory. Sreenath and Singh [6] and Rong et al. [4] applied the idea of CF in their systems, and used MovieLens data [19] for experimental analysis. However, using the movie data set to study web service recommendation is not convincing.

There are several SOM-based methods to visualize data structure. U-Matrix is the most popular one that displays the local distance structures of the input vectors. U*-Matrix, an enhancement of U-Matrix, combines the density and distance information for visualization. Color assignment is also employed to show the approximate cluster structures [26], [27]. To exploit data topology in visualization, Tasdemir and Mere´nyi [28] introduced a weighted Delaunay triangulation. Different from our approach that directly clusters the web services to form a generalized map, data clusters can be computed by applying clustering techniques to the trained prototypes, and clusters can be visualized on top of the map [12].

7. Conclusion and Future Work

In this an innovative approach to web service recommendation and visualization is presented. Different from previous work, our algorithm employs the characteristic 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 underlying 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. Then also demonstrate that the online time complexity of our approach is better than the traditional CF algorithms.

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 performance, 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 composition). 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.

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 recommendation.

References

[1] L.-J. Zhang, J. Zhang, and H. Cai, Services Computing. Springer and Tsinghua Univ., 2007.
[2] 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.
[3] S. Rosario, A. Benveniste, S. Haar, and C. Jard, "Probabilistic QoS and Soft Contracts for Transaction-Based Web Services Orchestrations," IEEE Trans. Services Computing, vol. 1, no. 4, pp. 187-200,Oct. 2008.v
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] X. Chen, X. Liu, Z. huang, and H. Sun, "Region KNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation," Proc. Int'l Conf. Web Services, pp. 9-16, 2010.
[9] 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.
[10] Hsu, and S.K. Halgamuge, "Class Structure Visualization with Semi-Supervised Growing Self-Organizing Maps," Neuro computing, vol. 71, pp. 3124-3130, 2008.
[11] T. Kohonen, "The Self-Organizing Map," Proc. IEEE, vol. 78, no. 9, pp 1464-1480, Sept. 1990.
[12] A. Skupin, "A Cartographic Approach to Visualizing Conference Abstracts," Computer Graphics and Applications, vol. 22, no. 1, 50-58, 2002
[13] F. Aurenhammer, "Voronoi Diagrams—A Survey of a Funda-mental Geometric Data Structure," ACM Computing Surveys, vol. 23, no. 3, pp. 345-405, 1991.
[14] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations:Item-to-ItemCollaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan./Feb. 2003.
[15] E. Rich, "User Modeling via Stereotypes," Cognitive Science, vol. 3, no. 4, pp. 329-354, 1979.
[16] C.D. Mining, P. Raghavan, and H. Schu¨tze, An Introduction to Information Retrieval. Cambridge Univ., 2009.
[17] 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.
[18] 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.
[19] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl, "MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System," Proc. ACM Int'l Conf. Intelligent User Interfaces, pp. 263-266, 2003.
[20] L.H. Ungar and D.P. Foster, "Clustering Methods for

Collaborative Filtering," Proc. AAAI Workshop Recommendation Systems, 1998.

[21] 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.

[22] 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.

Author Profile

Saradhambal. G received B.E degree in Computer Science from Anna University 2012. Currently doing M.E degree in Computer Science and Engineering and project work related to web services.

Priyaradhikadevi. T received M. Tech degree in Computer Science from Anna University. Pursuing Ph. D and published many international and national journals .Two books has been published. Participated in many national and international conferences.