Journal of Web Engineering, Vol. 15, No.5&6 (2016) 397-411

STUDYING THE PERFORMANCE OF QoS SPECIFIC WEB SERVICE RECOMMENDATION SYSTEM USING VIRTUAL REGIONS

KALPANA R, SARULADHA K, JAYABHARATHY J

Pondicherry Engineering College, India

{rkalpana, charuladha, bharathyraja}@pec.edu

Received November 13, 2014
Revised December 25, 2015

The emergence of Internet and web services made a tremendous impact on the data retrieval of the users. The users are sophisticated to utilize the web services based on the recommendation systems. There are various categories of recommendation system reported in the literature. Web Service Recommender Systems (WSRS) based on Collaborative Filtering (CF) achieves best Quality of Service (QoS) results. The users are classified based on the similar IP addresses and regions are created. In this paper a new recommendation system is proposed where the users are classified and grouped together based on virtual regions. The virtual regions are found based on the QoS parameters computed based on round trip time of the services. The proposed Virtual Region based Filtering algorithm (VRF) for web service recommendation significantly improves the prediction accuracy and time complexity. The developed approach is tested with real-world web service QoS data sets. The proposed system achieves improved performance with respect to the parameters such as Round Trip Time (RTT) and Mean Absolute Error (MAE).

Key words: Web Service Recommendation System, Filtering approach, QoS requirements, Region Similarity.
Communicated by: M. Gaedke & T. Tokuda

1 Introduction

Web services are software components designed to support interoperable machine-to-machine interaction over a network. In business process applications and business intelligence applications, the need for web services is detrimentally growing everyday and it made a new paradigm shift from the development of monolithic applications to the dynamic setup of business logic applications. The web services have attracted wide attentions from both industry and academia, and the number of public web services is steadily increasing every day.

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. The service engineers in turn will identify the optimal service from the functionally equivalent candidates. The optimal web service identification is difficult within the given constraints as the service users will not be aware of the performance of each service uniquely. Hence, there is a need to identify effective approaches or techniques to service selection and recommendation.

QoS is the broad collection of measures which evaluates networking technologies and techniques. The goal of QoS is to provide guarantees on the ability of the network to deliver predictable results. Elements of the network performance within the scope of QoS often include availability(uptime), bandwidth(throughput), latency(delay) and error rate. With the widespread proliferation of web services, QoS will become a significant factor in distinguishing the success of the service providers. QoS determines the service usability and utility, both of which influence the popularity of the service. 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 process. Some QoS properties like reputation and reliability are difficult to be evaluated, since long-duration observation and a number of invocations are required.

The major contributions are:

1) A QoS based Web Service Recommendation System is proposed which considers both functional and non-functional requirements. This approach not only takes the physical location into consideration but all the factors which results into a set of RTT value for a particular web service.
2) Virtual Region set formed by this approach is better classified which results in more accurate recommendation. For similarity computation RTT is calculated for individual regions.
3) The model-based and memory based CF algorithms for web service recommendation, which significantly improves the recommendation accuracy and time complexity compared with earlier web service recommendation algorithms.

The remainder of this paper is organized as follows: Section 2 elaborates the current research in QoS based web service recommendation system. Section 3 explains the architecture of the system based on virtual regions. In Section 4 the results are tabulated and analysed. Section 5 concludes with future findings.

2 Current Practice and Research

Service-oriented computing (SOC) [2] enables organizations and individual users to discover openly-accessible capabilities realized as services over the Internet. However, service registries can potentially be very large preventing organizations from discovering services in real-time. In fact, consumers may not be aware of the services that can be of most benefit to them. A web service recommender system that proactively discovers and manages web services was introduced. As an innovation, real, fully-operational web services currently available on the Internet have been analyzed. Using these general naming tendencies coupled with enhanced syntactical methods, it is able to aggregate services by their messages and accurately suggests candidate services to users as a part of daily routines.

Collaborative filtering [5], [9] or recommender systems use a database about user preferences to predict additional topics or products a new user might like. Several algorithms designed for this task, including techniques based on correlation coefficients, vector-based similarity calculations, and statistical Bayesian methods are described. The predictive accuracy of the various methods in a set of representative problem domains is compared. Two basic classes of evaluation metrics are used. The first characterizes accuracy over a set of individual predictions in terms of average absolute deviation. The second estimates the utility of a ranked list of suggested items. This metric uses an estimate of the probability that a user will see a recommendation in an ordered list. Experiments were run for datasets associated with 3 application areas, 4 experimental protocols, and the 2 evaluation metrics for the various algorithms. Results indicate that for a wide range of conditions, Bayesian networks with decision trees at each node and correlation methods outperform Bayesian-clustering and vector-similarity methods. Between correlation and Bayesian networks, the preferred method depends on the nature of the dataset, nature of the application (ranked versus one-by-one presentation), and the availability of votes with which to make predictions. Other considerations include the size of database, speed of predictions, and learning time [12].

Web service plays an important role in implementing Service Oriented Architecture (SOA) for achieving dynamic business process. With the increased number of Web services advertised in public repository, it is becoming vital to provide an efficient Web service discovery and selection mechanism with respect to a user requirement. Considerable efforts have been made to solve this problem among which semantic based Web service discovery has been attained much importance by researchers in academic and industry community. However, there is a challenge in the semantic based web service discovery process, that is, among the retrieved set of semantically equivalent Web service candidates. Inspired by collaborative filtering idea, a web service ranking framework is proposed in which a set of users with similar interest will be firstly identified. Afterwards, association rules will be found out by analyzing all Web service composition transactions related to that set of users. By combining user group and association rule mined from that group, a personalized Web service ranking mechanism is achieved [8].

Service-Oriented Architecture (SOA) provides a flexible framework for service composition. The standard-based protocols (such as SOAP and WSDL) will help for composite services which can be constructed by integrating atomic services developed independently. Algorithms are needed to select service components with various QoS levels according to some application-dependent performance requirements. a broker-based architecture is designed to facilitate the selection of QoS-based services. The objective of service selection is to maximize an application-specific utility function under the end-to-end QoS constraints. The problem is modeled into two ways: the combinatorial model and the graph model. The combinatorial model defines the problem as a Multi dimension Multi choice 0-1 Knapsack Problem (MMKP). The graph model defines the problem as a Multi Constraint Optimal Path (MCOP) problem [3].

Web is becoming a programmable platform, with countless services blooming every day in various forms like Feeds, REST APIs and Widgets, etc. Although the existing technologies, such as Mashups, have reduced the challenges to build new applications by composing these services, it's still far from enabling the non-technical users to solve their situational problems by correlating and consuming these services. Hyper Service technology empowers a much more flexible way to link and explore existing services for solving various situational problems. In Hyper Service, the service metadata, service linkages and user behaviors are indexed and managed; Based on the user's input keywords and navigation context, a group of relevant services are dynamically searched, ranked and recommended for facilitating future navigations; the service navigation is smoothed by a web2.0 style exploratory user interface. A prototype system is also presented to demonstrate the effectiveness of our Hyper Service research work. [13]

Consumers need to make prediction on quality of unused web services before selecting. Usually, this prediction is based on other consumers' experiences. Being aware of different QoS experiences of consumers, a collaborative filtering based approach is proposed to make similarity mining and prediction from consumers' experiences. This approach can make significant improvement on the effectiveness of QoS prediction for web services. [9]

As the abundance of Web services on the World Wide Web increase, designing effective approaches for Web service selection and recommendation has become more and more important. A Web service recommender system (WSRec) has been created to attack this crucial problem. WSRec includes a user-contribution mechanism for Web service QoS information collection and an effective and novel hybrid collaborative filtering algorithm for Web service QoS value prediction. WSRec is implemented by Java language and deployed to the real-world environment. To study the prediction performance, a total of 21,197 public Web services are obtained from the Internet and a large-scale real-world experiment is conducted, where more than 1.5 million test results are collected from 150 service users in different countries on 100 publicly available Web services located all over the world. The comprehensive experimental analysis shows that WSRec achieves better prediction accuracy than other approaches [11].

Web services are loosely coupled software components, published, located, and invoked across the web. The growing number of web services available within an organization and on the Web raises a new and challenging search problem: locating desired web services. Traditional keyword search is insufficient in this context: the specific types of queries users require are not captured, the very small text fragments in web services are unsuitable for keyword search, and the underlying structure and semantics of the web services are not exploited. [14]

The Self-Organizing Map (SOM) is a powerful method for visualization, cluster extraction, and data mining. It has been used successfully for data of high dimensionality and complexity where traditional methods may often be insufficient. In order to analyse data structure and capture cluster boundaries from the SOM, one common approach is to represent the SOM's knowledge by visualization methods. Different aspects of the information learned by the SOM are presented by existing methods, but data topology, which is present in the SOM's knowledge, is greatly underutilized. Data topology can be integrated into the visualization of the SOM and thereby provide a more elaborate view of the cluster structure than existing schemes. It has been achieved by introducing a weighted Delaunay triangulation (a connectivity matrix) and draping it over the SOM. This new visualization, CONNvis, also shows both forward and backward topology violations along with the severity of forward ones, which indicate the quality of the SOM learning and the data complexity. CONNvis greatly assists in detailed identification of cluster boundaries. We demonstrate the capabilities on synthetic data sets and on a real 8D remote sensing spectral image [6], [15], [16].

The health Geographical Information System (GIS) has been used in many organizations for the management and visualization of public health data. As epidemiology information has become a part of health data repository in the health data management system, many health researchers have dedicated their research areas to geographical epidemiology information analysis and visualization. The Population Health Epidemiology Unit of the Department of Health and Human Services (DHHS) in Tasmania uses the Web-based Epidemiology system (`WebEpi') to conduct monitoring and surveillance of the health of Tasmanian population. The epidemiology data self-organizing map (SOM) have been used as a tool to recognize patterns with data sets measuring epidemiology data and related geographical information. Google Maps services offer Web GIS Application Programming

Interface (API) and GIS views. The integration of SOM and Google Maps facilitates the epidemiology data pattern recognition and geo-visualization which enables health research to be conducted in a novel and effective way [16].

The QoS properties such as response time and availability are prone to change and they could be determined and objectively measured by individual users [1]. 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*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.

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. In the last step, we aggregate highly correlated regions to form a certain number of large regions.

3 Architecture of Web Service Recommendation System (WSRS) based on Virtual Regions

The basic idea of this new system is that users having similar web experiences should be grouped together. Inspired by the success of Web 2.0 websites that emphasize information sharing, collaboration, and interaction, we employ the idea of user-collaboration in our web service recommender system. Different from sharing information 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 the recommendation accuracy could be improved considerably.

We present an online service searching scenario to show the research problem of this paper. As Figure 1 depicts, John is a software engineer working in India. He needs a file hosting service. After searching a service registry located in US, He gets a list of recommended services in ascending order of the service average response time. John tries the first two services provided by a Russian company and finds that the response time is much higher than his expectation. He 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. The problem that John 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 John needs to try the recommended services one by one.

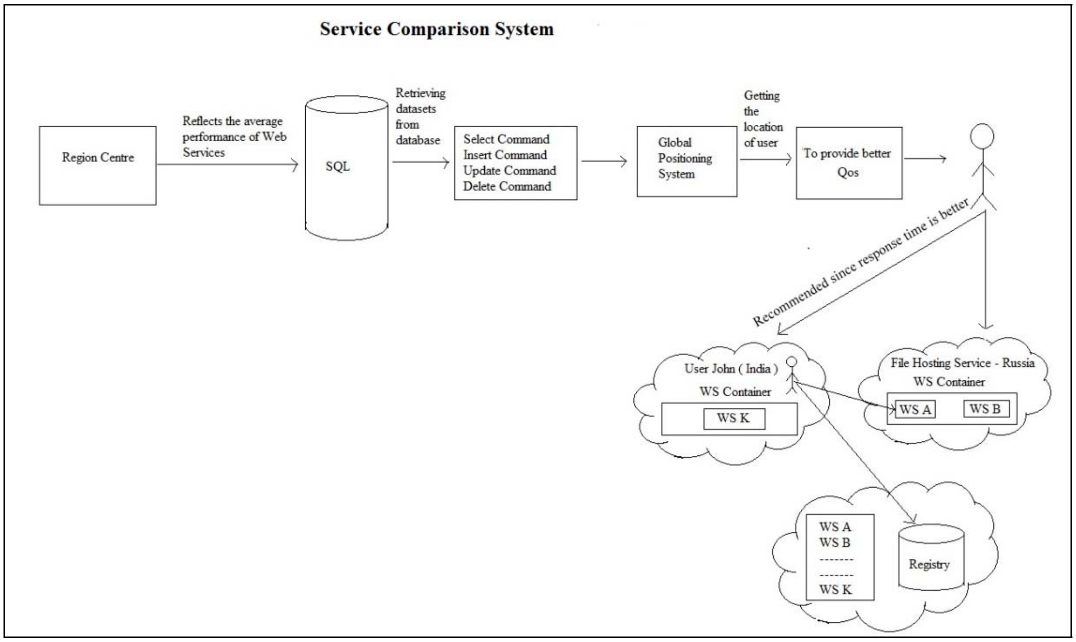

Figure 1. Scenario Illustrating Recommender System without Region Factor

To address this challenge, a more accurate approach to service recommendation with consideration of the region factor is proposed. By this way, users are allowed to know more about the overall performance of the recommended services, and thus trust the recommendations.

The Web Service Recommendation System (WSRS) architecture given in Figure 2 uses personalized web services for the end users. The architecture depicts how the user will be put into the virtual region and which parameters contribute together to place the user into the virtual region. We will have many numbers of users such as (U_1,U_2,U_3…U_n) and the predefined set of services such as (S1,S2,S3….Sn). In this architecture diagram, consider some sample services, such as mail services, file services and image services. Each user will be utilizing all the services and the response time (i.e.) Round Trip Time (RTT) should be calculated for each services by the individual users. Services should be categorized into most used services, standard services and least used services. When the first user wishes to join a virtual region, there is no virtual region formed at this time. So the first user is joined in the first virtual region without any comparison. When a second user enters the system, its RTT vector is compared with the first user of each virtual region by using the Pearson Correlation Coefficient (PCC). The advantage of this new system over the earlier regions based WSRS is the improved time complexity and the better Quality of Service (QoS), since the user has been categorized based on their RTT value experienced from the services.

However, the response time of the service will be affected by the parameters namely, such as user location, user network conditions, etc. These parameters which will enforce a service delay in joining the users to the virtual region can be modelled as follows: Harsanyi (1955) frames the group utility function G_u in equation which is adapted as virtual region servicing function.

$$G_u(X) = \sum_{i=1}^{n} K_i U_i (x_i) \tag{1}$$

where $U_i(x)$ is the i th user's preferences for the service x_i of some alternative, U_i value is computed and lies in the range between 0 and 1. And the k_i s are constants which reflect the relative user context weight accorded to the i^{th} user's preferences and n is the number of users in the virtual region. The computation of the user context weights are based on parameters like user location, user network conditions, etc.,

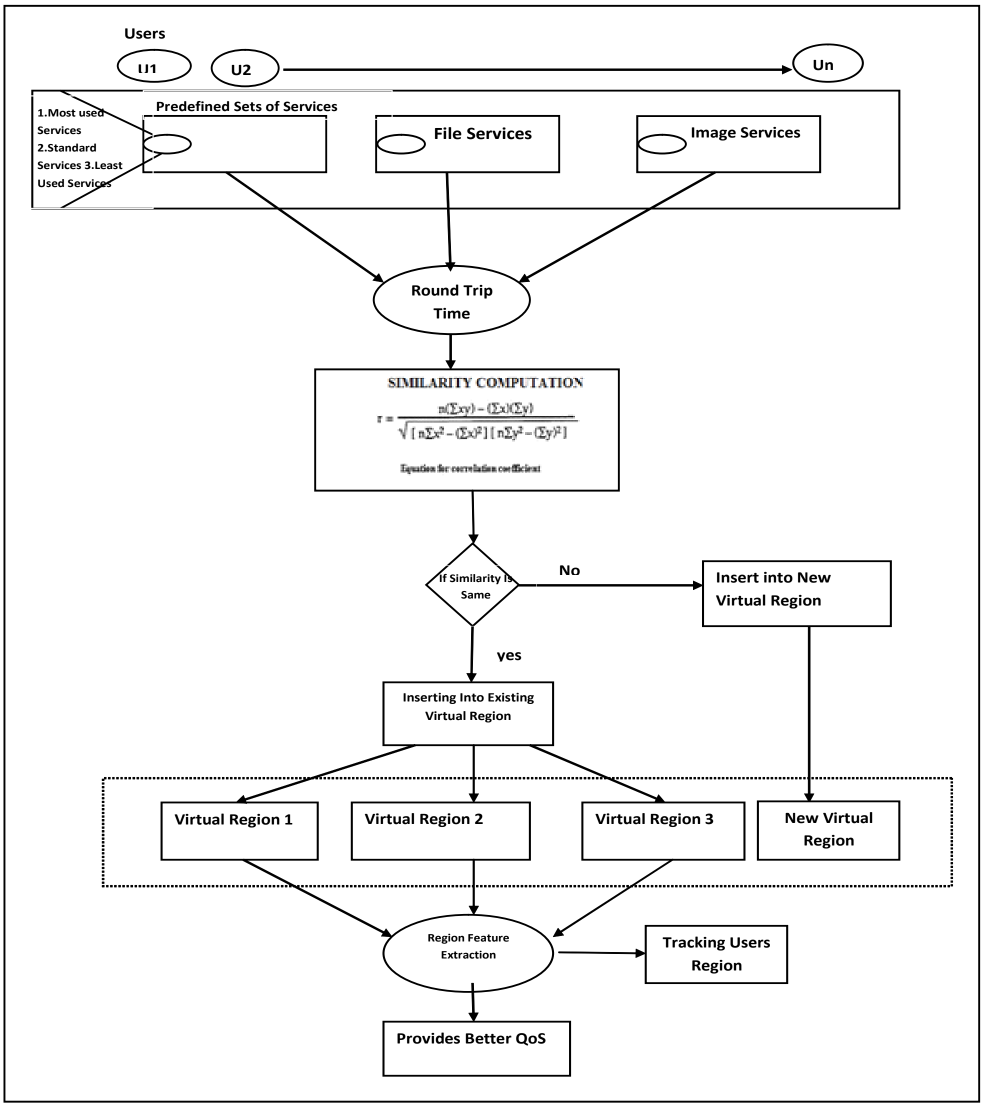

Figure 2. Architecture of WSRS based on Virtual Regions

Based on the collected QoS records, the recommendation approach is designed. Services with the best predicted QoS will be recommended to the current user. All the users having RTT value for different or same vector set of services within a user selected threshold are grouped together as one region. The newly formed and transformed regions are called '**Virtual Regions**'.

In this paper, emphasis is given 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. This system is studied using the response time (also called RTT). It is assumed to have n users and m services. The relationship between users and services is denoted by a n*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.

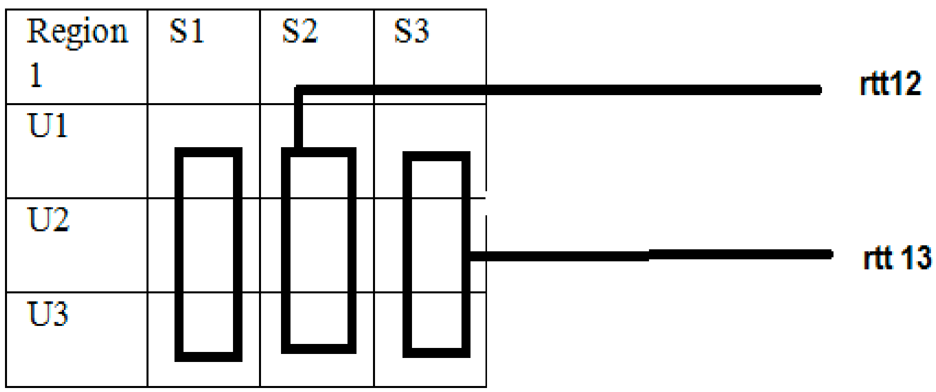

Figure 3. Region Separation based on Individual RTT Values

Individual RTT is provided for each service for the computation of similarity. Even if two places are not physically close but they are having similar RTT for services, they are grouped under one region, forming a virtual region. Correspondingly the region features are calculated based on the modified filtering algorithm as given in Figure 3. In Figure 3, U1, U2, U3 denotes the users and S1, S2, and S3 represents the services. rtt12 represents the RTT of service 2 observed by user 1.

The new recommendation system using virtual regions is designed for improving the performance in terms of the RTT value experienced by the users while using the services. The proposed system involves the following modules.

3.1 Virtual Region Creation

Each user i (i □ {1,2….,n}) is associated with a row vector R_i representing his/her observed RTT values on different web services. The user a (a □ {1,2,….n}) is called the active user or current user if he/she has provided some RTT records and needs service recommendations. Figure 4 represents the virtual region creation process based on the RTT values.

A virtual region is defined as a group of users who have similar QoS profiles. Each user is a member of exactly one virtual region. Virtual regions need to be internally coherent, but clearly different from each other. The virtual region creation phase is designed as a two-step process. In the first step, the similarity between the users is calculated using Pearson Correlation Coefficient (PCC). In second step, highly correlated users are aggregated to form a certain number of larger regions.

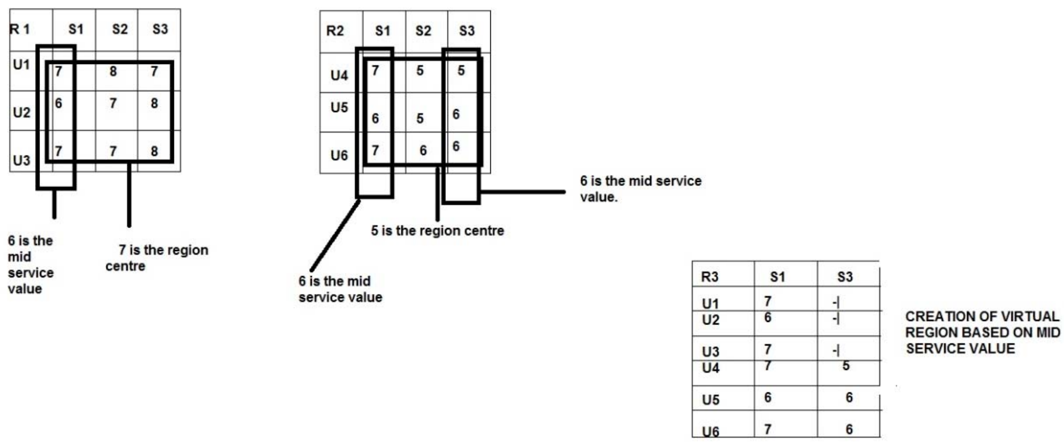

Figure 4. Creation of Virtual Regions

.

3.2 Virtual Region formation by Similarity Computation

The virtual region formation starts with an empty set of virtual regions. By following the steps given below a virtual region is created:-

1: A set of 'n' services are categorized together. A vector set of RTT values for these 'n' services would serve as parameter. The similarity value will be calculated based on this vector set..

2: When the first user enters to join a virtual region, there is no virtual region formed at that time. So the first user is joined in the first virtual region without any comparison or computation.

3: When a second user enters the system, its RTT vector is compared with the first user's RTT of each virtual region by using the Pearson Correlation Coefficient (PCC) as given in Equation (2)

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Equation for correlation coefficient (2)

were, n = Number of values or elements

$\sum x$ = sum of 1st values list
$\sum y$ = sum of 2nd values list
$\sum xy$ = Sum of the product of 1st and 2nd values
$\sum x^2$ = Sum of squares of 1st values
$\sum y^2$ = Sum of squares of 2nd values

4 : As PCC gives a value between -1 to +1 , a new user is added to a existing region based on the PCC value. If this region gives the best PCC similarity value for the user out of all the existing regions and this value is greater than a fixed threshold value to maintain the quality.
5: If the best PCC value for the user fails to cross the threshold value then this new user is taken as the first member of a new region.

3.3 Addition of a user to virtual region

In the previous section, which type of user can be grouped together in the same virtual region is discussed. But grouping of users itself is not sufficient. It has to be maintained in a matrix form for faster computation.

1: A m x n matrix is maintained for each user where ‘m’ is the number of users and ‘n-1’ is the number of services.

2: As one can see an extra ‘n’ th column has been added to the matrix which contains the PCC coefficient value obtained for each user when compared to the first user of the region.

3: Users are added to a region based on ascending or descending order of the PCC value of the user.

4: If a user ‘m’ is added to the region such that the newly added user don’t have any RTT value for a service ‘n’ i.e the user has never used the service ‘n’ , in that case an ‘nth’ column with a sign -| is added to the ‘m’ th user’s row.

5: Similarly if a user ‘m’ is added to the virtual region such that there exists a service ‘q’ which is only used by the user ‘m’ till now then a column for service ‘q’ is added to the matrix such that every user other than ‘m’ in that matrix gets a value -| for the RTT of service ‘q’.

6. Above listed steps are repeated till all the users are categorized.

3.4 RTT Value Prediction

As the virtual regions are created it is important to discuss how the prediction values will be generated from this matrix.

1: It is easily seen that the users are positioned in the matrix in an ascending or descending order based on their PCC value.

2: It is required to choose the “k” most significant services:

- Select (K/2) users having RTT value for the requested service, both above and below the requesting user.
- As the users are arranged on the basis of increasing or decreasing order, automatically the users having nearest QoS experience are selected.
- The user who is having the least PCC variance from the PCC value of the requesting user is the most significant user, while the user with the highest PCC variance from the PCC value of the requesting user is the least significant user.

3: Then a prediction for RTT value of a service ‘n’ for user ‘m’ is calculated by following the steps given below:

- Choose the “k” most significant users to retrieve the prediction value of RTT.
- Then the following formulae in Equation 2 is used to compute the RTT:-

  $$\{p*k + q*(k-1)+\ldots\ldots\ldots\ldots Y*(1)\}/ \{k+(k-1) + (k-2)\} \qquad (3)$$

  where ‘p’ is the most significant and ‘y’ is the least significant of “k” most significant users.

This equation results in the predicted value of RTT for the ‘n’th service for ‘m’th user.

Thus users are classified based on the virtual regions and retrieved RTT values.

4 Experimental Analysis

All results were compiled on an Intel Core 2 Duo Desktop clocking 2.83 GHz with 2GB RAM running on Windows 7. The programs were written in JAVA and the results were analysed.

Sparse training matrix was generated by removing the certain number of RTT records of the training users from the general training matrix. In addition to that some records of the active users were removed because they only invoke a small number of web services in reality. Prediction Performance was evaluated by comparing the proposed Virtual Region based Filtering approach (VRF) with User-based CF algorithm using PCC (UPCC), Item-based CF algorithm using PCC (IPCC) and WSRec which combines UPCC and IPCC. The density of the region is formulated based on the number of the users in a particular virtual region to that of the maximum number of users. Mean Absolute Error (MAE), the well-known statistical accuracy metric is used to measure the prediction accuracy. MAE is the average absolute deviation of predictions to the ground truth data. For all test services and test users:

$$\boxed{\text{MAE}=\sum(\text{ actual value of RTT} - \text{Predicted value of RTT})/ \text{Total no. of values predicted}}$$

Smaller MAE indicates better Prediction accuracy

4.1 Comparing Mean Absolute Error (MAE) Involving Different Methods

Table 1. Comparison of Prediction Performance

METHOD	**DENSITY=0.1**			**DENSITY=0.2**		
	Active Users =10	**Active Users =20**	**Active Users =30**	**Active Users =10**	**Active Users =20**	**Active Users =30**
IPCC	1163.22	1158.74	1150.56	1097.32	1089.94	1081.47
UPCC	1270.34	1133.75	1074.64	1161.64	840.36	668.57
WSRec	966.64	794.74	762.97	962.47	783.46	737.25
VRF	635.38	614.49	610.35	448.38	442.85	439.64

Table 1 shows the prediction performance of different methods employing the 0.1 and 0.2 density training matrix. It is that the Virtual Region based Filtering approach (VRF) significantly improves the prediction accuracy and outperforms other methods consistently. The performance of UPCC, WSRec, and VRF 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 other hand, there is only a slight improvement in values 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 VRF. Because even services provided by the same company will hardly have similar response times to different users.

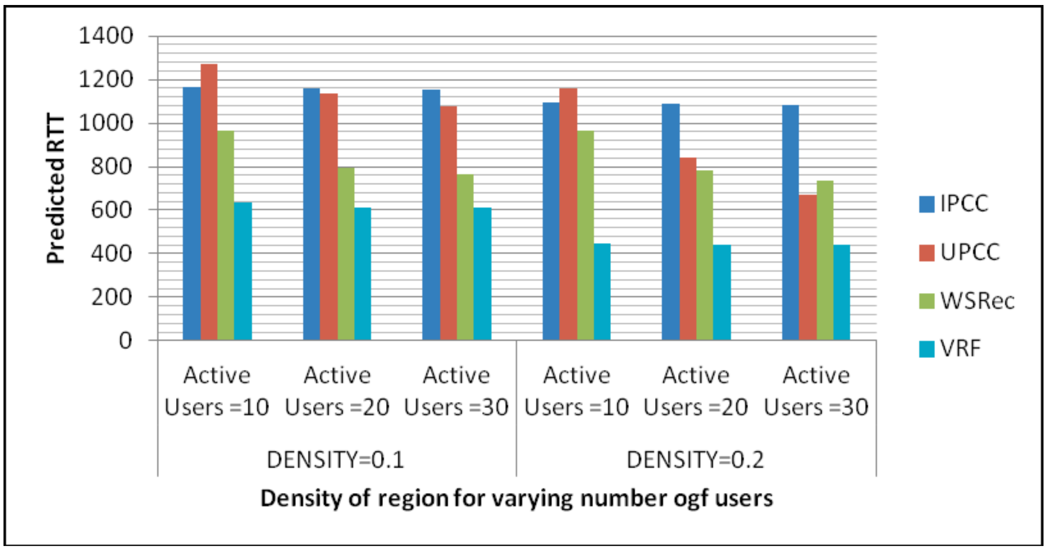

Figure 5. Density of Virtual Region Vs Predicted RTT

The number of RTT values given by active users are varied from 10, 20 to 30, and name them. From Table 1, it shows the findings for the density 0.1 and 0.2 respectively. For each density, there will be many active users and other users. From the above observation and from Figure 5, it is found that VRF approach achieves the better performance when compared to other methods.

1) RTT

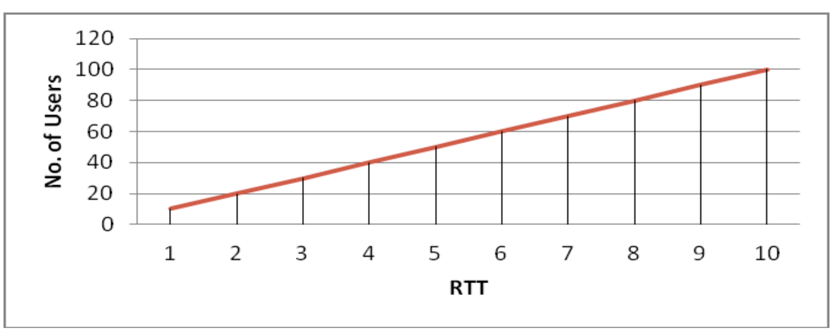

Figure 6. No. of Users Vs RTT

The chart given in Figure 6 shows the relationship between number of users and RTT achieved when virtual regions are used. Y axis gives the details of the number of users and X axis gives information regarding response time achieved.

It is evident from the figure 6 that it is linear in nature i.e as the number of users increases the response time also increases. The reason for increased response time is because of the many users who are trying to connect/request the same services repeatedly which induces the heavy traffic gained by the server and subsequently the response time for the service increases.

2) Prediction Analysis Vs Performance Analysis

The chart given in Figure 7 compares the Predicted RTT values with Actual RTT values. X axis represents the number of users and Y axis represents the RTT value in seconds.

Figure 7. Comparison of RTT Values

It is evident from the chart that the average difference between the predicted RTT value and actual RTT value is always less than or equal to one second.

3) Mean Absolute Error(MAE)

In the chart given in Figure 8, X-axis, represents the density of virtual region for varying number of users and Y-axis represents the MAE.

MAE can be calculated from the different methods such as item-based collaborative filtering algorithm using PCC, User based collaborative filtering algorithm using PCC and Virtual region based Filtering algorithm using PCC. Here, the MAE decreases when the density of the virtual region increases. In addition to that, virtual region based filtering algorithm gives less MAE value compared to that of IPCC and UPCC algorithms.

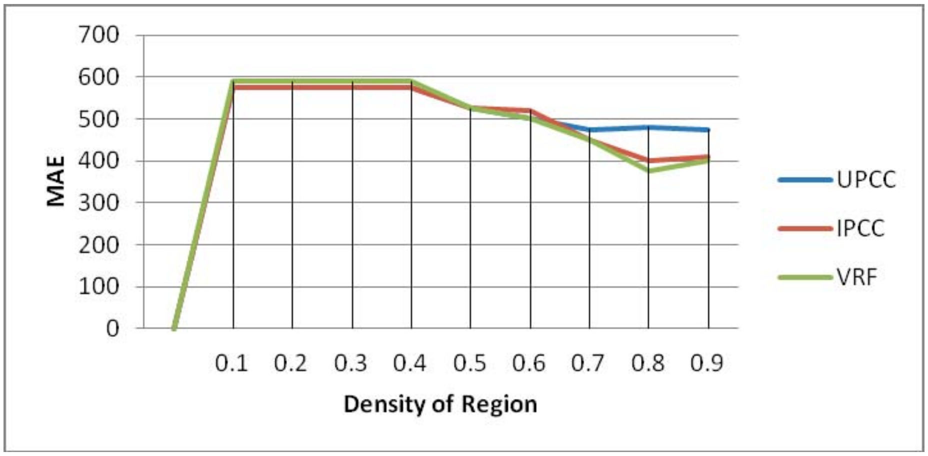

Figure 8. Density of Virtual Region Vs MAE

5 Conclusions and Future Work

In this paper, an innovative approach for web service recommendation is presented. The VRF algorithm employs the characteristic of QoS by clustering users into different regions. Based on the region feature, a refined filtering 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. The users physical location, will investigate more contextual information that influences the client side QoS performance. Experimental results show that the VRF approach significantly improves the prediction accuracy than the existing methods regardless of the sparseness of the training matrix. In this paper, the recommendation approach based on virtual regions considered the correlation between QoS records and users' physical locations by using IP addresses, which has achieved good prediction performance. The RTT values are compared for prediction and actual values. The MAE is calculated and charted for IPCC, UPCC and VRF. It is proved from the chart that VRF produces better results compared to other algorithms.

For the visualization of the recommendation results, we plan to add more user interactions such as searching the web services on the QoS map, zooming in and zooming out. In addition to that a searching algorithm can be explored to search the particular web service from the particular service providers irrespective of the regions.

References

1. Xi Chen, Zibin Zheng, Xudong Liu, Zicheng Huang and Hailong Sun, Personalized QoS-Aware Web Service Recommendation and Visualization, IEEE Transactions on Services Computing, 2011, (6)1,35-47.
2. L.-J. Zhang, J. Zhang, and H. Cai, Services Computing, Springer and Tsinghua Univ., 2007.

3. T. Yu, Y. Zhang, and K.-J. Lin, Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints, ACM Trans. Web, 2007, (1)1, 1-26.
4. 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, 2008, (1)4, 187-200.
5. Y.H. Chen and E.I. George, A Bayesian Model for Collaborative Filtering, Proc. Seventh Int'l workshop Artificial Intelligence and Statistics Seventh, http://www.stat.wharton.upenn.edu/~edgeorge/ Research_papers/Bcollab.pdf, 1999.
6. Hsu, and S.K. Halgamuge, Class Structure Visualization with Semi-Supervised Growing Self-Organizing Maps Neurocomputing, 2008, 71,3124-3130.
7. B. Mehta, C. Niederee, A. Stewart, C. Muscogiuri, and E.J. Neuhold, An Architecture for Recommendation Based Service Mediation, Semantics of a Networked World, 2004, 3226, 250-262.
8. W. Rong, K. Liu, and L. Liang, Personalized Web Service Ranking via User Group Combining Association Rule, Proc. Int'l Conf. Web Services, 2009, 445-452.
9. 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, 2007, 439-446.
10. R.M. Sreenath and M.P. Singh, Agent-Based Service Selection, J. Web Semantics, 2003, (1) 3, 261-279.
11. 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, 2009, 437-444.
12. 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), 1998, 43-52.
13. C. Zhao, C. Ma, J. Zhang, J. Zhang, L. Yi, and X. Mao, Hyper Service: Linking and Exploring Services on the Web, Proc. Int'l Conf. Web Services, 2010, 17-24.
14. 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, 2004, 372-383.
15. K. Tasdemir and E. Mere´nyi, Exploiting Data Topology in Visualization and Clustering of Self-Organizing Maps, IEEE Trans. Neural Networks, 2009, (20)4, 549-562.
16. J. Zhang, H. Shi, Y. Zhang, Self-Organizing Map Methodology and Google Maps Services for Geographical Epidemiology Mapping, Proc. Digital Image Computing: Techniques and Applications, 2009, 229-235