National Conference on Innovative Research & Industrial Applications

www.worldconferences.org

Location-Aware Web Service Recommender System with Optimal QoS Performance

SHAIK AFROZ[1], SHAIK NAZEER[2]

[1]PG Scholar, Dept of CSE, Quba College of Engineering & Technology, Venkatachalam, Nellore, AP, India.

[2]Assistant Professor, Dept of CSE, Quba College of Engineering & Technology, Venkatachalam, Nellore, AP, India.

Abstract: Web services are integrated software components for the support of interoperable machine-to-machine interaction over a network. In recent years Web services have been widely employed for building service-oriented applications in both industry and academia. The number of publicly available Web services is steadily increasing on the Internet. However, this proliferation makes it hard for a user to select a proper Web service among a large amount of service candidates. An inappropriate service selection may cause many problems to the resulting applications. In this paper, we propose a novel collaborative filtering-based Web service recommender system to help users select services with optimal Quality-of-Service (QoS) performance. Compared with existing service recommendation methods, our approach achieves considerable improvement on the recommendation accuracy. Comprehensive experiments are conducted involving more than 1.5 million QoS records of real-world Web services to demonstrate the effectiveness of our approach.

Keywords: Quality of Service(QoS), Recommendation, Web Service, Collaborative Filtering.

I. INTRODUCTION

Web services are designed for machine to machine interaction over a network. Web service employs WSDL (Web Service Description Language) for interface description and SOAP (Simple Object Access Protocol) for exchanging structured information. Benefiting from the cross-language and cross-platform characteristics, Web services have been widely employed by both enterprises and individual developers for building service oriented applications. The adoption of Web services as a delivery model in business has fostered a paradigm shift from the development of monolithic applications to the dynamic set-up of business processes. When developing service-oriented applications, developers first design the business process according to requirements, and then try to find and reuse existing services to build the process. Currently, many developers search services through public sites like Google Developers, Yahoo! Pipes, programmable Web, etc. However, none of them provide location-based QoS information for users. Such information is quite important for software deployment especially when trade compliance is concerned. Some Web services are only available in EU, thus software employing these services cannot be shipped to other countries. Without knowledge of these things, deployment of service-oriented software can be at great risk. Since selecting a high quality Web service among a large number of candidates is a non-trivial task, some developers choose to implement their own services instead of using publicly available ones, which incurs additional overhead in both time and resource. Using an inappropriate service, on the other hand, may add potential risk to the business process. Therefore, effective approaches to service selection and recommendation are in an urgent need, which can help service users reduce risk and deliver high-quality business processes. Quality-of-Service (QoS) is widely employed to represent the non-functional characteristics of Web services and has been considered as the key factor in service selection. QoS is defined as a set of properties including response time, throughput, availability, reputation, etc. Among these QoS properties, values of some properties (e.g., response time, user-observed availability, etc.) need to be measured at the client-side. It is impractical to acquire such QoS information from service providers, since these QoS values are susceptible to the uncertain Internet environment and user context (e.g., user location, user network condition, etc.).

Therefore, different users may observe quite different QoS values of the same Web service. In other words, QoS values evaluated by one user cannot be employed directly by another for service selection. It is also impractical for users to acquire QoS information by evaluating all service candidates by themselves, since conducting real world Web service invocations is time consuming and resource-consuming. To attack this challenge, this paper investigates personalized QoS value prediction for service users by employing the available past user experiences of Web services from different users. Our approach requires no additional Web service invocations. Based on the predicted QoS values of Web services, personalized QoS-aware Web service recommendations can be produced to help users select the optimal service among the functionally equivalent ones. From a large number of real-world service QoS data collected from different locations, we find that the user observed Web service QoS performance has strong correlation to the locations of users. Google Transparency Report1 has similar observation on Google services.

II. RELATED WORK

A. Collaborative Filtering: Collaborative Filtering (CF) is widely employed in commercial recommender systems, such as Netflix and Amazon.com. The basic idea of CF is to predict and recommend potential favorite items for a particular user employing rating data collected from other users. CF is based on processing the user-item matrix. Breese et al divide the CF algorithms into two broad classes: memory based algorithms and model-based algorithms. The most analyzed examples of memory-based collaborative filtering include user-based approaches, item-based approaches, and their fusion. User-based approaches predict the ratings of users based on the ratings of their similar users, and item-based approaches predict the ratings of users based on the information of item similarity. Memory-based algorithms are easy to implement, require little or no training cost, and can easily take ratings of new users into account. However, memory based algorithms do not scale well to a large number of users and items due to the high computation complexity. Model-based CF algorithms, on the other hand, learn a model from the rating data using statistical and machine learning techniques. Examples include clustering models, latent semantic models, latent factor models, and so on. These algorithms can quickly generate recommendations and achieve good online performance. However, these models must be rebuilt when new users or items are added to the system.

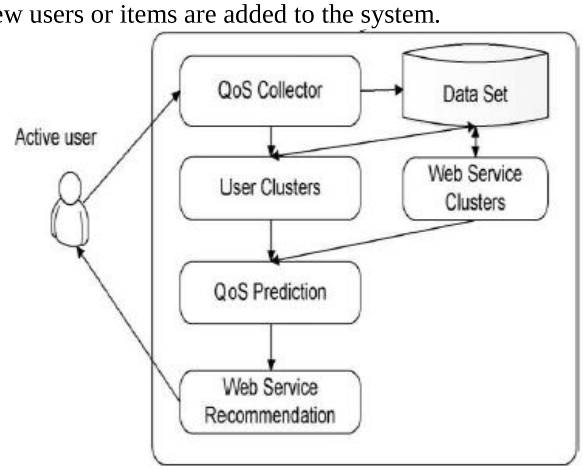

Fig 1. System overview of LoRec. B. Service Selection and Recommendation.

Service selection and recommendation have been extensively studied to facilitate Web service composition in recent years. Wang et al. present a Web service selection method by QoS prediction with mixed integer program. Zhang et al. provide a fine grained reputation system for QoS-based service selection in P2P system. Zheng et al provide a QoS-based ranking system for cloud service selection. Zhu et al. employ clustering techniques to their QoS monitoring agents and provide Web service recommendations based on the distance between each user and their agents. El Hadadd et al propose a selection method considering both the transactional properties and QoS characteristics of a Web service. Hwang et al. use finite state machine to model the permitted invocation sequences of Web service operations, and propose two strategies to select Web services that are likely to successfully complete the execution of a given sequence of operations. Kang et al. propose AWSR system to recommend services based on users' historical functional interests and QoS preferences. Barakat et al. model the quality dependencies among services and proposes a Web service selection method for Web service composition. Alrifai and Risse propose a method to meet users' end-to-end QoS requirements employing integer programming (MIP) to find the optimal decomposition of global QoS constraints into local constraints.

III. PRELIMINARY

A. System Overview Web 2.0 applications such as social networking sites and self publishing sites encourage users to share their knowledge and learn from others. LoRec employs the idea of user collaboration and provides a platform for users to share observed Web service QoS values and search Web services. This system will generate personalized service recommend-dations based on user shared QoS values. The more QoS records users contribute, the more accurate the recommendations will be, since more information can be mined from the user contributed QoS values. In this paper, we assume that users are trustworthy. How to detect and handle malicious users and inaccurate QoS values will be addressed in our future work. Fig. 1 shows the architecture of our LoRec recommender system, which includes the following procedures:

1. Web service users log on to LoRec system and share observed Web service QoS records with other users. In this paper, users who have submitted Web service QoS records to LoRec are called training users. If a training user requires Web service recommendation, then the user becomes an active user. QoS values of training users will be employed to make personalized recommendation for the active user.
2. LoRec clusters training users into different regions according to their physical locations and past Web service usage experiences.
3. LoRec clusters functionally similar Web services based on their QoS similarities.
4. LoRec maps the active user to a user region based on historical QoS and user location.
5. The recommender system predicts QoS values of candidate Web services for the active user and recommends the best one.
6. The active user receives the predicted QoS values of Web services as well as the recommendation results, which can be employed to assist decision making (e.g., service selection, service composition, service ranking, etc.).

A. Region Definition and Features

1. User Regions and Service Regions

Given a recommender system consisting of m users and n Web services, the relationship between users and Web services can be denoted by an m x n user-item matrix. An entry in this matrix ru,i represents a vector of QoS values (e.g., response time, failure rate, etc.) observed by user u on Web service i. If user u has never used Web service i before, then r u, i = null. A service region is a group of services with similar QoS performance. In LoRec, service regions are used to discover potential services and recommend them to active users. A user region is defined as a group of users who are closely located with each other and have similar Web service QoS usage experience. Each user belongs to exactly one region. Building regions help LoRec identify relationships in the QoS data set that might not be logically derived through casual observation.

2. Region Centers

Region center is a feature employed by both user region and service region. A user region center reflects the average performance of Web services observed by a set of similar users who belong to one region. A user region center is defined as the median vector of all QoS vectors associated with the region users. Median is the numeric value separating the higher half of a sample from the lower half. When there is an even number of samples, the median is defined to be the mean of the two middle values. The i^{th} element of the median vector of a region center represents the median QoS value of the ith service observed by users in the region.

3. Sensitive Web Services

Besides region centers, QoS fluctuation is another feature that deserves attention. From a large scale real data analysis, we discover that some QoS properties (e.g., response time) usually varies from one user region to another. Some services have unexpected long response time in certain user regions, and some services are even inaccessible to a few user regions. Inspired by the three sigma rule which is often applied to test outliers, we employ a similar method to distinguish services with unstable performance and regard them as user region sensitive services. For ease of discussion, let's pick one QoS property r (i.e., response time) as an example. The set of non-zero QoS values of service s, $s, \hat{r}_s = \{r_{1,s}, r_{2,s}, \dots, r_{k,s}\}, 1 \le k \le m,$ collected from users of all regions is a sample from the population of service s. To estimate the mean μ and the standard deviation σ of the population, we use two robust measures: median and Median Absolute Deviation (MAD). MAD is defined as the median of the absolute deviations from the sample's median

$$med = median_i(r_{i,s}), i = 1, \dots, k.$$

$$MAD = median_i(|r_{i,s} - med|), i = 1, \dots k. \quad (1)$$

Based on median and MAD, the two estimators can be calculated by

$$\hat{\mu} = median_i(r_{i,s}), i = 1, \dots, k \quad (2)$$

$$\hat{\mu} = MAD_i(r_{i,s}), i = 1, \dots, k. \quad (3)$$

IV. METHODOLOGY

Values of some QoS properties (e.g., response time) on the same Web service vary quite differently from user to user. Through the analysis of a real world Web service QoS data set2, which contains 1.5 millions service invocation records evaluated by users from more than twenty countries, we find that some QoS properties highly relate to the physical locations of users.

A. Phase 1: User Region Creation In this phase, users will be clustered into different regions according to their locations and historical QoS records. At the beginning, we retrieve users' approximate locations by their IP addresses.3 The location information reveals a user's country, city, latitude/longitude, ISP and domain name. Then users from the same city will be grouped together to form initial regions. These small regions will be aggregated into large ones with a bottom-up hierarchical clustering method. The clustering method has two parts: initialization and aggregation. In the initialization part, we select non sensitive user regions for aggregation, and compute the similarity between each region pair.

1. Select the most similar region pair (regioni, regionj), merge the two regions to regioni if their similarity exceeds the similarity threshold μ_u , otherwise stop this region aggregation process. To merge the two regions,

- Compute the sensitivity and region center of this newly merged region regioni. Remove this region from aggregation process if it becomes a sensitive one.
- Remove similarities between regionj and other existing regions.
- Update similarities between regioni and other existing regions.

2. Repeat the above step.

B. Phase 2: Service Region Creation

Normally, each user only uses a limited amount of Web services. Compared with the large number of services on the Internet, the number of services with user submitted QoS records is relatively small. Thus, it is difficult to find similar users, and predicting missing QoS values only from user's perspective is not enough. Clustering Web services can help LoRec find potential similar services. Different from retrieving user location from an IP address, LoRec directly clusters Web services based on their QoS similarity. This is because some companies regard the physical location of data center as a secret and use IP address to hide the real locations. Take Google for example. It has data centers located in Asia, Europe, America, etc, but physical locations retrieved from Google's IP addresses used in different country-specific versions of Google Search are all listed to Mountain View, California. Another reason is due to the use of the distributed system architecture. To enhance user interaction and to minimize delay, service providers will route user requests to different servers according to user locations or application types. Usually the server that processes requests is different from the one that responds to the users. Thus, retrieving a service location from an IP address does not prove much value. In LoRec, Web services are aggregated with a bottom-up hierarchical clustering algorithm. We use median vector rather than mean vector as the cluster center to minimize the impact of outliers. The similarity between two clusters is defined as the similarity of their centers. Each Web service is regarded as a cluster at the outset. The algorithm aggregates the pairs of the most similar clusters until none of the pairs' similarities exceeds threshold μ_w.

C. Phase 3: Personalized QoS Prediction

The first two phases aggregate users and Web services into a certain number of clusters based on their respective similarities. QoS predictions can be generated from both service regions and user regions. With the compressed QoS data, searching neighbors and making Web service QoS predictions for an active user can be computed faster than conventional methods.

1. **Prediction from User Perspective** Instead of computing the similarity between the active user and each training user, we only compute the similarity between the active user and each region center. Moreover, users in the same region are more likely to have similar QoS experience on the same Web service, especially on those region-sensitive ones.
2. **Prediction from Service Perspective** Clustering Web services provides another way to view and utilize the data set. It can enhance the prediction accuracy when we only have limited knowledge of user preference. To predict the QoS value of service s observed by user a from the service perspective, we use the Web service cluster center value of user a as a rough prediction if the center has the record of a; otherwise, we do not predict from the service perspective.

D. Phase 4: Web Service Recommendation

Web service QoS prediction is used in different ways in LoRec to facilitate Web service recommendation. First, when a user searches Web services using LoRec, predicted QoS values will be shown next to each candidate service, and the one with the best predicted value will be highlighted in the search result for the active user. It will be easier for the active user to decide which one to have a try. Moreover, LoRec selects the best performing services (services with the best submitted QoS) and services with the best predicted QoS from the whole service repository for the active user so that he/she can quickly find potential valuable ones instead of checking the service one by one.

V. EXPERIMENTS

In this experiment, we crawl publicly available Web services from three sources 1) well-known companies (e.g., Google, Amazon, ect.); 2) portals listing publicly available Web services (e.g., xmethods.net, webservicex.net, etc.); and 3) Web service search engines (e.g., seekda.com, esynaps.com, etc.). Java classes are generated using WSDL2Java tool of Axis2 package. To obtain QoS values of Web services, we employ 150 computers in 24 countries from Planet-Lab to monitor 100 real Web services in 22 countries. About 1.5 millions Web service invocation records are collected in two days' time. For each user (a computer node from Planet-Lab), there are around 100 profiles, and each profile contains the response time (also called Round Trip Time, RTT) records of 100 services. We randomly extract 20 profiles from each node, and generate 3000 users with RTTs ranging from 2 to 31407 milliseconds. We divide the 3000 users into two groups, one as training users and the rest as active (test) users. To simulate the real situation, we randomly remove a certain number of RTT records of the training users to obtain a sparse training matrix. We also remove some records of the active users, since active users usually use a small number of Web services in reality. We apply Mean Absolute Error (MAE) to measure the prediction accuracy of the recommendation algorithm. The more accurately the algorithm predicts, the better the recommendations are. MAE is the average absolute deviation of predictions to the ground truth data, where smaller MAE indicates better prediction accuracy

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \widehat{r_{i,j}}|}{N} \quad (4)$$

TABLE 1: Time Usage Comparison of Online QoS Prediction

Method	Online Time Duration (s)
IPCC	2.437
UPCC	5.218
WSRec	6
RegionKNN	0.094
LoRec	0.141

Where, $r_{i,j}$ denotes the expected QoS value of Web service j observed by user i, is the predicted QoS value, and N is the number of predicted values. MAE reflects how close predictions are to the eventual outcomes on average, which gives an overview of the prediction quality. Compared with the amount of services on the Internet, the number of services consumed by each user is small. The data set of recommender systems is usually sparse. We examine how data sparseness impacts the prediction results from two aspects: the density of training matrix which indicates how many QoS records are collected from all users, and the number of QoS values given by active users.

VI. CONCLUSION

This paper presents a QoS-aware Web service recommendation approach. The basic idea is to predict Web service QoS values and recommend the best one for active users based on historical Web service QoS records. We combine prediction results generated from service regions and user regions, which achieves better results than existing approaches. We also find that the combination result is much better than the result from any single method, either the prediction generated from user regions or the one generated from Web service regions. This is because these two methods analyze the problem from different aspects and the combination of them counteracts the error of individual methods. In our future work, we will consider several aspects to further improve the proposed Web service recommendation approach. In terms of the clustering method, we will consider probabilistic ones like EM to improve the scalability of LoRec.

VII. REFERENCES

[1] R. Burke, ‘‘Hybrid Recommender Systems: Survey and Experiments,’’ User Model. User-Adapt. Interact., vol. 12, no. 4, pp. 331- 370, Nov. 2002.
[2] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, ‘‘Personalized QoS-Aware Web Service Recommendation and Visualization,’’ IEEE Trans. Serv. Comput., vol. 6, no. 1, pp. 35-47R1st Quart., 2013.
[3] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, ‘‘Item-BasedCollaborative Filtering Recommendation Algorithms,’’ in Proc. 10th Int’l WWW Conf., 2001, pp. 285-295.
[4] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, ‘‘Personalized QoS Prediction for Web Services via Collaborative Filtering,’’ in Proc. 5th ICWS, 2007, pp. 439-446.
[5] R. Jin, J. Chai, and L. Si, ‘‘An Automatic Weighting Scheme for Collaborative Filtering,’’ in Proc. 27th Int’l ACM SIGIR Conf., 2004, pp. 337-344.
[6] G. Kang, J. Liu, M. Tang, X. Liu, B. Cao, and Y. Xu, ‘‘AWSR: Active Web Service Recommendation Based on Usage History,’’ in Proc. IEEE 19th ICWS, 2012, pp. 186-193.
[7]K.Karta, ‘‘An Investigation on Personalized Collaborative Filtering for Web Service Selection,’’ Honours Programme Thesis, Univ. Western Australia, Brisbane, Qld., Australia, 2005.