QoS Based Approach For Web Service Recommendation

Pooja S. Chame
Department of Information Technology
Shah And Anchor Kutchhi Engineering College, Chembur,
Mumbai, India

Swati Deshpande
Department of Information Technology
Shah And Anchor Kutchhi Engineering College,
Chembur, Mumbai, India

Abstract: Web services are standard means of interoperating between software applications running on a variety of platforms and frameworks spread across a network. Web services are recently playing a crucial role in enhancing service orientation approach in many fields throughout the globe. There is a boost in publicly available web services owing to their varied structure, facilities, and user convenience. These varied characteristics often lead to the growing confusion among the users for utilizing a particular service which, in turn, leads to improper selection and degraded performance of the application. This paper provides a work to select a proper web service for their application based on collaborative filtering with optimal quality of performance. Service recommendations are made by our software by clumping users and values on location details and QoS and clump results. The Recommendation accuracy of our system was sizably improved as compared to existing system. The successfullness of our approach was based on large size QoS records.

Keywords: Ad-hoc, MANET, Caching, Routing;

I. INTRODUCTION

The web infrastructure is used to describe, publish, characterize and invoke web services. Three important standards which define web services are (1) Web Service Description Language (WSDL) for presenting service interfaces, (2) Universal Description, Discovery and Integration (UDDI) registries for publishing, and (3) Simple Object Access Protocol (SOAP) for message transporting. There are numerous approaches for web service discovery which prove efficient only if the service providers publish relevant service specifications [1][2][3]. Furthermore, a third party may test a web service, verify their description and publish their QoS. The static nature of this approach makes it prone to errors.

On the other hand, there are service clients who previously have experience in using web services and therefore can help in selecting services with adequate quality. This principle is extensively used by (collaborative) recommendation and reputation systems [6][7]. A group of clients with common interests form a virtual community where they can exchange the experience, i.e., the knowledge achieved after having interaction with a web service. To join a virtual community that shares experience in retrieving of web services, the user must (1) install the remote client, (2) generate reminder for service invocations using the latest version of the Axis tool. No more involvement, user-to-user or user-to-system communications are needed, excluding for submitting requests.

If the user does not to need to find for new web services, the system can be used for service monitoring on the client side. The SICS remote client supply an interface for the user to call the system by submitting requests. Request may consist of textual details of the goal, name of the expected operation, description of its input/output parameters, description of a expected web service and its features (provider, etc.).

A. SYSTEM FOR WEB SERVICE DISCOVERY

This section gives a confession of the process of web service discovery within the system. The motivation for adopting the implicit culture approach for web service discovery branch from the difficulty of developers in finding and selecting web services suitable for their applications [10]. The system is designed for the use by a virtual community, giving suggestions about web services suitable for this community. In our domain, developers and their applications execute actions on web services. Types of actions determine by the SICS and will be made clear later in detail. Behaviour, agents and objects also may have more than one attributes, i.e., features helpful for their scrutiny. Information about a web service (id,
name, etc.) is stored as an attribute of an object activity. The description of the entire set of the stored attributes is not needed for considerate the current paper.

In order to use the system, each user must install a remote client. The aim of the remote client is to interact with the SICS, and forward user requests and store observations about user actions, applications and behaviour of web services. For observations of interaction with a web service we have extended the Java Stubs Writer class of the open-source Apache Framework 2. This class makes stubs for web service invocation. The modification that we implemented allows the stubs to report the information about the communication among a user application and a web service to the IC-Service, using the remote client. Thus, to join a virtual community that shares experience in retrieving of web services, the user must (1) install the remote client, (2) generate reminder for service invocations using the latest version of the Axis tool. No more involvement, user-to-user or user-to-system communications are needed, excluding for submitting requests. If the user does not need to find for new web services, the system can be used for service monitoring on the client side. Run-time web service monitoring is necessary for real-world service oriented systems where control of service quality is needed [11][12].

The SICS remote client supply an interface for the user to call the system by submitting requests. Request may consist of textual details of the goal, name of the expected operation, description of its input/output parameters, description of a expected web service and its features (provider, etc.). By configuring the similarity algorithm it is possible to define whether these requirements are

II. RELATED WORK

Collaborative Filtering methods, content based methods and hybrid methods are three different methods that are universally used in Web service recommendation system.

A. CF METHODS

The memory based and model based methods are two different type of CF techniques that are universally used in recommendation systems. Well known memory based methods include user based approaches [7] and item based approaches [8]. Memory based Collaborative Filtering techniques have been recently adopted to provide QoS-aware recommendations [9, 10]. Shao et al. [9] propose a typical user-based CF method to anticipate QoS which supposes that same type users likely to receive same QoS from same services, and they use Pearson Correlation Coefficient (PCC) to calculate similarity between users. Zheng et al. develop a model which enhance the user-based Collaborative Filtering by fusing item-based CF [10].

The model-based method allows the system to make Clever forecast for the CF algorithms tasks based on some Learned models [5, 6]. Matrix factorization (MF) is one of the ideal works. In [11], MF is used to develop a global model for forecast QoS data, which can accomplish higher prediction accuracy. Yu et al. [13] recommend a matrix completing approach using an effective iterative algorithm

B. CONTENT BASED METHODS

The content based methods mainly concentrate on providing a mechanism to characterize users’ choice, resource, and the description of Web services, and recommendations are generated based on the predefined semantic models. Zhao et al. [15] provide a way to model services and their correlation by semantic algorithm. Based on the input keywords, users can get a set of recommendations with correlation to the query. Blake and Nowlan [24] compute a recommendation score by matching strings collected from the user’s operational sessions and the description of the Web services. Based on this score, they inspect whether a user is interested in the service. Mehta et al. [5] add quality and usage pattern to the service information to provide more information to discover a service that meets user requirements. Maamar et al. [6] introduce a model for the context of Web service interactions and highlighted the Resource on which the Web service performed.

C. HYBRID METHOD

Combination CF with other techniques can supply more exact expectation, they are widely used. Numerous hybrid models have been presented that involve other related factors to improve service recommendation quality, such as users’ locations [16, 17], social network information [18] and temporal effects [19]. Chen et al. [16] introduce a CF algorithm which takes into account of users’ physical locations and design a region model for large-scale Web service...
recommendation. Tang et al. [17] determine a location aware CF model by incorporating locations of both users and services. Tang et al. [12] present a trustworthy recommendation method with social network which accommodate some social relation. Amina et al. [18] denote an approach that integrates ARIMA and GARCH models to catching the QoS attributes' changeability.

All the mentioned approaches do not take into consideration the cold-start problem in service recommendation. There are some approaches focusing on defeating the problem. For instance, Yu [12] integrates MF with decision tree grasping to bootstrap service recommendation. MF is used to forecast Qos data and then handle new user issue. Bobabilla et al. [25] design new similarity metrics using optimization based on neural learning which supply greater accuracy to check new user cold start situations. However, works [12, 25] are both target on providing more precise classification of new users, while we mainly deal with dynamic scenario. Ling et al [14] prepare an online framework for CF to handle different scenario. We advantage their theory of online learning to handle cold-start problem in service recommendation. 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.

WSDL DOCUMENT PARSING AND INFORMATION EXTRACTION

The Web Service Description Language (WSDL) is an XML-based language, designed according to standards specified by the W3C that provides a model for describing web services. It characterize one or more services as collections of network endpoints, or ports. It provides the specification necessary to use the web service by describing the communication protocol, the message format required to communicate with the service, the actions that the client can invoke and the service location. Two versions of WSDL recommendation endure the 1.1 version, which is used in almost all existing systems, and the 2.04 version which is calculated to replace 1.1. These two versions are functionally absolutely similar but have substantial differences in XML structure.

To manage efficiently web services descriptions, we extract all features that describe a web service from the WSDL document and store them into a relational database. We recognize both WSDL versions (1.1 and 2.0). During this process, we proceed in three steps (see Figure 1). The first step for checking availability of web service and validating the content of Web service Description Language document. The second step is to collect the WSDL document and read it from the WSDL URI to fetch all details of the document. In this step we describe the features to extract from the WSDL document: (1) the name, the documentation and the version of the WSDL, (2) WSDL types used by messages to transmit information between web services. Data types are often specified using a XML Schema Definition (XSD). We extract all kind of elements and types that can be as simple or complex types as a set of elements and/or attributes, and (3) a set of services declared in the WSDL document. For each service we extract the name, the documentation and a set of endpoints. Then, for each endpoint we extract the name, the address (which defines the connection point to web service, It is typically represented by a simple HTTP URL) and the binding (Name, Type, Style, Transport protocol).

The binding specifies the interface as well as defining the SOAP binding style (RPC/Document) and transport SOAP protocol. The interface defines the operations to be performed for a web service, and the messages that are used to perform the operation. We also extract for each operation the name, the documentation, the input and the output parameters. The input/output parameters can be referred to the previously extracted types/elements. Finally, the third step is dedicated to save the extracted WSDL features into a database. The extracted information will be used during the generation of representations (B, RBTT and SR). Before presenting the methods for calculating representations, we discuss some text-processing standard used thereafter.

✓ **TAG REMOVAL:** This step deletes all HTML tags, CSS components, symbols (punctuation, etc.).

✓ **DIVIDE AND REMOVE A STOP WORDS:** Some terms are posed by several words separated by a capital letter; we use therefore regular expression to extract these words. To emphasize, the application of the regular expression \([A-Z][a-z]+\) on this string "Get All Country Currencies Response" produces 'Get', 'All', 'Country', 'Currencies' and 'Response'. Moreover, to extract the potential content words, we remove all.

✓ **WORD STEMMING:** In this step we use the Porter Stemmer [21] to remove words, which have the same stem. Words with the same stem will usually have the duplicate meaning. For example, 'computer', 'computing' and 'compute' have the stem 'compute'.

![](http://www.datascience.net/services2/zip_longer.png)

**Figure 3: Process Flow**

**REPRESENTATIONS OF WEB SERVICES**

In this section, we present a generation method of the traditional representation of web service and we introduce two new representations. Let us note that the generated representations are Victoria.

A. **BASELINE REPRESENTATION (B)**

A web service can be characterized by a textual description extracted from WSDL document or given by its
provider when publishing in the UDDI. The current UDDI registry only allows searching web services by their textual description. The first representation is centered on textual description s of services and their offered functions. This is produced from the web service descriptions and enriched by integrating the descriptions of operations offered by services. Let us remark that the major disadvantage is that most web services have a poor or an empty description. The types are used by messages to transmit information between web services. Consequently, WSDL types are nice features to describe the operations of a service and are the most descriptive element in WSDL document. For this reason, we extract all type names (elements, complex types, simple types, attributes, catalog) and apply the textual processing (see Section 3 -steps 2 and 3) to produce a set of words. Thus, we use the obtained set of words to construct the new representation and we consider it as a baseline representation (B) for a web service.

B. RULES BASED TEXT TAGGING OF WEB SERVICES DESCRIPTIONS (RBTT)

The main focus is how to recognize significant parts or entities in the text description and how to use the filtered information to describe the web service? Our approach consists in the definition of extraction rules to identify, retrieve and comment relevant multi-word terms from web service descriptions. This approach has been already used for biological data [19]. The processing steps (Tokenization, Part-Of-Speech tagging, Extraction and output generation) of Information Retrieval have been developed as modules using the Lingua Stream platform5. Lingua Stream [20] is an integrated experimentation environment targeted to scientist in natural language processing (NLP). Let us note that we use Tree Tagger6 for the Part-Of-Speech tagging step. The extracted information is given in a form of a XML file. In the context of the web service descriptions, we have destine a Rules set into Prolog giving back the Definite Clause Grammar (DCG) to recognize three types of information: web service names (namespace), intention of the web service(purpose), and the domain of utilization (domain). Let us denote that we do not use patterns in the sense of Information Extraction that is without a prior on the form of the expressions.

**Figure 4: Structure of Rules**

III. EXISTING SYSTEM

When building 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. now a days, multiple developers search services through public sites like Google Programmers (developers.google.com), Yahoo Pipes (pipes.yahoo.com), programmable Web (pro-grammableweb.com), etc. Nonetheless, none of them provide location-based QoS information for users. Such detail is more important for software deployment especially when trade compliance is concerned. Some Web services are only present in EU, thus software employing these services cannot be shipped to other countries. Without information of these things, deployment of service-oriented software can be at great risk.

- Web service QoS performance has minimal correlation to the locations of users.
- No Web Service Recommendation.
- Service-oriented software can be at more risk.

IV. PROPOSED SYSTEM

We introduced a innovative collaborative filtering-based Web service recommender system to help users select services with optimal Quality-of-Service (QoS) performance. Our recommender system exploit the location information and QoS values to cluster users and services, and makes personalized service recommendation for users based on the clustering results. correlated with existing service recommendation methods, our approach achieves considerable improvement on the recommendation accuracy. Broad experiments are conducted containing more than 1.5 million QoS records of active world Web services to identify the effectiveness of our approach.

- Avoid risk and deliver high-quality business processes
- Web service QoS outcome has big correlation to the locations of users.
- Client Collaboration.
- Location detailed is also considered when clustering users and services.
- Service region and User region.

**Figure 5: Architecture of proposed system**

COLLABORATIVE FILTERING

Collaborative Filtering (CF) is widely employed in commercial recommender systems, such as Netflix and Amazon.Com. The basic concept of CF is to predict and recommend potential favourite items for a particular user employing rating data collected from other users. CF is relay on processing the user-item matrix, classify the CF algorithms into two broad classes: memory relay algorithms and model relay algorithms. The most evaluated 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 same kind of users, and item-based ways anticipate the ratings of users based on the information of item similarity. Memory-based algorithms are simplest 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 huge number of users and items due to the high computation complexity.

A. QUALITY OF SERVICES

FIRST, we nominate a novel location-aware Web service recommendation approach, which significantly improves the recommendation accuracy and time complexity compared with existing service recommendation algorithms.

SECOND, we done inclusive experiments to measure our approach by employing a real world Web service QoS data set. Excess than 1.5 millions real-world Web service QoS records from more than 20 countries are busy in our experiments. Broad analysis on the impact of the algorithm parameters is also provided.

B. 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 × n user-item matrix. An entry in this matrix row, i represents a vector of QoS values (e.g., response time, failure rate, etc.) observed by user u on Web service.

A service region is a group of services with same QoS outcome. In LoRec, service regions are used to find potential services and recommend them to active users. A user region is defined as a group of users who are nearly located with each other and have similar Web service QoS usage experience. Each user exactly from one region. Developing regions help LoRec identify relationships in the QoS data set that might not be logically derived through casual observation.

C. SENSITIVE WEB SERVICES

DEFINITION 1: The sensitivity of a region is the fraction between the numbers of sensitive services in the region over the total number of services.

DEFINITION 2: A region is a sensitive region if its region sensitivity exceeds the predefined sensitivity threshold (lambda).

D. WEB SERVICE RECOMMENDATION

Web service QoS forecast 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 displayed 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 simplest 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 immediately find potential valuable ones instead of checking the service one by one.

V. METHODOLOGY

Values of som 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 (see Section 5 for details), 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. For example, the response time of a service observed by closely located users usually fluctuates mildly around a certain value. On the other hand, the response time observed by users who are far away from each other sometimes varies significantly. Based on this finding, our recommendation algorithm takes location information.

A. 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 [20]. The clustering method has two parts: initialization and aggregation. In the initialization part, we select no sensitive user regions for aggregation, and compute the similarity between each region pair with Eq. (5). To aggregate regions,

✓ Select the most similar region pair merge the two regions if their similarity exceeds the similarity threshold u. Otherwise stop this region aggregation process. To merge the two regions,

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

✓ Repeat the above step.

Threshold u is a tunable parameter that can be adjusted to trade off accuracy for time and space requirements.

B. 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 w.

C. 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 neighbours and making Web service QoS predictions for an active user can be computed faster than conventional methods.

D. WEB SERVICE RECOMMENDATION

Web service QoS prediction is used in number of ways in LoRec to facilitate Web service recommendation. First, when a user searches Web services using LoRec, predicted QoS values will be displayed 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 simpler for the live 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 immediately discover potential valuable ones instead of checking the service one by one.

V. RESULT

In this system we propose an innovative CF algorithm for QoS-based web service recommendation. To address the third problem and enable an improved understanding of QoS based approach for web service recommendation the web service recommendation 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 basic structure of the QoS space by using map metaphor such as dots, areas, and spatial arrangement.

VI. CONCLUSION

This paper presents a QoS-aware Web service recommendation approach. The basic idea is to conclude 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.

ACKNOWLEDGEMENT

I would like to thank my Project Guide Prof. Swati Deshpande, for his exemplary counsel, valuable feedback and constant fillip throughout the duration of the project. Her suggestions were of immense help throughout my project work. Working under her was an extremely knowledgeable experience for me.

REFERENCES


