Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Collaborative Filtering Approach For QoS Prediction

G.Janani alias Pandeeswari
PG Scholar, Department of Computer Science And Engineering, R.V.S School of Engineering, Dindigul.

Abstract- Many researchers propose that, not only functional but also non-functional properties, also known as quality of service (QoS), should be taken into consideration when consumers select services. 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, this paper proposes a collaborative filtering based approach to making similarity mining and prediction from consumers' experiences. Experimental results demonstrate that this approach can make significant improvement on the effectiveness of QoS prediction for web services.
Keywords: Web services, Service recommendation, QoS, QoS prediction, collaborative filtering,

I. INTRODUCTION

Web services have become the primary source for constructing software system over Internet. The quality of whole system greatly dependents on the QoS of Web services, so QoS information is an important indicator for service selection. Web service QoS prediction is an important step in selecting services. During the process of selecting services, consumers would like to use the QoS data acquired by themselves. But a single consumer cannot have sufficient pre-acquired data for all candidate web services. To make a comparatively accurate prediction, a consumer could reuse others' usage experiences. A straightforward reuse mechanism is to calculate the arithmetic mean of all consumers' QoS data. However, this mechanism does not take consumers' differences into consideration. According to our experiences, consumers often have different experiences on the quality of the same web service. These differences are caused by many factors, such as network environment, programming languages, and so on. Averaging all consumers' QoS data results in inaccurate prediction because it ignores the factor that consumers' have different characteristics and different QoS experiences. To address this problem, this paper proposes an approach to predicting QoS for web services, taking the similarity among consumers' experiences into consideration. The basic assumption of this approach is that the consumers, who have similar historical experiences on some services, would have similar experiences on other services. Our approach consists of two steps. First, we figure out the similarity between each two consumers with their historical QoS data. Second, we predict the quality of unused service for consumers based on the data published by other consumers regarding the similarity.

II. APPROACH

The basic idea of our approach is to find out similarity between the consumers with collected QoS data and then to make prediction for the unused services based on the similarity. Collaborative filtering has a very similar motivating problem space with ours and is easy to implement. The main idea of collaborative filtering is to recommend items to consumers regarding other customers' experiences [9]. Collaborative filtering has been adopted in many applications for recommendations and achieved good results. However, there still exist two points which make collaborative filtering technique unable to be used directly for our problem. First, users' experiences in collaborative systems are often subjective ratings; on the contrary, in our motivating problem, the experiences are objective QoS data. Second, the result of predicting in collaborative filtering system is represented as a number, which is different from a vector in our problem.

Our approach starts from transforming the motivating problem into the standard form of collaborative filtering and normalizes the objective QoS data into a uniform range. With an modified collaborative filtering based algorithm, our approach gives out the predicting results of QoS data.

Our approach consists of four key steps:

1) Data Preparation,

2) Normalization,

3) Similarity Mining and

4)Prediction Making.

Data Preparation makes our collected QoS data consistent with the format of collaborative filtering. Normalization makes the values of QoS property into a uniform scope based on Gaussian approach. Similarity Mining calculates the similarity with Pearson Correlation Coefficient between two consumers based on their historical experiences. Prediction Making makes linear prediction for each QoS property based on the similarity and combine the prediction values into a quality vector.

2. Data Preparation

Data collected from consumers are denoted as $T = <D1; D2; :::; Dl >$. Prediction in our approach is made for single QoS property, that is, we make prediction separately for response time, availability, etc.First, we transform T into several vectors, and each of the vector contains QoS data of a unique quality property. Second, we set a threshold to exclude some consumers' data. There would be so many services but a consumer may only has experiences on few of them. The consumer who provides a small amount of data will not make contributions to but only disturb the statistical result. For example, there are ten services which could satisfy consumer ui's functional requirements, and ui wants to know the quality of service $s1$. Consumer uj only provides QoS data of two of these ten, say, $s1$ and $s2$, but quality data of $s2$ provided by uj has much similarity with that provided by ui. This similarity comes accidentally but it still suggests that ui would have similar experience on $s1$ with uj . Then uj 's QoS data make much impact on predicting, which is just a coincidence and had better be avoided in QoS prediction.

3. Normalization

Values of QoS properties can be data type or ratio type. The value of ratio type property varies in a limited range, such as 0~ 100%. The value of data type property may vary in different ranges. A typical data type property is response time, which is possible to vary in range of [0,1s] for one kind of consumers, but in range of [10s, 20s] for the other king of consumers. Data type properties should be normalized to make data in different range be fair on making prediction. To be consistent with standard collaborative filtering format, we normalize the QoS data from each consumer separately.

The values which are extremely large or small will disturb the result of normalization. To avoid this, we use Gaussian approach [18] to normalize QoS data as shown in Equation (1), where pki denotes the arithmetic mean of QoS data collected from consumer ui on k-th property. σ is the standard deviation of consumer ui's QoS data. We use 3 $\sigma\ i$ because of the 3 - σ rule, which helps to normalize the value into the range of [0,1]. It can be proven that the probability of the normalized value being in the range of [0,1] is approximately 99% [18]. We map the value out of the range of [0,1] to 0 or 1.

4. Similarity Mining

In our approach, the similarity between ui and uj on the k-th property indicates the probability of how $p^k{}_{i,r}$ is similar to $p^k{}_{j,r}$. Similarity is mined from consumers' historical experiences.As proposed by Jonathan L. Herlocker, etc. [9], when the value is continuous, Pearson correlation coefficient [17] is satisfactory for measuring similarity. Because the values of QoS properties are almost continuous, we adopt the Pearson correlation coefficient as the metric of consumers' similarity. The common presentation of Pearson correlation coefficient is

$$w^r{}_{a,b} = \sum i{=}1..m \wedge p^r{}_{a,i} \ \bar{\varepsilon}\ R \wedge p^r{}_{b,i} \ \bar{\varepsilon}\ R((p^r{}_{a,i} - p^r{}_a)(p^r{}_{b,i} - p^r{}_b)) \qquad (1)$$

$$\sigma_{\ a} = \sqrt{\textstyle\sum_{j=1..m \wedge pra,j} {}^{\varepsilon}{}_{R}}(p^r{}_{a,i} - p^r{}_a)^2 \qquad (2)$$

$$p^r{}_a = \sum j{=}1..n \wedge p^r{}_{a,j}{}^{\varepsilon}\ R\ p^r{}_{a,j} \div n \qquad (3)$$

In Equation (1), $w^r{}_{a,b}$ is the similarity weighting between consumer ua and ub on the r-th property.

5. Prediction Making

The prediction of $p^r{}_{a,i}$ is based on $p^r{}_a$ and the expected deviation between $p^r{}_{a,i}$ and $\bar{p}^r{}_a$. The expected deviation is often a weighted mean of all $p^r{}_{b,i}(b \neq a)$. The basic equation in common collaborative filtering system for predicting $p^r{}_{a,i}$ is (4) [9]:

$$p^r{}_{a,i} = p^r{}_a + \sum\nolimits^n_{u=1}(p^r{}_{u,i} - p^r{}_u) * w^r{}_{a,b} \div \sum\nolimits^n_{u=1} w^r{}_{a,b} \qquad (4)$$

When all the $p^r{}_{i,j}$ ($p^r{}_{i,j}$ = ?) have been predicted, the predicted values are combined into the format of $Q_{i,j}$.

III. EXPERIMENT DESIGN

Our experiment design consists of three models:

1. Consumer model
2. parameter model and
3. QoS model.

1. Consumer Model

The consumer model describes the definition of service consumers and consumer related environment. We developed client programs for each real web services and distributed these client programs to volunteers. The volunteers are distributed. These volunteers use different operation systems, such as Windows and Linux and they run the client programs in different network environments. To simulate more service consumers, we provide a proxy server list for each volunteer and the address of proxy servers are found on the Internet and we do not address where the proxy servers exactly locate. Volunteers can modify the list and add proxy servers by themselves. Client program on the volunteer's machine can invoke service through proxy server. We use <*Volunteer,Proxy* > to denote a virtual entity: service consumer. The client programs are developed with Java JDK 1.5, and the SOAP engine is Apache Axis 1.4. The client program could switch proxy server automatically and invoke the services through these proxy servers.

2. Parameters Model

In our experiment, there are two types of parameters: (1)invoking parameters, which are the parameters defined in SOAP message, and (2) running parameters, which specify how the client program runs. Basically, the data types of invoking parameters include string, integer and float. To relieve the volunteers from the burden of manually specifying the parameters in SOAP, the client program could automatically generate them. When integer or float-type parameters are required, client program will randomly generated the data. Although our experiment does not depend on volunteers' preferences, we still hope the string-type parameters to reflect the volunteers' preference to some extent. The process of generating string-type parameters goes as follows: when the client program runs at the volunteer's machine for the first time, the volunteer should specify a file directory in the computer and the client program could get all the file names in this directory. These file names are put into a table. Volunteers could modify this table as their preferences. When the client program needs a string-type parameter, it randomly picks up one in this table. Running parameters should be set by the volunteers before they run the client programs. These parameters specify:1) how many times to execute the client program per day, 2) what time to start the client programs, 3) the time interval between every two times of executing the client programs and 4) how many times the client program should invoke each service during each execution.

3. QoS Model

The invoking record is defined as a vector *R =<IPc, IPp, sn, st, et, lat, input, output >*. IP*c* denotes the IP address of the volunteer, and the IP*p* denotes the IP address of the proxy server, *sn* is the name of service, *st* denotes the time when client program sends the request, *et* denotes when the client program receives the response, *lat* is the time interval of TCP handshake between client program and service provider, *input* and *output* are request and response SOAP message. The invoking records with the same *IPc* and *IPp* are regarded as from the same service consumer. Three QoS properties, response time, availability and network latency are considered to evaluate our approach.
Response time is the time interval between a consumer sending message and receiving response [7]. For each record R, the value of response time is calculated by *et-st*. The $p^{res}_{i,j}$, which denotes the response time measured by consumer u*i* on service s*j* , is the arithmetic mean is calculated from R with identical *sn*. Availability is the probability that service will be available for use. The equation for calculating the value of availability is

$$Ava_{i,j} = (Na - Nf)/Na \quad (5).$$

IV. RELATEDWORK

QoS-aware service discovery is a hot topic and researchers have proposed to extend standard UDDI registry to enable it. Our prediction mechanism is based on these solutions and aims to be a good complement to them. Ran proposed to extend UDDI's "tModel" to describe QoS information and complement some interfaces to facilitate service retrieving[12]. UDDIe [13] is an another extended UDDI registry developed by Cardiff University. In UDDIe, QoS information is described as extended attributes of business service (which is one of core elements in UDDI data model). Accordingly, UDDI's standard interfaces are extended to enable QoS-aware service discovery and QoS information publishing.
UX is another extended UDDI which is proposed by Zhou, etc.[4]. This proposal adds a new server, named UX, to store QoS information, which is collected during interacting between consumers and services.Many researchers have proposed that service selection
should take user preferences into consideration.
Balke and Wagner employ service usage patterns and users' needs together with users' preference to enhance service discovery [3]. They also put forward a mechanism which enables cooperative discovery of web services [2].
Maamar, etc.propose a model for the context of web service interactions [14]. Besides user preferences, this work highlights the resource on which the web service is performed. These researches focus on providing a mechanism to formalize service consumers' preferences, history, scenario and resource of service provider and are based on the pre-setup semantic and ontology models that need much manual work.
Our work is quite different because in our approach, the similarity among consumers is mined among QoS data, which is collected automatically from interactions among consumers and services. Collaborative filtering is a widely used method to make recommendations.
Jonathan et al. propose an algorithm framework for performing collaborative filtering and this framework also provides some optimizing mechanisms [9]. This work is the basis of our work. To the best of our knowledge, there is no preceding work which proposes making QoS predicting with collaborative filtering technique. Most of the research for service recommendation is based on formalizing consumers' preferences

or service rating. And more, our approach have modified the calculation of the traditional collaborative filtering algorithm and achieves slightly enhancement.

Maximilien and Singh propose a conceptual model for web service reputation [11]. In this model, users could rate services and share their experiences.

N. Kokash proposes an approach to discovering services based on user previous experiences. This approach formalizes interactions between consumers and services into an implicit culture, which can be used to make recommendations to newcomers of service registries[6].

In previous work, the similarity among consumers is measured from their preferences and the ratings are based on the subjectivity of consumers. Our approach is different because the similarity mining is based on QoS data, which are objective and collected from interactions between consumers and real web services.

V. DISCUSSION AND FUTUREWORK

Our approach is based on the assumption that the consumers, who have similar historical experiences on some services, would have similar experiences on other services. The similarity between consumers is decided by many explicit and inexplicit features, such as users' preferences and network environment. Our experimental results show the reasonability of this assumption when these features are stable (we could assume that in a not long time interval, the network environment is stable). When consumers are located in unstable environment, it needs some complementary mechanisms in our approach. In our future work, we will try some AI techniques to make our approach adaptable to such situation. In our approach, the prediction is made for each QoS property separately. This separated prediction is reasonable when the QoS properties are independent, but many QoS properties are correlative, such as response time and network latency. Prediction making on the correlative QoS properties separately may lead to some errors. A possible way to solve this problem is to use regression techniques to revise some results.

In our future work, we'll try linear regression and some other regression techniques to tackle this problem. There are many optimization techniques in collaborative filtering which have proposed in [9]. One technique which is the most possible to refine our approach is the window based method. The basic idea of it is that, when making prediction, only consumers' data whose similarity is bigger than a pre-set threshold are taken into consideration. This and some other optimization mechanisms will be tried in our later refined version of this work.

Other recommending techniques, such as data mining, will also be tried in our system to improve the accuracy of predicting. To be simple and easy to understand, our experimental study have tried three QoS properties and achieved a satisfactory result. An ongoing work is to add more QoS properties, such as reliability, into our prototype system. Another future work is that we will deploy our system on the Internet to collect more real data to testify the approach.

VI. CONCLUSION

Predicting QoS for web service is a very important step for service selecting. Considering the effect of similarity among users' experiences on web services, we propose an approach based on collaborative filtering technique to improve the effectiveness of predicting.

The main contributions of this paper include:

(1) a collaborative filtering based approach for mining service consumers' similarity and predicting QoS based on objective QoS data, and

(2) a QoS data collecting mechanism and an experimental design which can be reused in other QoS prediction systems.

Our approach can significantly improve the effectiveness of predicting web services QoS.

REFERENCES

[1] B. Marko and S. Yoav. Fab: content-based, collaborative recommendation. In *Communications of the ACM*, volume 40, pages 66–72, New York, NY, USA, 1997.ACM Press.
[2] WT. Balke andW. Matthias. Cooperative Discovery for User-Centered Web Service provisioning. In *Web Services. International Conference on*, pp. 399-404, 2003.
[3] WT. Balke and W. Matthias. Towards Personalized Selection of Web Services. In *WWW*, 2003.
[4] C. Zhou, LT. Chia, B. Silverajan, and BS. Lee. Ux: an Architecture Providing QoS-Aware and Federated Supportfor UDDI. In *Web Services, 2003. International Conference on*, pages 171-176.
[5] C.K. Fung, P.C.K. Hung, and D.H. Folger. Achieving Survivability in Business Process Execution Language forWeb Services (BPEL) with Exception-Flows. *e-Technology, e-Commerce and e-Service, 2005. The 2005 IEEE International Conference on*, pages 68 – 74.
[6] N. Kokash, A. Birukou, and VD. Andrea. Web Service Discovery Based on Past User Experience. Technical report, WPI, 2007.
[7] F. Rosenberg, C. Platzer, and S. Dustdar. Bootstrapping Performance and Dependability Attributes of Web Services.In *Web Services. International Conference on*,2006, pages 205-212.
[8] C. K. Fung, P. C. K. Hung, R. C. Linger, G. Wang, and GH. Walton. A service-oriented composition framework with qos management. In *International Journal of Web Services Research*, volume Vol. 3, 2006.
[9] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl.An Algorithmic Framework for Performing Collaborative Filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA, 1999. ACM Press.
[10] A.K. Jain, M.N. Murty, and P.J. Flynn. Data Clustering:a Review. volume 31, pages 264–323, New York,NY, USA, 1999. ACM Press.
[11] E.M. Maximilien, and M.P. Singh. Conceptual Model of Web Service Reputation. *SIGMOD Rec.*, 31(4):36–41, 2002.
[12] S. Ran. A Model for Web Services Discovery with Qos. In *ACM SIGecom Exchanges*, Volume 4 , Issue 1,pages 1 - 10, 2003.
[13] A. ShaikhAli, O.F. Rana, R. Al-Ali, D.W. Walker.UDDIe: an Extended Registry forWeb services. In *Applications and the Internet Workshops, 2003. Proceedings.2003 Symposium on*, pages 85- 89, 2003.
[14] Z. Maamar, S.K. Mostefaoui, and Q.H. Mahmoud. Context for Personalized Web Services. In *System Sciences. Proceedings of the 38th Annual Hawaii International Conference on*, pages 166b– 166b, 2005.
[15] A.S. Bilgin, and M.P. Singh. A DAML-Based Repository for QoS-Aware Semantic Web Service Selection.In *Web Services, International Conference on*, page 368-375, 2005.
[16] Y. Liang, Y. Yan, and H. Liang. Composing Business Processes with Partial Observable Problem Space in Web Services Environments. In *Web Services. International Conference on*, pages 541 - 548, 2006.
[17]] RS. Pindyck, and DL. Rubinfeld. Econometrics Models and Economic Forecasts. McGraw-Hill, ISBN: 0- 07-118831-2, 2000.
[18] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, TS. Huang. Supporting Similarity Queries in MARS. In *ACM Multimedia*, pages 403-413, 1997.

