A QoS evaluation algorithm for web service ranking based on Artificial Neural Network

Lu-yi Guo
School of Software
Shanghai Jiao Tong University, SJTU
Shanghai, P.R.China
guoluyi@sjtu.edu.cn

Hao-peng Chen
School of Software
Shanghai Jiao Tong University, SJTU
Shanghai, P.R.China
Chen-hp@sjtu.edu.cn

Guang Yang
School of Software
Shanghai Jiao Tong University, SJTU
Shanghai, P.R.China
Allen.young@live.cn

Ruo-yu Fei
School of Software
Shanghai Jiao Tong University, SJTU
Shanghai, P.R.China
feisapphire@sjtu.edu.cn

Abstract—**As the number of web service providers grows, redundancy becomes prevalent with many WS providers offering the same or similar services. Many models have been proposed to measure the QoS (quality of service).This paper is trying to address the web service ranking problem based on QoS. A web service relevancy ranking algorithm based on QoS parameters has been presented for the purpose of finding the best available web service. In this ranking model, we try to find an automatic and objective way to recommend a web service. The ranking process will reduce correlation degree and extract user preference. Attributes weight will be studied and adjusted through neural network. By this ways, the accuracy of the web service ranking is improved.**

Keywords-QoS; ranking; principal component analysis; BP algorithm

I. INTRODUCTION

Web service technology is one of the most promising technologies in distributed computing area. Lately, developments in web service discovery approaches mainly focus on the concept of the QoS (quality of service). With the sharp increase of the service number, many web services are in fact providing the same functions which make QoS a very important issue in distinguishing and ranking services with similar functionality.

Previous researches have discussed about WS QoS models, definition, classification and QoS modeling and so on. Paper [1,2] propose a QoS model. Service qualities are classified into four categories: user's point of view, system level view, service level view, and business level view. Then the researchers select 5 service attributes to evaluate the web service: execution price, execution duration, reputation, successful execution rate and availability. Paper [3] introduces a method to extend the Web Service Repository Builder (WSRB) architecture by offering a quality-driven discovery of web services and uses a combination of web service attributes as constraints while searching for relevant web services. Paper [4] proposes a higher level framework for WS performance analysis and a recommendation based on the performance experienced by the client. The framework is divided into an ongoing analysis process and an on demand recommendation. Other approaches focus on improving the selection process of web services. Paper [5,6] develops a middleware for enhancing web service composition for monitoring QoS metrics. However, many of the researches mainly focus on the QoS model establishment and ranking the services in a static way. These studies may be more reasonable if the ranking process can be extended to a dynamic way.

The purpose of this paper is to consider the dynamic factors in service running and to adjust the ranking of service based on the user preference.

The rest of this paper is organized as follows: Section 2 proposes a principal component analysis (PCA) method to initial attributes weight. Then gives a train algorithm for weight adjusting based on neural network. Section 3 discusses the simulation and the results. Section 4 is the conclusion and the future work

II. RANKING MODEL

A. Assumptions

The model deals with the web services which are in the same domain.This evaluation model is used for those measurable QoS performance.

Service consumers and providers do not interact with each other directly except binding and invoking. Information communication is through the service registry. The QoS mentioned in this paper deals with the single service, QoS of composite service will be studied in the future work.

Only objective quality attributes in discrete value forms which can be observed by sever are considered. For example, response time and bandwidth are both objective and observable, however, robustness or reliability do not meet our restriction since they are not observable for a single interaction.

Only one service registry is focused in this paper.

We assume that the performance information is provided by the third party software plugged in service provider. It is considered to be objective and real-time. The details of Qos monitoring have been discussed in paper [7,8].

A web service with the highest evaluation value is considered to be the integrated optimal one.

This paper is supported by the National High-Tech Research Development Program of China (863 program) under Grant No. 2007AA01Z139

978-0-7695-3336-0/08 $25.00 © 2008 IEEE
DOI 10.1109/CSSE.2008.1108

The assumptions mentioned above are mainly to simplify the web service evaluation model, and it can be extended.

B. Web Service evalution and fix weight based on PCA

Assuming there are m web services providing the same functions, each service has n QoS attributes, so it can be described as (1).

In order to measure a web service and evaluate it, we need a basic evaluation function as formula (2).

$$Q = \begin{bmatrix} q_{11} & q_{12} & . & . & q_{1n} \\ q_{21} & q_{22} & . & . & q_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ q_{m1} & q_{m2} & . & . & q_{mn} \end{bmatrix} \quad (1)$$

$$score_i = \sum_{j=1}^{n} w_j * q_{ij} \quad (2)$$

ω represents the weight metrics. Q represents the attributes matrix. From the function above we can see that the comprehensive score of a web service relies on the quality and the weight of service attributes. The qualities of a service are fed back by the service provider automatically. Here we only concentrate on the attribute weight change.

There are several approaches to confirm the weights. Some papers adopt expert analysis method to establish weight, there are some problems. First, for different domain we need different experts. Second, it is not so reliable if we only invite a few experts to determine weight because of Man's subjective factors. Third, it has difficulty in dealing with the extension of QoS attributes. According to the expert analysis, we should invite experts again to determine weights when the QoS attributes extend.

To resolve the problems above, we suggest a method which will deduce weights objectively and automatically. Here we propose the principal component analysis (PCA) method.

PCA is a main method of covariance structure analysis in multivariate statistical. And it is also an important method of feature extraction on originally swatch in multivariate information classification [9]. It is always used to determine which vectors are significant in the data set X (set X here is a m*n dimensional vector samples). Principal component is a linear combination of random variable in algebra, but in geometry these linear combination means a new coordinate, which is obtained by rotating the original coordinate. As we know, the QoS attributes we need to calculate may have relationships with each other, for example, throughput, response time, waiting time, length of waiting queue, cost of service and so on. Intuitively, web services of large throughput may lead to less waiting time, and the shorter the waiting queue is, the less time a customer may need to wait. During our selecting based on QoS, many attributions are related with the same factor. For example, when a customer needs a web service which can run fast, there are some factors we need to take into account, like response time, throughput, latency and so on. PCA is always used to reduce dimension and correlation. For our calculation, we need PCA to reduce the correlation in initial and determine initial weight

C. PCA calculation

1) Standardize initial data matrix.

Assume there are n QoS attributes and m web services (sample size is m), X_{ij} is the j attribute of i sample, the initial data matrix is the same as (1).

We need to standardize the attribute due to the different dimensions of different attributes:

$$ST_j = \frac{q_j - E(q_j)}{\sqrt{\mathrm{var}(q_j)}} \quad \text{(j=1…..n)} \quad (3)$$

Formula (3) is used to standardize the data matrix.

$$ST = \begin{bmatrix} st_{11} & st_{12} & ... & st_{1n} \\ st_{2n} & st_{2n} & ... & st_{2n} \\ ... & ... & ... & ... \\ st_{m1} & st_{m2} & ... & st_{mn} \end{bmatrix} \quad (4)$$

ST is the standard form of Q.

$$st_{ij} = \frac{q_{ij} - \overline{q_j}}{s_j} \quad \text{where} \quad s_j^2 = \frac{1}{m}\sum_{i=1}^{m}(q_{ij} - \overline{q_j})^2$$

$$\overline{q_j} = \frac{1}{m}\sum_{i=1}^{m} q_{ij} \qquad \text{i=1,2…..m; j=1,2….n}$$

2) Compute correlation coefficient matrix.

$$R = \begin{bmatrix} r_{11} & r_{12} & ... & r_{1n} \\ r_{2n} & r_{2n} & ... & r_{2n} \\ ... & ... & ... & ... \\ r_{m1} & r_{m2} & ... & r_{mn} \end{bmatrix} \quad (5)$$

$$r_{ij} = \frac{\sum_{k=1}^{m}(q_{ki} - \overline{q_i})(q_{kj} - \overline{q_j})}{\sqrt{\sum_{k=1}^{m}(q_{ki} - \overline{q_i})^2}\sqrt{\sum_{k=1}^{m}(q_{kj} - \overline{q_j})^2}}$$

$$= \frac{1}{m}\sum_{k=1}^{m} st_{ki} * st_{kj} \quad \text{i=1,2…..m; j=1,2….n} \quad (6)$$

3) Solve principal componet.

The method to solve principal component is presented in this part. There are p nonnegative eigenvalues of correlation coefficient matrix according to $|R - \lambda I| = 0$, λ are sorted as $\lambda_1 \geq \lambda_2 \geq ..... \geq \lambda_n \geq 0$

Then we can solve the eigenvectors corresponding to λ, L is the eigenvector matrix of correlation coefficient matrix R. $L_k^T = (l_{1k}, l_{2k} ... l_{nk})$, k=1,2….n. The Kth principal component is:

$$P_k = \sum_{j=1}^{n} l_{jk} * ST_j \quad \text{k=1,2…n} \quad (7)$$

There are n eigenvector according to the n dimension matrix and a n dimension matrix will have n principal components. But we can see in fact there are only a few components are effective, while others are of little influence. We preserve those principal components whose cumulative contribution is more than 95%. Cumulative contribution is $\sum_{j=1}^{p}\lambda_j / \sum_{j=1}^{n}\lambda_j$.

The kth evaluation value of ith sample is : $Z_{ik} = \sum_{j=1}^{p} l_{jk} * st_{ij}$

Comprehensive evaluation value is $SCORE_i = \sum_{k=1}^{p} b_k * z_{ik}$

$b_k = \lambda_k / \sum_{i=1}^{n}\lambda_i$ This represents the contribution rate of variance of the kth principal component.

D. Optimizing attribute weight

The initial weight of web service is determined by PCA method presented above, which indeed is a static method. In fact, as the running of the service, the user's preference will change due to difference reasons. For example the users will be more interested in an expensive service with high performance instead of a cheap one with low performance. As the running of the services, we need to extend the algorithm to a dynamic process.

In order to solve this problem, we should focus on the weight adjust factor. Our purpose is to find an objective and automatic way which can adaptively adjust the weight. We can see the neural network is a good way to do so. Artificial neural network is a non-linear dynamic system composed by a lot of highly complex, distributing, and parallel information-processing units. It can learn from the former experiences through adjusting the connection weights and can use the knowledge learned before. In neural network, BP algorithm is a simple structured and easily implemented method and can be used to learn the weight information in our model

When a customer selects a web service, we record his selection and put it into a sample set. With the accumulation of the consumption record, we can extract some information like user preferences.

Generally speaking, a neural network needs a fixed expected value. The output compares with the expected value, shortens the distance with expected value during learning process. In our model, we use Comprehensive average score here as the standard quantity. Comprehensive average value reflects the average value of customer's selections during a period of time. To calculate the preference we use mean square error to shorten the distance between the standard score and a new sample's score.

In neural network computing, it needs to decide the number of neurons. In this paper we figure out the neurons according to the result of PCA. Neurons number of Input and output layer can be confirmed by the QoS attribute number and result number. There are n neurons according to n service attributes. Here the output layer contains one neuron which means the comprehensive value. Hidden neurons number is a contradictive issue with no final verdict now. In general word, more complex problem needs more hidden unit, more hidden unit will be easier converged. But at the same time, too many hidden units will lead to more computational complexity. According to the research of Charence N.W.Tan and Gerhard E.Wittig, Hidden layer neuron number equals the sum of input layer and output layer ones or the network architecture forms like pyramid will work better. In our model, hidden neurons are the principal components calculated by PCA. The weights between input and hidden layer indicate the proportion of attributes to a single principal component. The weights between hidden layer and output layer indicate the contribution of each principal component to comprehensive evaluation value

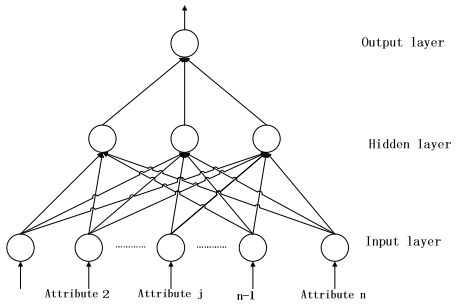

Figure 1. BP network architecture

The learning process includes forward propagation and back propagation. During forward propagation, input information is managed by hidden layer and passed to output layer. If output layer cannot get expected output, the information will be transferred to back propagation process. During the back propagation, feedback path shorten the distance between actual output and expected output by regulating weight of each neuron. Then we repeat iterations in this way for reducing the errors to the permissible error range.

Mean square error is used to present the distance between network-output and expected one. Given as:

$E = (EV - SCORE_i)^2$

EV is the excepted value. SCORE is the output of the network.

Weight correction formulation (8) is described as below:

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j y_i + \alpha[w_{ij}(t) - w_{ij}(t-1)] \quad (8)$$

$w_{ij}(t)$ is the weight joining upper and sub layer neurons at the time t. y_i is the desired output at time t, η is the step factor which controls convergence rate. δ_j is the error weight adjusting factor, denoted as (9)

$$\begin{cases} \delta_j = x_j(1-x_j)\sum_{k=0}^{n}\delta_k w_{ki} & \text{to hidden neurons} \\ \delta_j = x_j(1-x_j)(t_j - x_j) & \text{to output node} \end{cases} \quad (9)$$

t_j is the expected output value.

E. Algorithm flow

Step1 initialize network weight: Let w_{ij} to be the weights from input layer to hidden layer and w_{jk} to be the weights from hidden layer to output layer. The appropriate

initialization method is to take the weights as the results of PCA.

Step2 input a training data and compute output of every layer neuron

Step3 calculate the error item δ_{jl}^{p}、 δ_{lj}^{p}. Let the difference between the desired value and the actual value of output layer as the error, and propagate backward the error to adjust the weights of connections and threshold values.

Step4 adjust the weight according to correction formula.

Step5 calculate the output and the error according to the new weight. If the error meets the demand which is the desired value, end study, if not, repeat step2-5 until meets demand

The analysis above shows an adaptive weight process. Through the stable operation of web service system, more and more information will be accumulated. With this information, BP network will learn continuously and will fit weight to user's demand stably

III. SIMULATION

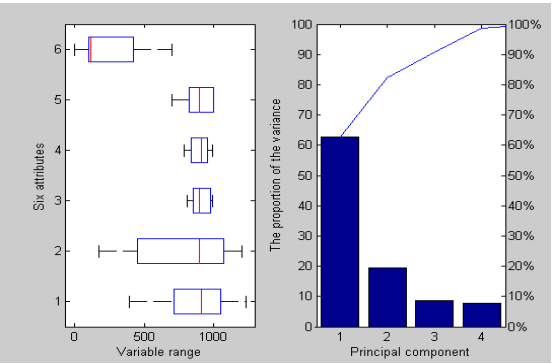

Figure 2. the attributes and the most important principal components

The left figure shows the range of QoS attributes. Six attributes [4]: response time (unit: milliseconds), throughput (unit: request/min), availability (unit: %), accessibility (unit: %), interoperability analysis (unit: % which means ratio of the errors and the warnings reported), cost of service (unit: cent per service request) are used in the simulation. The right one shows the first four principal components whose total variance contribution is more than 95%. The single rectangle in the right figure means the proportion of the variance. These four principal components will be selected as hidden layer neurons according to our model.

The attribute availability is a ratio of the time period when a web service is available. Accessibility is the probability a system is operation normally and can process requests without any delay. In our simulation, the time period is set 7 days. For the beauty of the figure, we enlarge some attributes value: throughput, availability, accessibility, interoperability analysis by multiplying a coefficient.

Figure 3 indicts a service ranking change after the neural network training. It shows that the weight of attributes can be changed by user's selection, which consequently affects the results of ranking.

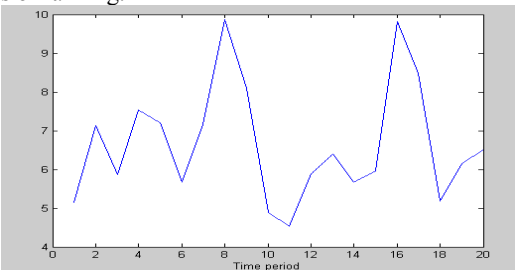

Figure 3. a service ranking change

IV. CONCLUSION AND FUTURE WORK

We propose a PCA and neural network algorithm to adjust QoS attributes weight dynamically, objectively and automatically.

Web service ranking in this paper is discussed under some precondition, in our future work, we will try to extend the model to several registry centers and to evaluate more complex attributes. Like reputation, penalty rates, reliability, fault rates and so on.

REFERENCES

[1] Shuping Ran. 'A model for web services discovery with QoS' ACM SIGecom Exchanges 2003.vol (4) :1-10

[2] Rao J,Kungas P,Matskin M 'Logic-Based Web service composition : From service description to process model. 'Proceeding of the IEEE Int'l Conf on Web Services. IEEE Computer Society,2004:446-45I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Gwyduk Yeom,Taewoong Yun, Dugki Min.'A QoS model and Testing Mechanism for Quality-driven web service selection' Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)

[4] Eyhab Al-Masri, Quasy H.Mahmoud 'QoS –based Discovery and Ranking of Web Service' Computer Communications and Networks, ICCN 2007. Proceedings of 16th International Conference on13-16 Aug. 2007 Page(s):529 - 534

[5] Charence N.W.Tan and Gerhard E.Wittig. 'QoS computation and Policing in Dynamic web service selection.' Proceeding of the 13th International Conference on World Wide Web, ACM Press.2004.66-73

[6] Maximilien, E.M.; Singh, M.P. 'A framework and ontology for dynamic Web services selection'. Internet Computing, IEEE Volume 8, Issue 5, Sept.-Oct. 2004, pp.84 – 93

[7] Deora V, Shao J, G Shercliff, et al. 'Incorporating QoS Specification in Service Discovery.' Proceeding of the 2nd Web Information Systems Workshop ,2004

[8] Tian M.Gramm A .Ritter H.Schiller J. 'Efficient selection and monitoring of QoS-aware web services with the WS-QoS framework.' IEEE/WIC/ACM International Conference on Web Intelligence(WI'04), BeiJing. China. 2004.152-158

[9] Gao Haibo, Hong Wenxue, Cui Jianxin and Xu Yonghong 'Optimization of Principal Component Analysis in Feature Extraction' Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, August 5 - 8, 2007, Harbin, China