Web Service Selection based on QoS Attributes using Entropy Discretization Method

S.Susila
Senior Lecturer
BHS Pilani Dubai

Dr.S.Vadivel
Professor
BHS Pilani Dubai

ABSTRACT
Using search engines (e.g. Google), service registries (UDDI), peer-to-peer networks, service portals, and various other sources, Web service interfaces can efficiently be searched. In order to find out relevant Web services, clients have to dedicate extreme amount of time to surf through available service resources and be capable to distinguish between services that share alike features. Discovering Web services all over diverse environments is becoming a difficult task and elevates a lot of anxieties such as performance, consistency, and sturdiness. This paper deals with ranking and selection of Web services on the basis of Entropy-Based Discretization with the help of using QoS constraints values provided by the client, and classifying them under corresponding service classifier. Using ranking (service classifier), client can easily choose the relevant Web service.

Keywords

Web service selection, Quality of service (Qos), entropy, WSRF (web service relevancy function), data normalization.

1. INTRODUCTION
This paper studies an empirical feature selection heuristics for classifying high-dimensional Qos data for web services. A feature's discriminating power can be measured by its entropy value. Based on this idea, we do not consider those features that are ignored by the entropy idea. Such a selection can usually reduce the dimensionality of the data by 90–95%. Then we rank the remaining features, and select features whose entropy is smaller than the average of all the remaining features' entropies. This round of selection can usually further reduce two thirds of the features. So, we can achieve a reduction from tens of thousands of features to only hundreds of important features. Furthermore, we also observe that learning algorithms, including our new tree classifier, generally improve their accuracy after the feature selection. This heuristics appears to be more systematic than the prevailing use of specific numbers of top-ranked features for classification.

The remainder of the paper is organized as follows: Section 2 presents related work done in selection of web services on the basis of Qos attributes of the web service. Section 3 explains about web service and the need to select web service. Section 4 describes the dataset used for web service selection. Section 5 explains our feature selection ideas and reviews a core discretization algorithm [15] that is used in our method for the first round of selection and proposes tree classifier consisting of cascading decision trees. Section 6 reports our results to show that our feature selection method is effective in the selection of web service.

2. RELATED WORK
Selection of Web Services on the Basis of Quality of Service Constraints was proposed in the paper [3]. In that QoS Manager had a role of being a moderator amongst the provider and the client. In paper [4] they discuss about Qos broker publish system that Extract the quality of service constraints in the issued WSDL and the values extracted are stored in QoSDB and the fundamental features are issued in the UDDI registryand service matching procedure is applied, and finally, service with the highest quality selected and proposed to the service requester. In paper[8] they formulates a robust QoS semantic framework for Web Services into three layers QoS-ontology, which can provide a standard model to formally describe arbitrary QoS parameters and exhibits properties [8][9][10]. Paper [11] dealt with, prominent works that apply fuzzy theory for representing imprecise QoS constraints and preferences and for developing QoS based ranking algorithm for Web services which can deal with fuzzy QoS values.In [12], the service selection for a Web service composition problem PWSDCP is dignified as contentment difficulty that belongs to a fuzzy attribute. Every QoS criterion has 5 fuzzy sets which describes the intensity of the attribute.

3. WEB SERVICES
"Web Services are encapsulated, loosely coupled contracted functions offered via standard protocols" where:

- "Encapsulated" defines the implementation of the assignment is absolutely not distinguished from the outsider.
- "Loosely coupled" defines varying the implementation of one operation does not require modification of the called operation.
- "Contracted" defines there are explicitly available definitions of the operation's activities, how to relate to the operation along with its input and output constraints.

Web services compose of the analogous 3 mechanism:

- A service-broker that acts as a request for web-service amongst a service requester and publisher.
- A service-publisher who provides his/her services to the broker of the service.
- A service-requestor who asks the service-broker where to find a suitable service-publisher and that attaches oneself to the publisher.

The communications amongst the web-services mechanism are demonstrated in the shown Figure:

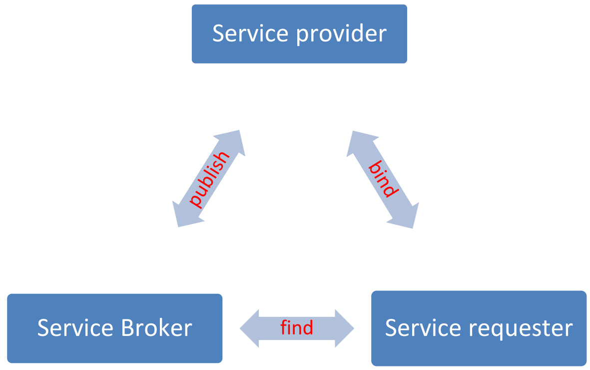

Fig 1: Web services components

Web service protocol stack.

The components of web service stack:

- "XML (Extensible Markup Language)".
- "SOAP (Simple Object Access Protocol)".
- "WSDL (Web Services Definition Language)".
- "UDDI (Universal Discovery Description Integration)".

3.1 The need for web service selection

One of the major challenges in finding out Web services is the truth that service registries do not give sufficient query elements for clients to clear appropriate service queries that can meet their requirements. For example, service registries let clients to carry out easy search queries such as searching by service or business name. However, finding out appropriate Web services could not be attained using simple keyword-based search method mainly as Web services multiply. Moreover, distinguishing Web services from each other using keyword corresponding techniques is not practical since little documented information is often made available in service discovery interfaces.Clients who are discovering appropriate Web services devote hours of searching through possible service resources by themselves. Hence, there is a desire to have a service broker that is able to collect Web service information from diverse environments (together with service portals, service registries, and search engines) and offering a central access point for clientele to clear their search queries in a resourceful method [8].

3.2 QOS for web services

The active e-business visualization calls for a flawless combination of business processes, Web services, and applications over the Internet. Carrying out QoS on the Internet is a vital and major challenge because of its vibrant and changeable nature.The dynamic electronic business idea requires a perfect arrangement of business procedures, web-services, and functions on the web. Implementing quality of service on the web is an essential and main test due to its exciting and variable character.

QoS concludes a comprehensive selection of processes that are comparable to the needs of service-requester with those of the service-publisher on the basis of the network properties available. By QoS, we talk about not ingenious configuration of web-services like reliability, ease of use, performance and security methods. The table below gives the different non-functional attributes of web services and their units.

Table 1: QWS Attributes and Units

ID	Parameter Name	Description	Units
1	Response Time	Time taken to send a request and receive a response	ms
2	Availability	Number of successful invocations/total invocations	%
3	Throughput	Total Number of invocations for a given period of time	Invokes/second
4	Success ability	Number of response / number of request messages	%
5	Reliability	Ratio of the number of error messages to total messages	%
6	Compliance	The extent to which a WSDL document follows	%
7	Best Practices	The extent to which a Web service follows	%
8	Latency	Time taken for the server to process a given request	Ms
9	Documentation	Measure of documentation (i.e. description tags) in WSDL	%
10	WsRF	Web Service Relevancy Function: a rank for Web Service Quality	%

3.3 About the dataset

The updated QWS Dataset Version 2.0 includes a set of 2,507 Web services and their QWS measurements that were conducted in March 2008 using our Web Service Broker (WSB) framework. Each row in this dataset represents a Web service and its corresponding nine QWS measurements (separated by commas). The first nine elements are QWS metrics that were measured using multiple Web service benchmark tools over a six-day period. The QWS values represent averages of the measurements collected during that period. The last two parameters represent the service name and reference to the WSDL document [17, 18, and 19].

3.4 Data Normalization

Mostly, all of the quality of service constraints varies from one another in direction as well as in value range of the utility increments. There is no comparison between them. Therefore, calculation of the weighted average of quality of service constraints is not useful. Constraint values must be transformed such that they reflect the true value in a standard range and also providing the same incrementing direction. Let's say that raw value of constraint, Q, is denoted by q, threshold value is denoted by qth and qmin denotes the minimum [3].

Data normalization of a constraint is calculated according to equation (1) if the effectiveness of it increases with the value of the constraint, q. Or else, equation (2) is applied.

$$Q' = \frac{(q-qmin)}{(qmax-qmin)'} \text{if } qmax - qmin \neq 0 \qquad (1)$$

$$Q = \frac{(qth-q)}{(qth-qmin)'} \text{if } qth - qmin \neq 0 \qquad (2)$$

On the whole, final rank value WSRF (web service relevancy function)is calculated using weighted sum of the quality of service constraints which were normalized, according to equation (3) and we get dataset.

$$V = \sum_{i=0}^{n} Wi \times Qi \qquad (3)$$

4. SERVICE CLASSIFICATION

The service classification characterizes different levels of service contributing qualities. There are four service classifications:

1. Excellent (High quality)
2. Good
3. Average
4. Poor (Low quality)

The classification is differentiated on the on the whole quality evaluation calculated by WsRF. Using WsRF values found for every Web service, we apply a classification format to relate each Web services to a particular service group. The classification can be useful to distinguish between ranges of services that offer the similar functionality. The part of the dataset is shown in figure.

Table 2

Sl. No	Response Time (ms)	Availa-bility	Throughput (invokes/	Sociability (%)	Reliability (%)	Compliance (%)	Best Practices (%)	Latency (ms)	Documen-tation (%)	WsRf	Classification
1	3321.4	89	1.4	96	73	78	80	2.6	96	-0.000053	POOR
2	302.75	89	7.1	90	73	78	80	187.75	32	0.000051	AVERAGE
3	482	85	16	95	73	100	84	1	2	0.000713	AVERAGE
4	126.17	98	12	100	67	78	82	22.77	89	0.000129	GOOD
5	107	87	1.9	95	73	89	62	58.33	93	0.000911	AVERAGE
6	107.57	80	1.7	81	67	78	82	18.21	61	0.0000803	AVERAGE
7	255	98	1.3	99	67	100	82	40.8	4	0.0000439	AVERAGE
8	136.71	76	2.8	76	60	89	6	11.57	8	0.0000503	AVERAGE
9	102.62	91	15.3	97	67	78	82	0.93	91	0.0001428	GOOD
10	93.37	96	13.5	99	67	89	58	41.66	93	0.000129	GOOD
11	133	86	7.7	95	73	78	84	10.67	9	0.0000676	AVERAGE
12	221.48	90	10.9	97	53	89	66	37.26	6	0.0000681	AVERAGE

5. ENTROPY DICRETIZATION METHOD

We first explain the basic idea of the entropy-based discretization method [15]. For a range of real values in which every point is associated with one of the two class labels, the distribution of the labels can have three main basic shapes as shown in figure 5.1:

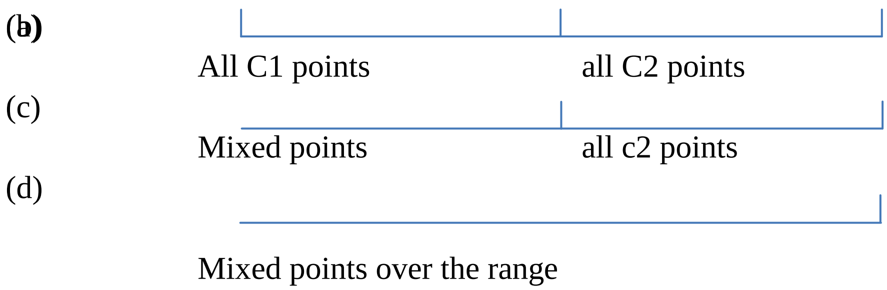

Figure 5.1: Distribution of range of points

a) Big intervals in each containing the same class of points
b) Big intervals but not all of them containing the same class of points
c) Class points randomly mixed over the range.

5.1 Entropy

Initially, the range of a continuous variable, from a database sample, is divided into intervals which contain at least one case each. This is done after sorting on the variable values. At most, there would be *m* intervals (*O (m))* for *m* cases. This converts the continuous variable into a discrete one, with *O* (*m)* values [16]. Entropy, or information, is maximized when the frequency probability distribution has the maximum number of values. Since there is a discrete partition for every distinct value in the continuous distribution in the database, there is no information or entropy loss from the database sample.

The entropy of a discrete random variable *X* is defined as

$$\text{Entropy (t)}=\sum_{i=0}^{c-1} p\left(\frac{i}{t}\right) \log 2\, p\left(\frac{i}{t}\right) \qquad (1)$$

We can use a measure called Information Gain, which calculates the reduction in entropy (*Gain in information*) that would result on splitting the data on an attribute, A.

$$\text{Gain (S,A) = Entropy (S)} \sum_{ve\,A} \frac{|Sv|}{|S|} \text{ Entropy } (Sv)$$

Where v is a value of A, |Sv| is the subset of instances of S where A takes the value v, and |S| is the number of instances

Using Entropy-based Discretization, classification of web services could be done easily. Using this classification the requester could choose the most suitable web service according to his/her requirements and preferences. Using Entropy-based Discretization we obtained a tree (Figure 6.1). In this tree, nodes belong to the QoS attributes. Tracing these nodes, we could reach leave nodes, which represent the classification of the web service into four classes.

In order to present the most suitable service to the service requester, we used the Quality of Service attributes, as they completely define a web service. In this paper, we proposed to use Entropy-based Discretization in order classify web services into four classes, i.e. Poor, Good, Average and Excellent. Using this method we obtained a classifier tree. Using this tree, we could classify any new web service into the four classes mentioned earlier by tracing the tree according to their QoS attribute values.

Many features are irrelevant to the classification. Takingsuch features into account during classification increases the dimensionality of theproblem, raises many computational difficulties, and potentially introduces noise effecton the classification accuracy [2, 12]. So, how to select important features for classificationis a problem that has been attracting tremendous research effort previously andcurrently. So we have considered only five attributes namely throughput, response time, successability, reliability and availability.

This classifier consisting of a committee of *cascading* decision trees. Each tree is constructed by using one of the top-ranked features as its root node. In our example we have used availability as root node. The learning phase of this classifier is to construct a certain number of trees.

We use the following steps to construct the tree:

Suppose n is the number of features describe a given data. To construct K (K<=n) number of trees, we use the following steps:

Step 1: Use gain ratios to rank all the features into an ordered list with the best feature at the first position.

Step 2:i=1

Step 3: Use the i th feature as root note to construct the i th tree.

Step 4: Increase i by 1 and goto Step 3, until i=K

Sorting the table (part of the table is given in fig)in ascending order according to Availability (Av)column,

Table 3

Sl. No	Response Time	Availability	Throughput	Successability	Reliability	Classification
1	255.08	12	8.1	13	53	POOR
2	64.96	18	4.3	18	60	POOR
3	68.91	19	4.4	20	60	POOR
4	451	23	1.8	24	42	AVERAGE
5	136.94	26	3.1	26	67	POOR
6	542.87	26	4.4	26	53	POOR
7	382.71	27	5.7	28	73	POOR
8	501.79	28	4.8	28	73	POOR
9	316.07	32	1	32	60	POOR

Then options for splitting points (T): 55 and 79

First, choosing T as 55:
For S1 (Av ≤ 55), we get:
P = 24
A = 3
E (S1) = 0.50325

For S2 (Av > 55), we get:
P = 11
A = 156
G = 96
C = 10
E (S2) = 1.35301

E (S|55) = 1.27653
Gain (55) = 0.26012
Now choosing T as 79, we get:

For S1 (Av ≤ 79), we get:
P = 32
A = 58
E (S1) = 0.93894
For S2 (Av > 79), we get:

P = 11
A = 156
G = 96
C = 10
E (S2) = 1.32086

E (S|79) = 1.20628
Gain (79) = 0.33037

Since Gain (79) > Gain (55), we choose splitting point as 79. Obtained incomplete tree as in figure.

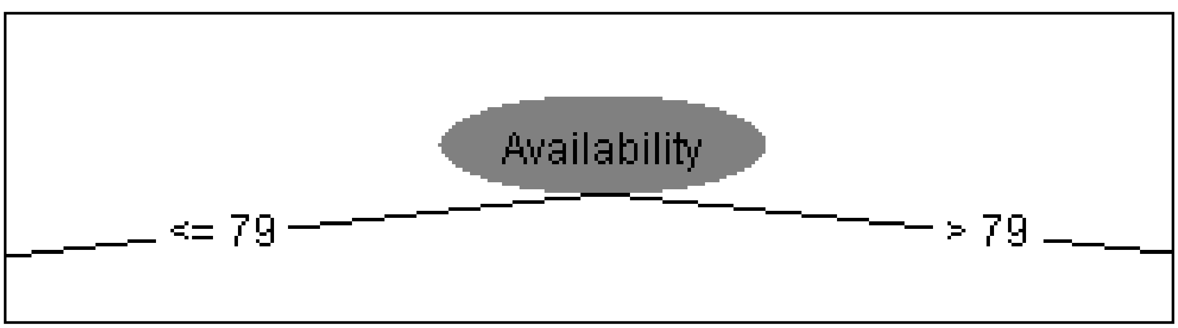

Fig 2: Obtained incomplete tree

Now, arrange the remaining table in ascending order according to Throughput (Tp). The algorithm is applied on to other columns of the table in the same way to get final tree like the one in figure.

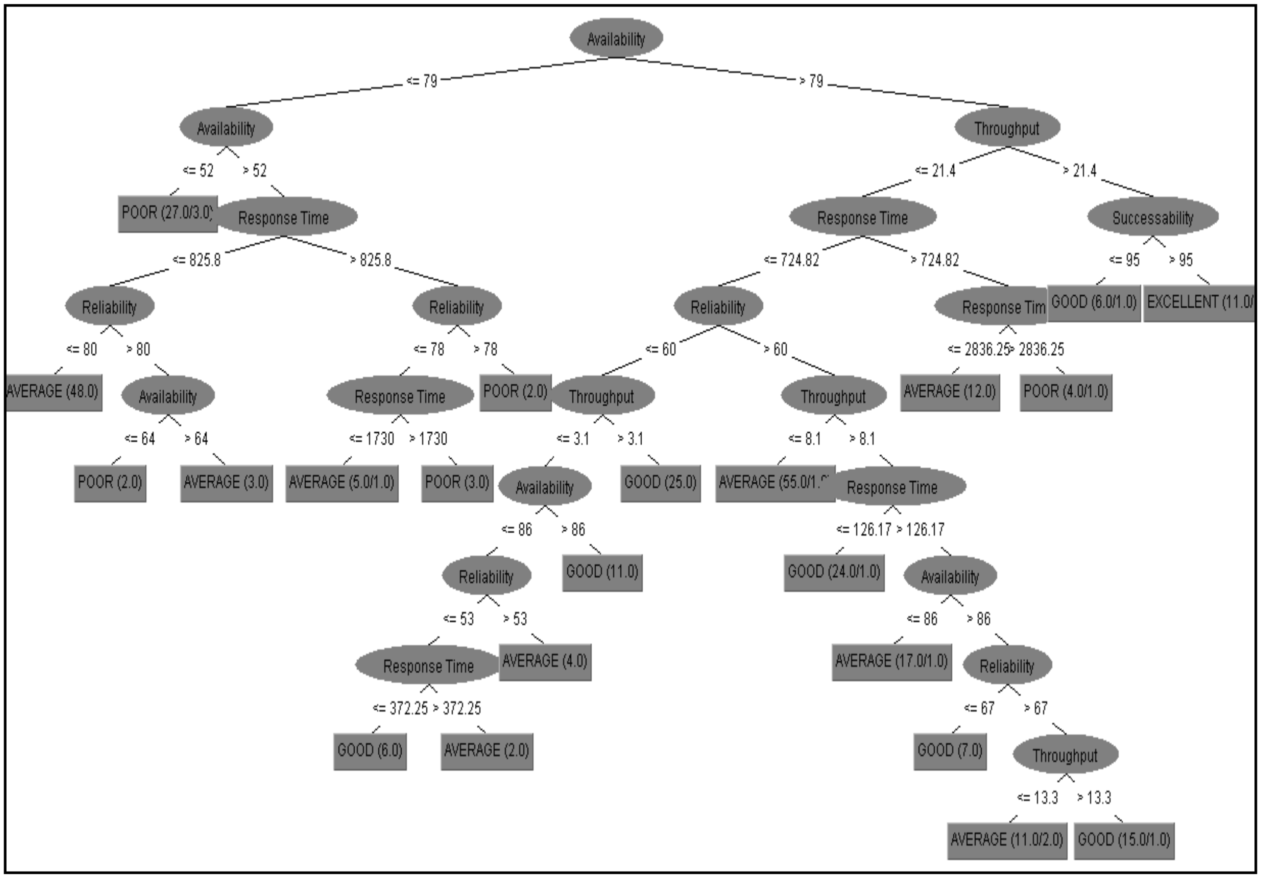

Fig 3: final tree

6. CONCLUSION

In the initial stages of service-oriented computation, in order to obtain a suitable web service one has to search into UDDI Business Registries (UBRs). Since the number of web services available were very less and countable in hundreds, there was no requirement for any advanced web-service search engine. But now, numbers of web services in the registry are increasing rapidly and hence, access points to the registries, i.e. WSDL, are no more a meagre as there are a lot of web registries, containing web services, are available on the internet.

Using Entropy-based discretization, classification of web services could be done easily. Using this classification the requester could choose the most suitable web service according to his/her requirements and preferences. Using Entropy-based Discretization we obtained a tree,in this tree, nodes belong to the QoS attributes. Tracing these nodes, we could reach leave nodes, which represent the classification of the web service into four classes.

7. REFERENCES

[1] Liang – Jie – Zhang, Jia Zhang, "Services Computing" Springer Berlin Acidelberg, Newyork.

[2] Thomas Erl, "Service Oriented Architecture: A Field Guide to Integrating XML and Web Services", Prentice Hall Publications, 2004.

[3] Jeberson R. Retna Raj, Sasipraba T., "Web Service Selection Based on QoS Constraints", IEEE, 2010.

[4] Liu Sha Guo Shaozhong Chen XinLanMingjing Zhengzhou, " A QoS Based Web Service Selection Model", International Forum on Information Technology and Applications, 2009.

[5] Manish Godse, UmeshBellur, Rajendra Sonar, "Automating QoS based Service Selection", 2010 IEEE International Conference on Web Services PP: 530-540.

[6] KyriakosKritikos and DimitrisPlexousakis, "Requirements for QoS Based Web Service

Description and Discovery”, IEEE Transactions on Services Computing, Vol. 2, 2009.

[7] Sidney Rosario, Albert Benveniste, “Probabilistic QoS and Soft Contracts for Transaction-Based

[8] Networks (ICCCN), 2007, pp. 529-534. (for QWS Dataset Version 1.0)

[9] Web Services Orchestrations”, IEEE Transactions on Services Computing, 2008.

[10] Qu Li-li, Chen Yan, “Qos Ontology Based Efficient Web Services Selection”, International Conference on Management Science & Engineering, 2009.

[11] Serhani M. A., Dssouli R., Hafid A., Sahraoui H., “A QoS broker based architecture for efficient web services selection”, IEEE International Conference on Web Services, 2005.

[12] Yu T., Lin K. J., “Service selection algorithms for Web services with end-to-end Qos constraints”, Proceeding of Information Systems and E-Business Management, 2005.

[13] Tran VuongXuan, Tsuji Hidekazu, “QoS based ranking for Web Services: Fuzzy Approach”, International Conference on Next Generation Web Services Practices, 2008.

[14] Lin M., Xie J., Guo H., Wang H., “Solving Qos-Driven Web Service Dynamic Composition as Fuzzy Constraint Satisfaction”, IEEE International Conference on E-Technology, E-Commerce and E-Service, 2005.

[15] Santhi T., Ananthanarayana V. S., D’MelloDemian Antony, “A QoS broker based architecture for web service selection”, International Conference on Modelling & Simulation, 2008.

[16] Menasce D. A., “QoS Issues in Web Services”, IEEE Internet Computing, 2002.

[17] Fayyad U., Irani K., “Multi-interval Discretization of continuou-valued attributes for classification learning, International Joint Conference on Artificial Intelligence, 1993.

[18] Ellis J. Clarke, Bruce A. Barton, “Entropy and MDL Discretization of Continuous Variables for Bayesian Belief Networks”, INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, 2000.

[19] Al-Masri, E., and Mahmoud, Q. H., "Discovering the best web service", (poster) 16th International Conference on World Wide Web (WWW), 2007, pp. 1257-1258. (for QWS Dataset Version 1.0 or QWS Dataset Version 2.0)

[20] Al-Masri, E., and Mahmoud, Q. H., "QoS-based Discovery and Ranking of Web Services", IEEE 16th International Conference on Computer Communications and or QWS Dataset Version 2.0)

[21] Al-Masri, E., and Mahmoud, Q.H., "Investigating Web Services on the World Wide Web", 17th International Conference on World Wide Web (WWW), Beijing, April 2008, pp. 795-804. (for QWS-WSDLs Dataset Version 1.0)