

## Research Article

### QOS Based Web Service Ranking Using Fuzzy C-means Clusters

<sup>1</sup>P. Parameswari and <sup>2</sup>J. Abdul Samath

<sup>1</sup>Department of MCA, Kumaraguru College of Technology,

<sup>2</sup>Department of MCA, Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India

**Abstract:** In service oriented computing ranking of best service from service registry is an essential process for service selection. The main objective of this research is to select a best service using fuzzy C-Means clustering. Identifying the best web service among all the existing services is a challenging issue. In the existing system, the ranking process uses a static priority of QoS parameters to find the best service. The first challenge is the customized prioritization of the QoS parameters and the second challenge is the multi-criterion analysis of the data. The proposed system identifies the best service using customized priority. The best service is obtained through a two-level process using fuzzy, c-means clustering algorithm for multi-criterion analysis and the threshold is calculated through the Manhattan distance algorithm. The empirical evaluation of the proposed system concludes that it reduces the time for service ranking.

**Keywords:** Fuzzy clustering, multi-criterion, QoS, ranking

#### INTRODUCTION

To find the best web service there is no particular matching condition that is most favorable. In finding the resemblance between parameters, the decrease of individual count to an overall resemblance leads to major information loss. Finding the appropriate measure of equity for these intermediate counts requires awareness of user choice, which is often not feasible or easy to obtain. This study proposes a Fuzzy C-Means clustering ranking methodology for user request, to select relatively best service from service repository. Efficiency and effectiveness are assessed based on relevant sets (Dimitrios *et al.*, 2010).

#### LITERATURE REVIEW

Finding a suitable set leads to problems such as, discovering a large number of functionally-equivalent services. This makes it difficult for users to choose which service is the best to be invoked. An algorithm for building a rule based model for ranking the web services based on Quality of Service (QoS) using fuzzy clustering and Particle Swarm Optimization (PSO) was used (Mohana and Dahiya, 2011). It contains a rule base, which consists of all the rules and a reference engine that triggers all rules to give the rank of the service as an output. The matching scores are not aggregated. The users could not formulate their QoS requirements easily and are accurately using the provided query languages. To tackle this issue and

emphasize more on the user-centered design of the service selection system and to enhance selection model that could handle both exact and fuzzy requirements and to return two categories of matching offers super-exact and partial matches and rank them using a two-level ranking algorithm (Chen and Delnavaz, 2011).

In other ways, the QoS prediction can be done by using a fuzzy clustering method by calculating the user similarity (Meng *et al.*, 2012). This method advances the prediction correctness and this is ensured by evaluating experiments with other methods. The worth of web services is assumed as a multi-dimensional object, where each dimension is one feature of the web service's non-functional properties. To get QoS values, real-world effect on the service candidates are always necessary. To evade the time-consuming and costly real-world service effect, a QoS ranking forecast framework (Zibin *et al.*, 2013) is designed by obtaining the assistance of the service selection knowledge of previous users.

The agent-based architecture is used for dynamic improvement. Design and development of an agent-based architecture and its QoS-based matching, ranking and selection algorithm for evaluating web services (Rajendran and Balasubramanie, 2010). The most suitable web service, according to the consumer's requirements is found out in the existing methods. This system is not implemented flexibly. Web service ranking algorithm based on the QoS parameter prediction was proposed. Here the creditable reports are marked then the QoS value was predicted using the

**Corresponding Author:** P. Parameswari, Department of MCA, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

fuzzy regression method and then finally the results are used as the service selecting basis (Zhai and Wei, 2012). Finding the creditable reports is a challenge. It is very difficult to precisely define the value of a QoS property. Fuzzy logic helps in representing imprecise QoS constraints. An overview of current research works which concentrates on developing QoS based ranking algorithms by using fuzzy logic. Such analysis helps to develop complete solutions in the field of Web service discovery and selection (Vuong and Hidekazu, 2008). For validation of partitions of object data produced by the Fuzzy C-Means (FCM) clustering algorithm; a number of functions are used. The role of the weighting exponent  $m$  of the FCM model-determines the validity of FCM partitions (Pal and Bezdek, 1995).

### METHODOLOGY

Our proposed method is a fusion of two systems which is for the selection of the best web service from a list of web services based on the QoS levels of each of the web services. In this system, the system consists of the process to cluster the services thereby ranking them. The new module is the gathering of input regarding the user preference of the QoS parameters and the level of values it must take based on fuzzy clusters.

The flow of the architectural model will be a looping structure which creates a loop at each step of scrutinizing the data in order to find the best service suiting the use preferences. The levels are the values that the user wants the parameter to take. Since its formation is a fuzzy cluster there are not definite values that the user can specify. The user can enter the range as high, medium and low. Our system has the following modules.

**Extracting order and level of parameter:** When the user has entered all the data to pursue the search, this is the first process. Here the parameter preferences and their levels are sorted so as to reduce the search as the user requested. The values are managed like a hash value for each item containing the parameter, preference, level. The structure of the hash is similar to the Table 1.

**Filtering based on search keyword:** The search is done purely based on the keyword entered by the user. All the web services having the keyword as the substring are filtered on the search. If the keyword is not contained in any web service name, then a empty message is displayed. The database is offline and only the web services in the database are filtered. This is due to the limited offline records used for the system.

**Calculation of fuzzy limits:** The entire limit is dynamically calculated based on the values for each web service against each QoS parameter in the database. The limit is calculated for each parameter for clustering based on the user entered preference. The formula is as follows:

Table 1: Value for each item

Preference	Parameter	Value
Values editable from 1-9 for each QoS parameter	The 9 parameters from the dataset	High/medium /low

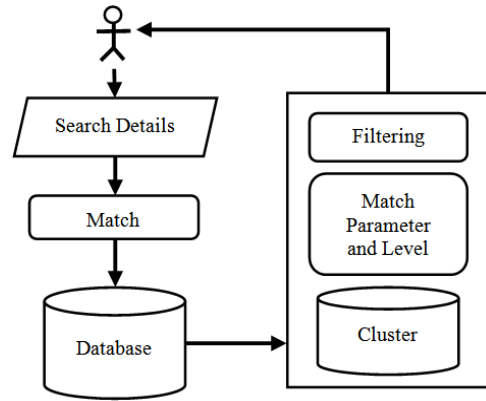


Fig. 1: Architectural diagram

$$\text{Value}_i = (\text{max} - \text{min}) / n$$

$$\text{Level1}_i = (m - 1) * \text{value}_i$$

$$\text{Level2}_i = m * \text{value}_i$$

Assume that the overlapping is 10%, hence:

$$\text{Threshold}_i = \text{level2}_i * 10/100$$

$$\text{Fuzzy limit}_i = \text{level2}_i + \text{threshold}_i$$

where,

- $i$  = The parameter 1-9
- Max = The maximum of parameter Value
- Min = The minimum of parameter Value
- Parameter value = The value of a web service against parameter  $i$
- Value = The Manhattan distance of the parameter  $i$
- $n$  = The number of levels, here its 3
- $m$  = The level (high = 2; medium = 1; low = 0)
- Level 1 = The lower limit for parameter  $i$  against selected level  $m$
- Level 2 = The upper limit for parameter  $i$  against selected level  $m$
- Number = The count of web services in the range according to selected level  $m$
- Threshold = Overlapping area of the levels for parameter  $i$ , assumed as 10%
- Fuzzy limit = Final limit for clustering into  $n$  levels

**Scrutinizing search according to parameter:** The order parameters as preferred by the user are the base for scrutinizing the web services. It is an internal

clustering of the data of the previous cluster. It can be represented through the Fig. 1. The search can be reduced to the section retrieved through the previous search. The first search is on the database over the search keyword. Later it is proceeded till the retrieval of the best service or till the clustering of all parameters. Due to the small dataset, the process is terminated within the maximum of evaluation of the first five high preferred parameters. The evaluation of all clusters can be viewed when dataset is large with varied values.

**Selection of best service:** The service that uniquely passes the selected order of the parameters is displayed as the best service for the user selected preferences. The valid preferences are recorded for future. But sometimes due to smaller dataset, there can be no data present to be retrieved for the given preference, this is handled by requesting the user to select a different set of preference and not recorded.

### RESULTS AND DISCUSSION

The first screen for the user is the screen for gathering the search keyword, parameter preferences and the levels. Only if all the preliminaries are satisfied the search is possible. The preliminaries in search keyword is not null, all the preferences are in the interval (Chen and Delnavaz, 2011; Zhai and Wei, 2012). Repeated preferences are considered as from

history and all levels must be selected either high or medium or low. The screen 1 is Fig. 2. The default marked preference is based on the history of the previous users. It is editable. The levels are unmarked and must be marked for the search along with the keyword. These values once entered in the left pane allow the user to search; else the message to enter all values is displayed in the right pane. Hence, only when all the values are entered the search is trailed.

The first process is the sorting of the parameter and their level according to the preference and it is displayed. Figure 3 shows the result of the sorted list. The parameters with the same preference are sorted based on the history and also the message stating the repetitions are considered as same preference and the search is continued. The output is extended to the search of the web services with the search keywords as the initial cluster. Then the search is scrutinized for the parameter with preference 1 and 2 with their chosen levels. Here in this case the first level search is based on the preferences and the Response Time and Availability with the second level search based on the levels are Low and high respectively. The result is the Fig. 3.

The left pane is only loaded and not the right pane. The right pane is for user editing if the preferences need to be changed at any point. The changing of preference is based on the user choice and expectation. Sometimes if the search closes abruptly, the alteration in preferences may be required.

HISTORY BASED USER PREFERENCE (Modify if Required)	FUZZY CLUSTER SELECTION (Select the Levels)
ResponseTime - 1	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
Availability - 2	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
Throughput - 3	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
Successability - 4	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
Reliability - 5	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
Compliance - 6	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
BestPractices - 7	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
Latency - 8	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
Documentation - 9	<input type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low

Fig. 2: Screen 1-to get values

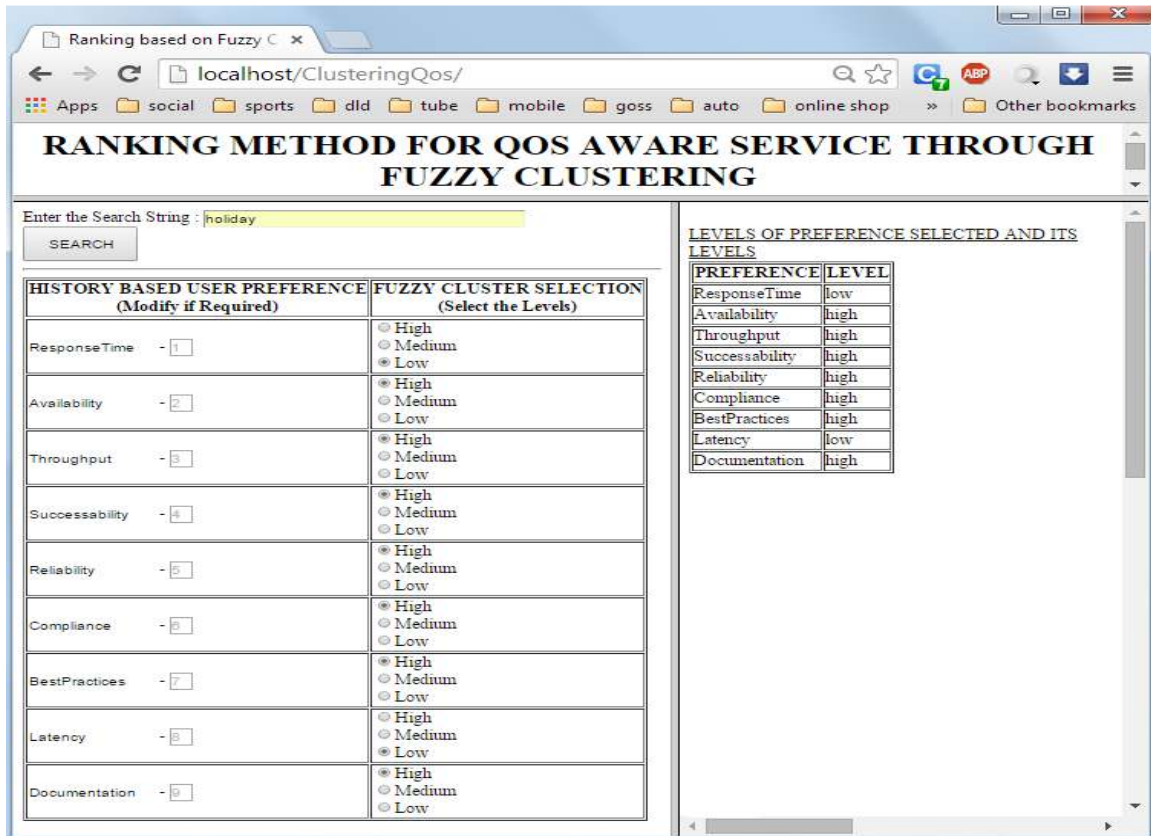


Fig. 3: Screen 2-sorting of preference values

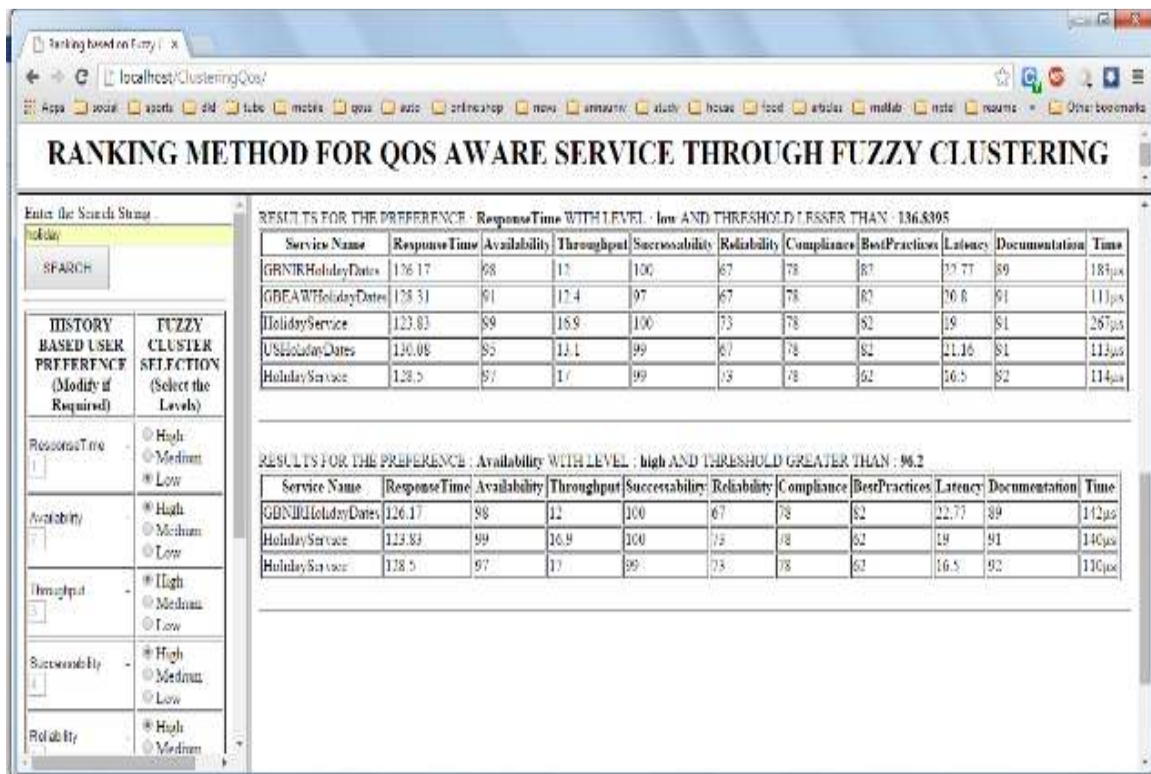


Fig. 4: Screen 3-search based on 1<sup>st</sup> and 2<sup>nd</sup> preference values



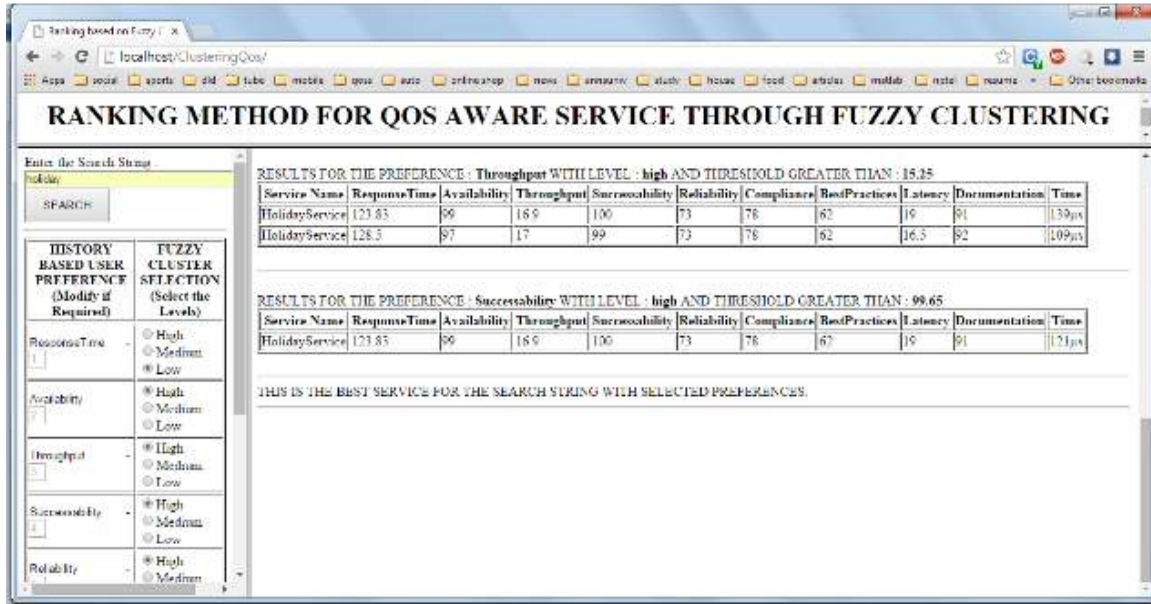


Fig. 5: Screen 4-search based on 3<sup>rd</sup> and 4<sup>th</sup> preference values

Table 2: Output consolidation

Response time	1	Low
Availability	2	High
Throughput	3	High
Success-ability	4	High
Reliability	5	High
Compliance	6	High
Best practice	7	High
Latency	8	Low
Documentation	9	High

Search string: Holiday; Best web service: Holiday service

Table 3: Best web service for user preference

Parameter	Threshold	Value achieved
Response time	<136.8395	123.83
Availability	>96.2000	99.00
Throughput	>15.2500	16.90
Success-ability	>99.6500	100.00
Reliability	= 73	73.00
Compliance	= 78	78.00
Best practice	= 62	62.00
Latency	= 19	19.00
Documentation	= 91	91.00

The parameter for first cluster is not considered for any of the latter clusters. Consider the Fig. 4 here the US Holiday Dates service has highest Response time 130 msec but it is avoided in the next cluster because it has a less availability. This shows the best service for the overall selection is displayed as the best and best service does not depend only on partial parameters in this method.

The proceeded search extends till the evaluation of the fourth parameter in the case of our selection and keyword in Fig. 5. The selection and keyword are displayed in Table 2. The Values of the best web service are displayed below along with their threshold calculated in the Table 3.

Since the best is reached at success-ability, the values for reliability, compliance, best practice, latency and documentation are assumed to be equal to the service retrieved.

## CONCLUSION

The proposed system retrieves the best service based on the user preferences and previous history of the users. The system is new and a two level search for refined and more specific search. The search is based on all the parameters entered by the user. The search is not restricted to the best service according to single or subset of parameters. The system can show diversity and be able to accept various user input rather than assuming the order of the parameter since each user requirement is unique in some way. Hence the proposed system shows a new dimension of search for the users requiring more specific search.

The main area of improvement can be the same as its advantage that is, requiring all the parameter and not specific for single parameter or partial set of parameter is not found here. The enhancement of expanding to partial and whole search can make the system versatile.

## REFERENCES

- Chen, D. and M. Delnavaz, 2011. User-centered Design of a QoS-Based Web Service Selection System. Springer-Verlag, London.
- Dimitrios, S., S. Dimitris, S. Alkis and S. Timos, 2010. Ranking and clustering web services using multicriteria dominance relationships. IEEE T. Serv. Comput., 3: 163-177.

- Meng, Z., L. Xudong, Z. Richong and S. Hailong, 2012. A web service recommendation approach based on QoS prediction using fuzzy clustering. Proceeding of the 9th IEEE International Conference on Services Computing, pp: 138-145.
- Mohana, R. and D. Dahiya, 2011. Optimized service discovery using QoS based ranking: A fuzzy clustering and particle swarm optimization approach. Proceeding of the 35th IEEE Annual Computer Software and Applications Conference Workshops, pp: 452-457.
- Pal, N.R. and J.C. Bezdek, 1995. On cluster validity for the fuzzy c-means model. *IEEE T. Fuzzy Syst.*, 3: 370-379.
- Rajendran, T. and P. Balasubramanie, 2010. An optimal agent-based architecture for dynamic web service discovery with QoS. Proceeding of the International Conference on Computing, Communication and Networking Technologies, pp: 1-7.
- Vuong, X.T. and T. Hidekazu, 2008. QoS based ranking for web services: Fuzzy approaches. Proceeding of the 4th International Conference on Next Generation Web Services Practices, pp: 77-82.
- Zhai, S. and J. Wei, 2012. A web service selecting method based on the prediction with QoS parameters. Proceeding of the IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems. Bangkok, Thailand, pp: 408-412.
- Zibin, Z., W. Xinmaio, Z. Yilei, M.R. Lyu and W. Jianmin, 2013. QoS ranking prediction for cloud services. *IEEE T. Parall. Distr.*, 24(6): 1213-1222.