Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 11, November-2018

QoS-based Web Service Ranking Model Considering Decision Making Methods

G. Vadivelou
Kanchi Mamunivar Centre for PG Studies,
Puducherry

E. Ilavarasan
Pondicherry Engineering College,
Puducherry

***Abstract* -- For selecting from a list of functionally similar services, users regularly decide based on multiple QoS criteria which they require on the target service. In selection process, different decision making strategies (compensatory and non-compensatory) are followed by various users. Recent works on QoS-based service selection systems are not considering these decision strategies while ranking the similar services, which are vital to generate exact ranking for individual users. This paper proposes a novel service ranking model based on decision strategy. Furthermore, since those different users follow diverse strategies in various contexts at different times, a learning to rank algorithm is used to learn a personalized ranking model for individual users constructed on how services are selected by them in the previous usage. Experiment result shows that the proposed approach is effective.**

Keywords – Decision strategies, learning to rank, Web services, Quality of Services (QoS)

I. INTRODUCTION AND BACKGROUND

Web services are self-contained, loosely coupled, discoverable, autonomous and dynamic entities that are available in the network. They are used to support the development of fast growing, reusable, low cost and interoperable software and applications [1][2]. Web services construction is established according to common standards that ensures the successful interaction between service providers, service requestors and service brokers.

Because of the huge number of web services, it is very difficult to discover and select the required services. Therefore, discovery and selection of web services is considered as an important challenge for the web service community. In the first phase which is the discovery phase, the user searches for a particular web service from the service directory or service registries and this search is based on the user's functional requirement. Functional requirements of web services determine the overall behavior of web services and determine if a service is relevant according to user's query. The discovery in this phase is to do the matchmaking between the functional requirements and the WSDL descriptions of web services. After the functional matchmaking, the user may get huge number of web services satisfying the functional requirements which the user has submitted.

In order to select the best functionally similar services, users must be further supported to select the suitable web service from the list, which is the selection phase. In this work, Quality of Service (QoS) attributes are considered as non-functional requirements. QoS are normally used to the show the difference between functionally equivalent services.

II. RELATED WORKS

QoS-based Web Service Selection

QoS-based service selection methods are commonly considered as an optimization problem, and thus different optimization models are used to solve this problem in order to attain the quality tradeoffs and optimizations.

In [3], users state their preferences by utility functions, quality distributions of providers are learned from a probabilistic trust model, and then the services which might maximize the anticipated utility are selected.

In [4], Mixed Integer Programming (MIP) is used to solve the problem of web service selection and match-making.

In [5], WS-QoSOnto is recommended to design and building QoS ontology for web services. The proposed ontology has several aspects to describe the QoS properties, trends, relationships and metrics. An Analytic Hierarchy Process (AHP) model is used to define the user preferences which is a multi-criteria decision making approach to develop a flexible and dynamic ranking algorithm. The main problem in this proposed model is that while calculating the overall service ranking, the proposed model totally overlooks the user defined QoS requirements and also does not take in the user defined constraints over the QoS properties.

Skyline computation is utilized to solve the problem of QoS based web service selection in [6]. In this paper, the authors deals with the issues that the users are required to convert personal preferences into numeric weights and the users may not be aware of converting these preferences into numeric weights. Also how to deal with the dynamic environment of the QoS properties in skyline approach are addressed.

Most of the above selection systems are not considering the variations of the individual decision strategies which are involved in the selection process.

Personalized Service Ranking

Personalized service ranking and selection are used in many of the research in recent years. Most of the efforts in this area uses recommendation algorithms. In [9], personal profiles are built based on the collaborative filtering method.

In [7], user similarities are measured as the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 11, November-2018

similarity between the rankings of their witnessed QoS values on commonly invoked services. Also, a QoS driven component ranking framework for cloud applications which uses the previous experiences from similar users is implemented.

In [8], former interactions between service providers and requestors are exhibited as a social network. Ranking of the web services is performed from three different aspects such as the service aspect, the consumer aspect and the non-functional properties aspect.

In [9], query and invocation logs are utilized in order to calculate user similarities and then services are ranked using most similar users' invocation accounts. Once functionally matching services are determined, ranking is done using the QoS requirements given by the current user based on how they were selected and invoked by former users with the same QoS requirements.

In [10], to calculate the similarity and to recommend services, functional interests as well as QoS preferences of users from their past usage history are utilized. Also the proposed framework finds the similarity between the consumer's functional requirements and web services. Then, a hybrid approach is proposed and developed which merges the functional similarity and non-functional similarity. Finally, the top *k* web service list is generated for the consumer considering the non-functional and functional requirements from the previous usage history.

Compared to the above mentioned research works, in this proposed work, the same problems are attacked from a totally different perception considering the previous invocation histories and query which are used. Assuming that various users might follow different decision strategies for different queries, and learning to rank algorithm [11] is used in order to learn a personalized decision-strategy-based ranking model.

Learning to Rank

Learning to rank has been widely used in information retrieval and web searching for ranking the documents and it is used to automatically construct a ranking model by means of training data for ranking objects. It has been proved in [6] that listwise ranking algorithms gives the better result compared to the pointwise and pairwise algorithms.

Freund et al. [12] developed a learning to rank algorithm called Rank Boost and he has applied it on information retrieval problem. Rank Boost is a pairwise boosting algorithm which is based on AdaBoost algorithm.

Burges et al. [13] presented the RankNet algorithm and proved the effectiveness in order to improve the search relevance. RankNet requires a label dataset for training the model. The method requires the pairwise preference information along with the gradient descent to training the model. The algorithm is simple to use and provides good performance when considering large amount of data.

Cao et al. [14] developed a listwise approach named ListNet for learning to rank and compared with the pairwise approaches such as Rank-Boost, Rank Net, Ranking SVM. Metzler and Croft [15] debated in detail the linear feature based models.

User Decision Strategies

In [16], two types of strategies such as compensatory and non-compensatory are discussed. Two usually used compensatory strategies are weighted additive (WA) and equal weight (EQW) and two generally used non-compensatory strategies are elimination by aspects (EBA) and lexicographic (LX).

In [17], more strategies which are classified based on their characteristics, such as whether all attribute values are processed or some of the attribute values are processed, whether they are attribute based or option based are discussed. The paper also illuminates the classification method which is used for detecting the user decision strategies.

According to [18], multiple strategies for making choices may be used by decision makers and then they may select strategies from a choice of strategies which represents the best accuracy and choice for the specific decision problem.

The diversity of the individual decision strategies is acknowledged in some domain-specific applications and then integrating multiple strategies which prove to offer a better result [19]. Also in this work, two aggregation strategies are joined using a single weighting strategy.

Compared to the above mentioned works, this proposed work do not want users to express their strategies explicitly, the ranking order of all alternative services instead of just finding the best one are generated automatically, and an automated solution is well-defined by using a learning to rank algorithm in order to find the optimal way for combining multiple strategies.

III. PROPOSED RANKING METHOD

After the functionally matched web services have been selected, next step is to select the web services based on the non-functional requirements. Here the main task is to find the list of optimal web services based on the user requirement on various QoS properties. This selection and ranking is based on the QoS weights, user constraints and requirements on the QoS properties which are already defined in the previous sections. For the selection and ranking of web services we are using the decision strategies which will be explained in this section.

There are many decision strategies reported in the literature [16] [17]. In this work, four of them, namely, Weighted Additive (WA) strategy, Majority of Confirming Dimensions (MD) strategy, Weighted Majority of Confirming Dimensions (WMD) and Lexicographic (LX) strategy [16] are considered. These strategies are used to reduce the number of alternatives and improve the processing efficiency. Since most of the decision strategies only target at finding optimal solutions without handling the constraints, in this work two rules are developed which could rank services based on how well they satisfy the constraints. Below, are the definitions and explanations of two rules.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 11, November-2018

- **Layer Rule:** Services satisfying all of the constraints (category 1) are ranked higher than services satisfying some of the constraints (category 2), which are ranked higher than services satisfying none of the constraints (category 3).
- **Quantity Rule:** A service satisfying more constraints is ranked higher. To simplify the case, we do not consider the user preferences on QoS criteria, and only count the number of satisfying criteria.

In the proposed work, the service ranking algorithm is based on one of the combinations between rules and strategies. The three rules (No Rule, Layer Rule, and Quantity Rule) decide how a user wants the system to handle the constraints, and the four decision strategies (WA, MD, WMD, and LX) define the preference-guided optimization process. If the number of satisfying constraints really matters to a user, Quantity Rule is applied in the ranking process. If the number of satisfying constraints is important but the exact number is not so critical, Layer Rule is applied. If a user does not care about satisfying constraints, No Rule is applied. If no rule is applicable, one of the four strategies is used directly to rank all the services. If Layer Rule is selected, services are first ranked based on the Layer Rule, and then services which fall into the same category are ranked according to one of the four strategies. If Quantity Rule is selected, services are first ranked based on the Quantity Rule, and then services which satisfy the same number of constraints are ranked according to one of the four strategies. There are in total 12 combinations, and thus we have 12 decision-strategy-based ranking algorithms.

A user may follow one decision strategy all the time when selecting services based on QoS criterion. However, it is more likely that a user may follow a few decision strategies and choose among them according to the context of search or the tasks service is used for. The user preferred strategy may also change over time. Below, a few scenarios are given that could describe the typical patterns when users follow multiple decision strategies. There could be more patterns, but in this work mainly these four are considered.

- **Pattern 1**: users follow one strategy all the time. In this case, users may only know one strategy, or are only comfortable with one strategy, and thus always use it.
- **Pattern 2**: users follow a few strategies with different probabilities. In this case, users are aware of a few decision strategies. The probability of following each strategy may depend on the context, tasks, the feasibility of the strategy, user's familiarity with the strategy, user's preference on the strategy, etc.
- **Pattern 3**: users follow a few strategies randomly. In this case, users are aware of a few decision strategies and use them constantly, however, without any obvious patterns or favorites.
- **Pattern 4**: users follow a few strategies among which some are dominating, e.g., their probabilities are much higher than the others. In this case, users have preferences on some strategies, so that they use them often, but they do not rule out other strategies and they still use them when necessary.

Among these four patterns, Pattern 4 can be considered as a special case of Pattern In order to understand which decision strategies users follow when selecting services, we could either ask them to specify explicitly or we can learn implicitly from history logs. In the former case, if they have knowledge on decision strategies, they could choose directly from a list of provided options, otherwise, if they want to spend time to fill in questionnaires, the system can identify the strategy by checking their answers.

In order for the system to learn user's service selection pattern or decision strategy pattern, the service invocation request is sent to the Invocation Proxy first, and then forwarded to the service provider. The delivered service also goes through the Invocation Proxy first and then to the user, in this way, the actual QoS data can be monitored and saved into the Monitored QoS Repository. The History Log keeps the records of all the users' search requests as well as invocation requests. Every record has the following information: user ID, query, matching services in the result list, and invoked service. With the history data, the Learning to Rank Component can learn the personalized service ranking algorithm for individual users, which could identify user's pattern on following decision strategies.

In the proposed service ranking problem, there are multiple ranking algorithms based on different decision strategies, and since users normally do not specify what strategy they follow for each selection process, we would like to learn a ranking model which could best mimic the way individual users switch between strategies according to the context and the tasks. The history log saved in the service registry can provide the training data for the learning algorithm. Through learning, the strategies a user follows as well as the best way of combining the corresponding ranking algorithms are identified, and eventually provide a personalized ranking algorithm for user's service selection. This personalized ranking algorithm is adaptive because it can be constantly learned and updated when new user data is available, and it is also extensible because more decision strategies or more QoS-based ranking algorithms can be fed into the learning model.

In this work, AdaRank [20] is used as the learning to rank algorithm. The reason for choosing it is that it can directly optimize the metrics used in the selection system, whereas many other algorithms such as RankBoost [21] define loss functions loosely related to those metrics. The metric considered in this work is Mean Reciprocal Rank (MRR) [22], which measures the accuracy of ranking based on the position of the selected result in the ranked result list, the higher the position, the higher the MRR value.

Since in the proposed system, personalized ranking is learned for each individual user, the history log is first partitioned on users. Then the training dataset for each user is represented as a collection of m records, and each record is represented as (q_i, rs_i, s_i), where q_i is the i^{th} query

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 11, November-2018

from the user, rs_i is a ranked list of returned services for query q_i, and s_i is the service the user selects from the list rs_i. The learning to rank algorithm is going to learn a ranking function, so that the ranking scores generated for the returned services for a query can optimize the performance measure MRR.

IV. EXPERIMENTS AND OBSERVED RESULTS

In the experiment, mainly the case when the explicit strategy information is not available is tested, which means the learned personalized ranking model is used to rank services. The proposed system was implemented using java language under the platform Windows 7 as the operating system and mySql as the database server.

There is no publicly available dataset for our experiment, and it is also hard to find many users to use our system so that we could collect enough usage data in the logs. Therefore, simulation is run to generate the dataset. In the simulated scenario, a user submits a QoS request, checks all the results returned from the system, and then selects one service based on a certain decision strategy. Since the decision making could be affected by the order of the results, the service selected by the user may not necessarily be the best service based on the strategy. We assume that the user is patient enough to review many results to find a good one so that it will be one of the top K results based on the strategy. Usually if the K value is not big, all the top K results can provide good results and thus the user is still satisfied with the selected service.

The QWS dataset [23] is used as our QoS dataset, which includes 2507 services. Only seven QoS properties out of the original 9 are considered, including availability, successability, throughput, documentation, compliance, best practices and reliability. There are 12 ranking algorithms considered in the work, and based on how they combine decision strategies and ranking rules, they are shown in the Table 1. Users may follow multiple strategies in different ways. Table 2 lists all the multi- strategy following patterns of users which are considered in the experiment conducted, together with their corresponding number of users.

Our dataset has are 440 users where each user has submitted 25 queries, and each query has 1 to 7 QoS requirements. For all queries, the matching services satisfying all its QoS requirements, the strategy user follows, and the service selected by the user based on the strategy. It is made sure that the number of strategies a user follows and the number of queries for every strategy matches with what are specified in the user pattern are saved. For example, if a user always uses one strategy, then all the queries from that user use that strategy for service selection and for a user using two strategies with chance 80%, 20%, then 80% of the queries use one strategy and 20% of the queries use the other one. The strategy is arbitrarily chosen from 12 strategies.

After the dataset was produced, the learning algorithm is applied where for each user's usage data, 60% is considered as training data set and 40% is used as testing dataset. The MRR metric is used for result evaluation.

In experiments conducted, the value 5 is assigned to K meaning the service chosen by the user could be one of top 5 results created on the user's strategy. Fig.1 shows the comparison between the proposed algorithm and that single strategy based ranking algorithms for every multi-strategy following patterns. The value of MRR is averaged on all the queries given by the user and the MRR value is averaged for each algorithm on all the users with the same strategy following pattern. The MRR value for the proposed algorithm is calculated on the testing data.

Fig.1 proves that the proposed learned ranking model merging multiple strategies performs much better than the ranking model considering only one strategy and also it is observed that the MRR value of the proposed algorithm is stable across all patterns. However, no individual algorithms perform steadily well for all patterns. Also it is observed that for each pattern, the best performing individual algorithm varies a lot. It shows that if the traditional ranking approach is used, considering only one strategy, it may work for some scenarios, but not all the time. Overall, the best performing algorithms are from the MD family, either one of the WMD or MD algorithms.

Many QoS-based service selection algorithms use the weighted sum (WA) as the default decision strategy. Therefore, the results from the proposed algorithm with the WA algorithm are matched as shown in Fig. 2. The perfection from proposed algorithm is obvious and if the proposed algorithm is integrated into any existing selection system, its accuracy can be improved.

V. CONCLUSIONS AND FUTURE WORKS

Regarding functionally similar services, the active user has to rank and select the services centered on non-functional requirements such as response time, throughput etc. Different users may follow diverse strategies during selection and ranking process which are classified as compensatory and non-compensatory decision strategies. The service selection system proposed in this work has been improved reflecting user's view for selecting and ranking services based on her QoS necessities and decision strategies.

As future work, the decision strategy may well be incorporated into the selection models namely CP, AHP as the base selection algorithms where only simple ranking rules are considered. To conclude, the influence of incorporating decision strategies into the service selection process in the context of service composition can also be tested.

Table 1: User Decision Strategies and Ranking Rules

User Decision Strategies and Ranking Rules	Abbreviations
Lexicographic	LX
Lexicographic + Layer Rule	LXL
Lexicographic + Quantity Rule	LXQ
Weighted Additive	WA
Weighted Additive + Layer Rule	WAL
Weighted Additive + Quantity Rule	WAQ
Majority of Confirming Dimensions	MD
Majority of Confirming Dimensions + Layer Rule	MDL
Majority of Confirming Dimensions + Quantity Rule	MDQ

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 11, November-2018

Weighted Majority of Confirming Dimensions	WMD
Weighted Majority of Confirming Dimensions + Layer Rule	WMDL
Weighted Majority of Confirming Dimensions + Quantity Rule	WMDQ

Table 2: User Pattern of Following Multiple Strategies

Pattern Name	Multi-Strategy Following Pattern of Users	Number of Users
O1	Always use one ranking strategy	10
U2	Uniformly use 2 ranking strategies	10
R2	Randomly use 2 ranking strategies	10
Dom	Some ranking strategies dominate	10
T91	Follow 2 strategies with probability 90%, 10%	10
T80	Follow 2 strategies with probability 80%, 20%	10

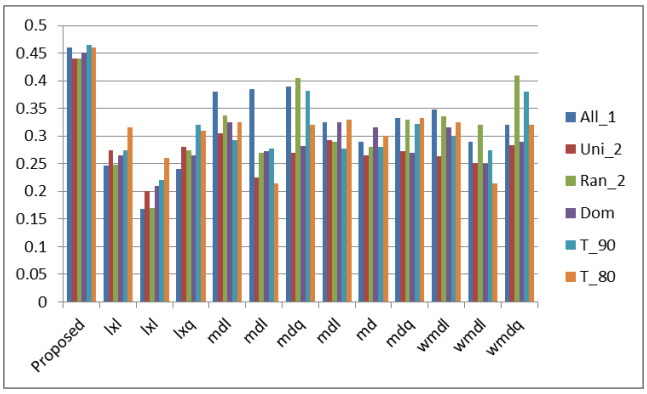

Fig.1: MRR values for all user patterns for proposed and individual algorithms

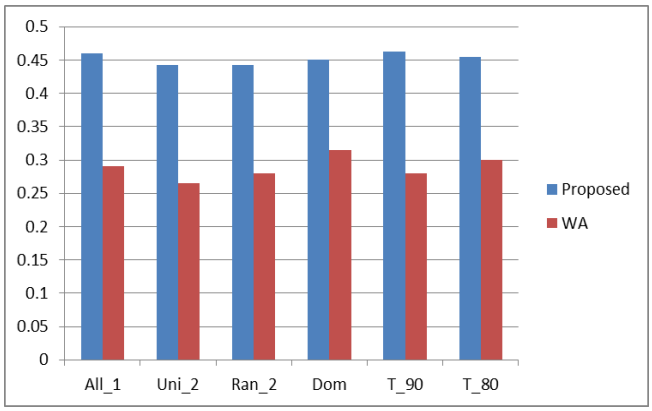

Fig. 2: Comparison of proposed method with WA method

REFERENCES

[1] M. Papazoglou, "Web Services: Principles and Technology", Pearson Education Limited, 2008.

[2] T. Erl, Service-Oriented Architecture, Concepts, Technology and Design, Prentice Hall, Indiana, 2006.

[3] C.W. Hang, and M.P. Singh, "From Quality to Utility: Adaptive Service Selection Framework", in *Proceedings of the International Conference on Service Oriented Computing*, pp. 456-470, 2010.

[4] K. Kritikos, and D. Plexousakis, "Mixed-Integer Programming for QoS- based Web Service Matchmaking", *IEEE Transactions on Service Computing*, 2(2), 122-139, 2009.

[5] V.X. Tran, H. Tsuji, and R. Masuda, "A New QoS Ontology and its QoS-based Ranking Algorithm for Web Services", *Simulation Modeling Practice and Theory*, 17(8), 1378-1398, 2009.

[6] Q. Yu, and A. Bouguettaya, "Computing Service Skyline from Uncertain QoWS", *IEEE Transactions on Services Computing*, 3(1), 16-29, 2010.

[7] Z. Zheng, Y. Zhang, and M.R. Lyu, "CloudRank: A QoS-Driven Component Ranking Framework for Cloud Computing", in *Proceedings of the 29th IEEE International Symposium on Reliable Distributed Systems*, pp. 184-193, 2010.

[8] M.O. Shafiq, R. Alhajj, and J. Rokne, "On the Social Aspects of Personalized Ranking for Web Services", in *Proceedings of the IEEE International Conference on High Performance Computing and Communications*, pp. 86-93, 2011.

[9] Q. Zhang, C. Ding, and C.H. Chi, "Collaborative Filtering Based ServiceRanking Using Invocation Histories", in *Proceedings of the International Conference on Web Services*, pp.195-202, 2011.

[10] G. Kang, J. Liu, M. Tang, X. Liu, B. Cao, and Y. Xu, "AWSR: Active Web Service Recommendation Based on Usage History", in *Proceedings of the 19th International Conference on Web Services*, pp.186-193, 2012.

[11] T.Y. Liu, "Learning to Rank for Information Retrieval", *Journal of Foundations and Trends in Information Retrieval*, 3(3), 225-331, March 2009.

[12] Y. Freund, R. Iyer, R.E. Schapire and Y. Singer, "An Efficient Boosting Algorithm for Combining Preferences", *Journal of Machine Learning Research*, 4, 933-969, 2003.

[13] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and G. Hullender, "Learning to Rank using Gradient Descent", in *Proceedings of the 22nd international conference on Machine learning*, pp. 89 - 96, 2005.

[14] Z. Cao, T. Qin, T.Y. Lin, M.F. Tsai and H. Li, "Learning to Rank: from Pairwise Approach to Listwise Learning to Rank using Gradient Descent Approach", in *Proceedings of the 24th international conference on Machine learning*, pp. 129-136, 2007.

[15] D. Metzler and W.B. Croft, "Linear feature-based models for information retrieval", *Journal of Information Retrieval*, 10(3), 257-274, June 2007.

[16] J.W. Payne, J.R. Bettman, and E.J. Johnson, "Adaptive Strategy Selection in Decision Making", *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14(3), 534-552, 1988.

[17] R. Riedl, E. Brandstätter, and F. Roithmayr, "Identifying Decision Strategies: A Process- and Outcome-based Classification Method", *Behavior Research Method*, 40(3), 795-807, 2008.

[18] J.W. Payne, J.R. Bettman, E. Coupey, and E.J. Johnson, "A Constructive Process View of Decision Making: Multiple Strategies in Judgment and Choice", *Acta Psychologica*, 80(1-3), 107-141, August 1992.

[19] I. Jeffreys, "The Use of Compensatory and Non-compensatory Multi-Criteria Analysis for Small-scale Forestry", *Small-scale Forest Economics, Management and Policy*, 3(1), 99-117, 2004.

[20] J. Xu, and H. Li, "AdaRank: A Boosting Algorithm for Information Retrieval", in *Proceeding of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 391-398, 2007.

[21] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer, "An Efficient Boosting Algorithm for Combining Preferences", *Journal of Machine Learning Research*, 4, 933-969, 2003.

[22] C.D. Manning, P. Raghavan, and H. Schütze, An Introduction to Information Retrieval, Cambridge University Press, Cambridge, England, 2009.

[23] E. Al-Masri, and Q.H. Mahmoud, "QoS-based Discovery and Ranking of Web Services", in *Proceedings of the 16th International Conference on Computer Communications and Networks*, pp. 529-534, 2007.