International Journal of Technology (2016) 5: 859-870
ISSN 2086-9614

A FOUR-LEVEL LINEAR DISCRIMINANT ANALYSIS BASED SERVICE SELECTION IN THE CLOUD ENVIRONMENT

S. Bharath Bhushan[1], Pradeep Reddy C.H.[1*]

[1] *School of Information Technology and Engineering, VIT University Vellore-632014, Tamil Nadu, India*

(Received: March 2016 / Revised: June 2016 / Accepted: July 2016)

ABSTRACT

The cloud is an outstanding platform to deal with functionally equivalent services which are exponentially increasing day-by-day. The selection of services to meet the client requirements is a subtle task. The services can be selected by ranking all the candidate services using their network and non-network Quality-of-Service (QoS) parameters, which is formulated as a NP hard optimization problem. In this paper, we proposed a linear discriminant analysis (LDA) based a four level matching model for service selection based on QoS parameters, which includes description matching of a service, matchmaking phase, LDA-based QoS matching and ranking. The LDA-service selection agent is deployed on each cloud to classify services into classes and rank the services based on the aggregate QoS value of each service. Finally, the test results show the efficiency in service selection with minimal discovery overhead, significant reduction in the computation time and the number of candidate services to be considered.

Keywords: Cloud computing; Linear Discriminant Analysis; Quality of Service; Ranking; Web service

1. INTRODUCTION

The merits of cloud powers a significant increase in number of similar services and service providers. The main cloud services: software, platform and infrastructure are provided as services on the basis of pay per use. So, selection of service provider and service selection plays a crucial role in their business activities. As the services are self-contained, loosely coupled processes deployed over a standard middleware platform can be described, published, discovered and invoked over a network. The challenge associated with cloud is in selecting the optimum required services, which are provided by different service providers with different QoS.

The selection of atomic service from a large number of similar services with a different quality of service is a multi-criteria decision problem Rajeswari et al. (2014). In this paper, we considered QoS parameters, such as response time, throughput, availability, successability and price Zhou et al. (2013). We proposed a four-level matching model to select a service from an optimal number of candidate services based on QoS criteria. The objective of our work is to achieve an efficiency in service selection with minimal discovery overhead, significant reduction in the number of candidate numbers to be considered and computation time.

Normally, the similarity computation is performed between a service request and published service based on which service is selected. In this process, the service is filtered based on service request

*Corresponding author's email: Pradeep1417@gmail.com, Tel. +91-416-2202749
Permalink/DOI: http://dx.doi.org/10.14716/ijtech.v7i5.3546

and then it assesses the QoS parameters of a service by selecting a criteria to obtain the overall score of filtered services. In order to reduce the number of candidate services and discovery overhead, the services are clustered based on QoS values by using linear discriminant analysis. None of the work has been carried out in LDA on service selection. The LDA is used for service discovery by reducing the dimension of service data and ignores the inequality of local data points of a similar class using matrix representation and calculates the overall score of QoS for each service, then ranks the services with the highest QoS values Izenman (2013). The intention of linear discriminant analysis is to detect a class to which services should belong to the closest mean. If the classes pose equal covariance and the predictor variables are subject to multivariate normal distribution, then LDA works more efficiently than any other of the discriminant analysis methods. In order to filter the candidate web services, we applied the term frequency-inverse document frequency (TFIDF) algorithm to compute the similarity degree between the service request and services published.

The contribution of this research is as follows:

1. The similar services are grouped into classes according to their QoS parameters
2. It will reduce discovery overhead
3. It reduces the number of candidate services
4. A significant reduction in computation time
5. It will ensure near optimal solution by selecting the best service from selected class

The rest of the paper is organized as follows: Related work about service selection based on QoS parameters in Section. 2. Section. 3 focuses on our proposed four level matching method and followed by an illustrative example. Experimental results and discussions are presented in Section. 4. Finally, we concluded our work in Section. 5.

2. RELATED WORK

Service selection has been a very important issue for service composition for years. This is because the improper selection of a service can affect the overall QoS of a composite service and this leads to user dissatisfaction. Researchers have adopted different approaches to select the best service from possible similar services.

In Zeng et al. (2004) they implemented a QoS based service selection through QoS ratings from the service requestor without including the context. Arasi et al. (2016) constructed a discriminant analysis model using a successability percentage of services. In Batra and Bawa (2011) they proposed a method to categorize the services into a set of already defined categories using principal component analysis. In Cardoso (2006) they computed the similarity between the service request and services published based on absolute distance between them. In Skoutas et al. (2007), the QoS requirement was detailed and structured into different classes, like security-related QoS, price-related QoS and run time-related QoS. In Papaioannou et al. (2006) they described QoS parameters which are based on QoS ontology models. In Tsesmetzis et al. (2006), they proposed a novel three-dimensional QoS model for web service discovery with a guaranteed QoS and distribution mechanism.

In Rajendran and Balasubramanie (2009), the services are selected by an agent-based web service discovery framework, which satisfies client preferences. Kalepu et al. (2004) proposed a QoS based web service selection and ranking by evaluating the reputation of a service in which it blindly considers the QoS values produced by users. Tian et al. (2004) presented a broker-based web service selection model that enables service selection based on QoS constraints. In D'Mello and Ananthanarayana (2008), also Aruna and Aramudhan, (2016) they explained the web service selection mechanism, which ranks the candidate services based on prospective levels of satisfaction of requests.

Shao-chang Li et al. (2010) proposed a heuristic algorithm for selecting services based on QoS parameters, concerning the degree of user satisfaction. Almulla et al. (2011) and Bhushan and Pradeep (2016) presented a ranking mechanism by using a fuzzy constraint satisfaction problem, in which QoS criteria are considered. Ardagna and Pernici, (2007) proposed a service composition method based on a linear programming model, but it is difficult to solve the complex problem.

In Benatallah et al. (2002), it is based on request parameters that the service selection is performed by considering the past and current execution history. Another set of authors Lim et al. (2011) and Vergin Raja Sarobin et al., (2016a or 2016b) proposed a web service selection within a community by calculating scores for other slave web services based on QoS parameters by a master web service.

3. PROPOSED FOUR-LEVEL MATCHING MODEL FOR SERVICE SELECTION

In this paper, we propose a four-level matching model for service selection based on QoS parameters, which includes description matching of a service, matchmaking process, LDA-based QoS matching and ranking method. The flowchart of the proposed method is shown in Figure 1.

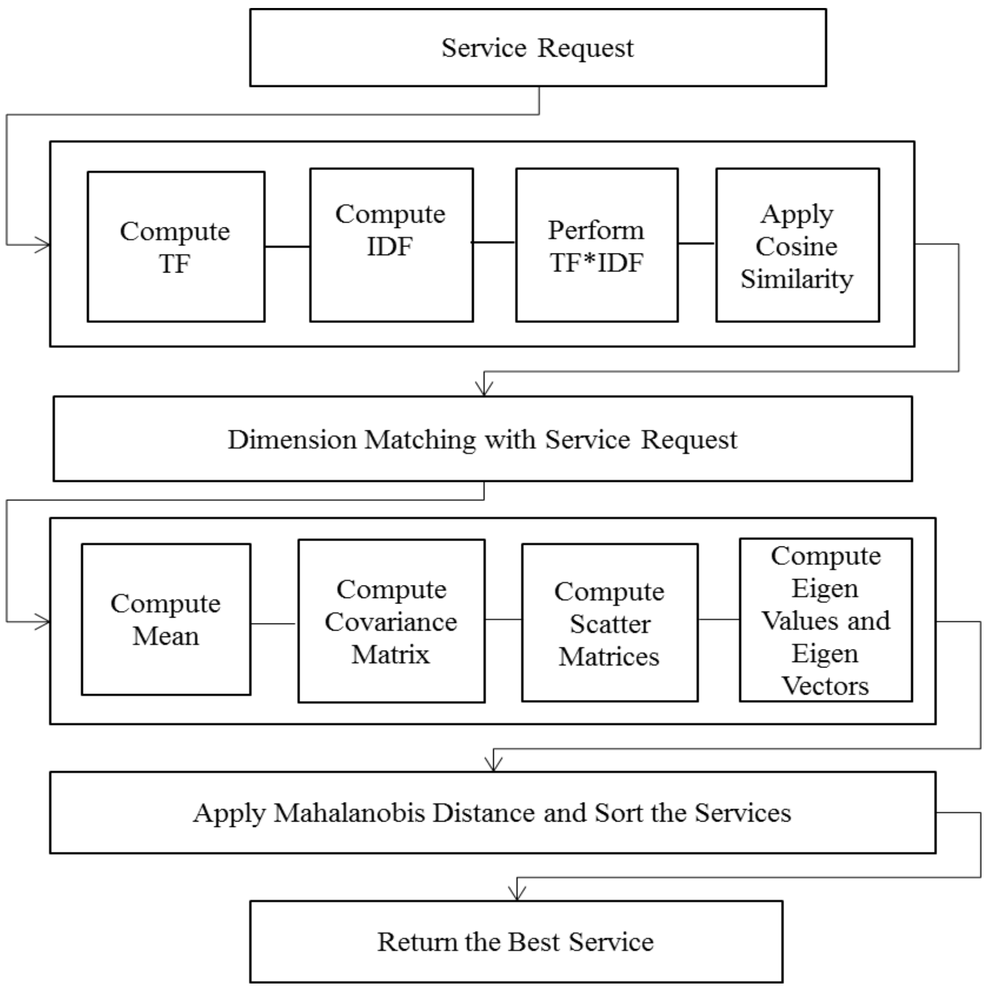

Figure 1 Flow chart of the proposed model

3.1. Description Matching of a Service

To find all similar services from multiple cloud domains, we employed term frequency (TF) - inverse document frequency (IDF) and cosine similarity to compute similarity scores of all the services with respect to the service request. Finally, the service set is filtered by applying thresholds.

The steps that are involved in description matching of a service is detailed below:

3.1.1. Term Frequency (TF)

TF measure will list down all the services based on service request. The web service description language (WSDL) will be a different size for each service, so we can adopt normalization to normalize the service description size.

3.1.2. Inverse Document Frequency (IDF)

In order to find similar services that are matched with a service request is our primary concern, but in Step 1 the terms involved in service request matching have equal importance. But in reality some terms have a minute capacity to decide relevant services and other terms will find more relevant services.

So, we need to weigh up and weigh down the terms for retrieving more relevant services from multiple clouds as shown in Equation 1:

$$IDF_{ci}(term_i) = 1 + \log_e\left(\frac{total\ number\ of\ services\ in\ a\ cloud}{number\ of\ services\ with\ service\ requester\ term\ in\ it}\right) \quad (1)$$

Aggregate the term count for all clouds by using Equation 2 shown below:

$$term_i = IDF_{c1} + IDF_{c2} + - - - - IDF_{cn} \quad (2)$$

Calculate IDF for each term in the service request from all clouds as shown in Equation 3:

$$\begin{gathered} Term_1 = IDF_{c1}(term_1) = 1 + \log_e(1) \\ Term_2 = IDF_{c1}(term_2) = 1 + \log_e(1) \\ \vdots \\ Term_n = IDF_{c1}(term_n) = 1 + \log_e(1) \end{gathered} \quad (3)$$

*3.1.3. Perform TF *IDF*

Multiply the normalized term frequency with its inverse service frequency on each service in every cloud.

3.1.4. Vector Space Model – Cosine Similarity

A vector is derived for each service, and the set of services from multiple clouds is viewed as a set of vectors in a vector space.

To find the similarity between any two services is given by the Equations 4 and 5:

$$Cosine\ similarity\ (S1, S2) = Dot\ product\left({(S1, S2)}/{\|d_1\| * \|d_2\|} \right) \quad (4)$$

$$\begin{gathered} Dot\ Product\ (S1, S2) = s_1[0] * s_2[0] + s_1[1] * s_2[2] \cdots\cdots + s_1[n] * s_2[n] \\ \|S_1\| = square\ root(s_1[0]^2 + s_1[1]^2 + \cdots + s_1[n]^2 \\ \|S_2\| = square\ root(s_2[0]^2 + s_2[1]^2 + \cdots + s_2[n]^2 \end{gathered} \quad (5)$$

where d_1,d_2 are service description documents. Based on the similarity score of all relevant services from multiple clouds, these will be reduced by applying a threshold filter.

3.2. Matchmaking Phase

The matchmaking phase selects the candidate cloud service providers based on the service request, which will be input for LDA based QoS matching. The cloud service provider must satisfy the following constraint.

3.2.1. Dimensions matching

The cloud service must have all QoS values which are in the service request. The cloud service QoS dimensions must be a superset of the service requestor QoS dimension. Equation 6 given below finds out whether the service meets the service requestor dimensions.

$$M(sp, sr) = \frac{|\{x|x \in sp \cap x \in sr\}|}{|\{x|x \in sr|}$$
$$M(sp, sr) = 1, \quad M(sp, sr) = 1$$
$$0, \ M(sp, sr) < 1 \tag{6}$$

where *sp*, *sr* is a service provider and a service requester.

3.3. Normalization

The values of various QoS parameters are different to performing numeric matching. So, the QoS parameters need to be quantified to have a uniform distribution. The QoS parameters are classified into positive and negative criteria based on their impact shown in the classification function.

The positive parameters are those with an increase in attribute value when the objective is increased and with such an increase in attribute value the objective function is going to decrease those which are negative criteria.

The positive parameters and negative parameters are normalized by Equations 7 amd 8 below:

$$N_{ij}^{+} = \frac{q_{ij} - q_j^{min}}{q_j^{max} - q_j^{min}} \quad if\ q_j^{max} - q_j^{min} \neq 0$$
$$1 \qquad if\ q_j^{max} - q_j^{min} = 0 \tag{7}$$

where N_{ij} is the normalized values of jth parameter of the ith service.

$$N_{ij}^{-} = \frac{q_j^{min} - q_{ij}}{q_j^{max} - q_j^{min}} \quad if\ q_j^{max} - q_j^{min} \neq 0$$
$$1 \qquad if\ q_j^{max} - q_j^{min} = 0 \tag{8}$$

where N_{ij} is the maximum value of the jth column of the QoS matrix, $q_j^{max} = \max(q_{i,j})$, $1 \leq i \leq n$ and N_{ij} is the minimum value of the jth column of the QoS matrix, if $q_j^{min} = \min(q_{i,j})$, $1 \leq i \leq n$. q_{min} denotes minimum QoS value and q_{max} represents the maximum QoS value. After normalization, all the QoS values lie between a [0,1] interval.

3.4. Classification Function

The classification function can be used to determine to which class each service most likely belongs. The classification function allows us to compute a classification score for each service in each class, by applying Equation 9.

$$S_i = c_i + w_{i1} * x_1 + w_{i2} * x_2 + \cdots + w_{im} * x_m \quad (9)$$

In this formula, the subscript i denotes the respective class, the subscripts $1,2 \cdots m$ denotes the m parameter, c_i is constant for the ith class, w_{ij} is the weight for the jth parameter in the computation of the classification score for the ith class. x_j is the observed value for the respective service for the jth parameter. S_i is the resultant classification score.

The services are classified into groups, which are represented as matrices consisting of similar services in a row and their QoS criteria are represented in columns as shown below in Equation 10.

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1m} \\ S_{21} & S_{22} & \cdots & S_{2m} \\ \vdots & \vdots & & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nm} \end{bmatrix} \quad (10)$$

The 4L-LDA-SC compute the service score is based on service requester weights as shown in the below column vector indicated in Equation 11:

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \quad (11)$$

where w_1, $w_2 \ldots . w_m$ are preferred weights of each QoS parameter.

3.5. Linear Discriminant Analysis

Linear discriminant analysis is used to classify the services into groups based on the service request. The proposed method classifies the services into two groups, which is a 2-class problem. The goal is to identify the best class with a good between class separability and to find the best service in the selected class with a good within class separability. The steps for performing linear discriminant analysis for service selection is illustrated below.

3.5.1. Compute Mean

Compute Mean is calcuted for two classes from the dataset. For two classes, compute the mean by using below Equation 12.

$$\mu_1 = \frac{1}{N_1} \sum_{x \in c_1} x$$
$$\mu_2 = \frac{1}{N_2} \sum_{x \in c_2} x \quad (12)$$

where N_1, N_2 is the total number of services in class $c_1$and c_2.

3.5.2. Compute Covariance Matrix

For two classes, compute the covariance matrix by using Equation 13 below:

$$S_1 = \sum_{x \in c_1} (x - \mu_1)(x - \mu_1)^T$$
$$S_2 = \sum_{x \in c_2} (x - \mu_2)(x - \mu_2)^T \quad (13)$$

where μ_1, μ_2 is the mean of two classes.

3.5.3. Compute the scatter matrices

Now we compute within-class and the between class scatter matrix, which helps in finding the best class and service with a good between class separability and within class separability.

3.5.3.1. Within-class scatter matrix (S_w)

The within-class scatter matrix S_w is computed by Equation 14 as follows:

$$S_w = S_1 + S_2 \tag{14}$$

where S_1, S_2 is the covariance matrix of two classes.

3.5.3.2. Between-class scatter matrix (S_B)

The between-class scatter matrix S_B is computed by Equation 15:

$$S_B = (\mu_1 - \mu_2)\,(\mu_1 - \mu_2)^T \tag{15}$$

where μ_1, μ_2 is the mean of two classes.

3.5.4. Compute the eigenvectors and eigenvalues

Compute the eigenvectors and corresponding eigenvalues for the scatter matrices by using Equation 16:

$$S_c^{-1} S_B C = \lambda_c \tag{16}$$

The eigenvector corresponding to the smallest eigenvalue leads to bad separability between the two classes and the eigenvector that corresponds to the highest eigenvalue leads to good separability between the two classes. To have a better discriminant class scatter matrix value, it should be large and within a class scatter matrix, it should be small.

3.5.5. Compute score

The Mahalanobis Distance (P.C. Mahalanobis, 1936) is used to find the best service from the selected class. The service with the highest score will be selected as the best service. Equation 17 below is used to calculate the Mahalanobis Distance.

$$d_m(x,y) = \sqrt{(x-m)^T S^{-1} (x-m)} \tag{17}$$

where *x* is the vector of service data, *m* is the vector of mean values, S^{-1} is the inverse covariance matrix, *T* is the indicates a vector should be transposed.

4. RESULTS AND DISCUSSION

In this paper, an LDA-based four-level matching model is proposed to rank similar services from multiple clouds based on QoS preferences. All experiments were implemented in MATLAB using an HP Pavilion dv6 laptop with 2.10 GHz Intel core processor and 2 GB RAM. The description and dimension matching is performed in the first and second stage as shown in Figure 1. Table 1 and Table 2 show the normalized QoS information of Class 1 and Class 2.

To compute between class scatter matrix and within class scatter matrix, we use the Equations 14 and 15, the resultant values are shown in Table 3 and Table 4. The S_b variable helps in identifying the best class and S_w variable finds the best service in the selected values with good within class separability.

Table 1 The normalized QoS information about Class 1 services after the description and matchmaking process

Services	Response Time	Throughput	Availability	Successability	Price
S2	0.1112	0.2816	0.5554	0.2008	0.3915
S3	0.1772	0.3392	0.2981	0.3534	0.3437
S6	0.2890	0.3392	0.4778	0.3333	0.3537
S8	0.1729	0.3392	0.1103	0.3333	0.3187
S9	0.2599	0.3392	0.3716	0.3212	0.3706
S11	0.3717	0.3392	0.1511	0.3855	0.3300
S14	0.3412	0.3392	0.1184	0.3614	0.3228
S16	0.5879	0.3392	0.3736	0.3333	0.2846
S18	0.4183	0.3392	0.2185	0.3453	0.2657

Table 2 The normalized QoS information about Class 2 services after the description and matchmaking process

Services	Response Time	Throughput	Availability	Successability	Price
S1	0.3311	0.3768	0.1166	0.2615	0.3833
S4	0.2902	0.3312	0.1747	0.3877	0.3565
S5	0.3720	0.3602	0.3011	0.2660	0.3450
S7	0.3720	0.3105	0.1709	0.4102	0.3450
S10	0.2902	0.3478	0.2837	0.4102	0.3450
S12	0.3720	0.3395	0.2722	0.1668	0.3066
S13	0.2902	0.2857	0.3687	0.3742	0.3028
S15	0.3720	0.3602	0.6455	0.1848	0.3028
S17	0.2902	0.2732	0.3589	0.4147	0.3028

Table 3 within Service Class S_w

QoS	Response Time	Throughput	Availability	Successability	Price
Response Time	10.6751	-17.6040	-0.5253	-1.8683	19.757
Throughput	-17.6040	133.3829	9.7370	21.5365	-33.834
Availability	-0.5253	9.7370	3.9565	4.5731	-0.3719
Successability	-1.8683	21.5365	4.5731	16.1673	-0.2224
Price	19.7579	33.8346	-0.3719	-0.2224	88.882

Table 4 Service between Class S_b

QoS	Response Time	Throughput	Availability	Successability	Price
Response Time	0.0443	-0.0019	0.0031	-0.0161	0.0015
Throughput	-0.1098	0.0046	-0.0077	0.0400	-0.0037
Availability	-0.0081	0.0003	-0.0006	0.0030	-0.0003
Successability	-0.0291	0.0012	-0.0020	0.0106	-0.0010
Price	0.0844	-0.0036	0.0059	-0.0307	0.0029

The optimum class selection is shown in Table 5, where C2 is the best class with a good score. From Table 6, S15 is the best service with 0.4740 aggregate score and S9 is the worst service with the least score.

Table 5 optimum class selection

Class	Score	Rank
C1	0.29816	2
C2	0.64124	1

Table 6 Overall score and ranks of services in C2

Sorted Services	Score	Rank
S15	0.4740	1
S5	0.3582	2
S10	0.3449	3
S1	0.3316	4
S4	0.2986	5
S12	0.2978	6
S13	0.2943	7
S17	0.2819	8
S7	0.2724	9

In Figure 2, we compared our proposed Four-Level-Linear Discriminant Analysis Based Service Selection (4L-LDA-SC) with QoS Based-Web Services Selection Method (WSSM-Q), QoS Aware Service Selection Based on Clustering (SCB-QC) methods with respect to the number of candidate services required for execution of tasks. The existing WSSM-Q method selects services based on QoS constraints by evaluating all the candidate services without filtering, which shows an impact on computation time and optimality The SCD-QC adopted a clustering technique to reduce the number of candidate services required for the evaluation process, but still it fails to attain an optimum number of candidate services. The 4L-LDA-SC dominates the other two methods with significant reduction in the number of candidate services required.

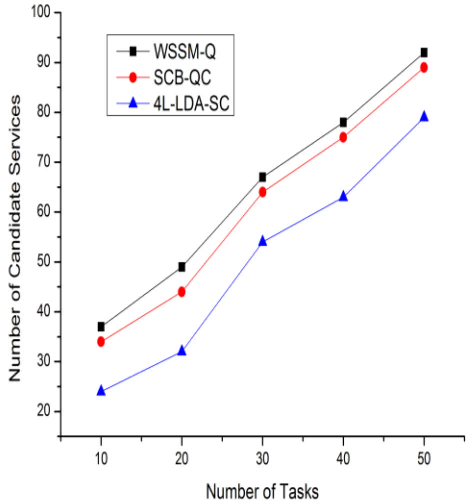

Figure 2 Candidate services discovered by varying number of tasks

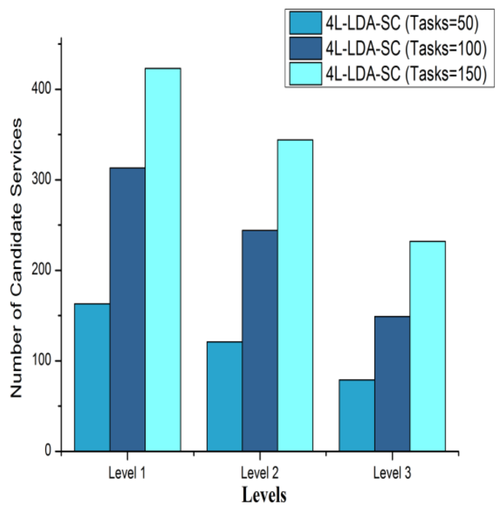

Figure 3 Total number of services discovered at each level

Figure 3 shows the importance of candidate service filtering at each level with respect to the variation of tasks. The candidate services at Level 3 will be considered for evaluation to find the best service.

In Figure 4, the proposed 4L-LDA-SC attains good computation time with respect to the variation of tasks from 20 to 100. The reduction in the number of candidate services has shown influence on computation time. The proposed method filters the candidate services by applying cosine similarity and dimension matching.

The quality of the proposed method is evaluated by comparing the overall classification function value of the selected services with an overall classification function value of the optimal selection obtained by the proposed method. Figure 5. shows the optimality value of different methods with respect to the variation of the number candidate per class. The proposed method achieves the best result with more than 96% optimality ratio on average.

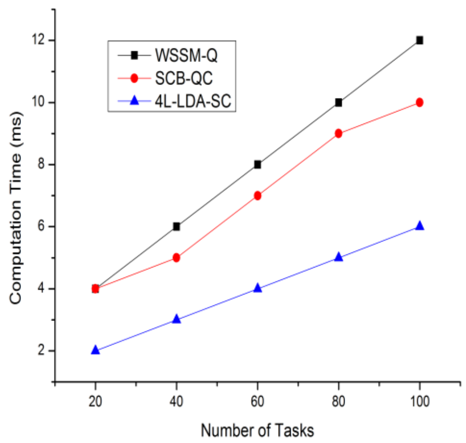

Figure 4 Computational time with respect to problem size

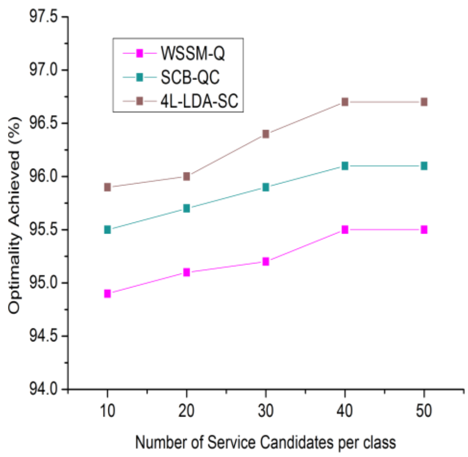

Figure 5 Optimality comparison with respect to the problem size

5. CONCLUSION

This paper proposes a four-level linear discriminant analysis based service selection (4L-LDA-SC) method to select a better service. The description and dimension matching of service filters the candidate services, which is input to LDA-based QoS matching. The candidate services are divided into two classes from which best class and service is selected. The proposed method achieves a reduced number of candidate services for evaluation and significant reduction in computation time over differing tasks. The 4L-LDA-SC method finds best service with more than 96% optimality ratio on average. In future work, we will use a vneural network for classification and prediction of services based on user preferences.

6. REFERENCES

Ardagna, D., Pernici, B., 2007. Adaptive Service Composition in Flexible Processes. *IEEE Transactions on Software Engineering*, Volume 33(6), pp. 369–384

Almulla, M., Almatori, K., Yahyaoui, H., 2011. A QoS-based Fuzzy Model for Ranking Real World Web Services. *In*: IEEE International Conference on Web Services (ICWS), pp. 203–210

Arasi, F.E.M., Govindarajan, S., Subbarayan, A., 2016. Discriminant Analysis of Web Services Successability. *Journal of Applied Sciences*, Volume 16(5), pp. 223–229

Aruna, L., Aramudhan, M., 2016. Framework for Ranking Service Providers of Federated Cloud Architecture using Fuzzy Sets. *International Journal of Technology*, Volume 7(4), pp. 643–653

Bhushan, S.B., Pradeep, R.C.H., 2016. A Network QoS Aware Service Ranking using Hybrid AHP-PROMETHEE Method in Multi-cloud Domain. *International Journal of Engineering Research in Africa*, Volume 24, pp. 153–164

Benatallah, B., Dumas, M., Sheng, Q.Z., Ngu, A.H., 2002. Declarative Composition and Peer-to-peer Provisioning of Dynamic Web Services. *In*: Proceedings of the 18th International Conference on Data Engineering, pp. 297–308

Batra, S., Bawa, S., 2011. Semantic Discovery of Web Services using Principal Component Analysis. *Physical Sciences*, Volume 6(18), pp. 4466–4472

Cardoso, J., 2006. Discovering Semantic Web Services with and without a Common Ontology Commitment. *In*: Services Computing Workshops, 2006. SCW'06. IEEE, pp. 183–190

D'Mello, D.A., Ananthanarayana, V.S., 2008. A QoS Model and Selection Mechanism for QoS-aware Web services. *In*: Proceedings of the International Conference on Data Management (ICDM 2008), Delhi, February, pp. 25–27

Kalepu, S., Krishnaswamy, S., Loke, S.W., 2004. Reputation=f(user ranking, compliance, verity). *In*: Proceedings of the IEEE International Conference on Web Services, pp. 200–207

Lim, E., Thiran, P., Maamar, Z., Bentahar, J., 2011. Using 3-way Satisfaction for Web Service Selection: Preliminary Investigation. *In*: IEEE International Conference on Services Computing (SCC), pp. 731–732

Li, S.C., Chen, H.P., Chen, X., 2010. A Mechanism for Web Service Selection and Recommendation based on Multi-QoS Constraints. *In*: The 6th World Congress on Services (SERVICES-1), pp. 221–228

Izenman, A.J., 2013. Linear Discriminant Analysis. *In*: Modern Multivariate Statistical Techniques, Springer New York, pp. 237–280

Papaioannou, I.V., Tsesmetzis, D.T., Roussaki, I.G., Anagnostou, M.E., 2006. A QoS Ontology Language for Web-services. *In*: The 20th International Conference on Advanced Information Networking and Applications, (AINA 2006),Volume 1, pp. 6

Rajeswari, M., Sambasivam, G., Balaji, N., Basha, M.S., Vengattaraman, T., Dhavachelvan, P., 2014. Appraisal and Analysis on Various Web Service Composition Approaches based on QoS Factors. *Journal of King Saud University-Computer and Information Sciences*, Volume 26(1), pp. 143–152

Rajendran, T., Balasubramanie, P., 2009. An Efficient Framework for Agent-based Quality Driven Web Services Discovery. *In*: International Conference on Intelligent Agent & Multi-Agent Systems, (IAMA 2009), pp. 1–2

Ran, S., 2003. A Model for Web Services Discovery with QoS. *ACM Sigecom Exchanges*, Volume 4(1), pp. 1–10

Skoutas, D., Simitsis, A., Sellis, T., 2007. A Ranking Mechanism for Semantic Web Service Discovery. *In*: IEEE Congress on Services, pp. 41–48

Tian, M., Gramm, A., Ritter, H., Schiller, J., 2004. Efficient Selection and Monitoring of QoS-Aware Web Services with the WS-QoS Framework. *In*: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 152–158

Tsesmetzis, D.T., Roussaki, I.G., Papaioannou, I.V., Anagnostou, M.E., 2006. QoS Awareness Support in Web-service Semantics. *In*: International Conference on Advanced International Conference on Telecommunications/Internet and Web Applications and Services, pp. 128–128

Vergin Raja Sarobin, M., Ann Thomas, Linda, 2016a. Improved Leach Algorithm for Energy Efficient Clustering of Wireless Sensor Network (WSN). *International Journal of Technology*, Volume 7(1), pp. 50–60

Vergin Raja Sarobin, M., Ganesan, R., 2016b. Bio-inspired, Cluster-based Deterministic Node Deployment in Wireless Sensor Networks. *International Journal of Technology*, Volume 7(4), pp. 673–682

Zhou, W., Wen, J., Gao, M., Liu, J., 2013. A QoS Preference-based Algorithm for Service Composition in Service-oriented Network. *Optik-International Journal for Light and Electron Optics*, Volume 124(20), pp. 4439–4444

Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H., 2004. QoS-aware Middleware for Web Services Composition. *IEEE Transactions on Software Engineering*, Volume 30(5), pp. 311–327