Diversified Quality Centric Service Recommendation

Yiwen Zhang[1], Lei Wu[1], Qiang He[2, 1*], Feifei Chen[3], Shuiguang Deng[4] and Yun Yang[2, 1]
[1]*School of Computer Science and Technology, Anhui University, Hefei, Anhui, China*
[2]*School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia*
[3]*School of Information Technology, Deakin University, Melbourne, Australia*
[4]*College of Computer Science and Technology, Zhejiang University, Hangzhou, China*
Email: zhangyiwen@ahu.edu.cn; wuleiahu@qq.com; qhe@swin.edu.au; feifei.chen@deakin.edu.au; dengsg@zju.edu.cn; yyang@swin.edu.au

Abstract—**In the past decade, a large number and variety of services have been published on the Internet, e.g., public services, web services, cloud services, etc., offering customers a wide range of choices for fulfilling their demands. This has made it a challenge for customers to select from many mutually-substitutable competing services, especially when they are offered with differentiated quality. Thus, service recommendation that helps customers find the right services has become of paramount research and practical importance. The correlations in customers' preferences for different quality dimensions make service recommendation an even more complicated problem. It has not been properly addressed by existing approaches. This paper proposes DQCSR (diversified quality centric service recommendation), an approach that finds diversified services that are representative in different quality dimensions with respect to customer's quality preferences. The results of experiments conducted on a real-world dataset demonstrate the representativeness and diversity in the recommendation results produced by DQCSR.**

Keywords-service recommendation; quality correlation; diversity

I. INTRODUCTION

The development and popularity of e-business, e-commerce and cloud computing have fueled the rapid growth in the number of services published on the Internet. These services offer customers a broad range of choices for fulfilling their demands. As the number of services continues to increase, it has become more and more difficult for customers to select the right services, especially when many mutually-substitutable services are available, e.g., restaurants, virtual machines, etc. Mutually-substitutable services usually differ from each other by their non-functional properties, e.g., ratings and prices of restaurants, latency and failure rates of web services, which are together referred to as *quality* in this paper. The key to finding the right services for a customer is to find those whose quality is preferable [1], [2] with respect to customers' personalized quality preferences.

Service recommendation approaches proposed in recent years can be categorized into four major categories, i.e., utility-based [3], [4], skyline-based [1], collaborative filtering (CF) based [5] and matrix factorization (MF) based [6]. None of those approaches considers customers' quality preferences. There is an urgent need for a personalized quality centric service recommendation approach for finding services that are most likely to fulfill the given customer's quality preferences.

Zhang et al. attempted to fulfill this need [7]. However, their approaches failed to consider the quality correlations among different dimensions of customers' quality preferences (referred to as *quality correlations* for short hereafter) [8]. For example, a customer may be willing to accept a reasonably higher latency if the failure rate of a service exceeds his/her expectation. There might also be customers that accept a restaurant's ratings slightly higher than his/her preference at a price lower than his/her budget. Personalized quality centric service recommendation must take such quality correlations into account to ensure both the *representativeness* and *diversity* in the recommendation result. Here, representativeness requires that for any service in the recommendation results there must not be any other services that are more similar to customer's quality preference in terms of all quality dimensions. For example, suppose that a customer's quality preferences are latency = 100ms, failure rate = 0.01 and service s_1 with latency = 95ms and failure rate = 0.05 and s_2 with latency = 90ms and failure rate = 0.03. Service s_1 is considered non-representative because s_2's quality is closer to customer's quality preferences than s_1's quality in both latency and failure. Formally speaking, s_2 dynamically dominates s_1. Diversity takes a step further by requiring that the recommendation results should include services that are representative in as many quality dimensions as possible. For example, given a customer's quality preferences for latency and failure rate, the recommendation results should ideally include three types of representative services with: 1) low latency 2) low failure rates; and 3) balanced latency and failure rates. Modeling quality correlations [9] is challenging because it is often too demanding for a customer to formularize or numerically specify his/her multi-dimensional quality correlations.

To address the above issues, this paper proposes DQCSR, a new approach for **D**iversified **Q**uality **C**entric **S**ervice **R**ecommendation. Given a customer's quality preferences s_r and a set of candidate services S, DQCSR first finds all services that are not dominated by any other services in S with respect to s_r. Those services are referred to as dynamic skyline services, denoted by S_{DSL}. Then, DQCSR partitions S_{DSL} with k-means into k clusters. Finally, DQCSR selects one service from each cluster based on cluster centers or coverage regions to generate the final recommendation results. In this way, DQCSR ensures both the representativeness and diversity in the recommendation results.

The major contributions of this paper include:

1. It is the first attempt to handle customers' uncertain quality correlations in service recommendation.
2. A method is proposed for finding representative services that are not dynamically dominated by any other services. This ensures the representativeness in the recommendation results.
3. A method is proposed for clustering dynamic skyline services. After that, two methods are proposed to generate the final recommendation results. This ensures the diversity in the recommendation results.
4. Extensive experiments are conducted on a real-world dataset to evaluate the effectiveness of DQCSR against four state-of-the-art approaches.

The remainder of this paper is organized as follows: Section II analyzes the research problem. Section III introduces DQSCR. Section IV evaluates DQSCR. Section V discusses the threats to validity. Section VI reviews the related work. Section VII concludes this paper and points out future work.

II. Problem Analysis

This section analyzes the problem studied in this research with a motivating example. Given a customer's quality preferences represented by s_r hereafter, we need to select a number of services from a set of nine candidate services $S = \{s_1, s_2, \ldots, s_9\}$ to generate the recommendation results, represented by R, as presented in Fig. 1. Table 1 lists the notations that are used in this paper.

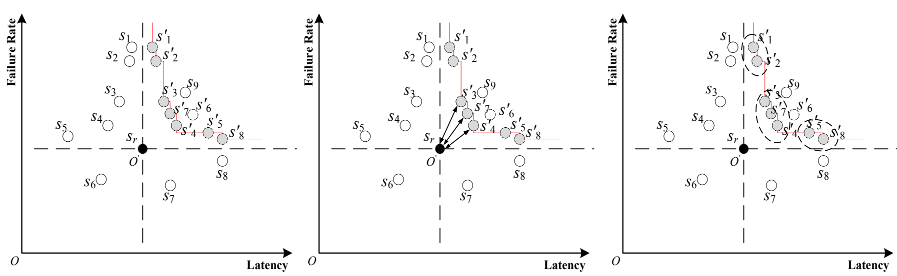
First of all, we need to prune the services that are not representative. This can be achieved by finding the dynamic skyline services. Fig. 1(a) presents the identification of dynamic skyline services in S, i.e., $S_{DSL} = \{s_1, s_2, s_3, s_4, s_5, s_7, s_8\}$ represented by light grey points. Services s_6 and s_9 are pruned because they are dynamically dominated by at least one of the services in S_{DSL}. In this example, s_6 has the lowest latency and failure rate. However, with such outstanding quality advantages, it is expected that its price will violate customer's preference for cost. In a 3-dimensional space with cost as the third dimension, s_6 might be one of the services most similar to s_r. However, in Fig. 1, it is not considered representative with respect to s_r. Fig. 1(a) illustrates a major limitation of the dynamic skyline technique – the number of dynamic skyline services can be excessive. This limitation is especially critical when the number of candidate services or the number of quality dimensions is large because it is very hard for the services in S to dynamically dominate each other.

Fig. 1(b) illustrates DSL-KNN, an approach proposed in [7] for tackling this limitation. This approach takes a step further to reduce the number of services in R to k by employing the top k nearest neighbors (KNN) method to select the top k services from S_{DSL} that are most similar to s_r. As demonstrated in Fig. 1(b), the services recommended by DSL-KNN are $\{s_3, s_4, s_7\}$ represented by dark grey points. Services s_1 and s_2 are both excluded from the results even though their latencies are most similar to the customer' preference. However, s_1 and s_2 are in fact preferable to customers who are willing to accept a high failure rate for a satisfactory latency. Similarly, s_5 and s_8 are of customers' interest who prioritize failure rate over latency. Thus, DSL-KNN cannot ensure the diversity in R.

To further reduce the number of services in S_{DSL} without losing diversity in R, S_{DSL} can be partitioned to cluster services that share similar quality representativeness. Take Fig. 1(c) for example, S_{DSL} is partitioned into three clusters, i.e., $\{s_1, s_2\}$, $\{s_3, s_4, s_7\}$ and $\{s_5, s_8\}$. In the meantime,

TABLE I. Notations

Notation	Description
S	A set of candidate services
s_i	The ith service in S
s_r	A customer's quality preferences
$s_i^{'}$	The projection of s_i in the new space with s_r as the origin
$q_p(s_i)$	The pth quality of service s_i
S_{SL}	The skyline services in S
$S_{DSL:}$	The dynamic skyline services in S
$s_i \rhd s_j$	s_i dominates s_j
$s_i \succ s_j$	s_i dynamically dominates s_j with respect to s_r
C	A cluster of services by clustering
$cr(s_i)$	The coverage region of s_i
$distance(s_i, s_j)$	The Euclidean distance between s_i and s_j



(a) DSL (b) DSL-KNN (c) DQSCR

Figure 1. Approaches for service recommendation. Service $s_i^{'}$ is the projection of s_i in the new space with s_r as the origin. The identification of dynamic skyline services will be discussed in detail in Section III-A. In this section, $s_i^{'}$ and s_i are interchangeable in the discussion.

services from different clusters are differentiated by their quality representativeness. Then, one service is selected from each cluster to generate the final k recommendation results. Here, k can be manually specified by the customer or empirically specified by the recommendation system. Fig. 1(c) shows that s_1, s_7 and s_8 are selected to be included in R. This way, the services in R are properly diversified.

III. RECOMMEDATION APPROACH

This section introduces DQCSR, our approach for diversified quality centric service recommendation that addresses the issues discussed in Section II. As demonstrated in Fig. 2, given a customer's quality preferences s_r, DQCSR goes through three phases to make service recommendations, i.e., dynamic skyline service identification, service clustering and service selection. This section will discuss those phases in detail.

A. Dynamic Skyline Service Identification

The identification of dynamic skyline service is based on the concept of skyline service. Given a set of points in a d-dimensional space, skyline calculation aims to find points that are not dominated by any other points. A point s_i dominates another point s_j, if s_i is better than or equal to s_j in all dimensions and strictly better in at least one dimension. In the context of this research, the dominance relations between two services are defined based on their d-dimensional quality values:

Definition 1 (Dominance). Given two services, $s_i, s_j \in S$, characterized by d-dimensional quality values, s_i dominates s_j, denoted by $s_i \rhd s_j$, if s_i is as good as or better than s_j in all quality dimensions and better in at least one quality dimension, i.e., $\forall\ p \in [1, d]$: $q_p(s_i) \leq q_p(s_j)$ and $\exists\ p \in [1, d]$: $q_p(s_i) < q_p(s_j)$.

Based on Definition 1, we formally define the concept of skyline services:

Definition 2 (Skyline Service). The skyline of S, denoted by S_{SL}, consists of the set of services in S that is not dominated by any other services in S, i.e., $S_{SL} = \{s_i \in S \mid \neg\exists\ s_j\text{: } s_j \rhd s_i\}$. The services in S_{SL} are referred to as skyline services.

Generally speaking, the skyline services have the best quality according to their absolute quality values in each quality dimension. However, given s_r, we need to identify the dynamic skyline services in S with respect to s_r. This can be achieved in a new d-dimensional space originated from the original space. First, each service $s \in S$ is mapped to a service $s' = (f_1(s), \ldots, f_d(s))$, where $f_i(s) = |q_i(s_r) - q_i(s)|$, $1 \leq i \leq d$. Then, the dynamic skyline of S with respect to functions $f_1, \ldots, f_d$, is obtained by calculating the ordinary skyline in the transformed d-dimensional space with s_r as the origin. Accordingly, dynamic dominance is defined as:

Definition 3 (Dynamic Dominance). Given two services, $s_i, s_j \in S$, characterized by d-dimensional quality values, and a reference service s_r, s_i dynamically dominates s_j with respect to s_r, denoted by $s_i \succ s_j$, if $\forall\, p \in [1, d]$: $|q_p(s_r) - q_p(s_i)| \leq |q_p(s_r) - q_p(s_j)|$ and $\exists\, p \in [1, d]$: $|q_p(s_r) - q_p(s_i)| < |q_p(s_r) - q_p(s_j)|$.

Based on Definition 3, we formally define the concept of dynamic skyline services:

Definition 4 (Dynamic Skyline Service). The dynamic skyline of S, denoted by S_{DSL}, consists the services that are not dynamically dominated by any other services, with respect to a given reference service s_r, i.e., $S_{DSL} = \{s_i \in S \mid \neg\exists\ s_j\text{: } s_j \succ s_i\}$. The services in S_{DSL} are referred to as dynamic skyline services.

Fig. 1(a) illustrates the identification of S_{DSL}. First, the original space is transformed into a new one with s_r as the new origin and the absolute distances to s_r as the mapping functions. Then, s_1, s_2, s_3, s_4, s_5, s_6, s_7 and s_8 are mapped into the new space where they are denoted by s_1', s_2', s_3', s_4', s_5', s_6', s_7' and s_8'. Service s_9 is already in the first quadrant of the new space and thus doesn't need to be mapped. All the services in the new space are collectively referred to as S'. The identification of S_{DSL} is equivalent to the identification of the skyline services in the new space, denoted by S_{SL}'. In Fig. 1(a), there is $S_{SL}' = \{s_1', s_2', s_3', s_4', s_5', s_7', s_8'\}$ in the new space. Accordingly, we can determine that $S_{DSL} = \{s_1, s_2, s_3, s_4, s_5, s_7, s_8\}$.

B. Service Clustering

To further reduce the number of services in S_{DSL} without losing their diversity, DQCSR partitions S_{DSL}, which is an NP-hard problem [10]. Thus, DQCSR employs the popular k-means algorithm for its popularity and high performance [11]. In addition, this allows customers to specify the number of types of representative services to be recommended. For example, a customer who is concerned about the latency and failure rate can choose $k = 3$. Given $k = 3$, k-means is expected to generally partition the services in S_{DSL} into three clusters, one cluster for low latency and high failure rate, one cluster for high latency and low failure rate and one cluster for balanced latency and failure rate.

The k-means algorithm partitions S_{DSL} into a total of k clusters. Then, exactly one service is selected from each cluster to generate R with final k services. The k-means is designed to maximize the similarity within the same cluster and the differences between different clusters. Thus, with k-means, DQCSR ensures that 1) services within the same cluster are similar in their quality representativeness; and 2) services in different clusters are different in their quality representativeness. Take Fig. 1(c) for example. Services s_1' and s_2' are both representatives with latency very similar to s_r.

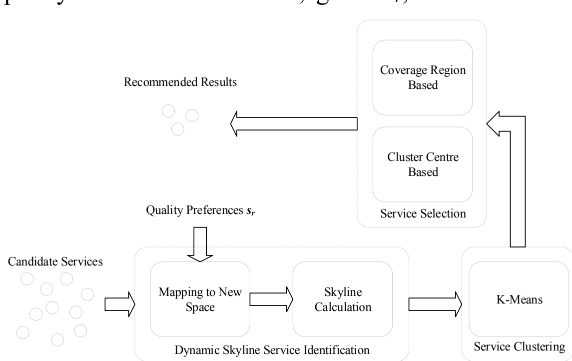


Figure 2. The Procedure of DQCSR.

Services $s_5^{'}$ and $s_8^{'}$ are both representative in failure rate. Services $s_3^{'}$, $s_4^{'}$ and $s_7^{'}$ are trade-off services that are most similar to s_r in general.

To eliminate the bias across quality dimensions with different units of measurement, DQSCR normalizes the quality of $s_i \in S$, $1 \leq i \leq n$, as well as s_r, with the min-max normalization technique, which has been widely employed in research on service-oriented computing [3], [12], [13].

C. Service Selection

Given k clusters returned by k-means, DQCSR employs two methods to select one service from each cluster to generate the recommendation results R. The first method is cluster center (CC) based and the second is coverage region (CR) based. Their differences are experimentally evaluated in Section IV.

Cluster Center based Selection. Algorithm 1 presents the pseudo-code of this method. Given a cluster of services denoted by C, the CC-based method calculates the center of C with formula (1) (line 3), denoted by c_C. Then, it calculates the Euclidean distance between each service s_i in C and c_C with formula (2) (lines 5-7). The service with the shortest distance is returned to be included in R (line 8). The computational complexity of Algorithm 1 is $O(|C|) = O(n)$.

$$q_p(c_C) = \frac{1}{|C|}\sum_{i=1}^{|C|} q_p(s_i) \quad p \in [1, d] \tag{1}$$

$$d(s_i, s_j) = \sqrt{\sum_{p=1}^{d} q_p(s_i) - q_p(s_j)} \tag{2}$$

where $q_p(s_i)$ is the p^{th} dimensional quality of s.

ALGORITHM 1: CC-based Service Selection

Input: C - service cluster
Output: s_C – service selected from C

```
1   Begin
2   S_rep ← null;
3   calculate the cluster center c_C
5   for each s in C do
6   |   s.distance ← distance(s, c_C);
7   end for
8   s_C ← min(s.distance) for all s in C
9   return s_C
10  End
```

ALGORITHM 2: CR-based Service Selection

Input: C - service cluster
Output: s - service selected from C

```
1   Begin
2   s ← null
3   for each s_i in C do
4   |   for each s_j (i ≠ j) in C do
5   |   |   r_i,j ← distance(s_i, s_j);
6   |   end for
7   |   r_i ← max(r_i,j) for all s_j in C (i ≠ j)
9   end for
10  s ← s_i with min(r_i) for all s_i in C
11  return s
12  End
```

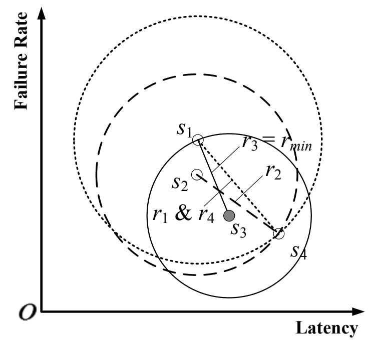


Figure 3. Identification of coverage region.

Coverage Region based Selection. First, we formally define coverage region:

Definition 5 (Coverage Region). The Given a cluster C of n services in a d-dimensional space, $C = \{s_1, s_2..., s_n\}$, for $s_i \in S$, $1 \leqslant i \leqslant n$, find $r_j = min(distance(s_i, s_j))$, $1 \leq j \leq n$ and $i \neq j$, the coverage region of s_i, denoted by $cr(s_i)$, is the region formed with s_i as the center and r_j as the radius.

Here, function $distance(s_i, s_j)$ calculates the Euclidean distance between s_i and s_j. Thus, in a two-dimensional space, a coverage region is a circle. In a three-dimensional space, it is a sphere, etc.

Algorithm 2 presents the pseudo-code of the CR-based selection method. Given a cluster C with n services, $C = \{s_1, \ldots, s_n\}$, It first calculates $r_1, \ldots, r_n$ (lines 3-9), then returns the center of the coverage region with $min(r_1, ..., r_n)$ (line 10). The computational complexity of Algorithm 2 is $O(|C|^2) = O(n^2)$.

Fig. 3 illustrates the identification of the coverage regions of four services in a cluster $C = \{s_1, s_2, s_3, s_4\}$ in a two-dimensional space. It is calculated that $r_1 = r_4 = distance(s_1, s_4)$, $r_2 = distance(s_2, s_4)$ and $r_3 = distance(s_1, s_3)$. There is $r_3 < r_2 < r_1 = r_4$. Thus, s_3 is returned. Given k clusters, DQCSR selects a total of k services to generate R.

IV. EXPERIMENTAL EVALUATION

This section experimentally evaluates DQCSR through comparison with four existing representative approaches in their recommendation accuracies and efficiency.

A. Experiment Setup

Dataset. The experiments were conducted on a widely used public real-world dataset named QWS [14]. This dataset contains 8-dimensional quality information on 2,507 real-world web services, including latency, availability, etc.

Comparing Approaches. We have implemented DQCSR-CC and DQCSR-CR, the former with the CC-based service selection method and the with the CR-based service selection method, both discussed in Section III-C. For comparison, we have implemented four existing representative approaches:

- **RS:** This baseline approach randomly selects k services from S.
- **UF:** This approach selects k services with the highest utility values, calculated with the widely used utility function [3], [4], [15], [16].

- **KNN:** This approach selects k services in S that are most similar to s_r.
- **DSL-KNN:** This approach models service recommendation as a k nearest neighbors problem [7]. It selects k services from S_{DSL}, i.e., the set of dynamic skyline services, which are most similar to s_r.

Evaluation Metrics. Given a set of candidate services S, we compare the recommendation accuracies achieved by the comparing approaches, denoted by R, measured by two metrics, diversity and representativeness.

Given two services s_i and s_j, the difference between them in a d-dimensional space can be measured by the Euclidean distance between them. A long distance indicates a significant difference. Thus, the diversity in R, i.e., how much the services in R differ from each other, can be measured by their average distance:

$$Diversity(R) = \frac{\sum_{s_i, s_j \in R, i \neq j} distance(s_i, s_j)}{|R| * (|R| - 1) / 2} \quad (3)$$

High diversity indicates high recommendation accuracy.

It is sometimes very challenging to ensure both the diversity and representativeness in R at the same time [7]. Thus, we also evaluate and compare the representativeness in R achieved by different approaches:

$$Representativeness(R) = |R_{DSL}|/|R| \quad (4)$$

where $|R_{DSL}|$ is the number of dynamic services in R. The services in R_{DSL} are representative because they are not dominated by any other services in R. Thus, high representativeness indicates high recommendation accuracy.

We also evaluate the efficiency of all approaches measured by computation time.

To simulate different recommendation scenarios, we have conducted three series of experiments. In each series, we vary one of the three experiment parameters. Table II presents the corresponding parameter settings. Each experiment is run for 100 times and the results are averaged. In each run, we randomly select n services from the QWS dataset as candidate services, and another one as s_r. Then, we run the comparing approaches to make recommendations. All experiments are implemented in Python 2.7 and conducted on a machine with Intel i7-4790 CPU 3.60GHz and 16 GB RAM, running Windows 10 x64 Professional.

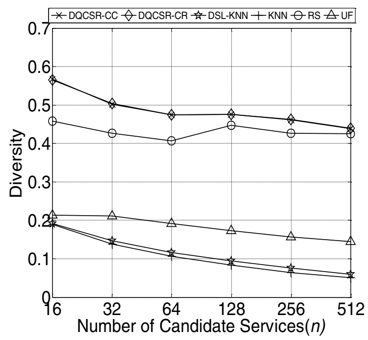
B. Experimental Results

Fig. 4 shows and compares the impact of the number of candidate services in S, denoted by n, on the recommendation accuracies achieved by the approaches. Fig. 4(a) shows that DQCSR-CR and DQCSR-CC achieve the highest diversity in their recommendation results, 0.487 and 0.486 on average. RS achieves the third highest diversity because it selects services randomly to recommend without considering the representativeness in R. The diversity achieved by the other four approaches are much worse than DQCSR-CR, DQCSR-CC and RS, as demonstrated by their very low average diversity, 0.181for UF, 0.114 for DSL-KNN and 0.106 for KNN. As n increases, the diversity achieved by all approaches decrease. The reason is that the increase in n will drive those approaches to eventually select services with similar quality characteristics. Take DQCSR-CR and DQCSR-CC for example. The increase in n results in an increase in $|S_{DSL}|$, i.e., the number of dynamic skyline services, because it becomes harder for the services in S to dominate each other. When there are more services in S_{DSL}, the services finally selected by DQCSR-CR and DQCSR-CC are closer to the centers of their corresponding clusters. Accordingly, the diversity achieved by DQCSR-CR and DQCSR-CC tends to converge to the average distance between the centers of the k clusters. Fig. 4(b) shows that DQCSR-CR and DQCSR-CC consistently obtain a representativeness value of 1.0, the same as DSL-KNN. This indicates the importance of the DSL operator which considers s_r when selecting representative services. When n increases, the k services in R achieved by KNN, RS or UF are more likely to be dynamically dominated by at least one of the other n - k services. As a result, the representativeness they achieve decreases as n increases.

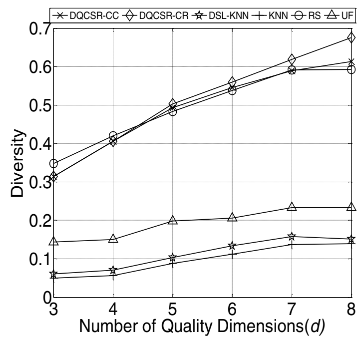
TABLEII. PARAMETER SETTINGS

Parameter	Experiment Series		
	A	B	C
Number of Candidate Services (n)	16 to 512	512	512
Number of Services to Recommend (k)	4	3 to 8	4
Number of Quality Dimensions (d)	4	4	3 to 8

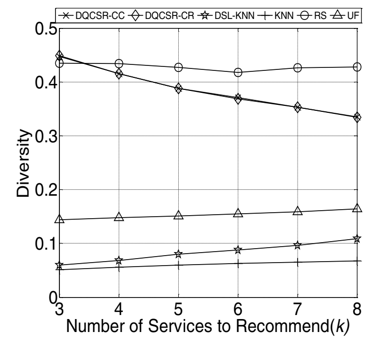
Fig. 5 shows the impact of the number of quality dimensions, denoted by d. Fig. 5(a) demonstrates that DQCSR-CR, DQCSR-CC and RS outperform the other approaches in ensuring the diversity in the recommendation results. The average diversity values achieved by DQCSR-CR, DQCSR-CC and RS are 0.513, 0.493 and 0.496, versus 0.194, 0.113 and 0.097 achieved by UF, DSL-KNN and KNN respectively. In general, the increase in d leads to increase in the diversity achieved by all comparing approaches, specifically from 0.314 to 0.675 for DQCSR-CR and 0.314 to 0.613 for DQCSR-CC. The increase in d differentiates the candidate services in different quality dimensions. Accordingly, the k services in R are more distributed in the d-dimensional space, resulting in a higher average distance between them. Fig. 5(b) shows that DQCSR-CR and DQCSR-CC achieve a representativeness value of 1.0 across all experiments in this series. This shows that DQCSR-CR and DQCSR-CC do not compromise the representativeness in their recommendation results when handling services with high dimensionality. When d increases, there are more dynamic skyline services because it is harder for services in S to dynamically dominate each other. Thus, it is more likely that some of the k services in the recommendation results obtained by the other four approaches happen to be dynamic services, which increase the representativeness values achieved by those approaches. However, their representativeness values are still significantly lower than those achieved by DQCSR-CR, DQCSR-CC.



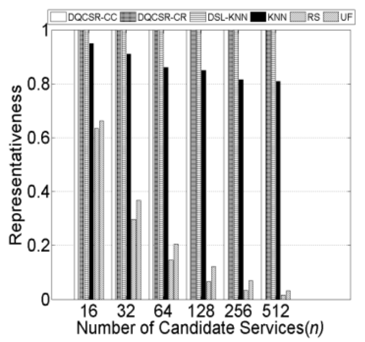
(a) Diversity vs. *n*



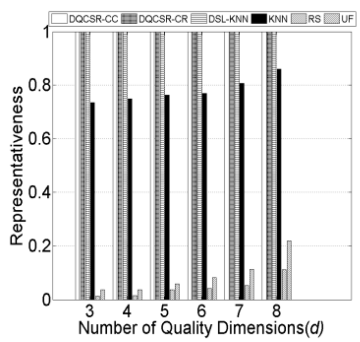
(a) Diversity vs. *d*



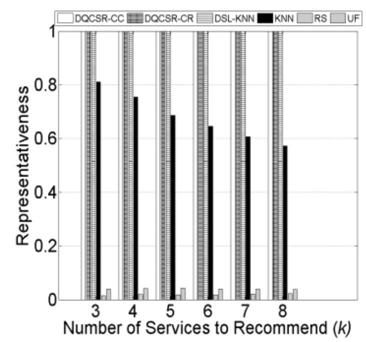
(a) Diversity vs. *k*



(b) Representativeness vs. *n*



(b) Representativeness vs. *d*



(b) Representativeness vs. *k*

Figure 4. Impact of *n* on recommendation accuracy. Figure 5. Impact of *d* on recommendation accuracy. Figure 6. Impact of *k* on recommendation accuracy.

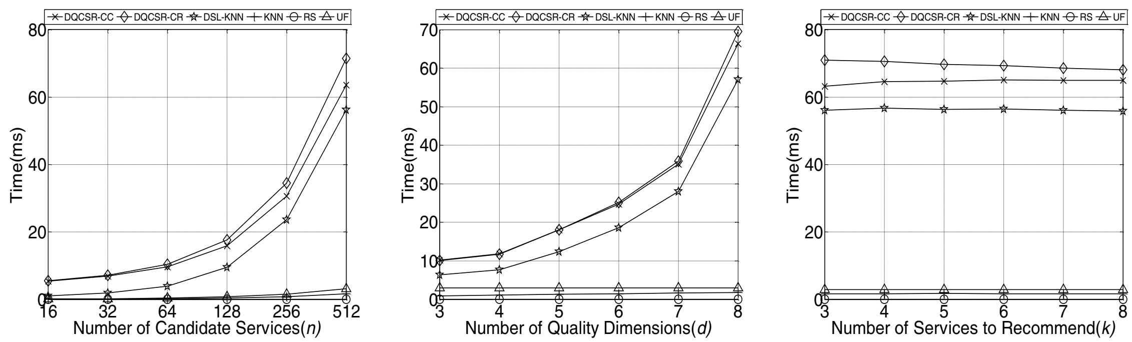
Fig. 6 shows the impact of the number of services to recommend, denoted by k. Fig. 6(a) shows that, in this series of experiments, DQCSR-CR and DQCSR-CC still outperform UF, DSL-KNN and KNN in ensuring diversity. The diversity achieved by DQCSR-CR and DQCSR-CC decrease as k increases. This is inevitable. Let us consider the cases with $k = 3$ and $k = 4$. Given S_{DSL}, suppose DQCSR-CR and DQCSR-CC have selected 3 services, i.e., s_1, s_2 and s_3, that have the minimum average distance between them. Now they need to select 1 more service to recommend, say s_4. The average distance between s_4 and s_1, s_2 and s_3 is larger than that between s_1, s_2 and s_3. Otherwise, s_4 would have been selected in the first place instead of one of s_1, s_2 and s_3. Thus, the inclusion of s_4 in R will decrease the average distance between the services in R and consequently decrease the diversity in R. In this series of experiments, RS achieves the highest diversity overall. Its search space is the entire S while the CR and CC operators of DQCSR-CR and DQCSR-CC, which are responsible for ensuring the diversity in R, can only select services from S_{DSL}. According to Definition 4, there is $|S_{DSL}| \leq |S|$. Thus, RS is more likely to be able to diversify R. Consistently with Fig. 4(b) and Fig. 5(b), Fig. 6(b) shows the ability of DQCSR-CR and DQCSR-CC to ensure the representativeness in R.

Figs. 4(a) and Fig. 6(a) show that DQCSR-CR and DQCSR-CC achieve the same diversity. In those experiments, given the same cluster, CR and CC operators select the same services to be included in R. Fig. 5(a) shows that as d increases, DQCSR-CR starts to outperform DQCSR-CC. This indicates that DQCSR-CR is more capable of diversifying R in scenarios with high quality dimensionality.

Fig. 7 shows the impacts of n, k and d on the computation time taken by different approaches to make recommendations. It shows that DQCSR-CR, DQCSR-CC and DSL-KNN always take more time than KNN, RS and UF, especially when n and d increase. This is due to the complexity of the dynamical skyline calculation, which is $O(dn^2)$. However, even in the largest-scale experiments, DQCSR-CR and DQCSR-CC can still make recommendations within 80 milliseconds, which are acceptable in, if not all, most cases.

V. THREATS TO VALIDITY

Threats to construct validity. One of the main threats to the construct validity of our evaluation lies in the comparison of recommendation accuracy with the existing recommendation methods, including a random approach, a utility-based approach and two KNN-based approaches. The random approach serves as the baseline for the comparison. The utility-based approach has been widely employed to evaluate services and select suitable service [1], [3], [4], [15], [16]. It does not consider customers' quality preferences and thus tends to obtain recommendation results less representative than our approach. Thus, the main threat to construct validity is whether the comparison with the



(a) Time vs. n (b) Time vs. d (c) Time vs. k

Figure 7. Impacts of factors on computation time.

selected existing approaches can properly demonstrate the effectiveness of our approach. To minimize this threat, we have also compared the effectiveness of our approach with the KNN-based approach as well as DSL-KNN. In this way, we can demonstrate the advantages of our approach in a straightforward and objective manner.

Threats to external validity. The main threat to the external validity of our evaluation is the representativeness of the dataset used in the evaluation, which might not be able to exactly represent all real-world applications. This threat is however minimal. First, the QWS dataset contains real information about real-world services [14] and has been employed in a broad range of research [3], [4], [17]. Second, the major information in a dataset that impacts our evaluation is the distributions of the quality values of the services, not their absolute quality values. In this regard, we varied the number of candidate services and the number of quality constraints being considered to simulate various applications of different characteristics. In this way, we can evaluate the proposed approach in different scenarios.

Threats to internal validity. The main threat to the internal validity of our evaluation is the comprehensiveness of the experiments. Due to the page limit, we have not been able to present the results of experiments under all parameter settings, e.g., more combinations of different parameter values. We believe that this threat is not significant. The diversity and representativeness in the recommendation results obtained from the experiments under other parameter settings might be different. However, the advantages of our approaches over the comparing methods are similar to those presented and discussed in Section IV.

Threats to conclusion validity. The main threat to the conclusion validity of our evaluation is the lack of statistical tests, e.g., chi-square tests. We could have conducted chi-square tests to draw conclusions in the evaluation. However, we ran the experiment for 100 times in each set and averaged the results each time that we varied the parameter setting. This led to a large number of test cases, which tend to result in a small p-value in the chi-square tests and lower the practical significance of the test results [18]. However, this number of runs is not even close to the number of observation samples that concern Lin et al. in [18]. Thus, the threat to the conclusion validity due to the lack of statistical tests might be high but not significant.

VI. RELATED WORK

Quality-aware service recommendation is a critical issue in service-oriented computing. Utility-based recommendation [3], [4], [15], [16] and skyline-based [1] recommendation are currently the two most popular approaches.

Utility-based recommendation has been widely employed in the past decade for its simplicity. The utility value of a service indicates how good its overall d-dimensional quality is in comparison with the other candidate services in S. Given S, it simply selects the services with the highest utility values. Skyline-based service recommendation was first employed by Alrifai et al. to select representative services [1]. In recent years, many researchers have tried to improve skyline-based service recommendation to accommodate more sophisticated recommendation scenarios [19]. The common and critical limitation of utility-based and skyline-based service recommendation is the lack of consideration for customers' quality preferences, which have always been a fundamental and critical issue in quality-aware service selection [3], [4], [16], [17] and skyline-based service composition [20], [21]. This renders the utility-based and skyline-based service recommendation obsolete.

The authors of [7] attempt to recommend representative services with respect to customers' quality preferences. However, their approach lacks the ability to diversify the recommendation results because it fails to accommodate customers' quality correlations, which has attracted many researchers' attention in recent years. To name a few, He et al. [4] built a service selection approach named CASS. It allows price discounts or quality premiums to be flexibly applied to service bundles to capture quality correlations. Deng et al. [22], [23] propose a service selection approach that prunes services based on quality correlations. Zhang et al. [24] propose an alliance-aware service composition approach that considers the quality correlations. Then, they proposed an index graph based approach that allows service composition approaches to efficiently query for optimal quality correlations. Du et al. [25] propose an approach for calculating the composite service skyline with quality correlations.

Our approach, named DQCSR, addresses the above issues and advances quality-centric service recommendation significantly. Through a procedure that consists of dynamic skyline service identification, service clustering and service selection, it ensures both the representativeness and diversity in the recommendation results with respect to customers' quality preferences.

VII. Conclusion and Future Work

In this paper, we proposed a quality centric approach named DQCSR for service recommendation with the objective to ensure both the diversity and representativeness in the recommendation results. DQCSR employs the dynamic skyline technique and k-means to achieve this objective. The results of extensive experiments conducted on a real-world dataset demonstrate that DQCSR significantly outperforms all existing representative approaches in diversifying the recommendation results without compromising representativeness.

In the future, we will attempt to combine the proposed approach with fuzzy logic to better accommodate the uncertainty in customers' quality preferences.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No.61872002), Australian Research Council Discovery Projects (No. DP170101932, DP180100212) and the Natural Science Foundation of Anhui Province of China (No.1808085MF197). Qiang He is the corresponding author of this paper.

References

[1] M. Alrifai, D. Skoutas, and T. Risse, "Selecting Skyline Services for QoS-Based Web Service Composition," in Proceedings of International Conference on World Wide Web (WWW), 2010, pp. 11-20.

[2] K. Benouaret, D. Benslimane, A. Hadjali, M. Barhamgi, Z. MAAMAR, and Q. Sheng, "Web Service Compositions with Fuzzy Preferences: A Graded Dominance Relationship Based Approach," ACM Transactions on Internet Technology, Vol. 13, 2014, pp. 1-33.

[3] I. Trummer, B. Faltings, and W. Binder, "Multi-Objective Quality-Driven Service Selection—A Fully Polynomial Time Approximation Scheme," IEEE Transactions on Software Engineering, Vol. 40, 2014, pp. 167-191.

[4] Q. He, J. Yan, H. Jin, and Y. Yang, "Quality-Aware Service Selection for Service-Based Systems Based on Iterative Multi-Attribute Combinatorial Auction," IEEE Transactions on Software Engineering, Vol. 40, 2014, pp. 192-215.

[5] Z. Zheng and M. R. Lyu, "Personalized Reliability Prediction of Web Services," ACM Transactions on Software Engineering and Methodology, Vol. 22, 2013, pp. 1-28.

[6] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative Web Service QoS Prediction via Neighborhood Integrated Matrix Factorization," IEEE Transactions on Services Computing, Vol. 6, 2013, pp. 289-299.

[7] Y. Zhang, X. Ai, Q. He, X. Zhang, W. Dou, F. Chen, L. Chen, and Y. Yang, "Personalized Quality Centric Service Recommendation," in Proceedings of International Conference on Service-Oriented Computing (ICSOC), 2017, pp. 528-544.

[8] Y. Wang, Q. He, D. Ye, and Y. Yang, "Service Selection Based on Correlated QoS Requirements," in Proceedings of IEEE International Conference on Services Computing (SCC), 2017, pp. 241-248.

[9] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, and Q. He, "Efficient Query of Quality Correlation for Service Composition," IEEE Transactions on Services Computing, 2018, DOI:10.1109/TSC.2018.2830773.

[10] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-Hardness of Euclidean Sum-of-Squares Clustering," Machine Learning, Vol. 75, 2009, pp. 245-248.

[11] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An Efficient K-means Clustering Algorithm: Analysis and Implementation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, 2002, pp. 881-892.

[12] Q. He, J. Han, Y. Yang, J. Grundy, and H. Jin, "QoS-Driven Service Selection for Multi-Tenant SaaS," IEEE Transactions on Software Engineering, Vol. 40, 2014, pp. 167-191.

[13] Q. He, J. Han, F. Chen, Y. Wang, R. Vasa, Y. Yang, and H. Jin, "Qos-Aware Service Selection for Customisable Multi-Tenant Service-Based Systems: Maturity and Approaches," in Proceedings of IEEE International Conference on Cloud Computing (CLOUD), 2015, pp. 237-244.

[14] E. Al-Masri and Q. H. Mahmoud, "Investigating Web Services on the World Wide Web," in Proceedings of International Conference on World Wide Web (WWW), 2008, pp. 795-804.

[15] M. Alrifa and T. Risse, "Combining Global Optimization with Local Selection for Efficient QoS-Aware Service Composition," in Proceedings of International Conference on World Wide Web (WWW), 2009, pp. 881-890.

[16] D. Ardagna and B. Pernici, "Adaptive Service Composition in Flexible Processes," IEEE Transactions on Software Engineering, Vol. 33, 2007, pp. 369-384.

[17] Q. He, R. Zhou, X. Zhang, Y. Wang, D. Ye, F. Chen, J. Grundy, and Y. Yang, "Keyword Search for Building Service-Based Systems," IEEE Transactions on Software Engineering, Vol. 43, 2017, pp. 658-674.

[18] M. Lin, H. C. Lucas Jr, and G. Shmueli, "Too Big to Fail: Large Samples and the *p*-Value Problem," Information Systems Research, Vol. 24, 2013, pp. 906-917.

[19] K. Benouaret, D. Benslimane, and A. Hadjali, "On the Use of Fuzzy Dominance for Computing Service Skyline Based on QoS," in Proceedings of IEEE International Conference on Web Services (ICWS), 2011, pp. 540-547.

[20] X. Zhao, L. W. Shen, X. Peng, and W. Zhao, "Finding Preferred Skyline Solutions for SLA-Constrained Service Composition," in Proceedings of IEEE International Conference on Web Services (ICWS), 2013, pp. 195-202.

[21] H. F. Zhang, K. Hwang, S. U. Khan, and Q. M. Malluhi, "Skyline Discovery and Composition of Multi-Cloud Mashup Services," IEEE Transactions on Services Computing, Vol. 9, 2016, pp. 72-83.

[22] S. Deng, H. Wu, and JL. Zhao, "Service Selection for Composition with QoS Correlations," IEEE Transactions on Services Computing, Vol. 9, 2016, pp. 291-303.

[23] S. Deng, L. Huang, D. Hu, J. Zhao and Z. Wu, "Mobility-Enabled Service Selection for Composite Services," IEEE Transactions on Services Computing, Vol. 9, 2016, pp. 394-407.

[24] Y. Zhang, G. Cui, S. Deng, and Q. He, "Alliance-Aware Service Composition Based on Quotient Space," in Proceedings of IEEE International Conference on Web Services (ICWS), 2016, pp. 340-347.

[25] Y. Du, H. Hu, W. Song, J. Ding, and J. Lü, "Efficient Computing Composite Service Skyline with QoS Correlations," in Proceedings of IEEE International Conference on Services Computing (SCC), 2015, pp. 41-48.