Variation-Aware Cloud Service Selection via Collaborative QoS Prediction

Hua Ma, Zhigang Hu, Keqin Li, *Fellow, IEEE*, and Haibin Zhu, *Senior Member, IEEE*

Abstract—As the number of cloud services (CSs) offering similar functionality is growing, more attention has been payed on the quality of service (QoS) of CSs. However, in a dynamic cloud environment, the explicit and inherent variation of QoS causes the single CS selection via collaborative filtering techniques (CSS-CFT) to be challenging. A variation-aware approach via collaborative QoS prediction is proposed to select an optimal CS according to users' non-functional requirements. Based on time series QoS data, this approach utilizes a set of specific cloud models to quantify the variation characteristics of QoS from the four aspects including central tendency, variation range, frequency of variation and period. To exactly identify the neighboring users for a current user, this paper employs the double Mahalanobis distances to measure the similarity of QoS cloud models. The variation-aware CSS-CFT is formulated as a multi-criteria decision-making problem, and an improved TOPSIS method is exploited to solve it, by considering both the objective QoS variation and subjective user preferences during different time periods. The experiments based on a real-world dataset demonstrate that the proposed approach can enhance the accuracy of CSS-CFT in a high-variance environment without noticeable increase of selection time, in comparison to the existing approaches.

Index Terms—Cloud model, cloud service selection, collaborative filtering, mahalanobis distance, QoS prediction, QoS variation, TOPSIS method, user preferences

1 Introduction

1.1 Motivation

Recently, cloud services (CSs) rapidly proliferate around the world [1]. The exploitation of CSs is progressively appealing due to the reduction on usage costs and the elasticity of computing power [2]. As the number of CSs offering similar functionality is growing [3], increasing attention has been payed on the quality of service (QoS) of CSs. For a CS, QoS is the description or measurement of its overall performance, particularly the performance seen by the users. To quantitatively measure QoS, several related parameters of a CS are often considered, such as response time, throughput, availability. Thus, QoS becomes an important differentiator among functionally equivalent CSs, describing how well a CS is performed [3]. From the different perspectives, researchers have proposed some approaches for the CS selection via collaborative filtering techniques (CSS-CFT) over the years to help a current user select an optimal CS from functionally equivalent ones [2]. The traditional CSS-CFT approaches function by the following steps [4]: 1) exploit the history QoS data about known CSs experienced by users to measure the similarity between a current user and other users; 2) identify the neighboring users for a current user; 3) predict the unknown CSs' QoS values for a current user by using the history data from the neighboring users; 4) on the basis of these predicted QoS values about candidates, select a most suitable CS with the optimal QoS for a current user meeting user preferences by evaluating their performance. However, in a dynamic cloud environment, the explicit and inherent variation of QoS causes the CSS-CFT to be challenging.

Challenge 1: *The explicit variation of QoS experienced by users make it complex to exactly identify the neighboring users for a current user when the context of users is unavailable*. The quality of experience (QoE) [1] of users is often different from the QoS claimed by the service providers [3]. In fact, the fluctuant performance of CSs usually leads to a wide variation range of QoE. According to the evaluation report,[1] the measured maximum values of some QoS parameters (e.g., throughput) may be ten times more than the minimum values. The wide variation range of QoE signifies the high uncertainty of CSs' QoS. Besides, the real QoE is affected by many factors [5], [6], [7], such as client devices, geographic or network locations and usage time period. These factors may result in a totally different QoS of the same CS experienced by different users. When these factors related to users are unknown, it is hard to precisely measure the similarity between a current user and other users based on uncertain QoS data. The CSS-CFT approaches rely on the history QoS data about known CSs experienced by users to calculate the user similarity. However, the high uncertainty of CSs' QoS will decrease the precision of neighboring user identification, which eventually is

- *H. Ma is with the College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China. E-mail: huama@hunnu.edu.cn.*
- *Z. Hu is with the School of Software, Central South University, Changsha 410075, China. E-mail: zghu@csu.edu.cn.*
- *K. Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA. E-mail: lik@newpaltz.edu.*
- *H. Zhu is with the Collaborative Systems Laboratory, Nipissing University, North Bay, ON P1B8L7, Canada. E-mail: haibinz@nipissingu.ca.*

Manuscript received 23 July 2017; revised 12 Jan. 2019; accepted 16 Jan. 2019. Date of publication 27 Jan. 2019; date of current version 9 Dec. 2021. (Corresponding author: H. Ma.)
Digital Object Identifier no. 10.1109/TSC.2019.2895784

1. http://www.yunzhiliang.net/cloudtest/cloudtest.html

bound to reduce the accuracy of CSS-CFT in a dynamic cloud environment.

Challenge 2: The inherent variation of QoS make it complex to select a suitable CS in accordance to the period preferences of the current user. Existing research [8], [9], [10], [11], [12], [13] revealed that the QoS of a CS usually changes over time due to fluctuation of loads, unstable network or resource sharing. CSs exhibit the best possible QoS at off-peak hours, and their performance deteriorates during peak hours. Furthermore, the users may have preferences for a CS's performance in different periods. For example, a stock exchange corporation in China that pays particular attention to certain busy periods for buying and selling stocks (e.g., the periods from 9:30 AM to 11:30 AM and from 1:00 PM to 3:00 PM), expects a cloud storage service to provide a high performance in concurrent reading and writing during these two periods. However, a CS with an intermediate level of concurrent reading capacity in other periods is also satisfactory. In contrast, for a logistics company, statistical data indicate that the peak time for querying express packages occurs between 12:00 PM and 2:00 PM, and between 6:00 PM and 8:00 PM. For this company, a CS is necessary with the superior concurrent reading performance during these two periods compared to the different busy periods of the previous case. Thus, to improve customer satisfaction, it is indispensable to account for the periodic variation of QoS and the user's requirements and preferences during different periods.

Recently, from the perspective of CS consumers, many solutions (e.g., MonSLAR [14], SLA-management [15]) are proposed to monitor the CSs' performance and collect the CSs' metric data by utilizing injected agents [16]. Especially, some researchers have carried out work over continuous monitoring of CSs. Rosaci et al. [5] demonstrated that agents deployed in a user client can easily capture QoS data of services. Zhang et al. [11] deployed a tool called WSMonitor on 142 computers located in 22 countries from PlanetLab project to collect the QoS data of services in 64 timeslots into WS-DREAM dataset #2. The above work makes it possible to thoroughly analyze the QoS variation of CSs based on time series data.

Considering that the variation of QoS is inevitable in a cloud environment, two complementary strategies are proposed to identify an excellent CS from candidates based on time series QoS data as follows: (1) the performance of an excellent CS should be stable, namely, the time series QoS data about it should have a good central tendency, a narrow variation range and a low frequency of variation; and (2) an excellent CS could meet the current user's requirements in different periods for the QoS performance. These two strategies are instructive to exactly select a CS with optimal QoS from candidates in light of the current user's period preferences. Furthermore, according to the above strategies, we can extract the variation characteristics of QoS from four aspects, including central tendency, variation range, frequency of variation and period, to evaluate CSs. By analyzing the four aspects of QoS variation, a more precise similarity measurement between a current user and other users can be computed.

Aiming at the variation of QoS in dynamic cloud environment, researchers proposed many approaches for solving CS selection from different perspectives, such as probabilistic models [17], [18], fuzzy models [19], [20], multi-objective optimization [13], prediction models [21], [22] and time-aware models [10], [23], [24], [25]. However, they fail to systematically model the four aspects of QoS variation hidden in time series data and harness them to improve the accuracy of CSS-CFT in accordance to user preferences during different periods. Thus, it is still an open question to study the variation-aware CSS-CFT.

The cloud model theory [26] has the advantages in discovering the latent variation features hidden in time series data and depicting the global and local variation features of time series data. In our previous work [8], we have introduced cloud model theory and presented a prediction approach for unknown QoS values by analyzing the time-varying characteristics of QoS. In this paper, we actually go beyond this to propose a variation-aware CS selection approach by exploiting cloud models to thoroughly analyze the four aspects of QoS variation in multiple periods and supporting the flexibly personalized settings of the length and number of periods.

1.2 Our Contributions

The main contributions of this paper are as follows:

1) To accommodate the uncertainty of CSs' QoS and the diversity of users' preferences in the dynamic cloud environment, we employ the cloud model theory to mathematically model the variation of QoS from four aspects including central tendency, variation range, frequency of variation and period, by utilizing a set of QoS cloud models to distinguish the comprehensive variation characteristics of QoS in multiple periods based on time series data. In contrast to the existing work, more aspects are considered for handling the QoS variation.
2) To exactly identify the neighboring users for a current user, the time series QoS data of every user about one CS are modeled as a set of QoS cloud models. A variation-aware method is presented to calculate the similarity between a current user and other users by employing the double Mahalanobis distances to measure the similarity of QoS cloud models during multiple periods. In contrast to the previous method [8], the experiments show that this method improves the accuracy of neighboring user identification and provides a strong support for predicting the QoS values based on collaborative filtering.
3) To select an optimal CS for a current user by considering both the objective QoS variation and the subjective user preferences in multiple periods, this paper formulates the variation-aware CS selection problem as a multi-criteria decision-making (MCDM) problem. An already proposed extension to TOPSIS (technique for order preference by similarity to an ideal solution) [27] method based on the Mahalanobis distance is successfully applied to solve this problem. The experiments based on a real-world dataset demonstrate that the proposed approach can enhance the accuracy of CSS-CFT in the high-variance cloud environment without noticeably increased selection time, in comparison to the existing approaches.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 gives the problem statement. Section 4 analyzes the variation of time series QoS

TABLE 1
Summary of CS Selection Approaches

Group	Title	Aspects of QoS variation	Metrics	Data representation	Selection method	CS type	Data source
Group #1: CS selection not concerning time series data	Mehdi et al. [17]	Central tendency, variation range	Response time, throughput, availability, and reliability	Statistical distribution	Machine learning techniques	Web services	QWS dataset
	Ma et al. [25]	Central tendency	Not limited to specific QoS parameters	Single-value number	CFT via linear regression	Web services	WS-DREAM dataset #1
	Ma et al. [19]	Central tendency, variation range	Not limited to specific QoS parameters	Interval number with 4 parameters	Possibility degree ranking	General	WS-DREAM dataset #1
	Sun et al. [20]	Central tendency, variation range	Not limited to specific QoS parameters	Fuzzy number	Fuzzy TOPSIS	General	Simulation
	Wang et al. [13]	Central tendency	QoS (not limited to specific parameters) and trust	Single-value number	Multi-objective optimization	General	QWS dataset
	Zheng et al. [21]	Central tendency	Not limited to specific QoS parameters	Single-value number	Greedy and QoS ranking prediction algorithms	General	WS-DREAM dataset #1
	Mao et al. [22]	Central tendency	Not limited to specific QoS parameters	Single-value number	Particle swarm optimization	Web services	WS-DREAM dataset #1
Group #2: CS Selection using time series data	Zhong et al. [28]	Central tendency, period	Popularity	Single-value number	Ranking via geometric mean calculation	Web services	ProgrammableWeb dataset
	Ye et al. [23]	Central tendency, period	Response time, throughput, and cost	Single-value number	Ranking via time series similarity	General	Simulation +real data
	Hu et al. [24]	Central tendency, period	Not limited to specific QoS parameters	Single-value number	Ranking via user-based and service-based prediction	Web services	WS-DREAM dataset #2
	Mehdi et al. [29]	Central tendency, period	Not limited to specific QoS parameters	Single-value number	Selection via power steady model	Web services	Simulation
	Ma et al. [10]	Variation range; period	Trustworthiness	Neutrosophic set	ELECTRE	General	WS-DREAM dataset #2
	Proposed approach in this paper	Central tendency; variation range; frequency of variation; period	Not limited to specific QoS parameters	A set of cloud models	Improved TOPSIS	General	WS-DREAM dataset #2

data. Section 5 presents the variation-aware neighboring user identification method. Section 6 proposes the variation-aware CS selection approach. Section 7 analyzes the experiments and results. Finally, the conclusions and further work are given in Section 8.

2 Related Work

2.1 Summary of CS Selection Approaches

We categorize the existing CS selection approaches into two groups and compare them from six dimensions with the proposed approach: aspects of QoS variation, metrics, data representation, selection method, CS type and data source. The results are shown in Table 1.

1) CS selection not concerning time series data. These approaches in Group #1 model the QoS variation based on the sample QoS data collected mainly sporadically, not the time series data, and select optimal CS for a current user using machine learning techniques [17], CFT via linear regression [25], possibility degree ranking [19], fuzzy TOPSIS [20], or optimization algorithms [13], [21], [22]. In contrast to the approaches in Group #2, they usually have good execution performance due to the lower computation complexity.

These approaches in Group #1 mainly employ single-value numbers, statistical distributions, interval numbers, or fuzzy numbers to represent the QoS values of CSs. The data representation methods based on single-value numbers [13], [21], [22], [25] use an integer or a real number to describe the CSs' performance and only depict the central tendency characteristic of QoS variation. The data representation method based on statistical distributions [17] using employ multinomial Dirichlet, generalized Dirichlet, or Beta-Liouville can capture both the central tendency and variation range characteristics of QoS variation. Ma et al. [19] proposed a data representation method based on interval numbers with four parameters (INF) and employed INF to depict the central tendency and variation range of QoS. Although Sun et al. [20] proposed a data representation method based on fuzzy number to convert the CSs' evaluations, these fuzzy numbers only capture the central tendency and variation range of QoS variation.

In general, these data representation methods used by these approaches in Group #1 can only capture the partial variation characteristics of CS's QoS, and are incapable to support time-aware personalized CS selection for a current user according to his/her period preferences.

2) CS selection using time series data. The approaches in Group #2 utilize time series QoS data to model the QoS variation, and select optimal CS for a current user using ranking via geometric mean calculation [28], ranking via time series similarity [23], ranking via user-based and service-based prediction [24], prediction via power steady model [29], or ELECTRE [10]. In contrast to the approaches in Group #1, they process larger volumes of data, usually with more execution time.

Although most of approaches in Group #2 employ single-value number to represent the QoS values, they could get more details of QoS variation than those in Group #1 by extracting the period or variation range characteristic from the time series data. Especially, the period characteristic facilitates to make accurate decisions of time-aware personalized CS selection for a current user according to his/her preferences.

However, there are still some limitations in these approaches, including a) only focusing on the separate moment that a user sends request [28], without consideration of the variation of CSs' QoS over time, b) only analyzing the correlations among QoS attributes to select optimal CS composition [23], not the comprehensive variation characteristics of every QoS parameter, c) lacking of consideration of the variation range characteristic of QoS variation, [23], [24], [28], [29], d) without consideration of the frequency of variation characteristic in all approaches. In sum, these approaches in Group #2 still fail to provide a systematic analysis to integrate all four aspects of QoS variation, and their accuracy is limited in variation-aware personalized CS selection.

In addition, Table 1 also shows that: a) most of CS selection approaches choose the WS-DREAM datasets, consisting

TABLE 2
Summary of Cloud Model Similarity Measurement

Methods	Computation complexity	Possibility of error	Calculation precision	Any parameter determined by empirical method
SCM [32]	High	High	Low	No
LICM [33]	Low	Medium	Medium	No
ECM and MCM [34]	Medium	High	Low	No
EDCM [35]	Low	Medium	Medium	No
VCM [8]	Low	Low	High	Yes
Proposed method	Low	Low	High	No

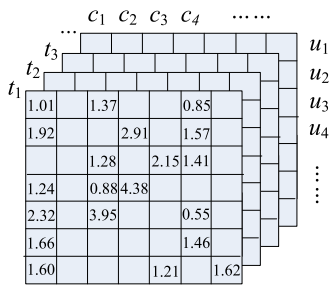

Fig. 1. An example of user-service-time matrix for response time.

of dataset #1 and dataset #2, as the data source in experiments. b) most of them are not limited to specific QoS parameters. c) most of approaches take the general CSs, not the specific SaaS/IaaS/PaaS CSs, as their research objects.

Aiming at the deficiency of the existing approaches, we systematically analyze all four aspects of QoS variation (i.e., central tendency, variation range, frequency of variation and period) and harness them to improve the similarity measurement between users and personalized CS selection.

2.2 Summary of Cloud Model Similarity Measurement

Gaussian distributions are found widely in nature and society. The Gaussian distribution functions with the parameters of expectation (*Ex*) and standard variance (*En*) are often used as the membership functions in fuzzy sets. However, Li et al. [30] found that a concept might have different meanings for different people, such that the membership degree is difficult to be identified precisely. Therefore, they introduced the hyper entropy (*HE*) as the standard variance of *En* and proposed the cloud model theory [30]. Cloud model theory is an effective tool in transforming between the qualitative concepts and their quantitative expressions, and can represent the fuzziness, the randomness and the relationships of uncertain concepts [26], [31]. Especially, the cloud model theory could provide the strong support for analyzing the latent features hidden in time series data, and clearly depict the global and local features of time series data [8], [9], [10]. Thus, with the cloud model theory, an effective mechanism to analyze the time-varying QoS characteristics of CSs could be established.

Currently some methods have been proposed to compute the similarity between two cloud models. We compare them from four dimensions with the proposed method: computation complexity, calculation precision, possibility of error and any parameter determined by empirical method. The results are summarized in Table 2.

There are some limitations with the first five methods [8] from Table 2, such as the time-consuming computation (e.g., SCM), obvious calculation errors (e.g., SCM, ECM and MCM), and unsatisfactory calculation precision (e.g., SCM, ECM and MCM). To overcome the limitations of the above methods, Ma et al. [8] presented a vector comparison method called VCM. In VCM method, a cloud model is viewed as a vector $\overrightarrow{E} = (Ex, En, He)$. First, the orientation similarity $O(\overrightarrow{E_i}, \overrightarrow{E_j})$ and the dimension similarity $D(\overrightarrow{E_i}, \overrightarrow{E_j})$ between two cloud model vectors are calculated from the perspectives of angle characteristics and numerical characteristics of vectors, respectively. Then, the synthetical similarity between two cloud models is computed by using a regulatory factor (α) to aggregate them as: $s(cm_i, cm_j) = \alpha \times O(\overrightarrow{E_i}, \overrightarrow{E_j}) + (1-\alpha) \times D(\overrightarrow{E_i}, \overrightarrow{E_j})$, where $\boldsymbol{\alpha}$ determines the weights of the orientation similarity and dimension similarity. However, the parameter $\boldsymbol{\alpha}$ is set manually based on an empirical method, and its appropriate value meeting the different application scenarios and datasets is still unknown. Thus, VCM is imperfect.

For decoupling the dependency of the regulatory factor, this paper uses a new method, called DMaCM elaborated in Section 5, by utilizing double Mahalanobis distances to improve the similarity measurement of QoS cloud models for evaluating the performance of CSs more exactly.

To the best of our knowledge, no existing research has systematically studied the variation characteristics of QoS from four aspects, and utilized the QoS cloud models to analyze the QoS variation in multiple periods, and employed the double Mahalanobis distances to measure the similarity of QoS cloud models. The proposed MCDM procedure using a TOPSIS method improved based on the Mahalanobis distance is the application of a successful extension of TOPSIS on the CS selection problem.

3 Problem Statement

In general, a user cannot experience every candidate CS due to the enormous number of candidate services with similar functions. Thus, the QoS data of these CSs unused by a current user is collected from other users who used them, and plays an important role in selecting a most suitable CS for the current user [6], [8], [25], [17]. To learn the performance of candidates, the continuous monitoring of QoS has been an urgent need currently. Different time periods are examined in this way which assists in a more fine-grained CS selection for the current user which might have different preferences for QoS over these periods. The time series data about one QoS parameter could be depicted with a user-service-time matrix. In this matrix, a period covers specific time points. An example of user-service-time matrix for response time is shown in Fig. 1.

This matrix records the time series data about response time of CSs invoked by users in multiple timeslots. Multiple user-service-time matrices are obtained when multiple QoS parameters are monitored. Obviously, the time series QoS data is complex, and the underlying complexity includes: (1) The data volume is massive. There is a great deal of CSs and users in practice. Moreover, if a CS is monitored by an agent every 15 minutes [11], [24], then there are 96 timeslots in one day; QoS may be involved to 5 parameters (e.g., response time, throughput, availability, successability,

TABLE 3
An Example of User-Time QoS Sub-Matrix

Users	Timeslots											
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
u_1	**1.01**	**1.97**	**1.86**	**3.63**	**1.05**	**2.44**	**2.21**	**2.93**	**1.58**	**3.83**	**1.23**	**3.97**
u_2	1.92	1.90	1.95	1.99	1.92	2.01	2.02	1.99	1.94	1.96	1.98	1.96
u_3	**1.24**	**1.63**	**1.87**	**3.57**	**1.11**	**2.18**	**2.40**	**1.59**	**2.96**	**3.72**	**3.70**	**1.21**
u_4	2.32	3.31	3.45	2.22	1.79	2.84	2.49	1.47	4.01	3.23	1.81	2.04
u_5	1.66	2.13	1.92	1.62	1.48	2.20	2.31	2.54	2.11	0.98	2.92	2.31

TABLE 4
User Similarity Values for Example 1

Measure	$S(u_1,u_2)$	$S(u_1,u_3)$	$S(u_1,u_4)$	$S(u_1,u_5)$
PCC similarity	**0.7028**	0.6226	0.4703	0.4023
COS similarity	0.5413	0.5593	0.5643	**0.5655**
KRCC similarity	**0.6473**	0.5606	0.4545	0.5534
ED similarity	**0.2132**	0.1920	0.1782	0.1870
NED similarity	0.1720	0.1739	0.1345	**0.1828**

reliability, latency[2]). (2) The data related to one user in user-service-time matrices is sparse as one user usually only invoked a small number of CSs from all available ones. (3) Most of all, the key variation characteristics of CSs' QoS are hidden in the massive and sparse time series. These complexities make the conventional methods face two challenges. The two challenges are how to exactly identify the neighboring users for a current user and how to select an appropriate CS with optimal QoS meeting user's period preferences. Two examples about these two challenges are demonstrated as follows.

Example 1. Challenge in exactly identifying the neighboring users for a current user.

To accurately predict the QoS of a CS that is not used by a current user, it is the key precondition to exactly identify the neighboring users for the current user. Those CSs that are invoked by the current user and other users are denoted as training CSs. Then, the user-time QoS sub-matrices about training CSs need to be extracted from Fig. 1. An example of the user-time QoS sub-matrix about the response time of the storage of data for a cloud storage service is shown in Table 3.

In Table 3, the time series QoS data consists of the monitoring values in 12 timeslots. We calculate the similarity between the current user and other users. Only the users with enough large similarity values may become the neighboring users of the current user. PCC similarity,[3] cosine (COS) similarity,[4] KRCC similarity,[5] Euclidean distance (ED) similarity and normalized Euclidian distance (NED) similarity[6] are widely used in collaborative filtering. The calculation formulas of the above similarity measurements are given in Appendix A.1, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TSC.2019.2895784. Due to the differences among measurement methods, the similarity values obtained by different methods need to be normalized within the range [0, 1] for the convenience of comparison. Let $S(u_i, u_j)$ be the normalized similarity between u_i and u_j. The closer $S(u_i, u_j)$ is to 1, the more u_j is similar to u_i. The normalized methods are given in Appendix A.2 available in the online supplemental material. Assuming u_1 is the current user, the similarity values between u_1 and other users are obtained, shown in Table 4.

From Table 4, the PCC, KRCC and ED similarities affirm that u_2 is more similar to u_1 than other users. u_5 is regarded as the most similar user to u_1 by the COS and NED similarities. However, we can draw a different conclusion by analyzing the time series variation characteristics of the response time, shown in Fig. 2a.

From Fig. 2a, obviously, u_3 should be the user most similar to u_1 because their response time values demonstrate the consistent change trend in most of timeslots, especially, the drastic fluctuation in some adjacent timeslots. On the contrary, the response time experienced by u_2 is stable from beginning to end, totally different from u_1. According to the previous research [7], if two users always obtain the similar abnormal data, they should be likely located in the similar network or geographic position. When they invoke one CS in the same or adjacent timeslots, they could experience a similar variation of QoS caused by the unstable network or the fluctuation of load. Thus, the traditional user similarity methods cannot exactly measure the variation of QoS, failing to support the comprehensive analysis of variation, and result in a limited identification accuracy of neighboring users in the unstable cloud environment.

Example 2. Challenge in selecting the appropriate CS with optimal QoS meeting user's period preferences.

Let $c_1 - c_5$ be 5 candidate CSs for a current user. Their QoS data in 12 timeslots, namely, the service-time QoS sub-matrix for one user, are shown in Table 5.

To evaluate the performance of every candidate CS, the distribution feature of time series QoS data can be systematically analyzed by the traditional statistical indicators, such as mode, median, mean, range, standard deviation (SD) and coefficient of variation (CV). The mode, median and mean illustrate the central tendency of data; the range, SD and CV measure the degree of dispersion of data. More details about the traditional indicators (e.g., mode, median, mean, range, SD, CV) can be found in Appendix A.3 available in the online supplemental material. The statistical results of time series QoS data of 5 candidate CSs are shown in Table 6.

In Table 6, although the optimal values of various indicators are rendered in bold, it is still difficult to judge which CS is the optimal one. In general, c_1 gains three optimal

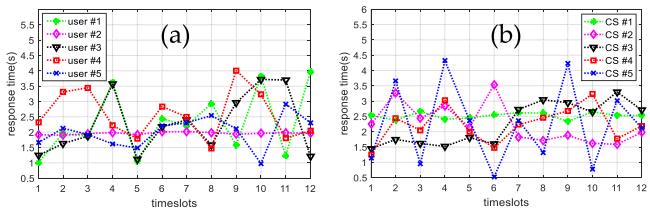

Fig. 2. Time series variation analysis. (a) Example 1; (b) Example 2.

2. http://www.uoguelph.ca/~qmahmoud/qws/
3. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient
4. https://en.wikipedia.org/wiki/Cosine_similarity
5. https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient
6. https://en.wikipedia.org/wiki/Euclidean_distance

TABLE 5
Example of the Service-Time QoS Sub-Matrix for One User

CSs	Timeslots											
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
c_1	2.53	2.38	2.66	2.41	2.45	2.55	2.62	2.61	2.34	2.65	2.53	2.53
c_2	2.25	3.27	2.44	2.84	2.14	3.54	**1.83**	**1.72**	**1.88**	**1.62**	**1.59**	**1.98**
c_3	**1.45**	**1.75**	**1.62**	**1.52**	**1.81**	**1.60**	2.73	3.04	2.95	2.64	3.30	2.71
c_4	1.26	2.44	2.05	3.03	2.00	1.47	2.23	2.45	2.68	3.23	1.77	2.19
c_5	1.14	3.66	0.95	4.32	2.37	0.52	2.37	1.32	4.23	0.77	3.02	2.12

values in the range, SD and CV indicators. However, its mean value is the largest in all CSs. Due to the different significances and measurement scales of these indicators, it is difficult to design a reasonable aggregation operator that could integrate all values of them for reflecting the comprehensive characteristics of CSs' performance hidden in time series data. In contrast, the line chart of time series data is a nice choice to reveal the important variation feature of QoS, as shown in Fig. 2b.

From Fig. 2b, the QoS of c_2 fluctuates greatly in $t_1 - t_6$, and exerts a stable performance in $t_7 - t_{12}$ with a small mean; the QoS of c_3 fluctuates greatly in timeslots $t_7 - t_{12}$, and exerts the stable performance in $t_1 - t_6$ with a small average value; although the QoS of c_4 is very unstable in 12 timeslots, its average value is smaller than other CSs and the maximum value of its response time is still within tolerable range; though the mean of c_5 is the smallest among CSs, c_5 shows the most unstable performance, especially with the quite large response time values in t_4 and t_9. The line chart is helpful to identify the variation feature of time series QoS data and to select the suitable CSs according to users' requirements. For example, if one user attaches great importance to $t_1 - t_6, c_3$ maybe the best candidate; if one user attaches great importance to $t_7 - t_{12}, c_2$ maybe the best choice. However, it will become infeasible to judge the performance of CSs based on the line chart when there are many candidate CSs or timeslots. From the above, we need a more effective method to comprehensively analyze the variation feature of time series QoS data and evaluate the performance of QoS for supporting the decision-making of CS selection.

4 Variation Analysis of Time Series QoS Data

In this section, we utilize the QoS cloud model to analyze the variation of CSs' QoS based on time series data. The definitions of some key symbols used in the following sections are shown in Table 7.

TABLE 6
Statistical Analysis of Time Series QoS Data for Example 2

Indicators	c_1	c_2	c_3	c_4	c_5
mode	2.53	1.59	1.45	**1.26**	2.37
median	2.53	**2.06**	2.23	2.21	2.25
mean	2.52	2.26	2.26	**2.23**	**2.23**
range	**0.32**	1.95	1.85	1.97	3.80
SD	**0.11**	0.65	0.69	0.58	1.34
CV	**4.22**	28.63	30.56	26.15	60.09

TABLE 7
Definitions of the Key Symbols

Symbols	Explanation
u_o	a current user
$C^T = \{c_1, c_2, \ldots, c_y\}$	C^T is the set of training CSs that are ever used by u_o; y is the number of training CSs
$U = \{u_1, u_2, \ldots, u_x\}$	U is a set of all users; x is the number of all users
$U^N = \{u_1, u_2, \ldots, u_X\}$	U^N is a set of neighboring users; X is the number of neighboring users
$C^C = \{c_1, c_2, \ldots, c_Y\}$	C^C is a set of candidate CSs that meeting the u_o's functional requirements; Y is the number of candidate CSs
$T = \{t_1, t_2, \ldots, t_N\}$	T is a set of timeslots; N is the number of timeslots
$P = \{p_1, p_2, \ldots, p_Z\}$	T is a set of periods; Z is the number of periods; one period consists of multiple timeslots
P^*	a whole period consisting of all timeslots
Ψ	the total number of QoS parameters

4.1 Four Aspects of QoS Variation Characteristics

This paper portrays the variation characteristics of CSs' QoS from four aspects as follows:

1) Central tendency: A central tendency is a typical value for a probability distribution.[7] The most common measures of central tendency are the arithmetic mean, the median and the mode. For the positively monotonic QoS parameters, such as throughput, the larger the central tendency, the better the QoS. For the negatively monotonic QoS parameters, such as response time, the smaller the central tendency, the better the QoS.
2) Variation range: It expresses the fluctuation range of most sample data. The simplest measurement of variation range is range which is simply the highest value minus the lowest value. The probability range of occurrence for most of data could also be used to measure the variation range. The latter should be more suitable to measure the variation range in a dynamic cloud environment. Obviously, the narrower the variation range is, the closer most of QoS data is to the central tendency value.
3) Frequency of variation: Frequency of variation can be measured by the number of variations within a given period. The lower the frequency of variation, the more stable is the QoS.
4) Period: The variation demonstrates different features in different periods. A period covering one hour or several hours may be fine-grained or coarse-grained, depending on the internal functional characteristics (e.g., computational complexity or storage latency) and external runtime environment (e.g., network throughput) of CSs. The existing research indicates

7. https://en.wikipedia.org/wiki/Central_tendency

TABLE 8
User similarity values in p_1 and p_2 for Example 1

Measure	$S(u_1,u_2)$		$S(u_1,u_3)$		$S(u_1,u_4)$		$S(u_1,u_5)$	
	p_1	p_2	p_1	p_2	p_1	p_2	p_1	p_2
PCC similarity	0.86	0.43	**0.99**	0.27	0.56	**0.47**	0.60	0.36
COS similarity	0.54	0.54	0.50	**0.59**	**0.55**	0.57	0.54	0.58
KRCC similarity	0.79	0.43	**0.94**	0.28	0.54	**0.44**	0.66	0.21
ED similarity	0.32	**0.25**	**0.67**	0.19	0.25	0.22	0.31	0.21
NED similarity	0.22	**0.24**	**0.48**	0.18	0.17	0.19	0.26	0.22

that the variation of one CS's QoS is often inconsistent in different periods. For example, the central tendency, variation range and frequency of variation of one CS's QoS during peak hours are quite different from them at off-peak hours. In the CS selection problem, users may have different preferences in periods for a CS. A CS possibly becomes an optimal one for a current user if it has good central tendency, narrow variation range and low frequency of variation in the specified periods that the user is concerned about.

Let P^* represent a coarse-grained period consisting of all timeslots. In Example 1, P^* consisting of 12 timeslots. Without loss of generality, assuming that P^* can be split into two fine-grained periods p_1 and p_2, p_1 covers timeslots #1-#6 and p_2 covers timeslots #7-#12. Based on fine-grained periods, the more valuable information can be explored by recalculating the user similarity and the statistical values of time series QoS data. The results are shown in Tables 8 and 9.

From Table 8, the majority of similarity methods, including PCC similarity, KRCC similarity, ED similarity and NED similarity, identify u_3 as the most similar to u_1 in p_1. However, these methods do not correctly select the most similar user for u_1 in p_2. The reason lies in that these methods cannot fully explore the time series data related to users and find their similar fluctuations features in some adjacent timeslots. In addition, some users with the distinctly different variation of QoS in p_1 and p_2 also obtained large similarity values, such as $S(u_1, u_2)$. Table 9 can clearly display that the QoS of c_3 outperforms c_2 in all six indicators in p_1, and the QoS of c_2 outperforms c_3 in all six indicators in p_2. However, from Table 9, we cannot discriminate the optimal CS in every period according to the six indicators. The different CS is selected as the optimal one by different indicator because every indicator only captures a local feature from one angle. To sum up, the variation analysis of QoS from the perspective of periodicity facilitates to seek the optimal CS meeting the user preference on usage periods.

TABLE 9
Statistics of QoS data in p_1 and p_2 for Example 2

Indicators	c_1		c_2		c_3		c_4		c_5	
	p_1	p_2	p_1	p_2	p_1	p_2	p_1	p_2	p_1	p_2
mode	2.38	2.34	2.14	1.59	1.45	2.64	1.26	1.77	**0.52**	**0.77**
median	2.49	2.57	2.64	**1.78**	**1.61**	2.84	2.03	2.34	1.76	2.25
mean	2.50	2.55	2.75	**1.77**	**1.63**	2.90	2.04	2.43	2.16	2.31
range	**0.28**	**0.31**	1.40	0.39	0.36	0.66	1.77	1.46	3.80	3.46
SD	**0.10**	**0.11**	0.57	0.15	0.14	0.25	0.64	0.50	1.56	1.23
CV	**4.16**	**4.42**	20.73	8.65	8.35	8.66	31.55	20.50	72.18	53.42

TABLE 10
QoSCMs for Example 1

Users	P^*			p_1			p_2		
	Ex	*En*	*HE*	*Ex*	*En*	*HE*	*Ex*	*En*	*HE*
u_1	**2.31**	**1.10**	**0.26**	**1.99**	**0.87**	**0.44**	**2.63**	**1.19**	**0.33**
u_2	1.96	0.04	0.01	1.95	0.04	0.01	1.98	0.03	0.01
u_3	**2.27**	**1.05**	**0.33**	**1.93**	**0.79**	**0.42**	**2.60**	**1.08**	**0.23**
u_4	2.58	0.82	0.24	2.66	0.68	0.19	2.51	0.93	0.22
u_5	2.02	0.50	0.13	1.84	0.31	0.10	2.20	0.54	0.37

To analyze the variation characteristics of time series QoS data more comprehensively, we introduce the cloud model theory and exploit the QoS cloud model to depict the four aspects of variation in multiple periods.

4.2 Variation Analysis via QoS Cloud Model

Based on the cloud model theory, a QoS cloud model (QoSCM) can be established based on time series data to analyze the variation of QoS. A QoSCM is defined as $\mathrm{QoSCM} = \{Ex, En, HE\}$. *Ex* is the expectation of QoS; *En* is the entropy of QoS, which is the standard variance of *Ex*; *HE* is the hyper entropy of QoS, which is the standard variance of *En*. An example of QoSCM is shown in Appendix Fig. A.1 available in the online supplemental material.

From Fig. A.1 available in the online supplemental material, a QoSCM utilizes the Gaussian distributions to describe the variation of QoS. As the expectation of QoS, *Ex* is the value with the largest occurrence possibility in the time series data, reflecting the centralized tendency of data. *En* describes the breadth of cloud, reflecting the range of variation; *HE* embodies the thickness of cloud, reflecting the frequency of variation. Thus, the QoSCM makes it possible to learn the integrated and qualitative variation features of QoS based on time series data. A QoSCM consists of many cloud drops. The time series QoS data of a CS concerning a certain user could be viewed as the cloud drops and sent to a reverse cloud generator (RCG) [36], where QoSCM is calculated by:

$$\begin{cases} Ex &= \frac{1}{N}\sum_{i=1}^{N} b_i \\ En &= \sqrt{\frac{\pi}{2}} \times \sigma = \sqrt{\frac{\pi}{2}} \times \frac{1}{N}\sum_{i=1}^{N} |b_i - Ex| \\ HE &= \sqrt{|S^2 - En^2|} = \sqrt{\left|\frac{1}{N-1}\sum_{i=1}^{N} (b_i - Ex)^2 - En^2\right|} \end{cases}, \tag{1}$$

where b_i is the QoS data obtained in timeslot #*i*; *Ex* is the mean of QoS data; $\boldsymbol{\sigma}$ is the standard deviation of Ex; S^2 is the sample variance of Ex; N is the number of timeslots. Given a set of data related to any one period, a QoSCM can be gained for this period by Eq. (1).

We define a set of QoSCMs to analyze the QoS variation of a CS in multiple periods. The sets of QoSCMs for Examples 1 and 2 in p_1 and p_2 are shown in Tables 10 and 11. From Table 10, the QoSCMs of u_3 and u_1 are very alike in p_1 and p_2, compared to other users. From Table 11, we cannot directly identify which one is the optimal CS in p_1 or p_2, however, c_2 in p_2 and c_3 in p_1 show better performance than in P^* compared to other CSs. In sum, it can be seen that the QoSCMs in multiple periods are helpful for recognizing the variation features of QoS. Based on the above variation

TABLE 11
QoSCMs for Example 2

CSs	P^*			p_1			p_2		
	Ex	*En*	*HE*	*Ex*	*En*	*HE*	*Ex*	*En*	*HE*
c_1	2.52	0.11	0.01	2.50	0.10	0.01	2.55	0.10	0.05
c_2	2.26	0.64	0.10	2.75	0.59	0.15	**1.77**	**0.16**	**0.04**
c_3	2.26	0.80	0.40	**1.63**	**0.13**	**0.04**	2.90	0.25	0.03
c_4	2.23	0.56	0.18	2.04	0.58	0.27	2.43	0.45	0.20
c_5	2.23	1.37	0.29	2.16	1.62	0.43	2.31	1.13	0.49

analysis, we propose the variation-aware methods to identify the neighboring users and to select the optimal CS in the next sections.

5 Variation-Aware Neighboring User Identification

Let u_{o} be the current user, $C^{\mathrm{T}} = \{c_1, c_2, \ldots, c_y\}$ be the set of training CSs that are ever used by u_{o}, and $U = \{u_1, u_2, \ldots, u_x\}$ be the set of all users. If u_i ever used a CS or multiple CSs in C^{T}, u_i is called a training user. The different set of training users can be found for different training CS. $P = \{p_1, p_2, \ldots, p_Z\}$ is the set of periods obtained by analyzing the application requirements of u_{o}. Every period might have the same or a different number of timeslots.

The time series data about a QoS parameter of a training CS c_k in every period is viewed as cloud drops and sent into the RCG for generating a QoSCM. Then, the QoSCMs of all training users associated with c_k for one QoS parameter q are obtained as follows:

$$CM^{k,q} = \begin{bmatrix} CM_o^{k,q} \\ CM_1^{k,q} \\ \vdots \\ CM_r^{k,q} \end{bmatrix} = \begin{bmatrix} cm_{o,1}^{k,q} & cm_{o,2}^{k,q} & \ldots & cm_{o,Z}^{k,q} \\ cm_{1,1}^{k,q} & cm_{1,2}^{k,q} & \ldots & cm_{1,Z}^{k,q} \\ \vdots & \vdots & cm_{i,j}^{k,q} & \vdots \\ cm_{r,1}^{k,q} & cm_{r,2}^{k,q} & \ldots & cm_{r,Z}^{k,q} \end{bmatrix}, \tag{2}$$

where $CM_{\mathrm{o}}^{k,q}$ and $CM_i^{k,q}$ are the QoSCM vectors of c_k relevant to u_{o} and training user u_i, respectively; r is the number of the training users associated with c_k; $cm_{i,j}^{k,q} = (Ex_{i,j}^{k,q}, En_{i,j}^{k,q}, HE_{i,j}^{k,q})$ is the QoSCM of c_k relevant to u_i in p_j.

Next, the similarity between $CM_{\mathrm{o}}^{k,q}$ and $CM_i^{k,q}$ is calculated. By integrating multiple QoS parameters and training CSs, the comprehensive similarity of every training user is obtained. The top K most similar users are chosen to form the set of neighboring users (U^{N}) for u_{o}, and a user should be removed from U^{N} if his/her similarity is equal to or smaller than 0 [4]. The QoS data from U^{N} are collected to predict the QoS of the CSs unused by u^{o}.

To overcome the limitations of the existing methods, this paper utilizes the double Mahalanobis distances to improve the similarity measurement of QoSCMs. The Mahalanobis distance is a method of measuring the distance of data covariance that can effectively calculate the similarity between two unknown sample sets. Unlike the Euclidean distance, the Mahalanobis distance is independent of the measurement scales, and it remains unaffected by the different dimensions between coordinates. Recently, the Mahalanobis distance has been applied in many research fields [37], [38].

Let $V = \{v_i\}$ be a vector set corresponding to a sample set. Every sample is observed in L indexes, and vector v_i consists of L dimensions. The Mahalanobis distance between vector v_i and vector v_j is calculated by:

$$D_{v_i,v_j}^{MaCM} = \sqrt{(v_i - v_j)\mathrm{H}^{-1}(v_i - v_j)^{\mathrm{T}}}, \tag{3}$$

where T is the transposition operation; H^{-1} denotes the inverse of the covariance matrix[8] of V, and it is a symmetric positive definite matrix, $\mathrm{H}^{-1} = \{h_{m,n}\}\,(1 \le m, n \le L)$. Then, the Mahalanobis distance is also defined by:

$$D_{v_i,v_j}^{MaCM} = \sqrt{\sum\nolimits_{1 \le m,n \le k} h_{m,n}(v_{i,m} - v_{j,m})(v_{i,n} - v_{j,n})}. \tag{4}$$

When H^{-1} is an identity matrix, the L dimensions of samples are within the same fluctuation range. Then, we get

$$D_{v_i,v_j}^{MaCM} = \sqrt{\sum\nolimits_{m=1}^{k} (v_{i,\mathrm{m}} - v_{j,m})^2}. \tag{5}$$

The Mahalanobis distance-based similarity of **c**loud **m**odels is noted as MaCM. Let $\overrightarrow{V_{\mathrm{o},j}^{k,q}} = (v_{\mathrm{o},j\,1}^{k,q}, v_{\mathrm{o},j\,2}^{k,q}, v_{\mathrm{o},j\,3}^{k,q}) = (Ex_{\mathrm{o},j}^{k,q}, En_{\mathrm{o},j}^{k,q}, HE_{\mathrm{o},j}^{k,q})$ and $\overrightarrow{V_{i,j}^{k,q}} = (v_{i,j1}^{k,q}, v_{i,j2}^{k,q}, v_{i,j3}^{k,q}) = (Ex_{i,j}^{k,q}, En_{i,j}^{k,q}, HE_{i,j}^{k,q})$ be the QoSCMs $cm_{\mathrm{o},j}^{k,q}$ and $cm_{i,j}^{k,q}$ respectively. The Mahalanobis distance between $cm_{\mathrm{o},j}^{k,q}$ and $cm_{i,j}^{k,q}$ is calculated by:

$$\begin{aligned} D_{\overrightarrow{V_{o,j}^{k,q}},\overrightarrow{V_{i,j}^{k,q}}}^{MaCM} &= D_{cm_{\mathrm{o},j}^{k,q},cm_{i,j}^{k,q}}^{MaCM} \\ &= \sqrt{\sum\nolimits_{1 \le m,n \le 3} h_{m,n}\left(v_{\mathrm{o},j\,m}^{k,q} - v_{i,j\,m}^{k,q}\right)\left(v_{\mathrm{o},j\,n}^{k,q} - v_{i,j\,n}^{k,q}\right)}. \end{aligned} \tag{6}$$

The precision of the Mahalanobis distance may decline when the size of the samples is small, especially in the case of $|V| > L$, where $|V|$ is the number of samples. In the CS selection problem, there may be a few users who invoked the same CSs with u_{o}. In this case, the Mahalanobis distance might become imprecise due to the insufficient numbers of samples. Therefore, this paper proposes the double Mahalanobis distance method to improve the similarity of QoSCMs by reducing the number of dimensions L from 3 to 2, denoted as DMaCM. The dimensionality reduction ensures that DMaCM obtains a more precise result than MaCM. In DMaCM, the vector $\overrightarrow{V_{i,j}^{k,q}}$ is divided into two sub-vectors: $\overrightarrow{V_{i,j1}^{k,q}} = (v_{i,j\,1}^{k,q}, v_{i,j\,2}^{k,q})$ and $\overrightarrow{V_{i,j\,2}^{k,q}} = (v_{i,j\,2}^{k,q}, v_{i,j\,3}^{k,q})$. The double Mahalanobis distances between $cm_{\mathrm{o},j}^{k,q}$ and $cm_{i,j}^{k,q}$ is calculated by:

$$D_{cm_{\mathrm{o},j}^{k,q},cm_{i,j}^{k,q}}^{DMaCM} = \frac{1}{2} \times \left(D_{\overrightarrow{V_{\mathrm{o},j1}^{k,q}},\overrightarrow{V_{i,j\,1}^{k,q}}}^{MaCM} + D_{\overrightarrow{V_{\mathrm{o},j2}^{k,q}},\overrightarrow{V_{i,j\,2}^{k,q}}}^{MaCM} \right). \tag{7}$$

8. https://en.wikipedia.org/wiki/Covariance_matrix

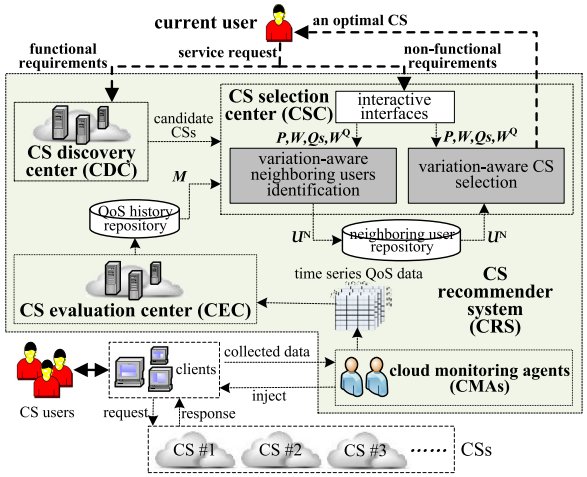

Fig. 3. A variation-aware CS recommender system.

The smaller the double Mahalanobis distance, the more similar the two QoSCMs. Thus, the similarity value between QoSCMs can be normalized by:

$$S(cm_{\mathrm{o},j}^{k,q}, cm_{i,j}^{k,q})^{DMaCM} = 1 \Big/ \left(1 + D^{DMaCM}_{cm_{\mathrm{o},j}^{k,q}, cm_{i,j}^{k,q}}\right). \tag{8}$$

Let $S_{i,j}^{k,q}$ be the similarity between u_i and u_{o} in period #j for c_k with respect to QoS parameter q. To aggregate the similarity values in Z periods, the weight vector of periods, namely, the period preference, reflecting the importance degree of every period for u_{o}, is defined as

$$W = [w_1, w_2, \ldots, w_Z], \tag{9}$$

where w_j is the period #j's weight; $0 \leq w_j \leq 1$; $\sum_{j=1}^{Z} w_j = 1$. Then, the similarity value in Z periods is obtained by:

$$\delta_i^{k,q} = \sum_{j=1}^{Z} S_{i,j}^{k,q} \times w_j. \tag{10}$$

In practice, u_{o} usually pay attention to multiple QoS parameters. Then, the comprehensive similarity of u_i for c_k with respect to all QoS parameters can be calculated by:

$$\delta_i^{k*} = \sum_{q=1}^{\Psi} \delta_i^{k,q} \times w_q^{\mathrm{Q}}, \tag{11}$$

where Ψ is the total number of QoS parameters and w_q^{Q} represents the weight of the qth QoS parameter. The fuzzy analytic hierarchy process (FAHP) method [39] can be used to objectively assign the weights of multiple QoS parameters. The default value of w_q^{Q} is $1/\Psi$ when u_{o} have no explicit preference for QoS parameters.

Next, employ the weighted average operator to calculate the comprehensive similarity between u_i and u_{o} in Z periods for all training CSs with respect to all QoS parameters as follows:

$$\delta_i = \frac{1}{|C^i|} \sum_{c_k \in C^i} \delta_i^{k*} = \frac{1}{|C^i|} \sum_{k=1}^{y} \sum_{q=1}^{N^{\mathrm{Q}}} \sum_{j=1}^{Z} S_{i,j}^{k,q} \times w_j \times w_q^{\mathrm{Q}}, \tag{12}$$

where C^i is the set of training CSs that are invoked by u_i; $|C^i|$ is the total number of training CSs in C^i.

After calculating the comprehensive similarities between u_{o} and all training users, a set of the K most similar users can be identified for u_{o}, denoted as Top-$K(u_{\mathrm{o}})$ [4], [24], [25], [28]. For u_{o}, the set of neighboring users are obtained by:

$$U^{\mathrm{N}}(u_{\mathrm{o}}) = \{u_i | u_i \in \mathrm{Top} - K(u_{\mathrm{o}}), \delta_i > 0, u_i \neq u_{\mathrm{o}}\}. \tag{13}$$

According to Eq. (13), the similarity of a neighboring user should be greater than 0. Note that the neighboring relations are not symmetrical. If u_i is in the $U^{\mathrm{N}}(u_j)$, it does not mean that u_j is in the $U^{\mathrm{N}}(u_i)$. The existing research on the recommendation algorithm via collaborative filtering [40] has indicated that even for a small value of K the recommendation algorithm can provide reasonably accurate results and increasing the value of K does not lead to significant improvements of accuracy. Due to the tradeoffs between performance and recommendation quality, the value of K is usually set as 10 [40].

The complexity of the neighboring user identification method is analyzed as follows: (1) Based on Eqs. (1), (2), the complexity of calculating the QoSCMs for all training users associated with all training CS for all QoS parameters is less than $\mathrm{O}(\Psi \times Z \times y \times x \times N)$. (2) Based on Eq. (8), the complexity of calculating the double Mahalanobis distances of all training users is less than $\mathrm{O}(\Psi \times Z \times y \times x^2)$. (3) Based on Eqs. (10), (11), (12), the complexity of calculating the comprehensive similarity of all training users is less than $\mathrm{O}(x \times \Psi \times Z \times y)$. (4) Based on Eq. (13), the complexity of selecting the set of neighboring users is $\mathrm{O}(x)$. To ensure that enough data is available in a period, the length of any a period should be larger than or equal to 2 hours, namely, $Z \leq 12$. When the monitoring frequency is 15 minutes, N is estimated as 96. Thus, considering that usually $\Psi \leq 5$ and $N << x$, the general complexity of this method is $\mathrm{O}(y \times x^2)$.

6 Variation-Aware CS Selection

6.1 Framework of a CS Recommender System

With the support of the variation-aware neighboring user identification, the framework of a variation-aware CS recommender system (CRS) can be established, shown in Fig. 3.

The core components of CRS include:

1) Cloud monitoring agents (CMAs): CMAs are injected into the clients of CS users for monitoring the performance of CSs. These agents collect the monitoring data about multiple QoS parameters usually every 15 minutes and submit them with the timestamps and users' time zone information to the CS evaluation center. The shared QoS data experienced by a user could be used to the reference information for helping other user to evaluate unknown candidate CSs, and also facilitate to exactly identify the neighboring users for this user in the future.
2) CS evaluation center (CEC): CEC receives the monitoring data from CMAs, and transforms them into a uniform format according to the time zone of CEC and stores them in the QoS history repository.
3) CS discovery center (CDC): CDC is responsible for finding the candidate CSs meeting the functional requirements of the current user (u_{o}). CDC can utilize the domain ontologies [20] to define the concept

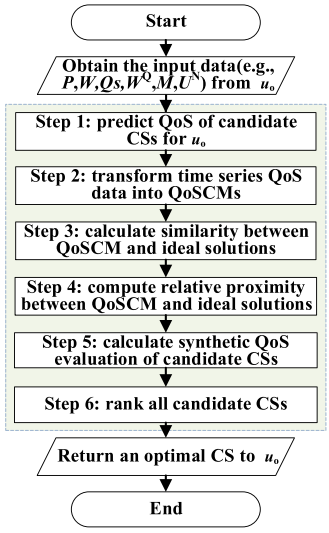

Fig. 4. MCDM procedure of variation-aware CS selection.

taxonomies and properties of CSs, and quantify the relations among concepts and between concepts and properties. The set of candidate CSs $C^{\mathrm{C}} = \{c_1, c_2, \ldots, c_Y\}$ will be exactly identified for u_{o} by calculating the similarity between the concepts from functional requirements and the ones from the functional descriptions of available CSs.

4) CS selection center (CSC): CSC is responsible for identifying the neighboring users and finding an optimal CS from candidates meeting the non-functional requirements of u_{o}. The main contributions of this paper focus on realizing the functionality of the CSC.

When u_{o} needs a CS, s/he submits the service request consisting of the functional and non-functional requirements to the CRS. CRS instructs u_{o} to input her/his requirements in a standardized way. For example, u_{o} can only use a given scale or name for each QoS parameter from a drop-down list box on a web page, which can reduce the workload of CDC. The non-functional requirements include: (1) the length and number of periods that need special attention (P); (2) the weight of periods (W); (3) the multiple QoS parameters (Qs) request by u_{o} (e.g., response time, throughput, availability, successability, reliability, latency); (4) the weights of QoS parameters (W^{Q}). To make the proposed approach function well, the time span of any a period is required to be larger than or equal to 2 hours for ensuring that enough time series data can be used to model variation characteristics of CSs' QoS. In addition, if u_{o} have no preference for W and W^{Q}, uniform weights are assigned to all QoS parameters or all periods. In this system, u_{o} can use the friendly "interactive interfaces" component to input the non-functional requirements, and the details are introduced in Appendix A.5 available in the online supplemental material.

After the service request is received, CRS executes the neighboring user identification and CS discovery in parallel. CRS delivers the functional requirements to CDC for the candidate CS matching. Meanwhile, CSC analyzes the time zone of u_{o}, extract the required history data about multiple QoS parameters, and transform it to the user's time zone. Based on P, W, Qs, W^{Q} and the relevant time series QoS data (M), CSC utilizes the "variation-aware neighboring user identification" component to acquire the neighboring users (U^{N}) for u_{o} and store them into the neighboring user repository. Then, $P, W, Qs, W^{\mathrm{Q}}, U^{\mathrm{N}}$ and M become the input data of the "variation-aware CS selection" component for selecting an optimal CS meeting the non-functional requirements from C^{C} for u_{o}. It is worth mentioning that the previous knowledge about U^{N} cached in the repository facilitates to enhance computation speed of neighboring user identification in the future.

In sum, this paper puts emphasis on the CS selection, neither on CS discovery nor on monitoring data collection. The existing approaches [20], [41] and techniques [16] could guarantee that the candidate CSs meeting user's functional requirements and the time series QoS data are available. Besides, this paper also does not concentrate on the implementation of a complete CS recommender system supporting the interactive visualization interfaces.

6.2 Procedure of Variation-Aware CS Selection

The variation-aware CS selection problem is formulated as an MCDM problem. In it, a period becomes a decision criterion for evaluating the QoS of CSs. To select the appropriate CS for u_{o}, the ranking values of candidate CSs should be computed based on their time series QoS data. To solve this problem, the TOPSIS method is improved to rank all candidate CSs by utilizing the Mahalanobis distance-based similarity measurement. The MCDM procedure of variation-aware CS selection is shown in Fig. 4.

TOPSIS [27] is a classic MCDM method. TOPSIS structures the positive and negative ideal solutions in an n-dimensional solution space and measures the relative proximity degree between evaluated objects and ideal solutions. Previous studies have applied it to solve many MCDM problems [20], [42]. The original TOPSIS method uses the Euclidean distance to calculate the similarity between the evaluated objects and the ideal solutions. However, the Euclidean distance cannot function well in TOPSIS method when the coordinates of the objects use the different scale [43], [44]. For CS selection decision involving QoSCMs, the measurement scales of coordinates, namely, the three numerical characteristics of a QoSCM, are just different, because *Ex* is usually many times greater than *En* and *HE*. In this case, errors will inevitably be introduced into decision results based on the Euclidean distances. Thus, for the three interdependent dimensions of QoSCMs with different measurement scales, we improve the original TOPSIS method by utilizing the Mahalanobis distance to measure the similarity between a QoSCM and the ideal solutions. Although the improvement of TOPSIS through the use of the Mahalanobis distance has been proposed for MCDM [43], [44]. We first apply it into the CS selection problem. Considering that enough candidate CSs are usually available, the double Mahalanobis distance is not used in TOPSIS for reducing the efforts in computation.

As such, the MCDM procedure for variation-aware CS selection via the improved TOPSIS method is as follows:

1) Based on the known QoS values from all neighboring users in U^{N}, predict the QoS of candidate CSs for u_{o} in every timeslot by using the user similarity as the weight. Obviously, the user with large similarity

value could provide more valuable reference data for u_o than those with small similarity value. Thus, the higher user similarity of a user u_i in U^N, the larger the weight. Taking one QoS parameter for example, for u_o, its predicted value of c_k in t_j can be calculated by:

$$b^{j*}_{o,k} = \sum\nolimits_{i=1}^{|U^N|} \left(b^j_{i,k} \times \delta_i\right) \Big/ \sum\nolimits_{i=1}^{|U^N|} \delta_i, \tag{14}$$

where δ_i is the comprehensive similarity between u_i and u_o; $b^j_{i,k}$ is the real QoS value of c_k in t_j from u_i; $|U^N| = X$, which represents the magnitude of U^N. Obviously, u_i with larger similarity value provides more valuable data for u_o.

In addition, for the three cases of cold start (e.g., a new user, a new CS, and the system startup), different prediction methods are provided: (a) For the new user case, namely, $U^N = \emptyset$, we have to use the average value of all users who invoked c_k as the predicted value of c_k; (b) for the new CS case, the average value of other candidates is viewed as the predicted value of c_k; (c) for the system startup case, all CSs and users are new, and the predicted value of c_k is unavailable. In this case, all candidate CSs will be directly recommended to u_o.

2) Transform the time series QoS data into QoSCMs. All QoS data in period #*j* is sent to the RCG, and a QoSCM is established for period #*j* according to Eq. (1). Thus, the QoSCM matrix for *Y* candidate CSs in *Z* periods for one QoS parameter *q* is defined as follows:

$$CM^q = \begin{bmatrix} CM^q_1 \\ CM^q_2 \\ \vdots \\ CM^q_Y \end{bmatrix} = \begin{bmatrix} cm^q_{1,1} & cm^q_{1,2} & \cdots & cm^q_{1,Z} \\ cm^q_{2,1} & cm^q_{2,2} & \cdots & cm^q_{2,Z} \\ \vdots & \vdots & cm^q_{k,j} & \vdots \\ cm^q_{Y,1} & cm^q_{Y,2} & \cdots & cm^q_{Y,Z} \end{bmatrix}, \tag{15}$$

where $cm^q_{k,j} = (Ex^q_{k,j}, En^q_{k,j}, HE^q_{k,j})$ represents the QoSCM of candidate CS #*k* in period #*j*.

3) Calculate the similarity between the QoSCM and the ideal solutions. First, extract the QoSCM matrix corresponding to period #*j* by:

$$CM^q_j = \begin{bmatrix} cm^q_{1,j} \\ cm^q_{2,j} \\ \vdots \\ cm^q_{Y,j} \end{bmatrix} = \begin{bmatrix} (Ex^q_{1,j}, En^q_{1,j}, HE^q_{1,j}) \\ (Ex^q_{2,j}, En^q_{2,j}, HE^q_{2,j}) \\ \vdots \\ (Ex^q_{Y,j}, En^q_{Y,j}, HE^q_{Y,j}) \end{bmatrix}. \tag{16}$$

Then, identify the positive and negative ideal solutions. An excellent CS should provide steady performance of QoS for u_o. The smaller *En* and *HE* mean the steadier QoS of CS. According to this principle, we observe all candidate CSs and compute the ideal solutions for the three features based on the best or worst possible value that can be taken based on the current performance of CSs. For the positively monotonic QoS parameters, the positive and negative ideal solutions for period #*j* are obtained by:

$$\begin{aligned} cm^{q+}_j &= \left\{ \max_{1\le k\le Y}\left\{Ex^q_{k,j}\right\}, \min_{1\le k\le Y}\left\{En^q_{k,j}\right\}, \min_{1\le k\le Y}\left\{HE^q_{k,j}\right\} \right\} \\ cm^{q-}_j &= \left\{ \min_{1\le k\le Y}\left\{Ex^q_{k,j}\right\}, \max_{1\le k\le Y}\left\{En^q_{k,j}\right\}, \max_{1\le k\le Y}\left\{HE^q_{k,j}\right\} \right\}. \end{aligned} \tag{17}$$

For the negatively monotonic QoS parameters, the positive and negative ideal solutions are identified as follows:

$$\begin{aligned} cm^{q+}_j &= \left\{ \min_{1\le k\le Y}\left\{Ex^q_{k,j}\right\}, \min_{1\le k\le Y}\left\{En^q_{k,j}\right\}, \min_{1\le k\le Y}\left\{HE^q_{k,j}\right\} \right\} \\ cm^{q-}_j &= \left\{ \max_{1\le k\le Y}\left\{Ex^q_{k,j}\right\}, \max_{1\le k\le Y}\left\{En^q_{k,j}\right\}, \max_{1\le k\le Y}\left\{HE^q_{k,j}\right\} \right\}. \end{aligned} \tag{18}$$

Finally, considering that enough candidate CSs are usually available, we can directly employ the typical Mahalanobis distance to calculate the similarity between the QoSCM of CS #*k* in period #*j* and the positive or negative ideal solutions for one QoS parameter *q* by:

$$\begin{aligned} D^{MaCM}_{cm^q_{k,j},cm^{q+}_j} &= \sqrt{\left(\overrightarrow{V^q_{k,j}} - cm^{q+}_j\right) \mathrm{H}^{-1} \left(\overrightarrow{V^q_{k,j}} - cm^{q+}_j\right)^{\mathrm{T}}} \\ D^{MaCM}_{cm^q_{k,j},cm^{q-}_j} &= \sqrt{\left(\overrightarrow{V^q_{k,j}} - cm^{q-}_j\right) \mathrm{H}^{-1} \left(\overrightarrow{V^q_{k,j}} - cm^{q-}_j\right)^{\mathrm{T}}}, \end{aligned} \tag{19}$$

where $\overrightarrow{V^q_{k,j}} = (v^q_{k,j\,1}, v^q_{k,j\,2}, v^q_{k,j\,3}) = (Ex^q_{k,j}, En^q_{k,j}, HE^q_{k,j})$.

4) Compute the relative proximity between a QoSCM and the ideal solutions. The relative proximity between a QoSCM and the ideal solutions for candidate CS #*k* in period #*j* for one QoS parameter *q* can be obtained by:

$$D(cm^q_{k,j}) = D^{MaCM}_{cm^q_{k,j},cm^{q-}_j} \Big/ \left(D^{MaCM}_{cm^q_{k,j},cm^{q+}_j} + D^{MaCM}_{cm^q_{k,j},cm^{q-}_j}\right), \tag{20}$$

where $D(cm^q_{k,j}) \in [0,1]$; the larger $D(cm^q_{k,j})$ means the better performance.

5) Calculate the synthetic QoS evaluation of candidate CS #*k*. The relative proximity value of candidate CS #*k* in *Z* periods for one QoS parameter *q* is obtained by:

$$T^q_k = \sum\nolimits_{j=1}^{Z} D(cm^q_{k,j}) \times w_j. \tag{21}$$

The synthetic evaluation value of candidate CS #*k* for Ψ QoS parameters are aggregated by:

$$T^*_k = \sum\nolimits_{q=1}^{\Psi} T^q_k \times w^{\mathrm{Q}}_q. \tag{22}$$

6) Rank all candidate CSs. The candidate CS with the largest synthetic QoS evaluation is recommended to u_o.

The complexity of the proposed CS selection method is analyzed as follows: (1) Based on Eq. (14), the complexity of predicting QoS values in all timeslots for all candidate CSs is $\mathrm{O}(Y \times \Psi \times X \times N)$. (2) Based on Eq. (15), the complexity of transforming time series QoS data into QoSCMs is $\mathrm{O}(Y \times \Psi \times Z \times N)$. (3) Based on Eqs. (16), (17), (18), (19), (20), (21), (22), the complexity of calculating synthetic evaluations of all candidate CSs is $\mathrm{O}(\Psi \times Z \times Y^2)$. (4) The

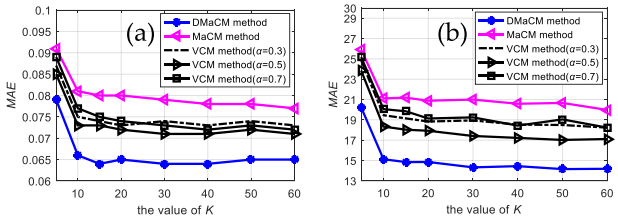

Fig. 5. Comparison analysis. (a) response time; (b) throughput.

complexity of ranking all candidate CSs is $O(Y)$. Considering that usually $\Psi \leq 5, Z \leq 12, Y << X \times N$, the general complexity of this method is $O(Y \times X \times N)$.

7 Experiments

7.1 Experimental Dataset and its Variation Analysis

WS-DREAM dataset #2[9] is used in the experiments. This dataset collects the real QoS data, including response time and throughput, from 142 users of 4,532 services around the world in 64 timeslots [11]. Compared with QWS that does not contain the time series data, this dataset is more suitable to the variation analysis of QoS, and most of the work also focuses on using it as shown in Table 1. The QoS of services in this dataset change significantly over time due to the fluctuation of loads and the instability of network. The analysis, as shown in Appendix Fig. A.2 available in the online supplemental material, reveals that the CV of response time of 3,873 services is larger than 1.0, so is the CV of throughput of 2,630 services. As we know, if the CV of a dataset is larger than 1.0, the distribution of it is considered high-variance.[10] We conducted experiments to illustrate the effectiveness of the QoS cloud models in analyzing the time series data and recognizing the variation of QoS. The detailed analysis is provided in Appendix A.7 available in the online supplemental material.

7.2 Metrics Indicators

Considering that the neighboring users is directly used to predict the QoS, we utilize mean absolute error (*MAE*) [24], [25] to evaluate the accuracy of the neighboring user identification method. *MAE* is defined by:

$$MAE = \frac{1}{total} \sum_{i=1}^{total} \sum_{j=1}^{N} \left| v_{i,j}^{\mathrm{B}} - v_{i,j}^{\mathrm{P}} \right|, \tag{23}$$

where $v_{i,j}^{\mathrm{B}}$ represents the real QoS experienced by the current user (u_{o}) in timeslot #*j* of the unknown CS in the *i*th experiment; $v_{i,j}^{\mathrm{P}}$ represents the predicted QoS value of it in timeslot #*j* in the *i*th experiment; *total* denotes the total number of experiments executed. Obviously, the smaller *MAE* means the higher accuracy.

Inspired by the difference degree [10] that is used to measure the accuracy of an ordered list of services in a CS ranking approach, we define the uniformity degree (U^{D}) to measure the accuracy of the selected optimal CS in a CS selection approach as follows:

$$U^{\mathrm{D}} = \frac{1}{total} \sum_{i=1}^{total} \frac{1}{O_i^{\mathrm{B}}}, \tag{24}$$

where O_i^{B} represents the selected optimal CS's order in the baseline list in the *i*th experiment. A baseline list of all candidate CSs can be obtained based on the real QoS evaluations by Eq. (22). Obviously, a larger U^{D} means a better accuracy.

7.3 Accuracy Analysis of Neighboring User Identification Method

First, to illustrate the accuracy of proposed neighboring user identification method, a case is analyzed based on Example 1. The details are provided in Appendix A.8 available in the online supplemental material. The case analysis indicates: (1) the MaCM method might be unsatisfactory in a coarse-grained period because it might mistakenly identify the most similar user; (2) in a fine-grained period, although the MaCM method could provide the correct result, it cannot exactly reflect the differences between users; (3) though the VCM method correctly identifies the most similar user when the regulatory factor (α) is set as different values, it is still not satisfactory enough like the DMaCM method in the measurement precision of user similarity. Besides, it is a remained problem how to determine the exact value of $\boldsymbol{\alpha}$ based on theoretical analysis or experimental verification; (4) the DMaCM method is superior to other methods, and its accuracy might be better in a fine-grained period than in a coarse-grained period. Moreover, the DMaCM method is not dependent on any parameter to exhibit a higher accuracy.

Next, we employ *MAE* to assess the accuracy of neighboring user identification methods. Considering that the previous research have demonstrated the better performance of VCM method compared to other methods [8], the DMaCM method is compared with the MaCM method and the VCM method in experiments.

First, u_{o} and an unknown CS are selected randomly from the dataset. Then, the neighboring users identified by different methods are used to predict the response time of unknown CSs for u_{o} by Eq. (14). We are looking for top *K* most similar users, and the values of *K* are set within from 5 to 60. Every experiment is conducted 50 times, and the average *MAE* values are recorded. In experiments, the overall period mapping to the first and 60th timeslot is partitioned into 6 periods. Based on the previous research [9], a suggested length of period is larger than 6 in the dataset. Considering that the total number of timeslots is limited, the more periods mean a smaller number of timeslots in a period, which make it difficult to employ the cloud model to accurately describe the variation of QoS. The weight of each period is regulated within the interval $[0.0, 1.0]$ for simulating the variation in user requirements. Fig. 5 shows the results.

Fig. 5 shows that the DMaCM method outperforms the MaCM method and the VCM method for both the response time and throughput. The performance of the VCM method is not stable when $\boldsymbol{\alpha}$ is set as different values. Fig. 5 also shows that a large value of *K* can enhance the accuracy of neighboring user identification of three methods to some

9. https://github.com/wsdream/WS-DREAM/tree/master/data
10. https://blogs.sas.com/content/iml/2014/11/19/coefficient-of-variation.html

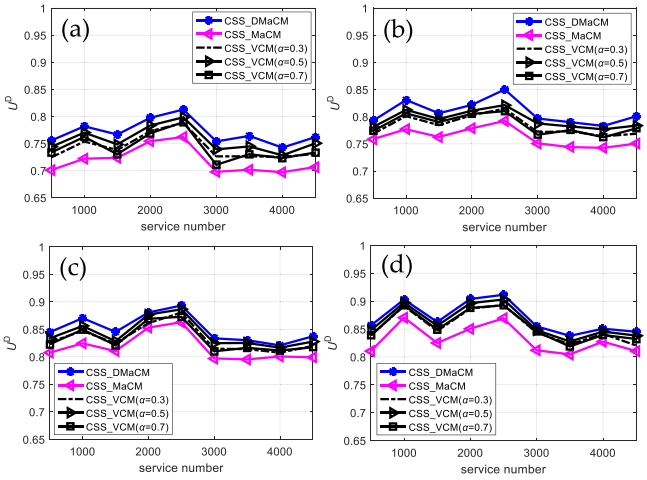

Fig. 6. U^{D} in different matrix density. (a)60%;(b)70%;(c)80%;(d)90%.

extent. However, obviously, the improvements are not significant when $K > 10$. The result is consistent with the previous research [40].

7.4 Accuracy Analysis of CS Selection

First, to illustrate the effectiveness of variation-aware CS selection, a case is analyzed based on Example 2. The details are provided in Appendix A.9 available in the online supplemental material. Next, we employ U^{D} to assess the accuracy of CS selection approaches.

7.4.1 Compare Variation-Aware CSS-CFT Approaches

The proposed variation-aware CS selection approach using the DMaCM similarity-based collaborative filtering, noted as CSS_DMaCM, is compared with the following two approaches: (a) CSS_MaCM, the variation-aware CS selection approach using the MaCM similarity-based collaborative filtering and (b) CSS_VCM, the variation-aware CS selection approach using the VCM similarity-based collaborative filtering. The two approaches also employ QoSCMs to model the variations of QoS. However, they use different methods to measure the similarity of QoSCMs and utilize the improved TOPSIS method based on the Mahalanobis distance to select the optimal CS.

Every experiment consists of 9 batches, in which 500 services are used in order. The services #1-#500 are used in the first batch; the services #501-#1000 are used in the second batch; and the services #4,001-#4,500 are used in the last batch. Every batch is executed repeatedly for 50 times and the average U^{D} values are recorded. In each batch, u_{o} is randomly selected from all 142 users involved in the dataset, and 8 services are chosen randomly as the candidate CSs. The first 60 timeslots are divided into 6 periods, and the weight of each period or each QoS parameter is regulated within the interval $[0.0, 1.0]$ for simulating the variation in user requirements. The top 10 most similar users are chosen (i.e., $K = 10$). These CS selection approaches are executed when the matrix density of the original QoS data is set as from 60 to 90 percent, respectively. The U^{D} values are obtained as shown in Fig. 6.

The results are analyzed as follows:

(a) The U^{D} values of CSS_VCM are obviously affected by α. CSS_VCM gets the better UD values in the case when $\alpha = 0.5$ than in other cases. Besides, as mentioned before, both VCM method and MaCM method are unable to precisely measure the similarity differences between QoSCMs, and inevitably produces errors in the process of CS selection. Thus, the accuracy of both CSS_VCM and CSS_MaCM is limited.

(b) CSS_DMaCM obtains the highest accuracy of service selection. On the one hand, CSS_DMaCM, using the DMaCM method to measure the similarity of QoSCMs, is capable of precisely distinguishing the slight differences between time series QoS data from two users when certain periods are checked, especially, in the application scenarios with the dramatic variation of QoS.

(c) The experimental results also demonstrate that the CV of the original data and the matrix density are closely related to the U^{D} values. The more greatly the performance fluctuations, the larger the CV is. The great uncertainty of performance seriously affects the similarity measurement, which subsequently disrupts the accuracy of CS selection approaches. For example, the largest U^{D} value is obtained when services #2000-#2500 are used in the experiments. The reason lies in that the CV values of the throughput data and the response time data related to services #2000-#2500 are smaller than other service set according to Appendix Fig. A.2 available in the online supplemental material. In addition, the low matrix density leads to the decreasing U^{D} values. This fact is verified in Fig. 6.

7.4.2 Compare other CSS-CFT Approaches

To further verify the advantages of CSS_DMaCM, we compare it with four CSS-CFT approaches from the related work, namely, SSPDR-I [18], RecINF [19], the time-aware service recommendation approach [24], called TaSRec, and time-aware service ranking approach [10], called TaSRank. Considering that no neighboring user identification method is given in SSPDR-I, to fairly compare the CS selection results with other approaches, the proposed DMaCM method is used to find the neighboring users for SSPDR-I. Besides, RecINF identifies the neighboring users according to the user features (e.g., the geographical and network locations). Due to the absence of user features information in WS-Dream dataset #2, we also employ the DMaCM method to identify the neighboring users for RecINF. In the following experiment, the matrix density varies from 60 to 90 percent, and other setups are same to the previous experiments. The results are shown in Fig. 7.

The results are analyzed as follows:

(a) From Fig. 7, the U^{D} values of SSPDR-I and TaSRec are lower than those of the other three approaches. The main reasons are as follows: (i) SSPDR-I exploits the probability distribution of QoS data to infer an INT representing the QoS variation and employs a possibility degree ranking method to select the optimal CS. However, the INT only captures a feature of QoS variation, namely, the variation range of QoS, which is not enough for CS selection in a high-variance cloud environment. (ii) TaSRec employs a time-aware PCC measurement to calculate user similarity, and ranks all

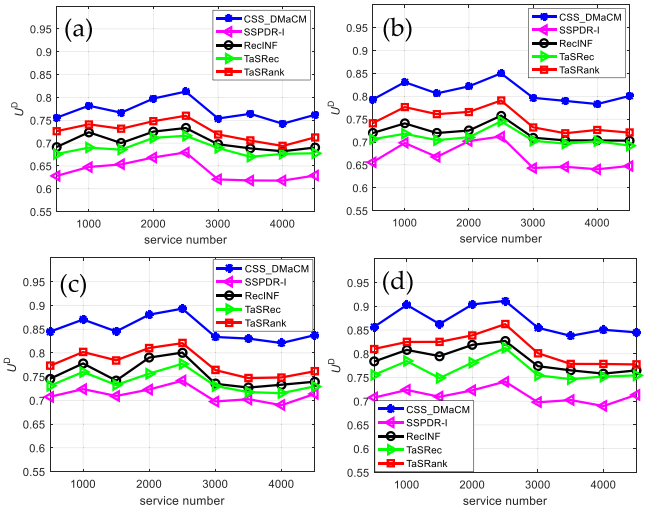

Fig. 7. U^{D} in different matrix density. (a)60%;(b)70%;(c)80%;(d)90%.

services by virtue of the user-based and service-based predictions. In TaSRec, the QoS data in the recent period have a greater contribution to the user similarity measurement than early period. Thus, TaSRec mainly focuses on the variation of QoS in the future period, and is limited to support CS selection meeting the user preferences for different periods. Besides, the PCC-based measurement method ignores the latent variation relations among the time series QoS data, difficult to choose the exact neighboring users in a high-variance environment

(b) RecINF obtains the larger U^{D} values than SSPDR-I and TaSRec. In RecINF, the traditional INT is extended to INF by defining an eigenvalue interval and INF is used to model the QoS variation, depicting the central tendency and variation range of QoS. Thus, RecINF is enable to improve the accuracy compared to SSPDR-I.

(c) TaSRank achieves a better accuracy than SSPDR-I and TaSRec. The reason lies in that TaSRank utilizes the interval neutrosophic set (INS) theory to assess the trustworthiness of CSs in multiple periods, capturing the QoS variation from two aspects (i.e., variation range and period). Although INS can effectively support the trustworthy CS selection with tradeoffs between performance-costs, INS cannot represent the central tendency and variation frequency of QoS. This causes the limited accuracy of TaSRank in a high-variance cloud environment.

(d) CSS_DMaCM gets the largest U^{D} values. This approach covers QoS variation in four aspects instead of one or two with respect to the other approaches. Moreover, it focuses on the time series QoS data in every period of the overall timeslots. The user similarity is separately measured in every period, and the data from every period is viewed as a whole to identify the implicit variation feature of QoS. Thus, it can reduce the errors of user similarity caused by the direct calculations based on the one-to-one sample matching. For example, in Table 3, we might firmly believe that the QoS values of u_1 in t_{11} and t_{12} are completely different with the values of u_3 from the perspective of an individual timeslot; however, we will come to a diametrically opposite conclusion when t_{11} and t_{12} are observed in a period together. In addition, CSS_DMaCM could take full advantage of the abnormal data for recognizing the variation of QoS, while other approaches fail to do this. These abnormal data may become the significant feature information to generate a more exact cloud model by the entropy and hyper entropy, which conforms to the real QoS situations of CSs.

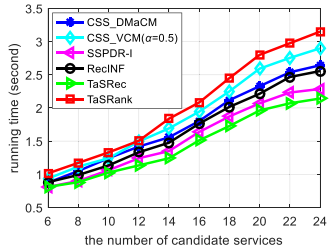

Fig. 8. Performance analysis.

7.5 Execution Time Analysis

Next, we compared the execution time of six approaches (i.e., CSS_DMaCM, CSS_VCM, SSPDR-I, RecINF, TaSRec, TaSRank). The experiments are executed in MATLAB 2016 via Dell notebook with Intel i7-6500U processor and 8G memory. The matrix density is 80 percent. Based on our survey, suppose that the number of training CSs is 10 and the number of candidate services ranges from 6 to 24. other setups are same to the previous experiment. The results are shown in Fig. 8.

The results display that the proposed approach can improve the accuracy of service selection without appreciably degrading performance. The computation process of SSPDR-I and TaSRec is simpler than other approaches, and requires less execution time. RecINF is slightly better than CSS_DMaCM because the calculations related to the multiple periods are not involved in it. Considering that the KRCC calculation [10] used in TaSRank involves the identification of concordant/discordant pairs between any two CSs and the VCM calculation [8] in CSS_VCM involves the integration of orientation similarity and dimension similarity, they are more complex and need more execution time than CSS_DMaCM. CSS_DMaCM obtains the better performance than TaSRank and CSS_VCM especially with the increasing number of candidate services because the optimized function from MATLAB is available to directly calculate the Mahalanobis distance. Considering that only top 10 most similar users are chosen and only 64 timeslots are involved in the dataset, the complexity of CSS_DMaCM $\mathrm{O}(Y \times X \times N)$ becomes linear, just shown in Fig. 8. In addition, from Fig. 8, the consuming time of CSS_DMaCM is only about 2.6 seconds even if the number of candidate services is up to 24. Considering the limited amount of functionally-equivalent services that might exist in the respective service repository, thus, the consuming time should be affordable to the users in order to obtain a more accurate service selection result.

7.6 Discussion

The experiments demonstrated the advantages of proposed approach compared to other approach. However, our work

still needs more improvements from the following aspects: (1) The implementation of CS recommender system has been not completed, and some components (e.g., CS discovery center, interactive interfaces) are being developed. (2) The cold start problem is still a critical challenge for variation-aware CS selection in which the time series data is required to accurately model the variation characteristics of CSs' QoS. (3) The proposed approach focuses on selecting an optimal CS. In fact, it is rational to help the current user acquire a right ranking list of all candidate CSs to cater the specific requirements in some cases that the top-most CS is not desirable for some reasons or even it has become unavailable or obsolete. (4) Although a real dataset is used in the experiments, the user features information is missing and only two QoS parameters are involved in it. Thus, we have to use the simulation method to randomly select the services and users. This is limited to seek more valuable results. Besides, probably it is a feasible idea to improve the CS selection by analyzing the variation features of QoS parameters from the domain-independent part and domain-specific part when the time series data of multiple QoS parameters are available.

8 Conclusion and Future Work

Aiming at the QoS variation in the dynamic cloud environment, this paper proposes a variation-aware CS selection approach via collaborative QoS prediction. We study the variation characteristics of QoS from the four aspects, including central tendency, variation range, frequency of variation and period, and utilize cloud model theory to derive a set of QoS cloud models that map to multiple aspects of QoS variation based on time series data. A neighboring user identification method based on the double Mahalanobis distance is presented to support the QoS prediction via collaborative filtering. To select an appropriate CS with optimal QoS in accordance to the user preferences, the variation-aware CS selection is formulated as an MCDM problem. An improved TOPSIS method is used to solve it by utilizing the Mahalanobis distance-based similarity measurement.

The case analysis and the experiments based on a real-world dataset reveal that compared to other approaches, the proposed approach can enhance the accuracy of CSS-CFT without noticeably increased selection time. This approach contributes to CS selection in a high-variance cloud environment, especially when users have different preferences over different time periods.

There are some topics that were not well discussed in this paper, which represent directions that we will focus on in the future. These topics include: (1) the cloud-based implementation of a complete CS recommender system providing a configurable module which is able to deliver service discovery functionality and an interactive visualization module which could enable the user to provide the right user input; (2) the realization of CS discovery through employing Semantic Web techniques that can enable the functional matchmaking of services in the most accurate manner; (3) the more effective cold start mechanism to deal with the new users problem by using the transfer learning techniques to extract the implicit user features (e.g., network location, geographical location, period preferences); (4) the variation-aware CSs ranking approach that can return an ordered list of services by analyzing the relative dominances between candidate CSs based on the time series QoS data; (5) the global CS selection for service composition by employing the big data analysis and machine learning method to predict the dynamic demands of users for CSs' resources in data-intensive applications.

Acknowledgments

Our deepest gratitude goes to the anonymous reviewers for their careful work and thoughtful suggestions that have helped us improve this paper. This work was supported by MOE (Ministry of Education in China) Humanities and Social Science Research Youth Fund Project (No. 18YJCZH124), Hunan Provincial Natural Science Foundation of China (No. 2017JJ2186) and National Natural Science Foundation of China (No. 61572525).

References

[1] P. Casas and R. Schatz, "Quality of experience in cloud services: Survey and measurements," *Comput. Netw.*, vol. 68, no. 1, pp. 149–165, 2014.
[2] L. Sun, H. Dong, F. K. Hussain, O. K. Hussain, and E. Chang, "Cloud service selection: State-of-the-art and future research directions," *J. Netw. Comput. Appl.*, vol. 45, no. 10, pp. 134–150, 2014.
[3] X. Zheng, L. Da Xu, and S. Chai, "QoS recommendation in cloud services," *IEEE Access*, vol. 5, pp. 5171–5177, 2017.
[4] Z. Zheng, H. Ma, M. R. Lyu, and I. King,"QoS-aware web service recommendation by collaborative filtering," *IEEE Trans. Services Comput.*, vol. 4, no. 2, pp. 140–152, Apr.-Jun. 2011.
[5] D. Rosaci and G. M. Sarné, "Recommending multimedia web services in a multi-device environment," *Inf. Syst.*, vol. 38, no. 2, pp. 198–212, 2013.
[6] J. Yin, W. Lo, S. Deng, Y. Li, Z. Wu, and N. Xiong, "Colbar: A collaborative location-based regularization framework for QoS prediction," *Inf. Sci.*, vol. 265, pp. 68–84, 2014.
[7] H. Ma, Z. Hu, L. Yang, and T. Song, "User feature-aware trustworthiness measurement of cloud services via evidence synthesis for potential users," *J. Visual Languages Comput.*, vol. 25, no. 6, pp. 791–799, 2014.
[8] H. Ma, H. Zhu, Z. Hu, W. Tang, and P. Dong, "Multi-valued collaborative QoS prediction for cloud service via time series analysis," *Future Generation Comput. Syst.*, vol. 68, no. 3, pp. 275–288, 2017.
[9] H. Ma, Z. Hu, K. Li, and H. Zhang, "Toward trustworthy cloud service selection: A time-aware approach using interval neutrosophic set," *J. Parallel Distrib. Comput.*, vol. 96, pp. 75–94, 2016.
[10] H. Ma, H. Zhu, Z. Hu, K. Li, and W. Tang, "Time-aware trustworthiness ranking prediction for cloud services using interval neutrosophic set and ELECTRE," *Knowl.-Based Syst.*, vol. 138, pp. 27–45, 2017.
[11] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for web services," in *Proc. IEEE 22nd Int. Symp. Softw. Rel. Eng.*, 2011, pp. 210–219.
[12] C. Yu and L. Huang, "Time-aware collaborative filtering for QoS-based service recommendation," in *Proc. IEEE Int. Conf. Web Serv.*, 2014, pp. 265–272.
[13] H. Wang, C. Yu, L. Wang, and Q. Yu, "Effective BigData-space service selection over trust and heterogeneous QoS preferences," *IEEE Trans. Serv. Comput.*, vol. 11, no. 4, pp. 644–657, Jul./Aug. 2018.
[14] S. Al-Shammari and A. Al-Yasiri, "MonSLAR: A middleware for monitoring SLA for RESTFUL services in cloud computing," in *Proc. 9th Int. Symp. Maintenance Evol. Serv.-Oriented Cloud-Based Environ.*, 2015, pp. 46–50.
[15] S. Anithakumari and K. Chandrasekaran, "Monitoring and management of service level agreements in cloud computing," in *Proc. Int. Conf. Cloud Autonomic Comput.*, 2015, pp. 204–207.
[16] H. J. Syed, A. Gani, R. W. Ahmad, M. K. Khan, and A. I. A. Ahmed, "Cloud monitoring: A review, taxonomy, and open research issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 11–26, 2017.

[17] M. Mehdi, N. Bouguila, and J. Bentahar, "Probabilistic approach for QoS-aware recommender system for trustworthy web service selection," *Appl. Intell.*, vol. 41, no. 2, pp. 503–524, 2014.
[18] H. Ma, Z. Hu, and M. Cai, "Trustworthy service selection integrating cloud model and possibility degree ranking of interval numbers," *Chin. J. Electron.*, vol. 26, no. 6, pp. 1177–1183, 2017.
[19] H. Ma and Z. Hu, "Recommend trustworthy services using interval numbers of four parameters via cloud model for potential users," *Frontiers Comput. Sci.*, vol. 9, no. 6, pp. 887–903, 2015.
[20] L. Sun, J. Ma, Y. Zhang, H. Dong, and F. K. Hussain, "Cloud-FuSeR: Fuzzy ontology and MCDM based cloud service selection," *Future Generation Comput. Syst.*, vol. 57, pp. 42–55, 2016.
[21] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang,"QoS ranking prediction for cloud services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1213–1222, Jun. 2013.
[22] C. Mao, J. Chen, D. Towey, J. Chen, and X. Xie, "Search-based QoS ranking prediction for web services in cloud environments," *Future Generation Comput. Syst.*, vol. 50, pp. 111–126, 2015.
[23] Z. Ye, S. Mistry, A. Bouguettaya, and H. Dong, "Long-term QoS-aware cloud service composition using multivariate time series analysis," *IEEE Trans. Serv. Comput.*, vol. 9, no. 3, pp. 382–393, May/Jun. 2016.
[24] Y. Hu, Q. Peng, and X. Hu, "A time-aware and data sparsity tolerant approach for web service recommendation," in *Proc. IEEE Int. Conf. Web Serv.*, 2014, pp. 33–40.
[25] Y. Ma, S. G. Wang, P. C. K. Hung, C.-H. Hsu, Q. B. Sun, and F. C. yang, "A highly accurate prediction algorithm for unknown web service QoS values," *IEEE Trans. Serv. Comput.*, vol. 9, no. 4, pp. 511–523, Jul./Aug. 2016.
[26] G. Wang, C. Xu, and D. Li, "Generic normal cloud model," *Inf. Sci.*, vol. 280, pp. 1–15, 2014.
[27] S. H. Zyoud and D. Fuchs-Hanusch, "A bibliometric-based survey on AHP and TOPSIS techniques," *Expert Syst. Appl.*, vol. 78, pp. 158–181, 2017.
[28] Y. Zhong, Y. Fan, K. Huang, W. Tan, and J. Zhang, "Time-aware service recommendation for mashup creation," *IEEE Trans. Serv. Comput.*, vol. 8, no. 3, pp. 356–368, 2015.
[29] M. Mehdi, E. Epaillard, N. Bouguila, and J. Bentahar, "Modeling and forecasting time series of compositional data: A generalized dirichlet power steady model," in *Proc. 9th Int. Conf. Scalable Uncertainty Manage.*, 2015, pp. 170–185.
[30] D. Li, H. Meng, and X. Shi, "Membership clouds and membership cloud generators," *J. Comput. Res. Develop.*, vol. 32, no. 6, pp. 15–20, 1995.
[31] H. Peng, H. Zhang, and J. Wang, "Cloud decision support model for selecting hotels on TripAdvisor.com with probabilistic linguistic information," *Int. J. Hospitality Manage.*, vol. 68, pp. 124–138, 2018.
[32] Y. Zhang, D. Zhao, and D. Li, "The similar cloud and the measurement method," *Inf. Control*, vol. 33, no. 2, pp. 129–132, 2004.
[33] G. Zhang, D. Li, P. Li, J. Kang, and G. Chen, "A collaborative filtering recommendation algorithm based on cloud model," *J. Softw.*, vol. 18, no. 10, pp. 2403–2411, 2007.
[34] H. Li, C. Guo, and W. Qiu, "Similarity measurement between normal cloud models," *Acta Electronica Sinica*, vol. 39, no. 11, pp. 2561–2567, 2011.
[35] L. Zhang, Y. Yang, and X. Zhao, "SaaS decision-making method based on cloud model," *Acta Electronica Sinica*, vol. 43, no. 5, pp. 987–992, 2015.
[36] C. Liu, M. Feng, X. Dai, and D. Li, "A new algorithm of backward cloud," *J. Syst. Simul.*, vol. 16, no. 11, pp. 2417–2420, 2004.
[37] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.
[38] F. Guo, W. Susilo, and Y. Mu, "Distance-based encryption: How to embed fuzziness in biometric-based encryption," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 247–257, Feb. 2016.
[39] L. Mikhailov and P. Tsvetinov, "Evaluation of services using a fuzzy analytic hierarchy process," *Appl. Soft Comput.*, vol. 5, no. 1, pp. 23–33, 2004.
[40] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.
[41] A. M. Alkalbani and F. K. Hussain, "A comparative study and future research directions in cloud service discovery," in *Proc. IEEE 11th Conf. Ind. Electron. Appl.*, 2016, pp. 1049–1056.
[42] R. Lourenzutti and R. A. Krohling, "A generalized TOPSIS method for group decision making with heterogeneous information in a dynamic environment," *Inf. Sci.*, vol. 330, pp. 1–18, 2016.
[43] A. Vega, J. Aguarón, J. Garcíaalcaraz, and J. M. Morenojiménez, "Notes on dependent attributes in TOPSIS," *Procedia Comput. Sci.*, vol. 31, pp. 308–317, 2014.
[44] X. Wang and L. Wang, "Applications of TOPSIS improved based on mahalanobis distance in supplier selection," *Control Decision*, vol. 27, no. 10, pp. 1566–1570, 2012.

Hua Ma received the BS, MS and PhD degrees all from Central South University, Changsha, China. His research interests focus on cloud computing and service computing. He is currently an associate professor and master supervisor with the College of Information Science and Engineering, Hunan Normal University, Changsha, China.

Zhigang Hu received the BS, MS and PhD degrees all from Central South University, Changsha, China. His research interests focus on high performance computing and cloud computing. He is currently a professor and PhD supervisor of Central South University, Changsha, China.

Keqin Li is a SUNY distinguished professor of computer science in the State University of New York. His current research interests include high-performance computing, cloud computing, big data computing, and so on. He has published more than 600 journal articles, book chapters, and refereed conference papers. He is currently serving or has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*. He is a fellow of the IEEE.

Haibin Zhu is a full professor of the Department of Computer Science and Mathematics, Nipissing University, Canada. He has more than 160 research publications. He is a senior member of IEEE and is serving and served as co-chair of the technical committee of Distributed Intelligent Systems of IEEE SMC Society, and associate editor of *IEEE Transactions on Systems, Man and Cybernetics (SMC): Systems*.