Temporal Pattern Based QoS Prediction

Liang Chen[1](✉), Haochao Ying[2], Qibo Qiu[2], Jian Wu[2], Hai Dong[1],
and Athman Bouguettaya[1]

[1] School of Computer Science and Information Technology,
RMIT, Melbourne, Australia
jasonclx@gmail.com, {hai.dong,athman.bouguettaya}@rmit.edu.au

[2] College of Computer Science and Technology, Zhejiang University,
Hangzhou, China
{haochaoying,vincent2014,wujian2000}@zju.edu.cn

Abstract. Quality-of-Service (QoS) is critical for selecting the optimal Web service from a set of functionally equivalent service candidates. Since QoS performance of Web services are unfixed and highly related to the service status and network environments which are variable against time, it is critical to obtain the missing QoS values of candidate services at given time intervals. In this paper, we propose a temporal pattern based QoS prediction approach to address this challenge. Clustering approach is utilized to find the temporal patterns based on services QoS curves over time series, and polynomial fitting function is employed to predict the missing QoS values at given time intervals. Furthermore, a data smoothing process is employed to improve prediction accuracy. Comprehensive experiments based on a real world QoS dataset demonstrate the effectiveness of the proposed prediction approach.

Keywords: Service Computing · QoS prediction · Temporal pattern

1 Introduction

A Service-Oriented Computing (SOC) paradigm and its realization through standardized Web service technologies provide a promising solution to the seamless integration of single-function applications to create new large-grained and value-added services. Web services are software systems designed to support interoperable machine-to-machine interaction over a network. Typically, a service-oriented application consists of multiple Web services interacting with each other in several tiers.

Quality of Service (QoS) has been widely employed for evaluating the non-functional characteristics of Web services [16]. With the explosive growth of functionality-equal services, non-functional characteristic of Web service is becoming a popular research concern and kinds of QoS-based approaches were proposed in various of Service Computing areas, such as service composition [1,2], fault-tolerant web services [5], and service selection [4,18].

© Springer International Publishing AG 2016

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 223–237, 2016.
DOI: 10.1007/978-3-319-48743-4_18

A common premise of previous research is that the values of QoS properties are already known and fixed. However, user-dependent QoS values always vary over time in the real-world scenario. Figure 1(a)[1] shows the variation curve of one service's response time (response time is one important QoS property) when continually invoked by the same user along 64 time intervals. It could be found that the response time varies largely from 1 s to 20 s. Actually, the QoS performance of Web services observed from the users perspective is usually quite different from that declared by the service providers in Service Level Agreement (SLA), due to the following reasons [17]:

- QoS performance of Web services is highly related to invocation time, since the service status (e.g., workload, number of clients, etc.) and the network environment (e.g., congestion, etc.) change over time.
- Service users are typically distributed in different geographical locations. The user-observed QoS performance of Web services is greatly influenced by the Internet connections between users and Web services. Different users may observe quite different QoS performance when invoking the same Web service.

Based on above reasons, it is becoming essential to collect time-aware QoS information of Web services for QoS-based Service Computing research issues. However, in reality, a service user usually only invokes a limited number of Web services, thus the QoS values of the other Web services are missing (unknown) for the target user. Without sufficient time-aware QoS information, the accuracy of QoS-based research work, i.e., QoS-based service selection, QoS-base service composition, could not be guaranteed. Therefore, it is becoming urgent to build a time-aware QoS prediction approach for efficiently estimating missing QoS values of Web services for target users.

In this paper, we propose to address the problem of time-aware QoS prediction by exploring the advantages of temporal patterns. Temporal patterns and related techniques have been used and demonstrated in social media area to solve the problems such as video popularity prediction in Youtube [12], retweet number prediction in Twitter [15], etc. An intuitive idea is that the influences of factors (i.e., network environment, location, etc.) behind the QoS temporal variation could be reflected in the uncovered patterns, and the missing values in each QoS carve could be predicted by using the most similar temporal patter to fit for. Particularly, a curve clustering approach is proposed to uncover QoS temporal patterns, and polynomial fitting function is employed to predict the missing QoS values. Moreover, A curve smoothing approach is employed to improve prediction accuracy, due to the noises in QoS curves. Experiments based on 20+ million service invocation records demonstrate the effectiveness of the proposed prediction approach.

In summary, this paper makes the following contributions:

1. We formally identify the critical problem of time-aware Web service QoS prediction and propose the concept of temporal pattern in this research area. Particularly, temporal patterns are extracted from QoS curves over time series.

[1] Due to the space limitation, Fig. 1 is placed in Page 5.

2. We propose a novel Temporal Pattern based QoS Prediction approach TPP, which utilizes temporal patterns to predict the missing QoS values via polynomial fitting. Moreover, a data smoothing process is employed to improve the prediction accuracy. We consider TPP as the first temporal pattern based QoS prediction approach.
3. Comprehensive experiments based on a real world Web service QoS dataset are implemented to evaluate the performances of TPP and other state-of-the-art approaches. Compared with other approaches, TPP achieves 35.8 %∼52.0 % improvement in terms of MRE metric.

The rest of this paper is organized as follows. Section 2 highlights the related work of QoS prediction. Section 3 formally define the problem and introduces the details of data smoothing, pattern clustering, and the prediction algorithm. Experimental results and analysis are presented in Sect. 4, whereas Sect. 5 concludes this paper.

2 Related Work

Quality of Service (QoS) has been widely employed for evaluating the non-functional characteristics of Web services [16]. Among QoS properties, values of server-side QoS (e.g., price, popularity) are identical for different users while others (e.g., response time, throughput) observed from the user-side may change over time due to the unpredictable network conditions and heterogeneous user environments [8]. With the explosive growth of functionality-equal services, non-functional characteristic of Web service is becoming a popular research topic and kinds of QoS-based approaches are proposed in various of Service Computing areas, such as service composition service composition [1] fault-tolerant web services [5] and service selection [4].

A common premise of previous research is that the values of user-dependent QoS properties are already known. However, in reality a user typically has engaged a limited number of Web services in the past and cannot exhaustively invoke all the available candidate services. Thus, it is fundamental to predict the missing QoS values for any QoS-based Service Computing research.

In web service QoS prediction, Collaborative filtering approaches have been widely adopted. Generally, traditional recommendation approaches could be classified into two categories: memory-based [13,19] and model-based [3]. Memory-based approaches, also known as neighborhood-based approaches, are one of the most popular prediction methods in collaborative filtering systems. Shao et al. [11] first use collaborative filtering approach to predict QoS values from similar users. Zheng et al. [20] propose a hybrid user-based and item-based approach to predict QoS values for the current user by employing historical web service QoS data from other similar users and similar web services. Although memory-based algorithms implement easily, high computation complexity makes it difficult to deal with a large and sparse time-aware dataset. Model-based algorithms employ statistical and machine learning techniques to learn a sophisticated model based on history QoS invocation records, including clustering models [14], latent semantic models [6], latent factor models [9], etc. Zheng et al. use PMF algorithm to predict missing failure probability values in user-service matrix [19], and propose NIMF to improve prediction accuracy by balancing the global information and local information [21]. Compared with memory-based approaches, model-based QoS prediction approaches usually have better performance but lack of interpretation.

Time is an important context factor which affects QoS prediction accuracy, since service status (e.g. number of clients and workload) and network environments (e.g. congestion) change over time. QoS values will fluctuate when the same user invoke the same service at different time interval. Limited QoS prediction works consider the influence of time to QoS values. Hu et al. propose a time-aware similarity model which considers two aspects: (1) More temporally close QoS experience from two users on a same service contributes more to the user similarity measurement; (2) More recent QoS experience from two users on a same service contributes more to the user similarity measurement [7]. Zhang et al. construct a three dimensional matrix by adding time factor, and then employ tensor factorization to extract user-specific, service-specific, and time-specific latent features from historical QoS values for prediction [17]. In this paper, we take advantage of model-based concept and propose a temporal pattern based approach with better interpretability. In this paper, we analyze a set of 430,000 response-time curves, each curve means one user invokes one service at 64 continuous time intervals. The surprising thing is that temporal patterns of QoS values could be accurately represented by using limited number of curves. Moreover, a data smoothing process is employed to improve the performance of QoS prediction.

3 QoS Prediction Based on Temporal Patterns

In this section, we first formally define the problem and analyze the research challenges in Sect. 3.1, and then introduce the details of corresponding solutions in Sects. 3.2 and 3.3, respectively. Finally, QoS prediction algorithm is presented in Sect. 3.4.

3.1 Problem Definition and Research Challenges

In previous works, most of QoS prediction approaches origin from recommender system. Concretely, they predict the missing QoS values in the user-service or user-service-time QoS matrix, which is generated from historical service invocation by users [11,17,22]. Unlike above works, we propose a novel method to predict QoS value based on temporal patterns in this paper.

Let U be the set of m users, S be the set of n Web services, and T be the set of c time intervals. From the collection of QoS attribute from user-side, the observed QoS value of user i invoking service j at time interval t_k can be formally represented by q_{ijk}, where $i \in 1, ..., m$, $j \in 1, ..., n$, $k \in 1, ..., c$ and q_{ijk} is one of QoS attributes (e.g., response time or throughput). For convenience, the length of time internals is fixed. For example, the real-world dataset employed in this paper is over 64 consecutive time slices at 15 min interval. Intuitively, the shape of q_{ij} measures how user i invokes service j changed over time. In practice, each user typically uses a few of services so that we can get the set H of complete curves which we know all QoS values of user invoked service at each time interval. However, component service can be replaced automatically in service-oriented architecture (SOA). Therefore, the records of user invoked service at some intervals may be missed, which formally represented by the set Δ. Our goal is to use the complete curves in H to predict the missing value in Δ.

In the scenario of QoS prediction, there are two challenges to efficiently predict the missing value. First, due to the influence of dynamic network conditions and varying server loads, the QoS value at each interval fluctuates quickly and may exist noise. If we directly use the original data, the performance of prediction may reduce. Secondly, To predict the missing value, the naive method is to compare the curve with missing value with each complete curve and then use the most similar curve to predict the missing value. However, although each user invokes a few services, millions of complete curves may be collected if we have large number of users and services. Therefore, this approach is time-consuming and not efficiently.

3.2 Data Smoothing

To deal with the first challenge, we design a data transformation method for QoS data to reduce noises. Figure 1 presents an example of a complete QoS (i.e., response time) curve of one user invoked a service. It is obvious that the curve is too diverse to directly compare with others by using distance measure. Fortunately, We can also observe that the QoS value at time t is close to the value at the previous $(t-1)$ and forward $(t+1)$ time slice. It is intuitive that

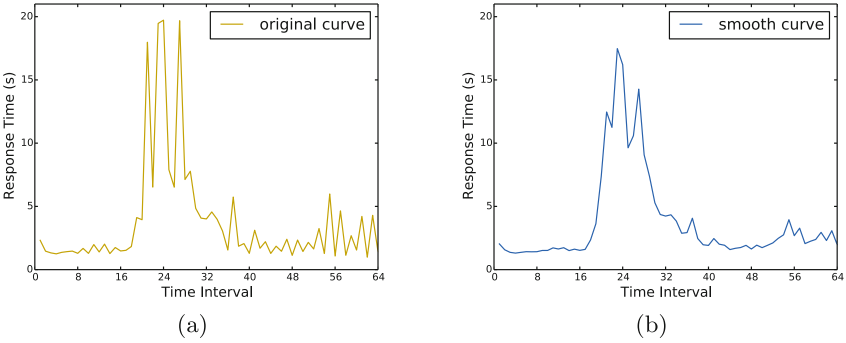


Fig. 1. An example of curves smooth in response time. (a) is the original curve and QoS values fluctuate sharply. (b) is the smoothed curve. To some extent this curve reduces noise and keep overall changing shape of QoS values.

the value in continue intervals should be similar. Based on this observation, the data transformation is defined as follows:

$$q_{ijk} = \begin{cases} \dfrac{2q_{ijk} + q_{ijk+1}}{3} & k = 1 \\ \dfrac{2q_{ijk} + q_{ijk-1}}{3} & k = c \\ \dfrac{q_{ijk-1} + 2q_{ijk} + q_{ijk+1}}{4} & \text{otherwise} \end{cases} \tag{1}$$

Our data smoothing method takes more weights to current observed QoS values and simultaneously consider QoS values in adjacent time. From Fig. 1(b), we can find that the smoothed curve retains the changing shape of QoS values and reduces noise to some extent.

3.3 Temporal Pattern Generation

To deal with the second challenge, we employ K-Means clustering algorithm to find the clusters of QoS curves that share distinct temporal pattern. The reason that we choose K-Means algorithm is its simpleness and efficiency.

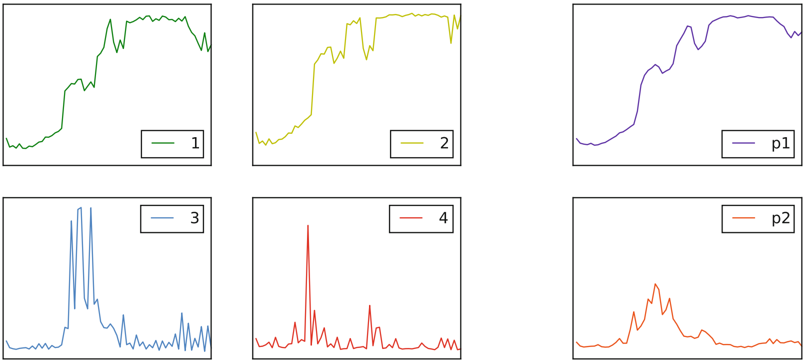
Given the set H of complete QoS curves and the number of clusters K, our goal is to find an assignment set C_k of curves for each cluster, and the centroid u^k of each cluster minimizes the following function:

$$F = \sum_{k=1}^{K} \sum_{q_{ij} \in C_k} d(q_{ij,} u^k) \tag{2}$$

where $d(q_{ij,} u^k) = \sum_{t=1}^{c} (q_{ijt}, u_t^k)^2$ is the square of Euclidean distance. We start the K-Means algorithm with random initial K centroids. As an iterative refinement algorithm, K-Means proceeds by alternating between two steps: assignment step and update step. In the assignment step, we assigns each curve to the cluster with the closest centroid based on $d(q_{ij,} u^k)$. After finding the new assignment set C_k for each curve, we calculate the new centroid for each C_k in the update step, according the average of all curves in C_k. Formally, the updated centroid should be as follows:

$$u^k = \frac{1}{|C_k|} \sum_{q_{ij} \in C_k} q_{ij} \tag{3}$$

After updating many times, the algorithm will converge when the assignment no longer changes. Finally, the centroid of each cluster represents the temporal pattern. Figure 2 presents an example for clustering four original curves. It is obvious that the two temporal patterns catch the most important characters in each cluster. In the next section, we will use these temporal patterns to predict the missing QoS values.



(a)

(b)

Fig. 2. An example of curves clustering. (a) is the four original curves of response time. After smoothing and clustering, (b) shows the centroids of two clusters.

3.4 QoS Prediction

After smoothing and clustering, we get K temporal patterns. Suppose that we have a curve q_{ij} with some QoS values at the corresponding time intervals missing. For simplify, q_{ij} misses the value in interval t. Now, the question is how to use these pattern to predict the missing QoS value q_{ijt}.

First, we compute the distance between the observed values of q_{ij} and each pattern in corresponding time interval under different metric. In the experiments, we compare three distance approaches (i.e., cosine, euclidean, and cityblock) and choose the best metric to measure the distance. After this step, the most similar pattern p can be obtained based on the distance. An intuitive way is to directly use the value of p in interval t to predict the q_{ijt}. However, it is unwise because the pattern p can not match q_{ij} completely and we can not eliminate the fixed distance in interval t. In this paper, we use a function to map the pattern p to the curve q_{ij}. In general, the map function is polynomial fitting function as follows:

$$\hat{q}_{ij} = map(p) = w_0 + w_1 p + w_2 p^2 + \cdots + w_d p^d \tag{4}$$

where w is the weights and d is the order.

After finding the order and weights of polynomial based on sum of least square between q_{ij} and $\hat{q}_{ij}$ in the observed values, the predicted value q_{ijt} could be obtained through $map(p_t)$. Note that we just compare the linear and square fitting in the experiments to avoid overfitting. The pseudo code of our algorithm for QoS prediction is provided in Algorithm 1.

4 Experiments

In this section, comprehensive experiments are implemented to evaluate the proposed approach based on a real-world dataset. Experimental evaluation will

Algorithm 1. Our QoS Prediction Algorithm.

```
Input : The set H of complete QoS curves; The number of clusters $K$; The set
        $\Delta$ of incomplete QoS curves
Output: The QoS prediction of unobserved value in $\Delta$
for $q \in$ H do
    smooth q by Equation 1;
random initial $K$ centroids $u^1, u^2, ..., u^K$;
repeat
    set each cluster $C_1, .., C_K$ to null;
    for $j = 1$ to $|\mathrm{H}|$ do
        $k \longleftarrow argmin_{k=1,..,K} d(p_j, u^k)$;
        $C_k \longleftarrow C_k \cup j$;
    for $i = 1$ to $K$ do
        $u_i \longleftarrow \frac{1}{|C_k|} \sum_{q_{ij} \in C_k} q_{ij}$;
until centroids converge;
for $q \in \Delta$ do
    find the most similar pattern $p$ based on observed value in $q$;
    polynomial fit $q \longleftarrow map(p)$;
    predict unobserved value of $q$ in each interval $t$ by $map(p_t)$;
```

answer the following questions: (1) What are the evaluation metrics? (2) How does our approach compare with other state-of-the-art ones? (3) What is the impact of data smoothing, similarity approach, and the order of polynomial fit?

4.1 Data Preprocessing

In the experiments, we mainly focus on Response Time (RT), one of the most important QoS properties, to evaluate QoS prediction methods. Response time (RT) is the length of time between the end of an inquiry on a computer system and the beginning of a response. All experiments are implemented in a machine with a 2.2 GHz Intel CPU and 16 GB RAM, running OS X Yosemite.

For the sake of application in practice, all experiments are implemented based on a public real-world Web service QoS dataset which is collected by 142 users invoking 4532 web services in 16 hours with a time interval of 15 min [17]. In particular, the users are 142 computers of PlanetLab[2] located in 22 countries, and the services are 4532 public available real world web services distributed in 57 countries. Through the observation, we find quite a lot of noises exist in the dataset. For example, the response time value will be set to −1, if the response time is over 20 s in this invocation. Furthermore, some Web services have not been invoked by any user. Thus, we do some data cleaning work on this dataset, and macroscopic statistics & data distribution of the generated dataset

[2] PlanetLab is a global research network that supports the development of new network services. Details could be found in https://www.planet-lab.org/.

are presented in Figs. 3 and 4, respectively. It could be found the experimental evaluation utilizes more than 20 million records, which partly demonstrate reliability and scalability of the experiments. It should be noted that the proposed approach could be utilized for the prediction of any other QoS property (e.g., throughput), even though only response time is studied in this paper.

Statistics	Values
#Users	135
#Services	3952
#Time slices	64
#Time interval	15min
#Records	20,138,880
RT scale	$(0, 20)$
RT mean	0.8442

Fig. 3. Statistics of QoS dataset

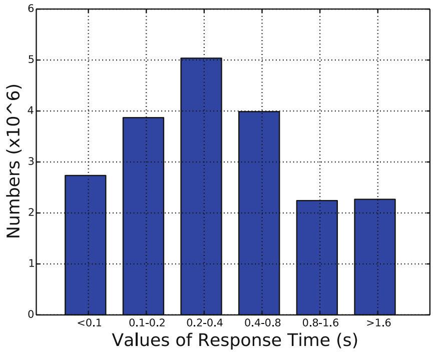


Fig. 4. RT value distribution

4.2 Evaluation Metric

We evaluate the prediction accuracy of our proposed approach in comparison with other existing methods by using the following metrics.

- **MAE** (Mean Absolute Error). MAE is average prediction accuracy between prediction results and corresponding observations, which is defined as follows:

$$MAE = \frac{\sum_{i,j} \left| \hat{R}_{ij} - R_{ij} \right|}{N} \tag{5}$$

  where R_{ij} denotes the real QoS value of service j observed by user i, $\hat{R}_{ij}$ is the predicted QoS value by a method, and N is the total number of predicted values.
- **NMAE** (Normalized Mean Absolute Error). NMAE normalizes the differences range of MAE by computing:

$$NMAE = \frac{MAE}{\sum_{ij} R_{ij}/N} \tag{6}$$

- **MRE** (Median Relative Error). MRE measures the median value of relative errors between observed value and predicted value:

$$MRE = median \left| \hat{R}_{ij} - R_{ij} \right| / R_{ij} \tag{7}$$

Due to the large variance of QoS values, we focus more on relative error metric, i.e., MRE, which is more appropriate for QoS prediction evaluation. Since many papers use MAE and NMAE, they are also included for comparison purpose.

4.3 Performance Comparisons

In order to show the effectiveness of our proposed QoS prediction approach, we compare the prediction accuracy of the following methods:

- **UPCC**: This method employs the information of similar users (measured by Pearson Correlation Coefficient) to predict the QoS values [3].
- **IPCC**: This method is widely-used in recommendation system, which employs the similarity between services for QoS prediction [10].
- **UIPCC**: This method combines UPCC and IPCC model, which fully uses the similarity of users and services [20].
- **PMF**: This is a classic matrix factorization method, which has been employed in [19]. User-service matrix is factorized into two matrices under low-rank assumption and then using the matrices predict QoS values.
- **WSPred**: This is a tensor factorization-based prediction method with average QoS value constraint [17].

In the experiments, user-service records are randomly divided into two parts: 80 % records as the training data and the rest 20 % as the testing data. In order to evaluate the performance of different approaches in reality, we randomly choose $\frac{m}{16}$ $(m = 1, 2, 3, 4, 5, 6, 7, 8)$ of the training data for pattern clustering, and the others (i.e., $\frac{16-m}{16}$ of the training data) for cross validation. Equation (1) is employed for data smoothing, and Eqs. (2) and (3) are employed for the pattern clustering. Through the observation of experimental results, we find that the proposed approach could get similar temporal patterns in any density setup. That is, the proposed pattern clustering approach is quite stable and even $\frac{1}{16}$ of training data is enough to get appropriate patterns. Polynomial fit is employed for QoS prediction, once temporal patters are generated. Since a user usually only invokes a small number of services, the testing matrix density is randomly thinned to the same $\frac{m}{16}$. The prediction accuracy is evaluated by comparing the original value and the predicted value of each removed entry in testing matrix. Without lost of generality, the number of patterns is set as 4 in this paper. Detailed impact of data smoothing, similarity approach, and polynomial order is studied in Sects. 4.4, 4.5, and 4.6, respectively.

The QoS value prediction accuracies evaluated by MAE, NAME, and MRE are shows in Table 1. For each row in the table, we highlight the best performer among all methods. As we can observe, our approach significantly outperforms the other ones over MRE, while still achieving best results on MAE and NMAE. Concretely, our approach achieves 35.8 %∼52.0 % improvement on MRE, 1.8 %∼2.7 % improvement on MAE, and 2.0 %∼3.0 % improvement on NMAE at different matrix densities. Note that all improvements are computed as the percentage of how much our approach outperforms the other most competitive approach.

Table 1. Comparison of performance (a smaller value means a better performance)

Method	Density = 2/16			Density = 3/16			Density = 4/16		
	MAE	NMAE	MRE	MAE	NMAE	MRE	MAE	NMAE	MRE
UPCC	0.5226	0.6211	0.5334	0.492	0.5845	0.477	0.4745	0.5637	0.4497
IPCC	0.5946	0.7066	0.6671	0.5675	0.6741	0.6395	0.5376	0.6386	0.5992
UIPCC	0.5215	0.6197	0.5225	0.4912	0.5835	0.473	0.4719	0.5606	0.4467
PMF	0.5219	0.6208	0.4764	0.4925	0.5855	0.4496	0.4765	0.5659	0.4327
WSPred	0.4583	0.5445	0.4519	0.4358	0.5168	0.4293	0.4253	0.504	0.4112
TPP	**0.4501**	**0.532**	**0.2167**	**0.4253**	**0.5025**	**0.2249**	**0.4138**	**0.4888**	**0.2308**
Improve. (%)	1.8 %	2.3 %	52.0 %	2.4 %	2.8 %	47.6 %	2.7 %	3.0 %	43.9 %
Method	Density = 5/16			Density = 6/16			Density = 7/16		
	MAE	NMAE	MRE	MAE	NMAE	MRE	MAE	NMAE	MRE
UPCC	0.462	0.549	0.4323	0.4517	0.5368	0.4185	0.4435	0.5272	0.4069
IPCC	0.5204	0.6184	0.5776	0.5071	0.6029	0.5606	0.4954	0.5891	0.5453
UIPCC	0.4588	0.5452	0.4307	0.4482	0.5327	0.4182	0.4394	0.5223	0.4072
PMF	0.4633	0.55	0.4262	0.4536	0.5386	0.4231	0.4444	0.5277	0.408
WSPred	0.4148	0.4913	0.3895	0.4125	0.4884	0.3894	0.4084	0.4834	0.3814
TPP	**0.4075**	**0.4817**	**0.2375**	**0.4026**	**0.4756**	**0.2419**	**0.3985**	**0.4709**	**0.2448**
Improve. (%)	1.8 %	2.0 %	39.0 %	2.4 %	2.6 %	37.9 %	2.4 %	2.6 %	35.8 %

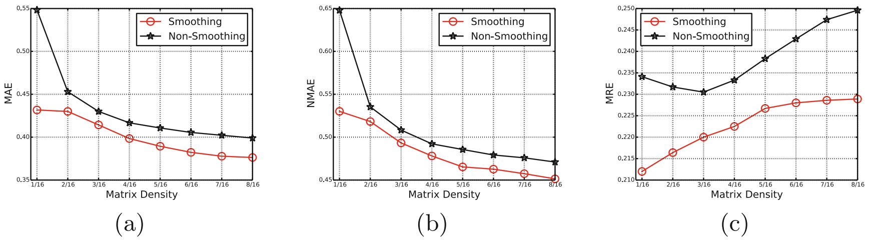
We also find that although UIPCC achieves higher accuracy than UPCC and IPCC over MAE and NMAE, and WSPred achieves better performance compared with the first three Collaborative Filtering based approaches (i.e., UPCC, IPCC, and UIPCC) and PMF, all these approaches have large errors over MRE. Thus, only focusing on minimizing the absolute error may lead to large relative error, which is not suitable for QoS prediction problem.

4.4 Impact of Data Smoothing

Data smoothing process is employed to reduce noises in QoS curves for the purpose of improving prediction accuracy, and is one of main contributions in this paper. In order to study its impact, we implement two versions of our proposed approach: one with the proposed data smooth process, i.e., Eq. (1), and the other without it. Figure 5 shows the prediction accuracy comparison between the above two versions. From Fig. 5, We can observe that the version with data smoothing largely outperforms the other version in terms of MAE, NMAE, and MRE. This is because the remove of noise points in QoS curves facilitates the generation of temporal patterns. In short, The process smooths out data fluctuations and improves QoS prediction accuracy.

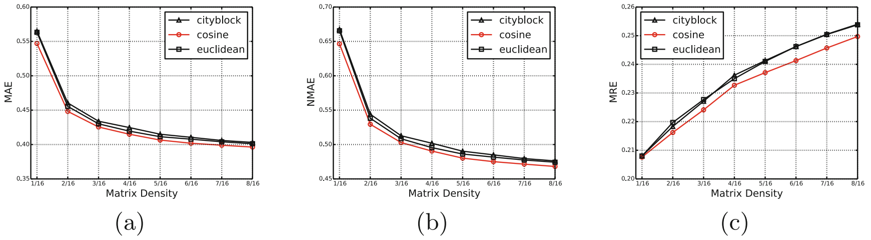
4.5 Impact of Similarity Approach

In the process of the proposed TPP approach, we have to choose the most similar pattern for the target QoS curve for predicting the missing values in



(a) (b) (c)

Fig. 5. Impact of data smoothing



(a) (b) (c)

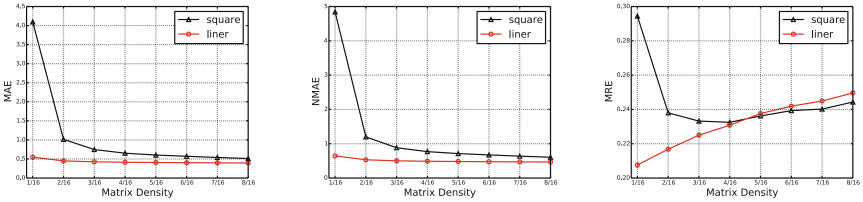
Fig. 6. Impact of similariy approach

the curve. Thus, the choice of similarity measure approach is very important for the final prediction accuracy. In the experiments, we employ three widely accepted approaches to compute the similarity between pattern and the choose data points in testing data. Specifically, the three similarity measure approaches are cosine, euclidean, and cityblock.

To present a comprehensive evaluation of these approaches, we vary the matrix density from 1/16 to 8/16. Other parameter settings are #pattern = 4, order of polynomial = 1. Figure 6 shows the performance comparison of different similarity approaches in terms of MAE, NAME, and MRE. From Fig. 6, we can find cosine similarity method always outperforms the other methods over three metrics when the data density varies from 1/16 to 8/16. This observation demonstrates that cosin similarity measurement is more suitable for computing similarity between curves. Furthermore, we can also observe that as the density increases, every similarity approach can achieve better prediction results in terms of absolute error metrics, i.e., MAE and NMAE. This is because more data points provided in testing data, more information could be gained for prediction. However, it is not suitable for the trend of relative error, i.e., MRE.

4.6 Impact of Order of Polynomial Fit

Once the optimal pattern is selected, polynomial fitting function is employed to predict the missing QoS values in testing data. In this section, we evaluate the impact of different polynomial fitting functions, that is, order of polynomial fit. For simplicity, we only compare the performance of QoS prediction when order



(a) (b) (c)

Fig. 7. Impact of order of polynomial fit

is 1 (liner) and 2 (square), since the trend could be easily illustrated by this comparison. Other parameter settings are #pattern = 4, similarity = cosine.

Figure 7 shows the prediction accuracy comparison of linear and square polynomial fit with the increase of density. For Fig. 7, we observe that the linear polynomial fit outperforms the square in most cases. As the increase of density, the prediction accuracy of square polynomial fit improves (the MAE and NMAE decreases) due to more information provided. However, compared with square one, it could be observed that linear polynomial fit is quite stable with the increase of density. That means, linear polynomial fit approach is very suitable for the case of cold-start and data sparsity, that is, online QoS prediction.

Further, we can find the prediction accuracy of square polynomial fit is quite bad when the density is 1/16, which means this sparsity condition causes an overfitting problem. From another perspective, the performance gap of linear and square polynomial fit decreases with the increase of matrix density. That means the overfitting phenomenon alleviates with more provided information. In all, linear polynomial fit is quite suitable for our problem.

5 Conclusion

With the explosive growth of functionality-equal services, non-functional characteristic of Web service is becoming a popular research concern and kinds of QoS-based approaches were proposed in various of Service Computing research areas. Since QoS performance of Web services are unfixed and highly related to the service status and network environments which are variable against time, it is critical to obtain the missing QoS values of candidate services at given time intervals. In this paper, we propose a temporal pattern based QoS prediction approach to address this challenge. Clustering approach is utilized to find the temporal patterns based on services QoS curves over time series, and polynomial fitting function is employed to predict the missing QoS values at given time intervals. Furthermore, a data smoothing process is employed to improve prediction accuracy. Comprehensive experiments based on a real world QoS dataset demonstrate the effectiveness of the proposed prediction approach.

For future work, we will investigate more techniques to improve the performance of temporal pattern generation and QoS prediction. Particularly, QoS curve shifting and scaling techniques will be introduced for better pattern generation, and machine learning techniques will be utilized to predict the missing QoS values based on the generated temporal patterns. Further, the datasets of other QoS properties (e.g., throughput) will also be employed to evaluate the performance of the proposed approach.

Acknowledgement. This research was made possible by NPRP 9-224-1-049 grant from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors. This research was partially supported by the Natural Science Foundation of China under grant of No. 61379119, Science and Technology Program of Zhejiang Province under grant of No. 2013C01073, the Open Project of Qihoo360 under grant of No. 15-124002-002.

References

1. Alrifai, M., Risse, T.: Combining global optimization with local selection for efficient QoS-aware service composition. In: Proceedings of the 18th International Conference on World Wide Web, pp. 881–890. ACM (2009)
2. Alrifai, M., Risse, T., Nejdl, W.: A hybrid approach for efficient web service composition with end-to-end QoS constraints. ACM Trans. Web (TWEB) **6**(2), 7 (2012)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, pp. 43–52. Morgan Kaufmann Publishers Inc. (1998)
4. Chen, L., Kuang, L., Wu, J.: Mapreduce based skyline services selection for QoS-aware composition. In: 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), pp. 2035–2042. IEEE (2012)
5. Fang, C.L., Liang, D., Lin, F., Lin, C.C.: Fault tolerant web services. J. Syst. Archit. **53**(1), 21–38 (2007)
6. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. (TOIS) **22**(1), 89–115 (2004)
7. Hu, Y., Peng, Q., Hu, X.: A time-aware and data sparsity tolerant approach for web service recommendation. In: 2014 IEEE 21th International Conference on Web Services (ICWS), pp. 33–40. IEEE (2014)
8. Menasce, D.: QoS issues in web services. IEEE Internet Comput. **6**(6), 72–75 (2002)
9. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2007)
10. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186. ACM (1994)
11. Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H.: Personalized QoS prediction for web services via collaborative filtering. In: IEEE International Conference on Web Services, ICWS 2007, pp. 439–446. IEEE (2007)
12. Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commun. ACM **53**(8), 80–88 (2010)
13. Wu, J., Chen, L., Feng, Y., Zheng, Z., Zhou, M.C., Wu, Z.: Predicting quality of service for selection by neighborhood-based collaborative filtering. IEEE Trans. Syst. Man Cybern.: Syst. **43**(2), 428–439 (2013)

14. Xue, G.R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y., Chen, Z.: Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 114–121. ACM (2005)
15. Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 177–186. ACM (2011)
16. Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for web services composition. IEEE Trans. Softw. Eng. **30**(5), 311–327 (2004)
17. Zhang, Y., Zheng, Z., Lyu, M.R.: WSPred: a time-aware personalized QoS prediction framework for web services. In: 2011 IEEE 22nd International Symposium on Software Reliability Engineering (ISSRE), pp. 210–219. IEEE (2011)
18. Zhao, L., Ren, Y., Li, M., Sakurai, K.: Flexible service selection with user-specific QoS support in service-oriented architecture. J. Netw. Comput. Appl. **35**(3), 962–973 (2012)
19. Zheng, Z., Lyu, M.R.: Personalized reliability prediction of web services. ACM Trans. Softw. Eng. Methodol. (TOSEM) **22**(2), 12 (2013)
20. Zheng, Z., Ma, H., Lyu, M.R., King, I.: QoS-aware web service recommendation by collaborative filtering. IEEE Trans. Serv. Comput. **4**(2), 140–152 (2011)
21. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Collaborative web service QoS prediction via neighborhood integrated matrix factorization. IEEE Trans. Serv. Comput. **6**(3), 289–299 (2013)
22. Zhu, J., He, P., Zheng, Z., Lyu, M.R.: Towards online, accurate, and scalable QoS prediction for runtime service adaptation. In: 2014 IEEE 34th International Conference on Distributed Computing Systems (ICDCS), pp. 318–327. IEEE (2014)