Temporal-Aware QoS Prediction via Dynamic Graph Neural Collaborative Learning

Shengxiang Hu[1], Guobing Zou[1(✉)], Bofeng Zhang[2,3], Shaogang Wu[1], Shiyi Lin[1], Yanglan Gan[4], and Yixin Chen[5]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai, China
{shengxianghu,gbzou}@shu.edu.cn
[2] School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China
bfzhang@sspu.edu.cn
[3] School of Computer Science and Technology, Kashi University, Xinjiang, China
[4] School of Computer Science and Technology, Donghua University, Shanghai, China
ylgan@dhu.edu.cn
[5] Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA
chen@cse.wustl.edu

Abstract. How to effectively predict missing QoS has become a fundamental research issue for service-oriented downstream tasks. However, most QoS prediction approaches omit high-order implicit invocation correlations and collaborative relationships among users and services. Thus, they are incapable of effectively learning the temporally evolutionary characteristics of user-service invocations from historical QoS records, which significantly affects the performance of QoS prediction. To address the issue, we propose a novel framework for temporal-aware QoS prediction by dynamic graph neural collaborative learning. Dynamic user-service invocation graph and graph convolutional network are combined to model user-service historical temporal interactions and extract latent features of users and services at each time slice, while a multi-layer GRU is applied for mining temporal feature evolution pattern across multiple time slices, leading to temporal-aware QoS prediction. The experimental results indicate that our proposed approach for temporal-aware QoS prediction significantly outperforms state-of-the-art competing methods.

Keywords: Web service · Temporal-aware QoS prediction · Dynamic user-service invocation graph · Graph convolutional network · Latent feature extraction

1 Introduction

With the rapid advancements of Internet technology, service-oriented architecture (SOA) has been widely used in real-world applications. As one of the

J. Troya et al. (Eds.): ICSOC 2022, LNCS 13740, pp. 125–133, 2022.
https://doi.org/10.1007/978-3-031-20984-0_8

key implementation techniques of SOA, web services have extremely promoted interoperatable machine-to-machine interactions. However, many services supply users with analogous functionalities. Quality of Service (QoS) [11] is applied to represent the non-functional characteristics of web services and differentiate those functionally equivalent ones. Because of the enormous number of users and services, it is impractical and time-consuming for users to invoke all web services and record the corresponding QoS values in the constantly changing network environment. Thus, it is of vital importance to precisely perform temporal-aware QoS prediction, which has become a challenging issue due to the sparsity of historical user-service invocations across multiple time slices in real scenarios.

Some recent investigations concentrate on collaborative filtering (CF) and neural network-based approaches for temporal-aware QoS prediction. They generally compose a sequence of QoS invocation matrices from different consecutive time slices, and extract the features of users and services at each time slice, then apply deep learning techniques, such as gate recurrent unit (GRU) [3] and long short-term memory (LSTM) [7], to learn the evolution pattern of QoS across multiple time slices. However, they mainly characterize a user in terms of those directly invoked services or a service in terms of those users who have directly invoked the service, without the consideration of high-order implicit invocation correlations between users and services through indirect interactions as well as the high-order collaborative relationships between similar users or services. Due to the lack of the extraction of high-order latent features that are hidden in the user-service interactions, it is still difficult in effectively encoding latent features of users and services, yielding to low accuracy of temporal-aware QoS prediction.

To address the issues, inspired by the developments of graph and Graph Convolutional Networks (GCNs) [2], we propose a novel framework for temporal-aware QoS prediction by dynamic graph neural collaborative learning. First, we formulate user-service historical QoS interactions as a temporal-aware service ecosystem, which is transformed into a dynamic user-service invocation graph across multiple time slices. Then, a GCN-based [2] graph neural collaborative feature extractor is learned to extract high-order latent features of users and services at each time slice, taking into account both indirect user-service invocation correlations and collaborative relationships by similar users or services. Finally, a multi-layer GRU [3] is applied for mining temporal feature evolution pattern across multiple time slices, leading to temporal-aware QoS prediction. To evaluate the effectiveness of our proposed approach for temporal-aware QoS prediction, extensive experiments are conducted on a large-scale real-world dataset. By comparing with several state-of-the-art baselines, experimental results demonstrate that our proposed approach receives the best prediction performance in multiple evaluation metrics. The main contributions of this paper are summarized as follows:

– We propose a novel dynamic graph neural collaborative learning framework for temporal-aware QoS prediction. It can more effectively reveal user-service invocation features at each time slice and mine temporal feature evolution pattern across multiple time slices for better QoS prediction.

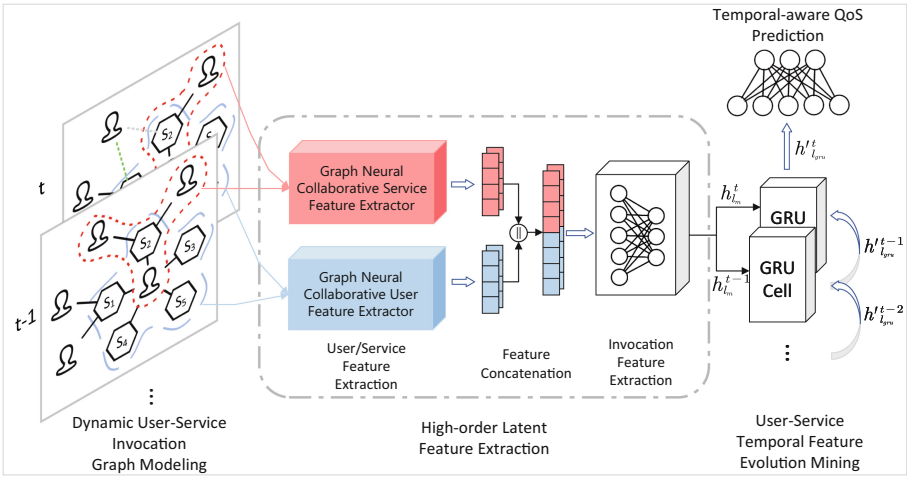

Fig. 1. The overall framework of our proposed approach.

- We propose a novel approach for extracting high-order latent features of a user and service by dynamic user-service invocation graph modeling and graph convolutional network learning. Compared to the existing approaches, the advantage is that we can more deeply reveal the latent features of users and services, with the consideration of both high-order user-service invocation correlations and collaborative relationships by similar neighborhoods.
- Extensive experiments are conducted on a large-scale real-world QoS dataset, and the results indicate that our approach receives superior performance for temporal-aware QoS prediction compared with baseline approaches.

The remainder of this paper is structured as follows. Section 2 elaborates the proposed approach. Section 3 shows and analyzes experimental results. Finally, Sect. 4 concludes the paper and discusses future work.

2 Approach

The overall framework of our proposed approach is illustrated in Fig. 1. It mainly consists of four stages, including dynamic user-service invocation graph modeling, high-order latent feature extraction, user-service temporal feature evolution mining, and temporal-aware QoS prediction.

2.1 Dynamic User-Service Invocation Graph Modeling

A temporal-aware service ecosystem can be formulated as $\xi =< U, S, T, R >$, where there are n users $U = \{u_i\}_{i=1}^{n}$, m web services $S = \{s_i\}_{i=1}^{m}$, t time

slices $T = \{1, 2, \ldots, t\}$, and a sequence of corresponding historical QoS matrix $R = \{R^i \in \Re^{n \times m}\}_{i=1}^{t}$, $r_{ij}^t \in R^t$ indicates the corresponding QoS value when a user $u_i \in U$ invokes a service $s_j \in S$ at time t. To model the high-order implicit invocation correlations and collaborative relationships among users and services, we transform ξ into a dynamic user-service invocation graph $\mathcal{G} = \{\mathcal{G}^i\}_{i=1}^{t}$. Each snapshot $\mathcal{G}^t =< V_u, V_s, E^t, W^t >$ is transformed from $\xi^t =< U, S, t, R^t >$ at time slice t. Here, $V_u = \{u_i\}_{i=1}^{n}$ is a set of n user vertices; $V_s = \{s_i\}_{i=1}^{m}$ is a set of m service vertices; E^t is a set of edges that represents user-service invocation relationships. If $r_{ij}^t \in R^t$, there exists an edge $e_{ij}^t = e_{ji}^t \in E^t$ between $u_i \in V_u$ and $s_j \in V_s$; W^t is a set of edge weights. If $e_{ij}^t \in E^t$, there exists a corresponding weight $w_{ij}^t \in W^t$, which can be converted from $r_{ij}^t \in R^t$.

The edge weight $w^t \in W^t$ measures the strength of the connection, i.e. the invocation relationship, between a user vertex and a service vertex at time slice t. Generally, a lower value implies a higher QoS under a negative QoS criteria, such as response time. It is observed that most of real QoS values are clustered around a certain value for a QoS criterion, but there are also a small number of outliers that may influence model training deviating from expectations. In order to ensure robustness of our proposed model, we further convert the original QoS value r_{ij}^t to a normalized range as the corresponding edge weight w_{ij}^t. By taking into account both the distribution characteristics of QoS values and practical observations, a heuristic conversion function is designed to project r_{ij}^t to w_{ij}^t under a negative QoS criterion. It is expressed as follows:

$$w_{ij}^t = \begin{cases} \frac{exp(r_{ij}^t) - exp(-1/r_{ij}^t)}{exp(1/r_{ij}^t) + exp(-1/r_{ij}^t)} & \text{if } r_{ij}^t > 1 \\ \frac{1}{exp(r_{ij}^t)} - \frac{1}{e} + \frac{exp(2)-1}{exp(2)+1} & otherwise \end{cases} \tag{1}$$

where w_{ij}^t denotes the associated weight for edge $e_{ij}^t \in E^t$. By using the conversion function, we project all of the QoS values to their corresponding edge weights for each time slice $t \in T$. Thus, the dynamic user-service invocation graph $\mathcal{G}$ can be generated, which is used to extract high-order latent features of users and services at each time slice.

2.2 High-Order Latent Feature Extraction of Users and Services

Based on $\mathcal{G}$, we extract the high-order latent feature of a target user u and service s at each time slice. We initially represent u and s with a randomized feature vector $x_u \in \Re^d$ and $x_s \in \Re^d$, respectively, where d specifies the dimension of the feature vector. It is intuitive that a user's feature can be partially reflected by the directly invoked services and indirectly characterized by the non-adjacent user and service neighbors. It can be performed by a multi-layer recursive way in a user-service invocation graph $\mathcal{G}^t$ at each time slice t. Analogously, we can also extract a service's latent feature with the consideration of user-service invocation correlations and collaborative relationships among services.

Here, we leverage the GCN's [2] message passing mechanism to capture high-order latent features of users (services) along the structure of $\mathcal{G}^t$. The procedure

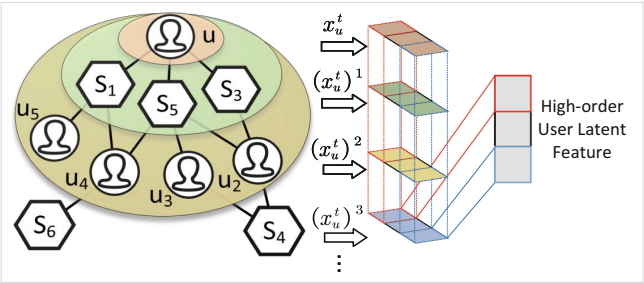

Fig. 2. High-order user latent feature extraction by graph neural collaborative feature extractor.

of high-order user latent feature extraction by graph neural collaborative feature extractor is illustrated in Fig. 2. which applies a recursive way of message propagation and aggregation. More specifically, $\mathcal{N}_u^t \subseteq V_s$ denotes the set of adjacent service vertices that are directly connected to u in $\mathcal{G}^t$, i.e., the first-hop service neighbors of u at time slice t. In such case, for each service $s' \in \mathcal{N}_u^t$, the message $m_{u \leftarrow s'}^t$ propagated from s' to u is calculated as follows:

$$m_{u \leftarrow s'}^t = \frac{exp(w_{us'}^t)}{\sum_{i \in \mathcal{N}_u^t} exp(w_{ui}^t)} W^1 x_{s'} \tag{2}$$

where $W^1 \in \Re^{d \times d}$ is a trainable weight matrix, and $w_{us'}^t$ denotes the weight associated with edge $e_{us'}^t$. With a larger $w_{us'}^t$, more messages are retained and s' contributes more to u's high-order latent feature. Following that, we aggregate messages from all of the u's first-hop neighbors in message aggregation:

$$x_u^t = x_u \tag{3}$$

$$(x_u^t)^1 = \alpha(x_u^t + \sum_{s' \in \mathcal{N}_u^t} m_{u \leftarrow s'}^t) \tag{4}$$

where $(x_u^t)^1$ signifies the representation of u that aggregates first-order messages, which implies the behavioral features embodied by the directly invoked services, α is the activation function. By stacking l_{gcn} message-passing procedures, we can aggregate messages from l_{gcn}-hop user and service neighbors, leading to the high-order connectivity characteristics of u. These heuristic information can strengthen the feature representation of a user by the latent invocation correlations between u and non-invoked services, as well as the latent collaborative relationships of the user neighbors who are structurally nearby vertices of u. The recursive aggregation of user representation can be expressed as:

$$(m_{u \leftarrow s'}^t)^{l_{gcn}-1} = \frac{exp(w_{us'}^t)}{\sum_{i \in \mathcal{N}_u^t} exp(w_{ui}^t)} W^{l_{gcn}} (x_{s'}^t)^{l_{gcn}-1} \tag{5}$$

$$(x_u^t)^{l_{gcn}} = \alpha(x_u^t + \sum_{s' \in \mathcal{N}_u^t} (m_{u \leftarrow s'}^t)^{l_{gcn}-1}) \tag{6}$$

where $W^{l_{gcn}}$ is the trainable weight for the l_{gcn}-th layer message propagation.

Through l_{gcn}-layers message passing, we obtain a series of user representations $x_u^t, (x_u^t)^1, \ldots, (x_u^t)^{l_{gcn}}$, which aggregates the user-service invocation correlations and collaborative relationships of users or services among different hops around the center of u. They are fused by a one-dimensional convolution layer to generate the high-order latent feature of u as follows:

$$(x_u^t)_i^* = \sum_{j=0}^{l_{gcn}} \omega_j (X_u^t)_{i,j},\ i \in [0, d) \tag{7}$$

where $(x_u^t)^* \in \Re^d$ is the extracted high-order latent feature of u, $\omega \in \Re^{l_{gcn}+1}$ denotes the convolution kernel, $X_u^t \in \Re^{d \times (l_{gcn}+1)}$ is the matrix of combining $(l_{gcn}+1)$ user representations x_u^t, $(x_u^t)^1$, $\ldots$, $(x_u^t)^{l_{gcn}}$. It is important to note that the procedure for extracting the high-order latent feature $(x_s^t)^*$ of a target service s is identical to the one of u.

Based on the high-order latent features of $(x_u^t)^*$ and $(x_s^t)^*$, they are concatenated as a whole that is fed into a l_m-layer multi-layer perceptron (MLP) to obtain the invocation feature $h_{l_m}^t$ of u and s at time slice t. Consequently, $h_{l_m}^t$ is used for mining temporal feature evolution between u and s.

2.3 User-Service Temporal Feature Evolution Mining

To reveal the evolution pattern of the user-service invocation features across multiple time slices, we mine the hidden temporal nonlinear relationship by a multi-layer GRU [3]. Given a set of extracted invocation features $H_k = \{h_{l_m}^{t-k+1}, h_{l_m}^{t-k+2}, \ldots, h_{l_m}^t\}$ of a current u and a target service s across k consecutive time slices, the hidden state of GRU layer can be calculated as follows:

$$z^t = \sigma(W_z \cdot [{h'}^{t-1} || h_{l_m}^t]) \tag{8}$$

$$r^t = \sigma(W_r \cdot [{h'}^{t-1} || h_{l_m}^t]) \tag{9}$$

$$\hat{s}^t = \tanh(W \cdot [(r^t \odot {h'}^{t-1}) || h_{l_m}^t]) \tag{10}$$

$${h'}^t = (1 - z^t) \odot {h'}^{t-1} + z^t \odot \hat{s}^t \tag{11}$$

where W_z, W_r, W are the trainable weight matrices, d' is the dimension of the GRU layer's output, and $\odot$ represents element-wise product. Due to traditional GRU is a shallow model with limited capacity to extract deep implicit features, we stack l_{gru} GRU layers. The hidden output of last GRU layer ${h'}_{l_{gru}}^t \in \Re^{d'}$ is used as the evolutionary invocation feature for temporal-aware QoS prediction.

2.4 Temporal-Aware QoS Prediction

Based on the evolutionary invocation feature of a current user u and target service s, we can predict the missing QoS $\hat{r}_{us}^{t+1}$ at time slice $t+1$, by a fully-connected neural network. The output layer is calculated as:

$$\hat{r}_{us}^{t+1} = ReLU(W_o {h'}_{l_{gru}}^t + b_o) \tag{12}$$

where W_o is a trainable weight matrix, b_o is a offset item, and $\hat{r}_{us}^{t+1}$ is the predicted QoS when a current user u invokes a target service s at time slice $t+1$. To train and optimize the model parameters, we take Mean Square Error as the loss that is defined as:

$$Loss = \frac{\sum_{u \in U} \sum_{s \in S} (\hat{r}_{us}^{t+1} - r_{us}^{t+1})^2}{n \times m} + \lambda \left\| \Theta \right\|_2^2 \tag{13}$$

where U, S represent the user and service set, respectively, and $|U| = n, |S| = m$. Θ is all the trainable parameters of our proposed model, λ controls the $L2$ regularization strength to prevent overfitting. We adopt mini-batch AdamW [4] to update and optimize the parameters.

Table 1. Results of temporal-aware QoS prediction among competing approaches.

Density	MAE				RMSE			
	0.05	0.1	0.15	0.2	0.05	0.1	0.15	0.2
UPCC	0.946	1.209	1.107	1.006	1.908	1.778	1.720	1.683
IPCC	1.135	1.041	0.994	0.989	2.255	1.867	1.795	1.795
WSRec	0.807	0.578	0.967	0.758	1.917	1.328	2.407	1.733
WSPred	0.781	0.689	0.673	0.663	1.707	1.633	1.608	1.593
PNCF	1.165	1.089	1.043	1.013	1.836	1.722	1.653	1.617
RNCF	1.048	1.010	0.974	0.958	1.616	1.546	1.503	1.470
TUIPCC	0.731	0.576	0.819	0.697	1.776	1.207	2.059	1.635
Ours	**0.574**	**0.526**	**0.489**	**0.462**	**1.284**	**1.193**	**1.158**	**1.123**
Gains	**22.5%**	**8.7%**	**27.4%**	**30.4%**	**20.6%**	**1.2%**	**23.6%**	**23.5%**

3 Experiments

3.1 Dataset

To validate the effectiveness of the proposed approach, we conduct extensive experiments on a large-scale real-world web service QoS dataset called WS-DREAM[1], which has been widely used in service computing for QoS prediction. WS-DREAM employed 142 distributed PlanetLab computers (i.e. users) located across 22 countries, to monitor a total of 4,500 publicly accessible real-world web services from 57 countries continuously in 64 different time slices at 15-minute interval. And a total of 27,392,643 detailed response-time values ranging from 0 s to 20 s are collected as the sub-dataset *rtdata* [11], on which our experiments are extensively conducted to demonstrate the superiority performance of the proposed temporal-aware QoS prediction approach. The overall data sparsity is approximately 66.98%.

[1] http://wsdream.github.io/dataset.

3.2 Experimental Results and Analyses

We evaluate the temporal-aware QoS prediction results by two widely adopted evaluation metrics: MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error). In addition, we compare our proposed approach with 7 state-of-the-art methods: UPCC [8], IPCC [5], WSRec [10], WSPred [9], PNCF [1], RNCF [3] and TUIPCC [6]. To thoroughly validate the effectiveness of our proposed approach for temporal-aware QoS prediction, we conduct extensive experiments on temporal QoS dataset with four different densities: 5%, 10%, 15%, and 20%, and report the MAE and RMSE, respectively. For all baseline approaches, we follow the optimal parameter settings specified in the corresponding papers.

The results are summarized in Table 1, with the best performance among baseline approaches highlighted in dark and the overall best results bolded. It is obvious from the results that our proposed approach outperforms all of the competing approaches at different QoS densities, with the relative improvements ranging from 8.7% to 30.4% on MAE and 1.2% to 23.6% on RMSE, respectively. In terms of MAE, TUIPCC receives superior performance among baseline approaches at QoS densities of 0.05 and 0.1, whereas WSPred achieves the best among baseline approaches at QoS densities of 0.15 and 0.2. As for RMSE, RNCF is better than the other baseline approaches for the densities of 0.05, 0.1, and 0.2, respectively. As can be seen from the above results, the baseline approaches suffer from instability for QoS prediction at different densities. For example, while TUIPCC achieves a lower MAE, it cannot perform very well on RMSE, indicating that it is unable to fit certain outliers when predicting the missing QoS. Therefore, compared to the baseline approaches, our proposed prediction model consistently achieves the lowest MAE and RMSE across all different QoS densities, revealing that it can predict QoS values more precisely with better robustness.

It concludes that two aspects may potentially contribute to the best performance of our proposed approach. First, an optimized dynamic neural graph collaborative learning model is designed to encode the high-order latent features of users and services, that overcomes the constraint of sparse historical QoS invocations across multiple time slices, leading to more precisely user-service invocation feature. Second, a multi-layer GRU is applied to boost the accuracy of QoS prediction by effectively mining the implicit temporal evolution patterns of user-service invocation features across multiple time slices.

4 Conclusion and Future Work

This paper proposes a novel framework for temporal-aware QoS prediction by dynamic graph neural collaborative learning. It first models a temporal-aware service ecosystem as a dynamic user-service invocation graph, which is then fed into a graph neural collaborative feature extractor for extracting high-order latent features of users and services at each time slice, considering both indirect user-service invocation correlations and collaborative relationships by similar users or services. Finally, a multi-layer GRU is employed to mine temporal

feature evolution patterns across multiple time slices, leading to vacant QoS prediction. Extensive experiments are conducted based on a large-scale QoS dataset in service computing to validate the superior prediction accuracy of our proposed approach, compared to state-of-the-art competing baselines on MAE and RMSE. In the future work, we are devoted to deeply investigating on how to effectively leverage the contextual information and graph structural properties of users and services to further strengthen the capability of temporal-aware QoS prediction.

Acknowledgements. This work was supported by National Natural Science Foundation of China (No. 62272290, 62172088), and Shanghai Natural Science Foundation (No. 21ZR1400400).

References

1. Chen, L., Zheng, A., Feng, Y., Xie, F., Zheng, Z.: Software service recommendation base on collaborative filtering neural network model. In: Pahl, C., Vukovic, M., Yin, J., Yu, Q. (eds.) ICSOC 2018. LNCS, vol. 11236, pp. 388–403. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03596-9_28
2. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
3. Liang, T., Chen, M., Yin, Y., Zhou, L., Ying, H.: Recurrent neural network based collaborative filtering for QoS prediction in IoV. IEEE Trans. Intell. Transp. Syst. **23**(3), 2400–2410 (2022)
4. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
5. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: International Conference on World Wide Web (WWW), pp. 285–295 (2001)
6. Tong, E., Niu, W., Liu, J.: A missing QoS prediction approach via time-aware collaborative filtering. IEEE Trans. Serv. Comput. (2021). https://doi.org/10.1109/TSC.2021.3103769
7. Wu, X., Fan, Y., Zhang, J., Lin, H., Zhang, J.: QF-RNN: QI-matrix factorization based RNN for time-aware service recommendation. In: IEEE International Conference on Services Computing (SCC), pp. 202–209. IEEE (2019)
8. Xue, G.R., et al.: Scalable collaborative filtering using cluster-based smoothing. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 114–121 (2005)
9. Zhang, Y., Zheng, Z., Lyu, M.R.: WSPred: a time-aware personalized QoS prediction framework for web services. In: IEEE International Symposium on Software Reliability Engineering, pp. 210–219. IEEE (2011)
10. Zheng, Z., Ma, H., Lyu, M.R., King, I.: WSRec: a collaborative filtering based web service recommender system. In: IEEE International Conference on Web Services (ICWS), pp. 437–444. IEEE (2009)
11. Zheng, Z., Zhang, Y., Lyu, M.R.: Investigating QoS of real-world web services. IEEE Trans. Serv. Comput. **7**(1), 32–39 (2014)