IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 17, 2019, accepted March 15, 2019, date of publication March 21, 2019, date of current version April 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906769

Research on the Influence of Sampling Methods for the Accuracy of Web Services QoS Prediction

JUN LI AND JIAN LIN
School of Mathematical and Electronic Information Engineering, Wenzhou University , Wenzhou 325035, China
Corresponding author: Jun Li (omama@wzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61402337, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ13F020011.

ABSTRACT In recent years, as the number of Web services, increases dramatically, the personalized Web service recommendation has become a hot topic in both academia and industry. The quality-of-service (QoS) prediction plays a key role in Web service recommendation systems. However, how to further improve the accuracy of QoS prediction is still a problem. Traditional QoS predicting models do not consider the impact of sampling methods on the accuracy of QoS prediction. However, the outstanding sampling method can train the predicting model more effectively and obtain higher accuracy. Therefore, it is necessary to study sampling methods based on the QoS dataset in order to obtain sample distribution closer to the original distribution, so as to improve the accuracy of the predicting models. In this paper, we first discuss how to apply several existing sampling methods to QoS datasets and then analyze their advantages and disadvantages. Finally, a novel sampling method, enhanced importance resampling (EIRS), is proposed and applied. The experiments on the real-world datasets show that our method can not only sample efficiently and accurately but also can greatly improve the accuracy of Web service QoS prediction.

INDEX TERMS Quality of service, Web services, sampling methods, enhanced importance resampling.

I. INTRODUCTION

With the rapid growth of the number of Web services, personalized Web service recommendation has become a hot topic in both academia and industry. Web service QoS(Quality of Service) prediction plays an important role in the process of personalized Web service recommendation. However, how to further improve the accuracy of Web services QoS prediction is still a problem. Traditional researches [1]–[3] mainly focus on increasing the complexity of the predicting models for fixing the problem and simply assume that the probability distribution of QoS datasets is uniform. However, in the real world, QoS datasets tend to follow a complex distribution, that the sampled data (training data of the predicting models) based on such assumption is biased and leads inaccurate prediction. Therefore, it is necessary to study sampling methods based on QoS dataset which can obtain sampling distribution closer to the original distribution, so as to improve the accuracy of predicting models.

In this paper, we firstly discuss how to apply several existing methods [4]–[7] to QoS datasets and then analyze their advantages and disadvantages. Finally, a novel sampling method (Enhanced Importance ReSampling, EIRS) is proposed and applied. Experiments on real-world datasets show that our method can not only sampling efficiently and accurately, but also can greatly improve the accuracy of Web service QoS prediction.

The remainder of the paper is organized as follows. Section IV discusses related works. Section III provides the background and motivations of our work. In Section IV, we firstly discuss how to apply several existing sampling methods to QoS datasets and analyze their advantages and disadvantages. Then a novel sampling method is proposed and applied based on QoS dataset. In section V, we discuss our experimental results in detail. Finally, we conclude our work in Section VI.

II. RELATED WORK

Collaborative filtering (CF)-based approach has been widely used in Web services QoS prediction. There are two main types of CF methods, memory-based CF method and model-based CF method. Memory-based CF methods can be further divided into three categories: User based CF methods [8], [9],

The associate editor coordinating the review of this manuscript and approving it for publication was Anton Kos.

Item based CF methods [10], [11] and hybrid based CF methods [12], [13]. The main steps of memory-based CF methods firstly obtain preferences of users, then calculate similarities between users or services and finally predict QoS values. Memory-based CF method is simple to be implemented and is a computational model of early commercial recommendation system. However, problems such as cold start and inability to handle large-scale and time-aware datasets hinder the popularity of memory-based CF methods.

The model-based prediction methods [14], [15] utilize statistical learning and machine learning techniques to mine and extract the learning model from the historical records of web service invocations, and achieve QoS prediction by matrix decomposition technique. Model-based CF methods can deal with sparse and large-scale datasets better than memory-based CF methods while predicting web services QoS. However, they are more complex and time-consuming. Furthermore, most of the recent model-based CF methods [11], [16] focus on adding additional domain information including context, time and location to improve the accuracy of QoS prediction. Although such additional information can improve the predicting accuracy, those models all use simple random sampling method to obtain training data from the original datasets, which makes the training data biased and leads to poor prediction accuracy.

In the field of statistics, many representative sampling methods such as Rejection Sampling method (RJS) [17], Metropolis-hastings sampling method (MHS) [18] and Importance ReSampling(IRS) [19] have been proposed. RJS is an advanced random sampling method for complex problems with high complexity. MHS is a sampling method based on Markov chain Monte Carlo (MCMC) stochastic process [20], random number sequences with specific probability are sampled to make the sample distribution approximately to target distribution and IRS is an effective sampling method for estimating the target distribution of original datasets. However, to the best of our knowledge, those sampling methods have not yet been used on the QoS dataset. Therefore, in section 4 of this paper, we will discuss them in detail and apply those sampling methods to the QoS datasets, and analyze their advantages and disadvantages.

III. MOTIVATION

In this section, we firstly observe the distribution of a real world QoS data (WSDream[1]), then we propose a framework of on-line Web service recommendation system and emphasize the importance of sampling in the process of QoS prediction.

A. OBSERVATIONS OF REAL WORLD QoS DATASET

WSDream is a real world QoS dataset which has been widely utilized by many mainstream predicting models. There are two sub-data sets in the dataset, Response Time (RT) and Throughput (TP) respectively. In Figure 1, the upper and lower parts show the distribution of the QoS value according to five randomly selected users based on RT and TP respectively. We can obviously see that the data present a long tail distribution rather than uniform distribution and most of the values are concentrated in a very small ranges.

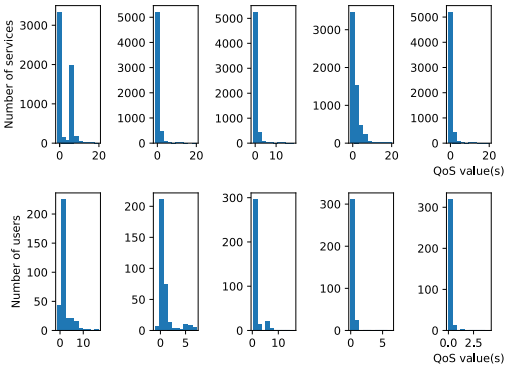

FIGURE 1. The distributions of QoS according to 5 randomly selected users.

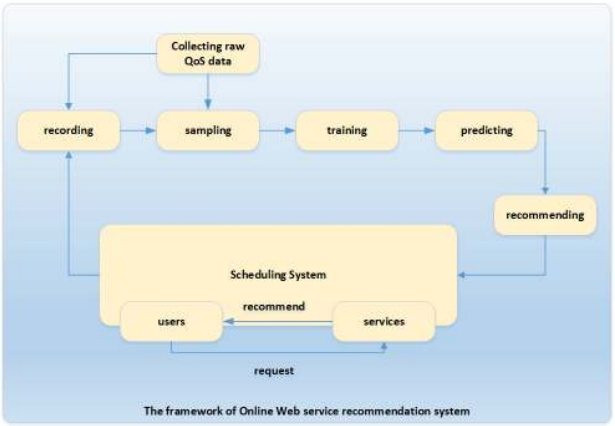

FIGURE 2. The framework of on-line web service recommendation system.

Traditional Web Services QoS prediction models use Simple Random Sampling method to obtain the samples, which mean they simply assume the distribution of the original data is uniform, resulting in inaccuracy prediction results.

B. THE FRAMEWORK OF ON-LINE WEB SERVICES RECOMMENDATION SYSTEMS

Figure 2 shows the framework of QoS prediction based on-line Web service recommendation system. We can see that the framework contains five steps. First, the system collects the original QoS values. Second, the system uses sampling method to obtain training data. Third, prediction algorithms are used to train the model. Fourth, the system predicts the QoS value based on the trained model and personal user requirements. Fifth, the system recommends the personalized web services. Finally, once the user selected one of the recommended services, the scheduling system will schedule the service to the user.

QoS Prediction is the key step in the on-line Web Services Recommendation System. It requires not only accuracy but

[1] github.com/wsdream/wsdream-dataset

also efficiency. The mainstream works focus on designing prediction models to improve the accuracy of QoS Prediction. However, such behavior often brings unnecessary system overhead and longer response time due to the complexity of the models. In order to improve the accuracy of recommendation without reducing user experience, we take the sampling step of on-line web service recommendation system into account, not only because the sampling step is off-line and has no affects of the user experience, but also a good sampling method effectively reduces the bias between training data and original data which helps improving the accuracy of predicting models.

IV. OUR WORK

As far as we know, existing works only use simple random sampling without considering the influence of different sampling methods when predicting Web services QoS. In this section, we will discuss how to apply different sampling methods to the QoS datasets and analyze their advantages and disadvantages, then propose a novel sampling method named Enhanced Importance ReSampling method (EIRS).

A. USER-BASED AND SERVICE-BASED RANDOM SAMPLING BASED ON QoS DATASET

Traditional simple sampling method (RS) assumes that the dataset is uniform distribution and samples globally according to a certain sampling density. However, we observed that some users or services data will never be sampled by using RS, resulting recommendation system unable to recommend services for such users. There are two variants of RS can fix such problem, one is user-based random sampling (URS) method which samples the data randomly according to each user for all users in the dataset and the other is service-based random sampling method (SRS) which samples the data randomly according to each service for all services in the dataset. RS, URS and SRS are all easy to be conducted on the QoS dataset.

B. DOMAIN BASED RANDOM SAMPLING BASED ON QoS DATASET

The domain information such as location and time is closely related to the QoS of Web services. By considering those domain information, domain based random sampling (DRS) method firstly divides the services into different domains and then samples the data randomly in each domain. When conducting a Domain based random sampling method (DRS) on WSDream, we firstly divide the dataset into different parts according to the 'AS' attribute which describe the location of services, then use RS on each part to obtain the samples. However, the sample distribution will be unbalanced because some parts have more data while others have less or even no data.

C. REJECTION SAMPLING BASED ON QoS DATASET

Rejection sampling (RJS) is an advanced random sampling method which can generate complex sample distribution.

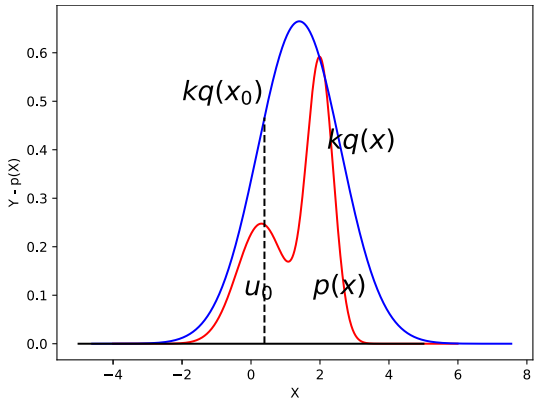

FIGURE 3. An example of RJS.

Algorithm 1 Rejection Sampling

Require: $q(x)$, k , $\tilde{p}(x)$
Ensure: S
1. Sampling s_i from the distribution q(x) randomly;
2. Generate a number u_i from the distribution U(0,1);
3. If $u_i < \frac{\tilde{p}(s_i)}{kq(s_i)}$, then accept s_i;
   If not, then reject the value and repeat the step 1-3;
   Until the new generated s_i is accepted.
4. Add s_i to S;
   Until a certain number of samples are obtained.
return S

Figure 3 shows an example of RJS, where q(x) represents a presumed sample distribution (reference distribution) which can be adjusted after the process of sampling, p(x) represents the distribution of the original dataset (target distribution) and k represents a parameter for scaling all x subject to $kq(x) \geq \tilde{p}(x)$, where $\tilde{p}(x)$ represents the distribution of the sampled data in the process of sampling (observation distribution).

RJS firstly samples the data x_0 randomly according to q(x), then samples the value u_0 randomly in the interval $[0, kq(x_0)]$ and compares $\tilde{p}(x_0)$ to u_0. If $u_0 < \tilde{p}(x_0)$, then accepts the sample with a certain probability, otherwise, rejects. The acceptance probability of the sample can be calculated according to equation (1)

$$p(accept) = \int \frac{\tilde{p}(x)}{kq(x)} q(x)dx = \frac{1}{k} \int \tilde{p}(x)dx \qquad (1)$$

The RJS can be conducted on the QoS dataset according to Algorithm 1.

The inputs of Algorithm 1 include the reference distribution q(x), the scale parameter k and the observation distribution $\tilde{p}(x_0)$. We choose the normal distribution for q(x) according to the distributions of WSDream and specify a large number k in order to cover the range of the target distribution p(x). However, in real applications, it is difficult to find a suitable q(x) because of that when the target distribution is a distribution with spikes, a large number of unwanted samples will be sampled. The algorithm terminates until a certain number of samples are obtained. However,

Algorithm 2 Metropolis-Hastings Sampling

Require: $\widetilde{p}(x)$, $q(x)$
Ensure: S
Initialize:
1. Pick an initial state s_0;
2. Set t=0;
Iterate:
1. Generate a candidate state s' , where $q(s'|s_t)$;
2. Calculate the acceptance probability
$A(s'|s_t) = \min\left(1, \frac{\widetilde{p}(s')}{\widetilde{p}(s_t)} \frac{q(s_t|s')}{q(s'|s_t)}\right)$;
3. Accept or Reject:
1. Generate a number $u \in [0, 1]$;
2. If $u \leq A(s'|x_t)$, accept, set $s_{t+1} = s'$, add s' to S;
3. If $u > A(s'|s_t)$, reject, set $s_{t+1} = s_t$;
4. Increment:
set t=t+1;
Until: a certain number of samples are obtained
return S

it converges slowly because lots of data are probably be rejected in the iteration step.

D. METROPOLIS-HASTINGS SAMPLING

Metropolis-hastings sampling (MHS) is based on Markov chain Monte Carlo (MCMC) stochastic process [20] . The basic idea of MHS is firstly constructing Markov Chain from the reference distribution q(x), then randomly selects an initial state of Markov Chain and begin to transfer until the state to be stable. Finally, the obtained state sequence can be used to estimate the target distribution.

Considering the complexity of distribution of the QoS dataset and in order to satisfy the fine stationary condition of Markov chain, we calculate the acceptance probability of samples according to equation (2):

$$A(j|i) = min\{1, \frac{\tilde{p}(j)q(i|j)}{\tilde{p}(i)q(j|i)}\} \tag{2}$$

where $A(j|i)$ represents the acceptance probability of sample j condition on the sampled sample j, $\tilde{p}(j)$ and $\tilde{p}(i)$ can be calculate by utilizing observation distribution $\tilde{p}(x)$, $q(i|j)$ and $q(j|i)$ can be calculate by reference distribution q(x). The pseudo code of MHS algorithm based on QoS dataset is described in Algorithm 2. We can see that in the iteration step of Algorithm 2, a candidate state s' is generated and then calculate the conditional probability $A(s'|s_t)$, where s_t is a sample already be sampled. If $A(s'|s_t)$, is larger than u which is a random number between [0, 1], accepts s', otherwise, rejects. Similar to RJS, MHS converges slowly because lots of data are probably be rejected in the iteration step.

E. IMPORTANCE RESAMPLING

The main idea of Importance Sampling is to find the function expectation of the target distribution, which can be described

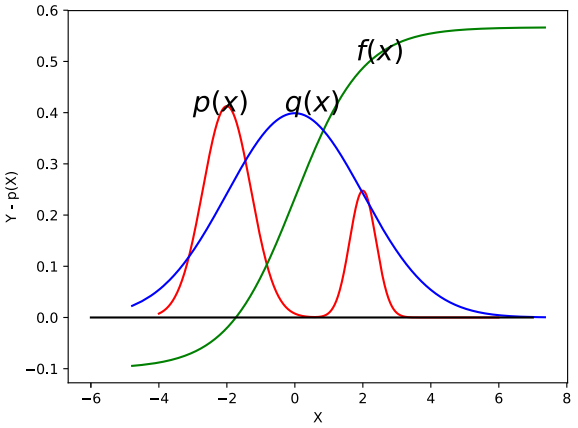

FIGURE 4. An example of IRS.

in equation (3):

$$E(f) = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx$$
$$\approx \frac{1}{N}\sum_{i}^{N} w(x_i)f(x_i),\ w(x_i) = \frac{p(x_i)}{q(x_i)} \tag{3}$$

where, E(f) represents the function expectation of the target distribution, f(x) is the function and p(x) is the target distribution. q(x) represents the reference distribution and $w(x_i)$ represents the weight of the sample x_i which can be calculated by $\frac{\tilde{p}(x_i)}{q(x_i)}$ where $\tilde{p}(x)$ is the observation distribution. From equation (3), we know that importance sampling aim to calculate the function expectation of the target distribution, the sampling distribution is still the same as q(x). However, the weights will greatly improve the information of the samples. For example, as shown in Figure 4, if $\frac{\tilde{p}(x_i)}{q(x_i)}$ is equal to 1, then $w(x_i)$ is equal to 1 according to equation (3), which means sample x_i must subject to the target distribution. Therefore, the distribution of samples with higher weight is more approximate to the target distribution.

Based on the idea described above, when conducting Importance ReSampling(IRS) based on QoS dataset, we can resample the samples according to the weights. The pseudo code of IRS algorithm based on QoS dataset is described in Algorithm 3. As shown in Algorithm 3, the inputs of Algorithm 3 contain four parameters, Q represents the original dataset, q(x) represents the reference distribution, interval represents the sampling interval (a certain interval which can be specified manually) and density represents the sampling density, respectively. Statistic represents a statistical function for estimating the observation distribution $\tilde{p}(x)$. We normalized the weight vector W before the step of resampling in order to make the weights more accuracy. *random*() is random function for resampling from the samples. Therefore, the output of Algorithm 3 is the vector S which store all the resampled samples with better distribution. It is worth to mention that, IRS coverages much faster that RJS and MHS, because there is no need to reject data in the iteration process.

Algorithm 3 Importance Resampling Method

Require: Q , $q(x)$,*interval*,*density*
Ensure: S
  for Q_i in Q **do**
    $\widetilde{p}(x) = statistic(Q_i, interval)$
    for $j = 0$ to interval **do**
      $weight_j = \frac{\widetilde{p}(x_j)}{q(x_j)}$
    end for
    for $j = 0$ to interval **do**
      $W_j = \frac{weight_j}{\sum_j^{interval} weight_j}$
    end for
    for w_j in W **do**
      $s_j = random(Q_i, \widetilde{p}(x_j), w_j, density)$
      $Add \quad s_j \quad to \quad S_i$
    end for
  end for
  return S

Algorithm 4 Enhanced Importance Resampling Method

Require: Q , $q(x)$,*interval*,*density*
Ensure: S
  for Q_i in Q **do**
    $\widetilde{p}(x) = statistic(Q_i, interval)$
    $\widetilde{p}_{invalid}(x) = statistic\_invalid(Q_i, interval)$
    for $j = 0$ to interval **do**
      $weight_j = \frac{\widetilde{p}(x_j)}{q(x_j}$,
    end for
    for $j = 0$ to interval **do**
      $W_j = \frac{weight_j}{\sum_j^{interval} weight_j} \times (1 + \frac{\widetilde{p}_{invalid}(x_j)}{\sum_j^{interval} \widetilde{p}_{invalid}(x_j)})$
    end for
    for w_j in W **do**
      $s_j = random(Q_i, \widetilde{p}(x_j), \widetilde{p}_{invalid}(x_j), w_j, density)$
      $Add \quad s_j \quad to \quad S_i$
    end for
  end for
  return S

F. ENHANCED IMPORTANCE RESAMPLING

Due to the dynamic and variability of Service-Oriented environment, a lot of services may be unavailable or interrupted, it leads many invalid QoS values existed in the original data. Therefore, we propose a novel sampling method (Enhanced importance resampling, EIRS) based on IRS. The details of EIRS can be described as follows: We firstly divide the dataset into n intervals and presume that the distribution of invalid data is uniform in each interval, then the probability of invalid values in the interval can be calculated according to equation (4).

$$p^j_{invalid} = \frac{N^j_{invalid}}{N^j_{all}} \tag{4}$$

where $p^j_{invalid}$ represents the probability of invalid data on interval j, $N^j_{invalid}$ is the total number of the invalid data on interval j, and N^j_{all} is the total number of data on interval j. Then, we normalized the probabilities according to equation (5)

$$\tilde{p}^j_{invalid} = \frac{p^j_{invalid}}{\sum_N p^k_{invalid}} \tag{5}$$

where $\tilde{p}^j_{invalid}$ represents the normalized probability of invalid data on interval j. Finally, we obtain the new weights according to equation (6)

$$E(f) = \int f(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{N}\sum_i^N w(x_i)f(x_i),$$
$$w(x_i) = \frac{p(x_i)}{q(x_i)} \cdot (1 + \tilde{p}^{ij}_{invalid}) \tag{6}$$

where $\tilde{p}^{ij}_{invalid}$ represents the normalized probability of invalid data on interval j which sample x_i belongs to. From equation (6), we know that larger the $\tilde{p}^{ij}_{invalid}$ is, the larger $w(x_i)$ is. Therefore, the samples with larger weight will be sampled more likely than the samples with smaller weight, which means the interval with higher probability of invalid data need to be sampled more in order to obtain enough valid data. The pseudo code of EIRS algorithm based on QoS dataset is described in Algorithm 4. In Algorithm 4, the statistic function *statistic_invalid* calculates the $\tilde{p}_{invalid}$ and W_j represents the new weight of the sample which belongs to interval j. By comparing to IRS, EIRS converges fast and is more suitable for QoS datasets.

V. EXPERIMENTS

A. DATASET AND SETUP

The dataset used in this paper is the WSDream, which is a widely used real world dataset in predicting models. The dataset contains two sub-data sets: response time (RT) and throughput (TP). Each dataset contains two dimensions which is user and service respectively. There are 339 users and 5825 services in each dataset. As a supplement to Figure 1, the upper parts of Figure 5 show the distributions of the RT data with 10 randomly selected services and the lower parts of Figure 6 show the distribution of TP data with 10 randomly selected services. It can be seen that, most of the QoS values of each service are concentrated within a very small range.

We apply all the sampling method described above based on the QoS dataset and analyze the performance of those methods in varies situation. All the experiments in this paper are implemented with C++ combined with python 3.5, conducted on a ThinkPad with an 2.2 GHz Intel Core i7 CPU and 16 GB 1600 MHz DDR3 RAM, running Ubuntu 16.

B. EVALUATION METHODOLOGY

The experiments utilize different sampling methods (RS, URS, SRS, DRS, RJS, MHS, IRS and EIRS) according to different density to generate the training data. Two main

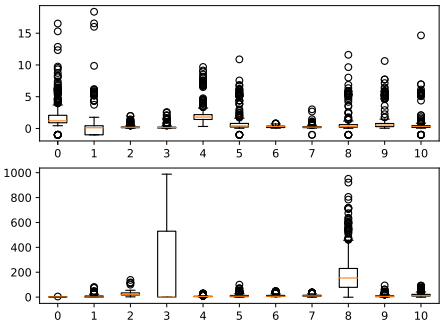

FIGURE 5. The distributions of QoS according with 10 randomly selected services.

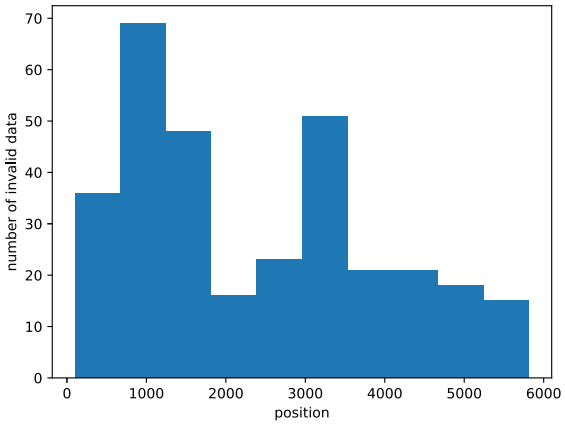

FIGURE 6. The distribution of invalid data on the dataset.

aspects of the sampling methods are evaluated which show the effectiveness and accuracy.

1) THE EFFECTIVENESS OF SAMPLING METHODS

a: METRICS ON THE PROXIMITY BETWEEN TWO PROBABILITY DISTRIBUTIONS

Generally, the measurement of the performance of sampling methods includes three evaluation indicators: unbiasedness, consistency and validity. However, due to the existence of invalid data, unbalance data distribution and the distribution of test data, appropriate evaluation indicators need to be chosen to validate the performance of sampling methods based on real QoS dataset. Here, we choose the Kullback-Leibler divergence(KLD) [21] and Wasserstein distance(WD) [22] to measure the proximity between the sampling distribution and the target distribution.

1) Kullback-Leibler divergence(KLD) is a similarity measurement method between two probability distributions. The formal description of KLD can be shown in equation (7)

$$D_{KL}(p|q) = \int p(x) ln \frac{p(x)}{q(x)} dx \tag{7}$$

where $D_{KL}(p|q)$ represents KLD between the probability distribution p(x) and the probability distribution q(x). According to the equation (7), we can see that KLD is affected by the ratio of distribution p(x) to q(x) which implies the deviation between two probability distributions. The bigger the ratio, the larger the deviation.

TABLE 1. Evaluation metrics of different sampling methods on RT dataset.

density	metrics	RS	URS	SRS	DRS	RJS	MHS	IRS	EIRS
0.05	spl_tst	0.0556	0.0526	0.0631	0.05	0.0555	0.0584	0.0525	0.0568
	kld	25.4239	25.6861	68.6555	26.8741	28.1309	26.5467	25.1156	23.1414
	wd	2.4207	2.4376	2.978	2.4206	2.5854	2.5147	2.3942	2.418
0.1	spl_tst	0.1178	0.111	0.118	0.1087	0.1173	0.1240	0.1108	0.1207
	kld	13.7336	14.8685	28.0394	14.3234	18.1188	16.0442	13.6889	13.1035
	wd	2.4004	2.4238	2.6574	2.4075	2.5596	2.4917	2.3586	2.3598
0.2	spl_tst	0.2671	0.2499	0.2528	0.2473	0.2650	0.2824	0.2497	0.275
	kld	7.9328	8.3677	10.2333	8.4853	12.6083	10.5467	7.6774	7.191
	wd	2.3974	2.4218	2.4773	2.4027	2.5582	2.4956	2.381	2.368
0.4	spl_tst	0.7287	0.6665	0.6606	0.6627	0.7138	0.7801	0.6661	0.7595
	kld	4.3767	4.6711	3.4288	4.7881	7.9984	6.2060	4.2806	4.0666
	wd	2.3974	2.4102	2.3932	2.3992	2.5201	2.4635	2.389	2.3832

2) Wasserstein distance (WD) is a distance measurement method between two probability distributions on a given metric space. The WD can be described in equation (8),

$$W(P_1, P_2) = inf_{\sim \prod(P_1,P_2)} \mathbb{E}_{(x,y)\sim\gamma} ||x - y|| \tag{8}$$

where $\prod(P_1, P_2)$ is a set of all possible joint distributions combined by distributions P1 and distributions P2. For every possible joint distribution γ, a sample $(x, y) \sim \gamma(x, y)$ can be sampled and the distance between x and y can be calculated by using the norm $||x - y||$. Therefore, the distance Expectation is $\mathbb{E}_{(x,y)\sim\gamma}||x - y||$ which can be calculated according to γ, and the lower bound of the expectation is the WD. Intuitively, distance Expectation can be understood as the consumption of moving pile P_1 to pile P_2 under the path planning of γ and WD is the minimum consumption under the optimal path. So the WD is also called the Earth-Mover distance. Comparing to KLD, the advantage of WD is that it can reflect the distance of two distributions even if the support sets of two distributions do not overlap or overlap little.

b: METRIC ON THE VALID RATE OF SAMPLING

The problem of invalid data always exists in the sampling process. Figure 6 shows the distribution of invalid data of the dataset, we can see that there are many invalid data in the dataset, especially in the interval [1000, 2000] and [3000, 4000]. Therefore, the distribution of invalid values must be taken into account in the sampling process in order to obtain high quality samples. We use the ratio of valid data in training data(R_{spl}) and the ratio of valid data in test data (R_{tst}) to define the validity of sampling data as $spl\_tst = \frac{R_{spl}}{R_{tst}}$, The purpose of dividing R_{spl} by R_{tst} is to ensure that the valid ratio of original dataset is unvaried in the process of sampling. *spl_tst* can reflect the actual effective information of the sampled data. The larger *spl_tst* indicates that the sampling method has better ability to obtain more valid training data.

c: RESULTS ANALYSIS

Table 1 and Table 2 show the KLD, wd and *spl_tst* performances of each sampling method on both RT and TP. From the point of view of KLD and WD, the results show that EIRS can obtain the minimum KLD and WD values in any cases

TABLE 2. Evaluation metrics of different sampling methods on TP dataset.

density	metrics	RS	URS	SRS	DRS	RJS	MHS	IRS	EIRS
0.05	spl_tst	0.0570	0.0526	0.0586	0.05	0.0568	0.0598	0.0524	0.0567
	kld	25.2773	27.0916	54.2201	27.6065	35.6419	26.6015	25.9775	24.4994
	wd	145.2512	144.6066	161.1968	144.3496	171.8512	154.8165	141.4795	140.2427
0.1	spl_tst	0.1209	0.111	0.1122	0.1087	0.1202	0.1273	0.1108	0.1206
	kld	15.3108	15.7397	22.2069	15.3485	24.9972	16.6898	15.0041	14.2012
	wd	144.7640	143.9657	148.3627	143.8272	170.4163	154.2393	142.428	141.298
0.2	spl_tst	0.2750	0.2499	0.2461	0.2472	0.2716	0.2913	0.2497	0.2749
	kld	8.5728	9.0413	8.3121	8.8696	17.5513	10.2605	8.6061	7.5669
	wd	144.4327	143.5836	143.4461	143.6209	167.5146	152.3375	143.0268	142.2296
0.4	spl_tst	0.7585	0.6665	0.6505	0.6627	0.7332	0.8161	0.6661	0.7594
	kld	4.4119	4.825	3.0716	4.7205	11.2565	5.7090	4.3987	3.9457
	wd	143.8257	143.5792	142.2855	143.8417	162.0932	149.8996	143.1825	142.8603

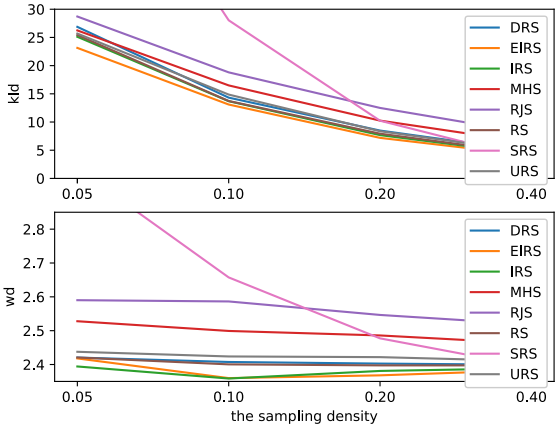

FIGURE 7. The variation of kld and wd according to different density.

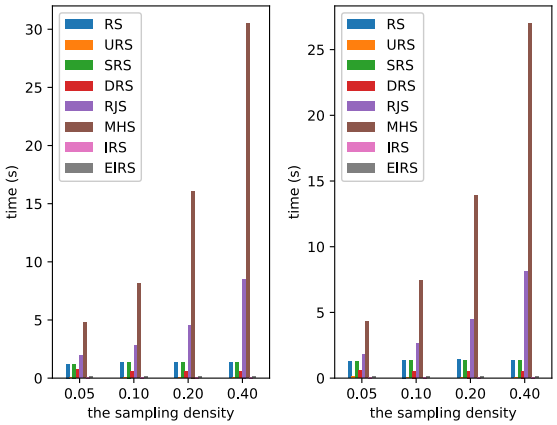

FIGURE 8. The running time of different sampling methods.

which verify that EIRS has the best performance among all of the sampling methods on measuring the proximity between sample distribution and target distribution. RJS and MHS perform poorly on both RT and TP due to the invalid data in the dataset. SRS performs poorly when the sampling density is low and became better with the increase of sampling density. From the point of view of *spl_tst*, EIRS also has better performance in most cases compared with RS, URS, SRS and DRS.

Figure 7 further verifies the high performance of EIRS on KLD and WD. In addition, we can see that the KLD is varied dramatically with varying the sample density while the WD maintain relatively stable. This result shows that WD is more robust than KLD in measuring the proximity between two distributions.

Figure 8 shows the running time of different sampling methods according to different sampling densities. We can see that the running time of advanced sampling methods such as RJS and MHS is much longer than other sampling methods due to high complexity of the models. However, IRS and EIRS are very efficient in all cases. In conclusion, EIRS can not only efficiently obtain the sample distribution which is closer to the target distribution, but also has a higher valid rate of training data than other sampling methods.

2) THE ACCURACY OF SAMPLING METHODS

a: METRICS ON THE ACCURACY

The Mean Absolute Error (MAE) and the Normalized Mean Absolute Error (NMAE) are widely used for evaluation the accuracy of predicting models. MAE reflects the absolute error of the prediction model and NMAE reflects the relative error of prediction models. The formula of MAE is shown in equation (9):

$$MAE = \frac{1}{N}\sum_{i,j}|q_{ij} - \hat{q_{ij}}| \tag{9}$$

where, q_{ij} represents the predicted value, $\hat{q_{ij}}$ represents the real value and N represents the total number of predictions. Based on the MAE, the formula of NMAE can be described in equation (10).

$$NMAE = \frac{MAE}{\frac{1}{N}\sum_{i,j}|q_{ij}|} \tag{10}$$

b: THE PREDICTION MODELS

- UPCC [10] is a user-based CF prediction model that calculates the similarity between users based on Pearson correlation coefficient. The model use RS to sample the training data.
- IPCC [10] is an item-based CF prediction model that calculates the similarity between services based on Pearson correlation coefficient. The model use RS to sample the training data.
- UIPCC [10] is a hybrid prediction model that linearly combines the predictions of UPCC and IPCC, and its accuracy is more precise than either. The model use RS to sample the training data.
- PMF [23] is a MF based prediction model based on probability model and matrix decomposition. The model use RS to sample the training data.
- RWEMF [16] is hybrid prediction model that combines the advantages of CF and MF. The model use RS to sample the training data.

c: RESULTS ANALYSIS

We firstly choose UIPCC to combine the different sampling method to verify the performance of EIRS. Table 5 and Table 6 show the MAE and NMAE of UIPCC combining with different sampling methods according to different sampling densities.

TABLE 3. MAE and NMAE with UIPCC (RT datasets).

density	metrics	RS	URS	SRS	DRS	RJS	MHS	IRS	EIRS
0.05	MAE	0.6284	0.6084	0.9927	0.6095	0.6747	0.6395	0.5889	0.5814
	NMAE	0.6894	0.6704	1.0936	0.6715	0.7526	0.7104	0.6515	0.6431
0.1	MAE	0.5583	0.535	1.0135	0.5353	0.5987	0.5685	0.5237	0.5151
	NMAE	0.5852	0.589	1.1165	0.5897	0.6756	0.6365	0.5785	0.5677
0.2	MAE	0.471	0.4584	1.0651	0.464	0.4995	0.483	0.4531	0.4441
	NMAE	0.5185	0.5048	1.1726	0.5108	0.5776	0.5507	0.4998	0.4889
0.4	MAE	0.4143	0.4171	1.109	0.4197	0.433	0.4151	0.413	0.4096
	NMAE	0.4568	0.4593	1.2199	0.4617	0.5236	0.4876	0.4556	0.4501

TABLE 4. MAE and NMAE with UIPCC (TP datasets).

density	metrics	RS	URS	SRS	DRS	RJS	MHS	IRS	EIRS
0.05	MAE	25.8789	26.3493	51.0864	26.4547	31.0257	26.9581	24.9071	23.8239
	NMAE	0.5453	0.5539	1.0747	0.5565	0.6689	0.5724	0.5283	0.5062
0.1	MAE	22.1046	22.2903	52.3392	22.0611	25.847	22.5996	21.2443	20.3763
	NMAE	0.4652	0.4687	1.0999	0.4636	0.5716	0.4844	0.4502	0.4322
0.2	MAE	18.8014	19.0229	55.9949	18.8095	21.047	18.4506	18.0921	17.5639
	NMAE	0.3963	0.3997	1.1766	0.3953	0.49	0.4032	0.3832	0.3722
0.4	MAE	15.3488	15.6829	57.1014	15.6574	16.3278	14.9398	15.0906	14.7301
	NMAE	0.3228	0.3297	1.2007	0.3292	0.4288	0.3402	0.3192	0.3116

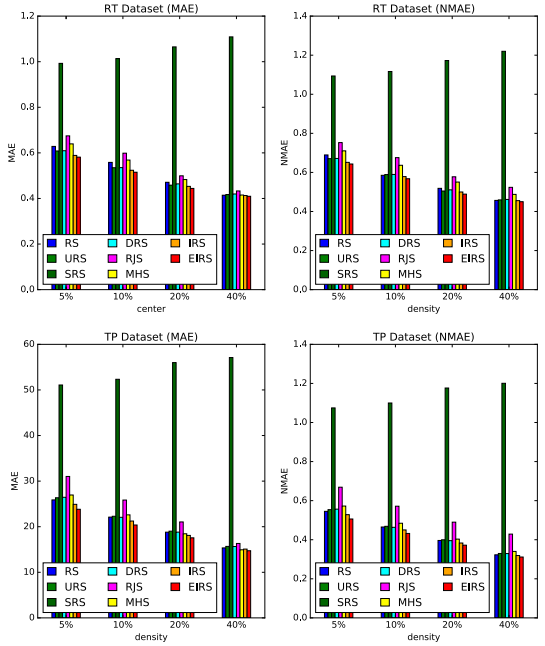

FIGURE 9. The MAE and NMAE of UIPCC combining with different sampling method according to different density.

The results show that the performance of EIRS on MAE and NMAE, we can see that in any case the MAE and NMAE of EIRS is always the lowest among all of the sampling methods which verify that EIRS is an excellent and stable sampling method.

Figure 9 shows the results of Table 3 and Table 4 more intuitively. On the RT dataset, it can be observed that all sampling methods (except SRS) perform better with increasing the sampling density. This is reasonable because of that the more data are obtained when the density is high. However, SRS performs poorly in both sparse and dense data. This is because of that despite the high density, the number of users is so small that training data has a lot of duplicate and invalid data. URS and DRS have significantly better performance than RS when the data is sparse (density is equal to 0.05 or 0.1). However, with the density of data increasing, RS performs better and catch up with or even exceed URS and DRS (when the density is equal to 0.4). In particular, the performance of MHS and RJS is poor in all cases. This is because of that influenced by rejection domain, there are more invalid data in training data. Furthermore, MHS and RJS are sensitive to UIPCC which is a CF based predicting model. Similar observations can be seen on the TP dataset, the subtle difference is that URS and DRS perform worse than RS in most cases. This result shows that most sampling methods such as URS, SRS. DRS, RJS and MHS are sensitive to the dataset and perform unstable.

TABLE 5. MAE and NMAE of prediction algorithm (RT datasets).

methods	metrics	UPCC	IPCC	UIPCC	PMF	RWEMF
RS	MAE	0.6583	0.6719	0.6284	0.6196	0.5119
	NAME	0.7252	0.7413	0.6894	0.6827	0.5759
URS	MAE	0.6195	0.6629	0.609	0.5734	0.506
	NAME	0.6836	0.7316	0.672	0.6327	0.5776
SRS	MAE	1.0947	0.6801	0.9927	0.8463	1.0108
	NAME	1.2059	0.7492	1.0936	0.9323	1.1135
DRS	MAE	0.6223	0.6801	0.6095	0.578	0.5251
	NAME	0.6856	0.7492	0.6715	0.6367	0.5784
RJS	MAE	0.692	0.7691	0.6786	0.5467	0.506
	NAME	0.7725	0.8585	0.7575	0.6102	0.5627
MHS	MAE	0.6469	0.7237	0.6351	0.5432	0.5107
	NAME	0.7189	0.8043	0.7058	0.6036	0.5723
IRS	MAE	0.5986	0.6676	0.5889	0.573	0.4986
	NAME	0.6622	0.7385	0.6515	0.6338	0.5516
EIRS	MAE	0.5856	0.6626	0.5774	0.5575	0.4951
	NAME	0.6476	0.7327	0.6386	0.6166	0.5475

TABLE 6. MAE and NMAE of prediction algorithm (TP datasets).

methods	metrics	UPCC	IPCC	UIPCC	PMF	RWEMF
RS	MAE	27.3259	28.5857	26.8493	26.8448	19.9955
	NMAE	0.5682	0.5947	0.5539	0.556	0.412
URS	MAE	27.0259	28.2857	26.3493	26.4448	19.5955
	NMAE	0.5682	0.5947	0.5539	0.556	0.412
SRS	MAE	59.2848	29.7132	52.256	83.5387	60.0657
	NMAE	1.2471	0.625	1.0993	1.7573	1.2635
DRS	MAE	27.5949	29.7132	26.7591	27.6782	19.8688
	NMAE	0.5805	0.625	0.5629	0.5822	0.418
RJS	MAE	32.5687	35.1533	31.5346	27.7825	19.1482
	NMAE	0.7024	0.7581	0.6801	0.5991	0.4129
MHS	MAE	28.4658	31.2388	27.6002	26.6521	18.5621
	NMAE	0.6044	0.6632	0.586	0.5659	0.3941
IRS	MAE	25.6509	29.1792	25.0884	26.2071	19.3165
	NMAE	0.544	0.6189	0.5321	0.5558	0.4097
EIRS	MAE	24.3962	28.4532	23.9278	26.2042	19.1171
	NMAE	0.5184	0.6046	0.5084	0.5568	0.4062

Tables 5 and 6 show the performance of different sampling methods combined with different predicting models when using a fixed sampling density (0.05). The results based on RT dataset show that the combination of EIRS with all the predicting models can significantly improve the prediction accuracy which mean the EIRS can help prediction models to improve the prediction accuracy in general. It is worth to mention that when EIRS is combined with better prediction model RWEMF, the MAE value is 0.4951, breaking through the bottleneck value of 0.5 which means that when excellent sampling method and excellent prediction model are combined, high quality prediction can also be obtained on sparse data. IRS has the same result as EIRS except the situation when combining with UPCC. This result shows that the prediction accuracy will be affected by invalid data. Similar results can be seen on TP dataset.

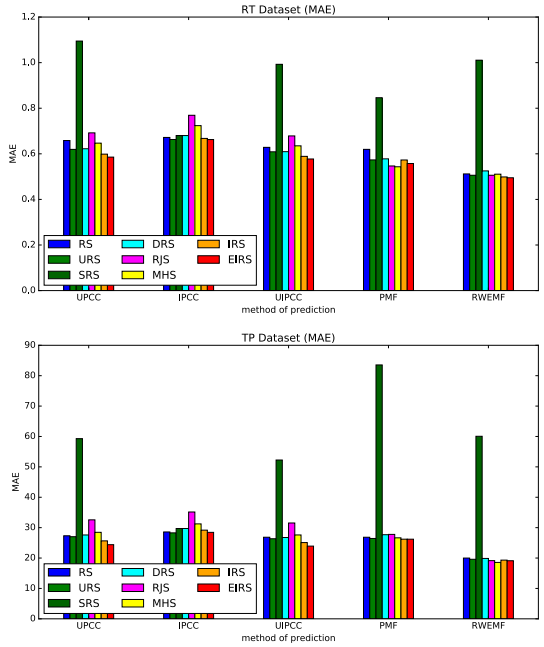

FIGURE 10. The MAEs of different predicting models combining with different sampling method.

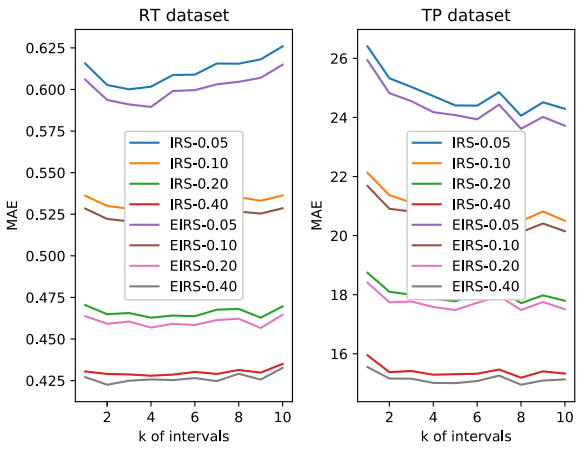

FIGURE 11. The MAEs of IRS and EIRS combined with UPCC according to different intervals.

Figure 10 shows the MAE of Tables 5 and 6 more intuitively. We can obviously see that, SRS performs poorly in all of the cases except in combining with IPCC (based on calculating the similarity between services). This is because of that the training data obtained by SRS is more suitable for item-based CF methods. URS and DRS perform better in combining with CF-based prediction methods especially UPCC and UIPCC (based on calculating the similarity between users) than MF-based prediction methods. Furthermore, better prediction accuracy are obtained when the sampling methods especially RJS and MHS combining with RWEMF which is more excellent MF-based predicting model. This is because of that excellent MF-based prediction model can better mine the latent features of the data and remedy the inaccuracy of the training data itself. To sum up, most sampling methods can improve the prediction accuracy only when combined with the appropriate prediction model, but our method EIRS can help the prediction model to improve the accuracy in general. However, in order to obtain high-quality prediction accuracy, we not only need good sampling method, but also need good prediction model.

3) EFFECT OF THE SAMPLING INTERVAL

In the process of sampling, the sampling interval affects the sampling distribution. Figure 11 shows the performance of IRS and EIRS combined with UPCC on MAE with different density according to different intervals. We can see that on the RT dataset, when the sampling interval is equal to 4, the result is the best in most cases. However, on the TP dataset the best result is obtained when the sampling interval is equal to 8. This result shows that an appropriate sampling interval is needed in different dataset.

VI. CONCLUSION AND FUTURE WORK

In this paper, we discuss how to apply different sampling methods to QoS datasets and analyze their advantages and disadvantages. In addition, we propose a novel sampling method (Enhanced Importance ReSampling method, EIRS) by considering the influence of invalid data of the training data. Experiments show that our method can sample Web service QoS data more stably and effectively, and improves the accuracy of predicting models higher than others. In the future, we will consider exploring more excellent sampling methods and designing more efficient prediction models to improve the accuracy of QoS prediction.

REFERENCES

[1] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based Web service recommender system," in *Proc. IEEE Int. Conf. Web Services*, Jul. 2009, pp. 437–444.

[2] P. Wang, A. K. Kalia, and M. P. Singh, "A collaborative approach to predicting service price for QoS-aware service selection," in *Proc. IEEE Int. Conf. Web Services*, Jun./Jul. 2015, pp. 33–40.

[3] F. Vahedian, R. Burke, and B. Mobasher, "Weighted random walk sampling for multi-relational recommendation," in *Proc. 25th Conf. User Model., Adaptation Personalization*, Jul. 2017, pp. 230–237.

[4] W. Zhang, M. Gen, and J. Jo, "Hybrid sampling strategy-based multiobjective evolutionary algorithm for process planning and scheduling problem," *J. Intell. Manuf.*, vol. 25, no. 5, pp. 881–897, Oct. 2014.

[5] J. Kawale, H. Bui, B. Kveton, L. Tran-Thanh, and S. Chawla, "Efficient Thompson sampling for online matrix-factorization recommendation," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2015, pp. 1297–1305.

[6] J. Li, M.-J. Er, and H. Yu, "Sampling and control strategy: Networked control systems subject to packet disordering," *J. Mag.*, vol. 10, no. 6, pp. 674–683, Apr. 2016.

[7] P. Singh, J. V. D. Herten, D. Deschrijver, I. Couckuyt, and T. Dhaene, "A sequential sampling strategy for adaptive classification of computationally expensive data," *Struct. Multidisciplinary Optim.*, vol. 55, no. 4, pp. 1425–1438, Apr. 2017.

[8] A. Agarwal, S. N. Negahban, and M. J. Wainwright, "Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1538–1546.

[9] S. Liu *et al.*, "Privacy-preserving collaborative Web services QoS prediction via differential privacy," in *Proc. Asia–Pacific Web*, 2017, pp. 200–214.

[10] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," *IEEE Trans. Serv. Comput.*, vol. 4, no. 2, pp. 140–152, Apr. Jun. 2011.

[11] Z. Xu, C. Chen, T. Lukasiewicz, Y. Miao, and X. Meng, "Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)* 2016, pp. 1921–1924.
[12] T. Shang, Q. He, F. Zhuang, and Z. Shi, "Extreme learning machine combining matrix factorization for collaborative filtering," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–8.
[13] G. Alexandridis, G. Siolas, and A. Stafylopatis, "Enhancing social collaborative filtering through the application of non-negative matrix factorization and exponential random graph models," *Data Mining Knowl. Discovery*, vol. 31, no. 4, pp. 1031–1059, Jul. 2017.
[14] R. Salakhutdinov and A. Mnih. "Probabilistic matrix factorization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.
[15] X. Wang, X. Fu, L. Liu, Q. Huang, K. Yue, "A probabilistic approach to analyzing the stochastic QoS of Web service composition," in *Proc. 12th Web Inf. Syst. Appl. Conf.*, Sep. 2015, pp. 147–150.
[16] H. Park, J. Jung, and U. Kang. (2017). "A comparative study of matrix factorization and random walk with restart in recommender systems." [Online]. Available: https://arxiv.org/abs/1708.09088
[17] J. Lv, Q. M. Liu, Y. J. Ren, W. F. Wang, T. Gong, and L. M. Li, "Application of a simple random sampling method on surveys at the community level," *Zhonghua Liu Xing Bing Xue Za Zhi*, vol. 31, no. 4, pp. 421–423, Apr. 2010.
[18] M.-S. Oh and J. O. Berger, "Adaptive importance sampling in monte carlo integration," *J. Stat. Comput. Simul.*, vol. 41, nos. 3–4, pp. 143–168, 1992.
[19] G. Bal and I. Langmore, "Importance sampling and adjoint hybrid methods in Monte Carlo transport with reflecting boundaries," *Physical*, p. 53, Apr. 2011. [Online]. Available: http://www.oalib.com/paper/3725222
[20] A. Shapiro, "Monte Carlo sampling methods," *Handbooks Oper. Res. Manage. Sci.*, vol. 10, no. 3, pp. 353–425, Dec. 2003.
[21] K.-L. Divergence, *Kullback–Leibler Divergence*. Saarbrücken, Germany: Alphascript, 2011, p. 844.
[22] E. Anderes, S. Borgwardt, and J. Miller, "Discrete Wasserstein barycenters: Optimal transport for discrete data," *Math. Methods Oper. Res.*, vol. 84, no. 2, pp. 389–409, 2015.
[23] J. Xu, Z. Zheng, and M. R. Lyu, "Web service personalized quality of service prediction via reputation-based matrix factorization," *IEEE Trans. Rel.*, vol. 65, no. 1, pp. 28–37, Mar. 2016.

JUN LI received the Ph.D. degree in computer science from Zhejiang University, China, in 2012. Since 2016, he has been an Associate Professor with Wenzhou University, China. His research interests include service computing and data mining.

JIAN LIN is currently pursuing the master's degree with the Department of Computer Software and Theory Electronic Information Engineering, Wenzhou University. His research interests include service computing and recommendation.