Noname manuscript No.
(will be inserted by the editor)

Differential Private Collaborative Web Services QoS Prediction

An Liu · Xindi Shen · Zhixu Li ·
Guanfeng Liu · Jiajie Xu · Lei Zhao ·
Kai Zheng · Shuo Shang*

Received: date / Accepted: date

Abstract Collaborative Web services QoS prediction has proved to be an important tool to estimate accurately personalized QoS experienced by individual users, which is beneficial for a variety of operations in the service ecosystem, such as service selection, composition and recommendation. While a number of achievements have been attained on the study of improving the accuracy of collaborative QoS prediction, little work has been done for protecting user privacy in this process. In this paper, we propose a privacy-preserving collaborative QoS prediction framework which can protect the private data of users while retaining the ability of generating accurate QoS prediction. We introduce differential privacy, a rigorous and provable privacy model, into the process of collaborative QoS prediction. We first present DPS, a method that disguises a user's observed QoS values by applying differential privacy to the user's QoS data directly. We show how to integrate DPS with two representative collaborative QoS prediction approaches. To improve the utility of the disguised QoS data, we present DPA, another QoS disguising method which first aggregates a user's QoS data before adding noise to achieve differential privacy. We evaluate the proposed methods by conducting extensive experiments on a real world Web services QoS dataset. Experimental results show our approach is feasible in practice.

Keywords Web services; collaborative QoS prediction; privacy-preserving

* corresponding author

A. Liu, X. Shen, Z. Li, G. Liu, J. Xu, L. Zhao
School of Computer Science and Technology, Soochow University, Suzhou, China
E-mail: anliu@suda.edu.cn

K. Zheng
Big Data Research Center, University of Electronic Science and Technology of China

S. Shang
King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia
E-mail: jedi.shang@gmail.com

1 Introduction

Quality of Service (QoS) has been widely used for describing nonfunctional characteristics of web services [20,23], for example, the response time of a Web service is the expected delay between the time when a request to the service is sent by a user and the time when the results are received by the user. QoS has been an important metric for users to decide which services should be used. As a result, QoS-based Web services selection, composition and recommendation have been studied extensively in the recent literature [25,21,22,41,42,39,29,53,55,40,47]. A common assumption of these proposed approaches is that accurate QoS values of Web Services are always available. It is, however, still an open problem to obtain accurate QoS values. On one hand, the QoS values advertised by Web service providers or third-party communities are not accurate to users, as they are susceptible to the uncertain Internet environment and the context where users are. On the other hand, it is impractical for users to directly evaluate the QoS of all available services due to the constraints of time, cost and other resources.

As an effective solution to this problem, personalized collaborative Web services QoS prediction [65,59] has received much attention recently. The basic idea is that similar users tend to observe similar QoS for the same service, so it is possible to predict the QoS value of the service observed by a user u based on the QoS values of the service observed by users similar to u. By this kind of computation, different users are typically given different QoS prediction values even for the same service and the final prediction values in fact depend on their specific contexts. Based on this basic idea, a variety of techniques have been employed to improve the accuracy of prediction [56,58,67,50,44,52,28].

Though many achievements have been attained on the study of improving the accuracy of collaborative QoS prediction, little work has been done for protecting user privacy in this process. In fact, the QoS values observed by users could be some kind of sensitive information, so users may not be willing to share them with others, in particular, the untrusted server (also called recommender) which is in charge of collaborative QoS prediction. For example, the observed response time reported by a user typically depends on his/her location [50], which means that the user's location could be deduced from the QoS information he/she provided. Consequently, an interesting but challenging problem is whether or not a recommender can make accurately personalized QoS prediction for users without knowing the exact value of their private data.

Homomorphic encryption [12] which allows computations to be carried out directly on ciphertexts seems to be a feasible way to solve the above problem. In [18], the authors present a secure way to protect users' private data during QoS prediction by combining homomorphic encryption with Yao's garbled circuits. Though its security is guaranteed by well-established cryptographic tools, the proposed approach cannot scale to large problems due to the prohibitive computation cost and communication cost of these expensive cryptographic tools. Hence, it is necessary to devise some scalable solutions based on lightweight privacy-preserving tools.

Randomized perturbation, another privacy-preserving technique proposed in [37], adds randomness from a specific distribution to the original data to prevent information leakage. Meanwhile, the injected randomness can be largely eliminated for some aggregation operations, which enables certain kinds of computation on the perturbed data. This technique is adopted in a recent work of privacy-preserving collaborative QoS prediction [69]. However, the distribution of randomness is chosen by experience, so the method itself cannot have provable privacy guarantee. What is worse, it is recognized that with the application of clustering on the perturbed data, adversaries can accurately infer users' private data with accuracy up to 70% [60]. In other words, methods based on randomized perturbation are useless in practice as they cannot provide sufficient security.

In this paper, we develop solutions to privacy-preserving collaborative Web services QoS prediction based on an emerging privacy model called differential privacy [8]. Differential privacy is lightweight compared with conventional public-key cryptosystems, but is a strong and provable privacy model. Our basic idea is to adopt differential privacy as a tool for privacy-preserving data publication, that is, every user adds random noise to his/her observed QoS data according to Laplace mechanism which is a classical way to achieve differential privacy. These disguised QoS values are then sent to the recommender for collaborative QoS prediction. In particular, we propose two methods to disguise users' QoS data. In the first method called DPS, every user considers his/her observed QoS data as a vector and calculates the sensitivity used by Laplace mechanism over the whole vector. The unique sensitivity is used to generate suitable noise to disguise the original QoS vector. In the second method called DPA, every user first performs some aggregation on his/her QoS vector and then calculates different sensitivities for different aggregated QoS values. For each aggregated QoS value, a specific sensitivity is used to control injected noise. By using different sensitivities, the utility of disguised QoS data can be improved significantly, which makes the final prediction based on DPA more accurate than that based on DPS.

To sum up, our work is to formulate a new framework to protect sensitive QoS data in the course of collaborative QoS prediction. More specifically, the contributions of our work can be summarized as follows:

- We propose a privacy-preserving solution to collaborative Web services QoS prediction where the original QoS data are disguised by noises generated according to Laplace mechanism. To the best of our knowledge, this is the first work that applies differential privacy to collaborative Web services QoS prediction.
- We propose two methods to disguise original QoS data. In the first method, we add noise to single QoS value and show how to run representative collaborative QoS prediction models on disguised QoS data. In the second method, we add noise to aggregated QoS value to improve the utility of the disguised QoS data.

- We evaluate our solution on a real world Web services QoS dataset. Experimental results show that our solution is feasible in practice.

The remainder of this paper is organized as follows: in Section 2 we introduce the basic knowledge of differential privacy. Section 3 presents our privacy-preserving collaborative QoS prediction framework and two methods of QoS disguising. Experimental results and analysis are given in Section 4. Finally, Section 6 concludes the paper after discussions of related work in Section 5.

2 Differential Privacy

In this section, we briefly introduce differential privacy [6], which is the building block of our approach to privacy-preserving collaborative QoS prediction. Generally speaking, differential privacy ensures that the outcome of a randomized computation is insensitive to the removal or addition of any record. It gives a rigorous and quantitative definition on the privacy leakage under a very strict attack model: an attacker cannot distinguish a record with a probability more than ϵ even the attacker has the knowledge of the entire dataset except the target one. The formal definition is as follows:

Definition 1 (ϵ-Differential Privacy [7]) A randomized function K gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(K)$,

$$\frac{Pr[K(D_1 \in S)]}{Pr[K(D_2 \in S)]} \leq exp(\epsilon).$$

In the above definition, the privacy parameter $\epsilon > 0$ is public, and a smaller ϵ yields a stronger privacy guarantee.

Differential privacy can be achieved by adding random noise with distribution like Laplace. A random variable has a Laplace(μ, b) distribution if its probability density function is:

$$f(x|\mu, b) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b})$$

where μ and b are the location parameter and scale parameter, respectively. For the sake of simplicity, we set $\mu = 0$, so the distribution can be regarded as the symmetric exponential distribution with the standard deviation of $\sqrt{2}b$.

To add noise with Laplace distribution, b is set to $\Delta f/\epsilon$ and the generation of noise is referred as $L(\Delta f/\epsilon)$ where Δf is global sensitivity, whose definition is as follows:

Definition 2 (Global Sensitivity [7]) For f: $D \rightarrow R^d$, the L_k-sensitivity of f is

$$\Delta f = max_{D_1,D_2} ||f(D_1) - f(D_2)||_k$$

As mentioned earlier, D_1, D_2 are neighboring if and only if they are different by one element. Kifer et al. [17] point out that there are two choices to form two datasets differing in at most one element, called unbounded DP and bounded DP. In Unbounded DP, D_1, D_2 are neighboring if D_1 can be obtained from D_2 by adding or removing one element. In bounded DP, D_1, D_2 are neighboring if D_1 can be obtained from D_2 by replacing one element in D_2 with another element. In this paper, we use the unbounded DP as its applicability is more extensive.

Laplace mechanism can be further divided into two cases, and their properties are shown in the following theorems.

Theorem 1 (Laplace Mechanism, the vector case [8]) *Given a function f whose value is a k-dimensional vector, the following computation maintains ϵ-differential privacy:*

$$X = f(x) + < X_1, X_2, \ldots, X_k >$$

where $X_1, X_2, \ldots, X_k$ are independent identically distributed random variables drawn from $L(\Delta f/\epsilon)$.

Theorem 2 (Laplace Mechanism, the scalar case [8]) *Given a function $f \colon D \to R^d$, the following computation maintains ϵ-differential privacy:*

$$X = f(x) + L(\Delta f/\epsilon)$$

The next two theorems describe the significant properties of differential privacy. We will use these properties to analyse the theoretical guarantee of our approach.

Theorem 3 (Sequential Composition [33]) *Let $A_1, A_2, \ldots, A_k$ be k algorithms (that take auxiliary inputs) that satisfy ϵ_1-DP, ϵ_2-DP, ..., ϵ_k-DP, respectively, with respect to the input dataset. Publishing $\mathbf{t} =< t_1, t_2, \ldots, t_k >$, where $t_1 = A_1(D), t_2 = A_2(t_1, D), \ldots, t_k = A_k(< t_1, \ldots, t_{k-1} >, D)$ satisfies $\sum_{i=1}^{k} \epsilon_i$-DP.*

Theorem 4 (Parallel Composition [33]) *Let $A_1, A_2, \ldots, A_k$ be k algorithms that satisfy ϵ_1-DP,ϵ_2-DP,...,ϵ_k-DP, respectively. Given a deterministic partitioning function f, let $D_1, D_2, \ldots, D_k$ be the resulting partitions of executing f on D. Publishing $A_1(D_1), A_2(D_2), \ldots, A_k(D_k)$ satisfies $(max_{i\in[1,\ldots,k]}\epsilon_i)-$DP.*

3 Differential Private Collaborative QoS Prediction

In this section, we present our differential private collaborative Web services QoS prediction approach. After introducing the system model in Section 4.1, we present a basic method by applying differential privacy directly on users' local QoS values in Section 4.2. To improve data utility, we present an advanced

Table 1 Summary of Notations

Notation	Meaning
q_{ui}	the original QoS value collected by user u for Web service i
q_u	the original QoS vector observed by user u on all m services
Q_{ui}	the disguised QoS value collected by user u for Web service i
Q_u	the disguised QoS vector observed by user u on all m services
$P(i)$	the prediction result for value i
S	the set of Web services
m	the number of Web services
n	the number of users
k	the number of countries
$\overline{q}_u$	the mean of q_u
ω_u	the standard deviation of q_u
$\hat{q}_{ui}$	the z-scored normalized QoS value collected by user u for service i
ϵ	the privacy parameter
$sim(u, v)$	similarity between two users u and v on the original QoS values
$Sim(u, v)$	similarity between two users u and v on the disguised QoS values
$Q_{n\times m}$	a user-service QoS matrix of size $n \times m$
$U_{n\times d}$	a user-factor matrix of size $n \times d$
$V_{m\times d}$	a service-factor matrix of size $m \times d$
b_u	the user bias
b_i	the service bias
$C_{m\times k}$	a service-country matrix of size $m \times k$
C_{ij}	the element in C represents whether service i locates in country j
A_u	the aggregated QoS vector for user u
A_{uj}	the j^{th} element in A_u

method by applying differential privacy on aggregated QoS values in Section 4.3. Before presenting the detailed algorithms, we summarize some important notations in the subsequent descriptions. Suppose there are n users and m Web services. Let Q denote a $n \times m$ user-service QoS matrix hold by the QoS predictor. The element q_{ui} in the matrix denotes the QoS value observed by a user u on a Web service i. If u did not use i before, then q_{ui} is set to 0. Let q_u denote a QoS vector observed by u on all m services. The mean and the standard deviation of q_u are denoted by $\overline{q}_u$ and ω_u, respectively. Table 1 summarizes the notations used in this paper.

3.1 System Model

Fig.1 shows the system model of privacy-preserving collaborative Web services QoS prediction. There are three roles in the system: users, Web services, and a Web services recommender. After invoking a Web service, an user observes a particular QoS value determined not only by the Web service itself but also by the user's context. In order to receive personalized recommendation, the user needs to report the observed QoS value to the recommender. This QoS value, however, needs to be disguised before being sent out to avoid the breach of the user's privacy, since the recommender is not a trusted party. After receiving disguised QoS values from different users, the recommender builds a disguised

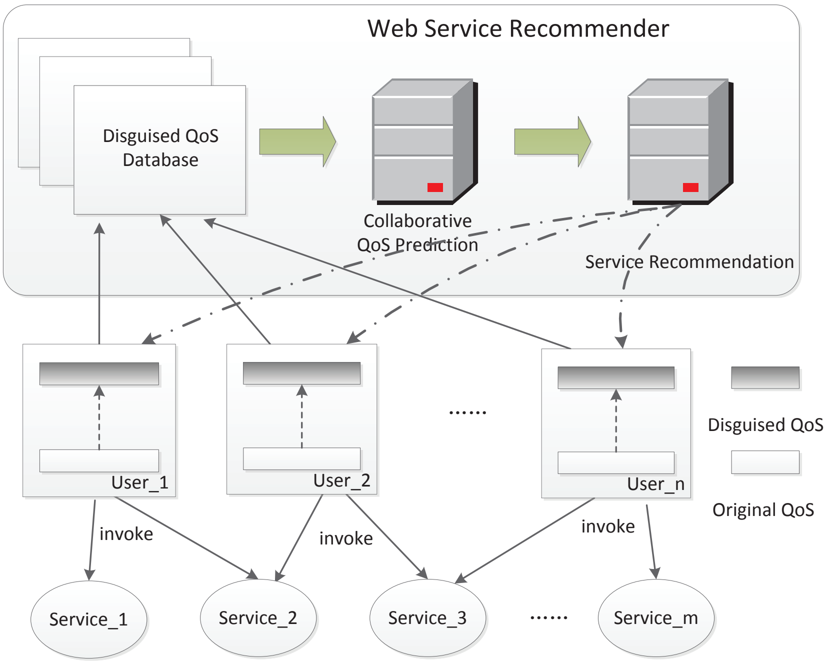


Fig. 1 System Model of Privacy-preserving Collaborative QoS Prediction

QoS database, based on which it can run a lot of collaborative QoS prediction algorithms, such as, UIPCC and MF [65]. The predicted QoS values are then fed into some well-known or customized recommendation algorithms to find suitable Web services to individual users.

The security of the above system model, that is, how well user privacy can be protected during recommendation, depends on the mechanism used for data disguising. The basic idea of data disguising is to perturb the raw data (i.e., the original QoS values) with randomness while keeping the following properties. First, randomness should be able to guarantee no sensitive information can be deduced from disguised QoS values. Second, the aggregation of these disguised QoS values should be evaluated with acceptable accuracy when the number of users is significantly large. Such a property is useful for computations that are based on aggregated data. As a result, it is possible to make accurate QoS prediction outcome without knowing the exact individual QoS values.

As mentioned in the introduction, [69] adopted randomized perturbation as the underlying mechanism of data disguising. However, this has proved to be unsafe in [60] as some sensitive information can be deduced by using the technique of clustering. In this paper, we use differential privacy to disguise original QoS values. An inherent problem of applying differential privacy is the trade-off between accuracy and privacy. The more the randomness, the

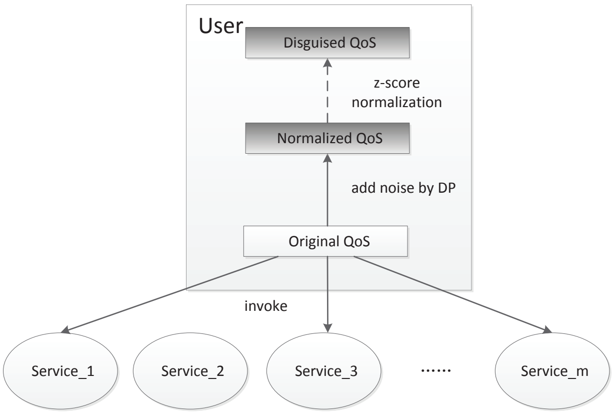


Fig. 2 Data Disguising by Adding Noise to Original QoS

bigger the gap between disguised QoS values and original QoS values. This indicates a higher level of privacy. Oppositely, the less the randomness, the more obvious the data characteristics. It is an open problem to deal with the trade-off between accuracy and privacy, and we will study it extensively in the experiments.

It should also be noted that many studies have investigated the application of differential privacy to recommender systems [31, 32]. In this paper, however, the recommender system is not assumed to be a trusted party. Therefore, the noise decided by differential privacy is not added by the recommender system but by individual users to their own QoS values, which is a major difference between our work and existing works. In particular, we propose two methods to protect users' sensitive QoS values. We first apply the vector case of Laplace mechanism to add noise, which can be performed only on users' local data. To improve data utility, we then use the scalar case of Laplace mechanism through introducing a public knowledge as auxiliary information. We first apply the vector case of Laplace mechanism to add noise, which can be performed only on users' local data. To improve data utility, we then use the scalar case of Laplace mechanism to add noise on the aggregated data through introducing a public knowledge as auxiliary information.

3.2 DPS: Differential Privacy on Simple Data

In this section we present, DPS, a method that disguises original QoS values based on differential privacy on simple data. We first introduce the procedure of DPS (see Figure 2) and then explain how to integrate DPS with two representative collaborative QoS prediction approaches: neighborhood-based and model-based.

We distinguish between disguised QoS and original QoS with upper case (e.g., Q_u and Q_{ui}) and lower case (e.g., q_u and q_{ui}), respectively. For simplicity, we compute Δf with L_1-norm:

$$\Delta f = max_{D_1,D_2}||f(D_1) - f(D_2)||_1$$

For user u, his/her observed QoS on service i, q_{ui}, is disguised as follows:

$$Q_{ui} = q_{ui} + L(\Delta f/\epsilon)$$

where Δf is the maximum difference between any two elements in the vector q_u, that is,

$$\Delta f = max|q_{ui} - q_{uj}|$$

After data disguising, each user u sends his/her disguised QoS values Q_u to the Web services recommender. The original sensitive data q_{ui} is protected by the injected randomness. To enable effective collaborative QoS prediction, however, we need to make accurate computation on the disguised data. Next, we will show how to achieve this in the aforementioned two representative collaborative QoS prediction methods.

3.2.1 Neighbourhood-based Method

A typical procedure of neighborhood-based collaborative QoS prediction consists of three primary steps [65]. The first step is to calculate the similarity between two users according to their observed QoS values. In particular, the similarity between two users u and v can be measured by pearson correlation coefficient as follows:

$$sim(u,v) = \frac{\sum_{s_i \in S}(q_{ui} - \bar{q}_u)(q_{vi} - \bar{q}_v)}{\sqrt{\sum_{s_i \in S}(q_{ui} - \bar{q}_u)^2 \sum_{s_i \in S}(q_{vi} - \bar{q}_v)^2}} \tag{1}$$

where S is a set of services that both u and v have invoked. After that, for every user u, top-k similar users (denoted as S_u) are identified based on the similarities obtained in the first step. Finally, the missing QoS value q_{ui} is evaluated based on these top-k similar users by the following equation:

$$P(q_{ui}) = \bar{q}_u + \sum_{v \in S_u} \frac{sim(u,v)(q_{vi} - \bar{q}_v)}{\sum_{v \in S_u} sim(u,v)} \tag{2}$$

When applying differential privacy into neighbourhood-based method, a key point is to ensure that the similarity value calculated on the disguised QoS values (denoted as $Sim(u,v)$) should approximate the similarity value $sim(u,v)$ calculated on the original QoS values, as this will improve the accuracy of collaborative QoS prediction. To achieve this approximation, z-score normalization is carried out by every user u on the QoS vector q_u he/she observed, as shown in the following equation:

$$\hat{q}_{ui} = (q_{ui} - \overline{q}_u)/\omega_u \tag{3}$$

where $\overline{q}_u$ and ω_u are the mean and standard deviation of QoS vector q_u, respectively.

Based on the procedure of z-score normalization, it is clear that we have $\omega_u = \sqrt{\sum_{s_i \in S}(r_{ui} - \bar{r}_u)^2 / c_u}$. By substituting $\sqrt{\sum_{s_i \in S}(r_{ui} - \bar{r}_u)^2}$ with $\omega_u / \sqrt{c_u}$, equation 1 can be rewritten as follows:

$$sim(u,v) = \frac{\sum_{s_i \in S}(q_{ui} - \overline{q}_u)(q_{vi} - \overline{q}_v)}{\omega_u \omega_v \sqrt{c_u c_v}} \tag{4}$$

Also, we have $q_{ui} - \overline{q}_u = \hat{q}_{ui} / \omega_u$ according to equation 3, so the above equation can be further rewritten as follows:

$$sim(u,v) = \frac{\sum_{s_i \in S} \hat{q}_{ui} \hat{q}_{vi}}{\sqrt{c_u c_v}} \tag{5}$$

Now we are in the position to compute $Sim(u,v)$ on the disguised QoS values Q_u and Q_v. By noting that noises are added into the normalized (original) QoS values, we evaluate the product of Q_u and Q_v as follows:

$$\begin{aligned} Q_u Q_v &= \sum\nolimits_{i=1}^{m} Q_{ui} Q_{vi} \\ &= \sum\nolimits_{i=1}^{m} (\hat{q}_{ui} + L(\Delta f_{\hat{q}_u} / \epsilon_{\hat{q}_u}))(\hat{q}_{vi} + L(\Delta f_{\hat{q}_v} / \epsilon_{\hat{q}_v})) \\ &= \sum\nolimits_{i=1}^{m} (\hat{q}_{ui} \hat{q}_{vi} + L(\Delta f_{\hat{q}_u} / \epsilon_{\hat{q}_u}) L(\Delta f_{\hat{q}_v} / \epsilon_{\hat{q}_v})) \\ &+ \sum\nolimits_{i=1}^{m} (\hat{q}_{ui} L(\Delta f_{\hat{q}_v} / \epsilon_{\hat{q}_v}) + \hat{q}_{vi} L(\Delta f_{\hat{q}_u} / \epsilon_{\hat{q}_u})) \end{aligned} \tag{6}$$

As $\hat{q}_{ui}$ and $L(\Delta f_{\hat{q}_v} / \epsilon_{\hat{q}_v})$ are independent and as $L(\Delta f_{\hat{q}_v} / \epsilon_{\hat{q}_v})$ is symmetric exponential distribution with $\mu = 0$, we have $\sum \hat{q}_{ui} L(\Delta f_{\hat{q}_v} / \epsilon_{\hat{q}_v}) \approx 0$. Likewise, we have $\sum \hat{q}_{vi} L(\Delta f_{\hat{q}_u} / \epsilon_{\hat{q}_u}) \approx 0$ and $\sum L(\Delta f_{\hat{q}_u} / \epsilon_{\hat{q}_u}) L(\Delta f_{\hat{q}_v} / \epsilon_{\hat{q}_v}) \approx 0$. Therefore, the following equation holds when the number of Web services (i.e., the size of Q_u) in the recommendation is significantly large:

$$Q_u Q_v \approx \sum \hat{q}_{ui} \hat{q}_{vi} = \hat{q}_u \hat{q}_v \tag{7}$$

Consequently, the similarity $Sim(u,v)$ calculated on the disguised QoS values approximates the similarity $sim(u,v)$ calculated on the original QoS values, as shown in the following equation:

$$Sim(u,v) = \frac{\sum_{s_i \in S} Q_{ui} Q_{vi}}{\sqrt{c_u c_v}} \approx \frac{\sum_{s_i \in S} \hat{q}_{ui} \hat{q}_{vi}}{\sqrt{c_u c_v}} = sim(u,v) \tag{8}$$

The above procedure is depicted in Fig 2. In summary, every user does z-score normalization on his/her observed QoS values over m services and then injects suitable noise into the normalized data according to the global sensitivity obtained based on his/her local data. Data utility can be largely preserved due to the fact that the aggregation of disguised QoS values approximates the aggregation of original QoS values. Therefore effective QoS prediction can be made while keeping individual QoS data private.

3.2.2 Model-based Method

Model-based approaches learn some models to fit the observed QoS data, and these models can be further used to predict the unknown QoS values. Matrix factorization (MF) [34] is a typical solution of model-based approaches which has been proved to be effective to improve the accuracy of prediction by learning latent factor models.

Recall that the observed QoS values of n users over m services are denoted by a matrix $Q_{n\times m}$. Taking it as input, MF aims to factorize this user-service QoS matrix into two latent matrices of a lower dimension d: a user-factor matrix $U_{n\times d}$ and a service-factor matrix $V_{m\times d}$. Then, vacant elements in $Q_{n\times m}$ can be approximated as the corresponding elements in the product of U and V, i.e., unknown QoS value q_{ui} is evaluated by $P(q_{ui}) = U_u \cdot V_i^T$. In other words, we can consider q_{ui} as the sum of $P(q_{ui})$ and δ_{ui} where δ_{ui} is regarded as the approximation error.

MF is typically transformed into an optimization problem, and a local optimal solution can be obtained by iteration. The objective function (or loss function) of MF is defined as:

$$min_{U,V} \sum_{q_{ui}\in Q} [(q_{ui} - U_u V_i^T)^2 + \lambda(||U_u||^2 + ||V_i||^2)] \tag{9}$$

The first part in the objective function is the squared difference between the existing QoS matrix and the predicted QoS matrix, but only for the elements that have been evaluated by users. The second part in the objective function is the regularization term, added to deal with overfitting caused by the sparsity of input matrix. By dealing with this optimization, we can finally obtain a user-factor matrix $U_{n\times d}$ and a service-factor matrix $V_{m\times d}$.

In the context of Web services QoS prediction, however, some other factors from both users and services may affect the predicted QoS values. To capture these factors, we adopt a biased matrix factorization model introduced in [57]:

$$q_{ui} = b_u + b_i + q'_{ui} + \delta_{ui} \tag{10}$$

where b_u is the user bias and b_i is the service bias. In order to reduce the effect of noise added to the original QoS values, we set $b_u = \bar{q}_u$. Within this model, the disguised QoS value Q_{ui} can be expressed as follows:

$$Q_{ui} = b_i + P(q_{ui}) + \delta_{ui} + L(\Delta f/\epsilon) \tag{11}$$

Then, we obtain the following new loss function:

$$min_{U,V} \sum_{Q_{ui}\in Q} [(Q_{ui} - b_i - U_u V_i^T)^2 + \lambda(||U_u||^2 + ||V_i||^2 + ||b_i||^2)] \tag{12}$$

At last, the predicted QoS values can be decided by combining b_i, U_i, V_j as follows:

$$P(Q_{ui}) = b_i + U_u V_i^T \tag{13}$$

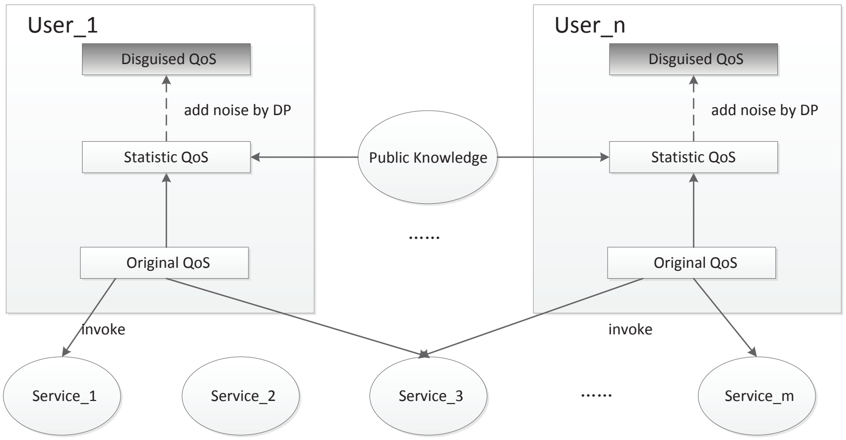


Fig. 3 Data Disguising by Adding Noise to Aggregated QoS

3.3 DPA: Differential Privacy on Aggregated Data

By adding noise to his/her local QoS data, every user can protect his/her sensitive data by differential privacy. This method is simple, however, the injected noise depends only on the sensitivity of his/her local QoS data, which sometimes is too large, resulting in a huge noise compared with the original data. This will decrease the data utility significantly, making prediction result unsatisfactory. To overcome this weakness, a possible solution is to add noise to the aggregation of user's QoS data while keeping the sensitivity of the aggregation low. We observe that, given a user, his/her observed QoS value, say the response time of a Web service, largely depend on where the service locates. For instance, a user in China generally has a better experience (i.e., quicker response time) when using services located in China than those in USA. In other words, services in the same country are more likely to have similar QoS values for the same user due to the similar network environment. Therefore, we decide to aggregate the QoS values of services in the same country and apply differential privacy on this these aggregated values. We will see later the effectiveness of this method called DPA as shown in the experiments on real Web services dataset. Figure 3 shows the basic idea of DPA and below we introduce the details of DPA which disguises original QoS values based on differential privacy on aggregated data.

To perform QoS aggregation over countries, we first introduce an $m \times k$ service-country matrix C where m is the number of services and k is the number of countries. The element c_{ij} in C indicates whether service i locates in the country j. If $c_{ij} = 1$, then Web service i belongs to country j. Let A_u denote the aggregated QoS vector for user u. Clearly, its size is k as there are k countries. Further, we consider a particular aggregation function, *sum*, to aggregate multiple QoS values. More specifically, the aggregated QoS vector A_u can be calculated as follows:

$$A_{uj} = \sum_{i \in S} q_{ui} c_{ij} + L(\Delta f_j / \epsilon) \tag{14}$$

where A_{uj} is the j^{th} element in A_u, Δf_j is the sensitivity of service QoS values over country j. It is easy to see that $\Delta f_j = max_{i \in S}\{q_{ui} c_{ij}\}$. Note that, this aggregated value cannot be sent to the recommender directly. Instead, it should be split into $\sum_{i \in S} c_{ij}$ pieces, each of which is a disguised QoS value. By considering the whole vector A_u, our objective is to reform a disguised QoS vector Q_u for the original QoS vector q_u. In particular, the value of $||Q_u C - A||_2$ should be as small as possible. Let l and r denote two error vectors, our problem is formally defined as follows:

$$\begin{aligned} &minimize: && \frac{1}{2}||l + r||_2^2 \\ &subject\ to: && -l \leq Q_u C - A \ \leq r \end{aligned} \tag{15}$$

The disguised QoS vector Q_u can be obtained by solving the above problem. Then it can be fed into memory-based or model-based collaborative QoS prediction algorithms discussed earlier.

Algorithm 1: Disguising QoS values using differential privacy on aggregated data.

```
   Data: original QoS vector $q_u$ of user $u$, service-country matrix $C$;
   Result: disguised QoS vector $Q_u$ of user $u$;
 1 for each country $j$ do
 2     $\Delta f_j = 0$;
 3     $A_{uj} = 0$;
 4     for each service $i$ do
 5         if $q_{ui} C_{ij} > \Delta f_j$
 6         $\Delta f_j = q_{ui} C_{ij}$;
 7         $A_{uj} = A_{uj} + q_{ui} C_{ij}$;
 8     end
 9     $A_{uj} = A_{uj} + L(\Delta f_j / \epsilon)$;
10 end
11 solve the problem defined in equation 15;
12 return $Q_u$;
```

Algorithm 1 shows the procedure of disguising QoS vectors using differential privacy on aggregated data. This algorithm takes as input the user u's original QoS vector and the service-country matrix and returns the disguised QoS vector Q_u as the output. In lines 4 - 8, for each country j, it calculates the aggregated QoS value for each country and gets the sensitivity according to these values. Noises are then added to each aggregated QoS value but split again by solving the problem defined in equation 15.

Publishing Q_u for every user u using the above algorithm satisfies ϵ-differential privacy. The reason is as follows. First, we note that noises are introduced only in Line 9 for every user u and the added noises satisfy Laplace mechanism. Therefore, ϵ-differential privacy holds for every user u. Further, the result of $Q_1, Q_2, \cdots, Q_n$ can be seen as the parallel execution of Algorithm 1 on n users, so publishing them satisfies ϵ-differential privacy according to Theorem 4.

Table 2 User-Service Matrix

	s_1	s_2	s_3	s_4	s_5	s_6	s_7
u_1	0.62	0.64	14.28	113.92	125.00	23.75	1.64
u_2	0.58	0.66	7.80	78.30	56.22	15.56	0.75
u_3	1.77	2.56	2.56	87.20	102.50	4.77	1.45

Table 3 Service-Country Matrix

	USA	China	France
s_1	1	0	0
s_2	1	0	0
s_3	0	1	0
s_4	0	0	1
s_5	0	0	1
s_6	0	1	0
s_7	1	0	0

We give a simple numerical example to illustrate our DPA algorithm. Tables 2 and 3 show a user-service matrix and a service-country matrix, respectively. We now consider the case where user u_1 wants to send his/her observed QoS values to the recommender in a privacy-preserving manner. As shown in Table 2, his/her QoS vector is $q_{u_1} = \{0.62, 0.64, 14.28, 113.92, 125.00, 23.75, 1.64\}$. Multiplying q_{u_1} by the service-country matrix C, we obtain the aggregated QoS vector $\{2.90, 38.03, 238.92\}$, where each element represents an aggregated QoS value for a specific country. Meanwhile, we have different sensitivities for different aggregated values. In this case, these sensitivities are $\{1.64, 23.75, 125\}$ respectively. Based on these sensitivities and the given privacy budget ($\epsilon = 1$), we add noises to the aggregated QoS values to achieve ϵ-differential privacy. Here, the disguised aggregated QoS vector is $A_u = \{1.89, 34.05, 251.50\}$. Finally we solve the problem defined in equation 15 to obtain the final disguised QoS vector $Q_u = \{0.63, 0.63, 17.02, 125.75, 125.75, 17.02, 0.63\}$. On the other hand, by using the proposed DPS algorithm, we can get the disguised QoS vector $Q'_u = \{9.11, 48.68, 277.25, 288.15, 138.21, 58.87, 25.54\}$. The utility of these two disguised vectors can be evaluated to some extent by Mean Absolute Error (MAE) which is defined to be the average of the absolute difference between the disguised value and the original value. After some simple calculation, we can learn that the MAE of Q'_u is 80.85 while that of Q_u is only 3.3. This simple example shows DPA can achieve a higher utility at the same privacy budget than DPS. In the next section, we will see this conclusion is supported by more results on real Web services QoS dataset.

4 Experiments

In this section, we conduct extensive experiments on a real Web services QoS dataset to evaluate our methods for differential private collaborative Web services QoS prediction. Firstly, we investigate the influence of z-score normalization during the process of QoS prediction. Then, a series of experiments study the trade-off between privacy and accuracy compared with the basic method and the advanced approach. The other experiments investigate the effects of the performance of the advanced approach on some important data features including the different scale of services and users.

4.1 Experimental Setup

A real Web services QoS dataset was introduced in [68], which includes QoS values of 5,825 real-world Web services observed by 339 users. This dataset is quite useful when studying the accuracy of QoS prediction. According to the dataset, we focus on two representative QoS attributes: response time (RT) and throughput (TP). Response time measures the time duration between user sending a request and receiving a response, while throughput stands for the data transmission rate of a user invoking a service. Table 4 describes the statistics of the dataset, where AVE and STD is the average and standard deviation of data respectively, density means the ratio of observed data to all data. More details of the dataset can be found in [68].

Table 4 Statistic of Dataset

QoS	USER	SERVICE	AVE	STD	DENSITY
RT(sec)	339	5825	0.90	1.973	94.8%
TP(kpbs)	339	5825	47.56	110.797	92.7%

We use cross validation to train and evaluate our QoS prediction methods. To simulate data sparsity in practice, we randomly remove entries from the full dataset and only keep a small density of historical QoS values as our training set. The removed data is treated as the testing set for evaluating the accuracy of our prediction methods.

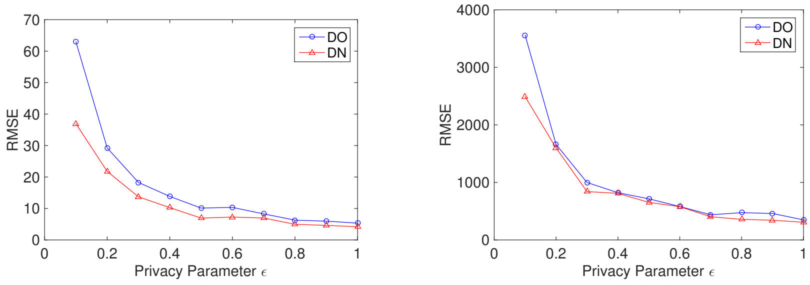
To quantize the accuracy of QoS prediction, we adopt Root Mean Square Error (RMSE) as it has been widely used in related studies (e.g., [1,32]):

$$RMSE = \sqrt{\frac{\sum_{t \in T}(R(t) - P(t))^2}{|T|}}$$

where T consists of all the QoS values needed to be predicted in the training set, $R(t)$ is t's real value that is available in the testing set and $P(t)$ is the predicted value of t generated by our prediction algorithms. Generally, a smaller RMSE indicates a better prediction.

4.2 Effect of Z-score Normalization

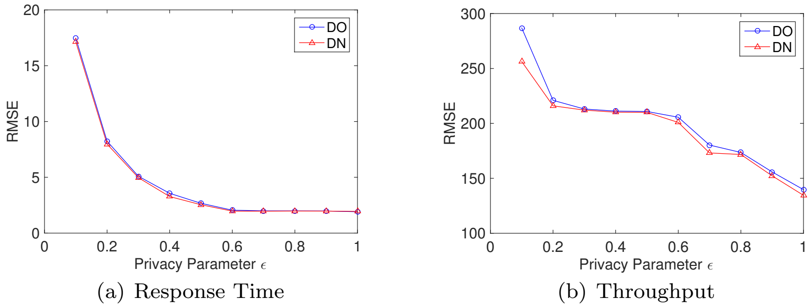
From the discussion in Section 3.2, noise can be added to the original QoS vector or the normalized QoS vector. Here, we use DO to denote the way of adding noise to the original QoS vector and DN to denote the way of adding noise to the normalized QoS vector. Fig 4 shows the prediction result of memory-based approach under different ways of adding noise. In Fig 4(a), the result of DN is better than DO, especially when the privacy budget ϵ is less than 0.7, which shows that z-score normalization can improve the utility of disguised QoS data. In Fig 4(b), we observe again that DN outperforms DO for the QoS metric throughput as it results in less RMSE. Therefore, we can conclude that using z-score normalization before adding noises can improve the prediction accuracy. We also study the prediction result of model-based approach under these two ways of adding noise. From Fig 11, it is clear that adding noise to the normalized QoS data leads to a more accurate prediction, no matter which kind of QoS metric is considered. In summary, doing z-score normalization on the original QoS data can improve the utility of disguised data.



(a) Response Time (b) Throughput

Fig. 4 Effect of z-score normalization on memory-based approach

The above experiment shows that z-score normalization on the user side can improve the utility of disguised QoS data. We note that this normalization can be also done at the recommender side. It is therefore interesting to investigate whether or not data utility can be further improved by doing normalization in two sides. To facilitate subsequent discussion, we use N-N to denote neither users nor the recommender performs z-score normalization, Z-N to denote only users perform z-score normalization, N-Z to denote only the



(a) Response Time

(b) Throughput

Fig. 5 Effect of z-score normalization on model-based approach

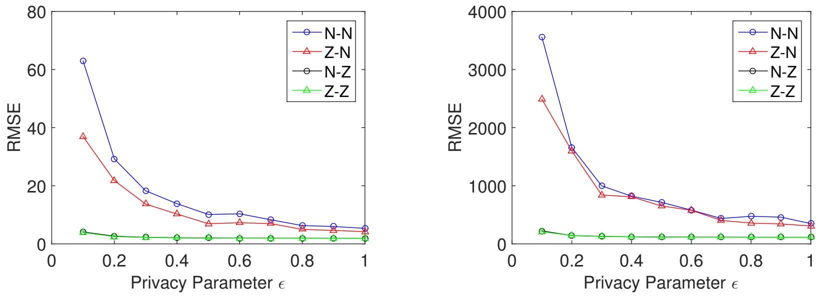
recommender performs z-score normalization, and Z-Z to denote both users and the recommender perform z-score normalization.

From Fig 6(a), it is clear that z-score normalization performed at the recommender side can greatly improve the prediction accuracy over the QoS metric RT, no matter whether or not users do z-score normalization at their sides. In particular, the superiority of z-score normalization at the recommender side is more obvious when the privacy parameter ϵ is smaller, say less than 0.4. The reason is that z-score normalization eliminates the difference between users' data, so that the similarities between users can be estimated more accurately. For the QoS metric TP, we can obtain the same conclusion from Fig 6(b). Therefore, it is useful for the recommender to normalize the disguised QoS data when using the memory-based approach.

For the model-based approach, we also find that z-score normalization at the recommender side is significant, as shown in Fig 7(a) for the QoS metric RT and Fig 7(b) for the QoS metric TP. An interesting point here is z-score normalization at the recommender side can bring more benefits for TP than RT. The reason is that the QoS data for TP are spread out over a wider range of values (this can be observed from the statistic of the dataset shown in Table 1), so z-score normalization has a better effect on TP than RT.

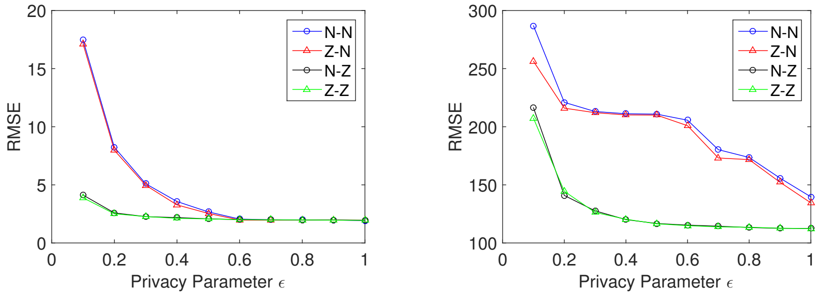
Based on the above results on two collaborative QoS prediction approaches, we can conclude that z-score normalization, especially at the recommender side, is beneficial to the utility of disguised data, which in fact determines the accuracy of QoS prediction.

Another interesting observation is z-score normalization also affects the running time of model-based approach. Figures 8(a) and 8(b) show the running time of model-based approach on RT and TP, respectively. Because model-based approach itself is an optimization problem, it is common for the running time to fluctuate. The effect of z-score normalization at user side on the running time is small, as the curves of N-N and Z-N are close and the curves of N-Z and Z-Z are close too. However, z-score normalization at the recommender side has a big influence on the running time, in particular, it reduces the running time significantly. This is because after z-score normalization the



(a) Response Time (b) Throughput

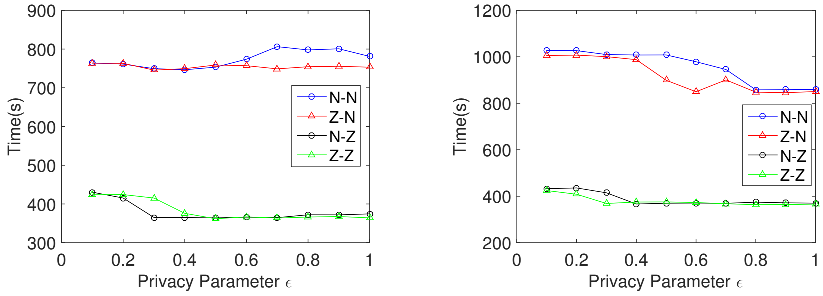
Fig. 6 Effect of two-sides z-score normalization on memory-based approach



(a) Response Time (b) Throughput

Fig. 7 Effect of two-sides z-score normalization on model-based approach

range of each element in the user-service matrix becomes smaller, so the goal of optimization can be faster to achieve.

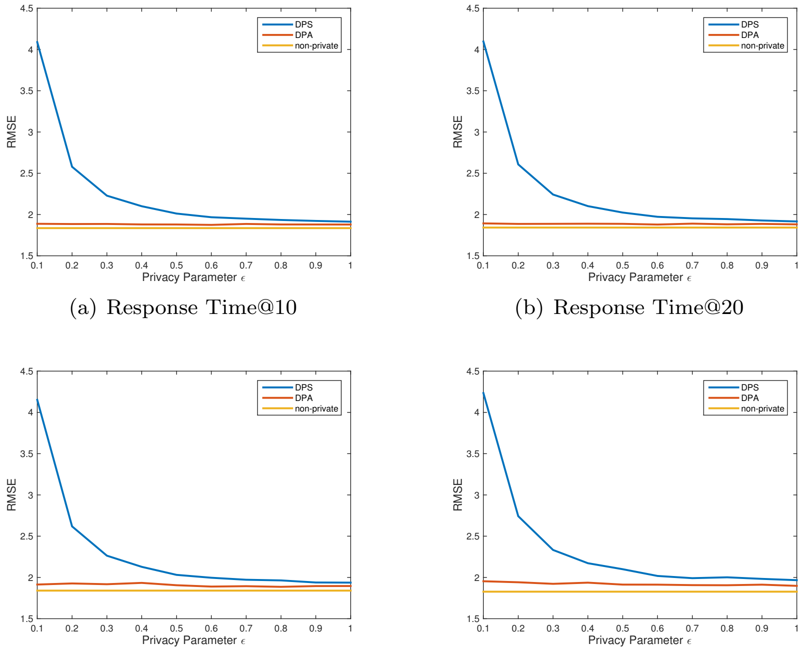


(a) Response Time (b) Throughput

Fig. 8 Effect of two-sides z-score normalization on running time of model-based approach

4.3 Privacy vs Accuracy

From the experiment results in Figures 9 and 10, we first find that the prediction accuracy of DPA and DPS both decrease when ϵ gets larger. This is because a larger ϵ means a looser privacy constraint, which has a smaller impact on the utility of the disguised QoS data. However, DPA achieves better prediction accuracy compared with DPS at the same privacy budget and the same number of recommendations. For small privacy budget say $\epsilon < 0.5$, DPA is much better than DPS, which means we should choose DPA to add noise when providing strong privacy guarantee.



(a) Response Time@10

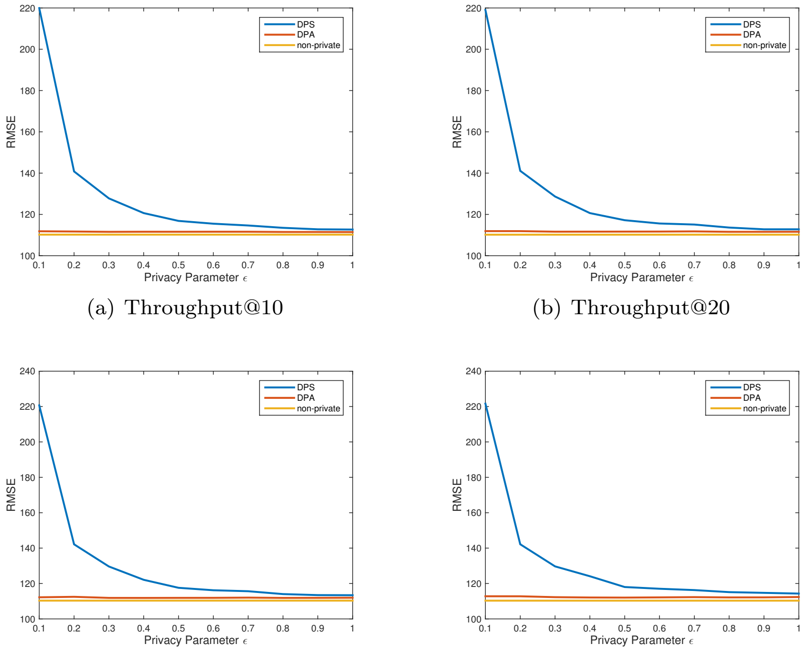
(b) Response Time@20

(c) Response Time@50

(d) Response Time@100

Fig. 9 DPS v.s. DPA on memory-based approach for RT

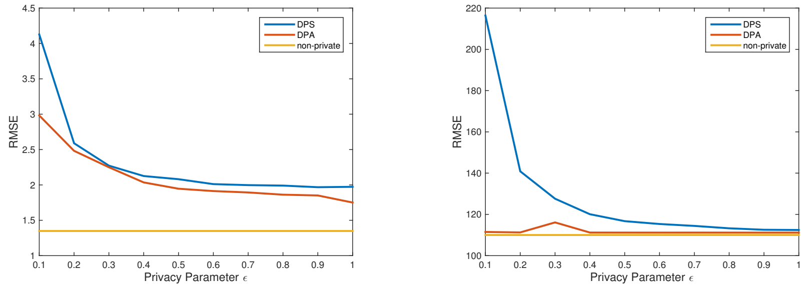
Fig 11 shows the performance of DPS and DPA on model-based approach. Obviously, with the same privacy parameter, more accurate prediction can be made by using DPA than DPS. For the QoS metric RT shown in Fig 11(a), the performance of DPA is slightly better than DPS, which is different from the memory-based approach where DPA is much better than DPS. The reason is that the distribution of RT is more balanced than TP, so the disguised QoS vectors are less influenced by these two methods. Fig 11(b) shows the weakness of DPS can be effectively addressed by DPA, that is, using DPA can significantly improve data utility.



(a) Throughput@10 (b) Throughput@20

(c) Throughput@50 (d) Throughput@100

Fig. 10 DPS v.s. DPA on memory-based approach for TP

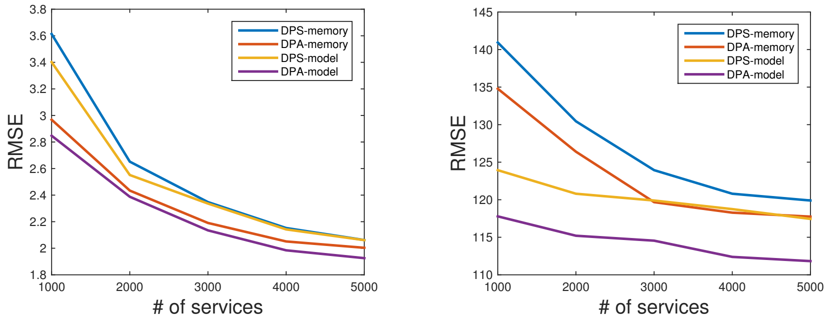


(a) Response Time (b) Throughput

Fig. 11 DPS v.s. DPA on model-based approach

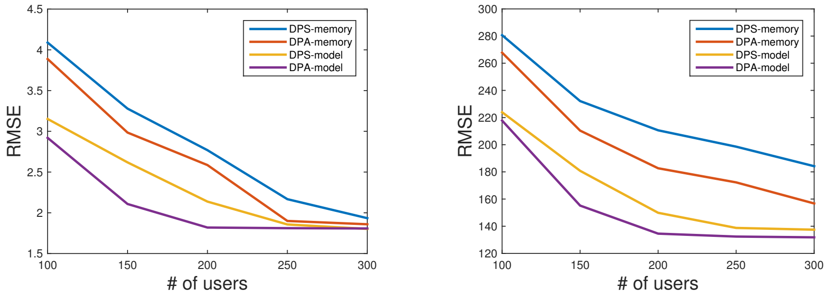
4.4 Influence of Data Size

The influence of data size on different methods (i.e., different data disguising methods plus different collaborative QoS prediction approaches) is also investigated here. In Fig 12, the number of users is set to 339 and the number of service is varying from 1000 to 5000 with a step 1000. Without losing generality, the services are selected randomly from the original dataset. It is obvious that a better prediction result can be obtained on a larger dataset, as more data can be used for recommendation. Meanwhile, DPA is better than DPS despite of data size and model-based approach outperforms memory-based approach. In Fig 13, the number of services is set to be 5825 and the number of users varies from 100 to 300 stepped by 50. Clearly, the number of users has a positive influence on the accuracy of all algorithms, which again means the more data are given, the better prediction can be made. Though prediction accuracy differs significantly on different data size, DPA is still better than DPS as it has a smaller influence on data utility.



(a) Response Time (b) Throughput

Fig. 12 Effect of number of services on different methods



(a) Response Time (b) Throughput

Fig. 13 Effect of number of users on different methods

5 Related Work

As mentioned in the introduction, many researches focused on improving the accuracy of Web services recommendation. Yao et al. [56] propose a hybrid approach that combines collaborative filtering and content-based recommendation for Web services. In [66], the authors combine user-based and item-based collaborative filtering to make accurate Web services QoS recommendation. Noting that different people have different requirements in the course of Web services recommendation, the authors in [11] take personalized preference into account when making recommendation.

Privacy preserving has aroused the concern of many people with the promotion of the QoS prediction. To enable collaborative QoS prediction to work well, the recommender system is used to collect the observed QoS values of the services which users have invoked. However, the QoS values may contain personal sensitive information so that users are unwilling to share them directly. Some researches point out that some QoS properties like response time and availability highly related to the users' physical locations [3,51,43,15,62,61, 63,64], which means that the users' location could be deduced from the QoS information.

In fact, there are a lot of researches using homomorphic encryption as a commonly used encryption method [9,27,2,24,26,14,54,38,46,45]. Homomorphic encryption [12] which allows computation carried out on ciphertext is a straightforward way to achieve privacy. Li et al. [18] propose a privacy-preserving collaborative QoS prediction framework via Yao's garbled circuit and homomorphic encryption, which protects the private data of users and retain the ability of generating accurate QoS prediction. In [10], the authors present a solution for privacy preserving recommendation via homomorphic encryption and data packing. Nikolaenko et al. [35,36,5,4] propose solutions on model-based recommendation algorithms: matrix factorization and ridge regression via Yao's protocol and homomorphic encryption.

Though these researches come up with secure multi-party computation protocols to afford strong privacy guarantee, it is obviously that the homomorphic encryption consumes plenty computation and communication. Hence, it is infeasible to deal with our problem by the usage of homomorphic encryption.

Sweeney [49] proposes the approach named k-anonymity which provides each record is indistinguishable from at least $k-1$ other records in a k-anonymized dataset. However, this approach can not protect privacy when the diversity of sensitive attributes is small or attackers have auxiliary information [19]. Another way to deal with the problem of privacy disclosure is randomized perturbation. Polat et al. [37] claim that accurate recommendation could still be obtained while randomness from a specific distribution are added to the original data to prevent information leakage. A recent work [69] uses the randomized perturbation as the data obfuscation techniques to form a simple yet effective privacy-preserving Web service recommendation framework. However, the range of randomness is chosen by experience and does not have provable privacy guarantee. What's worse, it is recognized that with the application of clustering on the perturbed data, adversaries can infer users' private data to a large extent [60].

Differential Privacy (DP) [6] is a recently proposed privacy model that can guarantee strong privacy independent of the available auxiliary information of the attackers. There are also a number of achievements that aim at privacy preserving recommender systems under the differential privacy paradigm. McSherry and Mironov [32] integrate differential privacy into non-social recommender systems. However, their work will lead to an unacceptable loss of utility when applied to the social recommendation. To overcome this weakness, Jorgensen and Yu [16] incorporate a clustering procedure that groups users according to the natural community structure of the social network and significantly reduces the amount of noises. Guerraoui et al. [13] propose a distance-based differential privacy framework as an extension of the notion of differential privacy to ensure this strong form of privacy.

Different from our work, these mentioned works are interested in preserving the collected users' data. There are also some works focusing on the privacy of user's data before being collected. [30] has presented a hybrid approach for privacy-preserving recommender system by combining randomized perturbation and differential privacy, the privacy of user's data is protected by randomized perturbation and the privacy of recommender results is guaranteed by the differential privacy. The approach proposed in this paper to perturb users' data is first introduced in [48], Shen et al. design a novel and practical privacy-built-in client under untrusted server settings, in which user's data are perturbed and anonymized on their private devices. They think whether rating is more important for specific scoring data. Based on this consideration, their work focus on the users history data. Nevertheless, our research concentrates on the values of QoS, which describe the quality of Web services.

6 Conclusion

In this paper, we have proposed a privacy-preserving solution to collaborative Web services QoS prediction via differential privacy. Compared with methods built on conventional public-key cryptosystems, our solution is lightweight but its security can be theoretically guaranteed as differential privacy gives a rigorous and quantitative definition on privacy leakage. To disguise users' QoS data, we have designed DPS which adds noise to single data and DPA which adds noise to aggregated data. We have also shown how to perform collaborative QoS prediction on disguised QoS data. Based on a real Web services QoS dataset, we have studied the performance of the proposed methods. Empirical results have shown that both methods can protect users' private QoS data with acceptable prediction accuracy loss, and DPA is superior to DPS as better data utility can be achieved on the disguised QoS data.

Acknowledgment

Research reported in this publication was partially supported Natural Science Foundation of China (Grant Nos. 61572336, 61572335, 61402313)

References

1. A. Berlioz, A. Friedman, M. A. Kaafar, R. Boreli, and S. Berkovsky. Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 107–114. ACM, 2015.
2. J. Canny. Collaborative filtering with privacy. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 45–57. IEEE, 2002.
3. X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun. Personalized qos-aware web service recommendation and visualization. *IEEE Transactions on Services Computing*, 6(1):35–47, 2013.
4. Z. Ding, B. Yang, Y. Chi, and L. Guo. Enabling smart transportation systems: A parallel spatio-temporal database approach. *IEEE Trans. Computers*, 65(5):1377–1391, 2016.
5. Z. Ding, B. Yang, R. H. Güting, and Y. Li. Network-matched trajectory-based moving-object database: Models and applications. *IEEE Trans. Intelligent Transportation Systems*, 16(4):1918–1928, 2015.
6. C. Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.
7. C. Dwork. Differential privacy. encyclopedia of cryptography and security, 2011.
8. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.
9. Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE transactions on information forensics and security*, 7(3):1053–1066, 2012.
10. Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE transactions on information forensics and security*, 7(3):1053–1066, 2012.
11. K. K. Fletcher and X. F. Liu. A collaborative filtering method for personalized preference-based service recommendation. In *Web Services (ICWS), 2015 IEEE International Conference on*, pages 400–407. IEEE, 2015.
12. C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.
13. R. Guerraoui, A.-M. Kermarrec, R. Patra, and M. Taziki. D 2 p: distance-based differential privacy in recommenders. *Proceedings of the VLDB Endowment*, 8(8):862–873, 2015.
14. C. Guo, C. S. Jensen, and B. Yang. Towards total traffic awareness. *SIGMOD Record*, 43(3):18–23, 2014.
15. C. Guo, B. Yang, O. Andersen, C. S. Jensen, and K. Torp. Ecosky: Reducing vehicular environmental impact through eco-routing. In *ICDE*, pages 1412–1415, 2015.
16. Z. Jorgensen and T. Yu. A privacy-preserving framework for personalized, social recommendations. In *EDBT*, pages 571–582, 2014.
17. D. Kifer and A. Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204. ACM, 2011.
18. L. Li, A. Liu, Q. Li, G. Liu, and Z. Li. Privacy-preserving collaborative web services qos prediction via yao's garbled circuits and homomorphic encryption. *JOURNAL OF WEB ENGINEERING*, 15(3-4):203–225, 2016.
19. N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.
20. A. Liu, Q. Li, L. Huang, and S. Wen. Shapley value based impression propagation for reputation management in web service composition. In *2012 IEEE 19th International Conference on Web Services, Honolulu, HI, USA, June 24-29, 2012*, pages 58–65, 2012.
21. A. Liu, Q. Li, L. Huang, and M. Xiao. Facts: A framework for fault-tolerant composition of transactional web services. *IEEE Transactions on Services Computing*, 3(1):46–59, 2010.
22. A. Liu, Q. Li, L. Huang, S. Ying, and M. Xiao. Coalitional game for community-based autonomous web services cooperation. *IEEE Transactions on Services Computing*, 6(3):387–399, 2013.

23. A. Liu, Q. Li, X. Zhou, L. Li, G. Liu, and Y. Gao. Rating propagation in web services reputation systems: A fast shapley value approach. In *Database Systems for Advanced Applications - 19th International Conference, DASFAA 2014, Bali, Indonesia, April 21-24, 2014. Proceedings, Part I*, pages 466–480, 2014.
24. A. Liu, Z. Li, G. Liu, K. Zheng, M. Zhang, Q. Li, and X. Zhang. Privacy-preserving task assignment in spatial crowdsourcing. *J. Comput. Sci. Technol.*, 32(5):905–918, 2017.
25. A. Liu, H. Liu, Q. Li, L.-S. Huang, and M.-J. Xiao. Constraints-aware scheduling for transactional services composition. *Journal of Computer Science and Technology*, 24(4):638–651, 2009.
26. A. Liu, W. Wang, S. Shang, Q. Li, and X. Zhang. Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. *GeoInformatica*, Aug 2017.
27. A. Liu, K. Zheng, L. Li, G. Liu, L. Zhao, and X. Zhou. Efficient secure similarity computation on encrypted trajectory data. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 66–77. IEEE, 2015.
28. J. Liu, K. Zhao, P. Sommer, S. Shang, B. Kusy, and R. Jurdak. Bounded quadrant system: Error-bounded trajectory compression on the go. In *ICDE*, pages 987–998, 2015.
29. J. Liu, K. Zhao, P. Sommer, S. Shang, B. Kusy, J. Lee, and R. Jurdak. A novel framework for online amnesic trajectory compression in resource-constrained environments. *IEEE Trans. Knowl. Data Eng.*, 28(11):2827–2841, 2016.
30. X. Liu, A. Liu, X. Zhang, Z. Li, G. Liu, L. Zhao, and X. Zhou. When differential privacy meets randomized perturbation: A hybrid approach for privacy-preserving recommender system. In *International Conference on Database Systems for Advanced Applications*, pages 576–591. Springer, 2017.
31. A. Machanavajjhala, A. Korolova, and A. D. Sarma. Personalized social recommendations: accurate or private. *Proceedings of the VLDB Endowment*, 4(7):440–450, 2011.
32. F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the net. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636. ACM, 2009.
33. F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30. ACM, 2009.
34. A. Mnih and R. R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.
35. V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh. Privacy-preserving matrix factorization. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 801–812. ACM, 2013.
36. V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft. Privacy-preserving ridge regression on hundreds of millions of records. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 334–348. IEEE, 2013.
37. H. Polat and W. Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 625–628. IEEE, 2003.
38. S. Shang, L. Chen, C. S. Jensen, J. Wen, and P. Kalnis. Searching trajectories by regions of interest. *IEEE Trans. Knowl. Data Eng.*, 29(7):1549–1562, 2017.
39. S. Shang, L. Chen, Z. Wei, C. S. Jensen, J.-R. Wen, and P. Kalnis. Collective travel planning in spatial networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(5):1132–1146, 2016.
40. S. Shang, L. Chen, Z. Wei, C. S. Jensen, K. Zheng, and P. Kalnis. Trajectory similarity join in spatial networks. *PVLDB*, 10(11):1178–1189, 2017.
41. S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng, and P. Kalnis. User oriented trajectory search for trip recommendation. In *Proceedings of the 15th International Conference on Extending Database Technology*, pages 156–167. ACM, 2012.
42. S. Shang, R. Ding, K. Zheng, C. S. Jensen, P. Kalnis, and X. Zhou. Personalized trajectory matching in spatial networks. *The VLDB Journal*, 23(3):449–468, 2014.
43. S. Shang, D. Guo, J. Liu, K. Zheng, and J. Wen. Finding regions of interest using location based social media. *Neurocomputing*, 173:118–123, 2016.
44. S. Shang, J. Liu, K. Zheng, H. Lu, T. B. Pedersen, and J. Wen. Planning unobstructed paths in traffic-aware spatial networks. *GeoInformatica*, 19(4):723–746, 2015.

45. S. Shang, H. Lu, T. B. Pedersen, and X. Xie. Finding traffic-aware fastest paths in spatial networks. In *SSTD*, pages 128–145, 2013.
46. S. Shang, H. Lu, T. B. Pedersen, and X. Xie. Modeling of traffic-aware travel time in spatial networks. In *MDM*, pages 247–250, 2013.
47. S. Shang, K. Zheng, C. S. Jensen, B. Yang, P. Kalnis, G. Li, and J. Wen. Discovery of path nearby clusters in spatial networks. *IEEE Trans. Knowl. Data Eng.*, 27(6):1505–1518, 2015.
48. Y. Shen and H. Jin. Epicrec: Towards practical differentially private framework for personalized recommendation. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 180–191. ACM, 2016.
49. L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
50. M. Tang, Y. Jiang, J. Liu, and X. Liu. Location-aware collaborative filtering for qos-based service recommendation. In *Web Services (ICWS), 2012 IEEE 19th International Conference on*, pages 202–209. IEEE, 2012.
51. M. Tang, Y. Jiang, J. Liu, and X. Liu. Location-aware collaborative filtering for qos-based service recommendation. In *Web Services (ICWS), 2012 IEEE 19th International Conference on*, pages 202–209. IEEE, 2012.
52. K. Xie, K. Deng, S. Shang, X. Zhou, and K. Zheng. Finding alternative shortest paths in spatial networks. *ACM Trans. Database Syst.*, 37(4):29:1–29:31, 2012.
53. Q. Xie, S. Shang, B. Yuan, C. Pang, and X. Zhang. Local correlation detection with linearity enhancement in streaming data. In *CIKM*, pages 309–318, 2013.
54. B. Yang, J. Dai, C. Guo, and C. S. Jensen. Pace: A PAth-CEntric paradigm for stochastic path finding. *VLDB Journal*, online first, 2017.
55. B. Yang, C. Guo, C. S. Jensen, M. Kaul, and S. Shang. Stochastic skyline route planning under time-varying uncertainty. In *ICDE*, pages 136–147, 2014.
56. L. Yao, Q. Z. Sheng, A. Segev, and J. Yu. Recommending web services via combining collaborative filtering with content-based features. In *Web Services (ICWS), 2013 IEEE 20th International Conference on*, pages 42–49. IEEE, 2013.
57. D. Yu, Y. Liu, Y. Xu, and Y. Yin. Personalized qos prediction for web services using latent factor models. In *Services Computing (SCC), 2014 IEEE International Conference on*, pages 107–114. IEEE, 2014.
58. Q. Yu, Z. Zheng, and H. Wang. Trace norm regularized matrix factorization for service recommendation. In *Web Services (ICWS), 2013 IEEE 20th International Conference on*, pages 34–41. IEEE, 2013.
59. Q. Zhang, C. Ding, and C.-H. Chi. Collaborative filtering based service ranking using invocation histories. In *Web Services (ICWS), 2011 IEEE International Conference on*, pages 195–202. IEEE, 2011.
60. S. Zhang, J. Ford, and F. Makedon. Deriving private information from randomly perturbed ratings. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 59–69. SIAM, 2006.
61. K. Zheng, S. Shang, N. J. Yuan, and Y. Yang. Towards efficient search for activity trajectories. In *ICDE*, pages 230–241, 2013.
62. K. Zheng, H. Su, B. Zheng, S. Shang, J. Xu, J. Liu, and X. Zhou. Interactive top-k spatial keyword queries. In *ICDE*, pages 423–434, 2015.
63. K. Zheng, Y. Zheng, N. J. Yuan, and S. Shang. On discovery of gathering patterns from trajectories. In *ICDE*, pages 242–253, 2013.
64. K. Zheng, Y. Zheng, N. J. Yuan, S. Shang, and X. Zhou. Online discovery of gathering patterns over trajectories. *IEEE Trans. Knowl. Data Eng.*, 26(8):1974–1988, 2014.
65. Z. Zheng, H. Ma, M. R. Lyu, and I. King. Wsrec: A collaborative filtering based web service recommender system. In *Web Services, 2009. ICWS 2009. IEEE International Conference on*, pages 437–444. IEEE, 2009.
66. Z. Zheng, H. Ma, M. R. Lyu, and I. King. Qos-aware web service recommendation by collaborative filtering. *IEEE Transactions on services computing*, 4(2):140–152, 2011.
67. Z. Zheng, H. Ma, M. R. Lyu, and I. King. Collaborative web service qos prediction via neighborhood integrated matrix factorization. *IEEE Transactions on Services Computing*, 6(3):289–299, 2013.

68. Z. Zheng, Y. Zhang, and M. R. Lyu. Distributed qos evaluation for real-world web services. In *Web Services (ICWS), 2010 IEEE International Conference on*, pages 83–90. IEEE, 2010.

69. J. Zhu, P. He, Z. Zheng, and M. R. Lyu. A privacy-preserving qos prediction framework for web service recommendation. In *Web Services (ICWS), 2015 IEEE International Conference on*, pages 241–248. IEEE, 2015.