Privacy-Preserving Collaborative Web Services QoS Prediction via Differential Privacy

Shushu Liu, An Liu*, Zhixu Li, Guanfeng Liu,
Jiajie Xu, Lei Zhao, and Kai Zheng

School of Computer Science and Technology, Soochow University, China
anliu@suda.edu.cn

Abstract. Collaborative Web services QoS prediction has become an important tool for the generation of accurate personalized QoS. 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 preserving technique, into the preprocess of QoS data prediction. We implement the proposed approach based on a general approach named Laplace mechanism and conduct extensive experiments to study its performance on a real world dataset. The experiments evaluate the privacy-accuracy trade-off on different settings and show that under some constraint, our proposed approach can achieve a better performance than baselines.

Keywords: collaborative QoS prediction; privacy-preserving; differential privacy; data distribution

1 Introduction

Quality of service (QoS) has been widely used for describing nonfunctional characteristics of web services. QoS-based Web services selection, composition and recommendation [1, 9, 10] have been discussed extensively in the recent literature. 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 service providers or third-party communities are not accurate to service users, as they are susceptible to the uncertain Internet environment and user context. On the other hand, it is impractical for service 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 [24, 22] which draw from personalized recommendation [18, 17]

* Corresponding author

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 based on the QoS values of the service observed by the similar users to this particular user. 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 depends on their specific context. Based on these provided QoS values, a variety of techniques have been employed to improve the quality especially accuracy of prediction [21, 19].

Though many achievements have been attainted 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 observed QoS values could be sensitive information, so users may not be willing to share them with others. For example, the observed response time reported by a user typically depends on her location [19], which means that the user's location could be deduced from the QoS information she provided. Consequently, an interesting but challenging question is whether or not a recommender system can make accurately personalized QoS prediction for users while protecting their privacy.

Homomorphic encryption [8] which allows computations to be carried out on ciphertext is a straightforward way to achieve privacy. However, all these operations require not only a large computation cost [7, 11], but also sustained communication between parties [3]. Not even to mention the difficulty to apply some complicated computations into the encrypted domain. Hence, it is infeasible to deal with our problem by the usage of Homomorphic encryption.

Another technique, randomized perturbation which is proposed by Polat et al. [15], claimed that accurate recommendation could still be obtained while randomness from a specific distribution are added to the original data to prevent information leakage. The same idea is introduced in a recent work [28] which is also achieved by adding random in a certain range to the original data. However, the range α of randomness was chosen by experience and does not have provable privacy guarantees. What's worse, it is recognized that with the application of the clustering on the perturbed data, adversaries can accurately infer users' private data with accuracy up to 70% [23].

Though the privacy protection of randomized perturbation is insecure, it inspires us to design a lightweight and provable perturbation. Specifically, we develop our privacy preserving QoS prediction for users with the integration of a strong and provable privacy model, differential privacy, which is the state-of-art technique for privacy preserving data publishing. Differential privacy [6] has drawn much research attention literally, as it aims at providing effective means to minimize the noise added to the original data with respect to a specific privacy.

Despite the prosperity of differential privacy, applications of QoS prediction is rather limited. To the best of our knowledge, [13, 12] are two differential privacy based privacy preserving recommendation systems which are the most related works to our problem. Machanavajjhala et al. [12] studied the privacy preserving of personalized social recommendation which is solely based on user's social graph. With differential privacy, sensitive links in the social graph can be

preserved effectively which means that attackers cannot deduce the existence of a single link in the graph by passively observing a recommendation result. But, it is also found that good recommendations were achievable only under weak privacy parameters, or only for a small fraction of users. McSheery and Mironov [13] applied differential privacy to collaborative filtering, a general solution for recommendation systems. They split the recommendation algorithms into two parts; they are the *learning phase* which can be performed with differential privacy guarantees, and *individual recommendation phase* in which the learned results are used for individual prediction. Different from the work done by [13, 12], we focus on the privacy guarantees of data publishing instead of knowledge learning and we explore additional approaches beyond those being investigated in [13], like latent factor models.

To sum up, the main contribution of this work is to formulate a differential privacy based privacy preserving collaborative Web Services QoS prediction. The task is non-trivial and our approach has the following advantage:

- For the approach we consider, privacy-preserving algorithms can be parameterized to essentially match the prediction to their non-private analogues.
- By integrating the privacy guarantees into an application, we can provide user with unfettered access to the raw data.
- Experiments on the real world dataset show that prediction accuracy on our disguised data is very close to that on users' private data.

This paper is organized as follows: Section 2 introduces some techniques used to building our privacy-preserving solution. Section 3 presents the system architecture of our privacy-preserving QoS prediction framework and the detail of our approach. Experimental results of proposed framework are presented in Section 4. Finally, Section 5 concludes the paper.

2 Differential Privacy

It's necessary to distinguish between differential privacy and traditional cryptosystems. Differential privacy gives a rigorous and quantitative definition on privacy leakage under a very strict attack model, and has it proved. Based on the idea of differential privacy, user can get privacy protection at utmost with ensuring the availability of data. The biggest advantage of this method is: although based on data distortion, the noise needed for perturbation is independent of data size. We can achieve high level of privacy protection by adding a very small amount of noise [20]. Despite many privacy preserving methods, like k-anonymity and l-diversity, have been proposed, differential privacy is still recognized as the most rigorous and robust privacy preserving model because of its solid mathematical foundation.

2.1 Security Definition Under Differential Privacy

There are two hypothesizes of differential privacy. On one hand, the output of any computation such as SUM, should not be affected by any operation like inserting

or deleting a record. On the other hand, 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 she has the knowledge of the entire dataset except the target one. The formal definition is as follows.

Definition 1 (ϵ-Differential Privacy [5]). 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 $S \subseteq Range(K)$,

$$\frac{Pr[K(D_1 \in S)]}{Pr[K(D_2 \in S)]} \leq exp(\epsilon) \tag{1}$$

D is a database of rows, D_1 is a subset of D_2 and the larger data set D_2 contains exactly one additional row. The probability space $Pr[.]$ in each case is over the coin flips of K. The privacy parameter $\epsilon > 0$ is public, and a smaller ϵ yields a stronger privacy guarantee.

Since differential privacy is defined under probabilistic, any method to achieve this is necessarily random. Some of these, like the Laplace mechanism [6], rely on adding controlled noise. Others, like the exponential mechanism [14] and posterior sampling [4], sample from a problem-dependent distribution instead. We will elaborate the construction in the following part.

2.2 Laplace Mechanism via Global Sensitivity

Apart from the definition of differential privacy, Dwork [6] also claimed that 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:

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

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 $\sqrt{2b}$.

To add noise with Laplace distribution, b is set to $\Delta f/\epsilon$ and the generation of noise is referred as:

$$laplace(\Delta f/\epsilon) \tag{3}$$

here, Δf is global sensitivity, the definition is given next. ϵ is privacy parameter which used to leverage the privacy. As we can see from the equation, the added noise is proportional to Δf , and is inversely proportional to ϵ.

Definition 2 (Global Sensitivity [5]). For f: $D \rightarrow R^d$, the L_k-sensitivity of f is

$$\Delta f = max_{D_1, D_2} ||f(D_1) - f(D_2)||_k \tag{4}$$

for all D_1, D_2 differing in at most one element and $|| \cdot ||_k$ denotes the L_k-norm.

3 Privacy Preserving Collaborative Web Service Prediction

3.1 System Model

As we have discussed in introduction, [23] has testified that randomized perturbation is not safe as it can be inferred by the technique of clustering, but the system model proposed by [28] is mature and suitable for many scenarios, so we adopt and adapt this model here. Specifically, each user disguises her observed QoS values of the services she has invoked and collected locally, and then sends to the server, the owner of all disguised QoS values. It's safe to upload QoS values since the server cannot derive any sensitive information about individual with disguised data. However, the data disguising scheme should still be able to allow the server to conduct collaborative filtering (either neighborhood-based or model-based) from the disguised data. Based on the predicted QoS values, the server can run a variety of applications such as QoS-based selection, composition and recommendation.

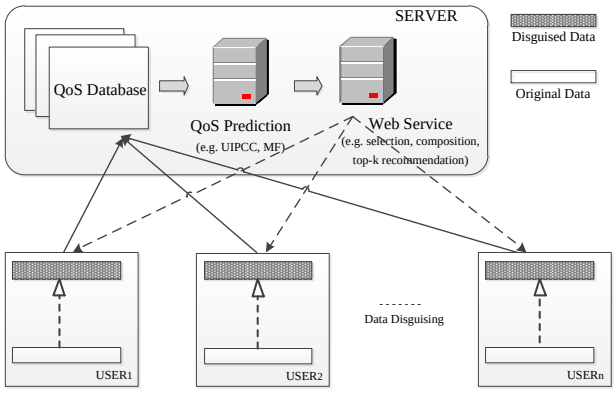


Fig. 1. Privacy Preserving Collaborative QoS Prediction

Data disguising is the key component of privacy preserving collaborative web service QoS prediction. The basic idea of data disguising is to perturb the raw data with randomness in such properties: a) randomness should be able to guarantee no sensitive information (such as each individual user's QoS value) can be deduced from perturbed data; b) though information of individual is limited, the aggregate information of these users can still be evaluated with decent accuracy when the number of users is significantly large. Such property is useful for computations that are based on aggregate information. For those computation, we can still generate meaningful outcome without knowing the

exact values of individual data items because the needed aggregate information can be estimated from the perturbed data.

Another focus of our method is the trade-off between accuracy and privacy. The more the number of randomness, the bigger the gap between disguised data and original data, which presents a higher level of privacy. Oppositely, the less the randomness number, the more obvious the data characteristics. For the computation based on context, this means a more accurate outcome. It has been an open problem to deal with the trade-off between the accuracy and privacy. In this paper, the privacy is parameterized by ϵ and given by each user. By taking advantages of differential privacy, the randomness number added in the observed QoS values is the least which preserves a decent accuracy with respect to a specific privacy.

3.2 Privacy Preserving Collaborative Web Service QoS Prediction

Collaborative filtering (CF) is a mature technique adopted by most modern recommender systems. In this section, we adopt two representative CF approaches: Neighborhood-based Collaborative Filtering and Model-based Collaborative Filtering. We will show how to integrate differential privacy into two representative CF approaches for Web services QoS prediction. More details about these two methods can be found in [24] and [16].

Differential Privacy Based Data Disguising We begin with the data disguising. We use r_{ui} to denote a QoS value collected by user u for web service i, r_u for the entire vector of QoS values evaluated by user u, and similarly, I_{ui} and I_u denote the binary elements and vectors indicating the presence of QoS values respectively. $c_u = |I_u|$ is the number of QoS values evaluated by user u. In our exposition, differential privacy is the key technique used for data disguising. Laplace mechanism [6] obtains ϵ-differential privacy by adding noise of Laplace distribution.

Definition 3 (Laplace Mechanism [5]). Given a function g: $D \rightarrow R^d$, the following computation maintains ϵ-differential privacy:

$$X = g(x) + Laplace(\Delta f/\epsilon) \tag{5}$$

We distinguish between disguised data and original data with upper case and lower case, respectively. ϵ is privacy parameter used to leverage the privacy and smaller ϵ provides a stronger privacy guarantee. Δf is global sensitivity, the definition is given forehead. here, we compute Δf with L_1-norm:

$$\Delta f = max_{D_1,D_2}||g(D_1) - g(D_2)||_1 \tag{6}$$

For simpleness, ϵ-differential privacy of each user u is achieved by the following equation:

$$R_{ui} = r_{ui} + Laplace(\Delta f/\epsilon) \tag{7}$$

where, Δf is defined as the maximum difference between QoS values, which is:

$$\Delta f = max(r_{ui} - r_{uj}) \tag{8}$$

After disguising, all user sends disguised QoS values R_u to server, sensitive information about original data r_{ui} is preserved by randomness. However, the aggregate information of users can still be estimated. Thus, QoS prediction can be performed with direct access to R_{ui} independently.

Collaborative Web Service QoS Prediction Next, we will show how to extend the two representative collaborative filtering approaches to perform our differential privacy based QoS prediction based on disguised data.

1) Neighbourhood-based Collaborative Filtering

Here,we divide all process into three parts: z-score normalization, data disguising and QoS prediction.

Step 1: to eliminate the difference between user data and facilitate better accuracy, the user needs to perform z-score normalization on the observed QoS data. Z-score normalization is performed on the QoS value with the following equation:

$$q_{ui} = (r_{ui} - \overline{r}_u)/\omega_u \tag{9}$$

where $\overline{r}_u$ is the mean and ω_u is the standard deviation of QoS vector r_u. After the normalization, QoS data have a zero mean and unit variance.

Step 2: user perform disguising on the normalized QoS value by:

$$Q_{ui} = q_{ui} + Laplace(\Delta f/\epsilon) \tag{10}$$

where, ϵ, the privacy parameter, is set by user u. Δf is defined according to the distribution of QoS value, which is: $\Delta f = max(r_{ui} - r_{uj})$. After disguising, the user sends their own disguised values Q_u to server, sensitive information about original data q_{ui} is preserved by randomness. Nevertheless, the aggregate information of users can still be estimated. Thus, QoS prediction can be performed with direct access to Q_{ui}.

During the process of QoS prediction, two types of similarity are calculated in order to improve prediction accuracy: *user similarity* and *service similarity*. In particular, the similarity between two users u and v are calculated based on the services they have commonly invoked using the following equations:

$$Sim(u,v) = \frac{\sum_{s_i \in S}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{s_i \in S}(r_{u,i} - \bar{r}_u)^2 \sum_{s_i \in S}(r_{v,i} - \bar{r}_v)^2}}, \tag{11}$$

where $S = S_u \bigcap S_v$ is the set of services that user u and user v have commonly invoked, $r_{u,i}$ is the QoS value of service i observed by user u, $\bar{r}_u$ is the average QoS value of all services observed by user u.

However, due to the disguising of QoS values, at server side we only have the disguised QoS value Q_{ui}, rather than true value q_{ui}. Therefore, we consider to employ Q_{ui} to approximately compute the similarity value as follows.

According to the z-normalization, $\omega_u = \sqrt{\sum_{s_i \in S}(r_{u,i} - \bar{r}_u)^2 / c_u}$, and by substituting this formula into computation, the similarity can be calculated as

$$Sim(u,v) = \frac{\sum_{s_i \in S}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\omega_u \omega_v \sqrt{c_u c_v}}, \tag{12}$$

also, we observe that during the z-normalization, $q_{ui} = (r_{ui} - \overline{r}_u)/\omega_u$. Then, it is easy to get that

$$Sim(u,v) = \frac{\sum_{s_i \in S} q_{u,i} q_{v,i}}{\sqrt{c_u c_v}}, \tag{13}$$

Next, we will prove that though with data disguising, the scalar product property between two vectors remains the same. To make it clear, we denote the two vectors as $a = (a_1, a_2, \cdots, a_n)$ and $b = (b_1, b_2, \cdots, b_n)$ respectively. After disguising, the two vectors are $A = (A_1, A_2, \cdots, A_n)$ and $B = (B_1, B_2, \cdots, B_n)$. We have

$$\begin{aligned} AB &= \sum_{i=1}^{n} A_i B_i \\ &= \sum_{i=1}^{n} (a_i + Laplace(\Delta f_a/\epsilon_a))(b_i + Laplace(\Delta f_b/\epsilon_b)) \\ &= \sum_{i=1}^{n} (a_i b_i + Laplace(\Delta f_a/\epsilon_a) Laplace(\Delta f_b/\epsilon_b)) \\ &+ \sum_{i=1}^{n} (a_i Laplace(\Delta f_b/\epsilon_b) + b_i Laplace(\Delta f_a/\epsilon_a)) \end{aligned}$$

because a_i and $Laplace(\Delta f_b/\epsilon_b)$ are independent vectors and $Laplace(\Delta f_b/\epsilon_b)$ is symmetric exponential distribution with $\mu = 0$, we have $\sum a_i Laplace(\Delta f_b/\epsilon_b) \approx 0$. Likewise, we have

$$\sum b_i Laplace(\Delta f_a/\epsilon_a) \approx 0 and \sum Laplace(\Delta f_a/\epsilon_a) Laplace(\Delta f_b/\epsilon_b) \approx 0. \tag{14}$$

Hence, we derive the following equation:

$$AB \approx \sum a_i b_i = ab. \tag{15}$$

Furthermore, we can get

$$Sim(u,v) \approx \frac{\sum_{s_i \in S} Q_{u,i} Q_{v,i}}{\sqrt{c_u c_v}}. \tag{16}$$

Note that that $Sim(u,v)$ is ranging from [-1, 1], and a larger value indicates that two users (or services) are more similar [24].

Based on the above similarity values, the QoS value of service i observed by user u can be predicted directly. We make use of the similar users to user u through the following equation:

$$q'_{u,i} = \bar{Q}_u + \sum_{v \in User} \frac{Sim(u,v)(q_{v,i} - \bar{q}_v)}{\sum_{v \in User} Sim(u,v)}, \tag{17}$$

Like the user based QoS prediction, the item based QoS prediction can be computed the same, or as proved in [24], these two ways can be combined together to improve the accuracy of QoS prediction. Due to the limit of space, we omit the description of these approaches here.

2) Model-based Collaborative Filtering

Matrix factorization (MF) [25] is a typical solution of model based collaborative filtering which improves the accuracy of prediction effectively by studying latent factor models.

Suppose that the observed QoS values of n users and m services are in a sparse matrix denoted by Q_{n*m} where each element q_{ij} reflects a QoS value of user i for service j. With the input of Q_{n*m}, MF aims to factorize the user-services matrix Q_{n*m} into two latent matrices of a lower dimension d: user-factor matrix U_{n*d} and service-factor matrix V_{m*d}. Then, vacant elements in Q_{n*m} can be approximated as the product of U and V, i.e., unknown QoS value q'_{ij} is evaluated by $q'_{ij} = U_i \cdot V_j^T$.

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

$$min_{U,V} \sum_{q_{ij} \in Q} [(q_{ij} - U_i V_j^T)^2 + \lambda(||U_i||^2 + ||V_j||^2)] \tag{18}$$

The first part is the squared difference between the existing QoS matrix and the predicted one, but only for elements that have evaluated by users. The latter part is the regularization term, added to deal with overfitting induced by the sparsity of input. By dealing with this optimization, we get user-factor matrix U_{n*d} and service-factor matrix V_{m*d} eventually.

Alternative least squares (ALS) and stochastic gradient descent (SGD) are two commonly used methods for solving this optimization problem. We take SGD as our solution since ALS is more difficult which requires the computation of inverse matrix. Iterative equations of SGD are as follows:

$$U_i \leftarrow U_i + \gamma((q_{ij} - U_i V_j^T)V_j - \lambda' U_i) \tag{19}$$

$$V_j \leftarrow V_j + \gamma((q_{ij} - U_i V_j^T)U_i - \lambda' V_j) \tag{20}$$

γ is the learning rate, λ' is the regularization coefficient. The selection of both parameters affects the result significantly. When the value of γ is big, it results in divergence and the results cannot get into convergence. To get a convergence, we set γ to a small value, 0.001 by experience, though it requires a longer training time. And λ' is set to 0.01 which is also selected by experience.

In the first iteration, U and V are set randomly. But a better selection can help to accelerate the computation effectively. Hence, we initialise U and V near the average of all QoS value that have been observed. The iteration will terminate when the objective function value is less than a certain threshold.

4 Experiments

In this section, we conduct three series of experiments on a real data set to evaluate our privacy-preserving QoS prediction framework.

4.1 Experimental Setup

We first note that a real Web services QoS dataset is introduced in [27, 26], 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 throughout (TP). Table 1 describes the statistics of the dataset, 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 [27, 26]. During the presentation of our experiment, we set the performance of RT in the left and TP in the right.

We use cross validation to train and evaluate the QoS prediction. The dataset here is collected motivated and complete, but in practice, for limited time and resource, a user usually invokes only a handful of services, and the density of data is under 10% generally. To simulate such sparsity in our experiment, we randomly remove entries from the full dataset and only keep a small density of historical QoS values as our training set. And the removed data is treated as testing set for accuracy evaluation.

Then, we perform algorithms of QoS prediction on training set and evaluate the prediction on testing set. Four algorithms are implemented and evaluated here. UIPCC which is proposed in [24] is a representative implementation of neighbourhood-based collaborative filtering and MF introduced in [25] is an implementation of model-based collaborative filtering. LUIPCC and LMF are two methods intergrading with differential privacy which is achieved by Laplace mechanism.

To quantize the accuracy of QoS prediction, we employ Root Mean Square Error (RMSE) as the metric which has been widely used in related work (e.g., [2, 13]):

$$RMSE = \frac{\sqrt{\sum_R (q_{ui} - q'_{ui})^2}}{|R|} \tag{21}$$

R consists of all value needed to be predicted in training set and $|R|$ is the number of set R. q'_{ui} is the predicted value of set R and q_{ui} is the corresponding value in testing set. Generally, a smaller RMSE indicates a better prediction.

Table 1. Statistic of Datasets

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%

Noted that, the default setting of parameters follows Table 2. We choose parameters of UIPCC and MF by experiences of [24] and [25]. Generally, ϵ is set to 0.5 by default which can preserve sufficient privacy.

Table 2. Parameter Setting

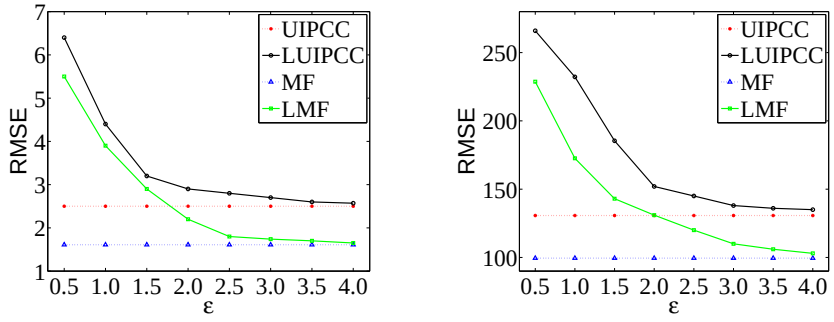
UIPCC	k=20	λ =0.1	-
MF	d=20	γ =0.001	λ' =0.01
Laplace	$\epsilon = 0.5$	-	-

4.2 Privacy vs Accuracy

Fig 2 is the comparison corresponding to RT and TP between our differential privacy based QoS prediction and original approaches under varying privacy. By introducing differential privacy into QoS prediction, users can achieve privacy. But for users who adopt our approaches, they do need to consider the trade-off between privacy and accuracy. On one hand, a user can attain high level privacy by adding more Laplace noises which definitely decreases the utility of data. On the other extremal hand, a user can get a 100% accuracy without adding any Laplace noises. To study the performance of changing accuracy, we perform algorithms of QoS prediction on testing set and evaluate the prediction on testing set. The privacy parameter, ϵ, changes from 0.5 to 4 stepped by 0.5. We can observe that both LUIPCC and LMF decrease in RMSE when ϵ gets larger. A larger ϵ means a looser privacy constraints and the utility of data is less limited, thus user can get a better accuracy. It is also noticeable that when ϵ gets larger, e.g., larger than 2.0 in Fig 2, our privacy-preserving approaches, both LUIPCC and LMF, can acquire almost the same or even more accuracy than UIPCC. Especially, when ϵ is as large as 4, the prediction accuracy of LMF is much better than the baseline UIPCC. Additionally, we find that MF outperforms UIPCC. This suggests the superior effectiveness of model-based approaches in capturing the latent structure of the QoS data, which conforms to the results reported in [25].

Another fact which requires our attention is that though a recent work [28] claims a better performance than both original original algorithms, UIPCC and MF, the randomness added to prevent the information leakage is not large enough, adversaries can accurately infer users' privacy data with the application of the clustering [23].

To sum up, our differential privacy based algorithms can provide a privacy preserving QoS prediction with a parameterized privacy. And the results show that recommendation on our disguised user data is very close to that on user's private data under a loose constraint.



(a) Response Time (b) Throughput

Fig. 2. Privacy vs Accuracy

4.3 Influence of data size

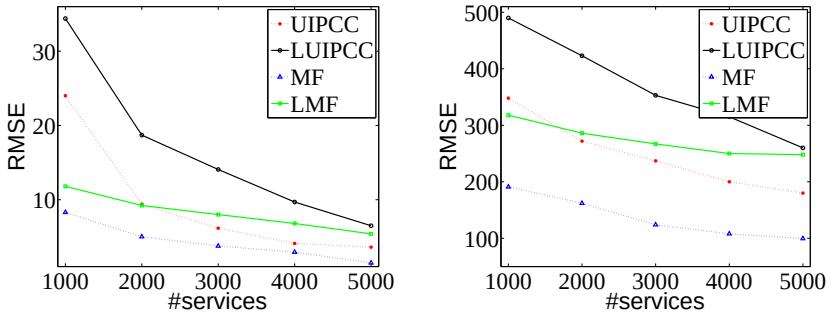
To evaluate the influence of data size, we design the experiments by changing the number of services and users respectively. In Fig 3, the number of users is set to 339 and the number of service is varying from 1000 to 5000 with a step 1000, where the service is selected randomly from the original dataset. And the other parameters of the experiment are set as table 2. We do the same experiment setting in Fig 4 which contains 5825 services.

It is straightforward that both the number of services and the number of users have a positive influence on the accuracy of algorithms which means that the more the data is given, the better the prediction can be. In other words, with more data, we can provide a better accuracy.

Another finding is that though the accuracy differs significantly between different data size, the trend of original algorithms and our differential privacy based algorithms are the same, such as the trend of UIPCC and LUIPCC or the trend of MF and LMF. It infers an dramatically advantage of differential privacy that the noise needed for data disguising is independent of data size, so users can achieve a high level of privacy protection by adding a very small amount of noise.

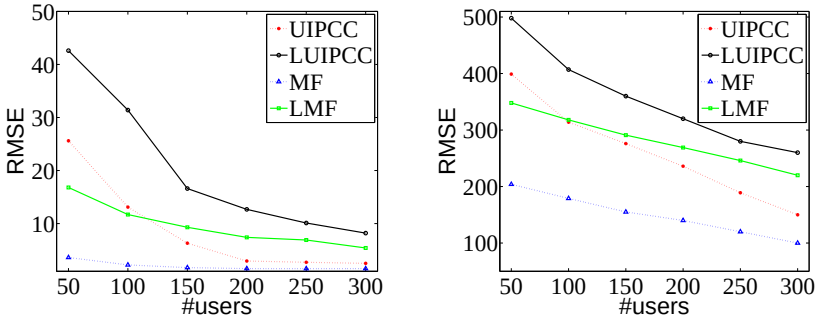
4.4 Influence of density

In addition to data size, density denoted as θ is also a subject to the performance of algorithms. Fig 5 presents the results of the accuracy comparison under different density. Though the influence of density on original algorithms is not obvious, it does have an significant influence of our differential based algorithms. The dataset with a higher density performs better. This result implies that the density is also a crucial factor for determining the performance of our differential privacy based approaches. More importantly, we can observe that when the number of services gets larger, the gap between traditional approaches and our differential privacy based approaches gets smaller. More precisely, in Fig 5,



(a) Response Time (b) Throughput

Fig. 3. Influence of Services



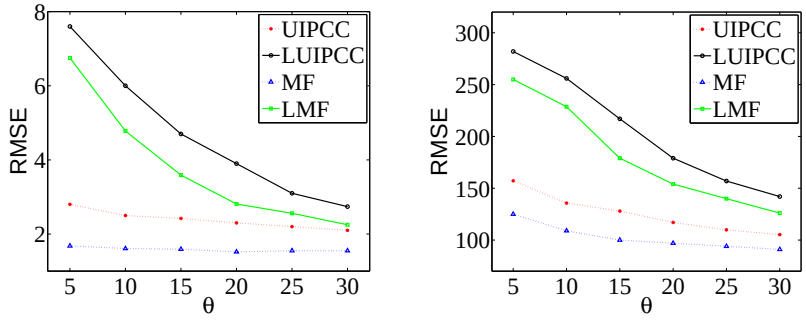
(a) Response Time (b) Throughput

Fig. 4. Influence of Users

when the density is set to 5, the gap between LUIPCC and UIPCC is 5. However, when the density increased to 30, the gap between LUIPCC and UIPCC decreases to 1. So, users are suggested to use the dataset with a higher density to preserves a closer prediction to original results.

5 Conclusion

To the best of our knowledge, this is the first piece of work that introduces differential privacy into a collaborative web services QoS prediction framework. Differential privacy gives a rigorous and quantitative definition on privacy leakage under very strict constraints. Based on the idea of differential privacy, users can get privacy protection at utmost with ensuring the availability of data. Empirical results show that our framework provides a secure and accurate collaborative Web services QoS prediction.



(a) Response Time (b) Throughput

Fig. 5. Influence of Density

Acknowledgment

Research reported in this publication was partially supported Natural Science Foundation of China (Grant Nos. 61572336, 61572335, 61402313) and Natural Science Foundation of Jiangsu Province of China under Grant No. BK20151223.

References

[1] An, L., Liu, H., Li, Q., Huang, L., Xiao, M.: Constraints-aware scheduling for transactional services composition. Journal of Computer Science and Technology 24(4), 638–651 (2009)
[2] Berlioz, A., Friedman, A., Kaafar, M.A., Boreli, R., Berkovsky, S.: Applying differential privacy to matrix factorization. In: The ACM Conference. pp. 107–114 (2015)
[3] Canny, J.: Collaborative filtering with privacy via factor analysis. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 238–245 (2002)
[4] Dimitrakakis, C., Nelson, B., Mitrokotsa, A., Rubinstein, B.I.P.: Robust and private bayesian inference. Arxiv 8776, 291–305 (2014)
[5] Dwork, C.: Differential Privacy (2006)
[6] Dwork, C., Mcsherry, F., Nissim, K.: Calibrating Noise to Sensitivity in Private Data Analysis. VLDB Endowment (2014)
[7] Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.: Generating private recommendations efficiently using homomorphic encryption and data packing. IEEE Transactions on Information Forensics Security 7(3), 1053–1066 (2012)
[8] Gentry, C.: A fully homomorphic encryption scheme (2009)
[9] Liu, A., Li, Q., Huang, L., Xiao, M.: FACTS: A framework for fault-tolerant composition of transactional web services. IEEE Trans. Services Computing 3(1), 46–59 (2010)
[10] Liu, A., Li, Q., Huang, L., Ying, S., Xiao, M.: Coalitional game for community-based autonomous web services cooperation. IEEE Transactions on Services Computing 6(3), 387–399 (2013)

[11] Liu, A., Zhengy, K., Liz, L., Liu, G., Zhao, L., Zhou, X.: Efficient secure similarity computation on encrypted trajectory data. In: IEEE International Conference on Data Engineering. pp. 66–77 (2015)
[12] Machanavajjhala, A., Korolova, A., Sarma, A.D.: Personalized social recommendations: accurate or private. VLDB Endowment (2011)
[13] Mcsherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 627–636 (2009)
[14] Mcsherry, F., Talwar, K.: Mechanism design via differential privacy. In: Foundations of Computer Science, 2007. FOCS '07. IEEE Symposium on. pp. 94–103 (2007)
[15] Polat, H., Du, W.: Privacy-preserving collaborative filtering using randomized perturbation techniques. In: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA. pp. 625–628 (2003)
[16] Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: International Conference on Neural Information Processing Systems. pp. 1257–1264 (2007)
[17] Shang, S., Chen, L., Wei, Z., Jensen, C., Wen, J.R., Kalnis, P.: Collective travel planning in spatial networks. IEEE Transactions on Knowledge Data Engineering 28(5), 1132–1146 (2016)
[18] Shang, S., Ding, R., Zheng, K., Jensen, C.S., Kalnis, P., Zhou, X.: Personalized trajectory matching in spatial networks. The VLDB Journal 23(3), 449–468 (2014)
[19] Tang, M., Jiang, Y., Liu, J., Liu, X.: Location-aware collaborative filtering for qos-based service recommendation. In: IEEE International Conference on Web Services. pp. 202–209 (2012)
[20] Yanga, L.I., Wen, W., Xie, G.Q.: Survey of research on differential privacy. Application Research of Computers 29(9), 3201–582 (2012)
[21] Yu, Q., Zheng, Z., Wang, H.: Trace norm regularized matrix factorization for service recommendation. In: IEEE International Conference on Web Services. pp. 34–41 (2013)
[22] Zhang, Q., Ding, C., Chi, C.H.: Collaborative filtering based service ranking using invocation histories. In: IEEE International Conference on Web Services. pp. 195–202 (2011)
[23] Zhang, S., Ford, J., Makedon, F.: Deriving private information from randomly perturbed ratings. In: Siam International Conference on Data Mining, April 20-22, 2006, Bethesda, Md, Usa (2006)
[24] Zheng, Z., Ma, H., Lyu, M.R., King, I.: Wsrec: A collaborative filtering based web service recommender system. In: IEEE International Conference on Web Services. pp. 437–444 (2009)
[25] Zheng, Z., Ma, H., Lyu, M.R., King, I.: Qos-aware web service recommendation by collaborative filtering. IEEE Transactions on Services Computing 4(2), 140–152 (2010)
[26] Zheng, Z., Zhang, Y., Lyu, M.R.: Distributed qos evaluation for real-world web services. In: IEEE International Conference on Web Services. pp. 83–90 (2010)
[27] Zheng, Z., Zhang, Y., Lyu, M.R.: Investigating qos of real-world web services. IEEE Transactions on Services Computing 7(1), 32–39 (2014)
[28] Zhu, J., He, P., Zheng, Z., Lyu, M.R.: A privacy-preserving qos prediction framework for web service recommendation. In: IEEE International Conference on Web Services. pp. 241–248 (2015)