Towards Privacy-Preserving Data Mining in Online Social Networks: Distance-Grained and Item-Grained Differential Privacy

Shen Yan[1,2,3], Shiran Pan[1,2,3], Yuhang Zhao[1,2], and Wen-Tao Zhu[1,2]⋆

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{yanshen, panshiran, zhaoyuhang}@iie.ac.cn
wtzhu@ieee.org
[2] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. Online social networks have become increasingly popular, where users are more and more lured to reveal their private information. This brings about convenient personalized services but also incurs privacy concerns. To balance utility and privacy, many privacy-preserving mechanisms such as differential privacy have been proposed. However, most existent solutions set a single privacy protection level across the network, which does not well meet users' personalized requirements. In this paper, we propose a fine-grained differential privacy mechanism for data mining in online social networks. Compared with traditional methods, our scheme provides query responses with respect to different privacy protection levels depending on where the query is from (i.e., is distance-grained), and also supports different protection levels for different data items (i.e., is item-grained). In addition, we take into consideration the collusion attack on differential privacy, and give a countermeasure in privacy-preserving data mining. We evaluate our scheme analytically, and conduct experiments on synthetic and real-world data to demonstrate its utility and privacy protection.

Keywords: Differential privacy, Online social networks, Privacy-preserving data mining, Collusion attack

1 Introduction

Online social networks (OSNs) are increasingly involved in our daily life. Driven by various OSN applications, large number of profiles are uploaded. Meanwhile, the tendency towards uploading personal data to OSNs has raised privacy concerns. Debates on privacy leakage in OSNs [1–3] have continued for years. Although users have certain degree of privacy concerns, actually such concerns

⋆ Corresponding author

would not be converted into any actions [1]. The contradiction between privacy and utility calls for better privacy mechanisms in OSNs.

Two main aspects of OSNs' privacy researches are privacy-preserving data mining (PPDM) [4] and privacy-preserving data publishing (PPDP) [5], which respectively achieves data mining and data sharing goals without privacy leakage. k-anonymity [6] is one of the most widely used anonymity models, which divides all records into several equivalence groups and hides individual records in groups. Later models like l-diversity [7], t-closeness [8], and (a, k)-anonymity [9] are proposed to enhance the privacy guarantee in various environments. Nevertheless, it has been shown that former anonymity models are susceptible to several background knowledge attacks [10]. The notion of differential privacy [11] is proposed in the context of statistical databases, which provides a mechanism resistent to background knowledge attacks. Differential privacy provides a quantifiable measurement of privacy via a certain parameter ϵ. A query is said to satisfy ϵ-differential privacy, if when an individual item in a data set is added or removed, the probability distribution of the query answers varies by a factor of $\exp(\epsilon)$. Thus curious users cannot deduce the individual item from the query answer.

Most previous work on differential privacy assigns the same privacy protection levels (i.e., ϵ) to all users in the network. However, in practice, heterogeneous privacy requirements are needed [12–15]. Users tend to allow people who are more closed to them (e.g., families, close friends) to get more accurate information, whereas those who are just acquaintances or strangers obtain obscure data. In addition, items in each user's data set require various privacy protection levels. For example, individual users may regard their telephone numbers as very private information that should not be leaked to strangers, whereas some business accounts need to publish their telephone numbers so as to get in touch with potential consumers.

Since existing privacy techniques cannot satisfy the increasing and heterogenous privacy demands, finer-grained privacy-preserving mechanisms are required in OSNs. In this paper, we propose a fine-grained differential privacy mechanism, and the merits of our scheme can be summarized as follows:

- First, each query satisfies ϵ-differential privacy, where ϵ varies with the distance between users (i.e., distance-grained).
- Second, each user can assign different privacy protection levels for different items in their respective data set (i.e., item-grained).
- Third, our scheme is resistent to the collusion attack in OSNs.

Our work adopts some techniques in [16, 17], and combines the merits of them. In addition, our work extends the application scenario in [16] to PPDM, and proposes a new method to resist the collusion attack.

This paper is organized as follows. We present the preliminaries in Section 2, and provide the problem formulation in Section 3. Section 4 gives an implementation approach to our scheme. The theoretical analysis is proposed in Section 5. Section 6 presents experimental results, followed by conclusions in Section 7.

2 Preliminaries

2.1 Differential Privacy

Differential privacy is originally introduced by Dwork [11]. The privacy guarantee provided by differential privacy is implemented via adding noise to the output of the computation, thus curious users cannot infer whether or not the individual data is involved in the computation.

Definition 1 (ϵ-Differential Privacy) *A random function M satisfies ϵ- differential privacy if for all neighboring data sets D and D', and for all outputs $t \in \mathbb{R}$ of this randomized function, the following statement holds:*

$$\Pr[M(D) = t] \leq \exp(\epsilon) \Pr[M(D') = t],$$

in which exp *refers to the exponential function. Two data sets D and D' are said to be neighbors if they are different in at most one item. ϵ is the privacy protection parameter which controls the amount of distinction induced by two neighboring data sets. A smaller value of ϵ ensures a stronger privacy guarantee.*

We can achieve ϵ-differential privacy by adding random noise whose magnitude is decided by the possible change in the computation output over any two neighboring data sets. We denote this quantity as the global sensitivity.

Definition 2 (Global Sensitivity) *The global sensitivity $S(f)$ of a function f is the maximum absolute difference obtained on the output over all neighboring data sets:*

$$S(f) = \max_{D \sim D'} |f(D) - f(D')|.$$

To satisfy the differential privacy definition, two main mechanisms are usually utilized: the *Laplace* mechanism and the *Exponential* mechanism. The *Laplace* mechanism is proposed in [18], which achieves ϵ-differential privacy by adding noise following *Laplace* distribution. Between the two mechanisms, the *Laplace* mechanism is more suitable for numeric outputs.

Definition 3 (Laplace Mechanism) *Given a function f: $D \to \mathbb{R}$, the mechanism M adds Laplace distributed noise to the output of f:*

$$M(D) = f(D) + V, \ \textit{where } V \sim Lap(\frac{S(f)}{\epsilon}),$$

where $Lap(\frac{S(f)}{\epsilon})$ has probability density function $h_\epsilon(x) = \frac{\epsilon}{2S(f)} \exp(\frac{-\epsilon|x|}{S(f)})$.

2.2 Related Work

The notion of differential privacy was introduced by Dwork [11]. After that, differential privacy has been a popular privacy-preserving mechanism in PPDM and PPDP. However, most previous work focuses on theoretical aspects. In this paper, we propose a fine-grained differential privacy mechanism in OSNs scenario.

The majority of previous work on fine-grained differential privacy provides user-grained privacy protection levels [14], where each user can set a personalized protection parameter (i.e., ϵ). Yuan et al. [19] considered personalized protection for graph data in terms of both semantic and structural information. However, considering the heterogeneous privacy need of different items in users' data sets, finer-grained mechanisms are required. Alaggan et al. introduced a mechanism [17] named *heterogeneous differential privacy* (HDP), HDP allows users set different privacy protection levels for different items in each data set, and Jorgensen et al. [15] extends HDP to *Exponential* mechanism of differential privacy. In our scheme, we adopt some techniques in [17] to realise the item-grained differential privacy.

Different from user-grained differential privacy, distance-grained differential privacy mechanism decides parameter ϵ by where the query is from. To the best of our knowledge, Koufogiannis et al. proposed the first distance-grained mechanism in [20], and later introduced a modified distance-grained differential privacy mechanism in PPDP [16].

Furthermore, few works take into consideration the collusion attack on differential privacy. Although the collusion attack does not leak private information directly, it expands the parameter ϵ, thus loosing the privacy guarantee of the system. Zhang et al. [21] designed a collusion resistent algorithm in PPDM which combines multiparty computation and differential privacy. Koufogiannis et al. [16, 20] proposed mechanisms to address this problem in PPDP.

Inspired by previous work, we design a fine-grained differential privacy mechanism in OSNs. Although some of the tools in [16, 17] are leveraged to provide a solution, we consider a different problem which applies to PPDM rather than PPDP. In PPDM scenario, the users related to a query can be two random users in the network, which makes resisting the collusion attack more difficult.

3 Problem Formulation

3.1 System Model

Consider an OSN represented as a graph $G = (N, E)$, where N is the set of nodes and E is the set of edges connecting pair of nodes in N. Each node in N represents a user in the network and each edge represents the friendship relationship between users. Each user owns a data set D, which is denoted by an n dimensional vector $D = [b_1, b_2, ..., b_n]$.[4]

There exist three parties in our model: server, data owner, and inquirer.

- Server usually is a service provider, such as Twitter, Facebook. The server stores the topology of G and users' data sets. In our scheme, we assume that the server is a trusted third party, which means that it will not leak any information to others, and will always return actual distance between users.

[4] Without of causing confusing, we interchangeably use node and user in this paper.

- Data owner is a user in the network, owning a data set D that to be queried.
- Inquirer is a user in the network, who submits a query Q to the server.

Moreover, each user in the network can be both data owner and inquirer.

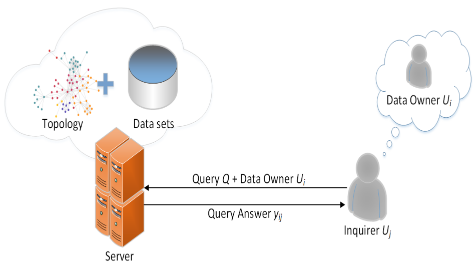


Fig. 1. System Illustration

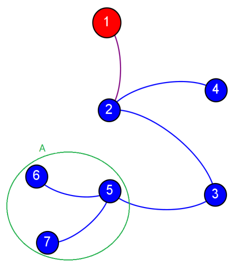


Fig. 2. A network example with 7 nodes, nodes 5-7 constitute group A.

In practice, a data mining task can be a numerical computation of D and Q, or a searching task on data sets. We define $f(D)$ as an arbitrary function of D, where $f(D)$ is defined by specific data mining tasks, and $M(D)$ is a modified $f(D)$ that satisfies differential privacy by adding noise V. As Fig. 1 shows, U_i is the data owner, U_j is the inquirer. When the server receives U_j's query Q, it computes $M(D)$, where D is the data set of U_i. After the computation, the server returns the query answer y_{ij} to U_j in proper methods (e.g., returns the IDs of the users whose computation results satisfy Q, or returns the computation results directly), where $y_{ij} = M(D) = f(D) + V$.

3.2 Fine-Grained Differential Privacy

We propose a fine-grained differential privacy mechanism which assigns different protection levels for different inquirers (i.e., distance-grained differential privacy) and simultaneously satisfies different privacy levels for each dimension of D (i.e., item-grained differential privacy).

A. Distance-grained Differential Privacy.

Practically, under a query, users are willing to give more accurate data to closed people, whereas return obscure result to people with distant relationship. In our scheme, we use shortest path length to measure the relationship between users.

More specifically, inquirer U_j gets a result that ensures $\epsilon(d_{ij})$-differential privacy, where d_{ij} denotes the shortest path length between the data owner U_i and inquirer U_j. If d_{ij} is smaller, $\epsilon(d_{ij})$ should be larger, i.e., $\epsilon(d_{ij})$ is inversely proportional to d_{ij}.

B. Item-grained Differential Privacy.

Usually, items in D have heterogeneous privacy expectations. For instance, D contains hobbies, locations, jobs, and other personal information of a specific user. The user may consider locations are sensitive information that should not be known by other users, whereas be willing to share his hobbies with others. Thus the user can set higher privacy levels (i.e., smaller ϵ) for the items of locations, In our scheme, every user can set a privacy weight vector $W = [w_1, w_2, ..., w_n]$, where each element w_k represents the privacy weight associated to the kth item b_k in D with $w_k \in [0, 1]$; smaller w_k ensures stronger protection, where 0 corresponds to absolute privacy while 1 refers to standard $\epsilon(d_{ij})$-differential privacy.

Consequently, whenever an inquirer U_j submits a query for data owner U_i to the server, he will get a perturbed answer satisfies $w_k\epsilon(d_{ij})$-differential privacy for each dimension.

3.3 Collusion Attack

In practice, users in the network can share information with each other. As Fig. 2 shows, assuming that there is a group A, each user in A has a query answer y_{ij} about data owner U_i's data set D. When users in group A share their query answers, they may derive a more accurate estimate of D, even though each user owns a highly noisy answer of the query.

Theorem 1 *For a group of users A, where each user has a query answer y_{ij} that satisfies $\epsilon(d_{ij})$-differential privacy, they can derive an estimator y_A that ensures $(\sum_{U_j \in A} \epsilon(d_{ij}))$-differential privacy under collusion.*

Proof: To simplify the proving process, we assume that group A has two users: U_{j_1} and U_{j_2}. We set $y_{ij_1} = M_1(D) = f(D) + V_1$, $y_{ij_2} = M_2(D) = f(D) + V_2$.

$$\begin{aligned}
&\frac{\Pr[M_1(D) = y_{ij_1}, M_2(D) = y_{ij_2}]}{\Pr[M_1(D') = y_{ij_1}, M_2(D') = y_{ij_2}]} \\
=&\frac{\Pr[M_1(D) = y_{ij_1}]\Pr[M_2(D) = y_{ij_2}]}{\Pr[M_1(D') = y_{ij_1}]\Pr[M_2(D') = y_{ij_2}]} \\
=&\frac{h_{\epsilon(d_{ij_1})}(y_{ij_1} - f(D))}{h_{\epsilon(d_{ij_1})}(y_{ij_1} - f(D'))}\frac{h_{\epsilon(d_{ij_2})}(y_{ij_2} - f(D))}{h_{\epsilon(d_{ij_2})}(y_{ij_2} - f(D'))} \\
=&\frac{\exp(\frac{-\epsilon(d_{ij_1})|y_{ij_1} - f(D)|}{S(f)})}{\exp(\frac{-\epsilon(d_{ij_1})|y_{ij_1} - f(D')|}{S(f)})}\frac{\exp(\frac{-\epsilon(d_{ij_2})|y_{ij_2} - f(D)|}{S(f)})}{\exp(\frac{-\epsilon(d_{i}j_2)|y_{ij_2} - f(D')|}{S(f)})} \\
\leq&\exp(\frac{\epsilon(d_{ij_1})|f(D') - f(D)|}{S(f)})\exp(\frac{\epsilon(d_{ij_2})|f(D') - f(D)|}{S(f)}) \\
\leq&\exp(\epsilon(d_{ij_1}))\exp(\epsilon(d_{ij_2})) = \exp(\epsilon(d_{ij_1}) + \epsilon(d_{ij_2})).
\end{aligned}$$

Therefore, when the collusion attack happened, the privacy protection parameter will be the sum of colluded users' $\epsilon(d_{ij})$.

Although collusion attacks will not leak data owners' information directly, it expands the parameter ϵ, thus loosing the privacy guarantee of the system.

According to the proof of Theorem 1, we can infer that the collusion attack works when the noise additions are independent of each other. Therefore, we should introduce correlations between the noise additions added to different query answers. In order to meet practical requirements, any group of users cannot derive a better estimator than the most accurate answer among the group, i.e., the answer of the inquirer who is closest to the data owner in the group.

Theorem 2 *For a group of users A, where each user has a query answer y_{ij} that satisfies $\epsilon(d_{ij})$-differential privacy, and the noise additions of y_{ij} are correlated to each other, the best estimator y_A ensures $(\max_{U_j \in A} \epsilon(d_{ij}))$-differential privacy.*

Proof: To simplify the proving process, we assume that group A has two users: U_{j_1} and U_{j_2}. $y_{ij_1} = M_1(D) = f(D) + V_1$, $y_{ij_2} = M_2(D) = f(D) + V_2$, where $\phi(V_2 - V_1)$ is the conditional probability density function of $\Pr[V_1|V_2]$.

$$\begin{aligned}
&\frac{\Pr[M_1(D) = y_{ij_1}, M_2(D) = y_{ij_2}]}{\Pr[M_1(D') = y_{ij_1}, M_2(D') = y_{ij_2}]} \\
=&\frac{\Pr[M_2(D) = y_{ij_2}]\Pr[M_1(D) = y_{ij_1}|M_2(D) = y_{ij_2}]}{\Pr[M_2(D') = y_{ij_2}]\Pr[M_1(D') = y_{ij_1}|M_2(D') = y_{ij_2}]} \\
=&\frac{h_{\epsilon(d_{ij_2})}(y_{ij_2} - f(D))}{h_{\epsilon(d_{ij_2})}(y_{ij_2} - f(D'))}\frac{\phi(y_{ij_1} - f(D) - (y_{ij_2} - f(D)))}{\phi(y_{ij_1} - f(D') - (y_{ij_2} - f(D')))} \\
=&\frac{\exp(\frac{-\epsilon(d_{ij_2})|y_{ij_2} - f(D)|}{S(f)})}{\exp(\frac{-\epsilon(d_{ij_2})|y_{ij_2} - f(D')|}{S(f)})}\frac{\phi(y_{ij_1} - y_{ij_2})}{\phi(y_{ij_1} - y_{ij_2})} \\
\leq&\exp(\epsilon(d_{ij_2})).
\end{aligned}$$

4 Scheme Implementation

In this paper, we take user matching as the example of data mining in OSNs, and give a particular implementation approach to our scheme.

User matching is a common application in OSNs, where the server computes the similarity between users' data sets. Therefore, D is data owner's profile; Q is another n dimension vector $[q_1, q_2, ..., q_n]$, which refers to the user attributes that the inquirer searching for, it can be inconsistent with the inquirer's profile; $f(D)$ is computing the similarity between Q and D.

There are many methods to evaluate the similarity between data (e.g., Euclidean distance, Cosine similarity). We choose Euclidean distance $f_Q(D) = \sqrt{\sum_{i=0}^{n}(b_i - q_i)^2}$ as the evaluation criterion (i.e, $f(D)$), where for binary vectors, $S(f) = \max\limits_{D \sim D'} |f_Q(D) - f_Q(D')| = 1$.

We define the similarity between user U_i and U_j as $\frac{1}{f_Q(D)}$. According to former analysis, the value of ϵ of each user should be inversely proportional to d_{ij}. We set $\epsilon(d_{ij}) = \frac{1}{d_{ij}+1}$.

The working flow of our scheme consists of three phases: creating correlations, query process, and updating perturbations.

4.1 Creating Correlations Phase

Creating correlations is the most crucial and innovative phase in our scheme. The correlation should ensure that each noise addition follows *Laplace* distribution, which makes the perturbations satisfy differential privacy.

Theorem 3 *Consider two noise additions $V_1 \sim Lap(\frac{1}{\epsilon_1})$, $V_2 \sim Lap(\frac{1}{\epsilon_2})$, where $\epsilon_1 < \epsilon_2$. V_1 and V_2 are correlated with a conditional probability Φ, where Φ has the density:*

$$\phi(x) = (1 - \frac{\epsilon_1^2}{\epsilon_2^2})\frac{\epsilon_1}{\sqrt{2\pi}}\sqrt{\epsilon_1 x}K_{-\frac{1}{2}}(\epsilon_1 x), \tag{1}$$

where K is the modified Bessel function of the second kind.

Proof: See the Appendix.

We denote (1) as *bessel* distribution, and its input parameters are ϵ_1 and ϵ_2.

Although Koufogiannis et al. [16] introduce similar method to address this problem by building correlations, we design a novel correlation creation mechanism, which is more suitable for PPDM.

We can see edges of the network as the bridges to build up the correlations between nodes. In our scheme, each edge is assigned a perturbation vector, in which each element represents the sampling noise for different distances. In addition, each node is assigned an initial perturbation, which will not change once the node joined in the network.

Algorithm 1 illustrates the method to assign perturbations over the network. In the algorithm, the largest dimension of each edge's perturbation vector is

Algorithm 1 Perturbation Assignment

```
for $node$ in $G.nodes()$ do
  $R_{node} \sim Laplace(1)$ // Assign initial perturbations
end for
$l = G.diameter()$
$\epsilon[l] = [1, \frac{1}{2}, ..., \frac{1}{l+1}]$
for $(j_1, j_2)$ in $G.edges()$ do
  $i = 0$
  while $i < l$ do
    $perturbation_{j_1,j_2}[i] = bessel(\epsilon[i], \epsilon[i+1])$
    $i = i + 1$
  end while
end for
```

decided by the diameter of the network. Real OSNs' diameters are usually small numbers: according to the networks' data in SNAP[5] database, the Facebook network with 4039 nodes has a diameter of 8, the Gowalla network with 196591 nodes only has a diameter of 14. This property of OSNs guarantees that we only need to save a vector of small size for each edge. In our scheme, *perturbation* is a sample of the *bessel* distribution density, which denotes the amount of noise added to the query answer when passing by the edge.

4.2 Query Process Phase

In query process phase, the inquirer submits a query to the server, and tells the server the data owner of this query. The server uses the scalar product of

Algorithm 2 Noise Computation

```
Input: Data Owner $U_i$, Inquirer $U_j$
  $path[\,] = shortest\_path(i, j)$ //$path$ consists of nodes in the shortest path sequentially
  $N = 0$
  for ($t$=0; $t < len(path) - 1$; $t$++) do
    $dis = shortest\_path\_length(i, path[t+1])$
    $N = N + perturbation_{path[t],path[t+1]}[dis]$
  end for
  $noise = N + R_i$
```

W and D to compute $f_Q(W \cdot D)$, and adds noise to the computation result, where the noise is the sum of the perturbations on each edge in the shortest path from the inquirer to the data owner (if there exists more than one shortest path between two nodes, the algorithm randomly picks one of the paths). We illustrate the noise computation process in Algorithm 2. The query answer is denoted as $y_{ij} = M(D) = f_Q(W \cdot D) + V$.

4.3 Updating Perturbations Phase

In addition to the collusion attack, repeat queries may also cause privacy leakage. If a certain inquirer receives two query answers with the same noise, he can deduce the noise and know the real data sequentially. So perturbation vectors should be updated after each query.

In Algorithm 3, we randomly update half of the perturbations used in the former query, which simultaneously ensures the correlations between noise additions and saves the computation cost.

5 Theoretical Analysis

Our scheme provides privacy protection for both the data sets and their corresponding privacy weight vectors. The privacy protections guaranteed by our scheme are:

[5] `https://snap.stanford.edu/data/index.html`

Algorithm 3 Perturbation Update

```
Input: $path$ of the former query
  $m = \lceil len(path)/2 \rceil$ //Compute the number of updated edges.
  $K = [\ ]$
  for ($t$=0; $t < m$; $t$++) do
    $K[t] = random(0, len(path) - 1)$
  end for
  for ($t$=0; $t < len(path)$; $t$++) do
    $j_1 = path[t]$, $j_2 = path[t+1]$
    if $t$ in $K$ then
      $perturbation_{j_1,j_2}[t+1] = bessel(\epsilon[t], \epsilon[t+1])$
    end if
  end for
```

- Each query answer provides $\epsilon(d_{ij})$-differential privacy for data owner's data set D. More specifically, each dimension satisfies $w_k\epsilon(d_{ij})$-differential privacy.
- Each query answer provides $\epsilon(d_{ij})$-differential privacy for data owner's privacy weight vector W.
- Noise additions for different inquirers are correlated.
- Noise additions before and after update are correlated .

5.1 Privacy Analysis

Theorem 4 *Our scheme ensures that each query answer satisfies $\epsilon(d_{ij})$-differential privacy.*

Proof: See data owner's data set as a whole, the query answer satisfies:

$$\begin{aligned}\frac{\Pr[M(D)=t]}{\Pr[M(D')=t]} &= \frac{h_{\epsilon(d_{ij})}(t-f(D))}{h_{\epsilon(d_{ij})}(t-f(D'))} = \frac{\frac{\epsilon(d_{ij})}{2S(f)}\exp(\frac{-\epsilon(d_{ij})|t-f(D)|}{S(f)})}{\frac{\epsilon(d_{ij})}{2S(f)}\exp(\frac{-\epsilon(d_{ij})|t-f(D')|}{S(f)})} \\ &= \exp(\frac{\epsilon(d_{ij})(|t-f(D)|-|t-f(D')|)}{S(f)}) \\ &\le \exp(\frac{\epsilon(d_{ij})|f(D')-f(D)|}{S(f)}) \le \exp(\epsilon(d_{ij})),\end{aligned}$$

where $h_\epsilon(\cdot)$ is defined in Theorem 1, thus proving the result.

Theorem 5 *The random function M provide $w_k\epsilon(d_{ij})$-differential privacy for each dimension of D.*

Proof: First, we use $S_k(f)$ to denote the local sensitivity of function f:

$$S_k(f) = \max_{D \sim D^{(k)}} |f(D) - f(D^{(k)})|,$$

where D and $D^{(k)}$ are neighboring data sets which are different in the kth element. In addition, $S(f) = \max(S_k(f))$.

In our scheme, the input of function f is $W \cdot D$ rather than D, and Alaggan et al. proves that $S_k(f) \leq w_k S(f)$ in [17].

So, we have:

$$\frac{\Pr[M(W \cdot D) = t]}{\Pr[M(W \cdot D') = t]} = \frac{h_{\epsilon(d_{ij})}(t - f(W \cdot D))}{h_{\epsilon(d_{ij})}(t - f(W \cdot D'))}$$
$$= \frac{\frac{\epsilon(d_{ij})}{2S(f)} \exp(\frac{-\epsilon(d_{ij})|t - f(W \cdot D)|}{S(f)})}{\frac{\epsilon(d_{ij})}{2S(f)} \exp(\frac{-\epsilon(d_{ij})|t - f(W \cdot D')|}{S(f)})} \leq \exp(\frac{\epsilon(d_{ij})|f(W \cdot D) - f(W \cdot D^{(k)})|}{S(f)})$$
$$\leq \exp(\frac{\epsilon(d_{ij}) S_k(f)}{S(f)}) \leq \exp(\frac{\epsilon(d_{ij}) w_k S(f)}{S(f)}) = \exp(w_k \epsilon(d_{ij})),$$

thus concluding the proof.

Theorem 6 *The random function M provide $\epsilon(d_{ij})$-differential privacy for each individual privacy weight W.*

Proof: W and W' are privacy weight vectors, and they differ in the kth element in each data set. So $W \cdot D$ and $W' \cdot D$ are neighboring data sets.

$$\frac{\Pr[M(W \cdot D) = t]}{\Pr[M(W' \cdot D) = t]} = \frac{h_{\epsilon(d_{ij})}(t - f(W \cdot D))}{h_{\epsilon(d_{ij})}(t - f(W' \cdot D))} = \frac{\frac{\epsilon(d_{ij})}{2S(f)} \exp(\frac{-\epsilon(d_{ij})|t - f(W \cdot D)|}{S(f)})}{\frac{\epsilon(d_{ij})}{2S(f)} \exp(\frac{-\epsilon(d_{ij})|t - f(W' \cdot D)|}{S(f)})}$$
$$\leq \exp(\frac{\epsilon(d_{ij})|f(W \cdot D) - f(W' \cdot D)|}{S(f)}) \leq \exp(\frac{\epsilon(d_{ij}) S(f)}{S(f)}) = \exp(\epsilon(d_{ij})),$$

thus concluding the proof.

5.2 Correlation Analysis

We use Monte-Carlo method to verify the correlation between noise. For universality, we simulate the process on the network example as shown in Fig. 2, where node 1 is the data owner, nodes 2-7 gets the query answer of the same query, respectively.

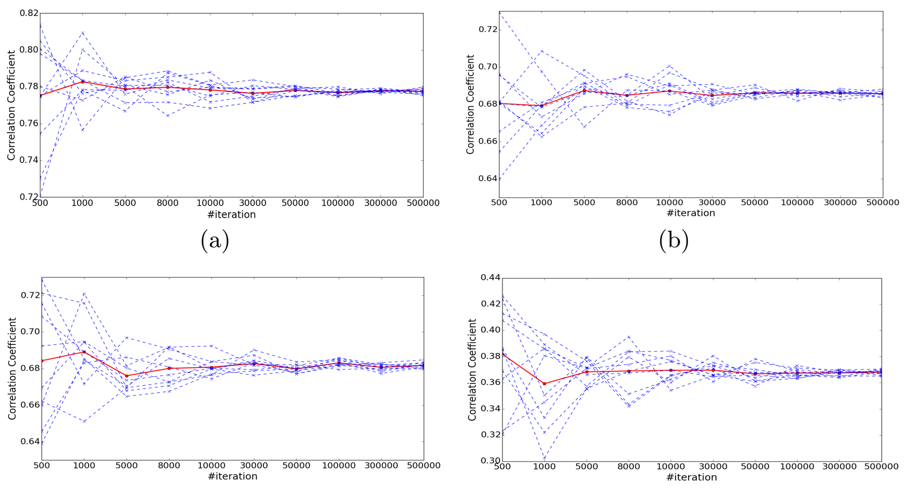
Specifically, we run the perturbation algorithm for over 100000 times, and use (2) to compute the Pearson Correlation Coefficient between results.

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}} \quad (2)$$

In Fig. 3, each blue dash line represents one set of results, the red solid line is the average of all computation results. As shown in Fig.3(a), after repeated iterations, the correlation coefficient between two different nodes converges to 0.78, which indicates that noise additions of different nodes are correlated.

Other than the collusion attack from a group of users, a single user can exploit the results he received to deduce more information. Therefore, to avoid this attack, different noise additions of the same query should be correlated, and they should be correlated to other nodes' noise additions as well.

In our scheme, we update the assigned perturbations after each query.



(a) (b) (c) (d)

Fig. 3. Correlation analysis by Monte-Carlo method: the x-dimension value is the iteration time in Monte-Carlo method, the y-dimension value is the correlation coefficient. (a) shows the correlation coefficient between two nodes' noise additions; (b)(c) show correlation coefficients after 50 and 100 updates, respectively; (d) shows the correlation of different nodes' noise after update.

Fig. 3(b) and Fig. 3(c) illustrate that the noise additions after 50 and 100 updates are still correlated with the original noise, and the correlation coefficient does not decrease with the increasing of the number of updates. We can observe that the correlation coefficient eventually converges to 0.68.

Meanwhile, Fig. 3(d) shows that the noise additions of different nodes after updates are still correlated.

6 Experiments

6.1 Data Sets

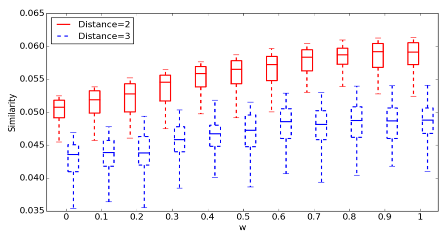
The experiments involve three networks' data: one synthetic network data and two real-world data. Real-world data come from the SNAP database.

- **BA Network:** BA model is a description of scale-free networks, which has the most similar topology to real OSNs. In our experiment, we use a BA network with 100 nodes and 291 edges. In addition, we build the data sets of each nodes by randomly choose 20 integer numbers from 0 to 10.

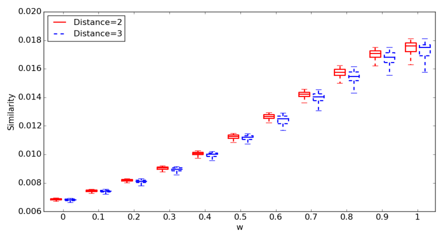
- **Facebook:** The Facebook network used in this experiment contains 213 nodes and 10305 edges. Each user keeps a data set as a 224 dimensional binary vector which includes user's name, locations, jobs, etc.
- **Gowalla:** Gowalla is a location-based social networking website where users share their locations by checking in. The network we use in this experiment consists of 34378 nodes and 626658 edges. The data that each user keeps is a two dimensional vector of latitude and longitude.

6.2 Privacy Performance

To demonstrate the privacy performance of our scheme, we compare the query results of different users. For better illustration, we modify W yet the query Q remains the same. Fig. 4(a) and Fig. 4(b) shows the similarity variation in BA network and Gowalla network, respectively. We vary two elements in W from 0



(a) BA



(b) Gowalla

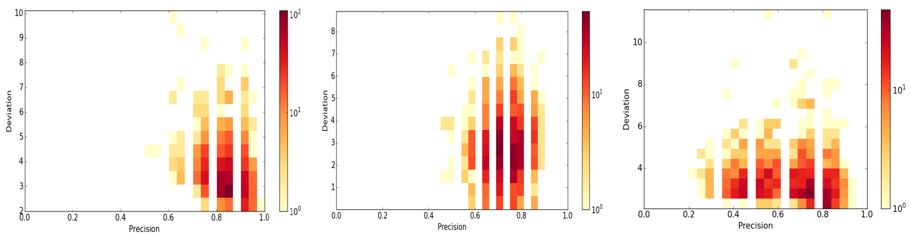
Fig. 4. The similarity between certain Q and D rises with the increase of w, where w represents the element we changed in private weight vectors, and $w \in [0, 1]$.

to 1 in each network, where the step size is 0.1.

The experiments show that the similarity rises with the increase of the elements in W, which is consist with our design goals that small w means stronger privacy protection. In addition, our results also illustrate that the similarities decrease as the distance increasing, which shows that our scheme realises the distance-grained differential privacy.

6.3 Utility Performance

In many practical user matching situations, the server often returns k IDs of users who have the top k similarities with Q instead of one user's ID. In our experiments, the server returns 20 most similar users' IDs to the inquirer. We evaluate the utility of our scheme by comparing the query answers with accurate query results. We define the precision as the ratio of correct IDs (i.e., IDs exist in accurate query result) to the size of query answer (i.e., 20). Fig. 5 shows the query



(a) BA (b) Facebook (c) Gowalla

Fig. 5. Precision of the query answers in (a)BA, (b)Facebook, and (C)Gowalla networks. Each point represents a query answer, where the x-dimension value is the precision, while y-dimension is the deviation between received query answers and real query answers. In the thermodynamic diagram, dark points represent the points shows up more frequently, yet the light color ones are those just occur occasionally.

precision in BA network, Facebook network, and Gowalla network, respectively. According to Section 6.2, privacy weight vectors have a great impact on the query answers. If a particular user wants to hide several elements in his data set (i.e., set quite small values to corresponding locations in the privacy weight vector), the query answers related to that user may significantly differ from the real answer. Therefore, to give a universal evaluation, we assume that all users set their privacy weight vectors as vectors of all ones, which satisfies the standard differential privacy protection.

In our scheme, query answers from distant data owners have larger noise, whereas close data owners add less noise to query answers. As a result, the IDs that inquirers received may contain more close data owners rather than real similar users, which may impact the precision of this experiment. However, from Fig. 5, we can observe that the majority of query answers have a precision value larger than 0.7, which proves that our scheme still provides good utility.

7 Conclusions

PPDM is an important issue in OSNs. However, most previous work assigns same privacy protection levels to all users in the network, which cannot satisfy users' personalized requirements. In this paper, we propose a scheme that provides fine-grained differential privacy for data mining in OSNs. Our scheme provides both distance-grained and item-grained differential privacy. Specifically, each inquirer in the network receives a query answer that satisfies $w_k\epsilon(d_{ij})$-differential privacy for each item in data owner's data set.

In addition, we investigate the collusion attack on differential privacy, and give a solution to this problem in PPDM. Finally, we conduct experiments on both synthetic and real-world data. Experiments show that our scheme guarantees good utility as well as good privacy protection.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China under Grant 61272479, the National 973 Program of China under Grant 2013CB338001, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA06010702.

References

1. Dwyer, C., Hiltz, S., Passerini, K.: Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In: 13th Americas Conference on Information Systems (AMCIS), pp. 339:1-339:13 (2007)
2. Zhang, C., Sun, J., Zhu, X., Fang, Y.: Privacy and security for online social networks: Challenges and opportunities. Network 24(4), 13-18. IEEE (2010)
3. Fogues, R., Such, J. M., Espinosa, A., Garcia-Fornes, A.: Open Challenges in Relationship-based privacy mechanisms for social network services. International Journal of Human-Computer Interaction 31(5), 350-370. (2015)
4. Wu, X., Zhu, X., Wu, G. Q., Ding, W.: Data mining with big data. IEEE Transactions on Knowledge and Data Engineering 26(1), 97-107. IEEE (2014)
5. Fung, B., Wang, K., Chen, R., Yu, P. S.: Privacy-preserving data publishing: A survey of recent developments. ACM Computing Surveys (CSUR) 42(4), 14. ACM (2010)
6. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(05), 557-570. (2002)
7. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD) 1(1), 3:1-3:52. ACM (2007)
8. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l -diversity. In: 23rd International Conference on Data Engineering (ICDE'07), pp. 106-115. IEEE (2007)
9. Wong, R. C. W., Li, J., Fu, A. W. C., Wang, K.: (α,k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In: 12th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'06), pp. 754-759. ACM (2006)
10. Dwork, C.: A firm foundation for private data analysis. Communications of the ACM 54(1), 86-95. ACM (2011)
11. Dwork, C.,: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1-12. Springer, Heidelberg (2006)
12. Baden, R., Bender, A., Spring, N., Bhattacharjee, B., Starin, D.: Persona: an online social network with user-defined privacy. ACM SIGCOMM Computer Communication Review 39(4), 135-146. ACM (2009)
13. Li, Y., Chen, M., Li, Q., Zhang, W.: Enabling multilevel trust in privacy preserving data mining. IEEE Transactions on Knowledge and Data Engineering 24(9), 1598-1612. IEEE (2012)
14. Ebadi, H., Sands, D., Schneider, G.: Differential privacy: Now it's getting personal. In: 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'15), pp. 69-81. (2015)
15. Jorgensen, Z., Yu, T., Cormode, G.: Conservative or liberal? Personalized differential privacy. In: 31st International Conference on Data Engineering (ICDE'15), pp. 13-17. IEEE (2015)

16. Koufogiannis, F., Pappas, G.: Diffusing Private Data over Networks. arXiv preprint arXiv:1511.06253. (2015)
17. Alaggan, M., Gambs, S., Kermarrec, A. M.: Heterogeneous differential privacy. arXiv preprint arXiv:1504.06998. (2015)
18. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265-284. Springer, Heidelberg (2006)
19. Yuan, M., Chen, L., Yu, P. S.: Personalized privacy protection in social networks. Proceedings of the VLDB Endowment 4(2), 141-150. ACM (2010)
20. Koufogiannis, F., Han, S., Pappas, G. J.: Gradual Release of Sensitive Data under Differential Privacy. arXiv preprint arXiv:1504.00429. (2015)
21. Zhang, N., Li, M., Lou, W.: Distributed data mining with differential privacy. In: IEEE International Conference on Communications (ICC'11). IEEE (2011)

Appendix

Proof of Theorem 3: Assume that $V_1 \sim Lap(\frac{1}{\epsilon_1})$, $V_2 \sim Lap(\frac{1}{\epsilon_2})$, where $\epsilon_1 < \epsilon_2$. The conditional probability $\Pr[V_1|V_2]$ has a density function $\phi(x)$. Additionally, $V_1 = h_{\epsilon_1}(x) = \epsilon_1 \exp(-\epsilon_1|x|)$, $V_2 = h_{\epsilon_2}(y) = \epsilon_2 \exp(-\epsilon_2|y|)$. We use $g(x, y)$ to denote the joint distribution density of V_1 and V_2. So $g(x, y)$ holds:

$$g(x, y) = \phi(y - x)h(x) \tag{3}$$

The density (3) should satisfy the following marginal distributions:

$$\int_{-\infty}^{\infty} g(x, y)dy = h_{\epsilon_1}(x) \quad \int_{-\infty}^{\infty} g(x, y)dx = h_{\epsilon_2}(y) \tag{4}$$

The equation (4) could be seen as a convolution operation $\int_{-\infty}^{\infty} \phi(y-x)h_{\epsilon_1}(x)dx = h_{\epsilon_2}(y)$. We use Convolution Theorem to solve this equation:

$$\mathcal{F}_{\phi}(s) = \frac{\mathcal{F}_{h_{\epsilon_2}}(s)}{\mathcal{F}_{h_{\epsilon_1}}(s)}, \tag{5}$$

where $\mathcal{F}$ denotes Fourier Transform. According to (5), we get:

$$\mathcal{F}_{\phi}(s) = \frac{\mathcal{F}_{h_{\epsilon_2}}(s)}{\mathcal{F}_{h_{\epsilon_1}}(s)} = \frac{1 - \frac{s^2}{\epsilon_2^2}}{1 - \frac{s^2}{\epsilon_1^2}} = (\frac{\epsilon_1}{\epsilon_2})^2(1 + \frac{\epsilon_2^2 - \epsilon_1^2}{\epsilon_1^2 + s^2})$$

We set $b(x) = |x|^{1-\frac{n}{2}} K_{\frac{n}{2}-1}(|x|)$, where K denotes the modified Bessel function of the second kind, and $\mathcal{F}_b(s) = \frac{(2\pi)^{\frac{n}{2}}}{1+s^2}$. So,

$$\mathcal{F}_{\phi}^{-1}(s) = (\frac{\epsilon_1}{\epsilon_2})^2[\delta(x) + (\frac{\epsilon_2^2}{\epsilon_1^2} - 1)\frac{\epsilon_1}{\sqrt{2\pi}}\sqrt{\epsilon_1 x}K_{-\frac{1}{2}}(\epsilon_1 x)]$$

$$\phi(x) \simeq (1 - \frac{\epsilon_1^2}{\epsilon_2^2})\frac{\epsilon_1}{\sqrt{2\pi}}\sqrt{\epsilon_1 x}K_{-\frac{1}{2}}(\epsilon_1 x)$$

More relevant details about the proof are given by Koufogiannis et al. in [16].