Privacy-aware Synthesizing for Crowdsourced Data

Mengdi Huai[1] , **Di Wang**[2] , **Chenglin Miao**[2] , **Jinhui Xu**[2] and **Aidong Zhang**[1]
[1]Department of Computer Science, University of Virginia, VA, USA
[2]Department of Computer Science and Engineering, SUNY at Buffalo, NY, USA
mh6ck@virginia.edu, {dwang45, cmiao, jinhui}@buffalo.edu, aidong@virginia.edu

Abstract

Although releasing crowdsourced data brings many benefits to the data analyzers to conduct statistical analysis, it may violate crowd users' data privacy. A potential way to address this problem is to employ traditional differential privacy (DP) mechanisms and perturb the data with some noise before releasing them. However, considering that there usually exist conflicts among the crowdsourced data and these data are usually large in volume, directly using these mechanisms can not guarantee good utility in the setting of releasing crowdsourced data. To address this challenge, in this paper, we propose a novel privacy-aware synthesizing method (i.e., PrisCrowd) for crowdsourced data, based on which the data collector can release users' data with strong privacy protection for their private information, while at the same time, the data analyzer can achieve good utility from the released data. Both theoretical analysis and extensive experiments on real-world datasets demonstrate the desired performance of the proposed method.

1 Introduction

In recent years, crowdsourcing has emerged as a popular and fast paradigm to solve many challenging data analysis tasks. Through the power of the crowd, the data collectors (e.g., hospitals, foundations and government agencies) can easily obtain large amounts of useful information. At the same time, the proliferation of new information techniques enables these data collectors to easily share their data that are collected from a crowd of users (e.g., patients, customers) with researchers or data analyzers. From such a wealth of shared data, researchers or data analyzers can discover useful knowledge or patterns to improve the quality of products, the management of public health, and so on. For example, in healthcare applications, the adverse events about a new drug can be easily collected by the hospitals from different patients. If the hospitals are willing to share these medical data, it would be very useful for the drug makers or medical research institutions to understand the efficacy of the drug.

Although the sharing of crowdsourced data brings many benefits, it may introduce privacy issues [Miao *et al.*, 2015; Shi and Wu, 2017; Miao *et al.*, 2017; Feng *et al.*, 2017]. Considering the above example, the hospital aims to collect the adverse events about a new drug from different patients. The patients usually trust the hospital and are willing to provide all the requested information. But if the hospital directly releases the patients' medical data to the drug makers, the private information of patients would be disclosed. Without effective privacy-preserving mechanisms, the patients may not allow their data to be released. Thus, it is essential to address how to enable the data collectors to release the crowdsourced data without disclosing users' private information.

Among existing privacy-preserving techniques, differential privacy (DP) has drawn significant attention as it can provide very rigorous privacy and utility guarantee [Dwork *et al.*, 2006]. However, this technique has several practical limitations when it is applied in the setting of releasing crowdsourced data. First of all, since the crowdsourced data on an object (e.g., the new drug) are usually collected from multiple users or sources, there inevitably exist conflicts among these data. The reasons include incomplete views of observations, environment noise, different knowledge bases and even the intent to deceive, etc. Directly applying DP on these data can not eliminate the conflicts, and this will certainly degrade the accuracy of the data analysis results. Additionally, DP is usually achieved by adding noise following the Laplace or exponential mechanisms [Dwork *et al.*, 2006]. The noise scale introduced by the Laplace mechanism is proportional to the number of data records, and such noise may make the data useless considering that crowdsourced dataset usually contains large amounts of data records. Although the noise introduced by the exponential mechanism does not depend on the number of data records, it depends on the domains of the input data [Dwork *et al.*, 2014], which may also make the crowdsourced data useless because these data usually have large domains.

To address the above challenges, in this paper, we propose a novel sampling-based **pri**vacy-aware **s**ynthesizing method for **crowd**sourced data (**PrisCrowd**). In this method, the data collector first learns the underlying patterns (i.e., densities) of the data for the objects through assigning each user a fine grained weight (or reliability degree) on each object. Then, for each object, the data collector samples a set of candidate synthetic data from the learned density. Finally, these synthetic data are subjected to our proposed privacy test, and the data collector only releases the synthetics that can pass the privacy test. The proposed method can not only extract high quality crowdsourced data via differentiating each user's fine grained reliability degrees on different objects but also achieve DP without injecting noise to the data. Both theoretical analysis and extensive experiments on real-world datasets are provided to verify the desirable performance of the proposed method.

2 Problem Setting

This paper considers a data releasing scenario, where a crowd of users and a data collector are involved. The users (or data sources) are the individuals (e.g., patients, customers) who can observe some objects (e.g., drugs, commodities) and provide claims for them. The data collector is an individual or institution (e.g., a hospital, an online store) who can collect the claims for these objects from a crowd of users and then release these claims to the public either voluntarily or for financial incentives. Here, we assume that the collector is trusted and the security threats mainly come from the public.

Problem formulation. Suppose there are N objects $\mathcal{O} = \{o_i\}_{i=1}^N$ which are observed by M users $\mathcal{U} = \{1, 2, ..., M\}$. For each object o_i, the claims of users are denoted as $\mathcal{X}_i = \{x_{i,s}\}_{s \in \mathcal{U}_i}$, where $x_{i,s}$ represents the claim provided by user s for object o_i and $\mathcal{U}_i$ represents the set of users who provide claims for this object. The claims collected by the data collector from all users are denoted as $\mathcal{X} = \{\mathcal{X}_i\}_{i=1}^N$, which need to be released to the public. Our goal in this paper is to design a mechanism based on which the data collector can release users' claims with strong privacy protection for their private information, while at the same time, the data analyzer can achieve good utility from the released data.

3 Preliminary

Definition 1 (Differential Privacy [Dwork *et al.*, 2006]**).** A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all neighboring datasets $D, D' \in \mathcal{X}^n$ and for all events S in the output space of $\mathcal{A}$, the following holds: $\Pr(\mathcal{A}(D) \in S) \leq e^{\epsilon} \Pr(\mathcal{A}(D') \in S) + \delta$.

The kernel density estimation (KDE) is a statistically-sound method to estimate a continuous distribution. Suppose there are n independent observations $X = \{x_1, ..., x_n\} \in \mathbb{R}^d$ following an unknown true density $f^*(x)$. The standard KDE $\tilde{f}(x)$ for the estimation of $f^*(x)$ at those points is defined as $\tilde{f}(x) = \frac{1}{n}\sum_{i=1}^n \mathcal{K}_{\mathcal{H}}(x, x_i)$. The following assumption will be used throughout the paper.

Assumption 1. For a vector $x_i \in \mathbb{R}^d$, we assume that the kernel function satisfies $\mathcal{K}_{\mathcal{H}}(x, x_i) = \mathcal{K}_{\mathcal{H}}(x - x_i)$. Furthermore, $\mathcal{K}_{\mathcal{H}}(x - x_i)$ is essentially a bump centered at x_i. More specifically, we take $\mathcal{K}_{\mathcal{H}}(x) = |\mathcal{H}|^{-\frac{1}{2}}\mathcal{K}(\mathcal{H}^{-\frac{1}{2}}z)$, where the kernel $\mathcal{K}$ itself is a probability density with zero mean and identity covariance and satisfying $\lim_{\|x\| \to \infty} \|x\|^d \mathcal{K}(x) = 0$.

Common choices for $\mathcal{K}$ that satisfy the above assumption include Gaussian and Epanechinikov kernels. As an example, Fig. 1 visualizes the construction of the standard KDE of 5 data points (black circles) using the well-known Gaussian kernel that is defined as $\mathcal{K}_{\mathcal{H}}(x - x_i) = (\frac{1}{\sqrt{2\pi}h})^d \exp(-\frac{\|x-x_i\|^2}{2h^2})$, where h is the bandwidth. The red curves are the component densities, and each red curve is a scaled version of the normal density curve centered at a datum. The standard KDE is obtained by summing these five scaled components.

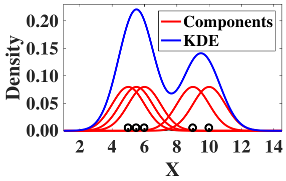


Figure 1: An example for the standard KDE

4 Methodology

4.1 Overview

To achieve the goal described in Section 2, we propose a novel privacy-aware synthesizing method for crowdsourced data (i.e., **PrisCrowd**), which contains two phases. In the first phase, we propose to use the weighted KDE as an intermediate representation of the raw data. This intermediate representation can well capture the statistical properties of the raw data. In the second phase, we first sample a set of candidate synthetic claims from the learned densities in the first phase, then each of these candidate claims is subjected to the proposed privacy test. If the claim passes the privacy test, it will be released, otherwise it will be discarded. The flowchart of the proposed two-phase method is shown in Fig. 2.

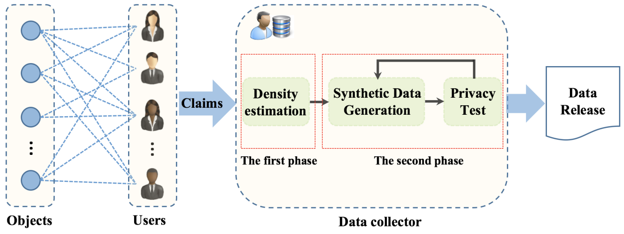


Figure 2: Privacy-aware synthesizing for crowdsourced data

4.2 Weighted KDE-based Data Representation

In order to share "wealth" data with the data analyzer, the data collector first needs to learn the characteristics or the underlying patterns of original data, i.e., the informative density distributions of objects. To estimate the density for each object, the standard kernel density estimation (KDE) can be adopted. Additionally, since different users may provide different claims for the same object, the reliability degrees (or weights) of these users should be taken into account when estimating the densities [Li *et al.*, 2014b; Li *et al.*, 2014a; Li *et al.*, 2016; Miao *et al.*, 2019]. However, the standard KDE cannot differentiate the importance of users (i.e., user reliability degrees). In order to learn users' reliability and compute the densities of objects simultaneously, we propose a novel method which can estimate users' global and local weights, and then combine them to learn objects' informative density distributions. A user's global weight reflects his capability to provide truthful information for all the objects, and the local weights represent that this user may have different confidence when providing claims for different objects. The advantage of the proposed method is that it can estimate reasonable reliability for each user, and in turn, learn the accurate informative density distributions for objects.

Global Weight Estimation

To evaluate the overall importance of users, the data collector assigns a global weight $g_s \in \mathbb{R}$ to each user s. Meanwhile, we can obtain a global density f_i^g for each object o_i, which should be close to the distribution of claims from reliable users. The distribution of the input claims $\mathcal{X}_i$ can be obtained by $\mathcal{K}_{\mathcal{H}_i}(x, \mathcal{X}_i)$ ($x \in \mathbb{R}$ is a variable), i.e., the kernel function associated with a reproducing kernel Hilbert space $\mathcal{H}_i$. To minimize the weighted deviation from the estimated density $Q = \{f_i^g(x)\}_{i=1}^N$ to the multi-user input $\mathcal{X} = \{\mathcal{X}_i\}_{i=1}^N$, we propose the following optimization framework

$$\min_{G,Q} \sum_{s \in \mathcal{U}} g_s \sum_{i \in \mathcal{E}_s} d_{\mathcal{H}_i}(\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s}), f_i^g(x)) \tag{1}$$

$$s.t. \sum_{s \in \mathcal{U}} \exp(g_s) = 1,$$

where $\mathcal{E}_s$ denotes the set of objects observed by user s, $G = \{g_s\}_{s \in \mathcal{U}}$ and the normalized squared loss $d_{\mathcal{H}_i}(\mathcal{K}_{\mathcal{H}_i}(\cdot, \cdot), f_i^g(x))$ is defined as $d_{\mathcal{H}_i}(\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s}), f_i^g(x)) = \|\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s}) - f_i^g(x)\|^2_{\mathcal{H}_i}$. The global loss function (i.e., Eq. (1)) extends the framework in [Li *et al.*, 2014b] from real space to Hilbert space. We can use an iterative procedure to solve it. Specifically, in the k-th iteration, g_s is updated as

$$g_s^{(k+1)} = -\log \frac{\sum_{i \in \mathcal{E}_s} d_{\mathcal{H}_i}(\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s}), f_i^{g(k)}(x))}{\sum_{s' \in \mathcal{U}} \sum_{i \in \mathcal{E}_{s'}} d_{\mathcal{H}_i}(\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s'}), f_i^{g(k)}(x))}, \tag{2}$$

where $f_i^{g(k)}(x) = \sum_{t \in \mathcal{U}_i} g_t^{(k)} \mathcal{K}_{\mathcal{H}_i}(x, x_{i,t}) / (\sum_{t \in \mathcal{U}_i} g_t^{(k)})$. Eq. (2) shows that a user's global weight is inversely proportional to the distance between its claims and the estimated global densities at the log scale. Users whose claims are close to the derived global densities will have higher global weights.

Local Weight Estimation

As described above, each user may have different confidence when providing claims for different objects. Thus, we need to model the local weight of each user on every object, which will in turn help to infer the accurate density estimations. A potential way to achieve this is to establish a square loss function. However, it leads to a problem that each user would receive the same local weight, and the trustworthiness of the claims provided by different users would be equal. In order to address this problem, we use Hampel loss function [Hampel *et al.*, 2011]:

$$\zeta_{q_1,q_2,q_3}(y) = \begin{cases} y^2/2, & 0 \leq y < q_1 \\ q_1 y - q_1^2/2, & q_1 \leq y < q_2 \\ \frac{q_1(y-q_3)^2}{2(q_2-q_3)} + \frac{q_1(q_2+q_3-q_1)}{2}, & q_2 \leq y < q_3 \\ q_1(q_2 + q_3 - q_1)/2, & q_3 \leq y, \end{cases}$$

where $q_1 < q_2 < q_3$ are predefined nonnegative parameters. These parameters allow us to decrease the trustworthiness of "bad" claims and increase that of "good" ones for each object, so the importance of users can be well distinguished.

Since we incorporate users' reliability into estimating the local densities, the local kernel density of object o_i can be defined as $f_i^l(x) = \sum_{s \in \mathcal{U}_i} l_{i,s} \mathcal{K}_{\mathcal{H}_i}(x, x_{i,s})$, where $l_{i,s}$ is the local weight of the user s on object o_i. Thus, the objective function for estimating $l_i = \{l_{i,s}\}_{s \in \mathcal{U}_i}$ is

$$J(l_i) = \min_{l_i} \sum_{s' \in \mathcal{U}_i} \zeta(\|\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s'}) - \sum_{s \in \mathcal{U}_i} l_{i,s} \mathcal{K}_{\mathcal{H}_i}(x, x_{i,s})\|), \tag{3}$$

where $\|\cdot\|$ denotes the difference between users' claims and the estimated local density $f_i^l(x)$. This objective function is not convex, i.e., Eq. (3) does not have a closed form solution. Fortunately, it is possible to approximate $l_i = \{l_{i,s}\}_{s \in \mathcal{U}_i}$ with a standard iteratively re-weighted least squares (IRWLS) algorithm. The iterative procedure for computing $\{l_{i,s}\}_{s \in \mathcal{U}_i}$ is

$$l_{i,s}^{(k+1)} = \frac{\zeta(\|\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s}) - \sum_{t \in \mathcal{U}_i} l_{i,t}^{(k)} \mathcal{K}_{\mathcal{H}_i}(x, x_{i,t})\|)}{\sum_{s' \in \mathcal{U}_i} \zeta(\|\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s'}) - \sum_{t \in \mathcal{U}_i} l_{i,t}^{(k)} \mathcal{K}_{\mathcal{H}_i}(x, x_{i,t})\|)},$$

where k denotes the number of iterations. This equation shows that users would receive lower weights when they provide "bad" claims which deviate largely from the center $f_i^l(x) = \sum_{t \in \mathcal{U}_i} l_{i,t} \mathcal{K}_{\mathcal{H}_i}(x, x_{i,t})$.

Combined Weight Estimation

For each user s, to measure the consistency degree of the global and local weights (i.e., g_s and $l_{i,s}$), we define a mixture weight, named combined weight $c_{i,s}$. To learn the combined weight, the relative entropy is employed, which minimizes the information loss between user's global weight and local weight. The smaller the relative entropy value of those weights, the higher the degree of their consistency. The objective of the combined model is

$$\min_{\{c_{i,s}\}_{s \in \mathcal{U}_i}} \sum_{s \in \mathcal{U}_i} c_{i,s} \log \frac{c_{i,s}}{l_{i,s}} + \sum_{s \in \mathcal{U}_i} c_{i,s} \log \frac{c_{i,s}}{g_s}.$$

$$s.t. \sum_{s \in \mathcal{U}_i} c_{i,s} = 1, c_{i,s} \geq 0. \tag{4}$$

By solving Eq. (4), we can obtain the combined weight $c_{i,s}$ of user s on the object o_i as $c_{i,s} = \sqrt{l_{i,s} g_s} / (\sum_{t \in \mathcal{U}_i} \sqrt{l_{i,t} g_t})$. Based on the learned combined weights, we can obtain the density of object o_i which is the weighted sum of claims in Hilbert space and is given as

$$f_i(x) = \sum_{s \in \mathcal{U}_i} \frac{\sqrt{l_{i,s} g_s}}{\sum_{t \in \mathcal{U}_i} \sqrt{l_{i,t} g_t}} \mathcal{K}_{\mathcal{H}_i}(x, x_{i,s}). \tag{5}$$

4.3 Privacy Test-based Synthetics Release

To provide strong privacy protection for users' private information, in this section, we propose a privacy test-based synthetics release method, which contains two steps: *Candidate synthetics generation* and *Privacy test for candidate synthetics*. In the first step, we sample a set of synthetic claims from the learned density in Eq. (5) as the candidate data to release. Then, in the second step, these sampled synthetics are subjected to a privacy test. If a synthetic claim passes the test, it will be released, otherwise it will be discarded.

Candidate Synthetics Generation

We first discuss how to generate the synthetic claims $\tilde{\mathcal{X}}_i$ for each object o_i. Specifically, we generate each element in $\tilde{\mathcal{X}}_i$ as follows:

1. Select a random integer $s \in \mathcal{U}_i$ with probability $c_{i,s}$;
2. Generate a synthetic claim $\tilde{x}_{i,s}$ through sampling from the probability distribution $\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s})$.

Here $c_{i,s}$ can be treated as the sampling probability that determines whether $x_{i,s}$ is selected or not. In this step, we aim to select some seed data (e.g., $x_{i,s}$) and then probabilistically transform them into the synthetic data. The sampling mechanism used here can increase the uncertainty of the adversary about whether a user's data is in the released dataset or not, and thus it can help to protect users' privacy to some extent. However, it is not enough to only use the sampling mechanism, directly releasing the sampled data can still violate users' privacy [Gehrke *et al.*, 2012]. To tackle this problem, we design the following privacy test mechanism to further prevent users' private information from being disclosed.

Privacy Test for Candidate Synthetics

To prevent an adversary from deducing that a particular claim in $\mathcal{X}_i$ is more responsible for generating the released synthetic data than other claims, the following randomized privacy test mechanism is proposed. Each candidate synthetic data in $\tilde{\mathcal{X}}_i$ is subjected to the randomized privacy test, and it is released only when it passes this test.

Suppose $k \geq 1$ and $\gamma > 1$ are the privacy parameters, and ϵ_0 is the randomness parameter. Let $\mathcal{M}(\cdot)$ denote the above synthetic data generation procedure, which samples a candidate synthetic based on a seed data. Given $x_{i,s} \in \mathcal{X}_i$, we use $\Pr\{\tilde{x}_{i,s} = \mathcal{M}(x_{i,s})\}$ to denote the probability that a synthetic data $\tilde{x}_{i,s}$ is generated based on $\mathcal{M}(\cdot)$. Then the privacy test procedure for $\tilde{x}_{i,s}$ is described as follows:

1. Randomize k by adding a noise: $\tilde{k} = k + z$, where $z \sim Lap(1/\epsilon_0)$ is sampled from the Laplace Distribution.
2. Let $a \geq 0$ be the integer that satisfies the inequalities $\gamma^{-a-1} < \Pr\{\tilde{x}_{i,s} = \mathcal{M}(x_{i,s})\} \leq \gamma^{-a}$.
3. Let k' be the number of records $x_{i,s'} \in \mathcal{X}_i$ that satisfies $\gamma^{-a-1} < \Pr\{\tilde{x}_{i,s} = \mathcal{M}(x_{i,s'}\} \leq \gamma^{-a}$.
4. If $k' \geq \tilde{k}$, $\tilde{x}_{i,s}$ passes the test, otherwise it fails.

Note that k' denotes the number of possible data seeds that can generate $\tilde{x}_{i,s}$ with a probability value falling into a very stringent interval $[\gamma^{-a-1}, \gamma^{-a}]$. The threshold parameter $\tilde{k}$ prevents releasing sensitive synthetic data. Under this randomized privacy test, a candidate synthetic data is released only when there are at least $\tilde{k}$ possible data seeds that can generate $\tilde{x}_{i,s}$. Intuitively, the larger the value of k, the larger the number of the possible seed data that are indistinguishable from $x_{i,s}$. Also, the less the value of γ, the more difficult to distinguish $x_{i,s}$ from other possible seed data. Algorithm 1 summarizes the proposed privacy test-based synthetics release procedure, in which m denotes the number of synthetic claims that need to be released for object o_i.

Algorithm 1 Private test-based synthetics release for o_i

```
Input: $\{c_{i,s}\}_{s\in\mathcal{U}_i}$, $\mathcal{X}_i = \{x_{i,s}\}_{s\in\mathcal{U}_i}$, $k$, $\gamma$, $\epsilon_0$, and $m$.
Output: The dataset $\tilde{\mathcal{X}}_i$ that can be released.
$\tilde{\mathcal{X}}_i = \emptyset$
while $|\tilde{\mathcal{X}}_i| < m$ do
    Select a random integer $s \in \mathcal{U}_i$ with probability $c_{i,s}$;
    Generate a synthetic claim $\tilde{x}_{i,s}$ based on the
    probability distribution $\mathcal{K}_{\mathcal{H}_i}(x, x_{i,s})$;
    Conduct randomized privacy test for $\tilde{x}_{i,s}$;
    if $\tilde{x}_{i,s}$ passes the privacy test then
        $\tilde{\mathcal{X}}_i = \tilde{\mathcal{X}}_i \bigcup \{\tilde{x}_{i,s}\}$;
    end if
end while
return $\tilde{\mathcal{X}}_i$;
```

4.4 Theoretical Analysis

Consistency Analysis

In Section 4.3, we generate the synthetic claims for object o_i by sampling from the mixture distribution $f_i(x)$, i.e., $\tilde{x}_{i,s} \sim f_i(x)$. After obtaining the dataset $\tilde{\mathcal{X}}_i = \{\tilde{x}_{i,s}\}_{s=1}^m$, a basic question here is that how well the generated dataset can reflect the original density function $f_i(x)$. Since each $\tilde{x}_{i,s}$ is sampled from $f_i(x)$ independently, the density function over $\{\tilde{x}_{i,s}\}_{s=1}^m$ can be denoted as $\tilde{f}_i(x) = \frac{1}{m}\sum_{s=1}^m \mathcal{K}_{\mathcal{H}_i}(x, \tilde{x}_{i,s})$. In Theorem 1, we provide the expected squared L_2-norm distance between $f_i(x)$ and $\tilde{f}_i(x)$.

Theorem 1. Under Assumption 1 for $\mathcal{K}_{\mathcal{H}_i}$ with the diagonal bandwidth matrix $\mathcal{H}_i = \hat{h}^2 I_d$, we further assume that the support of $\mathcal{K}(z)$ satisfies $\|z\| \leq 1$. Then, the expected squared L_2-norm distance between $f_i(x)$ and $\tilde{f}_i(x)$, i.e., $J = \mathbb{E}[\int (f_i(x) - \tilde{f}_i(x))^2 dx]$, satisfies

$$J \leq 4A\hat{h} + A^2\hat{h}^2 V + \frac{B}{m\hat{h}^d} + \frac{ABV}{m\hat{h}^{d-1}}, \qquad (6)$$

where $A = \sup_{x\in\mathbb{R}^d} \|\nabla f_i(x)\|$, $B = \int (\mathcal{K}(z))^2 dz$ and V is the volume of the support of $f_i(x)$. The expectation is respected to $\{\tilde{x}_{i,s}\}_{s=1}^m \sim f_i(x)$. This theorem is a general result for d dimensional case, in this paper, the value of d is 1.

Privacy Analysis

Next, we conduct privacy analysis for Algorithm 1. Based on Theorem 2, we know that the proposed algorithm is differentially private.

Theorem 2. Note that the input parameters of Algorithm 1 include $\{c_{i,s}\}_{s\in\mathcal{U}_i}$, $k \geq 1$, $\gamma > 0$, and ϵ_0. For any neighboring datasets $\mathcal{X}_i$ and $\mathcal{X'}_i$ such that $|\mathcal{X}_i|, |\mathcal{X'}_i| \geq k$ and any integer $1 \leq t < k$, we have that Algorithm 1 is (ϵ, δ)-differentially private, where $\epsilon = \epsilon_0 + \log(1 + \frac{\gamma}{t}\frac{\max_{s\in\mathcal{U}_i} c_{i,s}}{\min_{s\in\mathcal{U}_i} c_{i,s}})$, $\delta = |\mathcal{U}_i| \max_{s\in\mathcal{U}_i} c_{i,s} e^{-\epsilon(k-t)}$.

Remark 1. Note that the proposed Algorithm 1 is different from the mechanism in [Bindschaedler *et al.*, 2017]. The probability of choosing the seed $x_{i,s}$ is non-uniform in Algorithm 1 while that is uniform in [Bindschaedler *et al.*, 2017].

Dataset	# users	# objects
Population	2,344	1,124
Stock	55	5,521
Indoor Floorplan	247	129

Table 1: The statistics of the adopted datasets.

The non-uniform property may generate different parameters of differential privacy. When $\max_{s\in\mathcal{U}_i} c_{i,s} = \min_{s\in\mathcal{U}_i} c_{i,s} = 1/|\mathcal{U}_i|$ (i.e., we uniformly sample the seed $x_{i,s}$), the above Theorem 2 is actually Theorem 1 in [Bindschaedler *et al.*, 2017]. Thus, Theorem 2 in our paper is a generalization of Theorem 1 in [Bindschaedler *et al.*, 2017]. Although the main idea of the proof for Theorem 2 is similar to that in [Bindschaedler *et al.*, 2017], the details are quite different: in [Bindschaedler *et al.*, 2017] the proof consider $\mathcal{X}'_i = \mathcal{X}_i \bigcup \{x_{i,s'}\}$ as the neighborhood dataset while ours consider $\mathcal{X}'_i = \{\mathcal{X}_i - \{x_{i,s}\}\} \bigcup \{x_{i,s'}\}$ as the neighborhood dataset. That is because if we add one data record, the probability of sampling seeds, i.e., $\{c_{i,s}\}$, will be totally changed. So the proof in [Bindschaedler *et al.*, 2017] cannot satisfy our case.

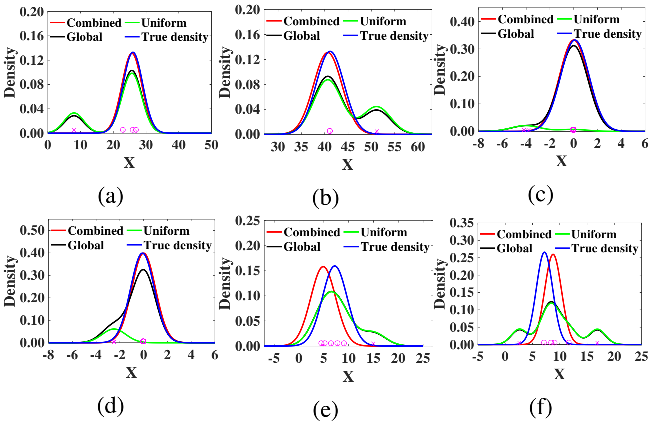
5 Experiments

Performance measure. To evaluate the performance of our method, we adopt two measure metrics: *the integrated squared error (ISE)* and *the squared integrated squared error (SISE)*. *ISE* is defined as: $\sum_{i=1}^{N}\int_{-\infty}^{+\infty}(f_i - \tilde{f}_i)^2dx$, where f_i and $\tilde{f}_i$ are respectively the original density and the density derived from the synthetic data for object o_i. *SISE* is defined as: $\sum_{i=1}^{N}(\int_{-\infty}^{+\infty}(f_i - \tilde{f}_i)^2dx)^2$. Compared with *ISE*, *SISE* tends to penalize more on the large distance and less on the small distance. Since the goal of the collector is to release the data whose pattern is similar to the true underlying pattern for the objects, the lower the *ISE* or *SISE*, the better the method.

Datasets. We adopt the following real-world datasets for our experiments: *Population Dataset* [Pasternack and Roth, 2010; Wan *et al.*, 2016], *Stock Dataset* [Li *et al.*, 2012], and *Indoor Floorplan Dataset* [Li *et al.*, 2014a]. The statistics of these datasets are provided in Table 1.

Baselines. Here, we adopt two baselines, i.e. *Basic* and *Uniform*. In the *Basic* method, the data collector adds three level noise to the original data: $\epsilon = 0.1$ (Strong), $\epsilon = 1$ (normal) and $\epsilon = 10$ (Weak). In the *Uniform* method, the collector treats all users equally and the entities' densities are learned with the uniformly weighted kernel density estimation. Here, the synthetic data generation and the privacy tests procedures are the same with those in our proposed method.

Case study. In order to investigate the advantages of the users' combined weights, we conduct case studies on the three real-world datasets. For each dataset, we randomly select two objects as the cases, and then estimate their densities. The estimated densities are shown in Fig. 3. The red line in each subfigure represents the density estimated based on users' combined weights. The black line represents the result estimated only based on the global weight of each user. We also conduct estimations without considering user quality, i.e., treating all users equally, and the estimated density for each object is represented with the green line. The results in Fig.3 show that the densities estimated based on users' combined weights are the closest to the true densities which are represented with the blue lines. Additionally, we show the claims of each object in this figure with magenta circles and crosses. We can see that some claims (i.e., the magenta crosses) are far away from others (i.e., the magenta circles). These claims are usually provided by the users with low weights, and they can be treated as outliers when estimating each object's density. The results show that the method based on the combined weight is more robust to outliers than the methods which only adopt users' global weights or treat all users equally. In other words, the estimated density for each object based PrisCrowd can well reflect the underlying true density of this object.



(a) (b) (c)

(d) (e) (f)

Figure 3: Case study on real-world datasets. (a) and (b): the two cases for Population dataset. (c) and (d): the two cases for Stock dataset. (e) and (f): the two cases for Indoor Floorplan dataset.

Accuracy comparison. In this experiment, we evaluate the accuracy (or quality) of the published synthetic data and explore whether these data can well reflect the underlying true densities of the objects. Here we assume that the data collector releases 30 synthetic claims for each object to the public. The parameters γ and k are set as 4 and 5 respectively. In order to evaluate the accuracy of the synthetic claims, we first derive the density (i.e., $\tilde{f}_i$) of each object from the synthetic data, and then calculate *ISE* and *SISE* for each dataset. The results are shown in Table 2, from which we can see the proposed approach performs much better than the baseline methods on all real-world datasets. That is to say, the synthetic data generated based on our proposed method could well preserve the characteristics of the underlying pattern for the objects. Additionally, the results also show that the advantages of our proposed approach on the Stock dataset is larger than that on the Population and Indoor Floorplan datasets. The reason is that there are more outlying data points in the Stock dataset, and our proposed approach is robust to these outliers while the baseline methods are very sensitive to them.

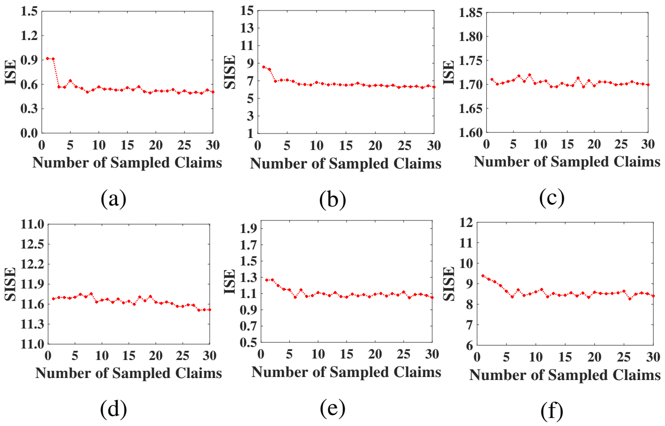
The effect of the number of sampled claims. In this experiment, we evaluate the effect of the number of sampled claims for each object on the performance of the proposed method. Here we vary the number of the sampled claims for each ob-

Measure	Method	Population	Stock	Indoor
ISE	**PrisCrowd**	**0.479**	**1.699**	**1.051**
	Uniform	0.628	17.628	1.220
	Basic(Strong)	1.183	12.799	1.943
	Basic(Normal)	1.119	9.867	1.937
	Basic(Weak)	0.866	2.125	1.882
SISE	**PrisCrowd**	**6.209**	**11.430**	**8.405**
	Uniform	8.502	47.217	11.112
	Basic(Strong)	12.013	40.420	15.111
	Basic(Normal)	11.768	35.391	15.046
	Basic(Weak)	10.149	15.412	14.723

Table 2: Accuracy comparison on the real-world datasets

ject from 1 to 30 and then calculate the *ISE* and *SISE* on the three real-world datasets. The results are shown in Fig. 4, from which we can see the *ISE* and *SISE* gradually get flattened with the increase of the number of the sampled claims for each object. Take the population dataset as an example, when the number of sampled claims is lager than 10, the values of *ISE* and *SISE* are almost the same. That is to say, the released data generated based on our proposed method could well reflect the underlying patterns of the objects even only a few claims are sampled for each object.

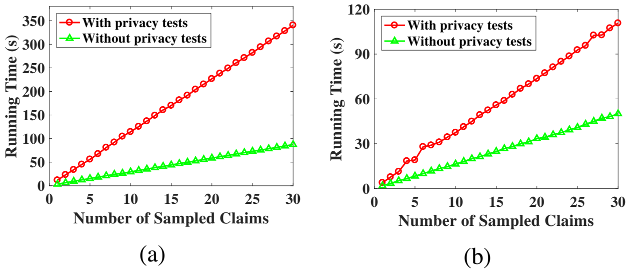
Computational cost. Next we evaluate the computational cost of the synthetic claims generation procedure, i.e., the second phase in our proposed method. In this experiment, we only generate synthetic claims for the objects whose ground truths can be achieved from the original datasets, and consider two scenarios, i.e., with privacy tests and without privacy tests. Then we vary the number of the sampled claims for each object from 1 to 30. The running time of the synthetic claims generation procedure for the Population and Indoor Floorplan datasets is shown in Fig. 5, from which we can see the running time in the two scenarios is approximately linear with respect to the number of sampled claims for each object. Additionally, the results also show that the privacy test step introduce extra computational cost during the released data generation procedure. This is because each candidate synthetic data record needs to be tested in the privacy test step. Since good utility can be achieved based on our proposed method even only a few synthetic claims are generated for each object, the computational cost is tolerable in practice.



(a) (b) (c)

(d) (e) (f)

Figure 4: Accuracy w.r.t. Number of Sampled Claims. (a) and (b): Population. (c) and (d): Stock. (e) and (f): Indoor Floorplan.



(a) (b)

Figure 5: Running time vs. number of sampled claims for each object. (a): Population. (b): Indoor Floorplan.

6 Related Work

Recently, various differential private data release approaches have been proposed. Those methods can be roughly partitioned into two categories: the interactive ones and the non-interactive ones. In an interactive method [Li *et al.*, 2010; Hardt and Rothblum, 2010; Roth and Roughgarden, 2010], a data analyzer can pose queries via a private mechanism, and a dataset owner answers these queries in response. In the non-interactive framework [Nissim *et al.*, 2007; Bindschaedler and Shokri, 2016; Blum *et al.*, 2013; Wang *et al.*, 2018; Wang *et al.*, 2019], a data owner releases the private version of the original data. Once data are published, the owner has no further control over the published data.

The method in our paper is non-interactive. The typical approach to protect data privacy in the non-interactive context is to directly add noise, which is taken by [Bindschaedler and Shokri, 2016; Blum *et al.*, 2013]. These works are either computationally infeasible on high-dimensional data, or practically ineffective because of their large utility costs. There are also some other works [Bindschaedler and Shokri, 2016] which release private data without adding noise, but they are unsuitable to be used in the newly appearing crowdsourcing setting considered in this paper where multi-sources provide multi-observations for multi-objects.

7 Conclusions

In this paper, we propose a novel privacy-aware synthesizing method for crowdsourced data. Based on this method, the data collector can release the crowdsourced data with strong privacy protection for users' private information, while at the same time, the data analyzer can achieve good utility from the released data. Both theoretical analysis and extensive experiments on real-world datasets verify the effectiveness of the proposed method.

Acknowledgements

This work was supported in part by the US National Science Foundation (NSF) under grants IIS-1514204 and CCF-1716400. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[Bindschaedler and Shokri, 2016] Vincent Bindschaedler and Reza Shokri. Synthesizing plausible privacy-preserving location traces. In *Proceedings of S&P*, pages 546–563, 2016.

[Bindschaedler *et al.*, 2017] Vincent Bindschaedler, Reza Shokri, and Carl A Gunter. Plausible deniability for privacy-preserving data synthesis. *Proceedings of the VLDB Endowment*, 10(5):481–492, 2017.

[Blum *et al.*, 2013] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):12, 2013.

[Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284, 2006.

[Dwork *et al.*, 2014] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[Feng *et al.*, 2017] Wei Feng, Zheng Yan, Hengrun Zhang, Kai Zeng, Yu Xiao, and Y Thomas Hou. A survey on security, privacy, and trust in mobile crowdsourcing. *IEEE Internet of Things Journal*, 5(4):2971–2992, 2017.

[Gehrke *et al.*, 2012] Johannes Gehrke, Michael Hay, Edward Lui, and Rafael Pass. Crowd-blending privacy. In *Proceedings CRYPTO*, pages 479–496. 2012.

[Hampel *et al.*, 2011] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: The approach based on influence functions*. Wiley Online Library, 2011.

[Hardt and Rothblum, 2010] Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proceedings of FOCS*, pages 61–70, 2010.

[Li *et al.*, 2010] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of PODS*, pages 123–134, 2010.

[Li *et al.*, 2012] Xian Li, Xin Luna Dong, Kenneth Lyons, Weiyi Meng, and Divesh Srivastava. Truth finding on the deep web: Is the problem solved? *Proceedings of the VLDB Endowment*, 6(2):97–108, 2012.

[Li *et al.*, 2014a] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. A confidence-aware approach for truth discovery on long-tail data. *PVLDB*, 2014.

[Li *et al.*, 2014b] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of SIGMOD*, pages 1187–1198, 2014.

[Li *et al.*, 2016] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 28(8):1986–1999, 2016.

[Miao *et al.*, 2015] Chenglin Miao, Wenjun Jiang, Lu Su, Yaliang Li, Suxin Guo, Zhan Qin, Houping Xiao, Jing Gao, and Kui Ren. Cloud-enabled privacy-preserving truth discovery in crowd sensing systems. In *Proceedings of SenSys*, pages 183–196, 2015.

[Miao *et al.*, 2017] Chenglin Miao, Lu Su, Wenjun Jiang, Yaliang Li, and Miaomiao Tian. A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems. In *Proceedings of INFOCOM*, pages 1–9, 2017.

[Miao *et al.*, 2019] Chenglin Miao, Wenjun Jiang, Lu Su, Yaliang Li, Suxin Guo, Zhan Qin, Houping Xiao, Jing Gao, and Kui Ren. Privacy-preserving truth discovery in crowd sensing systems. *ACM Transactions on Sensor Networks (TOSN)*, 15(1):9, 2019.

[Nissim *et al.*, 2007] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of STOC*, pages 75–84, 2007.

[Pasternack and Roth, 2010] Jeff Pasternack and Dan Roth. Knowing what to believe (when you already know something). In *Proceedings of Coling*, pages 877–885, 2010.

[Roth and Roughgarden, 2010] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of STOC*, pages 765–774, 2010.

[Shi and Wu, 2017] Xinghua Shi and Xintao Wu. An overview of human genetic privacy. *Annals of the New York Academy of Sciences*, 1387(1):61–72, 2017.

[Wan *et al.*, 2016] Mengting Wan, Xiangyu Chen, Lance Kaplan, Jiawei Han, Jing Gao, and Bo Zhao. From truth discovery to trustworthy opinion discovery: An uncertainty-aware quantitative modeling approach. In *Proceedings of SIGKDD*, pages 1885–1894, 2016.

[Wang *et al.*, 2018] Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical risk minimization in non-interactive local differential privacy revisited. In *Proceedings of NeurIPS*, pages 965–974, 2018.

[Wang *et al.*, 2019] Di Wang, Adam Smith, and Jinhui Xu. Noninteractive locally private learning of linear models via polynomial approximations. In *Algorithmic Learning Theory*, pages 897–902, 2019.