DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies ; 2018 (3):84–104

Takao Murakami*, Hideitsu Hino, and Jun Sakuma

Toward Distribution Estimation under Local Differential Privacy with Small Samples

Abstract: A number of studies have recently been made on discrete distribution estimation in the local model, in which users obfuscate their personal data (e.g., location, response in a survey) by themselves and a data collector estimates a distribution of the original personal data from the obfuscated data. Unlike the centralized model, in which a trusted database administrator can access all users' personal data, the local model does not suffer from the risk of data leakage. A representative privacy metric in this model is LDP (Local Differential Privacy), which controls the amount of information leakage by a parameter ϵ called privacy budget. When ϵ is small, a large amount of noise is added to the personal data, and therefore users' privacy is strongly protected. However, when the number of users N is small (e.g., a small-scale enterprise may not be able to collect large samples) or when most users adopt a small value of ϵ, the estimation of the distribution becomes a very challenging task.

The goal of this paper is to accurately estimate the distribution in the cases explained above. To achieve this goal, we focus on the EM (Expectation-Maximization) reconstruction method, which is a state-of-the-art statistical inference method, and propose a method to correct its estimation error (i.e., difference between the estimate and the true value) using the theory of Rilstone et al. We prove that the proposed method reduces the MSE (Mean Square Error) under some assumptions. We also evaluate the proposed method using three large-scale datasets, two of which contain location data while the other contains census data. The results show that the proposed method significantly outperforms the EM reconstruction method in all of the datasets when N or ϵ is small.

Keywords: Data privacy, Location privacy, Local differential privacy, EM reconstruction method

DOI 10.1515/popets-2018-0022

Received 2017-11-30; revised 2018-03-15; accepted 2018-03-16.

1 Introduction

With the widespread use of personal computers, GPS-equipped devices (e.g., mobile phones, in-car navigation systems), and IoT devices (e.g., smart meters, home monitoring devices), personal data are increasingly collected and analyzed for various purposes. For example, a great amount of location data (a.k.a. Spatial Big Data [46]) can be analyzed to find commonly frequented public areas [51], or can be made public to provide traffic information to users [26]. Power-consumption data from smart meters can be analyzed to extract typical daily consumption patterns in households [23], or to identify the right customers to target for demand response programs [8]. Personal data (e.g., age, gender, income, marital satisfaction) collected via survey sampling can be used to infer the statistics (e.g., histogram, heavy hitters) of a target population.

While these data are useful for discovering knowledge or improving the quality of service, the collection of personal data can lead to a breach of users' privacy. For example, users' home/workplace pairs [18], long-term properties (e.g., age, job position, smoking habit) [36], and social relationship [14] can be inferred from their disclosed locations. In-home activities (e.g., presence/absence, appliance use, sleep/wake cycle) can also be inferred from power-consumption data [35]. Furthermore, various kinds of personal data from different sources can be linked and aggregated into a *user profile* [21, 42], and can be provided to malicious parties.

PPDM (Privacy Preserving Data Mining) algorithms [1] have been widely studied to protect users' privacy while keeping data utility. According to their architecture, they can be divided into the following two categories: centralized model and local model (or local privacy model) [13]. In the centralized model, there is a trusted database administrator, who can access to all users' personal data. When the administrator provides the data to a data analyst (who is possibly malicious),

***Corresponding Author: Takao Murakami:** National Institute of Advanced Industrial Science and Technology (AIST), E-mail: takao-murakami@aist.go.jp

Hideitsu Hino: University of Tsukuba / RIKEN Center for AIP, E-mail: hinohide@cs.tsukuba.ac.jp

Jun Sakuma: University of Tsukuba / RIKEN Center for AIP, E-mail: jun@cs.tsukuba.ac.jp

he/she replaces user IDs with pseudonyms and obfuscates the data (e.g., by adding noise, generalization, adding dummy data). The obfuscation algorithm is designed so that the original data are not recovered from the obfuscated data (while enabling data analysis). In this model, however, the original data of all users may be leaked from the database to a malicious adversary by illegal access or internal fraud. This issue is crucial in recent years, in which the number of data leakage incidents is increasing. For example, the number of U.S. data breaches was increased by 40% in 2016 [10].

The local model is designed to be more secure against such a data leakage. In this model, users do not assume a trusted party that can access to their personal data. The users obfuscate their personal data (e.g., add noise to the data, generalize the data) by themselves, and send them to a data collector (or data analyst). The data collector does not observe the original data, but observes only the obfuscated data. Based on the obfuscated data, he/she infers the statistics (e.g., histogram, heavy hitters [39]) of the original data or provides a service (e.g., provides POI (point of interest) information nearby the noisy location [4]) to the users.

In this paper, we focus on the problem of discrete distribution estimation in the local model, in which data are represented as discrete values and a data collector estimates a distribution (i.e., multinomial distribution) of the original personal data from the obfuscated data. Examples of the personal data include locations, power-consumption data, responses in a survey, and radiation levels [43] (continuous data such as locations and power-consumption data are discretized into bins). We refer to this problem as *LPDE (Locally Private Distribution Estimation)* for short. LPDE is composed of the following two phases: (1) obfuscating the personal data (i.e., obfuscation phase) and (2) estimating the discrete distribution (i.e., distribution estimation phase).

More formally, suppose that N users $u_1, \cdots, u_N$ send their obfuscated data to the data collector. Let $\mathcal{X} = \{x_1, \cdots, x_K\}$ (alphabet of size K) be a space of personal data, $\mathcal{Y} = \{y_1, \cdots, y_L\}$ (alphabet of size L) be a space of obfuscated data, and X_n (resp. Y_n) be a random variable representing personal data (resp. obfuscated data) of u_n. Each user u_n ($1 \leq n \leq N$) obfuscates his/her personal data X_n via an obfuscation mechanism G, and sends the obfuscated data Y_n to a data collector (in this paper, we assume that each user sends only one sample; we discuss the case where each user sends multiple samples in Section 5.4). The obfuscation mechanism G can be represented as a $K \times L$ transition matrix, whose (i, j)-th element G_{ij} is a transition probability from x_i to y_j: $G_{ij} = \Pr(Y = y_j | X = x_i)$ ($1 \leq i \leq K$, $1 \leq j \leq L$). We assume that the original data $X_1, \cdots, X_N$ are independent and identically distributed (i.i.d.) with distribution $\mathbf{p} = (p_1, \cdots, p_K)^T$ (we regard $\mathbf{p}$ as a K-dimensional vector), where $p_i = \Pr(X = x_i)$ ($1 \leq i \leq K$). The data collector estimates $\mathbf{p}$ from the obfuscated data $Y_1, \cdots, Y_N$ with the help of the knowledge of G. Fig. 1 shows the overview of LPDE.

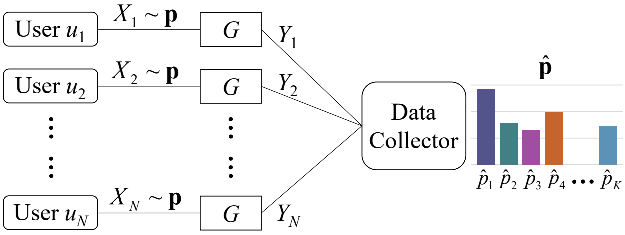


Fig. 1. Overview of LPDE. $\hat{\mathbf{p}} = (\hat{p}_1, \cdots, \hat{p}_K)^T$ is an estimate of $\mathbf{p} = (p_1, \cdots, p_K)^T$.

A number of studies have recently been made on LPDE (e.g., [3, 17, 25, 30, 31, 37, 40, 44]), mainly from the perspective of privacy metrics, obfuscation mechanisms, and statistical inference methods. A representative privacy metric in the local model is *LDP (Local Differential Privacy)* [11]. LDP is a variant of differential privacy [12] in the local model, and provides privacy guarantees against adversaries with arbitrary background knowledge. Roughly speaking, LDP guarantees that the adversary cannot infer the value of the original data from the obfuscated data with a certain degree of confidence. The amount of information leakage can be bounded by a parameter ϵ called *privacy budget*, and a large amount of noise is added to the personal data when ϵ is small (i.e., high privacy regime). The K-RR (K-ary Randomized Response) [30, 31] and the RAPPOR (Randomized Aggregatable Privacy-Preserving Ordinal Response) [17] are examples of obfuscation mechanisms satisfying LDP.

As for statistical inference methods, the matrix inversion method [3, 25, 30] and the EM (Expectation-Maximization) reconstruction method [2, 3] (which is also called the iterative Bayesian technique [3]) are well-known approaches. The matrix inversion method is used when the output alphabet size L is equal to the input alphabet size K. It infers a discrete distribution $\mathbf{p}$ by multiplying an empirical distribution of the obfuscated data $Y_1, \cdots, Y_N$ by an inverse matrix G^{-1}. A major drawback of this method is that some elements in an estimate $\hat{\mathbf{p}}$ of $\mathbf{p}$ can be negative, especially when the number of users N is small. The EM reconstruction method is a more sophisticated way to infer $\mathbf{p}$. This method is based on the EM algorithm [22], and iteratively estimates $\mathbf{p}$ until convergence. The feature of this method is that the final estimate (converged value) $\hat{\mathbf{p}}$ is equal to the ML (Maximum Likelihood) estimate in the probability simplex (i.e., $\hat{p}_1, \cdots, \hat{p}_K \geq 0$, $\sum_{i=1}^K \hat{p}_i = 1$), irrespective of the number of users N (see Section 3.2 for more details). It is reported in [3] that the EM reconstruction method significantly outperforms the matrix inversion method.

One of the main challenges in LPDE is how to accurately estimate the true discrete distribution $\mathbf{p}$ when the number of users N is small, or when users add a large amount of noise to their personal data (i.e., high privacy regime in which the privacy budget ϵ is small). It is well known that the ML estimate (i.e., the estimate by the EM reconstruction method) converges to the true value as the sample size goes to infinity. However, the number of users N can be small in practice due to various reasons. For example, N can be small when a data collector is a small-scale enterprise. Although Google implemented the RAPPOR in the Chrome browser and collected a dozen million samples [17], small-scale enterprises may not be able to collect such large samples. For another example, N can be small when a data collector estimates a distribution for people at a certain place or at a certain time. Furthermore, even if N is large, the *effective sample size* can be small in the high privacy regime. Duchi *et al.* [11] showed that for $\epsilon \in [0, \frac{22}{35}]$, the effective sample size required to achieve a certain level of estimation errors (minimax rate) is $4\epsilon^2 N$ (i.e., it decreases quadratically with decrease in ϵ). Thus, it is very challenging to accurately estimate $\mathbf{p}$ when ϵ is small (e.g., $\epsilon = 0.01$ or 0.1).

1.1 Our Contributions

The goal of this paper is to accurately estimate a discrete distribution $\mathbf{p}$ when the number of users N or the privacy budget ϵ is small. To achieve this goal, we focus on a *distribution estimation phase* in LPDE. Specifically, we focus on the EM reconstruction method [2, 3], which is a state-of-the-art statistical inference method, and propose a method to correct its estimation error (i.e., difference between the estimate and the true value) to significantly improve the estimation accuracy.

Our contributions are summarized as follows:

- We propose a method to correct an estimation error of the EM reconstruction method based on the theory of Rilstone *et al.* [41]. A major problem here is that the estimated error value may not be accurate when N or ϵ is small. To address this issue, the proposed method multiplies the estimated error value by a *weight parameter* α, and automatically determines an optimal value of α. We also prove that the proposed method reduces the MSE (Mean Square Error) under some assumptions (Section 4).
- We evaluate the proposed method using three large-scale real datasets: the People-flow dataset [45], the Foursquare dataset [50], and the US Census (1990) dataset [33]. The first and second datasets contain location data, while the third dataset contains census data. The results show that the proposed method significantly outperforms the existing inference methods (i.e., the matrix inversion method [3, 25, 30] and the EM reconstruction method [2, 3]) in all of the datasets when N or ϵ is small (Section 5).

More specifically about the second contribution, we consider the fact that the required privacy level can vary from user to user in practice [29]. Conservative users would require high level of privacy, whereas liberal users would not mind low level of privacy. Some liberal users might not mind it even if some of their personal data (e.g., visited sightseeing places, innocuous responses in a survey) are made public, and consequently they might not use an obfuscation method.

Taking this into account, we evaluated the proposed method (denoted by **Proposal**) in a scenario where the privacy budget ϵ is different from user to user (those who do not use an obfuscation method can be modeled by setting ϵ to ∞). We also generalize the matrix inversion method (denoted by **MatInv**) and the EM reconstruction method (denoted by **EM**) to such a scenario (see Section 3.2), and evaluate these methods for comparison. The results show that **Proposal** outperforms **MatInv** and **EM** when the total number of users N is small (e.g., $N \approx 1000$) or when N is large but most users adopt a small value of ϵ (e.g., $\epsilon = 0.1$).

In addition to the above-mentioned methods, we also evaluate two methods that estimate $\mathbf{p}$ *without the knowledge of the obfuscation mechanism.* The first one estimates $\mathbf{p}$ as an empirical distribution of the obfuscated data $Y_1, \cdots, Y_N$ (denoted by **ObfDat**). The second one always estimates $\mathbf{p}$ as a uniform distribution: $\hat{\mathbf{p}} = (\frac{1}{K}, \cdots, \frac{1}{K})^T$ (denoted by **Uniform**). In our experiments, we show that all of **Proposal**, **MatInv**, and **EM** performed worse than these two methods when both N and ϵ (adopted by most users) are extremely small. This is because the variance of the estimate $\hat{\mathbf{p}}$ is very small in **ObfDat** and **Uniform** (in particular, the variance of $\hat{\mathbf{p}}$ is always 0 in **Uniform**). On the other hand, the variance of $\hat{\mathbf{p}}$ is very large in **Proposal**, **Mat-Inv**, and **EM** when both N and ϵ are extremely small. We show this limitation, and provide a guideline for when to use the proposed method by thoroughly evaluating the effects of N and ϵ on the estimation accuracy.

2 Preliminaries

2.1 Notations

Table 1 shows the basic notations used in the rest of this paper. It should be noted that we denote an obfuscation mechanism of user u_n by $G^{(n)} \in [0,1]^{K \times L}$ (instead of G). This is because the privacy budget ϵ can be different from user to user, as described in Section 1.1.

Each user u_n ($1 \leq n \leq N$) obfuscates his/her personal data X_n via the obfuscation mechanism $G^{(n)}$, and sends the obfuscated data Y_n to a data collector (we discuss the case where each user sends multiple samples in Section 5.4). The (i,j)-th element of $G^{(n)}$ is a transition probability from x_i to y_j: $G^{(n)}_{i,j} = \Pr(Y_n = y_j | X_n = x_i)$ ($1 \leq i \leq K$, $1 \leq j \leq L$). The original data $X_1, \cdots, X_N$ are independent and identically distributed (i.i.d.) with distribution $\mathbf{p} = (p_1, \cdots, p_K)^T \in [0,1]^K$. We denote a set of all original data $\{X_1, \cdots, X_N\}$ and a set of all obfuscated data $\{Y_1, \cdots, Y_N\}$ by $\mathbf{X}$ and $\mathbf{Y}$, respectively. The data collector estimates $\mathbf{p}$ from $\mathbf{Y}$ with the help of the knowledge of $G^{(1)}, \cdots, G^{(N)}$. We denote the estimate of $\mathbf{p}$ by $\hat{\mathbf{p}} = (\hat{p}_1, \cdots, \hat{p}_K)^T \in \mathbb{R}^K$.

We also denote the probability simplex by $\mathcal{C}$; i.e., $\mathcal{C} := \{\mathbf{p} | p_1, \cdots, p_K \geq 0, \sum_{i=1}^K p_i = 1\}$. Moreover, we define the following K-dimensional vector:

$$\mathbf{g}_n := (G^{(n)}_{1,\pi(Y_n)}, \cdots, G^{(n)}_{K,\pi(Y_n)})^T \in [0,1]^K, \tag{1}$$

where $\pi(Y_n)$ is an index of the alphabet in $\mathcal{Y}$ that is equal to Y_n (i.e., if $Y_n = y_j$, then $\pi(Y_n) = j$). $\mathbf{g}_n$ will be used in Sections 3.2 and 4.

2.2 Utility Metrics

In this paper, we use the MSE (Mean Square Error) and the JS (Jensen-Shannon) divergence [34] as utility metrics to measure the difference between the true distribution $\mathbf{p}$ and its estimate $\hat{\mathbf{p}}$.

Table 1. Basic notations used in this paper ($1 \leq n \leq N$).

Symbol	Description
$\mathbb{R}$	Set of real numbers.
N	Number of users.
u_n	n-th user.
$\mathcal{X} = \{x_1, \cdots, x_K\}$	Space of original data.
$\mathcal{Y} = \{y_1, \cdots, y_L\}$	Space of obfuscated data.
X_n	n-th user's original data.
Y_n	n-th user's obfuscated data.
$G^{(n)} \in [0,1]^{K \times L}$	n-th user's obfuscation mechanism.
$\mathbf{X} = \{X_1, \cdots, X_N\}$	Set of all original data.
$\mathbf{Y} = \{Y_1, \cdots, Y_N\}$	Set of all obfuscated data.
$\mathbf{p} = (p_1, \cdots, p_K)^T$	Distribution of the original data.
$\hat{\mathbf{p}} = (\hat{p}_1, \cdots, \hat{p}_K)^T$	Estimate of $\mathbf{p}$.
$\mathcal{C}$	$:= \{\mathbf{p} \| p_1, \cdots, p_K \geq 0, \sum_{i=1}^K p_i = 1\}$ (i.e., probability simplex).
$\pi(Y_n)$	index of the alphabet in $\mathcal{Y}$ equal to Y_n (i.e., if $Y_n = y_j$, then $\pi(Y_n) = j$).
$\mathbf{g}_n$	$:= (G^{(n)}_{1,\pi(Y_n)}, \cdots, G^{(n)}_{K,\pi(Y_n)})^T$.

MSE (Mean Square Error). The MSE is one of the most popular metrics to measure the quality of an estimator. Given $\mathbf{p}$ and $\hat{\mathbf{p}}$, the squared error (i.e., l_2 loss) is computed as follows:

$$||\hat{\mathbf{p}} - \mathbf{p}||_2^2 = \sum_{i=1}^{K} (p_i - \hat{p}_i)^2. \tag{2}$$

It should be noted that the original data $\mathbf{X}$ are randomly generated from $\mathbf{p}$, and the obfuscated data $\mathbf{Y}$ are randomly generated from $\mathbf{X}$ using the obfuscation mechanisms $G^{(1)}, \cdots, G^{(N)}$. Since $\hat{\mathbf{p}}$ is computed from $\mathbf{Y}$, the squared error can be changed depending on $\mathbf{Y}$.

The MSE is an expectation of the squared error:

$$\mathbf{MSE} := \mathbb{E}[||\hat{\mathbf{p}} - \mathbf{p}||_2^2] \tag{3}$$

(or the sample mean of the squared errors over multiple realizations of $\mathbf{Y}$). Based on the bias-variance decomposition [22], it can be decomposed as follows:

$$\mathbf{MSE} = ||\mathbf{Bias}(\mathbf{p}, \hat{\mathbf{p}})||_2^2 + \mathbf{Var}(\hat{\mathbf{p}}), \tag{4}$$

where

$$\mathbf{Bias}(\mathbf{p}, \hat{\mathbf{p}}) := \mathbb{E}[\hat{\mathbf{p}}] - \mathbf{p} \tag{5}$$

$$\mathbf{Var}(\hat{\mathbf{p}}) := \mathbb{E}[||\hat{\mathbf{p}} - \mathbb{E}[\hat{\mathbf{p}}]||_2^2]. \tag{6}$$

Fig. 2 shows the relationship between the MSE, bias, and variance. Note that the bias and variance are highly dependent on inference methods. For example, when we always estimate $\mathbf{p}$ as $\mathbf{p} = (\frac{1}{K}, \cdots, \frac{1}{K})^T$ (i.e., **Uniform**), the variance is always 0 (i.e., $\mathbf{Var}(\hat{\mathbf{p}}) = 0$). When we use the EM reconstruction method, the bias is much smaller (as shown in our experiments) and the variance is larger than 0.

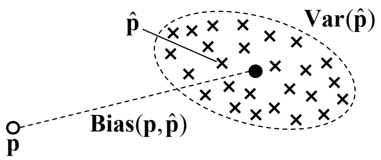

$\hat{\mathbf{p}}$
$\mathbf{Var}(\hat{\mathbf{p}})$
$\mathbf{Bias}(\mathbf{p}, \hat{\mathbf{p}})$
$\mathbf{p}$

Fig. 2. Relationship between the MSE, bias, and variance. The closed circle (•) is the expectation of $\hat{\mathbf{p}}$ (i.e., $\mathbb{E}[\hat{\mathbf{p}}]$). The MSE can be decomposed into $\mathbf{Bias}(\mathbf{p}, \hat{\mathbf{p}})$ and $\mathbf{Var}(\hat{\mathbf{p}})$.

JS (Jensen-Shannon) divergence. Since the JS divergence [34] is related to the KL (Kullback-Leibler) divergence [9], we first explain the KL divergence. The KL divergence between $\mathbf{p}$ and $\hat{\mathbf{p}}$ is given by

$$D(\mathbf{p} \parallel \hat{\mathbf{p}}) := \sum_{i=1}^{K} p_i \log \frac{p_i}{\hat{p}_i}. \tag{7}$$

Although the KL divergence can also measure the difference between $\mathbf{p}$ and $\hat{\mathbf{p}}$, it becomes infinite when $p_i > 0$ and $\hat{p}_i = 0$ for some $i \in \{1, \cdots, K\}$.

To avoid this problem, we use the JS divergence. The JS divergence between $\mathbf{p}$ and $\hat{\mathbf{p}}$ is given by

$$\mathbf{JSD}(\mathbf{p} \parallel \hat{\mathbf{p}}) := \frac{1}{2} D(\mathbf{p} \parallel \mathbf{m}) + \frac{1}{2} D(\hat{\mathbf{p}} \parallel \mathbf{m}), \tag{8}$$

where $\mathbf{m} = (\mathbf{p} + \hat{\mathbf{p}})/2$. In contrast to the KL divergence, the JS divergence is always finite.

It can be seen from (2) and (3) that the errors in small values of $\mathbf{p}$ has a small impact on the MSE. On the other hand, the errors in small values of $\mathbf{p}$ can make a large impact on the JS divergence (due to the logarithm in (7)). Thus, the MSE is suitable for evaluating the errors in large values of $\mathbf{p}$, whereas the JS divergence is suitable for evaluating the errors in small values of $\mathbf{p}$.

2.3 Privacy Metrics

In this paper, we use LDP (Local Differential Privacy) [11] as a privacy metric in the local model:

Definition 1 (ϵ-LDP [11])**.** *An obfuscation mechanism $G^{(n)}$ provides ϵ-LDP (ϵ-local differential privacy) if for all $i, j \in \{1, \cdots, K\}$ and all $S \subset \mathcal{Y}$, we have*

$$G^{(n)}(S|X_n = x_i) \le e^{\epsilon} G^{(n)}(S|X_n = x_j), \tag{9}$$

where $G^{(n)}(S|X_n = x_i) = \Pr(Y_n \in S|X_n = x_i)$ and $\epsilon \ge 0$.

LDP guarantees that an adversary who obtains obfuscated data Y_n cannot infer, for any pair of x_i and x_j, whether $X_n = x_i$ or $X_n = x_j$ with a certain degree of confidence. When ϵ is close to 0, it seems for the adversary that all of $x_1, \cdots, x_K \in \mathcal{X}$ are almost equally likely. Therefore, LDP with a small value of ϵ guarantees strong privacy protection.

3 Related Work

We now review the previous work related to ours. We describe obfuscation mechanisms and statistical inference methods in Sections 3.1 and 3.2, respectively.

3.1 Obfuscation Mechanisms

Obfuscation mechanisms that satisfy LDP have been widely studied in recent years. The K-RR (K-ary Randomized Response) [30, 31] is one of the simplest mechanisms satisfying LDP. This mechanism is a generalization of Warner's binary randomized response [49] to K-ary alphabets.

In the K-RR, the output range is identical to the input domain; i.e., $\mathcal{X} = \mathcal{Y}$. Let $G^{KRR} \in [0,1]^{K \times K}$ be the K-RR. The (i,j)-th element of G^{KRR} is given by

$$G_{i,j}^{KRR} = \begin{cases} \frac{e^{\epsilon}}{K-1+e^{\epsilon}} & (\text{if } j = i) \\ \frac{1}{K-1+e^{\epsilon}} & (\text{if } j \ne i) \end{cases} \tag{10}$$

$(1 \le i, j \le K)$. The K-RR satisfies ϵ-LDP.

Another example is the RAPPOR (Randomized Aggregatable Privacy-Preserving Ordinal Response) [17], which is used in the Chrome browser. The simplest configuration of the RAPPOR is called the *basic one-time RAPPOR*. The basic one-time RAPPOR is a mechanism with the output alphabet $\mathcal{Y} = \{0,1\}^K$ of size 2^K. Specifically, it deterministically maps x_i $(1 \le i \le K)$ onto $e_i \in \{0,1\}^K$, where e_i is the i-th standard basis vector. Then it flips each bit of e_i with probability $1/(1+e^{\epsilon/2})$. This mechanism also satisfies ϵ-LDP.

Kairouz *et al.* [30, 31] theoretically analyzed the K-RR and the basic one-time RAPPOR. In [30], they proved that under l_1 and l_2 losses, the K-RR and the basic one-time RAPPOR are order optimal in the low privacy regime (e.g., $\epsilon = \ln(K)$) and high privacy regime (e.g., $\epsilon = 0.01, 0.1$), respectively. In [31], they also proved that the K-RR is optimal in that it maximizes the mutual information $I(X;Y)$ between the original data X and the obfuscated data Y in the low privacy regime.

Other promising obfuscation mechanisms have also been studied in the literature. Kairouz *et al.* [30] proposed the O-RR, which is an extension of the K-RR using hash functions and cohorts. They showed that the performance of the O-RR meets or exceeds that of K-RR and the basic one-time RAPPOR in both low and high privacy regimes. Sei *et al.* [44] proposed an extension of the K-RR that not only randomizes the data but also adds multiple dummy samples. They showed that it outperforms the K-RR for $\epsilon \in [0.1, 1]$ using several artificial datasets.

In this paper, we use the K-RR as an obfuscation mechanism satisfying LDP due to the following reasons: (1) it is simple and widely used; (2) the output alphabet size L is small (not 2^K but K); (3) it can provide the optimal data utility in the low privacy regime [30, 31].

3.2 Statistical Inference Methods

The data collector computes an estimate $\hat{\mathbf{p}} = (\hat{p}_1, \cdots, \hat{p}_K)^T$ of the distribution $\mathbf{p}$ based on obfuscated data $\mathbf{Y} = \{Y_1, \cdots, Y_N\}$. The matrix inversion method [3, 25, 30] and the EM reconstruction method [2, 3] are existing methods to compute $\hat{\mathbf{p}}$ from $\mathbf{Y}$. Although both of them assume that all users $u_1, \cdots, u_N$ use the same obfuscation mechanism G $(= G^{(1)} = \cdots = G^{(N)})$, we generalize these methods to the case where $G^{(1)}, \cdots, G^{(N)}$ can be different, as we describe in detail below.

Matrix Inversion Method. Assume that the output alphabet size L is equal to the input alphabet size K, and that all users use the same obfuscation mechanism G $(= G^{(1)} = \cdots = G^{(N)})$. Let $\mathbf{q} = (q_1, \cdots, q_K)^T \in [0, 1]^K$ be a distribution of obfuscated data, which is given by

$$\mathbf{q}^T = \mathbf{p}^T G. \tag{11}$$

Let further $\hat{\mathbf{q}} = (\hat{q}_1, \cdots, \hat{q}_K)^T \in [0, 1]^K$ be an empirical distribution of $\mathbf{Y}$. The matrix inversion method computes $\hat{\mathbf{p}}$ by multiplying $\hat{\mathbf{q}}$ by an inverse matrix G^{-1}

$$\hat{\mathbf{p}}^T = \hat{\mathbf{q}}^T G^{-1}. \tag{12}$$

As the number of users N goes to infinity, the empirical distribution $\hat{\mathbf{q}}$ converges to the true distribution $\mathbf{q}$. Therefore, $\hat{\mathbf{p}}$ also converges to the true distribution $\mathbf{p}$.

We generalize the matrix inversion method to the case where there are multiple obfuscation mechanisms $G_1, \cdots, G_M$ $(M \ll N)$ and each user chooses one of the mechanisms to obfuscate his/her data (e.g., each user chooses one mechanism out of G_1, G_2, and G_3, each of which is corresponding to the high, middle, and low privacy regime, respectively). Let N_m $(1 \leq m \leq M)$ be the number of users who use G_m $(N = \sum_{m=1}^{M} N_m)$, and $\hat{\mathbf{q}}_m \in [0, 1]^K$ be an empirical distribution of the obfuscated data generated using G_m. Then, $\hat{\mathbf{p}}$ can be computed as follows:

$$\hat{\mathbf{p}}^T = \frac{1}{M} \sum_{n=1}^{M} \hat{\mathbf{q}}_m^T G_m^{-1}. \tag{13}$$

As $N_1, \cdots, N_M$ go to infinity, $\hat{\mathbf{p}}$ converges to $\mathbf{p}$ (in the same way as the original matrix inversion method).

However, when the number of users N $(= \sum_{m=1}^{M} N_m)$ is small, many elements in $\hat{\mathbf{p}}$ can be negative. Kairouz *et al.* [30] considered two methods to constrain $\hat{\mathbf{p}}$ to the probability simplex $\mathcal{C}$. The first method is a *normalized decoder*, which truncates the negative elements of $\hat{\mathbf{p}}$ to 0 and renormalizes $\hat{\mathbf{p}}$ so that the sum is 1. The second method is a *projected decoder*, which projects $\hat{\mathbf{p}}$ onto the probability simplex $\mathcal{C}$ so that the Euclidean distance between the two points is minimized (using the algorithm in [48]). We evaluate both methods in Section 5.

EM reconstruction Method. The EM reconstruction method is a more sophisticated method to infer $\mathbf{p}$. It regards $\mathbf{X}$ as a latent variable (or hidden variable), and infers $\mathbf{p}$ from $\mathbf{Y}$ using the EM (Expectation-Maximization) algorithm [22], which guarantees that the log-likelihood function $L_{\mathbf{Y}}(\mathbf{p}) := \log \Pr(\mathbf{Y}|\mathbf{p})$ is increased at each iteration (EM cycle). Although this method assumes that mechanisms $G^{(1)}, \cdots, G^{(N)}$ are the same [2, 3], we describe this method in a general case where $G^{(1)}, \cdots, G^{(N)}$ can be different.

Specifically, the following algorithm is equivalent to the EM algorithm (we can show this equivalence in the same way as [2]; we omit the proof for lack of space):

Algorithm 1 (EM reconstruction Method):

1. Initialize $\hat{\mathbf{p}} = (\hat{p}_1, \cdots, \hat{p}_K)^T \in \mathcal{C}$ (e.g., the empirical distribution of $\mathbf{Y}$ can be used: $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{q}}$ [3]).
2. Compute $\hat{\mathbf{p}}^{(new)} = (\hat{p}_1^{(new)}, \cdots, \hat{p}_K^{(new)})^T$ as follows:

$$\hat{p}_k^{(new)} = \frac{1}{N} \sum_{n=1}^{N} \frac{\hat{p}_k G_{k,\pi(Y_n)}^{(n)}}{\sum_{l=1}^{K} \hat{p}_l G_{l,\pi(Y_n)}^{(n)}}. \tag{14}$$

Repeat the update by (14) until convergence.

The feature of the EM reconstruction method is that the final estimate (converged value) $\hat{\mathbf{p}}$ is equal to the ML (Maximum Likelihood) estimate in the probability simplex $\mathcal{C}$, irrespective of the number of users N.

This can be explained as follows. The ML estimate in the probability simplex $\mathcal{C}$ maximizes the log-likelihood function $L_{\mathbf{Y}}(\mathbf{p})$ ($= \log \Pr(\mathbf{Y}|\mathbf{p})$) over $\mathcal{C}$. Since all of the obfuscated data $Y_1, \cdots, Y_N$ are independent, $L_{\mathbf{Y}}(\mathbf{p})$ can be written, using $L_n(\mathbf{p}) := \log \Pr(Y_n|\mathbf{p})$, as follows:

$$L_{\mathbf{Y}}(\mathbf{p}) = \sum_{n=1}^{N} L_n(\mathbf{p}) = \sum_{n=1}^{N} \log \mathbf{p}^T \mathbf{g}_n \tag{15}$$

(note that $L_n(\mathbf{p}) = \log \Pr(Y_n|\mathbf{p}) = \log \sum_{k=1}^{K} p_k G_{k,\pi(Y_n)}^{(n)}$ $= \log \mathbf{p}^T \mathbf{g}_n$). $\log \mathbf{p}^T \mathbf{g}_n$ is strictly concave in $\mathbf{p}$, and the sum of strictly concave functions is strictly concave. Thus, $L_{\mathbf{Y}}(\mathbf{p})$ in (15) is strictly concave, and has a unique global maximum over $\mathcal{C}$. It follows from (14) that the estimate of the EM reconstruction method is always in $\mathcal{C}$. In addition, each EM cycle in the EM algorithm is guaranteed to increase $L_{\mathbf{Y}}(\mathbf{p})$ [22]. Therefore, the estimate of the EM reconstruction method converges to the ML estimate in $\mathcal{C}$, whose log-likelihood is the global maximum over $\mathcal{C}$.

Note that *this property holds irrespective of the number of users N*. When N is small, many elements of $\hat{\mathbf{p}}$ can be negative in the matrix inversion method. On the other hand, the elements of $\hat{\mathbf{p}}$ are always non-negative in the EM reconstruction method, since it is equal to the ML estimate in $\mathcal{C}$. Thus, the EM reconstruction method can significantly outperform the matrix inversion method [3]. We also show this in Section 5.

4 Estimation Error Correction of the EM Reconstruction Method

The EM reconstruction method is a state-of-the-art statistical inference method, whose estimate is equal to the ML estimate in $\mathcal{C}$, irrespective of the number of users N. However, even this method may not accurately estimate the distribution $\mathbf{p}$ when N or ϵ is small, since the estimation error increases with decrease in N and ϵ.

To address this issue, we propose a method to reduce an estimation error (i.e., difference between the estimate $\hat{\mathbf{p}}$ and the true value $\mathbf{p}$) of the EM reconstruction method. Here we briefly describe its outline. We first formalize the expectation of the estimation error $\mathbb{E}[\hat{\mathbf{p}}] - \mathbf{p}$ (i.e., bias) in the EM reconstruction method up to order $O(N^{-1})$, which is denoted by $\mathbf{a}_{-1} \in \mathbb{R}^K$, based on the theory of Rilstone *et al.* [41]. We then replace the expectation $\mathbb{E}$ in $\mathbf{a}_{-1}$ with the empirical mean over N samples $Y_1, \cdots, Y_N$. We denote the result value by $\hat{\mathbf{a}}_{-1}$. It is important to note here that the estimate $\hat{\mathbf{p}}$ is also computed based on the N samples $Y_1, \cdots, Y_N$. Since both $\hat{\mathbf{a}}_{-1}$ and $\hat{\mathbf{p}}$ are computed based on the N samples $Y_1, \cdots, Y_N$, *we can regard $\hat{\mathbf{a}}_{-1}$ as a rough approximation of $\hat{\mathbf{p}} - \mathbf{p}$ (i.e., estimation error vector), whereas $\mathbf{a}_{-1}$ approximates* $\mathbb{E}[\hat{\mathbf{p}}] - \mathbf{p}$. We also prove that, under some assumptions, the MSE of the EM reconstruction method is reduced by subtracting $\hat{\mathbf{a}}_{-1}$ from $\hat{\mathbf{p}}$ (**Proposition 1** in Section 4.4). Note that this correction method was used to reduce the bias of the estimate [5, 41]. However, we prove a more general result that applying this correction can lead to a reduction in the MSE.

The proposed method computes $\hat{\mathbf{a}}_{-1}$ as an estimate of $\hat{\mathbf{p}} - \mathbf{p}$, and subtracts it from $\hat{\mathbf{p}}$. However, $\hat{\mathbf{a}}_{-1}$ may not be accurately computed when N or ϵ is small. Thus, the proposed method multiplies $\hat{\mathbf{a}}_{-1}$ by a weight parameter α and automatically determines an optimal value of α.

We first describe the theory of Rilstone *et al.* in Section 4.1. We then describe the proposed algorithm in Section 4.2, and describe how to determine an optimal value of α in Section 4.3. We finally provide a theoretical analysis of the MSE in Section 4.4.

4.1 Theory of Rilstone et al.

Given a set of N samples $\mathbf{Y} = \{Y_1, \cdots, Y_N\}$, Rilstone *et al.* [41] considered an estimate $\hat{\mathbf{p}} \in \mathbb{R}^K$ that is written as a solution to the following estimation equation:

$$\sum_{n=1}^{N} \mathbf{s}_n(\hat{\mathbf{p}}) = \mathbf{0}, \tag{16}$$

where $\mathbf{s}_n : \mathbb{R}^K \to \mathbb{R}^K$ is a function that takes $\hat{\mathbf{p}}$ as input and outputs a K-dimensional vector based on the n-th sample Y_n. The expectation of $\mathbf{s}_n$ is $\mathbf{0}$ only at the true value $\mathbf{p}$: $\mathbb{E}[\mathbf{s}_n(\mathbf{p})] = \mathbf{0}$. The class of estimators characterized by (16) include the ML estimator, generalized method of moments (GMM), and least squares (LS).

Rilstone *et al.* [41] showed that $\mathbb{E}[\hat{\mathbf{p}}] - \mathbf{p}$ (i.e., bias) of the estimate that satisfies (16) can be written as follows:

$$\mathbb{E}[\hat{\mathbf{p}}] - \mathbf{p} = \mathbf{a}_{-1} + O(N^{-3/2}). \tag{17}$$

$\mathbf{a}_{-1} \in \mathbb{R}^K$ is a dominant bias term of order $O(N^{-1})$, which is called the second-order bias[1], and is given by

$$\mathbf{a}_{-1} = \frac{1}{N}\mathbf{Q}\left\{\mathbb{E}[\mathbf{V}_n\mathbf{Q}\mathbf{s}_n] - \frac{1}{2}\mathbb{E}[\nabla^2\mathbf{s}_n]\mathbb{E}[\mathbf{Q}\mathbf{s}_n \otimes \mathbf{Q}\mathbf{s}_n]\right\}, \tag{18}$$

[1] Note that the first-order bias, which is a bias term of order $O(N^{-1/2})$, is given by $\mathbb{E}[-\mathbf{Q}\mathbf{s}_n(\mathbf{p})]$ [41]. However, it is zero since $\mathbb{E}[\mathbf{s}_n(\mathbf{p})] = \mathbf{0}$ (i.e., $\mathbb{E}[-\mathbf{Q}\mathbf{s}_n(\mathbf{p})] = -\mathbf{Q}\mathbb{E}[\mathbf{s}_n(\mathbf{p})] = \mathbf{0}$).

where $\mathbf{s}_n$ is a shorthand for $\mathbf{s}_n(\mathbf{p})$, ∇ is a vector differential operator (i.e., $\nabla = \frac{\partial}{\partial \mathbf{p}} = (\frac{\partial}{\partial p_1}, \cdots, \frac{\partial}{\partial p_K})$), $\otimes$ is a tensor product, and

$$\mathbf{Q} = \mathbb{E}[\nabla \mathbf{s}_n(\mathbf{p})]^{-1} \in \mathbb{R}^{K \times K} \tag{19}$$

$$\mathbf{V}_n = \nabla \mathbf{s}_n(\mathbf{p}) - \mathbb{E}[\nabla \mathbf{s}_n(\mathbf{p})] \in \mathbb{R}^{K \times K}. \tag{20}$$

4.2 Proposed Algorithm

We now describe the proposed method. We first exploit the fact that the estimate of the EM reconstruction method is equal to the ML estimate (as described in Section 3.2), and apply the theory of Rilstone *et al.* to the EM reconstruction method.

Specifically, we exploit the fact that *maximizing the log-likelihood function* $L_{\mathbf{Y}}(\mathbf{p})$ $(= \sum_{n=1}^{N} L_n(\hat{\mathbf{p}}))$ *in (15) is equivalent to solving the following equation:*

$$\sum_{n=1}^{N} \nabla L_n(\hat{\mathbf{p}}) = \mathbf{0} \tag{21}$$

(i.e., the gradient of the log-likelihood function ∇L_n is $\mathbf{0}$). By regarding ∇L_n in (21) as $\mathbf{s}_n$ in (16), we can apply the theory of Rilstone *et al.* to the EM reconstruction method.

Since $L_n(\mathbf{p}) = \log \mathbf{p}^T \mathbf{g}_n$ (see (15)), $\mathbf{s}_n$ $(= \nabla L_n)$, $\nabla \mathbf{s}_n$, and $\nabla^2 \mathbf{s}_n$ are written as follows:

$$\mathbf{s}_n(\mathbf{p}) = \frac{1}{\mathbf{p}^T \mathbf{g}_n} \mathbf{g}_n \in \mathbb{R}^K \tag{22}$$

$$\nabla \mathbf{s}_n(\mathbf{p}) = -\frac{1}{(\mathbf{p}^T \mathbf{g}_n)^2} \mathbf{g}_n \mathbf{g}_n^T \in \mathbb{R}^{K \times K} \tag{23}$$

$$\nabla^2 \mathbf{s}_n(\mathbf{p}) = \frac{1}{(\mathbf{p}^T \mathbf{g}_n)^3} \mathbf{g}_n \otimes \mathbf{g}_n \otimes \mathbf{g}_n \in \mathbb{R}^{K \times K \times K} \tag{24}$$

By using (23), $\mathbf{Q}$ in (19) and $\mathbf{V}_n$ in (20) are written as follows:

$$\mathbf{Q} = \mathbb{E}[\nabla \mathbf{s}_n(\mathbf{p})]^{-1} = -\mathbb{E}\left[\frac{1}{(\mathbf{p}^T \mathbf{g}_n)^2} \mathbf{g}_n \mathbf{g}_n^T\right]^{-1} \tag{25}$$

$$\mathbf{V}_n = \nabla \mathbf{s}_n(\mathbf{p}) - \mathbb{E}[\nabla \mathbf{s}_n(\mathbf{p})] \tag{26}$$

$$= \mathbb{E}\left[\frac{1}{(\mathbf{p}^T \mathbf{g}_n)^2} \mathbf{g}_n \mathbf{g}_n^T\right] - \frac{1}{(\mathbf{p}^T \mathbf{g}_n)^2} \mathbf{g}_n \mathbf{g}_n^T. \tag{27}$$

Note that $-\mathbb{E}[\nabla \mathbf{s}_n(\mathbf{p})]$ $(= -\mathbb{E}[\nabla^2 L_n(\mathbf{p})])$ is the Fisher information matrix [9]. Since $-\mathbf{Q}$ is the inverse of this matrix, it provides the lower bound of covariance matrix (i.e., Crámer-Rao inequality [9]).

The second-order bias $\mathbf{a}_{-1}$ in the EM reconstruction method is given by substituting (22), (24), (25), and (27) into (18). We replace the expectation $\mathbb{E}$ in $\mathbf{a}_{-1}$ with the empirical mean over N samples $Y_1, \cdots, Y_N$. Since we do not know the true value $\mathbf{p}$, we also replace $\mathbf{p}$ with the estimate $\hat{\mathbf{p}}$ (i.e., plug-in estimate), as done in [5, 41]. The result value, denoted by $\hat{\mathbf{a}}_{-1}$, can be written as follows:

$$\begin{aligned}\hat{\mathbf{a}}_{-1} =& \frac{1}{N} \hat{\mathbf{Q}} \left\{ \frac{1}{N} \sum_{n=1}^{N} (\hat{\mathbf{V}}_n \hat{\mathbf{Q}} \mathbf{s}_n(\hat{\mathbf{p}})) \right. \\ & \left. - \frac{1}{2N^2} \sum_{n=1}^{N} (\nabla^2 \mathbf{s}_n(\hat{\mathbf{p}})) \sum_{n=1}^{N} (\hat{\mathbf{Q}} \mathbf{s}_n(\hat{\mathbf{p}}) \otimes \hat{\mathbf{Q}} \mathbf{s}_n(\hat{\mathbf{p}})) \right\},\end{aligned} \tag{28}$$

where

$$\hat{\mathbf{Q}} = -\hat{\mathbf{S}}^{-1} \in \mathbb{R}^{K \times K} \tag{29}$$

$$\hat{\mathbf{V}}_n = \hat{\mathbf{S}} - \frac{1}{(\hat{\mathbf{p}}^T \mathbf{g}_n)^2} \mathbf{g}_n \mathbf{g}_n^T \in \mathbb{R}^{K \times K} \tag{30}$$

$$\hat{\mathbf{S}} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(\hat{\mathbf{p}}^T \mathbf{g}_n)^2} \mathbf{g}_n \mathbf{g}_n^T \in \mathbb{R}^{K \times K} \tag{31}$$

($\mathbb{E}$ in $\mathbf{a}_{-1}$, $\mathbf{Q}$, and $\mathbf{V}_n$ is now replaced by the empirical mean in $\hat{\mathbf{a}}_{-1}$, $\hat{\mathbf{Q}}$, and $\hat{\mathbf{V}}_n$, respectively). We emphasize again that both $\hat{\mathbf{a}}_{-1}$ and $\hat{\mathbf{p}}$ are computed based on the N samples $Y_1, \cdots, Y_N$. Therefore, we use $\hat{\mathbf{a}}_{-1}$ as an estimate of $\hat{\mathbf{p}} - \mathbf{p}$ (whereas $\mathbf{a}_{-1}$ approximates $\mathbb{E}[\hat{\mathbf{p}}] - \mathbf{p}$).

However, there is a major problem in computing $\hat{\mathbf{Q}}$ in (29). When N or ϵ is small, the rank of $\hat{\mathbf{S}}$ in (31) becomes much smaller than K (note that as ϵ goes to 0, $\mathbf{g}_n \mathbf{g}_n^T$ in (29) approaches to $(1/K^2)\mathbf{J}_K$, where $\mathbf{J}_K$ is the $K \times K$ all-ones matrix whose rank is 1). When $\hat{\mathbf{S}}$ is highly rank deficient, we cannot compute $\hat{\mathbf{Q}}$ in (29), which is the inverse of $-\hat{\mathbf{S}}$.

To avoid this problem, we add a small positive value λ (> 0) to the diagonal elements of $\hat{\mathbf{S}}$ to make $\hat{\mathbf{S}}$ a full rank matrix. Such a regularization is known as the Tikhonov regularization [20]. That is, we compute $\hat{\mathbf{Q}}$ as follows:

$$\hat{\mathbf{Q}} = -(\hat{\mathbf{S}} + \lambda \mathbf{I}_K)^{-1} \in \mathbb{R}^{K \times K}, \tag{32}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix.

We compute $\hat{\mathbf{a}}_{-1}$ from $\hat{\mathbf{p}}$ and $\mathbf{g}_n$ $(1 \leq n \leq N)$ by substituting (30), (31), and (32) into (28) (we can compute $\hat{\mathbf{a}}_{-1}$ with time complexity $O(NK^2)$; for details, see Appendix B). It should be noted, however, that $\hat{\mathbf{Q}}$ in (32) may not be accurately computed, since the matrix $\lambda \mathbf{I}_K$ is added to $\hat{\mathbf{S}}$ in (32). As a consequence, $\hat{\mathbf{a}}_{-1}$ may also not be accurately computed. To address this issue, we multiply $\hat{\mathbf{a}}_{-1}$ by a *weight parameter* α (> 0). In other words, we do not trust $\hat{\mathbf{a}}_{-1}$ itself, but trust the direction of $\hat{\mathbf{a}}_{-1}$. We denote the corrected estimate in the proposed method by $\tilde{\mathbf{p}} = (\tilde{p}_1, \cdots, \tilde{p}_K)^T \in \mathbb{R}^K$; i.e., $\tilde{\mathbf{p}} = \hat{\mathbf{p}} - \alpha \hat{\mathbf{a}}_{-1}$. We describe how to determine an optimal value of α in Section 4.3.

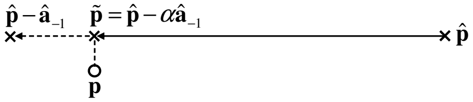

$\hat{\mathbf{p}}-\hat{\mathbf{a}}_{-1}$
$\tilde{\mathbf{p}}=\hat{\mathbf{p}}-\alpha\hat{\mathbf{a}}_{-1}$
$\hat{\mathbf{p}}$
$\mathbf{p}$

Fig. 3. Corrected estimate $\tilde{\mathbf{p}}$ in the proposed method. We multiply $\hat{\mathbf{a}}_{-1}$ by α, and subtract it from an estimate $\hat{\mathbf{p}}$ of the EM reconstruction method.

In summary, the proposed algorithm is as follows:

Algorithm 2 (Proposed Algorithm):

1. Compute an estimate $\hat{\mathbf{p}}$ in the EM reconstruction Method (using **Algorithm 1**).
2. Compute $\hat{\mathbf{a}}_{-1}$ in (28) from $\hat{\mathbf{p}}$ and $\mathbf{g}_n$ $(1 \leq n \leq N)$ by (30)-(32) (see Appendix B for details).
3. Correct the estimation error of $\hat{\mathbf{p}}$ as follows:

$$\tilde{\mathbf{p}} = \hat{\mathbf{p}} - \alpha\hat{\mathbf{a}}_{-1} \tag{33}$$

(see Section 4.3 for optimization of α).

Fig. 3 shows the corrected estimate $\tilde{\mathbf{p}}$ in the proposed method. Note that the corrected estimate $\tilde{\mathbf{p}}$ in (33) can deviate from the probability simplex $\mathcal{C}$ slightly (this deviation is very small if $\alpha\hat{\mathbf{a}}_{-1}$ are appropriate). Thus, after the step 3 in **Algorithm 2**, we use the normalized decoder, which is described in Section 3.2, to constrain $\tilde{\mathbf{p}}$ to the probability simplex $\mathcal{C}$.

4.3 Optimization of the Weight Parameter

We now describe how to determine an optimal value of the weight parameter α (> 0) in the proposed method. The proposed method determines α by running a *simulation* that simulates a real distribution estimation task using some distribution $\mathbf{p}' = (p'_1, \cdots, p'_K)^T$ (we use a symbol with "′" to represent a variable in the simulation; we describe what to use as $\mathbf{p}'$ later in detail).

Fig. 4 shows the overview of determining α based on the simulation. We first generate personal data $\mathbf{X}' = \{X'_1, \cdots, X'_N\}$ from the distribution $\mathbf{p}'$ $(X'_1, \cdots, X'_N$ are i.i.d. with $\mathbf{p}')$. Then we obfuscate the n-th personal data X'_n using the n-th obfuscation mechanism $G^{(n)}$ $(1 \leq n \leq N)$. Let $\mathbf{Y}' = \{Y'_1, \cdots, Y'_N\}$ be the obfuscated data generated from $\mathbf{X}'$. We compute a corrected estimate $\tilde{\mathbf{p}}' = (\tilde{p}'_1, \cdots, \tilde{p}'_K)^T$ in the proposed method from $\mathbf{Y}'$ by running **Algorithm 2** for various values of α (e.g., $\alpha \in \{10^{-10}, 10^{-9}, \cdots, 1\}$). We evaluate the estimation error of $\tilde{\mathbf{p}}'$ (e.g., squared error $||\tilde{\mathbf{p}}' - \mathbf{p}'||_2^2$) for each value of α, and select α that achieves the smallest error.

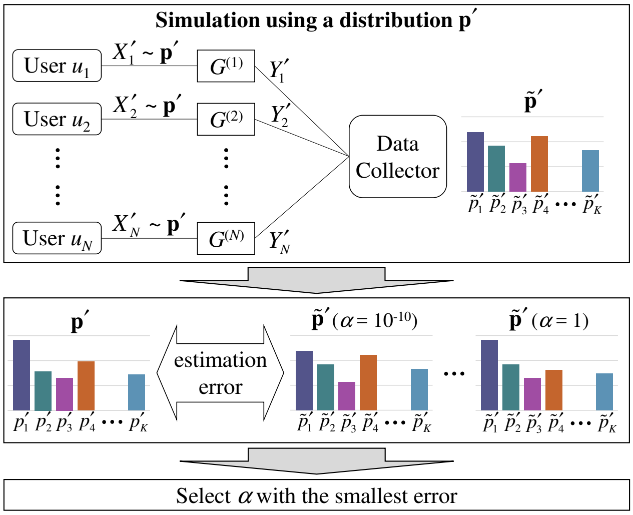

Simulation using a distribution $\mathbf{p}'$
User u_1
$X'_1 \sim \mathbf{p}'$
$G^{(1)}$
Y'_1
User u_2
$X'_2 \sim \mathbf{p}'$
$G^{(2)}$
Y'_2
User u_N
$X'_N \sim \mathbf{p}'$
$G^{(N)}$
Y'_N
Data Collector
$\tilde{\mathbf{p}}'$
$\tilde{p}'_1$ $\tilde{p}'_2$ $\tilde{p}'_3$ $\tilde{p}'_4$ ··· $\tilde{p}'_K$
$\mathbf{p}'$
p'_1 p'_2 p'_3 p'_4 ··· p'_K
estimation error
$\tilde{\mathbf{p}}'(\alpha = 10^{-10})$
$\tilde{p}'_1$ $\tilde{p}'_2$ $\tilde{p}'_3$ $\tilde{p}'_4$ ··· $\tilde{p}'_K$
···
$\tilde{\mathbf{p}}'(\alpha = 1)$
$\tilde{p}'_1$ $\tilde{p}'_2$ $\tilde{p}'_3$ $\tilde{p}'_4$ ··· $\tilde{p}'_K$
Select α with the smallest error

Fig. 4. Overview of determining the weight parameter α. We run a simulation using a distribution $\mathbf{p}'$, and compute a corrected estimate $\tilde{\mathbf{p}}'$ in the proposed method for various values of α (e.g., $\alpha \in \{10^{-10}, 10^{-9}, \cdots, 1\}$). Then we select α that achieves the smallest estimation error.

Note that after running the steps 1 and 2 in **Algorithm 2**, we can compute $\tilde{\mathbf{p}}'$ for each value of α by just running the step 3 in **Algorithm 2** (i.e., we run the steps 1 and 2 in **Algorithm 2** just for once, and then run the step 3 for many times). Therefore, the time complexity of determining α is $O(NK^2)$ (i.e., the same as **Algorithm 2**) in total.

We now describe what to use as $\mathbf{p}'$ in the simulation. When no prior information is available about the true distribution $\mathbf{p}$, we should use a uniform distribution as $\mathbf{p}'$. It should be noted, however, that the obfuscated data $\mathbf{Y}$ are available as prior information about $\mathbf{p}$. Taking this into account, we use the empirical distribution of $\mathbf{Y}$ as $\mathbf{p}'$. In our experiments, we also confirmed that the proposed method that determines α using the empirical distribution of $\mathbf{Y}$ provided the better performance than the proposed method that determines α using the uniform distribution.

Remark. It should be noted that although we start with the formalization of the second-order bias $\mathbf{a}_{-1}$ in the EM reconstruction method, the proposed method does not guarantee that the bias of the EM reconstruction method is reduced. In fact, the bias of the proposed method was larger than that of the EM reconstruction method in our experiments (as shown in Appendix C). We consider this is because we applied the Tikhonov regularization to compute $\hat{\mathbf{Q}}$ in (32). A regularization method is generally used to significantly reduce the variance by introducing a bias in the estimate. The Tikhonov regularization can also introduce a bias [28].

However, we emphasize that the goal of this paper is to accurately estimate $\mathbf{p}$ when N or ϵ is small, and that we utilize $\hat{\mathbf{a}}_{-1}$ as a means to reduce the estimation error. In our experiments, we show that it significantly reduces the variance (at the cost of increasing the bias), and therefore reduces the MSE and the JS divergence. We also prove that the proposed method reduces the MSE under some assumptions in Section 4.4.

It should also be noted that the weight parameter α can influence the estimation accuracy. In this paper, we generate artificial data from the empirical distribution of $\mathbf{Y}$ in an analogous way to the parametric bootstrap method [15], and chooses a hyper-parameter α (i.e., performs a kind of model selection) using the artificial data. It is also known that the bootstrap method is used for model selection [27]. However, we may be able to choose a better α by improving our method in several directions. For example, although we choose α that minimizes the squared error $||\tilde{\mathbf{p}}' - \mathbf{p}'||_2^2$ based on one simulation in our experiments, we may be able to choose a better α by running multiple simulations and using the MSE as a metric (at the cost of computational time necessary to optimize α). We may also be able to choose a better α by extending our method to the Bayesian framework in the same way as [19]. We leave such improvements as future work.

4.4 Theoretical Analysis

We provide a theoretical analysis of the MSE in the proposed method. Specifically, we show that the proposed method can reduce the second-order MSE, which is an MSE term of order $O(N^{-3/2})$, to zero under some assumptions.

According to the theory of Rilstone *et al.* [41], the MSE of the EM reconstruction method (denoted by $\mathbf{MSE_{EM}}$) can be written as follows[2]:

$$\mathbf{MSE_{EM}} = b_{-1} + b_{-3/2} + O(N^{-2}), \tag{34}$$

where

$$b_{-1} = \mathbb{E}\left[\mathbf{d}^T\mathbf{d}\right] \in \mathbb{R} \tag{35}$$

$$b_{-3/2} = -\mathbb{E}\left[\mathbf{d}^T\left\{2\mathbf{Q}\mathbf{V}_n\mathbf{d} - \mathbf{Q}\mathbb{E}[\nabla^2\mathbf{s}_n][\mathbf{d}\otimes\mathbf{d}]\right\}\right] \in \mathbb{R} \tag{36}$$

$$\mathbf{d} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{d}_n \in \mathbb{R}^K \tag{37}$$

$$\mathbf{d}_n = \mathbf{Q}\mathbf{s}_n \in \mathbb{R}^K. \tag{38}$$

b_{-1} is a term of order $O(N^{-1})$, and is called the first-order MSE. $b_{-3/2}$ is a term of order $O(N^{-3/2})$, and is called the second-order MSE.

We now consider the proposed method with the weight parameter $\alpha = 1$, which corrects the estimate $\hat{\mathbf{p}}$ of the EM reconstruction method by subtracting $\hat{\mathbf{a}}_{-1}$ in (28) from $\hat{\mathbf{p}}$:

$$\tilde{\mathbf{p}} = \hat{\mathbf{p}} - \hat{\mathbf{a}}_{-1} \tag{39}$$

Let $\mathbf{MSE_{Proposal}}$ be the MSE of this method. To simplify our theoretical analysis, we assume the following two assumptions: (i) $\hat{\mathbf{a}}_{-1}$ is evaluated at $\mathbf{p}$, (ii) $\mathbf{Q}$, $\mathbf{V}_n$, and $\mathbb{E}[\nabla^2\mathbf{s}_n]$ are perfectly estimated: $\mathbf{Q} = \hat{\mathbf{Q}}$, $\mathbf{V}_n = \hat{\mathbf{V}}_n$, and $\mathbb{E}[\nabla^2\mathbf{s}_n] = \frac{1}{N}\sum_{n=1}^{N}(\nabla^2\mathbf{s}_n)$. In this case, $\hat{\mathbf{a}}_{-1}$ in (28) can be written as follows:

$$\hat{\mathbf{a}}_{-1} = \frac{1}{N}\mathbf{Q}\left\{\frac{1}{N}\sum_{n=1}^{N}(\mathbf{V}_n\mathbf{Q}\mathbf{s}_n) - \frac{1}{2N}\mathbb{E}[\nabla^2\mathbf{s}_n]\sum_{n=1}^{N}(\mathbf{Q}\mathbf{s}_n\otimes\mathbf{Q}\mathbf{s}_n)\right\}. \tag{40}$$

It should be noted that although we make some ideal assumptions, we still replace the two expectation terms $\mathbb{E}$ in $\mathbf{a}_{-1}$ (see (18)) with the empirical mean over N samples $Y_1, \cdots, Y_N$. We prove that *the second-order MSE is reduced to zero by these replacements.*

Namely, we prove the following result:

Proposition 1.

$$\mathbf{MSE_{Proposal}} = b_{-1} + O(N^{-2}). \tag{41}$$

The proof is given in Appendix A. **Proposition 1** indicates that the estimation error can be reduced by subtracting $\hat{\mathbf{a}}_{-1}$ from $\hat{\mathbf{p}}$ (since $\hat{\mathbf{a}}_{-1}$ may not be accurately computed, we multiply $\hat{\mathbf{a}}_{-1}$ by α in practice). It should be noted, however, that the first-order MSE b_{-1} is not reduced in this case. This can be explained by the fact that **Proposal** provides almost the same performance as **EM** when N is large in our experiments. However, we emphasize that it is still beneficial to reduce $b_{-3/2}$ when N is small, since the term of order $O(N^{-3/2})$ is large in this case. In fact, **Proposal** significantly outperforms **EM** when N or ϵ is small, as shown in our experiments.

[2] Note that in [41], the MSE is represented in the form of a matrix (i.e., error covariance matrix). $\mathbf{MSE_{EM}}$ can be written as (34) by computing the trace of the MSE matrix in [41].

5 Experimental Evaluation

5.1 Experimental Set-up

We evaluated the proposed method by conducting experiments using three real datasets: the People-flow dataset [45], the Foursquare dataset [50], and the US Census (1990) dataset [33]. The first two datasets contain location data, while the third dataset contains census data. We used these datasets because they are large-scale datasets (we used the data of 303916, 251689, and 2458285 people in the People-flow, Foursquare, and US Census datasets, respectively). In the following, we describe these datasets in detail:

- **People-flow dataset:** The People-flow dataset (1998 Tokyo metropolitan area) [45] contains mobility traces (time-series location trails) of 722000 people in the Tokyo metropolitan area in 1998. In this paper, we used this dataset for estimating a geographic population distribution in the period of one day. To this end, we extracted mobility traces of 303916 people on the first of October, and used the first location sample for each user (we excluded the remaining 418084 people since they had no location samples on the first of October). We divided the Tokyo metropolitan area into 20×20 regions, excluded 119 regions in the sea, and used the remaining 281 land regions as input alphabets ($K = 281$).
- **Foursquare dataset:** The Foursquare dataset (global-scale check-in dataset) [50] was collected from April 2012 to September 2013. It contains location check-ins by 266909 people all over the world. Since many of these check-ins were located in 415 cities, we focused on these cities. We extracted 251689 people who had at least one check-in in these cities, and used the first location check-in for each user (we excluded the remaining 15220 people who had no check-ins in these cities). We used the 415 cities as input alphabets ($K = 415$).
- **US Census (1990) dataset:** The US Census (1990) dataset [33] was collected as part of US census in 1990. It contains responses from 2458285 people (each user provided one response), where each response has 68 attributes. We used the responses from all people, and used age, sex, income, and marital status as attributes. Each attribute has 8, 2, 5, and 5 category IDs depending on their value, as shown in Table 2. We regarded a sequence of these category IDs as a single category ID. Thus, the total number of category IDs is 400 ($= 8 \times 2 \times 5 \times 5$).

  We used these category IDs as input alphabets ($K = 400$).

Table 2. Attributes (age, sex, income, and marital status) and category IDs in the US Census (1990) dataset.

Attribute	Category ID (Value)
Age	**0 (0), 1 (1-12), 2 (13-19), 3 (20-29), 4 (30-39), 5 (40-49), 6 (50-64), or 7 (65-)**
Sex	**0 (male) or 1 (female)**
Income	**0 ($0), 1 ($1-$14999), 2 ($15000-$29999), 3 ($30000-$60000), or 4 ($60000-)**
Marital status	**0 (now married, except separated), 1 (widowed), 2 (divorced), 3 (separated), or 4 (never married)**

For each dataset, we used a frequency distribution of all people (303916, 251689, and 2458285 people in the People-flow, Foursquare, and US Census datasets, respectively) as $\mathbf{p}$ (i.e., distribution of the original data). We randomly selected N users from these people. Here we attempted 100 cases to randomly select N users, and ran, for each case, the following experiments.

We conducted experiments, in which each user u_n ($1 \leq n \leq N$) obfuscates his/her personal data X_n (i.e., region ID, city ID, or category ID) via the obfuscation mechanism $G^{(n)}$, and a data collector computes an estimate $\hat{\mathbf{p}}$ of the distribution $\mathbf{p}$ based on the obfuscated data $\mathbf{Y} = \{Y_1, \cdots, Y_N\}$. As an obfuscation mechanism $G^{(n)}$, we used the K-RR (i.e., G^{KRR} in (10)). As for the privacy budget ϵ, we considered four values: $\epsilon = 0.1$, 2, $\ln(K)$, and ∞, each of which is corresponding to the high, middle, low, and "no" privacy regime, respectively. We denote the number of users who set $\epsilon = 0.1$, 2, $\ln(K)$, and ∞ by N_1, N_2, N_3, and N_4, respectively (i.e., $N = \sum_{i=1}^{4} N_i$).

We set $\epsilon = 0.1$ and 2 in the high and middle privacy regime, respectively, since many studies used these values [24] and $\epsilon = 0.1$ offers reasonably strong privacy protection [32]. We set $\epsilon = \ln(K)$ in the low privacy regime, since a user sends different data (i.e., $X_n \neq Y_n$) with probability 50% even in this case. In other words, $\epsilon = \ln(K)$ can still provide plausible deniability. This value of ϵ was also used in [30]. When $\epsilon = \infty$, G^{KRR} in (10) is equivalent to the identity matrix $\mathbf{I}_K$. This means that those who do not use an obfuscation method can be modeled by setting ϵ to ∞, as described in Section 1.1.

However, many users might care about their privacy and prefer the high or middle privacy regime (i.e., $\epsilon = 0.1$ or 2). Taking this into account, we set N_3 and N_4 much smaller than N_1 and N_2. Specifically, we first set N_1, N_2, N_3, and N_4 so that $N_1 : N_2 : N_3 = 10 : 10 : 1$

and changed N_4 from 0 to N_3 (e.g., $(N_1, N_2, N_3) = (500, 500, 50)$ and $N_4 \in [0, 50]$). We then evaluated the performance in the case where we significantly increased only N_1 (i.e., most users select the high privacy regime in which $\epsilon = 0.1$). To more thoroughly evaluate the effects of N and ϵ on the performance, we also set $N_1 = N_4 = 0$ and evaluated the performance for various values of N_2 and N_3.

As a statistical inference method, we evaluated the following methods for comparison:

- **Uniform:** A method that always estimates $\mathbf{p}$ as a uniform distribution: $\hat{\mathbf{p}} = (\frac{1}{K}, \cdots, \frac{1}{K})^T$.
- **ObfDat:** A method that estimates $\mathbf{p}$ as an empirical distribution of the obfuscated data $Y_1, \cdots, Y_N$.
- **MatInv$_{\mathbf{norm}}$:** The matrix inversion method using the normalized decoder (described in Section 3.2).
- **MatInv$_{\mathbf{proj}}$:** The matrix inversion method using the projected decoder (described in Section 3.2).
- **EM:** The EM reconstruction method (described in Section 3.2).
- **Proposal:** The proposed method.

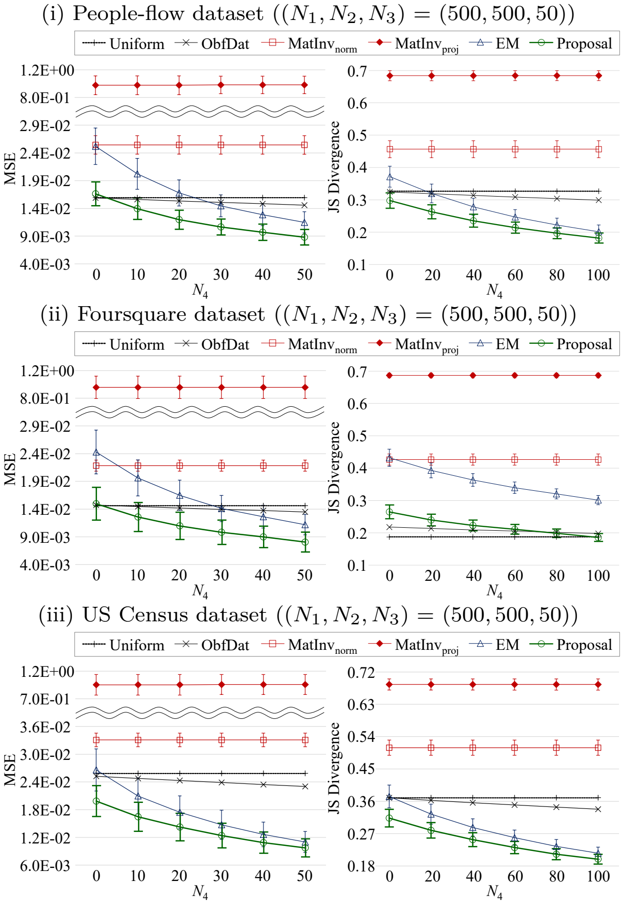
In **EM**, we used the empirical distribution of $\mathbf{Y}$ as an initial value of $\hat{\mathbf{p}}$ (i.e., $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{q}}$) in the same way as [3]. In **Proposal**, we set $\lambda = 10^{-3}$, and attempted various values for α: $\alpha \in \{c_1 \times 10^{c_2} | c_1 \in \{1, \cdots, 9\}, c_2 \in \{-1, \cdots, -10\}\}$. Then we selected α that achieved the smallest squared error (i.e., $||\tilde{\mathbf{p}}' - \mathbf{p}'||_2^2$).

After computing the estimate $\hat{\mathbf{p}}$ (or the corrected estimate $\tilde{\mathbf{p}}$ in **Proposal**), we evaluated the MSE and the JS divergence. Specifically, we computed the average of the squared error $||\hat{\mathbf{p}} - \mathbf{p}||_2^2$ (or $||\tilde{\mathbf{p}} - \mathbf{p}||_2^2$) over 100 runs (i.e., 100 cases to randomly select N users), and used it as the MSE. In other words, we computed the sample mean of 100 squared errors as the MSE. Similarly, we averaged the JS divergence over 100 runs. We also evaluated, for both the squared error and the JS divergence, the standard deviation over 100 runs.

5.2 Experimental Results

We first evaluated the MSE and the JS divergence in the case where $(N_1, N_2, N_3) = (500, 500, 50)$ and $N_4 \in [0, 50]$. Fig. 5 shows the results.

It can be seen that the MSE and the JS divergence of **MatInv$_{\mathbf{norm}}$** and **MatInv$_{\mathbf{proj}}$** are very large. This is because many elements in the estimate $\hat{\mathbf{p}}$ were negative, as described in Section 3.2. In particular, the performance of **MatInv$_{\mathbf{proj}}$** is much worse than that of the other inference methods. This is because the estimate $\hat{\mathbf{p}}$ that contained many negative elements was projected to a vertex of the probability simplex (i.e., each element in $\hat{\mathbf{p}}$ was either 0 or 1). On the other hand, **EM** outperforms **MatInv$_{\mathbf{norm}}$** and **MatInv$_{\mathbf{proj}}$** in most cases (in the same way as [3]), since the elements in $\hat{\mathbf{p}}$ are always nonnegative, as described in Section 3.2.


(i) People-flow dataset ($(N_1, N_2, N_3) = (500, 500, 50)$)

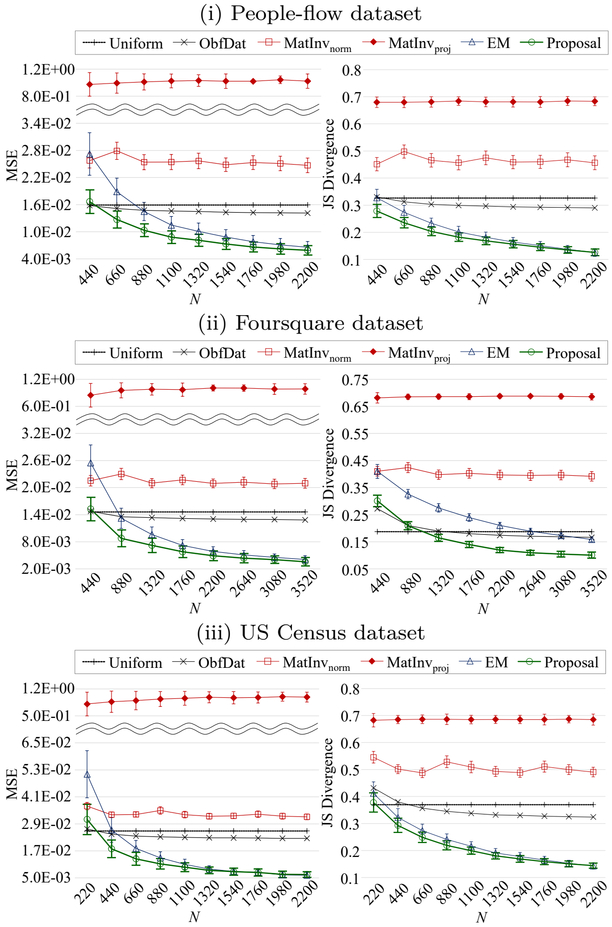
(ii) Foursquare dataset ($(N_1, N_2, N_3) = (500, 500, 50)$)

(iii) US Census dataset ($(N_1, N_2, N_3) = (500, 500, 50)$)

Fig. 5. The MSE and the JS divergence in the case where $(N_1, N_2, N_3) = (500, 500, 50)$ and $N_4 \in [0, 50]$. The error bars show standard deviations (we do not show error bars in **Uniform** and **ObfDat**, since the standard deviations are very small in these methods; in particular, the standard deviation is 0 in **Uniform**).

It can also be seen that **Proposal** outperforms **EM**, which shows that the estimation accuracy is improved by correcting the estimation error (although the error bars overlap in many cases, we show later that there is a very high correlation between 100 squared errors of **Proposal** and those of **EM**). In particular, **Proposal** significantly outperforms **EM** when N_4 is small. This is because the estimation error of **EM** is large in this case and is corrected by **Proposal**.

However, when N_4 is small, **Uniform** or **ObfDat** provides the best performance in some cases (e.g., the MSE in the People-flow dataset, the JS divergence in

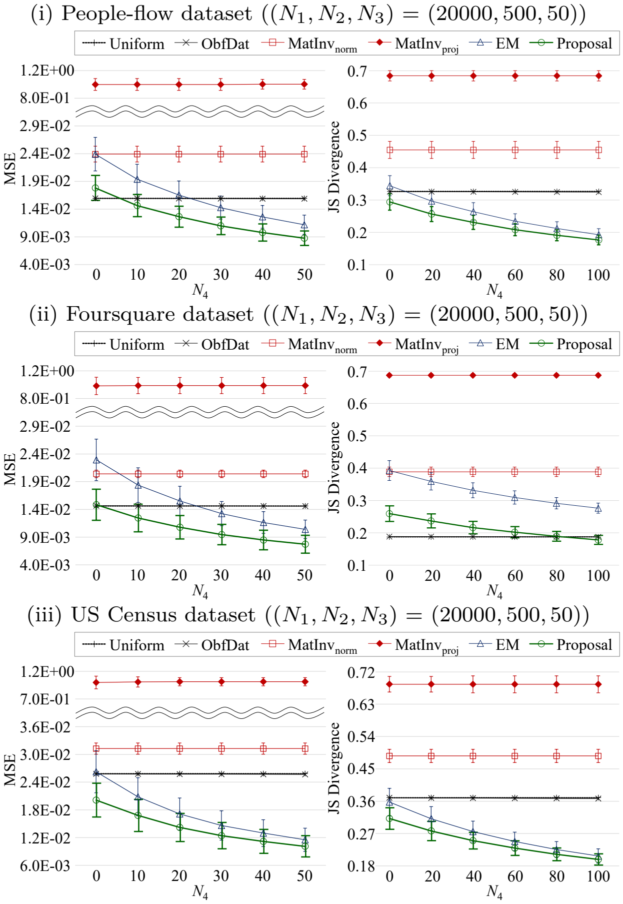


(i) People-flow dataset

(ii) Foursquare dataset

(iii) US Census dataset

Fig. 6. The MSE and the JS divergence in the case where N is changed while fixing $N_1 : N_2 : N_3 : N_4 = 10 : 10 : 1 : 1$. The error bars show standard deviations.



(i) People-flow dataset $((N_1, N_2, N_3) = (20000, 500, 50))$

(ii) Foursquare dataset $((N_1, N_2, N_3) = (20000, 500, 50))$

(iii) US Census dataset $((N_1, N_2, N_3) = (20000, 500, 50))$

Fig. 7. The MSE and the JS divergence in the case where $N_1 = 20000$. The error bars show standard deviations.

the Foursquare dataset). This is because the variance of the estimate $\hat{\mathbf{p}}$ was very small in **Uniform** and **ObfDat** (in Appendix C, we also show the results of the bias and variance for each method; see Appendix C for details). More specifically, the variance of $\hat{\mathbf{p}}$ was always 0 in **Uniform**, as described in Section 2.2. The variance of $\hat{\mathbf{p}}$ was also close to 0 in **ObfDat**, since only a small number of people sent the original data X_n (i.e., $X_n = Y_n$). In other words, the empirical distribution of the obfuscated data $Y_1, \cdots, Y_N$ was close to the uniform distribution. If the variance is larger than the bias of these methods, the MSE is also larger.

To investigate the relationship between the total number of users N $(= \sum_{i=1}^{4} N_i)$ and the performance, we changed N while fixing the ratio of $N_1 : N_2 : N_3 : N_4$. Specifically, we changed N while fixing $N_1 : N_2 : N_3 : N_4 = 10 : 10 : 1 : 1$. Fig. 6 shows the results. It can be seen that when N is very small, the MSE of **Uniform** or **ObfDat** is the smallest in all of the datasets. It can also be seen that the MSE and the JS divergence of **Proposal** rapidly decrease as N increases. **Proposal** provides the best performance with respect to both the MSE and the JS divergence when N is more than or equal to 660, 1320, and 440 in the People-flow, Foursquare, and US Census datasets, respectively.

We also evaluated the performance in the case where we significantly increased only N_1 (i.e., the number of users with $\epsilon = 0.1$). Specifically, we set $(N_1, N_2, N_3) = (20000, 500, 50)$ and $N_4 \in [0, 50]$. Fig. 7 shows the results. Fig. 7 is very similar to Fig. 5, and the MSE and the JS divergence are only slightly decreased (or not decreased) by increasing N_1. This is because when $\epsilon = 0.1$, the probability of sending the original data X_n (i.e., $X_n = Y_n$) was very small (0.39%, 0.27%, and 0.28%, in the People-flow, Foursquare, and US Census datasets, respectively). Since these users sent different data (i.e., $X_n \neq Y_n$) in most cases, they did not contribute much to the estimation accuracy. This is consistent with the

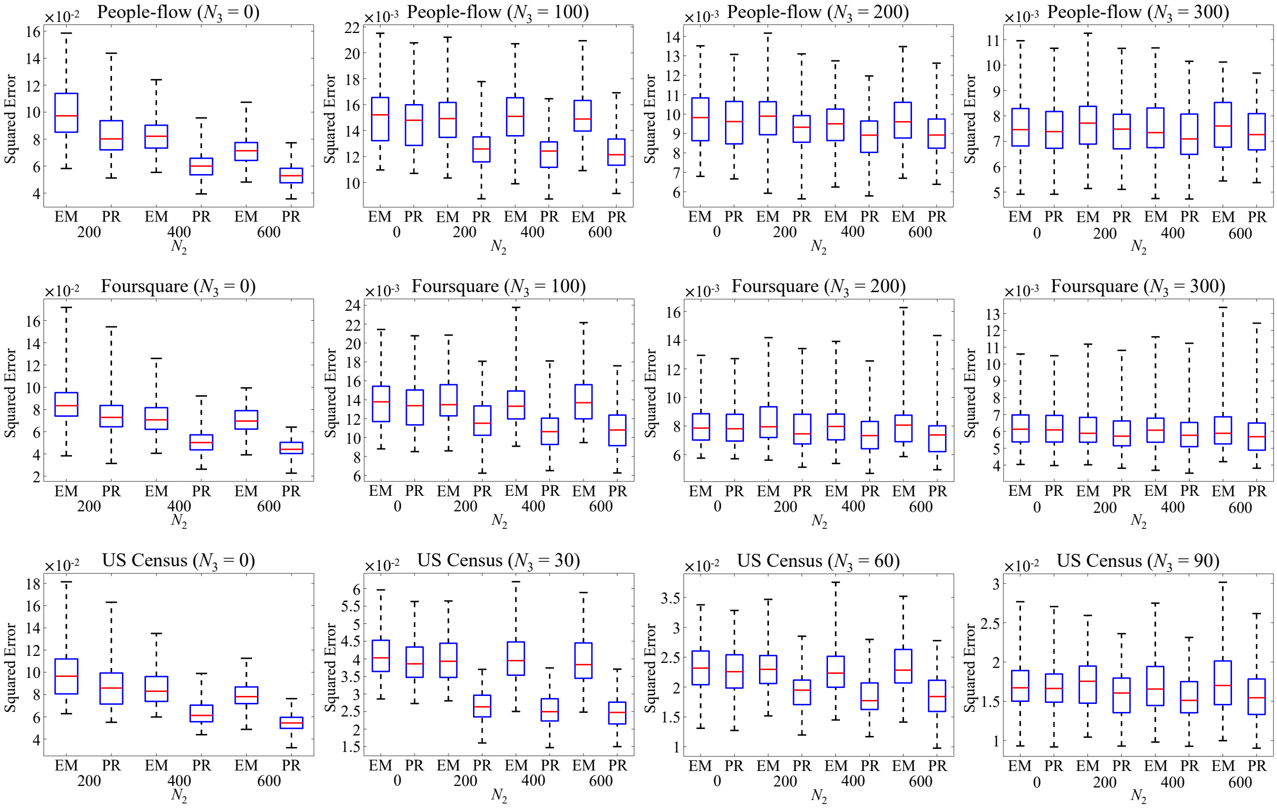


Fig. 8. Box plots of 100 squared errors for **EM** and **Proposal** in the case where N_2 and N_3 are changed and $N_1 = N_4 = 0$ (EM: **EM**, PR: **Proposal**). The ends of the whiskers represent the minimum and maximum values. The bottom and top of the box represent the first and third quartiles, respectively. The red band inside the box represents the median.

fact that the effective sample size decreases quadratically with decrease in ϵ [11].

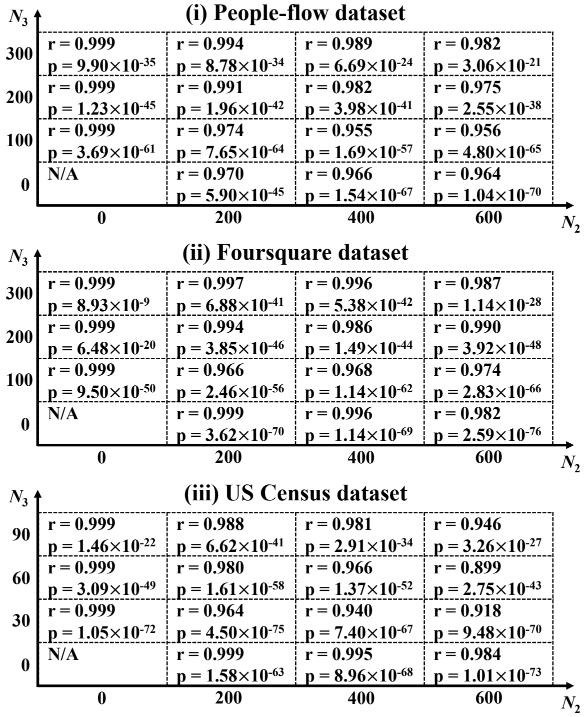
To more thoroughly evaluate the effects of N and ϵ on the performance, we finally set $N_1 = N_4 = 0$ and evaluated 100 squared errors and the MSE for various values of N_2 and N_3 (we do not show the JS divergence for lack of space). Fig. 8 shows the box plots of 100 squared errors for **EM** and **Proposal**. We also computed the correlation coefficient r between 100 squared errors of **Proposal** and those of **EM**, and the p-value p of the t-test for paired samples. Fig. 9 shows the results. In addition, Fig. 10 shows the best inference method, which achieves the smallest MSE among the six methods, for each case.

It can be seen from Fig. 8 that **Proposal** significantly outperforms **EM** when N_3 is small. This is because the estimation error of **EM** is large in this case and is corrected by **Proposal**. It can also be seen from Fig. 9 that the correlation coefficient r is very close to one in most cases. This means that when the MSE of **Proposal** is smaller than that of **EM**, **Proposal** outperforms **EM** in almost all of the 100 runs. Consequently, the difference between 100 squared errors of **Proposal** and those of **EM** is statistically significant ($p < 0.05$).

However, it can be seen from Fig. 10 that **Uniform** or **ObfDat** provides the best performance when N_2 and N_3 are very small (e.g., $N_3 = 0$). This is because the variance of the estimate $\hat{\mathbf{p}}$ was very small in **Uniform** and **ObfDat**, as previously explained. In addition, **Proposal** provides almost the same performance as **EM** when N_2 and N_3 are large. From Fig. 8 and 10, we conclude that **Proposal** is effective especially when N_3 is about 100 to 200 in the People-flow dataset, about 100 to 200 in the Foursquare dataset, and about 30 to 60 in the US Census dataset.

5.3 Visualization of Distributions

In Section 5.2, **Proposal** significantly outperformed **EM** in the case where the number of users N was small or when most users adopted a small value of ϵ. To explain how the proposed method corrected the estimation



(i) People-flow dataset

N_3 \ N_2	0	200	400	600

300	r = 0.999
	p = 9.90×10^{-35}
		r = 0.994
		p = 8.78×10^{-34}
			r = 0.989
			p = 6.69×10^{-24}
				r = 0.982
				p = 3.06×10^{-21}

200	r = 0.999
	p = 1.23×10^{-45}
		r = 0.991
		p = 1.96×10^{-42}
			r = 0.982
			p = 3.98×10^{-41}
				r = 0.975
				p = 2.55×10^{-38}

100	r = 0.999
	p = 3.69×10^{-61}
		r = 0.974
		p = 7.65×10^{-64}
			r = 0.955
			p = 1.69×10^{-57}
				r = 0.956
				p = 4.80×10^{-65}

0	N/A	r = 0.970
		p = 5.90×10^{-45}
			r = 0.966
			p = 1.54×10^{-67}
				r = 0.964
				p = 1.04×10^{-70}

(ii) Foursquare dataset

N_3 \ N_2	0	200	400	600

300	r = 0.999
	p = 8.93×10^{-9}
		r = 0.997
		p = 6.88×10^{-41}
			r = 0.996
			p = 5.38×10^{-42}
				r = 0.987
				p = 1.14×10^{-28}

200	r = 0.999
	p = 6.48×10^{-20}
		r = 0.994
		p = 3.85×10^{-46}
			r = 0.986
			p = 1.49×10^{-44}
				r = 0.990
				p = 3.92×10^{-48}

100	r = 0.999
	p = 9.50×10^{-50}
		r = 0.966
		p = 2.46×10^{-56}
			r = 0.968
			p = 1.14×10^{-62}
				r = 0.974
				p = 2.83×10^{-66}

0	N/A	r = 0.999
		p = 3.62×10^{-70}
			r = 0.996
			p = 1.14×10^{-69}
				r = 0.982
				p = 2.59×10^{-76}

(iii) US Census dataset

N_3 \ N_2	0	200	400	600

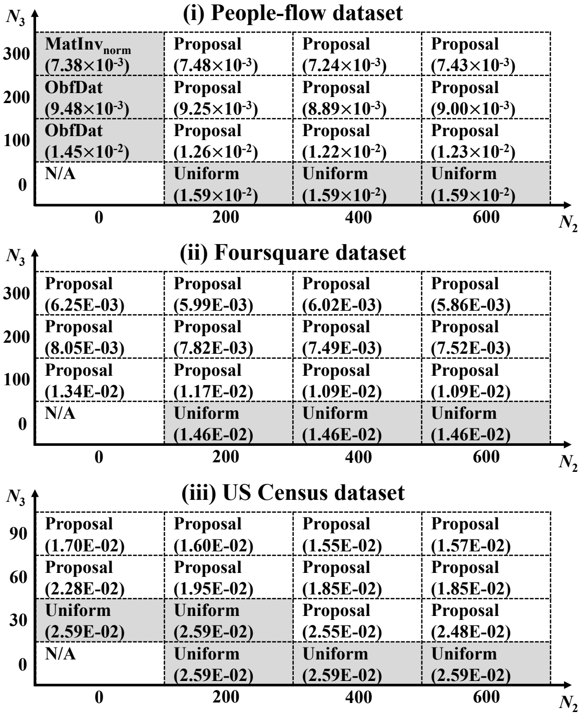
90	r = 0.999
	p = 1.46×10^{-22}
		r = 0.988
		p = 6.62×10^{-41}
			r = 0.981
			p = 2.91×10^{-34}
				r = 0.946
				p = 3.26×10^{-27}

60	r = 0.999
	p = 3.09×10^{-49}
		r = 0.980
		p = 1.61×10^{-58}
			r = 0.966
			p = 1.37×10^{-52}
				r = 0.899
				p = 2.75×10^{-43}

30	r = 0.999
	p = 1.05×10^{-72}
		r = 0.964
		p = 4.50×10^{-75}
			r = 0.940
			p = 7.40×10^{-67}
				r = 0.918
				p = 9.48×10^{-70}

0	N/A	r = 0.999
		p = 1.58×10^{-63}
			r = 0.995
			p = 8.96×10^{-68}
				r = 0.984
				p = 1.01×10^{-73}

Fig. 9. Correlation coefficients and p-values in the case where N_2 and N_3 are changed and $N_1 = N_4 = 0$. "r" represents the correlation coefficient between 100 squared errors of **Proposal** and those of **EM**. "p" represents the p-value of the t-test for paired samples.



(i) People-flow dataset

N_3 \ N_2	0	200	400	600
300	MatInv$_{\text{norm}}$ (7.38×10^{-3})	Proposal (7.48×10^{-3})	Proposal (7.24×10^{-3})	Proposal (7.43×10^{-3})
200	ObfDat (9.48×10^{-3})	Proposal (9.25×10^{-3})	Proposal (8.89×10^{-3})	Proposal (9.00×10^{-3})
100	ObfDat (1.45×10^{-2})	Proposal (1.26×10^{-2})	Proposal (1.22×10^{-2})	Proposal (1.23×10^{-2})
0	N/A	Uniform (1.59×10^{-2})	Uniform (1.59×10^{-2})	Uniform (1.59×10^{-2})

(ii) Foursquare dataset

N_3 \ N_2	0	200	400	600
300	Proposal (6.25E-03)	Proposal (5.99E-03)	Proposal (6.02E-03)	Proposal (5.86E-03)
200	Proposal (8.05E-03)	Proposal (7.82E-03)	Proposal (7.49E-03)	Proposal (7.52E-03)
100	Proposal (1.34E-02)	Proposal (1.17E-02)	Proposal (1.09E-02)	Proposal (1.09E-02)
0	N/A	Uniform (1.46E-02)	Uniform (1.46E-02)	Uniform (1.46E-02)

(iii) US Census dataset

N_3 \ N_2	0	200	400	600
90	Proposal (1.70E-02)	Proposal (1.60E-02)	Proposal (1.55E-02)	Proposal (1.57E-02)
60	Proposal (2.28E-02)	Proposal (1.95E-02)	Proposal (1.85E-02)	Proposal (1.85E-02)
30	Uniform (2.59E-02)	Uniform (2.59E-02)	Proposal (2.55E-02)	Proposal (2.48E-02)
0	N/A	Uniform (2.59E-02)	Uniform (2.59E-02)	Uniform (2.59E-02)

Fig. 10. Best inference method, which achieves the smallest MSE, in the case where N_2 and N_3 are changed and $N_1 = N_4 = 0$. The value inside of the parenthesis represents the MSE. A cell in which **Proposal** is not the best method is shaded in gray.

error in the EM reconstruction method, we visualize the true distribution $\mathbf{p}$, the estimate $\hat{\mathbf{p}}$ in **EM**, and the corrected estimate $\tilde{\mathbf{p}}$ in **Proposal** using the People-flow dataset.

Fig. 11 shows the true distribution $\mathbf{p}$. Fig. 12 shows $\hat{\mathbf{p}}$ and $\tilde{\mathbf{p}}$ in the special wards of Tokyo in the case where $(N_1, N_2, N_3, N_4) = (500, 500, 50, 20)$. In Fig. 12, we show 100 values of the estimates in each of the fifteen regions (from upper left to lower right) as a box plot.

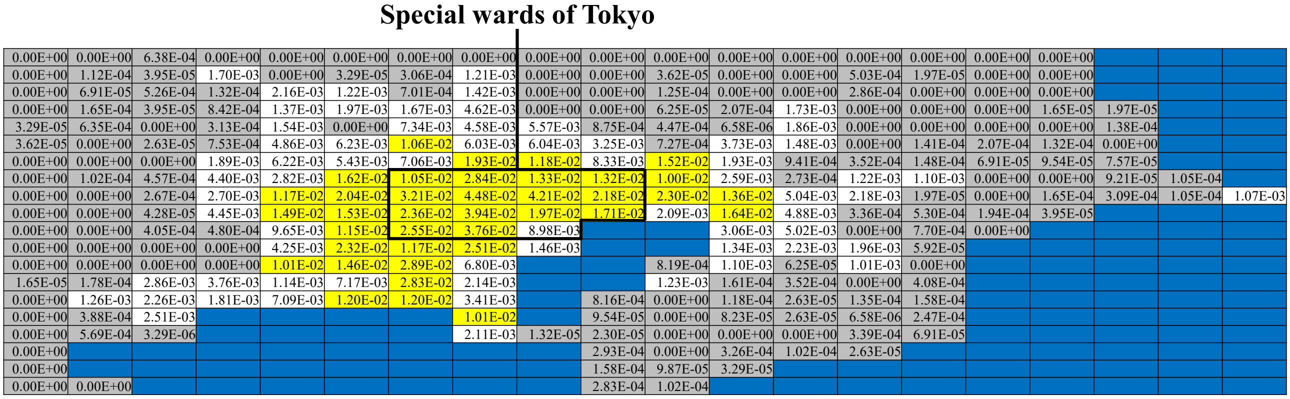
It can be seen that a variance of is smaller in **Proposal** in all of the regions. The maximum value of $\hat{p}_i$ in **EM** is much larger than the true value. **EM** also estimates $\hat{p}_i$ to be very close to zero in many cases (e.g., Regions #1, #3, #4, and #15). **Proposal** corrects these over/underestimated values. We consider this is the reason **Proposal** significantly outperformed **EM**.

5.4 Discussions on the Case of Multiple Samples Per User

In our experiments, we assumed that each user sends only one sample, and the data collector estimates the distribution $\mathbf{p}$. We finally discuss the extension of our results to case where each user sends multiple samples.

For example, suppose that user u_n obfuscates t samples $X_n^{(1)}, \cdots, X_n^{(t)}$ using the obfuscation mechanism $G^{(n)}$, which satisfies ϵ-LDP, and sends the obfuscated samples $Y_n^{(1)}, \cdots, Y_n^{(t)}$ to the data collector. Then, it follows from by the composition theorem [13] that the t samples are protected by (ϵt)-LDP. Therefore, if user u_n wants to protect t samples by ϵ-LDP, he/she can satisfy this privacy requirement by using, for each sample, the obfuscation mechanism satisfying (ϵ/t)-LDP.

When the number of samples t is large, each privacy budget ϵ/t can be very small and therefore a large amount of noise is added to each sample. A recent study [38] also showed that the data utility can be completely destroyed in the case of time-series location data. It should be noted, however, that the number of samples t can be different from user to user, and ϵ/t can be large for users whose t is small. For example, users who send only a small number of their locations (e.g., $t = 2$ or 3) may not have to add a large amount of noise to each location. If many users adopt a small value of ϵ/t and some users adopt a large value of ϵ/t, the proposed method would work well in the same way as in Fig. 7.


Special wards of Tokyo

0.00E+00	0.00E+00	6.38E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00			
0.00E+00	1.12E-04	3.95E-05	1.70E-03	0.00E+00	3.29E-05	3.06E-04	1.21E-03	0.00E+00	0.00E+00	3.62E-05	0.00E+00	0.00E+00	5.03E-04	1.97E-05	0.00E+00	0.00E+00			
0.00E+00	6.91E-05	5.26E-04	1.32E-04	2.16E-03	1.22E-03	7.01E-04	1.42E-03	0.00E+00	0.00E+00	1.25E-04	0.00E+00	0.00E+00	2.86E-04	0.00E+00	0.00E+00	0.00E+00			
0.00E+00	1.65E-04	3.95E-05	8.42E-04	1.37E-03	1.97E-03	1.67E-03	4.62E-03	0.00E+00	0.00E+00	6.25E-05	2.07E-04	1.73E-03	0.00E+00	0.00E+00	0.00E+00	1.65E-05	1.97E-05		
3.29E-05	6.35E-04	0.00E+00	3.13E-04	1.54E-03	0.00E+00	7.34E-03	4.58E-03	5.57E-03	8.75E-04	4.47E-04	6.58E-06	1.86E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.38E-04		
3.62E-05	0.00E+00	2.63E-05	7.53E-04	4.86E-03	6.23E-03	1.06E-02	6.03E-03	6.04E-03	3.25E-03	7.27E-04	3.73E-03	1.48E-03	0.00E+00	1.41E-04	2.07E-04	1.32E-04	0.00E+00		
0.00E+00	0.00E+00	0.00E+00	1.89E-03	6.22E-03	5.43E-03	7.06E-03	1.93E-02	1.18E-02	8.33E-03	1.52E-02	1.93E-03	9.41E-04	3.52E-04	1.48E-04	6.91E-05	9.54E-05	7.57E-05		
0.00E+00	1.02E-04	4.57E-04	4.40E-03	2.82E-03	1.62E-02	1.05E-02	2.84E-02	1.33E-02	1.32E-02	1.00E-02	2.59E-03	2.73E-04	1.22E-03	1.10E-03	0.00E+00	0.00E+00	9.21E-05	1.05E-04	
0.00E+00	0.00E+00	2.67E-04	2.70E-03	1.17E-02	2.04E-02	3.21E-02	4.48E-02	4.21E-02	2.18E-02	2.30E-02	1.36E-02	5.04E-03	2.18E-03	1.97E-05	0.00E+00	1.65E-04	3.09E-04	1.05E-04	1.07E-03
0.00E+00	0.00E+00	4.28E-05	4.45E-03	1.49E-02	1.53E-02	2.36E-02	3.94E-02	1.97E-02	1.71E-02	2.09E-03	1.64E-02	4.88E-03	3.36E-04	5.30E-04	1.94E-04	3.95E-05			
0.00E+00	0.00E+00	4.05E-04	4.80E-04	9.65E-03	1.15E-02	2.55E-02	3.76E-02	8.98E-03			3.06E-03	5.02E-03	0.00E+00	7.70E-04	0.00E+00				
0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.25E-03	2.32E-02	1.17E-02	2.51E-02	1.46E-03			1.34E-03	2.23E-03	1.96E-03	5.92E-05					
0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.01E-02	1.46E-02	2.89E-02	6.80E-03			8.19E-04	1.10E-03	6.25E-05	1.01E-03	0.00E+00					
1.65E-05	1.78E-04	2.86E-03	3.76E-03	1.14E-03	7.17E-03	2.83E-02	2.14E-03			1.23E-03	1.61E-04	3.52E-04	0.00E+00	4.08E-04					
0.00E+00	1.26E-03	2.26E-03	1.81E-03	7.09E-03	1.20E-02	1.20E-02	3.41E-03		8.16E-04	0.00E+00	1.18E-04	2.63E-05	1.35E-04	1.58E-04					
0.00E+00	3.88E-04	2.51E-03					1.01E-02		9.54E-05	0.00E+00	8.23E-05	2.63E-05	6.58E-06	2.47E-04					
0.00E+00	5.69E-04	3.29E-06					2.11E-03	1.32E-05	2.30E-05	0.00E+00	0.00E+00	0.00E+00	3.39E-04	6.91E-05					
0.00E+00									2.93E-04	0.00E+00	3.26E-04	1.02E-04	2.63E-05						
0.00E+00									1.58E-04	9.87E-05	3.29E-05								
0.00E+00	0.00E+00								2.83E-04	1.02E-04									

Fig. 11. Visualization of the true distribution $\mathbf{p}$ in the People-flow dataset. the true probability p_i is shown in the corresponding land region (sea regions are marked in blue). The special wards of Tokyo are marked by a solid black frame. The regions whose probabilities are lower than 0.1% (resp. higher than 1%) are marked in gray (resp. yellow).

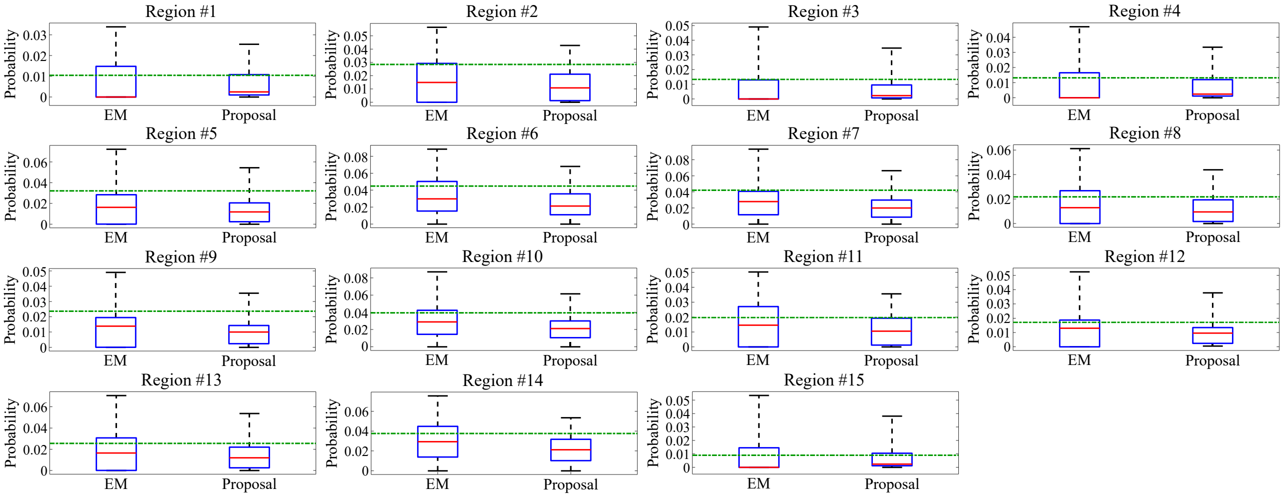


Fig. 12. Visualization of the estimate $\hat{\mathbf{p}}$ in **EM** and the corrected estimate $\tilde{\mathbf{p}}$ in **Proposal** in the special wards of Tokyo ($(N_1, N_2, N_3, N_4) = (500, 500, 50, 20)$). 100 values of the estimates in each region (from upper left to lower right) are shown as a box plot. The ends of the whiskers represent the minimum and maximum values. The bottom and top of the box represent the first and third quartiles, respectively. The red band inside the box represents the median. A green chain line represents the true probability p_i.

6 Conclusion

In this paper, we proposed a method to correct an estimation error of the EM reconstruction method. The experimental results using three large-scale real datasets showed that the proposed method significantly outperforms the existing inference methods when N is small or when most users adopt a small value of ϵ.

In this paper, we used LDP as a privacy metric, and showed the effectiveness of the proposed method using two location datasets (and one census dataset). In the field of location privacy, a privacy metric called *geo-indistiguishability* [4, 6, 7, 16] is also widely studied. Geo-indistiguishability can be regarded as a variant of LDP that multiplies ϵ in (9) by the distance (e.g., Euclidean distance) between x_i and x_j. It allows the adversary to infer approximate information about the original location (e.g., the fact that the user is in Paris), but does not allow him/her to infer the exact location (e.g., exact home address). By relaxing the privacy requirements in this way, the amount of noise added to the location data can be significantly reduced, and consequently the distribution estimation error (e.g., MSE, JS divergence) can be significantly reduced. As future work, we would like to evaluate the proposed method using geo-indistiguishability as a privacy metric.

Acknowledgement: This study was supported in part by JSPS KAKENHI 16K16069 and 16H02864. The authors would also like to thank the reviewers for many valuable comments.

References

[1] Aggarwal CC, Yu PS (2008) Privacy-Preserving Data Mining. Springer

[2] Agrawal D, Aggarwal CC (2001) On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'01), pp 247–255

[3] Agrawal R, Srikant R, Thomas D (2005) Privacy preserving OLAP. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data (SIGMOD'05), pp 251–262

[4] Andrés ME, Bordenabe NE, Chatzikokolakis K, Palamidessi C (2013) Geo-indistinguishability: Differential privacy for location-based systems. In: Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS'13), pp 901–914

[5] Bao Y, Ullah A (2007) The second-order bias and mean squared error of estimators in time-series models. Journal of Econometrics 140(2):650–669

[6] Bordenabe NE, Chatzikokolakis K, Palamidessi C (2014) Optimal geo-indistinguishable mechanisms for location privacy. In: Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS'14), pp 251–262

[7] Chatzikokolakis K, ElSalamouny E, Palamidessi C (2017) Practical mechanisms for location privacy. Proceedings on Privacy Enhancing Technologies (PoPETs) 2017(4):210–231

[8] Chelmis C, Kolte J, Prasanna VK (2015) Big data analytics for demand response: Clustering over space and time. In: Proceedings of 2015 IEEE International Conference on Big Data (BigData'15), pp 2223–2232

[9] Cover TM, Thomas JA (2006) Elements of Information Theory, Second Edition. Wiley-Interscience

[10] Data Breaches Increase 40 Percent in 2016, Finds New Report from Identity Theft Resource Center and CyberScout (2017) http://www.idtheftcenter.org/2016databreaches.html

[11] Duchi JC, Jordan MI, Wainwright MJ (2013) Local privacy and statistical minimax rates. In: Proceedings of the IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS'13), pp 429–438

[12] Dwork C (2006) Differential privacy. In: Proceedings of the 33rd international conference on Automata, Languages and Programming (ICALP'06), pp 1–12

[13] Dwork C, Roth A (2014) The Algorithmic Foundations of Differential Privacy. Now Publishers

[14] Eagle N, Pentland A, Lazer D (2009) Inferring friendship network structure by using mobile phone data. Proceedings of the National Academy of Sciences (PNAS) 106(36):15,274–15,278

[15] Efron B, Hastie T (2016) Computer Age Statistical Inference. Cambridge University

[16] Eltarjaman W, Dewri R, Thurimella R (2017) Location privacy for rank-based geo-query systems. Proceedings on Privacy Enhancing Technologies (PoPETs) 2017(4):19–38

[17] Úlfar Erlingsson, Pihur V, Korolova A (2014) RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS'14), pp 1054–1067

[18] Freudiger J, Shokri R, Hubaux JP (2011) Evaluating the privacy risk of location-based services. In: Proceedings of the 15th international conference on Financial Cryptography and Data Security (FC'11), pp 31–46

[19] Gelman A, Meng XL, Stern H (1996) Posterior predictive assessment of model fitness via realized discrepancies. Statistica Sinca 6:733–807

[20] Groetcsh C (1984) The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind. Pitman Advanced Publishing Program

[21] Hasan O, Habegger B, Brunie L, Bennani N, Damiani E (2009) A discussion of privacy challenges in user profiling with big data techniques: The EEXCESS use case. In: Proceedings of 2013 IEEE International Congress on Big Data (BigData Congress'13), pp 25–30

[22] Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning. Spinger, 2nd edition

[23] Hino H, Shen H, Murata N, Wakao S, Hayashi Y (2013) A versatile clustering method for electricity consumption pattern analysis in households. IEEE Transactions on Smart Grid 4(2):1048–1057

[24] Hsu J, Gaboardi M, Haeberlen A, Khanna S, Narayan A, Pierce BC, Roth A (2014) Differential privacy: An economic method for choosing epsilon. In: Proceedings of the 2014 IEEE 27th Computer Security Foundations Symposium (CSF'14), pp 398–410

[25] Huang Z, Du W (2008) OptRR: Optimizing randomized response schemes for privacy-preserving data mining. In: Proceedings of IEEE 24th International Conference on Data Engineering (ICDE'08), pp 705–714

[26] Hull B, Bychkovsky V, Zhang Y, Chen K, Goraczko M (2006) CarTel: A distributed mobile sensor computing system. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'06), pp 125–138

[27] Ishiguro M, Sakamoto Y, Kitagawa G (1997) Bootstrapping log likelihood and EIC, an extension of AIC. Annals of the Institute of Statistical Mathematics 49(3):411–434

[28] Johansen TA (1997) On Tikhonov regularization, bias and variance in nonlinear system identification. Automatica (3):441–446

[29] Jorgensen Z, Yu T, Cormode G (2015) Conservative or liberal? Personalized differential privacy. In: Proceedings of IEEE 31st International Conference on Data Engineering (ICDE'15), pp 1023–1034

[30] Kairouz P, Bonawitz K, Ramage D (2016) Discrete distribution estimation under local privacy. In: Proceedings of the 33rd International Conference on Machine Learning (ICML'16), pp 2436–2444

[31] Kairouz P, Oh S, Viswanath P (2016) Extremal mechanisms for local differential privacy. Journal of Machine Learning Research 17(1):492–542

[32] Li N, Lyu M, Su D (2016) Differential Privacy: From Theory to Practice. Morgan & Claypool Publishers

[33] Lichman M (2013) UCI machine learning repository. URL http://archive.ics.uci.edu/ml

[34] Lin J (1991) Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory 37(1):145–151

[35] Lisovich MA, Mulligan DK, Wicker SB (2010) Inferring personal information from demand-response systems. IEEE Security & Privacy 8(1):11–20

[36] Matsuo Y, Okazaki N, Izumi K, Nakamura Y, Nishimura T, Hasida K (2007) Inferring long-term user properties based on users' location history. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI'07), pp 2159–2165

[37] Pastore A, Gastpar M (2016) Locally differentially-private distribution estimation. In: Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT'16), pp 2694–2698

[38] Pyrgelis A, Troncoso C, Cristofaro ED (2017) What does the crowd say about you? Evaluating aggregation-based location privacy. Proceedings on Privacy Enhancing Technologies (PoPETs) 2017(4):76–96

[39] Qin Z, Yang Y, Yu T, Khalil I, Xiao X, Ren K (2016) Heavy hitter estimation over set-valued data with local differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16), pp 192–203

[40] Quercia D, Leontiadis I, McNamara L, Mascolo C, Crowcroft J (2011) Spotme if you can: Randomized responses for location obfuscation on mobile phones. In: Proceedings of the 2011 31st International Conference on Distributed Computing Systems (ICDCS'11), pp 363–372

[41] Rilstone P, Srivastava V, Ullah A (1996) The second-order bias and mean squared error of nonlinear estimators. Journal of Economics 75(2):369–395

[42] Schiaffino S, Amandi A (2009) Intelligent user profiling. In: Bramer M (ed) Artificial Intelligence, Springer-Verlag, pp 193–216

[43] Schubert E, Zimek A, Kriegel HP (2014) Generalized outlier detection with flexible kernel density estimates. In: Proceedings of the 14th SIAM International Conference on Data Mining (SDM'14), pp 542–550

[44] Sei Y, Ohusuga A (2017) Differential private data collection and analysis based on randomized multiple dummies for untrusted mobile crowdsensing. IEEE Transactions on Information Forensics and Security 12(4):926–939

[45] Sekimoto Y, Shibasaki R, Kanasugi H, Usui T, Shimazaki Y (2011) Pflow: Reconstructing people flow recycling large-scale social survey data. IEEE Pervasive Computing 10(4):27–35

[46] Shekhar S, Evans MR, Gunturi V, Yang K (2012) Spatial big-data challenges intersecting mobility and cloud computing. In: Proceedings of the 11th ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'12), pp 1–12

[47] van der Vaart AW (1998) Asymptotic Statistics. Cambridge University Press

[48] Wang W, Carreira-Perpiñán MÁ (2013) Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. CoRR abs/1309.1541, URL http://arxiv.org/abs/1309.1541

[49] Warner SL (1965) Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association 60(309):63–69

[50] Yang D, Zhang D, Qu B (2016) Participatory cultural mapping based on collective behavior data in location based social network. ACM Transactions on Intelligent Systems and Technology 7(3):30:1–30:23

[51] Zheng Y, Zhang L, Xie X, Ma WY (2009) Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web (WWW'09), pp 791–800

A Proof of Proposition 1

In this appendix, we prove **Proposition 1** in Section 4.4. The difference between the estimate $\hat{\mathbf{p}}$ of the EM reconstruction method and the true value $\mathbf{p}$ can be written as follows (see Lemma 3.1 in [41]):

$$\begin{aligned}\hat{\mathbf{p}} - \mathbf{p} = & -\mathbf{d} + \mathbf{Q}\left(\frac{1}{N}\sum_{n=1}^{N}\mathbf{V}_n\right)\mathbf{d} - \frac{1}{2}\mathbf{Q}\mathbb{E}[\nabla^2\mathbf{s}_n][\mathbf{d}\otimes\mathbf{d}] \\ & + O(N^{-3/2}).\end{aligned} \tag{42}$$

By using (37), (38), (40), and (42), the difference between the estimate of the proposed method (i.e., $\hat{\mathbf{p}} - \hat{\mathbf{a}}_{-1}$) and the true value $\mathbf{p}$ can be written as follows:

$$\hat{\mathbf{p}} - \hat{\mathbf{a}}_{-1} - \mathbf{p} = -\mathbf{d} + \mathbf{c}_1 - \mathbf{c}_2 + O(N^{-3/2}), \tag{43}$$

where

$$\mathbf{c}_1 = \frac{1}{N^2}\mathbf{Q}\left\{\sum_{\substack{n=1 \\ n\neq n'}}^{N}\sum_{n'=1}^{N}(\mathbf{V}_n\mathbf{d}_{n'})\right\} \tag{44}$$

$$\mathbf{c}_2 = \frac{1}{2N^2}\mathbf{Q}\left\{\mathbb{E}[\nabla^2\mathbf{s}_n]\sum_{\substack{n=1 \\ n\neq n'}}^{N}\sum_{n'=1}^{N}(\mathbf{d}_n\otimes\mathbf{d}_{n'})\right\}. \tag{45}$$

We note here that $\mathbf{d}$, $\mathbf{c}_1$, and $\mathbf{c}_2$ are terms of order $O(N^{-1/2})$, $O(N^{-1})$, and $O(N^{-1})$, respectively. This can be explained as follows. Since $\mathbb{E}[\mathbf{s}_n] = \mathbf{0}$ (as described in Section 4.1), the expectation of $\mathbf{d}_n$ in (38) is also $\mathbf{0}$: $\mathbb{E}[\mathbf{d}_n] = \mathbf{Q}\mathbb{E}[\mathbf{s}_n] = \mathbf{0}$. Therefore, $\mathbf{d}$ in (37) is a term of order $O(N^{-1/2})$ (due to the central limit theorem [47]). In addition, it follows from (20) that $\mathbb{E}[\mathbf{V}_n] = \mathbf{0}$. Then, since both $(\frac{1}{N}\sum_{n=1}^{N}\mathbf{V}_n)$ and $\mathbf{d}$ in (42) are terms of order $O(N^{-1/2})$, both the second term and the third term in (42) are terms of order $O(N^{-1})$. Since $\hat{\mathbf{a}}_{-1}$ in (40) converges to $\mathbf{a}_{-1}$ in (18) as N increases, $\hat{\mathbf{a}}_{-1}$ is also a term of order $O(N^{-1})$ (in the same way as $\mathbf{a}_{-1}$). Therefore, both $\mathbf{c}_1$ and $\mathbf{c}_2$ are terms of order $O(N^{-1})$ (note that the right-hand side of (43) is obtained by subtracting $\hat{\mathbf{a}}_{-1}$ from the right-hand side of (42)).

Thus, $\mathbf{MSE_{Proposal}}$ can be written, using (35) and (43), as follows:

$$\mathbf{MSE_{Proposal}} = \mathbb{E}[||\hat{\mathbf{p}} - \hat{\mathbf{a}}_{-1} - \mathbf{p}||_2^2] \quad (46)$$
$$= \mathbb{E}[|| - \mathbf{d} + \mathbf{c}_1 - \mathbf{c}_2 + O(N^{-3/2})||_2^2] \quad (47)$$
$$= b_{-1} - 2\mathbb{E}[\mathbf{d}^T\mathbf{c}_1] + 2\mathbb{E}[\mathbf{d}^T\mathbf{c}_2] + O(N^{-2}). \quad (48)$$

In the following, we show that both $\mathbb{E}[\mathbf{d}^T\mathbf{c}_1]$ and $\mathbb{E}[\mathbf{d}^T\mathbf{c}_2]$ are 0. We begin with $\mathbb{E}[\mathbf{d}^T\mathbf{c}_1]$, which can be written, using (37) and (44), as follows:

$$\mathbb{E}[\mathbf{d}^T\mathbf{c}_1] = \frac{1}{N^3}\sum_{m=1}^{N}\sum_{n=1}^{N}\sum_{\substack{n'=1\\ n\neq n'}}^{N}\mathbb{E}[\mathbf{d}_m^T\mathbf{Q}\mathbf{V}_n\mathbf{d}_{n'}]. \quad (49)$$

As previously described, $\mathbb{E}[\mathbf{d}_n] = \mathbb{E}[\mathbf{V}_n] = \mathbf{0}$. In addition, $\mathbf{d}_1, \cdots, \mathbf{d}_N$ are independent, and $\mathbf{V}_1, \cdots, \mathbf{V}_N$ are also independent. Using these facts, we have

$$\mathbb{E}[\mathbf{d}_m^T\mathbf{Q}\mathbf{V}_n\mathbf{d}_{n'}] = \begin{cases} \mathbb{E}[\mathbf{d}_n^T\mathbf{Q}\mathbf{V}_n]\mathbb{E}[\mathbf{d}_{n'}] = 0 & (\text{if } m = n) \\ \mathbb{E}[\mathbf{d}_{n'}^T\mathbf{Q}\mathbb{E}[\mathbf{V}_n]\mathbf{d}_{n'}] = 0 & (\text{if } m = n') \\ \mathbb{E}[\mathbf{d}_m^T\mathbf{Q}]\mathbb{E}[\mathbf{V}_n]\mathbb{E}[\mathbf{d}_{n'}] = 0 & (\text{if } m \neq n, m \neq n'). \end{cases} \quad (50)$$

By (49) and (50), we have

$$\mathbb{E}[\mathbf{d}^T\mathbf{c}_1] = 0. \quad (51)$$

We can show that $\mathbb{E}[\mathbf{d}^T\mathbf{c}_2]$ is 0 in the same way. $\mathbb{E}[\mathbf{d}^T\mathbf{c}_2]$ can be written, using (37) and (45), as follows:

$$\mathbb{E}[\mathbf{d}^T\mathbf{c}_2] = \frac{1}{2N^3}\sum_{m=1}^{N}\sum_{n=1}^{N}\sum_{\substack{n'=1\\ n\neq n'}}^{N}\mathbb{E}[\nabla^2\mathbf{s}_n]\cdot\mathbb{E}[\mathbf{Q}^T\mathbf{d}_m\otimes\mathbf{d}_n\otimes\mathbf{d}_{n'}], \quad (52)$$

where $\mathbf{A}\cdot\mathbf{B}$ ($\mathbf{A},\mathbf{B} \in \mathbb{R}^{K\times K\times K}$) represents the tensor contraction of $\mathbf{A}$ and $\mathbf{B}$ as follows:

$$\mathbf{A}\cdot\mathbf{B} = \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k=1}^{K}[\mathbf{A}]_{i,j,k}[\mathbf{B}]_{i,j,k} \quad (53)$$

($[\mathbf{A}]_{i,j,k}$ and $[\mathbf{B}]_{i,j,k}$ are the (i,j,k)-th element of $\mathbf{A}$ and $\mathbf{B}$, respectively). Since $\mathbb{E}[\mathbf{d}_n] = \mathbf{0}$ and $\mathbf{d}_1, \cdots, \mathbf{d}_N$ are independent, we have

$$\mathbb{E}[\mathbf{Q}^T\mathbf{d}_m\otimes\mathbf{d}_n\otimes\mathbf{d}_{n'}] = \begin{cases} \mathbb{E}[\mathbf{Q}^T\mathbf{d}_n\otimes\mathbf{d}_n]\otimes\mathbb{E}[\mathbf{d}_{n'}] = \mathbf{0} & (\text{if } m = n) \\ \mathbb{E}[\mathbf{Q}^T\mathbf{d}_{n'}\otimes\mathbb{E}[\mathbf{d}_n]\otimes\mathbf{d}_{n'}] = \mathbf{0} & (\text{if } m = n') \\ \mathbb{E}[\mathbf{Q}^T\mathbf{d}_m]\otimes\mathbb{E}[\mathbf{d}_n]\otimes\mathbb{E}[\mathbf{d}_{n'}] = \mathbf{0} & (\text{if } m \neq n, m \neq n'). \end{cases} \quad (54)$$

By (52) and (54), we have

$$\mathbb{E}[\mathbf{d}^T\mathbf{c}_2] = 0. \quad (55)$$

By (48), (51), and (55), the equation (41) holds. □

It should be noted that we can immediately derive from (43) that both b_{-1} and $b_{-3/2}$ will be reduced to zero by subtracting $(\hat{\mathbf{a}}_{-1} - \mathbf{d})$ from $\hat{\mathbf{p}}$ (since the MSE will be $\mathbb{E}[||\hat{\mathbf{p}} - (\hat{\mathbf{a}}_{-1} - \mathbf{d}) - \mathbf{p}||_2^2] = \mathbb{E}[||\mathbf{c}_1 - \mathbf{c}_2 + O(N^{-3/2})||_2^2] = O(N^{-2})$ in this case). However, the estimation of $\mathbf{d}$ in (37) is very challenging, since it becomes zero after replacing $\mathbf{p}$ with $\hat{\mathbf{p}}$: $\frac{1}{N}\sum_{n=1}^{N}\mathbf{d}_n(\hat{\mathbf{p}}) = \mathbf{Q}(\frac{1}{N}\sum_{n=1}^{N}\mathbf{s}_n(\hat{\mathbf{p}})) = \mathbf{0}$ (by (16)). In other words, *we cannot use a plug-in estimate of* $\mathbf{d}$. Thus, we leave finding a practical method to reduce both b_{-1} and $b_{-3/2}$ for future work.

B Computation of $\hat{\mathbf{a}}_{-1}$ in the Proposed Method

Here we describe how to compute $\hat{\mathbf{a}}_{-1}$ in (28) from $\hat{\mathbf{p}}$ and $\mathbf{g}_n$ ($1 \leq n \leq N$) by using (30), (31), and (32) with time complexity $O(NK^2)$.

We begin by explaining the first term of (28):

$$\frac{1}{N}\sum_{n=1}^{N}(\hat{\mathbf{V}}_n\hat{\mathbf{Q}}\mathbf{s}_n) \in \mathbb{R}^K. \quad (56)$$

$\hat{\mathbf{V}}_n \in \mathbb{R}^{K\times K}$, $\hat{\mathbf{Q}} \in \mathbb{R}^{K\times K}$, and $\mathbf{s}_n \in \mathbb{R}^K$ are computed from $\hat{\mathbf{p}}$ and $\mathbf{g}_n$ (see (22), (30), (31), and (32)). A straightforward computation of (56) requires time complexity of $O(NK^3)$, since the multiplication of $\hat{\mathbf{V}}_n \in \mathbb{R}^{K\times K}$ and $\hat{\mathbf{Q}} \in \mathbb{R}^{K\times K}$ requires time complexity of $O(K^3)$. However, we can compute (56) with time complexity $O(NK^2)$ by transforming $\hat{\mathbf{V}}_n\hat{\mathbf{Q}}\mathbf{s}_n$ as follows:

$$\hat{\mathbf{V}}_n\hat{\mathbf{Q}}\mathbf{s}_n = (\mathbf{s}_n^T(\hat{\mathbf{V}}_n\hat{\mathbf{Q}})^T)^T = (\mathbf{s}_n^T\hat{\mathbf{Q}}^T\hat{\mathbf{V}}_n^T)^T. \quad (57)$$

We can compute the multiplication of $\mathbf{s}_n^T \in \mathbb{R}^K$ by $\hat{\mathbf{Q}}^T \in \mathbb{R}^{K\times K}$ with time complexity $O(K^2)$. Similarly, we can compute the multiplication of $\mathbf{s}_n^T\hat{\mathbf{Q}}^T \in \mathbb{R}^K$ by $\hat{\mathbf{V}}_n^T \in \mathbb{R}^{K\times K}$ with time complexity $O(K^2)$. Thus, we can compute (57) with time complexity $O(K^2)$, and (56) with time complexity $O(NK^2)$.

We then explain the second term of (28), which is more complicated:

$$\frac{1}{2N^2}\sum_{n=1}^{N}(\nabla^2\mathbf{s}_n)\sum_{n=1}^{N}(\hat{\mathbf{Q}}\mathbf{s}_n\otimes\hat{\mathbf{Q}}\mathbf{s}_n) \in \mathbb{R}^K. \quad (58)$$

Let $\mathbf{A} \in \mathbb{R}^{K\times K\times K}$ be $\sum_{n=1}^{N}(\nabla^2 \mathbf{s}_n)$ in (58). Let further $\mathbf{B} \in \mathbb{R}^{K\times K}$ be $\sum_{n=1}^{N}(\hat{\mathbf{Q}}\mathbf{s}_n \otimes \hat{\mathbf{Q}}\mathbf{s}_n)$ in (58). Then, (58) can be written as follows:

$$\frac{1}{2N^2}\mathbf{AB} \in \mathbb{R}^K. \tag{59}$$

Note that $\mathbf{AB}$ is a K-dimensional vector, whose i-th element $[\mathbf{AB}]_i$ is written as follows:

$$[\mathbf{AB}]_i = \sum_{j=1}^{K}\sum_{k=1}^{K}[\mathbf{A}]_{i,j,k}[\mathbf{B}]_{j,k} \tag{60}$$

($[\mathbf{A}]_{i,j,k}$ is the (i,j,k)-th element of $\mathbf{A}$, and $[\mathbf{B}]_{j,k}$ is the (j,k)-th element of $\mathbf{B}$). In the following, we describe how to compute $\mathbf{AB}$ in detail.

$[\mathbf{A}]_{i,j,k}$ can be expressed, using (24), as follows:

$$[\mathbf{A}]_{i,j,k} = \sum_{n=1}^{N}\frac{1}{(\hat{\mathbf{p}}^T\mathbf{g}_n)^3}[\mathbf{g}_n]_i[\mathbf{g}_n]_j[\mathbf{g}_n]_k \tag{61}$$

($[\mathbf{g}_n]_i$ is the i-th element of $\mathbf{g}_n \in [0,1]^K$). $[\mathbf{B}]_{j,k}$ can be expressed as follows:

$$[\mathbf{B}]_{j,k} = \sum_{n=1}^{N}[\hat{\mathbf{Q}}\mathbf{s}_n]_j[\hat{\mathbf{Q}}\mathbf{s}_n]_k \tag{62}$$

($[\hat{\mathbf{Q}}\mathbf{s}_n]_j$ is the j-th element of $\hat{\mathbf{Q}}\mathbf{s}_n \in \mathbb{R}^K$).

$[\mathbf{AB}]_i$ in (60) can be written, using (61), as follows:

$$[\mathbf{AB}]_i = \sum_{n=1}^{N}\frac{1}{(\hat{\mathbf{p}}^T\mathbf{g}_n)^3}[\mathbf{g}_n]_i u_n, \tag{63}$$

where

$$u_n = \sum_{j=1}^{K}\sum_{k=1}^{K}[\mathbf{g}_n]_j[\mathbf{g}_n]_k[\mathbf{B}]_{j,k} \in \mathbb{R}. \tag{64}$$

Therefore, we can compute $\mathbf{AB} \in \mathbb{R}^K$ as follows:

1. Compute $\mathbf{B} \in \mathbb{R}^{K\times K}$ from $\hat{\mathbf{Q}}\mathbf{s}_n \in \mathbb{R}^K$ by using (62).
2. Compute $\mathbf{u} = (u_1, \cdots, u_N)^T \in \mathbb{R}^N$ from $\mathbf{g}_n$ $(1 \leq n \leq N)$ and $\mathbf{B}$ by using (64).
3. Compute $\mathbf{AB} \in \mathbb{R}^K$ from $\hat{\mathbf{p}}$, $\mathbf{g}_n$ $(1 \leq n \leq N)$, and $\mathbf{u}$ by using (63).

Both the steps 1 and 2 require time complexity of $O(NK^2)$, and the step 3 requires time complexity of $O(NK)$. Thus, we can compute $\mathbf{AB}$ with time complexity of $O(NK^2)$. By using (59), we can compute the second term of (28) with time complexity of $O(NK^2)$.

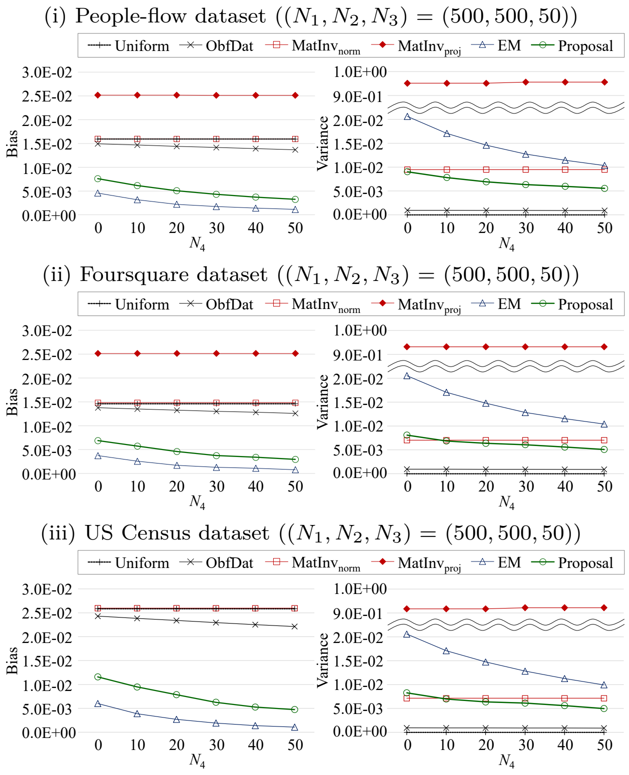
Therefore, we can compute $\hat{\mathbf{a}}_{-1}$ in (28) with time complexity of $O(NK^2)$.

C Analysis of the Bias and the Variance in Our Experiments

In this appendix, we decompose the MSE in Fig. 5, 6, and 7 into the bias and variance, and show the results of the bias and variance.

Fig. 13, 14, and 15 show the empirical bias and sample variance corresponding to Fig. 5, 6, and 7, respectively. The empirical bias and sample variance were computed by replacing the expectation $\mathbb{E}$ in (5) and (6) with the empirical mean over N samples $Y_1, \cdots, Y_N$. It can be seen that the variance of the estimate $\hat{\mathbf{p}}$ is 0 in **Uniform**, and is close to 0 in **ObfDat**, as described in Section 5.2. This is the reason **Uniform** or **ObfDat** provided the best performance when N is very small.

It can also be seen that the bias of **Proposal** is larger than that of **EM**. We consider this is because we applied the Tikhonov regularization to compute $\hat{\mathbf{Q}}$ in (32), as described in Section 4.2. However, **Proposal** significantly reduces the variance, and therefore reduces the MSE and the JS divergence in Fig. 13, 14, and 15. We consider this is because $\hat{\mathbf{a}}$ can be used as a rough approximation of $\hat{\mathbf{p}} - \mathbf{p}$, as described in Section 4.2. This result is also consistent with **Proposition 1**, which states that the MSE is reduced (especially when N is small) by subtracting $\hat{\mathbf{a}}$ from $\hat{\mathbf{p}}$.

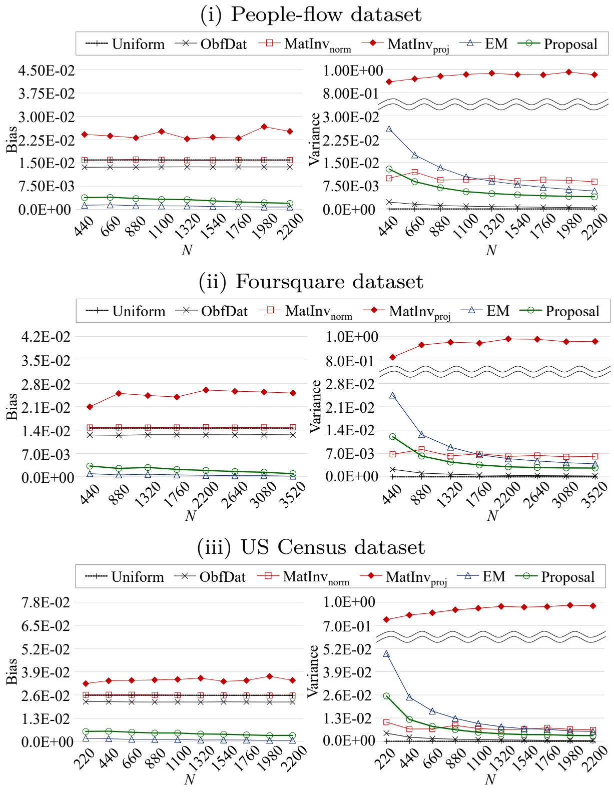


(i) People-flow dataset $((N_1, N_2, N_3) = (500, 500, 50))$

(ii) Foursquare dataset $((N_1, N_2, N_3) = (500, 500, 50))$

(iii) US Census dataset $((N_1, N_2, N_3) = (500, 500, 50))$

Fig. 13. The bias and the variance in the case where $(N_1, N_2, N_3) = (500, 500, 50)$ and $N_4 \in [0, 50]$.

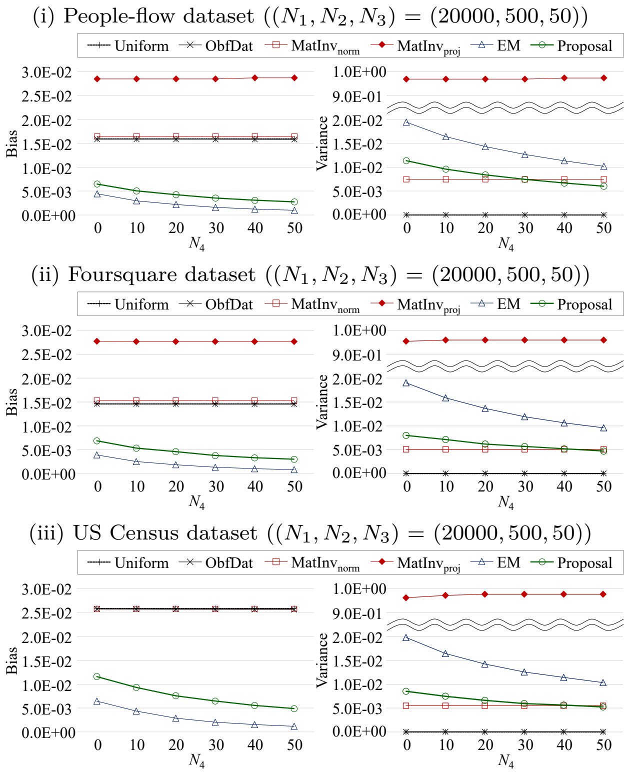


(i) People-flow dataset

(ii) Foursquare dataset

(iii) US Census dataset

Fig. 14. The bias and the variance in the case where N is changed while fixing $N_1 : N_2 : N_3 : N_4 = 10 : 10 : 1 : 1$.



(i) People-flow dataset $((N_1, N_2, N_3) = (20000, 500, 50))$

(ii) Foursquare dataset $((N_1, N_2, N_3) = (20000, 500, 50))$

(iii) US Census dataset $((N_1, N_2, N_3) = (20000, 500, 50))$

Fig. 15. The bias and the variance in the case where $N_1 = 20000$.