Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2024) 32: 34 – 50

doi:10.55730/1300-0632.4054

Differentially private online Bayesian estimation with adaptive truncation

Sinan YILDIRIM[1,2*]

[1]Faculty of Engineering and Natural Sciences, Sabancı University, İstanbul, Turkiye
[2]Center of Excellence in Data Analytics (VERİM), Sabancı University, İstanbul, Turkiye

Received: 06.01.2023 • **Accepted/Published Online:** 04.01.2024 • **Final Version:** 07.02.2024

Abstract: In this paper, a novel online and adaptive truncation method is proposed for differentially private Bayesian online estimation of a static parameter regarding a population. A local differential privacy setting is assumed where sensitive information from individuals is collected on an individual level and sequentially. The inferential aim is to estimate, on the fly, a static parameter regarding the population to which those individuals belong. We propose sequential Monte Carlo to perform online Bayesian estimation. When individuals provide sensitive information in response to a query, it is necessary to corrupt it with privacy-preserving noise to ensure the privacy of those individuals. The amount of corruption is proportional to the sensitivity of the query, which is determined usually by the range of the queried information. The proposed truncation technique adapts to the previously collected data to adjust the query range for the next individual. The idea is that, based on previous data, one can carefully arrange the interval into which the next individual's information is to be truncated before being distorted with privacy-preserving noise. In this way, predictive queries are designed with small sensitivity, hence small privacy-preserving noise, enabling more accurate estimation while maintaining the same level of privacy. To decide on the location and the width of the interval, an exploration-exploitation approach is employed, *a la* Thompson sampling, with an objective function based on Fisher information. The merits of the methodology are shown with numerical examples.

Key words: Differential privacy, Bayesian statistics, sequential Monte Carlo, online learning

1. Introduction

During the past couple of decades, there has been a rapid increase in the amount of collected data as well as concerns about individuals' privacy. This has made privacy-preserving data analysis a popular and important subject in data science. Along the way, differential privacy has become a popular framework for privacy-preserving data sharing algorithms [1, 2].

There are two conflicting interests in privacy-preserving data analysis: (i) The individuals of a population who contribute to a data set with their sensitive information want to protect their privacy against all possible adversaries. Conflicting with that, it is desirable to be able to estimate a common quantity of interest regarding the population based on sensitive data with reasonable accuracy. To put the conflict in a statistical context, let $X_t \sim \mathcal{P}_\theta$ be the sensitive information of t'th individual of a sample randomly chosen from a large population with a population distribution $\mathcal{P}_\theta$. The goal is to estimate θ while also protecting the privacy of the individuals contributing to the sample, i.e. without revealing much information about X_ts individually.

This paper studies a setting where sensitive information from individuals is collected on an individual

*Correspondence: sinanyildirim@sabanciuniv.edu

level, i.e. every individual randomly corrupts their information before sharing it. In that way, the proposed methodology fits in the framework of local differential privacy [3], where there is no need for a central aggregator. Local differential privacy also addresses concerns about the possibility of an adversary having access to the sensitive data stored in the database, thereby offering a stronger notion of privacy.

We are particularly interested in online Bayesian estimation of θ as we sequentially collect $Y_1, Y_2, \ldots$, which are the corrupted versions of $X_1, X_2, \ldots$ respectively. The cases where individuals contribute to a data set sequentially in time are not common: Imagine, for example, web users registering to a web application after entering their information, patients being admitted to a hospital, customers applying for a bank loan, etc. The presence of such scenarios enables two methodological opportunities/challenges:

1. One can (and/or should) estimate the static parameter on the fly, that is, update the estimate as data are being received.

2. As the parameter is being estimated, one can adaptively adjust the query for the next individual's information to make the response as informative as possible. For example, if, based on the noisy income values collected so far from 100 individuals, it has been estimated that the mean income of the population is around $\hat{\mu}$, the next individual can be asked to provide their income information after truncating it to an interval around $\hat{\mu}$, such as $[\hat{\mu} - \Delta, \hat{\mu} + \Delta]$, and then privatizing it by adding noise.

   The motivation behind pursuing such an adaptive truncation technique is to improve the estimation performance with less noisy data while maintaining a given level of privacy. As we shall see below, the standard deviation of the privacy-preserving noise added to the outcome of a query is proportional to the sensitivity of the query. By default, the queried information may be unbounded or have very large ranges, which renders many practical privacy-preserving mechanisms useless. Continuing with the income example above, assume that the natural limits of an income are $[x_{\min}, X_{\max}]$ so that a query that directly asks for income information has a sensitivity of $X_{\max} - x_{\min}$, which is expectedly large. With adaptive truncation, the query interval for 101'th individual would be $[\hat{\mu} - \Delta, \hat{\mu} + \Delta]$ with sensitivity 2Δ.

This paper contributes to the literature on differential privacy by addressing the two challenges described above with a novel methodology. For the first challenge, that is, online estimation of θ, we propose a sequential Monte Carlo (SMC) method [4] for online Bayesian inference. For the second challenge, we propose an adaptive truncation method that employs an exploration-exploitation heuristic to maximize the aggregate information in the sequence of observations $Y_1, Y_2, \ldots$ about θ. The Fisher information is chosen as a measure of informativeness as suggested in [5, 6]. The exploration part of the proposed approach can be seen as an instance of Thompson sampling [7] from reinforcement learning, as we will show in subsection 3.2. The exploitation part consists of finding the truncation points that make the resulting observations most informative in terms of Fisher information. Finally, for the exploitation step, we pay special attention to location-scale families and show that the maximization task can be performed for all time steps once and for all. To the best of our knowledge, this is the first work that tackles the problem of online differentially private Bayesian estimation with adaptive queries.

The presented methodology has potential use for scenarios where a stream of sensitive data is collected from individuals and needs to be processed privately and efficiently for various data-based goals varying from inference to decision-making. Such scenarios are ample in today's world, where valuable and sensitive personal data such as health data, financial transactions, and internet activity, continually cumulate and can be used for various useful purposes in anomaly detection, analysis of health data, user recommendation etc.

The organization of the paper is as follows. In subsection 2.1, the basic concepts of differential privacy are introduced, and in subsection 2.2 the existing related works in the literature are discussed. Section 3 contains a discussion of the problem of online parameter estimation using privatized noisy statistics of the sensitive data, as well as the proposed methodology in general. Subsections 3.1 and 3.2 contain the technical details of the presented methodology. Section 4 contains the numerical experiments. Finally, concluding remarks and possible future work are given in Section 5. This paper includes an Appendix section for some deferred details.

2. Background and related work

This section is reserved for an introduction to the basic concepts of differential privacy, followed by a discussion of the data privacy literature relevant to this work.

2.1. Differential privacy

Let $\mathcal{X}$ be a set of individual data values and $\mathscr{X} = \bigcup_{n=1}^{\infty} \mathcal{X}^n$ be the set of data sets. Define the Hamming distance between the data sets $\boldsymbol{x}, \boldsymbol{x}' \in \mathscr{X}$ as the number of different elements between those data sets, denoted by $h(\boldsymbol{x}, \boldsymbol{x}')$. We call two data sets $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}^n$ neighbours if $h(\boldsymbol{x}, \boldsymbol{x}') = 1$. A randomized algorithm can be defined as a couple $\mathcal{A} = (A, \mu)$, where $A : \mathscr{X} \times \mathcal{E} \mapsto \mathcal{Y}$ is a function and μ is a probability distribution on $\mathcal{E}$, which represents the randomness intrinsic to $\mathcal{A}$. Upon taking an input $\boldsymbol{x} \in \mathscr{X}$, the randomised algorithm $\mathcal{A}$ generates random numbers $\omega \sim \mu(\cdot)$ in $\mathcal{E}$ and outputs $A(\boldsymbol{x}, \omega)$. A differential private algorithm ensures a certain sense of similarity between the probability distributions of $A(\boldsymbol{x}, \omega)$ and $A(\boldsymbol{x}', \omega)$ when $\boldsymbol{x}$ and $\boldsymbol{x}'$ are neighbors.

Definition 1 (Differential privacy (DP) [1]) *A randomised algorithm $\mathcal{A} = (A, \mu)$ is (ϵ, δ)-DP if*

$$\mathbb{P}\left[A(\boldsymbol{x}, \omega) \in S\right] \leq e^{\epsilon} \mathbb{P}\left[\mathcal{A}(\boldsymbol{x}', \omega) \in S\right] + \delta, \quad \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathscr{X} \, s.t. \, h(\boldsymbol{x}, \boldsymbol{x}') = 1, \quad \forall S \subseteq \mathcal{Y}.$$

where the randomness is with respect to $\omega \sim \mu(\cdot)$. We say $\mathcal{A}$ is ϵ-DP when $\delta = 0$.

As far as privacy is concerned, both privacy parameters (ϵ, δ) are desired to be as small as possible. The following theorem states that (ϵ, δ)-DP is maintained by postprocessing the output of an (ϵ, δ)-DP algorithm.

Theorem 1 (Postprocessing) *Define functions $A_1 : \mathscr{X} \times \mathcal{E}_1 \mapsto \mathcal{Y}_1$ and $A_2 : \mathcal{Y}_1 \times \mathcal{E}_2 \mapsto \mathcal{Y}_2$; and probability distributions μ_1, μ_2 on $\mathcal{E}_1$, $\mathcal{E}_2$, respectively. Furthermore, let $A : \mathscr{X} \times \mathcal{E}_1 \times \mathcal{E}_2 \mapsto \mathcal{Y}_2$ be defined by $A(\boldsymbol{x}, \omega_1, \omega_2) = A_2(A_1(\boldsymbol{x}, \omega_1), \omega_2)$, and $\mu = \mu_1 \otimes \mu_2$. Then, if $\mathcal{A}_1 = (A_1, \mu_1)$ is (ϵ, δ)-DP, $\mathcal{A} = (A, \mu)$ is (ϵ, δ)-DP, too.*

Let $\varphi : \mathscr{X} \mapsto \mathbb{R}$ be a function and assume that $\varphi(\boldsymbol{x})$ is queried. One common way of achieving differential privacy, in this case, is the *Laplace mechanism* [8], which relies on the L_1-*sensitivity* of φ, given by

$$\Delta\varphi = \sup_{\boldsymbol{x}, \boldsymbol{x}' : h(\boldsymbol{x}, \boldsymbol{x}') = 1} |\varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x}')|. \tag{1}$$

Theorem 2 (Laplace mechanism) *The algorithm that returns $\varphi(\boldsymbol{x}) + \Delta\varphi V$ given the input $\boldsymbol{x} \in \mathscr{X}$, where $V \sim \text{Laplace}(1/\epsilon)$, is ϵ-DP.*

Other useful definitions of data privacy have close relations to differential privacy. Some important examples are Gaussian differential privacy [9] and zero-concentrated differential privacy [10], both of which promote the Gaussian mechanism [2] (where V in Theorem 2 has a normal distribution) as its primary mechanism for providing privacy. The Gaussian mechanism can also provide (ϵ, δ)-DP for $\delta > 0$ if the variance is modified to depend on δ also.

For the rest of the paper, we will consider the Laplace mechanism to provide ϵ-DP for the sake of simplicity. We remark, however, that other additive mechanisms to provide privacy in other senses also fit into our methodology with minor changes. In particular, our methodology applies to the Gaussian mechanism in an almost identical manner.

2.2. Related work

Differentially private Bayesian inference of θ has been the subject of several recent studies, with Monte Carlo being the main methodological tool for inference. Differentially private stochastic gradient MCMC algorithms were proposed in [11, 12], while some reversible differentially private MCMC algorithms were proposed in [13–15]. Those algorithms require as many interactions with sensitive data as the number of iterations they run for. An alternative scheme to that is called input perturbation, where the sensitive data are shared privately once and for all, and all the subsequent Bayesian inference is performed on the perturbed data without further interaction with the sensitive data [6, 16–21]. All the cited works above consider differentially private Bayesian inference conditional on a batch (static) data set. Unlike those works, in this paper, we consider the case with continual observations, where data from the individuals are collected sequentially in a privacy-preserving way.

Differentially private estimation has been studied under the assumption of continual observation in several works that are initiated by [22]; other important contributions include [23, 24]. However, those works are usually applied to online tracking of dynamic summaries of data, such as the count of a certain property, rather than estimating a static parameter of the population from which the data are being received. In particular, they do not consider Bayesian estimation.

Locally differentially private protocols for estimation of frequency distributions [25], which is somewhat more specific than the general parameter estimation problem, have been studied in several works. For frequency estimation with LDP protocols, Barnes et al. [5] and Steinberger [26] consider Fisher Information as the utility metric for finding nearly optimal LDP protocol for parameter estimation. Lopuhaä-Zwakenberg et al. [27] use Fisher information for comparing the utility of various LDP protocols for frequency estimation and finding the optimal one. The Fisher information is also proposed in [6] for statistic selection for differentially private Bayesian estimation. In these works, the estimation task is mainly assumed a static one, and the main focus is on a single protocol that is chosen once and for all for a given estimation task. In contrast to these approaches, we incorporate the Fisher information into our estimation method algorithm in a dynamic way to adapt to cumulating information in time. To the best of our knowledge, the proposed method in this paper is the first one that adaptively uses Fisher information to dynamically optimise the query in a local differential privacy setting.

3. Differentially private online parameter estimation with adaptive queries

In this section, we describe in detail the methodological contributions of the paper. First, we formalize the inference problem and provide a general framework of the methodology, which is outlined in Algorithm 1. Then, in subsections 3.1 and 3.2 we give the details required to implement Algorithm 1.

Problem definition: In this paper, we assume a sequence of i.i.d. data points

$$X_t \overset{\text{i.i.d.}}{\sim} \mathcal{P}_\theta, \quad t \geq 1,$$

where X_t is some sensitive information that belongs to the t'th individual sampled from a population. We want to estimate the unknown parameter θ of the population distribution $\mathcal{P}_\theta$. However, we are not allowed to access to X_t's directly; instead, individuals share their information through a function $s_t : \mathcal{X} \mapsto \mathbb{R}$ and with privacy-preserving noise as

$$Y_t = s_t(X_t) + \Delta s_t V_t, \quad V_t \sim \text{Laplace}(1/\epsilon), \quad t \geq 1, \tag{2}$$

where Δs_t is the sensitivity defined as in (1). We consider online estimation of θ when $\{Y_t\}_{t\geq 1}$ are observed sequentially in time. The recursion that corresponds to a sequential estimation procedure can be written down generically as

$$\Theta_t = G(\Theta_{t-1}, Y_{1:t}, s_{1:t}).$$

The update function G produces Θ_t using all the information up to time t, which includes Θ_{t-1}, the functions $s_{1:t}$, and the observations $Y_{1:t}$. Generally, Θ_t is not necessarily a point estimate but a collection of variables needed to construct the estimation of θ at time t. For example, in SMC for Bayesian estimation, Θ_t can correspond to the particle system at time t. Details of such an algorithm will be provided in subsection 3.1.

This paper focuses on the possibility of choosing s_t adaptively so that θ is estimated with improved accuracy relative to its nonadaptive counterpart. The choice of s_t is important because s_t determines how much information is contained in Y_t about θ in two ways [6]: (i) The first way is related to the sufficiency or informativeness of s_t in a classical way. For example, if the population distribution were $\mathcal{P}_\theta = \mathcal{N}(\theta, 1)$ with an unknown mean θ, then $s_t(x_t) = x_t$ would be a better choice than $s_t(x_t) = |x_t|$ since $|x_t|$ masks the information that is contained in x_t about θ. (ii) Secondly, the standard deviation of the privacy-preserving noise is proportional to the sensitivity Δs_t. A large Δs_t leads to the perturbation of useful information with too much privacy-preserving noise. (As an extreme case, think of an unbounded s_t.) On the flip side, making Δs_t too small could result in a small amount of information about θ. (Imagine a constant $s_t(\cdot)$, which has $\Delta s_t = 0$ but carries no information about θ.) Therefore, truncation and sensitivity establish a trade-off.

Example 1 *Assume that our goal is to learn the average income θ of the individuals in a given population, with a population distribution $\mathcal{N}(\theta, \sigma^2)$, where σ^2 is known. Assume that data is collected from (some of) the individuals in this population in a sequential way. However, since the income information is sensitive, each individual's income is recorded (or shared by the individual) with privacy-preserving noise. For practical applications, we must ensure a finite sensitivity, which is usually achieved either by truncating the income value into the natural boundaries of the information or more generally into an interval $[l, r]$*

$$Y_t = \min\{\max\{X_t, l\}, r\} + (r - l)V_t, \quad V_t \sim \text{Laplace}\,(1/\epsilon)\,.$$

If the interval $[l, r]$ is wide, true income X_t is not likely to be truncated but Y_t suffers a large noise for ensuring the given level of privacy. On the other hand, if $[l, r]$ is small, X_t is likely to be truncated but Y_t is less noisy. This makes a trade-off between truncation and privacy-preserving noise, the two undesired components in terms of statistical inference. It would therefore be reasonable to adjust the interval adaptively as we collect data, where the interval for receiving the t'th individual's data is denoted by $[l_t, r_t]$. We aim to set $[l_t, r_t]$ so that it will

likely contain the true value and it is small so that the required privacy-preserving noise has a small variance. We could do that by positioning $[l_t, r_t]$ *around the most recent estimate of* θ *if it is a location parameter.*

General framework: The general online estimation method with adaptive functions s_t is given in Algorithm 1. The algorithm outlines the general idea in this paper: We gain knowledge about θ as we observe Y_t's; which we use to adapt the statistic s_{t+1} such that the new observation Y_{t+1} carries more information about θ than it would with an arbitrary choice of s_{t+1}.

Algorithm 1 is ϵ-DP. Each observation Y_t belongs to an individual and is shared with ϵ-DP. Furthermore, all the updates in Algorithm 1 are performed using the shared data $\{Y_t\}_{t\geq 1}$ and *not* the private data $\{X_t\}_{t\geq 1}$. Therefore, by Theorem 1, those updates do not introduce any further privacy leaks. A more formal statement in Proposition 1.

Algorithm 1: Differentially private online learning - general scheme.

Initialise the estimation system Θ_0 and $s_1(\cdot)$.
for $t = 1, 2, \ldots$ **do**
The function s_t is revealed to individual t, which shares his/her data X_t as

$$Y_t = s_t(X_t) + \Delta s_t V_t, \quad V_t \sim \text{Laplace}\,(1/\epsilon)$$

Update the estimation system Θ_t as

$$\Theta_t = G(\Theta_{t-1}, Y_{1:t}, s_{1:t}) \tag{3}$$

Update the function

$$s_{t+1} = H(\Theta_t) \tag{4}$$

end

Proposition 1 *Algorithm 1 is* ϵ*-DP.*

Proof Let $R_t = (\Theta_t, S_t, Y_t)$ be the revealed outputs of Algorithm 1 at time t. For any $n \geq 1$, the conditional distribution of $R_{1:n}$ at $r_{1:n} = (\theta_{1:n}, s_{1:n}, y_{1:n})$ is given by

$$P(\mathrm{d}r_{1:n}|X_{1:n} = x_{1:n}) = \prod_{t=1}^{n} \text{Laplace}(y_t - s_t(x_t), \Delta s_t/\epsilon)\mathrm{d}y_t \prod_{t=1}^{n} P(\mathrm{d}s_t|\theta_t)p(\mathrm{d}\theta_t|\theta_{1:t-1}, y_{1:t}, s_{1:t}),$$

where we used $\text{Laplace}(y; b)$ to denote the pdf of $\text{Laplace}(b)$. The ratio between the conditional distributions with $x_{1:n}$ and $x'_{1:n}$ for any neighbour pair $x_{1:n}, x'_{1:n}$ differing by some k'th element is given by

$$e^{-\epsilon} \leq \frac{P(\mathrm{d}r_{1:n}|X_{1:n} = x_{1:n})}{P(\mathrm{d}r_{1:n}|X_{1:n} = x'_{1:n})} = \frac{\prod_{t=1}^{n} \text{Laplace}(y_t - s_t(x_t), \Delta s_t/\epsilon)}{\prod_{t=1}^{n} \text{Laplace}(y_t - s_t(x'_t), \Delta s_t/\epsilon)} = \frac{\text{Laplace}(y_k - s_k(x_k), \Delta s_k/\epsilon)}{\text{Laplace}(y_k - s_k(x'_k), \Delta s_k/\epsilon)} \leq e^{\epsilon},$$

where the first equality is because the other factors do not depend on $x_{1:n}$ (or $x'_{1:n}$), the second equality is because $x_{1:n}$ and $x'_{1:n}$ differ by the k'th element only. □

Remark 1 *We remark that Algorithm 1 also satisfies ϵ-local differential privacy [3], since ϵ-DP is satisfied for every user's piece of data. This can be seen easily by checking that for every n the ratio*

$$e^{-\epsilon} \leq \frac{P(\mathrm{d}r_n | X_{1:n-1} = x_{1:n-1}, X_n = x_n, R_{1:n-1} = r_{1:n-1})}{P(\mathrm{d}r_n | X_{1:n-1} = x_{1:n-1}, X_n = x'_n, R_{1:n-1} = r_{1:n-1})} \leq e^{\epsilon}$$

for every $x_{1:n-1}$, x_n, x'_n, and $r_{1:n}$.

In the subsequent subsections 3.1 and 3.2, we describe the methods for the updates in (3) and in (4).

3.1. Sequential Monte Carlo for online Bayesian estimation

In this section, we focus on G in (3), which stands for the parameter estimation update upon receiving a new observation. We consider the functions s_t given and present an SMC method for online Bayesian estimation of θ. SMC is a popular numerical method for online Bayesian inference; see [4, 28] for some pioneer works. Let $p_\theta(\cdot)$ be the probability density (or mass) function (pdf or pmf) of $\mathcal{P}_\theta$. With a prior distribution $\eta(\theta)$ on θ, the following sequence of posterior distributions is targeted sequentially with SMC.

$$p^{\epsilon}_{s_{1:t}}(\theta, x_{1:t} | y_{1:t}) \propto p^{\epsilon}_{s_{1:t}}(\theta, x_{1:t}, y_{1:t}) = \eta(\theta) \prod_{k=1}^{t} p_\theta(x_k) \mathrm{Laplace}\left(y_k - s_k(x_k), \Delta s_k / \epsilon\right), \quad t = 1, \ldots, n. \tag{5}$$

A Monte Carlo approximation is indeed necessary for those posterior distributions since they are intractable having no closed form. At time t, SMC approximates the posterior distribution in (5) with a discrete probability distribution having $N > 1$ particles (points of mass) $\{(\theta^{(i)}, x^{(i)}_{1:t}); i = 1, \ldots, N\}$ with particle weights $\{w^{(i)}_t; i = 1, \ldots, N\}$ as

$$p^{\epsilon,N}_{s_{1:t}}(\mathrm{d}(\theta, x_{1:t}) | y_{1:t}) = \sum_{i=1}^{N} w^{(i)}_t \delta_{(\theta^{(i)}, x^{(i)}_{1:t})}(\mathrm{d}(\theta, x_{1:t})).$$

By marginalizing out the $x_{1:t}$ component in the above approximation, we can also obtain the particle approximation of the marginal posterior distribution of θ given the observations.

$$p^{\epsilon,N}_{s_{1:t}}(\mathrm{d}\theta | y_{1:t}) = \sum_{i=1}^{N} w^{(i)}_t \delta_{\theta^{(i)}}(\mathrm{d}\theta).$$

At time-step t, the particles and their weights from time $t-1$ are updated after the resampling, rejuvenation, propagation, and weighting steps. The propagation and weighting steps are necessary to track the evolving posterior distributions, while the rejuvenation and resampling steps prevent the particle approximation from collapsing to a single point. The update at a single time-step of SMC is detailed in Algorithm 2. The algorithm is an instance of the resample-move algorithm of [28], specified for the sequence of posteriors in (5). The most common resampling step is multinomial sampling, where the N new particles are sampled independently from the categorical distribution with support points being the existing N particles and the probabilities being their weights. For the rejuvenation step, one common type of MCMC move consists of (i) an update of x_k, $k = 1, \ldots, t$, with a Metropolis-Hastings (MH) move with invariant distribution $p_{\theta,s_k}(x_k | y_k) = p_\theta(x_k) p^{\epsilon}_{s_k}(y_k | x_k)$, which is followed by (ii) an update of θ using an MH move with invariant distribution $p(\theta | x_{1:t}) \propto \eta(\theta) \prod_{k=1}^{t} p_\theta(x)$. One such MCMC move is shown in Algorithm 4 in Appendix B.

In general, the computational cost of SMC for processing n observations is $\mathcal{O}(Nn^2)$ since an $\mathcal{O}(tN)$ operation is needed to rejuvenate the particles at time t. The cost may be reduced in some cases: The cost for updating $x_{1:t}$ can be reduced by updating a random subset, of a fixed size, of x_k's at each time step t. The cost for updating θ may be reduced depending on the model specifics, for example by using a Gibbs move for θ if the posterior distribution $p(\theta|x_{1:t})$ is tractable.

Algorithm 2: SMC update at time t.

Input: Particles at time $t-1$, $(\theta_{t-1}^{(1:N)}, x_{1:t-1}^{(1:N)})$, particle weights $w_t^{(i)}$ observation y_t, function s_t, DP parameter ϵ

Output: The particle system at time t

Resampling: Resample particles according to their weights:

$$(\theta^{(1:N)}, x_{1:t-1}^{(1:N)}) \leftarrow \text{Resample}((\theta^{(1:N)}, x_{1:t-1}^{(1:N)}); w_{t-1}^{(1:N)})$$

for $i = 1, \ldots, N$ **do**

  Rejuvenation: Update $(\theta^{(i)}, x_{1:t-1}^{(i)})$ using an MCMC move that targets $p^{\epsilon}_{s_{1:t-1}}(\theta, x_{1:t-1}|y_{1:t-1})$.

  Propagation: Sample $x_t^{(i)} \sim \mathcal{P}_{\theta^{(i)}}$ and append particle i as $(\theta^{(i)}, x_{1:t}^{(i)}) = (\theta^{(i)}, (\widetilde{x}_{1:t-1}^{(i)}, x_t^{(i)}))$,

end

Weighting: Calculate $w_t^{(i)} \propto \text{Laplace}\left(y_t - s_t(x^{(i)}), \Delta s_t/\epsilon\right)$ for $i = 1, \ldots, N$ s.t. $\sum_{i=1}^N w_t^{(i)} = 1$.

3.2. Adaptive truncation for the transformation

In this section, we focus on H in Algorithm 1 and describe a method to determine the function s_t adaptively so that the estimation performance of SMC is better over a version where an arbitrary s_t is used. We confine to s_t that corresponds to truncating x_t into an interval $[l_t, r_t]$,

$$s_t(x) = T_{l_t}^{r_t}(x) := \min\{\max\{x, l_t\}, r_t\},$$

so that the sensitivity is $\Delta s_t = r_t - l_t$. We assume X_t is univariate; for multivariate X_t the truncation approach can be applied to each component.

How should we choose the truncation points l_t, r_t? Recall the trade-off mentioned earlier: A larger $r_t - l_t$ renders truncation less likely but leads to a larger noise in Y_t; whereas a smaller $r_t - l_t$ renders truncation more likely but leads to a smaller noise in Y_t. Another critical factor is the location of l_t, r_t relative to θ. For example, when θ is a location parameter, an interval (l_t, r_t) around θ may be preferred.

Following the works like [5, 6], we use the Fisher information as the amount of information that an observation carries about the population parameter. The Fisher information associated to $Y = T_l^r(X) + (r-l)V$ when $V \sim \text{Laplace}(1/\epsilon)$ can be expressed as

$$F_{l,r}^{\epsilon}(\theta) = \mathbb{E}\left[\nabla_\theta \log p_{l,r}^{\epsilon}(Y|\theta) \nabla_\theta \log p_{l,r}^{\epsilon}(Y|\theta)^T\right], \tag{6}$$

where $p_{l,r}^{\epsilon}(y|\theta)$ is the pdf of the marginal distribution of $Y = y$ given θ. According to this approach, we set the truncation points l_t, r_t to those l, r values that jointly maximize $F_{l,r}^{\epsilon}(\theta)$. When θ is multidimensional, an overall *score function* $\text{sc}(\cdot)$ can be used to order the Fisher information matrices. An example of such a score function is the trace, or a weighted (harmonic) average of the diagonals, of $F_{l,r}^{\epsilon}(\theta)$.

$F_{l,r}^{\epsilon}(\theta)$ is smaller for a smaller ϵ, due to more noisy observations. However, it is not obvious how $F_{l,r}^{\epsilon}(\theta)$ behaves with l, r. The exact calculation of $F_{l,r}^{\epsilon}(\theta)$ is not possible in general as the truncation of X between

l, r, if nothing else, introduces an intractability in the calculations. That is why we numerically approximate the Fisher information using the Monte Carlo technique proposed in [6],

$$F^{\epsilon}_{l,r}(\theta) \approx \frac{1}{M}\sum_{j=1}^{M} \nabla_\theta \log \widetilde{p^{\epsilon}_{l,r}}(y^{(j)}|\theta) \nabla_\theta \log \widetilde{p^{\epsilon}_{l,r}}(y^{(j)}|\theta)^T, \quad y^{(1)}, \ldots, y^{(M)} \overset{\text{i.i.d}}{\sim} p^{\epsilon}_{l,r}(y|\theta), \tag{7}$$

where each gradient term in the sum is calculated using Algorithm 5 in Appendix B.

3.2.1. Exploration-exploitation for interval selection

We adjust the interval $[l, r]$ to better estimate θ, and yet the adjustment is based on $F(\theta)$, which itself depends on θ. Therefore, we are adapting the intervals based on a technique that requires the knowledge of θ which we want to estimate in the first place. This situation necessitates an exploration-exploitation approach. When we have little knowledge about θ, we should let our adaptive algorithm have more freedom to locate the truncation interval; but as we learn θ by receiving more and more observations, the location of the interval should be chosen with less variety. Our exploration-exploitation approach consists of two steps. Let $\text{sc} : \mathbb{R}^{d\times d} \mapsto \mathbb{R}$ be a score function for Fisher information matrices, where d is the dimension of θ. Given Θ_t,

1. Draw ϑ randomly from the particle approximation,

$$\vartheta \sim p^{\epsilon,N}_{l_{1:t},r_{1:t}}(\theta|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{\theta^{(i)}}(\mathrm{d}\theta), \tag{8}$$

   which corresponds to setting $\vartheta = \theta^{(i)}$ w.p. $w_t^{(i)}$.
2. Determine the interval for the next observation as

$$l_{t+1}, r_{t+1} = \arg\max_{l,t} \text{sc}(F^{\epsilon}_{l,r}(\vartheta)). \tag{9}$$

After determining $[l_{t+1}, r_{t+1}]$, the next data point X_{t+1} is shared as

$$Y_{t+1} = T^{r_{t+1}}_{l_{t+1}}(X_{t+1}) + (r_{t+1} - l_{t+1})V_{t+1}, \quad V_{t+1} \sim \text{Laplace}(1/\epsilon). \tag{10}$$

The exploration size is decreased as t increases, that is, as more data are observed. Remarkably, this is automatically handled by the posterior distribution in (8), which is spread over a wide region at the beginning but gets more concentrated as more data are received.

Thompson sampling. The exploration-exploitation approach described above can be likened to Thompson sampling in reinforcement learning (see e.g., [7]), albeit with latent 'rewards': Using the terminology from reinforcement learning, in our case, the 'action' is the choice of the interval $[l_t, r_t]$, 'state' is Y_t, the 'model parameter' is θ, 'past observations' at time t are the states $Y_1, \ldots, Y_t$, and the 'objective function' is $\text{sc}(F_{l,r}(\theta))$. The exact implementation of Thompson sampling requires sampling from $p_{l_{1:t},r_{1:t}}(\mathrm{d}\theta|Y_{1:t})$. As often done in practice, we approximate that step sample from the particle approximation $p^{N}_{l_{1:t},r_{1:t}}(\mathrm{d}\theta|Y_{1:t})$.

3.2.2. Location and scale parameters and truncation

In principle, the maximization step in (9) can be applied to any population distribution $\mathcal{P}_\theta$ for sensitive data. However, location-scale distribution families deserve particular interest due to their common use and certain desirable properties. It is intuitive to suppose that the best truncation points for a location-scale distribution can be obtained simply by scaling and shifting the best truncation points calculated for some *base* distribution. We show here that this is indeed the case. For a general population distribution $\mathcal{P}_\theta$, the maximization (9) needs to be performed afresh for each ϑ_t. For location-scale families, however, the computationally intensive part of (9) can be done once for some base distribution and its result can easily be applied for all ϑ_t by scaling and shifting. Below we explain how that is possible.

Definition 2 *A distribution family* $\{f(\cdot;m,c) : (m,c) \in \mathbb{R}\times(0,\infty)\}$ *is a location-scale family with a base distribution* $g(x)$ *if for all* $(m,c) \in \mathbb{R}\times(0,\infty)$ *we have* $f(x;m,c) = \frac{1}{c}g((x-m)/c)$ *for all* $x \in \mathcal{X}$. *In particular,* $f(x;0,1) = g(x)$.

Assume that $\mathcal{P}_\theta$ is a member of a location-scale family, e.g., a normal distribution with θ being the vector of the mean and the standard deviation. When $\vartheta_t = (m,c)$ is sampled in Step 1 above, consider formalizing the truncation points as

$$l_{t+1} = ac + m, \quad r_{t+1} = bc + m, \tag{11}$$

where a and b are the free parameters. Then, the problem in (9) reduces to choosing the best a, b that maximize $\mathrm{sc}(F^\epsilon_{ac+m,bc+m}(m,c))$, where $F^\epsilon_{ac+m,bc+m}(m,c)$ is the Fisher information associated to the random variable

$$Y = T^{bc+m}_{ac+m}(X) + c(b-a)V, \quad V \sim \text{Laplace}(1/\epsilon), \quad X \sim \mathcal{P}_{(m,c)}. \tag{12}$$

We show that for location-scale families, a uniformly best pair a, b over all possible values (m,c) exists.

Theorem 3 *For any* $a, b \in \mathbb{R}$ *such that* $a < b$, $\epsilon > 0$ *and* $(m,c) \in \mathbb{R}\times[0,\infty)$, *let* $\mathrm{sc} : \mathbb{R}^{2\times 2} \mapsto \mathbb{R}$ *be a score for Fisher information matrices. Then, for all pairs* a, b *and* a', b' *such that* $a < b$ *and* $a < b$, *either one of the three holds*

$$\begin{aligned}
&\mathrm{sc}(F^\epsilon_{ac+m,bc+m}(m,c)) > \mathrm{sc}(F^\epsilon_{a'c+m,b'c+m}(m,c)), \quad \forall (m,c) \in \mathbb{R}\times(0,\infty);\\
&\mathrm{sc}(F^\epsilon_{ac+m,bc+m}(m,c)) < \mathrm{sc}(F^\epsilon_{a'c+m,b'c+m}(m,c)), \quad \forall (m,c) \in \mathbb{R}\times(0,\infty);\\
&\mathrm{sc}(F^\epsilon_{ac+m,bc+m}(m,c)) = \mathrm{sc}(F^\epsilon_{a'c+m,b'c+m}(m,c)), \quad \forall (m,c) \in \mathbb{R}\times(0,\infty).
\end{aligned}$$

A proof of Theorem 3 is given in Appendix A. Theorem 3 implies that it suffices to find

$$(a^*, b^*) = \arg\max_{a,b} \mathrm{sc}(F^\epsilon_{a,b}(0,1)), \tag{13}$$

the best a, b for the base distribution, i.e. for $(m,c) = (0,1)$. Then, it is guaranteed that those a^*, b^* are the best choices for all (m,c) values when the intervals are chosen according to (11). Therefore, maximization for interval selection needs to be done only once, implying significant computational savings.

4. Numerical results

In this section, we demonstrate the merits of our method, in comparison to its nonadaptive counterparts, both on simulated and real data sets.

4.1. Experiments on simulated data

In our experiments with simulated data, we take $\mathcal{N}(\mu, \sigma^2)$ as the population distribution, so that $\theta = (\mu, \sigma)$, and aim to estimate both μ and σ. Sensitive data $X_1, \ldots, X_n$ of length $n = 1000$ are generated from $\mu = 50$ and $\sigma^2 = 10$. The parameters are taken a priori independent with $\mu \sim \mathcal{N}(0, 10^4)$ and $\sigma^2 \sim \mathcal{IG}(1, 1)$, where $\mathcal{IG}(\alpha, \beta)$ is the inverse gamma distribution with shape α and scale β. The SMC method is combined with the exploration-exploitation strategy described in subsection 3.2, for the choice of the truncation points.

Note that $\mathcal{N}(\mu, \sigma^2)$ is a location-scale distribution with μ and σ being the location and scale parameters. Therefore, we apply (12) to generate the noisy observations, where a and b are the optimal truncation points corresponding to the $\mathcal{N}(0, 1)$. The Fisher information matrix is a 2×2 matrix, corresponding to the bivariate parameter μ, σ. For the example's sake, we consider that the primary goal is to estimate μ while σ^2 is of secondary importance. Thus, we chose the score function as the first entry of the Fisher information matrix, that is, $\mathrm{sc}(F^{\epsilon}_{a,b}(0,1)) = F^{\epsilon}_{a,b}(0,1)[1,1]$. The maximization in (13) is performed by Monte Carlo estimation of $F^{\epsilon}_{a,b}(0,1)$ on the 50×50 grid spanning $[-3,3] \times [3,3]$ of (a, b) points. The Monte Carlo estimation is performed as in (7) with $M = 1000$, where the gradient terms in (7) are approximated using Algorithm 5 with samples of size $10,000$. The best $[a, b]$ intervals were numerically found as $[-0.06, 0.06]$, $[-0.12, 0.12]$, $[-0.54, 0.54]$, and $[-0.96, 0.96]$ for $\epsilon = 1, 2, 5, 10$, respectively.

Algorithm 3 summarizes the entire course of one run of SMC with adaptive truncation for the normal distribution, which we call "SMC-adaptive". For each of $\epsilon = 1, 2, 5, 10$, we repeat this experiment 30 times independently.

Algorithm 3: SMC-adaptive: Differentially private SMC with adaptive truncation for the normal distribution.

First, find a, b that maximize $\mathrm{sc}(F^{\epsilon}_{a,b}(0,1))$ the Fisher information matrix of $Y = T^b_a(X) + (b-a)V$, when $X \sim \mathcal{N}(0,1)$, $V \sim \mathrm{Laplace}(1/\epsilon)$.
Start with, l_1, r_1.
for $t = 1, \ldots, n$ **do**
  Generate $Y_t = T^{r_t}_{l_t}(X_t) + (r_t - l_r)V_t$, where $X_t \sim \mathcal{N}(\mu, \sigma^2)$ and $V_t \sim \mathrm{Laplace}(1/\epsilon)$.
  Update the particle system of SMC using Algorithm 2 with $N = 1000$ particles to construct the SMC approximation of the posterior $p^{\epsilon}_{l_{1:t}, r_{1:t}}(\theta | Y_{1:t})$
  Sample $\vartheta = (m, c)$ from the SMC approximation $p^{\epsilon, N}_{l_{1:t}, r_{1:t}}(\theta | y_{1:t})$.
  Determine the new truncation points $l_{t+1} = m + ca$, $r_{t+1} = m + cb$.
end

We compared SMC-adaptive to two nonadaptive algorithms. The first one is the same SMC method in Algorithm 2, but with constant truncation points, $l_c = \mu - 10\sigma$ and $r_c = \mu + 10\sigma$ for all t. We call this algorithm "SMC-nonadaptive". The second method is an MCMC sampling method that targets the conditional distribution of θ given the entire batch of the observations at once, $p^{\epsilon}_{l_c, r_c}(\theta | Y_{1:n})$, where the observations are generated using the same truncation points for all X_t as in SMC-nonadaptive as

$$Y_t = T^{r_c}_{l_c}(X_t) + (r_c - l_c)V_t, \quad V_t \overset{\text{i.i.d.}}{\sim} \mathrm{Laplace}(1/\epsilon), \quad t = 1, \ldots, n. \tag{14}$$

The model for the random variables $\{\theta, X_{1:n}, Y_{1:n}\}$ is a latent variable model for independent observations. That is why as the MCMC method we chose the MHAAR algorithm proposed in [29, Section 3], which is well suited to such latent variable models. The interval $[l_c, r_c]$, chosen for the nonadaptive methods, represents the

situation in many practical applications where there is not much strong *a priori* knowledge available about θ. The comparison with the nonadaptive version of the SMC aims to show the merit of adaptive truncation. Moreover, the comparison with the MCMC method aims to show the merit of adaptation even when online estimation is not required.

Figure 1 displays the performance of the two SMC methods for a single run, for each ϵ. The scatter plots of the particles at every 20th time step as well as the mean estimates are shown. Further, the truncation points are also shown in the plots for the location parameter μ. Observe the decreasing amount of spread of the particles as t. Also, as expected, accuracy increases with ϵ. We also observe the benefit of the adaptive truncation method relative to its nonadaptive counterpart when we compare the particle distributions: the particles of the SMC algorithm with adaptive truncation get more concentrated around the true values and do that much more quickly than those of the nonadaptive version. The posterior means, shown with red lines, also show the advantage of the truncation method.

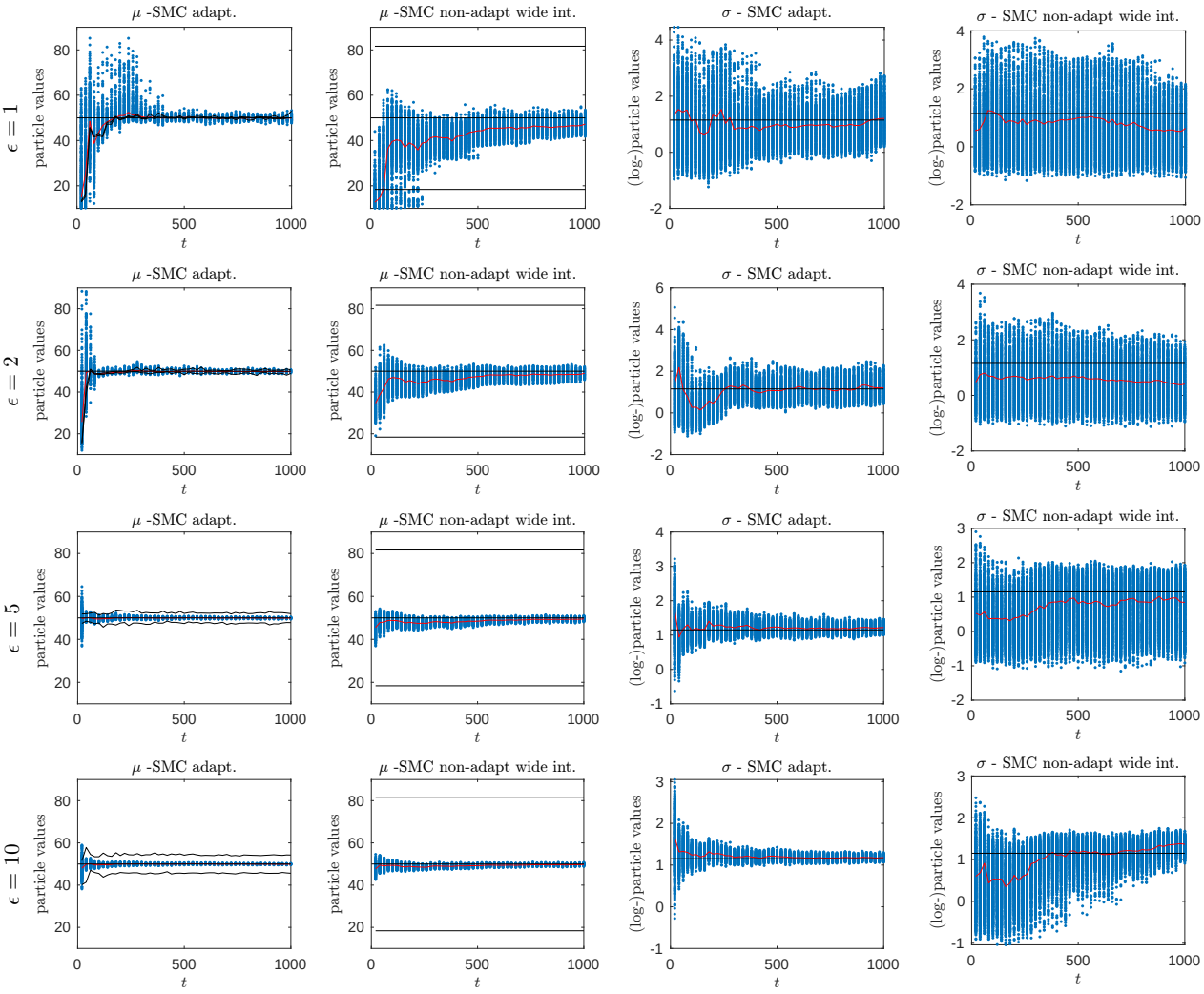

Figure 1. The particle distribution of SMC (every 20th time-step shown) (blue points) and the estimate of the posterior means (red line) versus time. Black lines indicate the true values. In plots for μ, truncation points are also shown.

While Figure 1 shows results by the SMC methods from a single run, Figure 2 shows the box plots of the mean posterior estimates of θ, obtained from 30 independent runs, of all the three methods under comparison, namely SMC-adaptive, SMC-nonadaptive, and the MCMC methods. The box plots clearly show that the adaptive truncation approach is beneficial in terms of estimation accuracy as our method beats the other two methods for both parameters and all the tried ϵ values.

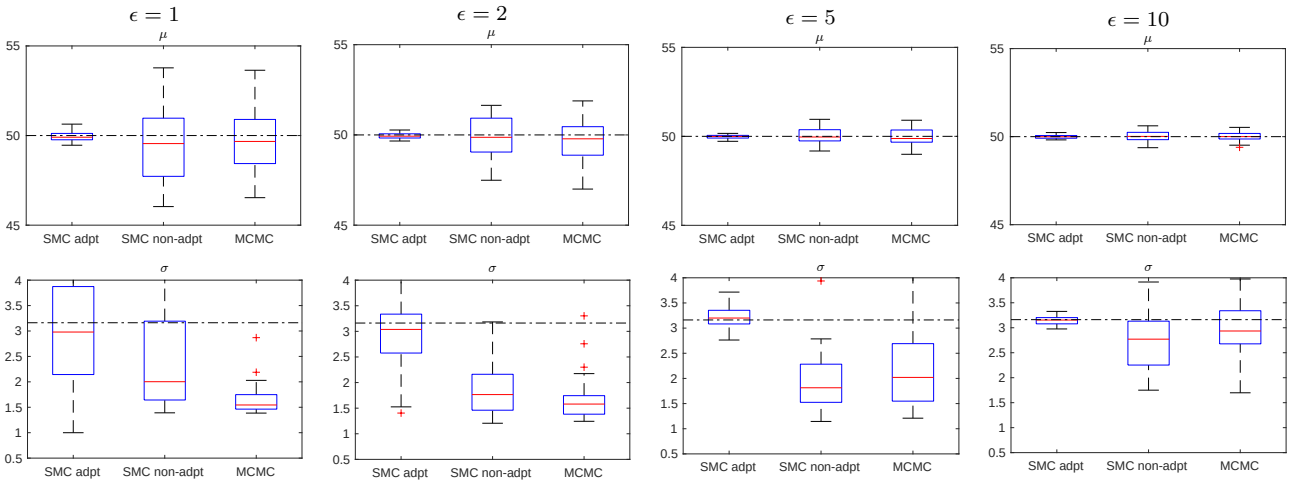

Figure 2. Box-plots of the posterior means, obtained from 30 runs.

Remark 2 *At this point, we remark that the underperformance of SMC-nonadaptive and MCMC are not due to the algorithmic inferiorities of their SMC or MCMC components. These methods do worse because they are nonadaptive, hence fed with less informative observations.*

4.2. Experiments on real data

In this part, we use a real data set that consists of annual median household incomes ($\times 10^{-3}$) of a total of 4033 small geographical units in the state of Indiana, US, for the year 2016. Hence, for this data set X_t denotes the median household income of the t'th geographical region. Removing the problematic entries, we ended up with $n = 3982$ households.

The data set fits naturally in the context of data privacy since the household income information is collected from individuals and can be considered sensitive data. The data set has a median income value for each geographical unit, typically or county or township, that contains a group of households.

As long as data privacy is concerned, it is reasonable to assume that a unit of data (corresponding to a single 'individual') is a household income. Note that the median of a group of household incomes has the same sensitivity as a single household income. That is, if the income for a household has some natural limits, those limits apply, equally tightly, to the median income of a group of households, too.

Similarly to the experiments above, we fitted a normal distribution to the sensitive data. The MLE solution based on the sensitive measurements $X_1, \ldots, X_n$ yields $\hat{\mu} = 51.78, \hat{\sigma} = 17.57$. These values are used as a benchmark for the performances of the methods that work on the privatized noisy data $Y_1, \ldots, Y_n$.

Differently than the earlier examples, this time we set the score function as

$$\mathrm{sc}(F^{\epsilon}_{a,b}(0,1)) = \left(\frac{1}{F^{\epsilon}_{a,b}(0,1)[1,1]} + \frac{1}{F^{\epsilon}_{a,b}(0,1)[2,2]} \right)^{-1},$$

the harmonic sum of the diagonals of $F^{\epsilon}_{a,b}(0,1)$. We picked this form to simulate a scenario where we want to estimate the population distribution itself, hence giving importance to both μ and σ (hence consider the Fisher information with respect to both parameters). The harmonic mean implies that the used score function somewhat accounts for the squared error in the estimates since the mean squared error is implied by the inverse of the Fisher information matrix.

As before, we compared the performance of our method SMC-adaptive (Algorithm 3) to the nonadaptive algorithms SMC-nonadaptive and MCMC, where the observations are generated as in (14). The constant truncation points for the nonadaptive algorithms are taken $l_c = \mu - 5\sigma$ and $r_c = \mu + 5\sigma$ for all t.

Figure 3 shows the results for a single run for SMC-adaptive and SMC-nonadaptive, which enable similar conclusions as we drew from simulated data examples. The nonadaptive methods do well for μ while our SMC-adaptive does well both for μ and σ^2. Overall, SMC-adaptive is more capable of estimating μ, σ^2 jointly than its nonadaptive competitors.

Another interesting observation here can be made from the truncation intervals for SMC-adaptive, shown by the wrinkled lines in the left-most plot in Figure 3. Note that the truncation points are not around the mean, in fact, they hardly overlap with the region where the particles concentrate. This is because those truncation points are determined such that the resulting observations are informative about σ^2 as well as μ.

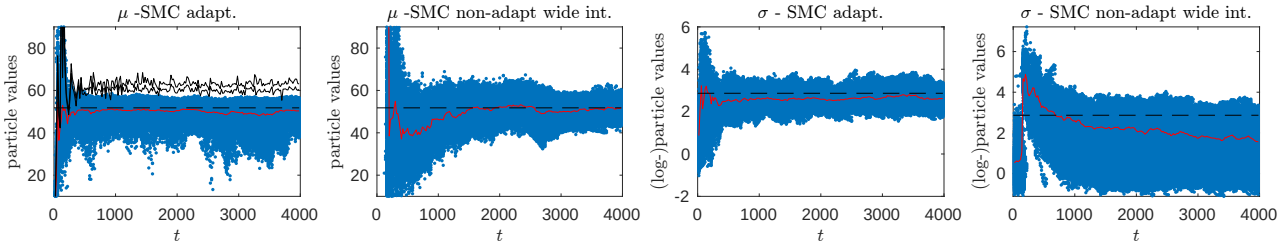

Figure 3. The particle distribution of SMC-adapt and SMC-nonadapt vs. observation number t for real data.

We performed 5 Monte Carlo runs on the real data, each time independently randomly permuting the sensitive data (so that the values are observed in a different order each time) and adding independent DP noise. The estimation results across those runs, in terms of posterior means for μ and σ, are shown in Table 1. The table includes the estimation results obtained from the MCMC method as well. As we can see, SMC-adaptive is not only stable but also the most accurate among the three methods. SMC-nonadaptive is the worst in terms of stability. MCMC also estimates μ fairly well but fails to estimate the σ with reasonable accuracy.

5. Conclusion

This paper presents a novel methodology for differentially private online Bayesian estimation with adaptive truncation. The proposed methodology is a working example of the general idea that, as we gain knowledge about the process that generates sensitive data, we can modify our 'query' about sensitive data to get more utility while maintaining the same level of privacy. The proposed method demonstrated its merits in the

Table 1. Posterior means across 5 Monte Carlo runs on random permutations of the same sensitive data.

Run no	μ (MLE: $\hat{\mu} = 51.78$)			σ (MLE: $\hat{\sigma} = 17.57$)		
	SMC-adaptive	SMC-nonadaptive	MCMC	SMC-adaptive	SMC-nonadaptive	MCMC
1	50.3756	51.8790	49.6149	13.6397	4.8232	2.1333
2	55.0282	535.2168	46.7601	9.0344	1.6814	1.6139
3	53.5980	48.4388	49.3032	10.2802	2.6478	1.4627
4	46.6303	51.8498	51.6290	20.5239	2.0033	3.6181
5	48.3601	51.4870	53.4291	15.7926	2.4522	1.6269

numerical experiments involving the normal distribution, one of the most commonly used distributions for modeling univariate i.i.d. data. It would be interesting to see the extension of the work to other distributions, especially multivariate distributions.

Although we considered the Laplace mechanism throughout, the methodology can be modified straightforwardly for other privacy mechanisms, such as the Gaussian mechanism, that provide different senses of privacy. All that changes throughout is the conditional distribution of Y_t given $x_t, \theta, s_t, \epsilon$.

We considered Bayesian inference in this work. Bayesian inference fits ideally into the exploration-exploitation framework by providing a proper sense of uncertainty about θ via the posterior distribution. However, its computational cost that grows quadratically with data size can be a concern when n is very large. A viable alternative is finding the maximum likelihood estimate of θ using an online gradient method, where the gradients can be calculated approximately using Monte Carlo as in Algorithm 5. The online gradient method can be advantageous in terms of computational load but it would be more challenging to tune the exploration-exploitation heuristic since a posterior distribution of θ would not be available.

Acknowledgment

This study was funded by The Scientific and Technological Research Council of Türkiye (TÜBİTAK) ARDEB through grant no 120E534. The author was supported by this grant.

References

[1] Dwork C. Differential privacy. In: Bugliesi M, Preneel B, Sassone V, Wegener I, editors, Automata, Languages and Programming; Berlin, Heidelberg. Springer Berlin Heidelberg; 2006. pp. 1-12.

[2] Dwork C, Roth A. The algorithmic foundations of differential privacy. Theoretical Computer Science 2013; 9 (3-4):211-407.

[3] Kasiviswanathan SP, Lee HK, Nissim K, Raskhodnikova S, Smith A. What can we learn privately? SIAM Journal on Computing 2011; 40 (3):793-826

[4] Chopin N. A sequential particle filter method for static models. Biometrika 2002; 89 (3):539-551

[5] Barnes LP, Chen WN, Özgür A. Fisher information under local differential privacy. IEEE Journal on Selected Areas in Information Theory 2020; 1 (3):645-659.

[6] Alparslan B, Yıldırım S. Statistic selection and MCMC for differentially private Bayesian estimation. Statistics and Computing 2022; 32 (5):66.

[7] Russo DJ, Roy BV, Kazerouni A, Osband I, Wen Z. A tutorial on Thompson sampling. Foundations and Trends in Machine Learning 2018; 11 (1):1-96.

[8] Dwork C. Differential privacy: A survey of results. In: Agrawal M, Du D, Duan Z, Li A, editors, Theory and Applications of Models of Computation; Berlin, Heidelberg. Springer Berlin Heidelberg; 2008. pp. 1-19.

[9] Dong J, Roth A, Su WJ. Gaussian differential privacy. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 2022; 84 (1):3-37.

[10] Bun M, Steinke T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Proceedings, Part I, of the 14th International Conference on Theory of Cryptography-Volume 9985; New York, NY, USA. Springer-Verlag New York, Inc.; 2016: 635-658.

[11] Wang YX, Fienberg S, Smola A. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In: Blei D, Bach F, editors, Proceedings of the 32nd International Conference on Machine Learning (ICML-15). JMLR Workshop and Conference Proceedings; 2015:2493-2502.

[12] Li B, Chen C, Liu H, Carin L. On connecting stochastic gradient MCMC and differential privacy. In: Chaudhuri K, Sugiyama M, editors, Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics; volume 89 of Proceedings of Machine Learning Research. PMLR; 2019: 557-566.

[13] Heikkilä M, Jälkö J, Dikmen O, Honkela A. Differentially private Markov chain Monte Carlo. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, editors, Advances in Neural Information Processing Systems; volume 32. Curran Associates, Inc.; 2019.

[14] Yıldırım S, Ermiş B. Exact MCMC with differentially private moves. Statistics and Computing 2019; 29 (5):947-963.

[15] Räisä O, Koskela A, Honkela A. Differentially Private Hamiltonian Monte Carlo. In: NeurIPS 2021 Workshop Privacy in Machine Learning; 2021.

[16] Foulds J, Geumlek J, Welling M, Chaudhuri K. On the theory and practice of privacy-preserving Bayesian data analysis. In: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence; UAI'16; Arlington, Virginia, USA. AUAI Press; 2016:192-201.

[17] Williams O, Mcsherry F. Probabilistic inference and differential privacy. In: Lafferty J, Williams C, Shawe-Taylor J, Zemel R, Culotta A, editors, Advances in Neural Information Processing Systems; volume 23. Curran Associates, Inc.; 2010.

[18] Karwa V, Slavković AB, Krivitsky P. Differentially private exponential random graphs. In: Domingo-Ferrer J, editor, Privacy in Statistical Databases; Cham. Springer International Publishing; 2014. pp. 143-155.

[19] Bernstein G, Sheldon DR. Differentially private Bayesian inference for exponential families. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, editors, Advances in Neural Information Processing Systems; volume 31. Curran Associates, Inc.; 2018.

[20] Park M, Vinaroz M, Jitkrittum W. ABCDP: Approximate Bayesian computation with differential privacy. Entropy 2021; 23 (8).

[21] Gong R. Exact inference with approximate computation for differentially private data via perturbations. Journal of Privacy and Confidentiality 2022; 12 (2).

[22] Dwork C, Naor M, Pitassi T, Rothblum GN. Differential privacy under continual observation. In: Proceedings of the Forty-Second ACM Symposium on Theory of Computing; STOC '10; New York, NY, USA. Association for Computing Machinery; 2010: 715-724.

[23] Chan T, Shi E, Song D. Private and continual release of statistics. ACM Transactions on Information and System Security 2011; 14 (3).

[24] Cao Y, Yoshikawa M, Xiao Y, Xiong L. Quantifying differential privacy under temporal correlations. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE); 2017. pp. 821-832.

[25] Wang T, Blocki J, Li N, Jha S. Locally differentially private protocols for frequency estimation. In: Proceedings of the 26th USENIX Conference on Security Symposium; SEC'17; USA. USENIX Association; 2017: 729-745.

[26] Steinberger L. Efficiency in local differential privacy. arXiv preprint arXiv:2301.10600 2023.

[27] Lopuhaä-Zwakenberg M, Boris S, Li N. Fisher information as a utility metric for frequency estimation under local differential privacy. In: Proceedings of the 21st Workshop on Privacy in the Electronic Society; WPES'22; New York, NY, USA. Association for Computing Machinery; 2022: 41-53.

[28] Gilks WR, Berzuini C. Following a moving target-Monte Carlo inference for dynamic Bayesian models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 2001; 63 (1):127-146.

[29] Andrieu C, Yıldırım S, Doucet A, Chopin N. Metropolis-Hastings with averaged acceptance ratios. arXiv:2101.01253 2020.

[30] Jones MC, Noufaily A. Log-location-scale-log-concave distributions for survival and reliability analysis. Electronic Journal of Statistics 2015; 9 (2):2732-2750.

[31] Geweke J. Bayesian inference in econometric models using Monte Carlo integration. Econometrica 1989; 57 (6):1317-1339.

A. Proof of Theorem 3

We prove the theorem by first showing that the distribution of Y in (12) belongs to a location-scale family. Lemmas 1 and 2 are used for establishing that. Given a distribution with density f on $\mathbb{R}$ and an interval $[l,r]$, we define $f_{[l,r]}$ to be f truncated to $[l,r]$, that is $f_{[l,r]}(x) \propto f(x)\mathbb{I}\{x \in [l,r]\}$. Lemma 1 states that the truncated version of a location-scale distribution is also a location-scale distribution.

Lemma 1 *Let $\{f(x;m,c); m,c) \in \mathbb{R}\times[0,\infty)\}$ be a location-scale family of distributions with location parameter m, scale parameter c, and base distribution g. For any a,b such that $a<b$; the family of truncated distributions $\{f_{[ca+m,cb+m]}(x;m,c) : (m,c) \in \mathbb{R}\times[0,\infty)\}$ is a location-scale family with location parameter m, scale parameter c, and base distribution $g_{[a,b]}$.*

Proof (Lemma 1) For all $x \in \mathcal{X}$, $a,b \in \mathbb{R}$ such that $a<b$, $(m,c) \in \mathbb{R}\times[0,\infty)$, we have

$$f_{[ca+m,cb+m]}(x;m,c) = \frac{f(x;m,c)\mathbb{I}\{x \in [ca+m, cb+m]\}}{\int_{ca+m}^{cb+m} f(u;m,c)du}.$$

Using $f(x;m,c) = g((x-m)/c)/c$, $\mathbb{I}\{x \in [ca+m,cb+m]\} = \mathbb{I}\{(x-m)/c \in [a,b]\}$, and $\int_{ca+m}^{cb+m} \frac{1}{c} g((u-m)/c)du = \int_a^b g(u)du$ by change of variables, we end up with

$$f_{[ca+m,cb+m]}(x;m,c) = \frac{g((x-m)/c)/c\mathbb{I}\{(x-m)/c \in [a,b]\}}{\int_a^b g(u)du} = \frac{1}{c} g_{[a,b]}((x-m)/c).$$

Hence, $f_{[ca+m,cb+m]}(x;m,c)$ is a location-scale distribution with location m, scale c, base distribution $g_{[a,b]}$. □

Let $f^{\epsilon}_{[ca+m,cb+m]}(x;m,c)$ be the distribution of Y defined in (12). Let $g^{\epsilon}_{a,b}$ be the distribution of $X_0+(b-a)V_0$ where $X_0 \sim g_{[a,b]}$, $V_0 \sim$ Laplace$(1/\epsilon)$, and assume that X_0 and V_0 are independent.

Lemma 2 *Given $a,b \in \mathbb{R}$ such that $a<b$ and $\epsilon \in (0,\infty)$, the distribution family $\{f^{\epsilon}_{[ca+m,cb+m]}(x;m,c) : m \in \mathbb{R}, c \in [0,\infty)\}$ is a location-scale family with location m, scale c and base distribution $g^{\epsilon}_{a,b}$.*

Proof (Lemma 2) By Lemma 1, the distribution of $T^{bc+m}_{ac+m}(X)$ is a location-scale distribution with location m, scale c, and base distribution $g_{a,b}$. For Y in (12), it can be checked that $Y = cY_0+m$ where $Y_0 = X_0+(b-a)V_0$, where $X_0 = (T^{bc+m}_{ac+m}(X)-m)/c \sim g_{a,b}$, $V_0 \sim$ Laplace$(1/\epsilon)$ and X_0 and V_0 are independent. Then, $Y_0 \sim g^{\epsilon}_{a,b}$, which does not depend on m and c. Hence we conclude. □

Finally, we proceed to the proof of Theorem 3.

Proof (Theorem 3) Since the distribution of Y is a location-scale distribution by Lemma 2, the Fisher information associated to it is given by $F^{\epsilon}_{ac+m,bc+m}(m,c) = \frac{1}{c^2} F^{\epsilon}_{a,b}(0,1)$, where $F^{\epsilon}_{a,b}(0,1)$ is the Fisher information matrix associated with the base distribution $g^{\epsilon}_{a,b}(x)$ (for explicit formulae, see, e.g., [30]), and depends on a, b, and ϵ, but not on m and c. Therefore, if $\mathrm{sc}(F^{\epsilon}_{a,b}(0,1)) > \mathrm{sc}(F^{\epsilon}_{a',b'}(0,1))$ (resp. $<$, $=$), then $\mathrm{sc}(F^{\epsilon}_{a,b}(m,c)) > \mathrm{sc}(F^{\epsilon}_{a',b'}(m,c))$ (resp. $<$, $=$) for any other $(m,c) \in \mathbb{R}\times(0,\infty)$. □

B. Supplementary algorithms

Algorithm 4 presents an MCMC algorithm for the rejuvenation step at time t of the SMC algorithm.

Algorithm 4: MCMC for $p^{\epsilon}_{s_{1:t}}(\theta, x_{1:t}|y_{1:t})$ - a single update.

Input: The current sample $(x_{1:t}, \theta)$, proposal distributions $q(x'|x)$ and $q(\theta'|\theta)$, ϵ
Output: The new sample
MH update for $x_{1:t}$**:**
for $k = 1 : t$ **do**
  Sample $x'_k \sim q(x'_k|x_k)$ and return x'_k as the new sample w.p.

$$\min\left\{1, \frac{p_\theta(x'_k)\text{Laplace}(y_k - s_k(x'_k), \Delta s_k/\epsilon)q(x_k|x'_k)}{p_\theta(x_k)\text{Laplace}(y_k - s_k(x_k), \Delta s_k/\epsilon)q(x'_k|x_k)}\right\};$$

  otherwise return x_k as the new sample.
end
MH update for θ**:** Sample $\theta' \sim q(\theta'|\theta)$ and return θ' as the new sample w.p. $\min\left\{1, \frac{q(\theta|\theta')\eta(\theta')\prod_{k=1}^{t} p_{\theta'}(x_k)}{q(\theta'|\theta)\eta(\theta)\prod_{k=1}^{t} p_\theta(x_k)}\right\}$; otherwise, return θ as the new sample.

Algorithm 5 approximates Fisher's identity for the score vector, $\nabla_\theta \log p^{\epsilon}_{\theta,s}(y) = \int \nabla \log p_\theta(x) p^{\epsilon}_{\theta,s}(x|y)\mathrm{d}x$, by using self-normalised importance sampling [31] with a sample of size N drawn from $\mathcal{P}_\theta$.

Algorithm 5: Monte Carlo calculation of the gradient.

Input: Parameter θ, observation $\underset{\sim}{y}$, DP parameter ϵ, truncation points l, r
Output: Gradient vector $\nabla_\theta \log p^{\epsilon}_{l,r}(y|\theta)$
for $i = 1, \ldots, N$ **do**
  Sample $x^{(i)} \sim \mathcal{P}_\theta$,
  Calculate $w^{(i)} = \text{Laplace}(y_k - T^r_l(x^{(i)}), \Delta s_k/\epsilon)$.
end
Calculate the (approximate) gradient as $\widetilde{\nabla_\theta \log p^{\epsilon}_{l,r}}(y|\theta) = \frac{\sum_{i=1}^{N} w^{(i)} \nabla_\theta \log p_\theta(x^{(i)})}{\sum_{i=1}^{N} w^{(i)}}$.