Master's thesis
Master's Programme in Data Science

Differentially Private Markov Chain Monte Carlo

Ossi Räisä

October 15, 2020

Supervisor(s): Associate Professor Antti Honkela

Examiner(s): Associate Professor Antti Honkela
Dr. Antti Koskela

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Ossi Räisä

Työn nimi — Arbetets titel — Title

Differentially Private Markov Chain Monte Carlo

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	October 15, 2020	28

Tiivistelmä — Referat — Abstract

ACM Computing Classification System (CCS):

Avainsanat — Nyckelord — Keywords

Differential Privacy, Markov Chain Monte Carlo

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

2. Background

2.1 Differential Privacy

Differential privacy [DR14] is a property of an algorithm that quantifies the amount of information about private data an adversary can gain from the publication of the algorithm's output. The most commonly used definition uses two real numbers, ϵ and δ, to quantify the information gain, or, from the perspective of a data subject, the privacy loss of the algorithm.

The most common definition is called (ϵ, δ)-DP, approximate DP or ADP [DR14]. The case where $\delta = 0$ is called ϵ-DP or pure DP.

Definition 1. *An algorithm $\mathcal{M} \colon \mathcal{X} \to \mathcal{U}$ is (ϵ, δ)-ADP if for all neighbouring inputs $x \in \mathcal{X}$ and $x' \in \mathcal{X}$ and all measurable sets $S \subset \mathcal{U}$*

$$P(\mathcal{M}(x) \in S) \leq e^{\epsilon} P(\mathcal{M}(x') \in S) + \delta.$$

The neighbourhood relation in the definition is domain specific. With tabular data the most common definitions are the add/remove neighbourhood and substitute neighbourhood.

Definition 2. *Two tabular datasets are said to be add/remove neighbours if they are equal after adding or removing at most one row to or from one of them. The datasets are said to be in substitute neighbours if they are equal after changing at most one row in one of them.*

The neighbourhood relation is denoted by $\sim$. The definitions and theorems of this section are valid for all neighbourhood relations.

There many other definitions of differential privacy that are mostly used to compute (ϵ, δ)-bounds for ADP. This thesis uses two of them: Rényi-DP (RDP) [Mir17] and zero-concentrated differential privacy (zCDP) [BS16]. Both are based on Rényi divergence [Mir17], which is a particular way of measuring the difference between random variables.

Definition 3. *For random variables with density or probability mass functions P and Q the Rényi divergence of order $1 < \alpha < \infty$ is*

$$D_\alpha(P \,||\, Q) = \frac{1}{\alpha - 1} \ln E_{x \sim Q} \left(\frac{P(x)^\alpha}{Q(y)^\alpha} \right).$$

Orders $\alpha = 1$ and $\alpha = \infty$ are defined by continuity:

$$D_1(P \,||\, Q) = \lim_{\alpha \to 1-} D_\alpha(P \,||\, Q),$$

$$D_\infty(P \,||\, Q) = \lim_{\alpha \to \infty} D_\alpha(P \,||\, Q).$$

Both Rényi-DP and zCDP can be expressed as bounds on the Rényi divergence between the outputs of an algorithm with neighbouring inputs:

Definition 4. *An algorithm $\mathcal{M}$ is (α, ϵ)-Rényi DP if for all $x \sim x'$*

$$D_\alpha(\mathcal{M}(x) \,||\, \mathcal{M}(x')) \leq \epsilon.$$

$\mathcal{M}$ is ρ-zCDP if for all $\alpha > 1$ and all $x \sim x'$

$$D_\alpha(\mathcal{M}(x) \,||\, \mathcal{M}(x')) \leq \rho\alpha.$$

Rényi-DP and zCDP bounds can be converted to ADP bounds [Mir17, BS16]:

Theorem 1. *If $\mathcal{M}$ is (α, ϵ)-RDP, $\mathcal{M}$ is also $(\epsilon - \frac{\ln \delta}{\alpha - 1}, \delta)$-ADP for any $0 < \delta < 1$. If $\mathcal{M}$ is ρ-zCDP, $\mathcal{M}$ is also $(\rho + \sqrt{-4\rho \ln \delta}, \delta)$-ADP for any $0 < \delta < 1$.*

A very useful property of all of these definitions is composition [DR14]: if algorithms $\mathcal{M}$ and $\mathcal{M}'$ are DP, the algorithm first computing $\mathcal{M}$ and then $\mathcal{M}'$, outputting both results, is also DP, although with worse bounds. More precisely

Definition 5. *Let $\mathcal{M} \colon \mathcal{X} \to \mathcal{U}$ and $\mathcal{M}' \colon \mathcal{X} \times \mathcal{U} \to \mathcal{U}'$ be algorithms. Their composition is the algorithm outputting $(\mathcal{M}(x), \mathcal{M}'(x, \mathcal{M}(x)))$ for input x.*

Theorem 2. *Let $\mathcal{M} \colon \mathcal{X} \to \mathcal{U}$ and $\mathcal{M} \colon \mathcal{X} \times \mathcal{U} \to \mathcal{U}'$ be algorithms. Then*

1. *If $\mathcal{M}$ is (ϵ, δ)-ADP and $\mathcal{M}'$ is (ϵ', δ')-ADP, then their composition is $(\epsilon + \epsilon', \delta + \delta')$-ADP [DR14]*

2. *If $\mathcal{M}$ is (α, ϵ)-RDP and $\mathcal{M}'$ is (α, ϵ')-RDP, then their composition is $(\alpha, \epsilon + \epsilon')$-RDP [Mir17]*

3. *If $\mathcal{M}$ is ρ-zCDP and $\mathcal{M}'$ is ρ'-zCDP, then their composition is $(\rho + \rho')$-zCDP [BS16]*

All of the composition results can be extended to any number of compositions by induction. Note that any step of the composition can depend on the results of the previous steps, not only on the private data. There are also other composition theorems for ADP that trade increased δ for decreased ϵ or vice-versa, but this thesis does not apply them directly.

As any algorithm that does not use private data in any way is $(0, 0)$-ADP, 0-zCDP and $(\alpha, 0)$-RDP with all α, Theorem 2 has the following corollary, called post-processing immunity:

Theorem 3. *Let $\mathcal{M}\colon \mathcal{X} \to \mathcal{U}$ be an ADP, RDP or zCDP algorithm with some privacy parameters. Let $f\colon \mathcal{U} \to \mathcal{U}'$ be any algorithm not using the private data. Then the composition of $\mathcal{M}$ and f is ADP, RDP or zCDP with the same privacy parameters.*

There are many different DP algorithms that are commonly used, which are also called mechanisms [DR14]. This thesis only requires one of the most commonly used ones: the Gaussian mechanism [DR14].

Definition 6. *The Gaussian mechanism with parameter σ^2 is an algorithm that, with input x, outputs a sample from $\mathcal{N}(x, \sigma^2)$, where $\mathcal{N}$ denotes the normal distribution.*

The RDP and zCDP bounds for the Gaussian mechanism are quite simple. The ADP bound is more complicated:

Theorem 4. *If for all inputs x and x', $||x - x'||_2 \leq \Delta$, the Gaussian mechanism is*

1. *$(\alpha, \frac{\alpha\Delta^2}{2\sigma^2})$-RDP [Mir17]*
2. *$\frac{\Delta^2}{2\sigma^2}$-zCDP [BS16]*
3. *n compositions of the Gaussian mechanism are $(\epsilon, \delta(\epsilon))$-ADP [SMM19] with*

$$\delta(\epsilon) = \frac{1}{2}\left(\operatorname{erfc}\left(\frac{\sigma(\epsilon - n\mu)}{\sqrt{2n}\Delta}\right) - e^{\epsilon}\operatorname{erfc}\left(\frac{\sigma(\epsilon + n\mu)}{\sqrt{2n}\Delta}\right)\right),$$

where $\mu = \frac{\Delta^2}{2\sigma^2}$ and erfc is the complementary error function.

The most common use case for the Gaussian mechanism is computing a function $f\colon \mathcal{X} \to \mathbb{R}$ of private data and feeding the result into the Gaussian mechanism to privately release the function value. The condition that the inputs of the Gaussian mechanism cannot vary too much leads into the concept of sensitivity of a function

Definition 7. *The l_p-sensitivity Δ_p, with neighbourhood relation $\sim$, of a function $f\colon \mathcal{X} \to \mathbb{R}^n$ is*

$$\Delta_p f = \sup_{x \sim x'} ||f(x) - f(x')||_p.$$

Theorem 4 implies that the value of any function with finite l_2-sensitivity can be privately released using the Gaussian mechanism with appropriate noise variance σ^2. Of course, the usefulness of the released value depends on the magnitude of σ^2 compared to the actual value.

2.2 Bayesian Inference and Markov Chain Monte Carlo

In Bayesian inference, the parameters of a statistical model are inferred from observed data using Bayes' theorem [GCS+14]. The result is not just a point estimate of the parameters, but a probability distribution describing the likelihood of different values of the parameters.

Bayes' theorem relates the *posterior* belief of the parameters $p(\theta \mid D)$ to the *prior* belief $p(\theta)$ through the observed data D and the likelihood of the data $p(D \mid \theta)$ as follows:

$$p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{\int p(D \mid \theta)p(\theta)d\theta}.$$

It is theoretically possible to compute $p(\theta \mid D)$ given any likelihood, prior and data, but the integral in the denominator is in many cases difficult to compute [GCS+14]. In such cases the posterior cannot be feasibly computed. However, many of the commonly used summary statistics of the posterior, such as the mean, variance and credible intervals, can be approximated from a sample of the posterior. *Markov chain Monte Carlo* (MCMC) is a widely used algorithm to obtain such samples [GCS+14].

MCMC algorithms sequentially sample values of θ with the goal of eventually having the chain of sampled values converge to a given distribution [GCS+14]. While this can be done in many ways, this thesis focuses on a particular MCMC algorithm: *Metropolis-Hastings* (MH).

The Metropolis-Hastings algorithm samples from a distribution π of θ_i by first picking a proposal θ^* from a proposal distribution $q(\theta_{i-1})$ at iteration i [GCS+14]. A density ratio is calculated

$$r = \frac{\pi(\theta^*)}{\pi(\theta_{i-1})} \frac{q(\theta_{i-1} \mid \theta^*)}{q(\theta^* \mid \theta_{i-1})},$$

and the proposal is accepted with probability $\min\{1, r\}$. If the proposal is accepted, $\theta_i = \theta^*$, otherwise $\theta_i = \theta_{i-1}$.

It can be shown that, with a suitable proposal distribution, the chain of θ_i values converges to π [GCS+14]. The Gaussian distribution centered at the current value is a commonly used proposal.

When MCMC is used in Bayesian inference, the distribution to approximate is

$$\pi(\theta) = p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{\int p(D \mid \theta)p(\theta)d\theta}.$$

The difficult integral $\int p(D \mid \theta)p(\theta)d\theta$ in the denominator cancels out when computing r, so only the likelihood and the prior are needed. For numerical stability, r is usually computed in log space, which makes the acceptance probability $\min\{1, e^{\lambda}\}$ where

$$\lambda = \ln \frac{p(D \mid \theta^*)}{p(D \mid \theta_{i-1})} + \ln \frac{p(\theta^*)}{p(\theta_{i-1})} + \ln \frac{q(\theta_{i-1} \mid \theta^*)}{q(\theta^* \mid \theta_{i-1})}.$$

The dataset D is typically a table with n independent rows. The likelihood is given as $p(D_j \mid \theta)$ for row D_j. Independence of the rows means that

$$p(D \mid \theta) = \prod_{j=1}^{k} p(D_j \mid \theta)$$

which means that the log likelihood ratio term of λ is

$$\ln \frac{p(D \mid \theta^*)}{p(D \mid \theta_{i-1})} = \sum_{j=1}^{n} \ln \frac{p(D_j \mid \theta^*)}{p(D_j \mid \theta_{i-1})}$$

Algorithm 1 puts all of this together to summarise the MH algorithm used for Bayesian inference.

Algorithm 1: Metropolis-Hastings: number of iterations k, proposal distribution q and initial value θ_0 and dataset D as input

for $1 \leq i \leq k$ **do**
  sample $\theta^* \sim q(\theta_{i-1})$
  $\ln p(D \mid \theta) = \sum_{j=1}^{n}(\ln p(D_j \mid \theta^*) - \ln p(D_j \mid \theta_{i-1}))$
  $\lambda = \ln p(D \mid \theta) + \ln p(\theta^*) - \ln p(\theta_{i-1}) + \ln q(\theta_{i-1} \mid \theta^*) - \ln q(\theta^* \mid \theta_{i-1})$
  $\theta_i = \begin{cases} \theta^* & \text{with probability } \min\{1, e^{\lambda}\} \\ \theta_{i-1} & \text{otherwise} \end{cases}$
end
return $(\theta_1, \ldots, \theta_k)$

2.3 The Banana Distribution

The banana distribution [TPK14] is a banana-shaped probability distribution that is a challenging target for MCMC algorithms. For this reason it has been used to test MCMC algorithms in the literature [TPK14].

Definition 8. *Let X have a bivariate Gaussian distribution with mean μ and covariance matrix Σ. Let*

$$g(x) = (x_1, x_2 - a(x_1 - m)^2 - b),$$

with $a, b, m \in \mathbb{R}$. *The banana distribution with parameters* μ, Σ, a, b *and* m *is the distribution of* $g(X)$. *It is denoted by* $\mathrm{Ban}(\mu, \Sigma, a, b, m)$.

In the literature, the banana distribution is simply used as the target to sample from, and is not the posterior in a Bayesian inference problem [TPK14]. To test differentially private MCMC algorithms, the target distribution must be the posterior of some inference problem, as otherwise there is no data to protect with differential privacy. Theorem 5 gives a suitable inference problem for testing DP MCMC algorithms.

Theorem 5. *Let*

$$\begin{aligned} \theta = (\theta_1, \theta_2) &\sim \mathrm{Ban}(0, \mathrm{diag}(\sigma_0^2, \sigma_0^2), a, b, m) \\ X_1 &\sim \mathcal{N}(\theta_1, \sigma_1^2) \\ X_2 &\sim \mathcal{N}(\theta_2 + a(\theta_1 - m)^2 + b, \sigma_2^2). \end{aligned}$$

Given data $x_1, x_2 \in \mathbb{R}^n$ *and denoting* $\tau_i = \frac{1}{\sigma_i^2}$, *the posterior of* θ *tempered with* T *is the banana distribution* $\mathrm{Ban}(\mu, \Sigma, a, b, m)$ *with*

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ji} \quad i \in \{1, 2\}$$

$$\mu = \left(\frac{Tn\tau_1\bar{x}_1}{Tn\tau_1 + \tau_0}, \frac{Tn\tau_2\bar{x}_2}{Tn\tau_2 + \tau_0}\right),$$

$$\Sigma = \mathrm{diag}\left(\frac{1}{Tn\tau_1 + \tau_0}, \frac{1}{Tn\tau_2 + \tau_0}\right).$$

Proof. Because

$$g^{-1}(y) = (y_1, y_2 + a(y_1 - m)^2 + b)$$

and the Jacobian determinant of g^{-1} is 1, for a positive-definite Σ the banana distribution has density

$$\frac{1}{2\pi\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(g^{-1}(x) - \mu)^T \Sigma^{-1} (g^{-1}(x) - \mu)\right)$$

With $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2)$ the density is

$$\frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\left(\left(\frac{x_1 - \mu_1}{\sigma_1}\right)^2 + \left(\frac{x_2 + a(x_1 - m)^2 + b - \mu_2}{\sigma_2}\right)^2\right)\right)$$

Denote $u = \theta_2 + a(\theta_1 - m)^2 + b$. The tempered posterior of θ is

$$
\begin{aligned}
p(\theta \mid X) &\propto p(X \mid \theta)^T p(\theta) \\
&= p(X_1 \mid \theta_1)^T p(X_2 \mid \theta_1, \theta_2)^T p(\theta) \\
&\propto \left(\prod_{i=1}^n \exp\left(-\frac{(x_{i1}-\theta_1)^2\tau_1}{2}\right)\right)^T \cdot \left(\prod_{i=1}^n \exp\left(-\frac{(x_{i2}-\theta_2-a(\theta_1-m)^2-b)^2\tau_2}{2}\right)\right)^T \\
&\cdot \exp\left(-\frac{1}{2}\left(\tau_0\theta_1^2 + \tau_0(\theta_2 + a(\theta_1-m)^2+b)^2\right)\right) \\
&= \exp\left(-\frac{1}{2}\Big(T\tau_1 \sum_{i=1}^n (x_{i1}-\theta_1)^2 + T\tau_2 \sum_{i=1}^n (x_{i2}-u)^2 + \tau_0\theta_1^2 + \tau_0 u^2\Big)\right) \\
&= \exp\left(-\frac{1}{2}\Big(T\tau_1 \sum_{i=1}^n (x_{i1}-\bar{x}_1)^2 + T\tau_1 n(\bar{x}_1-\theta_1)^2\right. \\
&\left. + T\tau_2 \sum_{i=1}^n (x_{i2}-\bar{x}_2)^2 + T\tau_2 n(\bar{x}_2-u)^2 + \tau_0\theta_1^2 + \tau_0 u^2\Big)\right) \\
&\propto \exp\left(-\frac{1}{2}\Big(T\tau_1 n(\bar{x}_1-\theta_1)^2 + T\tau_2 n(\bar{x}_2-u)^2 + \tau_0\theta_1^2 + \tau_0 u^2\Big)\right) \\
&= \exp\left(-\frac{1}{2}\Big(T\tau_1 n\bar{x}_1^2 - 2T\tau_1 n\bar{x}_1\theta_1 + nT\tau_1\theta_1^2 + \tau_0\theta_1^2\right. \\
&\left. + T\tau_2 n\bar{x}_2^2 - 2T\tau_2 n\bar{x}_2 u + nT\tau_2 u^2 + \tau_0 u^2\Big)\right) \\
&\propto \exp\left(-\frac{1}{2}\Big((Tn\tau_1+\tau_0)\theta_1^2 - 2T\tau_1 n\bar{x}_1\theta_1 + (Tn\tau_2+\tau_0)u^2 - 2T\tau_2 n\bar{x}_2 u\Big)\right) \\
&= \exp\left(-\frac{1}{2}\Big((Tn\tau_1+\tau_0)\left(\theta_1^2 - \frac{2T\tau_1 n\bar{x}_1\theta_1}{Tn\tau_1+\tau_0}\right) + (Tn\tau_2+\tau_0)\left(u^2 - \frac{2T\tau_2 n\bar{x}_2 u}{Tn\tau_2+\tau_0}\right)\Big)\right) \\
&\propto \exp\left(-\frac{1}{2}\Big((Tn\tau_1+\tau_0)\left(\theta_1 - \frac{T\tau_1 n\bar{x}_1}{Tn\tau_1+\tau_0}\right)^2 + (Tn\tau_2+\tau_0)\left(u - \frac{T\tau_2 n\bar{x}_2}{Tn\tau_2+\tau_0}\right)^2\Big)\right)
\end{aligned}
$$

As $p(\theta \mid X)$ is proportional to the density of a banana distribution, the posterior is the banana distribution $\mathrm{Ban}(\mu, \Sigma, a, b, m)$ with

$$
\mu = \left(\frac{Tn\tau_1\bar{x}_1}{Tn\tau_1+\tau_0}, \frac{Tn\tau_2\bar{x}_2}{Tn\tau_2+\tau_0}\right),
$$

$$
\Sigma = \mathrm{diag}\left(\frac{1}{Tn\tau_1+\tau_0}, \frac{1}{Tn\tau_2+\tau_0}\right).
$$

$\square$

3. Differentially Private MCMC

As seen in Section 2.1, an algorithm can be made differentially private by adding Gaussian noise the its output. The noise could also be added to any intermediate value calculated by the algorithm, and post processing immunity will guarantee that the same DP bounds that hold for releasing the intermediate value also hold for releasing the final result of the algorithm.

In 2019, Yildirim and Ermis [YE19] realised that if Gaussian noise is added to the exact value of λ, the noise can be corrected for yielding a differentially private MCMC algorithm which converges to the correct distribution. In the same year, Heikkilä et. al. [HJDH19] developed another DP MCMC algorithm, called DP Barker, which uses subsampling to amplify privacy.

3.1 DP Penalty

In 1999, Ceperley and Dewing [CD99] developed a variant of Metropolis-Hastings called the penalty algorithm, where only a noisy approximation of λ is known. They developed the algorithm for simulations in physics where computing λ requires computing energies of complex systems, which can only be approximated. The penalty algorithm modifies the acceptance probability to account for the noise added to λ and still converges to the correct distribution if the noise is Gaussian with known variance.

The DP penalty algorithm adds Gaussian noise to the value of λ, and uses the penalty algorithm to correct the acceptance probability so that the algorithm still converges to the correct distribution [YE19]. The corrected acceptance probability for Gaussian noise with variance σ^2 is

$$\min\{1, e^{\lambda - \frac{1}{2}\sigma^2}\}$$

Theorem 6 gives the number of iterations DP penalty can be run for when the privacy cost is computed through zCDP, which is what Yildirim and Ermis prove in their paper [YE19]. A tighter, but harder to use, bound can be reached without using zCDP. This is given by Theorem 7.

Theorem 6. *Let $\epsilon > 0$, $0 < \delta < 1$, $\alpha > 0$ and $\tau > 0$. Let*

$$\rho = (\sqrt{\epsilon - \ln \delta} - \sqrt{-\ln \delta})^2$$

$$c(\theta, \theta') = \sup_{D_j, D'_j} (p(D_j \mid \theta') - p(D_j \mid \theta) - (p(D'_j \mid \theta') - p(D'_j \mid \theta)))$$

$$\sigma^2(\theta, \theta') = \tau^2 n^{2\alpha} c^2(\theta, \theta')$$

Then DP penalty can be run for

$$k = \lfloor 2\tau^2 n^{2\alpha} \rho \rfloor$$

iterations when using σ^2 as the variance of the Gaussian noise.

Theorem 7. *Let $\epsilon > 0$ and $\tau > 0$. Define c and σ as in Theorem 6. The DP penalty algorithm, after running for k iterations using σ as the noise variance, is $(\epsilon, \delta(\epsilon))$-DP for*

$$\delta(\epsilon) = \frac{1}{2}\left(\operatorname{erfc}\left(\frac{\tau n^\alpha (\epsilon - k\mu)}{\sqrt{2k}}\right) - e^\epsilon \operatorname{erfc}\left(\frac{\tau n^\alpha (\epsilon + k\mu)}{\sqrt{2k}}\right)\right)$$

where $\mu = \frac{1}{\tau^2 n^{2\alpha}}$.

Proof. DP penalty is an adaptive composition of Gaussian mechanisms that release noisy values of $\lambda(\theta, \theta')$. The sensitivity of $\lambda(\theta, \theta')$ is $c(\theta, \theta')$. For the tight ADP bound used here, the sensitivity must be constant in each iteration. This is achieved by releasing $\frac{\lambda(\theta,\theta')}{c(\theta,\theta')}$ instead, which has sensitivity 1. $c(\theta, \theta')$ does not depend on D, so $\lambda(\theta, \theta')$ can be obtained from $\frac{\lambda(\theta,\theta')}{c(\theta,\theta')}$ by post processing.

Adding Gaussian noise with variance σ_n^2 to $\frac{\lambda(\theta,\theta')}{c(\theta,\theta')}$ is equivalent to adding Gaussian noise with variance $\sigma_n^2 c^2(\theta, \theta')$ to $\lambda(\theta, \theta')$. Setting $\sigma_n^2 = \tau^2 n^{2\alpha}$ and plugging into the ADP bound of Theorem 4 proves the claim. □

Theorem 7 is harder to use than Theorem 6 because the number of iteration DP penalty can be run for given an (ϵ, δ)-bound cannot be computed analytically for the former. However, the maximum number of iterations can be solved for numerically.

Theorems 6 and 7 require a bound on sensitivity of the log likelihood ratio. If there is a bound

$$|\ln p(D_j \mid \theta') - \ln p(D_j \mid \theta)| \leq L||\theta - \theta'||_2$$

for all D_j, θ and θ' then

$$c(\theta, \theta') \leq 2L||\theta - \theta'||_2$$

The former bound is true in some model, such as logistic regression. In other models it can be forced by clipping the log likelihood ratios to the interval $[-L||\theta - \theta'||_2, L||\theta - \theta'||_2]$. This will remove the guarantee of eventually converging to the correct posterior, but if L is chosen to be large enough, the clipping will not affect the acceptance decision frequently. As a tradeoff, picking a large L will increase the variance of the Gaussian noise and slow down convergence through it.

Yildirim and Ermis [YE19] propose two potential ways to improve the performance of the penalty algorithm. The first improvement is only proposing changes in one dimension in a multidimensional problem. This decreases $||\theta - \theta'||_2$, which means that it decreases the noise variance.

The second improvement is called *guided random walk* (GRW) [YE19]. In GRW, proposals change only one dimension, as above. Additionally, a direction is associated with each dimension, and proposals are only made the current direction of the chosen dimension. After an accepted proposal, the direction is kept the same, but after a reject it is switched. This means that the chain can move towards areas of higher probability faster because, after some initial proposals are rejected, the directions for each dimension point towards the area of high probability, so all proposals are towards it. Without GRW, most proposals would move the chain away from the area of high probability, and would likely be rejected.

3.2 DP Barker

The DP Barker algorithm of Heikkilä et. al. [HJDH19] is based on the Barker acceptance test [Bar65] instead of the Metropolis-Hastings test. Instead of using the MH acceptance probability, the Barker acceptance test samples $V_{log} \sim \text{Logistic}(0, 1)$ and accepts if

$$\lambda + V_{log} > 0$$

If Gaussian noise with variance σ^2 is added to λ, there exists a correction distribution V_{corr} such that $\mathcal{N}(0, \sigma^2) + V_{corr}$ has the same distribution as V_{log}. Because the variance of V_{log} is $\frac{\pi^2}{3}$ [HJDH19], the variance of V_{corr} must be $\frac{\pi^2}{3} - \sigma^2$ which means that there is an upper bound to the noise variance: $\sigma^2 < \frac{\pi^2}{3}$. Testing whether $\lambda + \mathcal{N}(0, \sigma^2) + V_{corr} > 0$ is equivalent to testing whether $\lambda + V_{log} > 0$, which means that it is possible to derive a DP MCMC algorithm based on the Barker acceptance test if the correction distribution can be sampled from.

However, the analytical form of V_{corr} is not known [HJDH19]. Heikkilä et. al. approximate the distribution with a Gaussian mixture model. This means that their algorithm only converges to an approximately correct distribution, but the approximation error can be made very small.

If the sum in λ was only computed over a subset of the data, the algorithm would take less computation to run, and would be less sensitive to changes in the data.

The latter property is called *subsampling amplification* of differential privacy [WBK19]. Using the λ computed with subsampling instead of the full data λ introduces an additional error that must be corrected for to have the algorithm converge to the correct distribution.

The *central limit theorem* (CLT) states that the distribution of a sum of random variables approaches a Gaussian distribution as more random variables are summed, if some conditions on the independence and variance of the random variables are met [HJDH19]. With the CLT, it can be argued that the error from using the subsampled λ instead of the full data λ has an approximately Gaussian distribution, if the subsample is large enough [HJDH19].

The variance of the error from subsampling can be estimated by the sample variance of the individual terms in the sum in λ [HJDH19]. This allows combining the errors from subsampling and the Gaussian noise from the Gaussian mechanism to a single Gaussian noise value. The V_{corr} distribution can then be used to approximate the Barker acceptance test as above. See algorithm 2 for the DP Barker algorithm. *

Heikkilä et. al. [HJDH19] do not directly bound the sensitivity of λ as is done in DP penalty, because the sample variance also depends on input data. Instead they directly bound the Rényi divergence between $\mathcal{N}(0, \sigma^2 - \sigma_b^2)$, where σ_b^2 is the batch sample variance, for two adjacent inputs. Subsampling amplification is accounted for with an amplification theorem for Rényi DP [WBK19].

Theorem 8. *If*

$$|\ln p(D_j \mid \theta') - \ln p(D_j \mid \theta)| \leq \frac{\sqrt{|B|}}{n}$$

$$\alpha < \frac{|B|}{5}, \alpha \in \mathbb{N}$$

for all $\theta, \theta' \in \Theta$*, all* D *and* $1 \leq j \leq n$*, running* k *iterations of DP Barker is* $(\alpha, k\epsilon(\alpha))$*-RDP, with*

$$\epsilon(\alpha) = \frac{1}{\alpha - 1} \ln \left(1 + q^2 \binom{\alpha}{2} \min\{4(e^{\epsilon'(2)} - 1), 2e^{\epsilon'(2)}\} + 2 \sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon'(j)} \right)$$

and

$$\epsilon'(\alpha) = \frac{5}{2|B|} + \frac{1}{2(\alpha - 1)} \ln \frac{2|B|}{|B| - 5\alpha} + \frac{2\alpha}{|B| - 5\alpha}$$

where n *is the number of rows in* D*,* $|B|$ *is the size of the minibatch and* $q = \frac{|B|}{n}$*.*

Like DP penalty, DP Barker requires a bound on the log likelihood ratio for one row of data. The bound can be forced through clipping if the model does not meet it, but because of the n in the denominator of the bound, it can get very tight for large

*See [HJDH19] for the sampling procedure of V_{corr}.

values of n. As a result, clipping may be needed for almost all log likelihood ratios, which may cause the algorithm to converge to a very different distribution from the posterior.

To alleviate the tight bound on log likelihood sensitivity, DP Barker is best used with a tempered likelihood [HJDH19]. In tempering, the log likelihood is multiplied by a number $T = \frac{n_0}{n} < 1$. This increases the variance of the resulting posterior and may lower modeling error in some cases [HJDH19].

Using the tempered likelihood, the log likelihood bound becomes

$$T|\ln p(D_j \mid \theta') - \ln p(D_j \mid \theta)| \leq \frac{\sqrt{|B|}}{n}$$

which is equivalent to

$$|\ln p(D_j \mid \theta') - \ln p(D_j \mid \theta)| \leq \frac{\sqrt{|B|}}{n_0}$$

Typically $n_0 \ll n$ for large datasets, so using a tempered likelihood requires significantly less clipping than a nontempered likelihood.

Algorithm 2: DP Barker

sample $\theta^* \sim q(\theta_{i-1})$
sample $B \subset \{1, \ldots, n\}$
for $1 \leq i \leq k$ **do**
  for $j \in B$ **do**
    $r_j = \ln \frac{p(\theta^* | D_j)}{p(\theta_{i-1} | D_j)}$
  end
  $\sigma_b^2 = \mathrm{Var}\{r_j \mid j \in B\}$
  $\lambda = \frac{n}{|B|} \sum_{j \in B} r_j + \ln \frac{p(\theta^*)}{p(\theta_{i-1})} + \ln \frac{q(\theta_{i-1}|\theta^*)}{q(\theta^*|\theta_{i-1})}$
  sample $s \sim \mathcal{N}(0, \sigma^2 - \sigma_b^2)$
  sample $c \sim V_{corr}^{\sigma^2}$
  $\theta_i = \begin{cases} \theta^* & \text{if } \lambda + s + c > 0 \\ \theta_{i-1} & \text{otherwise} \end{cases}$
end
return $(\theta_1, \ldots, \theta_k)$

3.3 Comparing DP Penalty and DP Barker

4. Variations of the Penalty Algorithm

4.1 The Penalty Algorithm with Subsampling

In the DP Barker algorithm, the log likelihood ratio is computed using only a subsample of the dataset to amplify privacy. Subsampling can also be used with the penalty algorithm in the same way, if the acceptance test is corrected for the subsampling.

As with DP Barker, the error from subsampling is approximately normally distributed by the central limit theorem. The variance of the subsampling error can be estimated from the sample variance of individual terms of the sum in the log likelihood ratio. This means that the penalty method can be used to correct for the subsampling error.

The acceptance probability with subsampling is

$$\min\{1, e^{\lambda - \frac{1}{2}(\sigma^2 + \sigma_b^2)}\},$$

where

$$\lambda = \frac{nT}{|B|} \sum_{j \in B} \ln \frac{p(D_j \mid \theta')}{p(D_j \mid \theta)} + \ln \frac{p(\theta')q(\theta \mid \theta')}{p(\theta)q(\theta' \mid \theta)},$$

and σ_b^2 is the sample variance of the log likelihood ratios in batch B. Denote

$$r_j = \ln \frac{p(D_j \mid \theta')}{p(D_j \mid \theta)},$$

$$R = \sum_{x \in B} r_j.$$

Then σ_b^2 can be estimated from the sample variance of r_j:

$$\begin{aligned}
\sigma_b^2 &= \mathrm{Var}\left(\frac{nT}{|B|} \sum_{j \in B} r_j\right) = \frac{nT^2}{|B|^2} \sum_{j \in B} \mathrm{Var}(r_j) = \frac{nT^2}{|B|} \mathrm{Var}(r_j) \\
&\approx \frac{(nT)^2}{|B|^2} \sum_{j \in B} \left(r_j - \frac{R}{|B|}\right)^2 = \frac{(nT)^2}{|B|^2} \left(\sum_{j \in B} r_j^2 - \frac{R^2}{|B|}\right).
\end{aligned}$$

Because σ_b^2 depends on the data, releasing λ privately is not enough, $\lambda - \frac{1}{2}\sigma_b^2$ must be released privately. This means that using subsampling requires adding additional noise to account for the sensitivity of $\frac{1}{2}\sigma_b^2$.

The sensitivity of $\lambda - \frac{1}{2}\sigma_b^2$ is

$$\Delta\lambda + \frac{1}{2}\Delta\sigma_b^2.$$

With the bound $r_j \leq L||\theta - \theta'||_2$ used in DP penalty, the bound sensitivity of λ is the same as without subsampling. The sensitivity of σ_b^2 must be bounded separately.

Lemma 1. *The sensitivity of $\frac{1}{2}\sigma_b^2$, with $r_j \leq L||\theta - \theta'||_2$, has upper bound*

$$\frac{1}{2}\Delta\sigma_b^2 \leq \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| L^2||\theta - \theta'||_2^2 + \frac{2(b-1)}{b}\left(\frac{nT}{b}\right)^2 L^2||\theta - \theta'||_2^2.$$

Proof. For datasets $D \sim D'$, that only differ in one element, denote the common part they have by D^*, and the differing element by $d \in D$ and $d' \in D'$

$$\begin{aligned}
\Delta\sigma_b^2 &= \sup_{D\sim D'} |\sigma_b^2(D) - \sigma_b^2(D')| \\
&= \left(\frac{nT}{b}\right)^2 \sup_{D\sim D'} \left|\sum_{d\in D} r^2(d) - \sum_{d\in D'} r^2(d) + \frac{1}{b}R^2(D') - \frac{1}{b}R^2(D)\right| \\
&= \left(\frac{nT}{b}\right)^2 \sup_{d,d',D^*} \left|r^2(d) - r^2(d') + \frac{1}{b}(R(D^*) + r(d'))^2 - \frac{1}{b}(R(D^*) + r(d))^2\right| \\
&= \left(\frac{nT}{b}\right)^2 \sup_{d,d',D^*} \left|r^2(d) - r^2(d') + \frac{1}{b}(R^2(D^*) + 2R(D^*)r(d') + r^2(d'))\right. \\
&\left. - \frac{1}{b}(R^2(D^*) + 2R(D^*)r(d) + r^2(d))\right| \\
&= \left(\frac{nT}{b}\right)^2 \sup_{d,d',X^*} \left|\left(1 - \frac{1}{b}\right)(r^2(d) - r^2(d')) + \frac{2}{b}D(X^*)(r(d') - r(d))\right| \\
&\leq \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| \sup_{d,d'} \left|(r^2(d) - r^2(d'))\right| + \frac{2}{b}\left(\frac{nT}{b}\right)^2 \sup_{d,d',D^*} \left|R(D^*)(r(d') - r(d))\right| \\
&= \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| \sup_{d,d'} \left|(r^2(d) - r^2(d'))\right| + \frac{2}{b}\left(\frac{nT}{b}\right)^2 \sup_{d,d'} |r(d') - r(d)| \sup_{X^*} |R(D^*)| \\
&\leq \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| \sup_{d,d'} \left|(r^2(d) - r^2(d'))\right| + \frac{2}{b}\left(\frac{nT}{b}\right)^2 \sup_{d,d'} |r(d') - r(d)|(b-1) \sup_{d} |r(d)|.
\end{aligned}$$

Plugging the bound $\sup_d |r(d)| \leq L||\theta - \theta'||_2$ into the last expression proves the claim. □

Theorem 9. *Let*

$$\Delta_\lambda = \frac{2nTL}{|B|}||\theta - \theta'||_2,$$

$$\Delta_\sigma = \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| L^2||\theta - \theta'||_2^2, + \frac{2(b-1)}{b}\left(\frac{nT}{b}\right)^2 L^2||\theta - \theta'||_2^2.$$

$$c(\theta, \theta') = \Delta_\lambda + \Delta_\sigma,$$

$$\sigma^2(\theta, \theta') = \tau c^2(\theta, \theta').$$

Then running DP penalty with subsampling for k iterations is $(\alpha, k\epsilon(\alpha))$-RDP, with

$$\epsilon(\alpha) = \frac{1}{\alpha - 1} \ln\left(1 + q^2\binom{\alpha}{2}\min\{4(e^{\epsilon'(2)} - 1), 2e^{\epsilon'(2)}\} + 2\sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon'(j)}\right),$$

and

$$\epsilon'(\alpha) = \frac{\alpha}{2\tau},$$

where n is the number of rows in D, $|B|$ is the size of the minibatch and $q = \frac{|B|}{n}$.

Proof. By Lemma 1, $\Delta_\sigma(\theta, \theta')$ an upper bound to the sensitivity of $\frac{1}{2}\sigma_b^2$, therefore $c(\theta, \theta')$ is an upper bound to the sensitivity of $\lambda - \frac{1}{2}\sigma_b^2$.

This means that a Gaussian mechanism taking a subsample B of the data as input and uses $\sigma(\theta, \theta')$ as the noise variance is $(\alpha, \epsilon'(\alpha))$-RDP with

$$\epsilon'(\alpha) = \frac{\alpha}{2\tau}.$$

By the subsampling amplification theorem [WBK19, Theorem 9] and the composition theorem of RDP (Theorem 2), the combination of subsampling and Gaussian mechanism is $(\alpha, k\epsilon(\alpha))$-RDP with

$$\epsilon(\alpha) = \frac{1}{\alpha - 1} \ln\left(1 + q^2\binom{\alpha}{2}\min\{4(e^{\epsilon'(2)} - 1), 2e^{\epsilon'(2)}\} + 2\sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon'(j)}\right)$$

when run for k iterations for integer $\alpha \geq 2$. □

4.2 DP Metropolis-Adjusted Langevin Algorithm

5. The Gauss-Bernoulli Algorithm

6. Experiments

6.1 Banana Distribution

7. Conclusions

Bibliography

[Bar65] Av A Barker. Monte carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18(2):119–134, 1965.

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, pages 635–658, 2016.

[CD99] DM Ceperley and Mark Dewing. The penalty method for random walks with uncertain energies. *The Journal of chemical physics*, 110(20):9812–9820, 1999.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[GCS+14] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman & Hall/CRC texts in statistical science series. CRC Press, Boca Raton, third edition, 2014.

[HJDH19] Mikko A. Heikkilä, Joonas Jälkö, Onur Dikmen, and Antti Honkela. Differentially private markov chain monte carlo. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 4115–4125, 2019.

[Mir17] Ilya Mironov. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 263–275, 2017.

[SMM19] David M. Sommer, Sebastian Meiser, and Esfandiar Mohammadi. Privacy loss classes: The central limit theorem in differential privacy. *PoPETs*, 2019(2):245–269, 2019.

[TPK14] Minh-Ngoc Tran, Michael K. Pitt, and Robert Kohn. Adaptive metropolis-hastings sampling using reversible dependent mixture proposals. *Statistics and Computing*, 26(1-2):361–381, 2014.

[WBK19] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled renyi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 1226–1235, 2019.

[YE19] Sinan Yildirim and Beyza Ermis. Exact MCMC with differentially private moves - revisiting the penalty algorithm in a data privacy framework. *Statistics and Computing*, 29(5):947–963, 2019.