Master's thesis
Master's Programme in Data Science

Differentially Private Markov Chain Monte Carlo

Ossi Räisä

February 17, 2021

Supervisor(s): Associate Professor Antti Honkela

Examiner(s): Associate Professor Antti Honkela
Doctor 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	February 17, 2021	62

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

This chapter covers the basics of differential privacy, Bayesian inference and Markov chain Monte Carlo algorithms needed in the later chapters. To keep the introduction brief, only directly needed concepts are covered, and much of the motivation behind the subjects is left out.

2.1 Differential Privacy

Differential privacy (DP) [DMNS06, 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. DP algorithms must necessarily[†] include randomness to mask influence of the private data, so all of the considered algorithms in this thesis are randomised. DP randomised algorithms are also called *mechanisms* in this thesis, and in the DP literature.

The most common definition is called (ϵ, δ)-approximate differential privacy (ADP) [DKM+06, DR14]. The case where $\delta = 0$ is called ϵ-DP or pure DP.

Definition 1. *A mechanism $\mathcal{M}\colon \mathcal{X} \to \mathcal{U}$ 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.*

[†]Unless the algorithm does not actually use the private data.

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 distance* 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*

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

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 a randomised algorithm with neighbouring inputs:

Definition 4. *A mechanism $\mathcal{M}$ 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 mechanisms $\mathcal{M}$ and $\mathcal{M}'$ are DP, the mechanism 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 mechanisms. Their composition is the mechanism 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 mechanisms. Then*

*Statistical divergences are commonly called distances, even though they typically are not metrics.

1. *If $\mathcal{M}$ is (ϵ, δ)-ADP and $\mathcal{M}'$ is (ϵ', δ')-ADP, then their composition is $(\epsilon+\epsilon', \delta+\delta')$-ADP [DKM+06]*

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 randomised 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 (ϵ, δ)-ADP, (α, ϵ)-RDP or ρ-zCDP mechanism. Let $f\colon \mathcal{U} \to \mathcal{U}'$ be any randomised algorithm not using the private data. Then the composition of $\mathcal{M}$ and f is (ϵ, δ)-ADP, (α, ϵ)-RDP or ρ-zCDP.*

There are many different DP mechanisms that are commonly used[DR14]. This thesis only requires one of the most commonly used ones: the Gaussian mechanism [DKM+06].

Definition 6. *The Gaussian mechanism with parameter σ^2 is a randomised algorithm that, with input data x and query $f\colon \mathcal{X} \to \mathbb{R}^d$, outputs a sample from $\mathcal{N}(f(x), \sigma^2)$, where $\mathcal{N}$ denotes the normal distribution.*

The privacy bounds of the Gaussian mechanism require that the values of the query f do not vary too much for neighbouring inputs. This requirement is formalised as a bound on the *sensitivity* of f.

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

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

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

Theorem 4. *If $\Delta_2 f \leq \Delta$, 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. *k compositions of the Gaussian mechanism, with queries f_i, where $\Delta_2 f_i \leq \Delta$ for $1 \leq i \leq k$, are $(\epsilon, \delta(\epsilon))$-ADP [SMM19] with*

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

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

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 utility of the released value depends on the magnitude of σ^2 compared to the actual value. As in Definition 5, each function f_i can depend on the output of the previous functions f_j, $j < i$.

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 $p(\theta \mid X)$ to the *prior* belief $p(\theta)$ through the observed data X and the likelihood of the data $p(X \mid \theta)$ as follows:

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

It is theoretically possible to compute $p(\theta \mid X)$ 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) [MRR+53] is a widely used method to obtain such samples.

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) [MRR+53, Has70].

At each iteration i, the Metropolis-Hastings algorithm samples θ_i from a distribution π of the parameters by first picking a proposal θ' from a proposal distribution $q(\cdot \mid \theta_{i-1})$ [MRR+53], where θ_{i-1} is the previously sampled value*. We shorten θ_{i-1} to θ in the following. The ratio of posterior and proposal densities is calculated

$$r(\theta, \theta') = \frac{\pi(\theta')}{\pi(\theta)} \frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)},$$

and the proposal is accepted with probability $\min\{1, r\}$. If the proposal is accepted, $\theta_i = \theta'$, otherwise $\theta_i = \theta$.

It can be shown that, with a suitable proposal distribution, the chain of θ_i values converges to π [Has70]. 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

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

The difficult integral $\int p(X \mid \theta)p(\theta)\mathrm{d}\theta$ 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 $\min\{1, e^{\lambda(\theta,\theta')}\}$ where

$$\lambda(\theta, \theta') = \ln \frac{p(X \mid \theta')}{p(X \mid \theta)} + \ln \frac{p(\theta')}{p(\theta)} + \ln \frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)}. \tag{2.1}$$

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

$$p(X \mid \theta) = \prod_{j=1}^{n} p(x_j \mid \theta),$$

which means that the log-likelihood ratio term of λ is

$$\ln \frac{p(X \mid \theta')}{p(X \mid \theta)} = \sum_{j=1}^{n} \ln \frac{p(x_j \mid \theta')}{p(x_j \mid \theta)}.$$

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

*The value of θ_0 for the first iteration is given as input to the algorithm.

inference.

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

for $1 \leq i \leq k$ **do**

denote $\theta = \theta_{i-1}$

sample $\theta' \sim q(\cdot \mid \theta)$

$\ln \frac{p(X|\theta')}{p(X|\theta)} = \sum_{j=1}^{n} (\ln p(x_j \mid \theta') - \ln p(x_j \mid \theta))$

$\lambda = \ln \frac{p(X|\theta')}{p(X|\theta)} + \ln p(\theta') - \ln p(\theta) + \ln q(\theta \mid \theta') - \ln q(\theta' \mid \theta)$

$$\theta_i = \begin{cases} \theta' & \text{with probability } \min\{1, e^{\lambda}\} \\ \theta & \text{otherwise} \end{cases}$$

end

return $(\theta_1, \ldots, \theta_k)$

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 from Equation 2.1, 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 Correcting MH with the penalty algorithm

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. The original algorithm was developed 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

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

Theorem 5 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 by using the ADP bound of Theorem 4, without using zCDP. This is given by Theorem 6.

Theorem 5. *Let $\epsilon > 0$, $0 < \delta < 1$, $\alpha > 0$ and $\tau > 0$. Let*

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

$$c(\theta, \theta') = \sup_{x_j, x_j'} (p(x_j \mid \theta') - p(x_j \mid \theta) - (p(x_j' \mid \theta') - p(x_j' \mid \theta)))$$

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

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 6. *Let $\epsilon > 0$ and $\tau > 0$. Define c and σ as in Theorem 5. The DP penalty algorithm, after running for k iterations using σ as the noise variance, is $(\epsilon, \delta(\epsilon))$-DP for*

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

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

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 X, 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 6 is harder to use than Theorem 5 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. Algorithm 2 shows a simple procedure that solves the maximum number of iterations

using the bisection method.

Algorithm 2: Maximise the number of iterations given ϵ, δ, τ and n. The zcdp-function computes the number of iterations Theorem 5 allows, and the adp-function computes $\delta(\epsilon)$ from Theorem 6. $\lfloor \cdot \rfloor$ is the floor function that rounds real numbers down. Note that the variables *low*, *high* and *new* are not necessarily integers, as Theorem 6 can handle a non-integer number of iterations.

$low = \text{zcdp}(\epsilon, \delta, \tau, n)$
$high = \max\{low, 1\}$
while $\text{adp}(\epsilon, high, \tau, n) < \delta$ **do**
  $high = high \cdot 2$
end
while $\lfloor high \rfloor - \lfloor low \rfloor > 1$ **do**
  $new = \frac{high+low}{2}$
  if $\text{adp}(\epsilon, new, \tau, n) > \delta$ **then**
    $high = new$
  end
  else
    $low = new$
  end
end
if $\text{adp}(\epsilon, \lfloor high \rfloor, \tau, n) < \delta$ **then**
  return $\lfloor high \rfloor$
end
else
  return $\lfloor low \rfloor$
end

Theorems 5 and 6 require a bound on sensitivity of the log-likelihood ratio. If there is a bound

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

for all D_j, θ and θ' then

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

The former bound is true in some models, such as logistic regression [YE19]. 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, called *one component updates* (OCU) in this thesis, 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 walk Metropolis Hastings* (GWMH) [YE19]. In GWMH, 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 GRMH, most proposals would move the chain away from the area of high probability, and would likely be rejected.

3.2 The Barker acceptance test

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, as long as σ^2 is not too large, there exists an approximate correction distribution V_{corr} such that $\mathcal{N}(0, \sigma^2) + V_{corr}$ has approximately 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 approximately 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 addi-

tional 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 [SPCC17]. 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 [SPCC17].

The variance of the error from subsampling can be estimated by the sample variance of the individual terms in the sum in λ [SPCC17]. 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 [HJDH19]. See Algorithm 3 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 7. *If*

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

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

for all $\theta, \theta' \in \Theta$, *all* X *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* $q = \frac{|B|}{n}$.

Algorithm 2 can modified to compute the maximum number of iterations Theorem 7 allows by computing ϵ-bounds given δ with Theorem 7 instead of computing

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

δ-bounds given ϵ. The variable *low* of Algorithm 2 must be initialised to 0, and the variable *high* must be initialised to a value greater than 0*.

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 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

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

which is equivalent to

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

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

Algorithm 3: DP Barker

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

end

return $(\theta_1, \ldots, \theta_k)$

*Choosing a value close to the maximum number of iterations speeds up the computation of the maximum number of iterations. The experiments in this thesis used the value 1024.

3.3 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

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

where

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

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

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

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

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

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

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 $X \sim X'$, that only differ in one element, denote the common part they have by X^*, and the differing element by $x \in X$ and $x' \in X'$

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

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

Theorem 8. *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 $\alpha \geq 2$. □

Theorem 8 does not give tight bounds for ADP, as it is based on RDP. Computing tight ADP bounds for the minibatch penalty algorithm requires an analogue of the ADP bound of Theorem 4 for the Gaussian mechanism with subsampling. An analytical and tight ADP bound for the subsampled Gaussian mechanism is not known, but the *Fourier accountant* of Koskela et al. [KJH20] can approximate the tight bound with a very small error.

The Fourier accountant takes the subsampling ratio q and noise variance, and computes the ADP bounds of the subsampled Gaussian mechanism with sensitivity 1. For the minibatch penalty algorithm, the released value $\lambda - \frac{1}{2}\sigma_b^2$ has sensitivity $c(\theta, \theta')$, and noise variance $\tau c^2(\theta, \theta')$, but if the released value is normalised to have sensitivity 1, the noise variance becomes τ, which makes the Fourier accountant easy to apply to the minibatch penalty algorithm.

The DP Barker algorithm, like minibatch penalty, uses RDP to compute privacy bounds. Using the Fourier accountant with DP Barker may be possible, but it is not as simple as with minibatch penalty, as DP Barker does not consider releasing the sample variance σ_b^2 directly.

As with the DP penalty and DP Barker algorithms, the maximum number of iterations for minibatch DP penalty can be computed using Algorithm 2. For Theorem 8, Algorithm 2 must be modified the as it is modified for DP Barker (Section 3.2). With the Fourier accountant, the only necessary modification is computing δ with the Fourier accountant*.

The appearance of n as a multiplier of Δ_λ and n^2 as a multiplier of Δ_σ causes the noise variance to increase with n, without a corresponding decrease in ϵ'. This makes

*The Fourier accountant handles the non-integer iteration numbers used by Algorithm 2 by internally rounding them down.

subsampled DP penalty unsuitable for problems with large datasets, unless tempering is used. Like with DP Barker, using tempering $T = \frac{n_0}{n}$ cancels n out of the noise variance.

The proposed variations of the penalty algorithm, one component updates and guided walk MH, are also applicable with subsampling. The sensitivity reduction from OCU may be especially useful as the variance sensitivity contains the square of the distance between the current and proposed parameter values, so reducing the distance by updating only a single component can greatly reduce the added noise.

4. Differentially Private Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC[†]) [DKPR87, Nea12] is a widely used MCMC algorithm that, like Metropolis-Hastings and the penalty algorithm, was originally developed for simulations in physics and chemistry, but has since been applied to Bayesian inference.

HMC simulates the trajectory of a moving particle in a way that allows it to draw samples from a target distribution [Nea12]. With perfect simulation, an acceptance test would not be needed, but exact simulation of the moving particle is typically not possible, so a Metropolis-Hastings acceptance test is used for proposals generated by the simulation. Despite imperfections, the simulation can generate long jumps that stay in areas of high probability, giving HMC a higher acceptance rate that Metropolis-Hastings using a Gaussian distribution as the proposal, at the cost of higher computational requirements.

This chapter first introduces HMC in the non-DP setting in Section 4.1. In Section 4.2 the HMC is made differentially private using the acceptance test of the penalty algorithm and adding noise to the proposal phase appropriately.

4.1 MCMC with Hamiltonian Dynamics

The motion of a particle on a frictionless surface of varying height is governed by the initial position and velocity of the particle, and the height of the surface at different positions [Nea12]. The kinetic energy of the particle with momentum[‡] $p \in \mathbb{R}^2$ and mass $m \in \mathbb{R}$ is $\frac{p^T p}{2m}$. The potential energy of the particle at position $\theta \in \mathbb{R}^2$ is proportional to the height of the surface, given by a continuously differentiable function $U(\theta)$. The sum of potential and kinetic energies, the total energy of the system, is called the Hamiltonian H. Hamilton's equations give the equations of motion for a system with

[†]HMC originally stood for hybrid Monte Carlo, but the name Hamiltonian Monte Carlo has become more common.

[‡]Momentum is velocity times mass.

a given Hamiltonian:

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial p}, \qquad \frac{dp}{dt} = -\frac{\partial H}{\partial \theta}.$$

Given $H(\theta, p) = U(\theta) + \frac{p^T p}{2m}$, Hamilton's equations become

$$\begin{aligned}\frac{d\theta}{dt} &= \frac{p}{m},\\ \frac{dp}{dt} &= -\nabla U(\theta).\end{aligned}$$

These equations define a function T_t taking an initial state (θ, p) of the particle to the state $T(\theta, p)$ after time t. Solving T_t analytically is usually not possible, but T_t can be approximately simulated by the *leapfrog method*. The leapfrog method sequentially updates an estimate (θ_i, p_i) of the state in steps L of η as follows:

$$\begin{aligned}p_{i+1/2} &= p_i - \frac{\eta}{2}\nabla U(\theta_i),\\ \theta_{i+1} &= \theta_i + \frac{\eta p_{i+1/2}}{m},\\ p_{i+1} &= p_{i+1/2} - \frac{\eta}{2}\nabla U(\theta_{i+1}).\end{aligned}$$

The simulation starts with $\theta_0 = \theta$ and $p_0 = p$ and the result is given by $T_{L\eta}(\theta, p) \approx (\theta_L, p_L)$.

The above equations for two-dimensional θ and p generalize to any number of dimensions d [Nea12]. The physical analogue would be a particle moving on a d-dimensional hypersurface with "height" given by $U(\theta)$. The mass of the particle, m, can also be generalized to any positive definite matrix $M \in \mathbb{R}^{p\times p}$, which changes division by m to multiplication by M^{-1}. Specifically, the kinetic energy of the particle becomes $\frac{1}{2}p^T M^{-1} p$ and $\frac{d\theta}{dt} = M^{-1}p$. The generalization to a mass matrix does not have a physical analogue, but it can be useful for MCMC.

Hamiltonian dynamics have three important properties [Nea12] that make them useful as a proposal mechanism for MCMC. They are reversibility, conservation of the Hamiltonian and conservation of volume.

Reversibility means that the function T_t is injective, meaning that it has an inverse T_{-t} where $T_t(\theta_1, p_1) = (\theta_2, p_2)$ implies that $T_{-t}(\theta_2, p_2) = (\theta_1, p_1)$ [Nea12]. For the Hamiltonian of the particle moving on a surface, the inverse is obtained by $T_t(\theta_2, -p_2) = (\theta_1, -p_1)$.

Conservation of the Hamiltonian follows from Hamilton's equations and the chain rule of derivatives:

$$\frac{dH}{dt} = \frac{\partial H}{\partial \theta}\frac{d\theta}{dt} + \frac{\partial H}{\partial p}\frac{dp}{dt} = \frac{\partial H}{\partial \theta}\frac{\partial H}{\partial p} - \frac{\partial H}{\partial p}\frac{\partial H}{\partial \theta} = 0.$$

Conservation of volume in the (θ, p)-space follows from the fact that the absolute value of the Jacobian determinant of T_t is 1 and the injectivity of T_t. A somewhat lengthier proof for the Jacobian determinant of T_t is given by Neal [Nea12].

To see why having $|\det J_f| = 1$, where J_f is the Jacobian matrix of an injective continuously differentiable function f, implies f preserving volume, recall that the volume of a set A is given by

$$\int_A 1 \mathrm{d}x.$$

The volume of the image of A under f, $f(A)$, is given by

$$\int_{f(A)} 1 \mathrm{d}x = \int_A |\det J_f(x)| \mathrm{d}x = \int_A 1 \mathrm{d}x.$$

where the first equality follows from the change of variables formula.

The leapfrog method does not conserve the Hamiltonian exactly, but it preserves volumes and is reversible exactly [Nea12]. These properties are important for its use as a proposal mechanism for MCMC.

The HMC algorithm samples from the distribution of [Nea12]

$$P(\theta, p) \propto \exp(-H(\theta, p)).$$

With $H(\theta, p) = U(\theta) + \frac{1}{2} p^T M^{-1} p$,

$$P(\theta, p) \propto \exp(-U(\theta)) \exp\left(\frac{1}{2} p^T M^{-1} p\right)$$

This means that the marginal distributions of θ and p are independent, the marginal distribution of p is a d-dimensional Gaussian distribution with mean 0 and covariance M, and the marginal distribution of θ can have any continuously differentiable log-likelihood $-U$. Because of this, θ is used to represent the variables of interest, while p is a set of auxiliary variables used for the proposals.

Generating a (θ, p)-sample is done in two stages [Nea12], which are technically two separate Metropolis-Hastings proposals and acceptance tests. In the first stage, p is proposed from the Gaussian distribution with mean 0 and covariance M. Because the proposal matches the marginal distribution of p, it is always accepted.

In the second stage, Hamiltonian dynamics for the particle on a surface are simulated with the leapfrog method, starting from the current value of θ and the proposed value of p. The end result is a proposal for both θ and p. Because the leapfrog simulation preserves volumes and is reversible by negating p, the acceptance probability of the proposal is simply

$$\min\{1, \exp(H(\theta, p) - H(\theta', p'))\}.$$

If the simulation conserved H exactly, the acceptance probability would always be 1. Because this is usually not possible, there is always some probability of rejecting, which

depends on the length of the jump taken by the leapfrog method, η, and the number of leapfrog steps, L. It might be expected that the errors from the leapfrog steps would accumulate during the simulation, but in practice the errors in different steps tend to cancel each other [Nea12], meaning that the acceptance probability mainly depends on η.

4.2 Differential Privacy with HMC

HMC only uses the model log-likelihood, and thus the data, through U. U is used in two ways. Firstly, its value is used in the acceptance test, and secondly, its gradients are used to obtain proposals. Adding appropriate amounts of noise to both accesses would make HMC private, but the addition of noise may break some of the properties required for the algorithm's correctness.

The addition of noise to the log-likelihood ratios can be corrected using the penalty acceptance test, i.e. changing the acceptance probability to

$$\min\left\{\exp\left(H(\theta,p) - H(\theta',p') - \frac{1}{2}\sigma_l^2\right)\right\},$$

where σ_l^2 is the variance of the noise added to the log-likelihood ratios. This is because with $U(\theta) = -\ln p(X \mid \theta)$, the difference of the Hamiltonians in the acceptance probability is

$$H(\theta,p) - H(\theta',p') = \ln p(X \mid \theta') - \ln p(X \mid \theta) + \frac{1}{2}(p^T M^{-1} p - p'^T M^{-1} p').$$

As with the penalty algorithm, the sensitivity of the log-likelihood ratios must be bounded. If a bound does not exist, clipping must be used, nullifying the asymptotic converge guarantee of the algorithm. However, in Section 6.3 it is shown that clipping low numbers of gradients does not affect convergence in practice.

Releasing the gradients privately requires a trade-off. The leapfrog method remains reversible with noisy and potentially clipped gradients, as is shown in the following, but it no longer simulates the correct Hamiltonian dynamics [CFG14]. This does not affect asymptotic convergence, but it can potentially lower acceptance rates. The dynamics can be corrected by adding a friction term to the equations of motion of the simulation [CFG14], but this removes volume preservation of the simulation, which means that the acceptance probability must be corrected.

The leapfrog update equations with noisy and clipped gradients become

$$\begin{aligned} p_{i+1/2} &= p_i - \frac{\eta}{2}(\text{clip}(\nabla U(\theta_i)) + \mathcal{N}(0,\sigma_g^2)) \\ \theta_{i+1} &= \theta_i + \eta M^{-1} p_{i+1/2} \\ p_{i+1} &= p_{i+1/2} - \frac{\eta}{2}(\text{clip}(\nabla U(\theta_{i+1})) + \mathcal{N}(0,\sigma_g^2)), \end{aligned}$$

where the clip function clips the Euclidean norm of its argument to a given bound. The value of the bound is omitted here to keep the equations clearer.

The second leapfrog update equation is unchanged from the non-DP case, and remains both volume preserving and reversible by negating p. Both the first and third equations apply the function $l_p \colon \mathbb{R}^{2d} \to \mathbb{R}^{2d}$,

$$l_p(\theta, p) = \left(\theta, p - \frac{\eta}{2}(\text{clip}(\nabla U(\theta)) + \mathcal{N}(0, \sigma_{g^2}))\right).$$

to θ and p.

The Jacobian of l_p is

$$J_{l_p}(\theta, p) = \begin{bmatrix} I_{d\times d} & 0_{d\times d} \\ C(\theta) & I_{d\times d} \end{bmatrix},$$

where $I_{d\times d}$ and $0_{d\times d}$ are the d-by-d identity and zero matrices, respectively, and $C(\theta) = J_{\text{clip}(\nabla U(\theta))}(\theta)$. It is possible that $C(\theta)$ does not exist, but this can only happen in a null set* if U is sufficiently well-behaved. With such U, the non-existence of $C(\theta)$ does not affect the volume preservation of l_p. Appendix B gives very general sufficient conditions for U, and proves that all of the models considered in Chapter 5 meet the conditions.

Because $J_{l_p}(\theta, p)$ is triangular, $\det J_{l_p}(\theta, p) = 1$ for all θ and p, which means that l_p preserves volume. As the step updating θ also preserves volume, the entire leapfrog simulation preserves volume.

Showing that the leapfrog simulation is reversible by negating p requires more thought, as the simulation is no longer deterministic. Recall that the MH acceptance test uses the acceptance probability

$$\min\left\{1, \frac{\pi(\theta')}{\pi(\theta)}\frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)}\right\},$$

where π is the target density and q is the proposal density. For HMC, q must be understood as a Dirac δ function, as the standard leapfrog is deterministic. The reversibility of the leapfrog by negating p means that $q(\theta', p' \mid \theta, p) = q(\theta, -p \mid \theta', -p')$. If p were negated after the leapfrog simulation, one would have another proposal distribution q', with $q'(\theta', p' \mid \theta, p) = q'(\theta, p \mid \theta', p')$. Running HMC with proposal q' instead of q does not change the output of the algorithm, as p is not stored and the kinetic energy term in the Hamiltonian is $\frac{1}{2}p^T M^{-1} p$, so $H(\theta, p) = H(\theta, -p)$. This means that the proposal ratio in the MH acceptance probability is one for HMC.

With the noisy gradient, the leapfrog proposal is no longer deterministic, so the reversibility by negating p must be proven separately.

*A set of zero Lebesgue measure.

The proof requires defining some new terminology. Let $l\colon \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ be a function giving a potentially random proposal for given (θ, p). Because l is potentially random, it has a (generalized) density $q_l(\theta', p' \mid \theta, p)$, where

$$P(l(\theta, p) \in A) = \int_A \mathrm{d}(\theta', p') q_l(\theta', p' \mid \theta, p).$$

Now reversibility by negating p can be defined:

Definition 8. *A proposal function l is reversible by negating p if*

$$q_l(\theta', p' \mid \theta, p) = q_l(\theta, -p \mid \theta', -p')$$

This can be simplified by defining $\bar{l}$ as the the proposal function that has density $q_{\bar{l}}(\theta', p' \mid \theta, p) = q_l(\theta, p \mid \theta', p')$, and defining l_- as $l_-(\theta, p) = (\theta, -p)$. Definition 8 can then be written as $\bar{l} = l_- \circ l \circ l_-$.

Lemma 2. *Let l_1 and l_2 be proposal functions in $\mathbb{R}^{2d}$. Then $\overline{l_2 \circ l_1} = \bar{l_1} \circ \bar{l_2}$. For a sequence $l_1, \ldots, l_k$ of proposal functions, $\overline{l_k \circ \cdots \circ l_1} = \bar{l_1} \circ \cdots \circ \bar{l_k}$.*

Proof.

$$\begin{aligned}
q_{\overline{l_2 \circ l_1}}(\theta_3, p_3 \mid \theta_1, p_1) &= q_{l_2 \circ l_1}(\theta_1, p_1 \mid \theta_3, p_3) \\
&= \int_{\mathbb{R}^{2d}} \mathrm{d}(\theta_2, p_2) q_{l_1}(\theta_2, p_2 \mid \theta_3, p_3) q_{l_2}(\theta_1, p_1 \mid \theta_2, p_2) \\
&= \int_{\mathbb{R}^{2d}} \mathrm{d}(\theta_2, p_2) q_{\bar{l_1}}(\theta_3, p_3 \mid \theta_2, p_2) q_{\bar{l_2}}(\theta_2, p_2 \mid \theta_1, p_1) \\
&= q_{\bar{l_1} \circ \bar{l_2}}(\theta_3, p_3 \mid \theta_1, p_1)
\end{aligned}$$

which means that $\overline{l_2 \circ l_1} = \bar{l_1} \circ \bar{l_2}$. The claim for sequences follows by induction. □

Theorem 9. *The leapfrog simulation with noisy and clipped gradients is reversible by negating p.*

Proof. The leapfrog simulation is a composition of proposal functions l_p and l_θ where

$$l_p(\theta, p) = \left(\theta, p - \frac{\eta}{2}(\mathrm{clip}(\nabla U(\theta)) + \mathcal{N}(0, \sigma_g^2))\right)$$

and

$$l_\theta(\theta, p) = (\theta + \eta M^{-1} p, p).$$

Specifically, the simulation is

$$(l_p \circ l_\theta \circ l_p) \circ \cdots \circ (l_p \circ l_\theta \circ l_p)$$

with L compositions of $l_p \circ l_\theta \circ l_p$. Denoting $g(\theta) = \text{clip}(\nabla U(\theta))$,

$$\begin{aligned} q_{l_p}(\theta', p' \mid \theta, p) &= \delta_{\theta'}(\theta) \cdot \mathcal{N}\left(p' \mid p - \frac{\eta}{2} g(\theta), \frac{\eta^2}{4}\sigma_g^2\right) \\ &= \delta_{\theta}(\theta') \cdot \mathcal{N}\left(-p \mid -p' - \frac{\eta}{2} g(\theta), \frac{\eta^2}{4}\sigma_g^2\right) \\ &= q_{l_p}(\theta, -p \mid \theta', -p') \end{aligned}$$

and

$$\begin{aligned} q_{l_\theta}(\theta, -p \mid \theta', -p') &= \delta_\theta(\theta' - \eta M^{-1} p') \cdot \delta_{-p}(-p') \\ &= \delta_{\theta'}(\theta + \eta M^{-1} p) \cdot \delta_{p'}(p) \\ &= q_{l_\theta}(\theta', p' \mid \theta, p), \end{aligned}$$

which means that both l_p and l_θ are reversible by negating p. Then the reverse of the leapfrog composition can be written as

$$\begin{aligned} \overline{(l_p \circ l_\theta \circ l_p) \circ \cdots \circ (l_p \circ l_\theta \circ l_p)} &= (\bar{l}_p \circ \bar{l}_\theta \circ \bar{l}_p) \circ \cdots \circ (\bar{l}_p \circ \bar{l}_\theta \circ \bar{l}_p) \\ &= l_- \circ (l_p \circ l_\theta \circ l_p) \circ \cdots \circ (l_p \circ l_\theta \circ l_p) \circ l_-, \end{aligned}$$

where the first equality uses Lemma 2 and the fact that the sequence of compositions is symmetric. The second equality uses the fact that l_p and l_θ are reversible by negating p. This means that the leapfrog simulation with noisy and clipped gradients is reversible by negating p. □

If the friction term that corrects the dynamics is used, the equations of motion become [CFG14]

$$\begin{aligned} \frac{d\theta}{dt} &= M^{-1} p \\ \frac{dp}{dt} &= -\nabla U(\theta) - \frac{1}{2}\sigma_g^2 M^{-1} p + \mathcal{N}(0, \sigma_g^2) \end{aligned}$$

The equation for θ is unchanged, but the equation for p has the additional term $-\frac{1}{2}\sigma_g^2 M^{-1} p$. For the leapfrog updates, l_θ does not change, and l_p changes to

$$l_p(\theta, p) = \left(\theta, p - \frac{\eta}{2}(\text{clip}(\nabla U(\theta)) + \frac{1}{2}\sigma_g^2 M^{-1} p + \mathcal{N}(0, \sigma_g^2))\right).$$

l_p no longer preserves volumes, as the Jacobian of l_p is

$$J_{l_p}(\theta, p) = \begin{bmatrix} I_{d\times d} & 0_{d\times d} \\ C(\theta, p) & I_{d\times d} - \frac{\eta}{4}\sigma_g^2 M^{-1} \end{bmatrix}$$

which means that generally $|\det J_{l_p}(\theta, p)| \neq 1$.

Algorithm 4: DP HMC

$G_{1,0} = \text{grad}(\theta_0, b_g)$
for $1 \leq i \leq k$ **do**
  $p_0 = \mathcal{N}_d(0, M)$
  $\theta_{i,0} = \theta_{i-1}$
  for $1 \leq j \leq L$ **do**
    $p_{j-\frac{1}{2}} = p_{j-1} + \frac{1}{2}\eta G_{i,j-1}$
    $\theta_{i,j} = \theta_{i,j-1} + M^{-1} p_{j-\frac{1}{2}}$
    $G_{i,j} = \text{grad}(\theta_{i,j}, b_g)$
    $p_j = p_{j-\frac{1}{2}} + \frac{1}{2}\eta G_{i,j}$
  end
  $r_l = \ln \frac{p(\theta_{i,L}|x_l)}{p(\theta_{i-1}|x_l)}$
  $R = \sum_{l=1}^{n} \text{clip}_{b_l||\theta_{i-1}-\theta_{i,L}||_2}(r_l)$
  $s_{var} = (2\sigma_l b_l ||\theta_{i-1} - \theta_{i,L}||_2)^2$
  $\Delta H = R + \frac{1}{2} p_0^T M^{-1} p_0 - \frac{1}{2} p_L^T M^{-1} p_L + \ln \frac{p(\theta_{i,L})}{p(\theta_{i-1})} + \mathcal{N}(0, s_{var})$
  $u = \text{Unif}(0, 1)$
  if $u < \Delta H - \frac{1}{2} s_{var}$ **then**
    $\theta_i = \theta_{i,L}$
    $G_{i+1,0} = G_{i,L}$
  end
  else
    $\theta_i = \theta_{i-1}$
    $G_{i+1,0} = G_{i,0}$
  end
end
where

$$\text{grad}(\theta, b) = \sum_{i=1}^{n} \text{clip}_b(\nabla \ln p(\theta \mid x_i)) + \nabla \ln p(\theta)$$

and $\text{clip}_b(\theta)$ clips the Euclidean norm of θ to be less than or equal to b.

DP HMC is shown in Algorithm 4. Like the DP penalty algorithm, DP HMC is a composition of Gaussian mechanisms, and its privacy bounds can be computed through Theorems 2 and 4 for zCDP.

Theorem 10. *Let* $\epsilon \geq 0$, $0 \leq \delta < 1$, $\tau_g > 0$, $\tau_l > 0$, $b_g > 0$, $b_l > 0$,

$$\sigma_l(\theta, \theta') = 2\tau_l \sqrt{n} b_l ||\theta - \theta'||_2,$$

$$\sigma_g = 2\tau_g \sqrt{n} b_g,$$

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

$$\rho_l = \frac{1}{2\tau_l^2 n},$$

$$\rho_g = \frac{1}{2\tau_g^2 n},$$

Then DP HMC can be run for

$$k = \left\lfloor \frac{\rho - \rho_g}{\rho_l + L\rho_g} \right\rfloor$$

iterations using σ_l^2 as log-likelihood ratio noise variance, σ_g^2 as gradient noise variance, clipping log-likelihood ratios with b_l and clipping gradients by b_g.

Proof. Each iteration of the outer for-loop computes the gradient L times and the log-likelihood ratio once. The gradient is also computed once before the outer for-loop. Releasing a single gradient has zCDP privacy cost of

$$\rho_g = \frac{4b_g^2}{2\sigma_g^2} = \frac{1}{2\tau_g^2 n},$$

where $\sigma_g = 2\tau_g \sqrt{n} b_g$. Releasing the log-likelihood ratio of θ and θ' has privacy cost

$$\rho_l = \frac{4b_l^2 ||\theta - \theta'||_2^2}{2\sigma_l(\theta, \theta')^2} = \frac{1}{2\tau_l^2 n},$$

where $\sigma_l(\theta, \theta') = 2\tau_l \sqrt{n} b_l ||\theta - \theta'||_2$.

The total zCDP budget with given (ϵ, δ)-bound is

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

This means that the number of iteration that the algorithm is allowed to run for is

$$k = \left\lfloor \frac{\rho - \rho_g}{\rho_l + L\rho_g} \right\rfloor$$ □

zCDP based privacy accounting is loose, so the above bound could be improved by using the tight bound for the Gaussian mechanism from Theorem 4. Because DP HMC has both different sensitivities and different noise variances between the log-likelihood ratios and gradients, Theorem 4 is not directly applicable. However, the bound of Theorem 4 does generalise to differing variances between composition.

Theorem 11. *Let $\epsilon \geq 0$, $0 \leq \delta < 1$, $\tau_l > 0$, $\tau_g > 0$, $b_l > 0$, $b_g > 0$,*

$$\sigma_l'^2 = \tau_l^2 n,$$

$$\sigma_g'^2 = \tau_g^2 n.$$

Then running DP HMC for k iterations with L leapfrog steps, using $2b_l\sigma_l'^2||\theta-\theta'||_2$ as the log-likelihood ratio noise variance, $2b_g\sigma_g'^2$ as the gradient noise variance and b_l and b_g as the log-likelihood and gradient clip bounds is $(\epsilon, \delta(\epsilon))$-ADP where

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

and

$$\mu = \frac{k}{2\sigma_l'^2} + \frac{kL+1}{2\sigma_g'^2}.$$

Proof. Sommer et al. [SMM19] prove the ADP bound of Theorem 4 with the *privacy loss distribution* (PLD) of the Gaussian mechanism. First, they show that the PLD of the Gaussian mechanism with sensitivity Δ and variance σ^2 is a Gaussian distribution $\mathcal{N}(\mu, 2\mu)$, where $\mu = \frac{\Delta^2}{2\sigma^2}$ [SMM19, Lemma 11]. Next, they show that the ADP bounds $(\epsilon, \delta(\epsilon))$ for a Gaussian PLD $\mathcal{N}(\mu, 2\mu)$ are given by [SMM19, Lemma 12]

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

Finally, they show that the PLD of an adaptive composition is the convolution of the PLDs of the individual parts of the composition [SMM19, Theorem 1]. As convolution of distributions corresponds to summing random variables, the PLD of a convolution of Gaussian mechanisms with PLDs $\mathcal{N}(\mu_i, 2\mu_i)$ is simply $\mathcal{N}(\sum \mu_i, 2\sum \mu_i)$.

Because the sensitivity of the log-likelihood ratio is $2b_l||\theta-\theta'||_2$ and the sensitivity of the gradient is $2b_g$, both the gradient and log-likelihood ratio are divided by their sensitivities before adding noise, which normalises both to have sensitivity one. Adding noise with variance $\sigma_l^2(\theta, \theta')$ and σ_g^2 to the log-likelihood ratio and gradient respectively is equivalent to adding noise with variance $\sigma_l'^2 = \tau_l^2 n$ and $\sigma_g'^2 = \tau_g^2 n$. to the normalised log-likelihood ratio and gradient.

DP HMC releases the log-likelihood ratio k times and the gradient $kL+1$ times. This means that the PLD of DP HMC is a Gaussian distribution $\mathcal{N}(\mu, 2\mu)$ with

$$\mu = \frac{k}{2\sigma_l'^2} + \frac{kL+1}{2\sigma_g'^2}.$$ □

The maximum number of iterations DP HMC can run for can be computed with Algorithm 2 by using Theorem 11 to compute δ, and using Theorem 10 to initialise the variable *low*.

Figure 4.1 shows an example of a leapfrog trajectory of DP HMC on a circular posterior (Section 5.3). Even with noisy and clipped gradients, the trajectory is able to stay inside the thin, circular area of high probability, while moving a substantial distance on the circle.

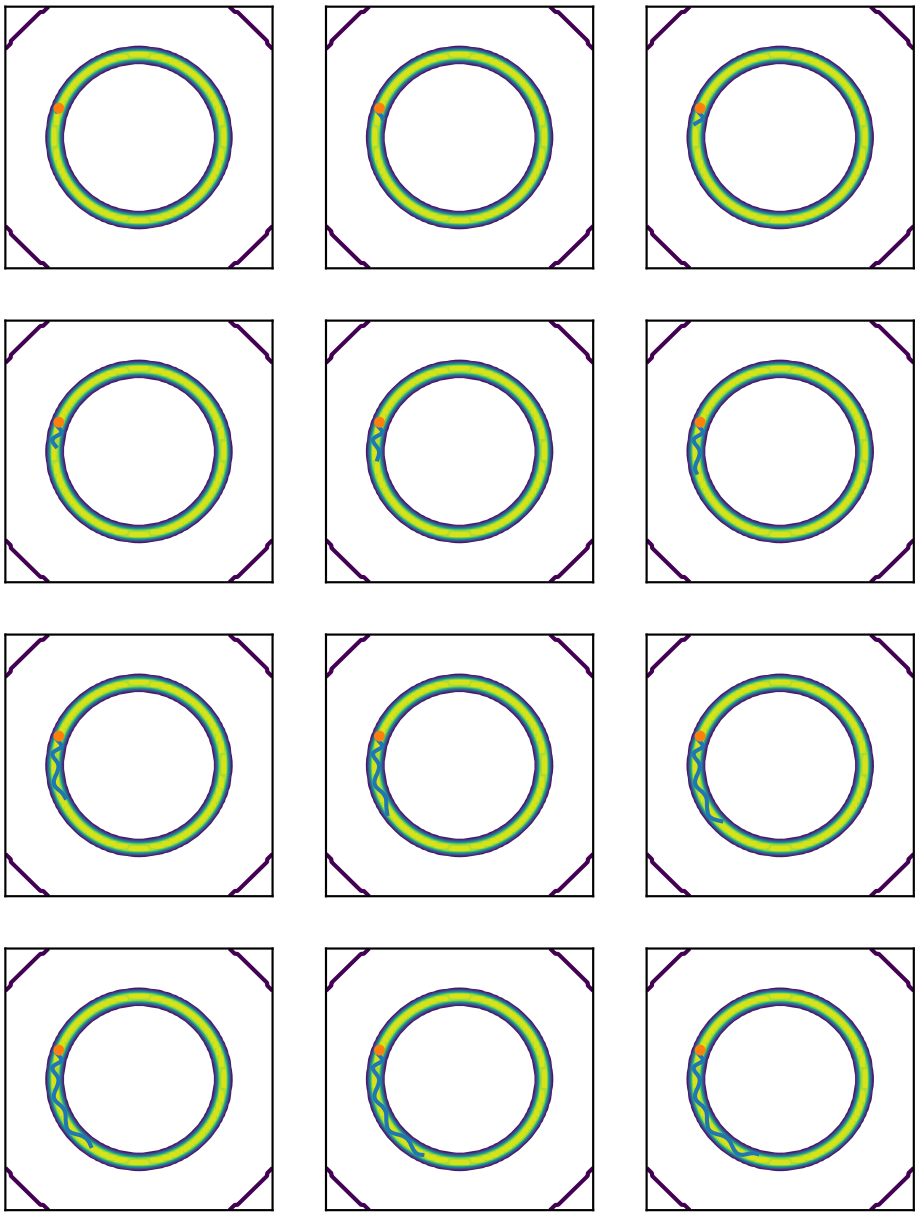

Figure 4.1: DP HMC proposal trajectory on a circular posterior (Section 5.3). The orange point is the starting point of the leapfrog simulation and the blue line shows the progress of the leapfrog steps. The background is a contour plot of the circular posterior density.

5. Evaluated Models

To compare the performance of the DP MCMC algorithms, they were run on several models. This chapter introduces these models in Sections 5.1, 5.2 and 5.3. Finally, the main performance metric used in the experiments is introduced in Section 5.4. The results and the specifics of the used models are discussed in Chapter 6.

5.1 The Gaussian Model

The simplest model used for the experiments is the Gaussian model with known covariance. The likelihood and prior for n points of d-dimensional data $X \in \mathbb{R}^{n \times d}$ and parameters $\theta \in \mathbb{R}^d$ are

$$\begin{aligned} \theta &\sim \mathcal{N}_d(\mu_0, \Sigma_0), \\ X_i &\sim \mathcal{N}_d(\theta, \Sigma), \end{aligned}$$

where X_i is a row of X, Σ is the known covariance, and μ_0 and Σ_0 are the prior hyperparameters. The posterior of this model is another d-dimensional Gaussian distribution with parameters expressible in closed form [GCS+14, Section 3.5], so it is easy to sample from the posterior.

5.2 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 9. *Let X have a d-variate Gaussian distribution with mean μ and covariance matrix Σ. Let*

$$g(x) = (x_1, x_2 - a(x_1 - m)^2 - b, x_3, \ldots, x_d),$$

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 12 gives a suitable inference problem for testing DP MCMC algorithms.

Theorem 12. *Let*

$$\begin{aligned}\theta = (\theta_1, \dots, \theta_d) &\sim \mathrm{Ban}(0, \sigma_0^2 I, a, b, m),\\ X_1 &\sim \mathcal{N}(\theta_1, \sigma_1^2),\\ X_2 &\sim \mathcal{N}(\theta_2 + a(\theta_1 - m)^2 + b, \sigma_2^2),\\ X_3 &\sim \mathcal{N}(\theta_3, \sigma_3^2),\\ &\vdots\\ X_d &\sim \mathcal{N}(\theta_d, \sigma_d^2).\end{aligned}$$

Given data $x_1, \dots, x_d \in \mathbb{R}^n$ *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*

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

$$\mu = \left(\frac{Tn\tau_1\bar{x}_1}{Tn\tau_1 + \tau_0}, \dots, \frac{Tn\tau_d\bar{x}_d}{Tn\tau_d + \tau_0}\right),$$
$$\Sigma = \mathrm{diag}\left(\frac{1}{Tn\tau_1 + \tau_0}, \dots, \frac{1}{Tn\tau_d + \tau_0}\right).$$

Proof. See Appendix A. □

The Gaussian distribution is a special case of the banana distribution, as setting $a = 0$ makes g the identity function. Similarly, setting $a = 0$ turns the Bayesian banana model into a Gaussian model.

5.3 Circle Model

Random walk MH algorithms may struggle with posteriors which are concentrated on long, but thin regions. Having a large proposal variance causes the algorithm to frequently jump out of the region of high probability, but lowering the variance to stay in the region causes the chain to take a very long time to move around in the region. An example of such a posterior is the circular posterior described in this section.

The circle posterior is obtained from a model where the log-likelihood is

$$\ln p(r \mid x, y) = -a(x^2 + y^2 - r^2)^2$$

where $r \in \mathbb{R}$ is an observed data point and $(x, y) \in \mathbb{R}^2$ are the parameters. The likelihood is circular in the (x, y)-plane, as it only depends on the squared distance of the point (x, y) from the origin. By choosing a flat prior, $p(x, y) = 1$ for all $(x, y) \in \mathbb{R}^2$, the posterior will be proportional to the likelihood, meaning that the posterior will also be circular.

As the prior does not integrate to 1, it is not a proper probability density, so Bayes' theorem does not guarantee that the posterior integrates to a finite value. In this case, for a single data point r,

$$\begin{aligned}
\int_{\mathbb{R}^2} p(r \mid x, y)\mathrm{d}x\mathrm{d}y &= \int_{\mathbb{R}^2} e^{-a(x^2+y^2-r^2)^2}\mathrm{d}x\mathrm{d}y \\
&= \int_0^{2\pi} \mathrm{d}\phi \int_0^{\infty} \mathrm{d}t \cdot t e^{-a(t^2(\cos^2\phi+\sin^2\phi)-r^2)^2} \\
&= \int_0^{2\pi} \mathrm{d}\phi \int_0^{\infty} \mathrm{d}t \cdot t e^{-a(t^2-r^2)^2} \\
&= \int_0^{2\pi} \mathrm{d}\phi \int_0^{\infty} \mathrm{d}u \cdot e^{-a(u-r^2)^2} \\
&< \infty.
\end{aligned}$$

As the inner integral is over an unnormalised Gaussian density and the outer integral is over a finite interval. For multiple data points, the integral of the likelihood is

$$\int_{\mathbb{R}^2} \prod_i^n p(r_i \mid x, y)\mathrm{d}x\mathrm{d}y \leq \prod_i^n \left(\int_{\mathbb{R}^2} p(r_i \mid x, y)^n \mathrm{d}x\mathrm{d}y \right)^{\frac{1}{n}} < \infty,$$

where the first inequality is Hölder's inequality applied $n - 1$ times, and the second follows from the single data point case, as $p(r \mid x, y)^n = e^{-an(x^2+y^2-r^2)^2}$. As the likelihood integrates to a finite value, it can be considered an unnormalised density and sampled from with MCMC.

Obtaining samples from the true posterior with the circle model is not as easy as with the banana and Gaussian models, so using MMD, the metric introduced in Section 5.4 to evaluate the performance of an MCMC algorithm on the circle is nontrivial. However, as the mean of the circle is in its center, and the samples from an MCMC algorithm are likely to lie on the circle, the distance of the mean of the sample from the origin indicates how evenly the samples a distributed throughout the circle.

It remains to show that the mean of the circle is actually the origin. It turns out

that this is fairly simple. For the mean of the x-coordinate

$$\begin{aligned}
E_x &= \int_{\mathbb{R}} \mathrm{d}x \cdot x \int_{\mathbb{R}} \mathrm{d}y p(r \mid x, y) \\
&= \int_{-\infty}^{\infty} \mathrm{d}x \cdot x \int_{-\infty}^{\infty} \mathrm{d}y \prod_i e^{-a(x^2+y^2-r_i^2)^2} \\
&= \int_{0}^{\infty} \mathrm{d}x \cdot x \int_{-\infty}^{\infty} \mathrm{d}y \prod_i e^{-a(x^2+y^2-r_i^2)^2} + \int_{-\infty}^{0} \mathrm{d}x \cdot x \int_{-\infty}^{\infty} \mathrm{d}y \prod_i e^{-a(x^2+y^2-r_i^2)^2} \\
&= \int_{0}^{\infty} \mathrm{d}x \cdot x \int_{-\infty}^{\infty} \mathrm{d}y \prod_i e^{-a(x^2+y^2-r_i^2)^2} - \int_{0}^{\infty} \mathrm{d}x \cdot x \int_{-\infty}^{\infty} \mathrm{d}y \prod_i e^{-a(x^2+y^2-r_i^2)^2} \\
&= 0.
\end{aligned}$$

The mean of the y-coordinate is calculated similarly.

5.4 Maximum Mean Discrepancy

The convergence of non-DP MCMC algorithms is typically assessed using $\hat{R}$ [GCS+14], which measures how well multiple chains started from different points have mixed together. The utility of a sample produced by an MCMC algorithm can be evaluated using effective sample size (ESS) [GCS+14], which is an estimate of the size of an uncorrelated sample of the posterior with the same estimation utility as the MCMC sample.

Both $\hat{R}$ and ESS require that the MCMC algorithm asymptotically targets the true posterior, as they cannot detect an algorithm that has converged to the wrong distribution. Because some of the DP MCMC algorithms use approximations that may cause the algorithms to not converge to the true posterior, such as clipping the log-likelihood ratios, $\hat{R}$ and ESS are not suitable for assessing the performance of the algorithms.

Because the DP MCMC algorithms may not converge to the correct distribution, their performance should be evaluated with a metric that measures how close to the true distribution they are. A very general such metric is maximum mean discrepancy (MMD) [GBR+12]. MMD between distributions p and q is defined as

$$\mathrm{MMD}(p, q) = \sup_{f \in \mathcal{F}} (E_{x \sim p} f(x) - E_{y \sim q} f(y))$$

where $\mathcal{F}$ is some class of functions. By choosing a suitable $\mathcal{F}$, $\mathrm{MMD}(p, q)$ can be estimated from a sample from p and q. The suitable classes $\mathcal{F}$ can be characterised by a kernel function $k \colon P \times Q \to \mathbb{R}$, where P and Q are the supports of p and q, respectively. The Gaussian radial basis function (RBF) kernel

$$k(x, y) = \exp\left(\frac{||x - y||_2^2}{2\sigma^2}\right)$$

is particularly well suited, as it has the property that $\mathrm{MMD}(p, q) = 0$ if and only if $p = q$. After choosing a kernel, $\mathrm{MMD}(p, q)$ may be estimated from finite samples of p and q. To evaluate MCMC algorithms, one of the samples is the output of the algorithm to be evaluated, and the other is a sample from the true posterior.

The choice of the σ parameter of k affects the way MMD evaluates different kinds of differences in p and q. For some preliminary experiments in this thesis, σ was chosen to be 1, which penalised error in the mean much more than errors in higher moments in the experiments. The σ used for the final experiments is chosen by picking 50 subsamples from both samples with replacement and setting σ to be the median between distances of points of the subsamples. This is following the procedure of Gretton et al. [GBR+12], with the addition of the subsampling step to handle samples of different sizes.

6. Experiments

This chapter contains the experiments performed on both non-DP and DP MCMC algorithms. Section 6.1 details the specific model parameters and other technical details of the experiments. Section 6.3 examines the effect clipping log-likelihood ratios, which is necessary for DP MCMC algorithms, but may adversely affect their convergence. Section 6.4 compares the different DP MCMC algorithms on several models.

6.1 Experimental Setup

Eight models were used in the experiments. The parameters are shown in Table 6.1 and contour plots of the 2-dimensional models are shown in Figure 6.1. All banana models had $b = m = 0$. The likelihood variance of the banana and 30-dimensional Gaussian models was $\sigma_1^2 = 20$, $\sigma_2^2 = 2.5$ and $\sigma_i^2 = 1$ for $i > 2$. The likelihood covariance for the correlated Gaussian model was

$$\begin{bmatrix} 1 & 0.999 \\ 0.999 & 1 \end{bmatrix}.$$

All experiments ran 20 chains for each value of ϵ for DP MCMC experiments, or clip bound for clipping experiments, for each algorithm included. Each of the 20 chains had a different starting point, but the starting points did not vary across algorithms, or values of ϵ or clip bound.

The first half of the obtained samples were discarded as they may not represent the true posterior, which is standard practice with MCMC [GCS+14]. The latter half was compared to a reference sample obtained from the true posterior, except in the circle experiment, where the mean of the sample was compared to the true posterior mean of $(0, 0)$.

Publicly available implementations of DP Barker[†] and the Fourier accountant[‡] were used, and the rest of the algorithms were implemented by the author. The code for running the experiments is freely available[§].

[†] https://github.com/DPBayes/DP-MCMC-NeurIPS2019
[‡] https://github.com/DPBayes/PLD-Accountant
[§] https://github.com/oraisa/masters-thesis

Table 6.1: Model parameters. n_0 determines tempering by $T = \frac{n_0}{n}$. For missing n_0, $T = 1$. Start deviation is the standard deviation of the random starting point in the DP experiments. a is the parameter determining how curved the banana distribution is, with $a = 0$ corresponding to a Gaussian distribution.

Name	Dim	n	n_0	Start Deviation	a	σ_0^2
Flat banana, d = 2	2	100000		0.02	20	1000
Flat banana, d = 10	10	200000		0.02	20	1000
Tempered banana, d = 2	2	100000	1000	0.02	20	1000
Tempered banana, d = 10	10	200000	1000	0.02	20	1000
High dimensional Gauss	30	200000		0.02	0	1000
Narrow banana	2	150000		0.02	350	1000
Correlated Gauss	2	200000		0.005	0	100
Circle	2	100000		0.1	1e-05	

6.2 Comparing Privacy Accounting Methods

DP penalty, minibatch DP penalty and DP HMC have two different methods to compute the number of iterations the algorithm can run for given a privacy bound and parameters for the algorithm. These privacy accounting methods are given by Theorems 5 and 6 for DP penalty, Theorems 10 and 11 for DP HMC, and Theorem 8 and the Fourier accountant [KJH20] for minibatch DP penalty.

Figure 6.2 compares the number of iterations each of the algorithms can run for for both accounting methods and different values of ϵ in the 2-dimensional flat banana experiment. The ADP based methods of Theorems 6 and 11, as well as the Fourier accountant, significantly outperform the other accounting methods. This makes it clear that the tight bounds given by the ADP based methods should be used in favor of loose methods. The Fourier accountant is not easily applicable to DP Barker, as DP Barker does not release the sample variance directly, it may still be possible to use the Fourier accountant with DP Barker. This is an interesting question for future research, as using a better privacy accountant may significantly improve the performance of DP Barker.

6.3 The Effects of Clipping

The first experiment evaluates the effect of clipping log-likelihood ratios. Both HMC and random walk Metropolis-Hastings (RWMH)* algorithms were run on 2 and 10 dimensional banana models. DP was not used so that error from the extra noise would not affect the results.

For both 2 and 10 dimensions, 500 samples from HMC and 3000 samples from RWMH were taken and the latter half of them were compared to 2000 samples from the true posterior. The reference posterior sample was also compared to other samples from the posterior to obtain a baseline. The sample sizes and other parameters of the algorithms were tuned so that the algorithms converged without clipping.

Figure 6.3 shows the results of the clipping experiment. The top left and bottom left panels show MMD as a function of the clip bound and the fraction of log-likelihoods that was actually clipped for both HMC and RWMH in the 2-dimensional model. The effect of clipping on MMD is nonexistent for all but the lowest clip bounds. The top and bottom right panels show results for the 10-dimensional model. This time there are chains that did not converge correctly with most clip bounds, but the chains with the higher bounds converged. Based on these results, if the clip fraction is less than

*Metropolis-Hastings using the Gaussian distribution as the proposal distributions.

10%, clipping is likely undetectable without a large sample.

6.4 Comparison of DP MCMC Algorithms

The experiments in this section compare the DP MCMC algorithms discussed in this thesis. The compared algorithms are the DP penalty algorithm with and without guided walk MH and one component updates [YE19] (Section 3.1), the DP Barker algorithm [HJDH19] (Section 3.2), the penalty algorithm with subsampling, again with and without GWMH and OCU (Section 3.3) and DP HMC (Section 4.2).

The δ for all experiments is $\frac{0.1}{n}$, and ϵ is varied. All algorithms used the best privacy accounting methods discussed in this thesis. For both variants of DP penalty and DP HMC, these are the PLD based Theorems 6 and 11. DP penalty with subsampling uses the Fourier accountant [KJH20] and DP Barker uses Theorem 7. The RDP based theorems for the minibatch algorithms are not tight for the ADP bounds that were used, so their results could be improved by using an accounting method for ADP.

The hyperparameters of the algorithms were tuned by running the algorithms with $\epsilon = 4$ and examining the results, trying to find hyperparameters that minimise MMD. Clip bounds were tuned so that less than 10% of the log-likelihood ratios were clipped, as the results of Section 6.3 show minimal effect on MMD at that point.

Figure 6.4 shows the MMD for each algorithm as a function of ϵ for the flat and tempered banana models with 2 and 10 dimensions. In the non-tempered models, the minibatch algorithms are not very useful, with the exception of DP Barker in 10 dimensions. Of the non-minibatch algorithms, with tempering HMC beats the penalty algorithms, but without tempering this is reversed.

Figure 6.5 shows MMDs for the more complicated models, the 30-dimensional Gaussian, the narrow banana and the highly correlated Gaussian. The minibatch algorithms were not included as the previous experiment indicates that they perform poorly without tempering. In the 30-dimensional Gaussian, DP penalty with OCU and GWMH beats the other two algorithms, with DP HMC narrowly beating DP penalty. In the narrow banana all algorithms are approximately equal, until $\epsilon = 4$ where DP penalty0 OCU+GWMH has huge variance in MMD, and in the higher values of ϵ where it performs extremely poorly. In the highly correlated Gaussian experiment, the penalty algorithms perform equally, and HMC beats both of them.

Figure 6.6 shows the fraction of log-likelihood ratios that were clipped for each ϵ and algorithm. Almost all runs had less than 10% clipping, with the exception of DP Barker, as adjusting its clip bound is not possible.

Figure 6.7 shows clipping for the harder models. Again, almost all runs have less than 10% clipping, with the exception of DP penalty OCU+GWMH on the narrow

banana, where the high amount of clipping could explain the poor performance of the algorithm with larger values of ϵ.

Figure 6.8 shows the results for the circle model. The distance of the sample mean from the true mean (the origin) were used instead of MMD, as computing MMD requires a sample from the posterior, which is not trivial to obtain for the circle model. Both algorithms perform well on average, with HMC slightly beating DP penalty on $\epsilon = 0.5$. Clipping is again low for both algorithms, and HMC has a fairly large variance in clipping between different runs.

Figure 6.9 shows the fraction of clipped gradients for DP HMC. Clipping gradients does not affect the asymptotic convergence of DP HMC, but it may lower the acceptance rate and thus the performance of the algorithm. As raising the clip bound to lower gradient clipping increases the noise added to gradient, thus also lowering the acceptance rate, it is not clear what an appropriate level of gradient clipping would be. The observed levels of clipping are fairly constant across values of ϵ, but vary in the models, and in the narrow banana model, there is very large variance in clipping.

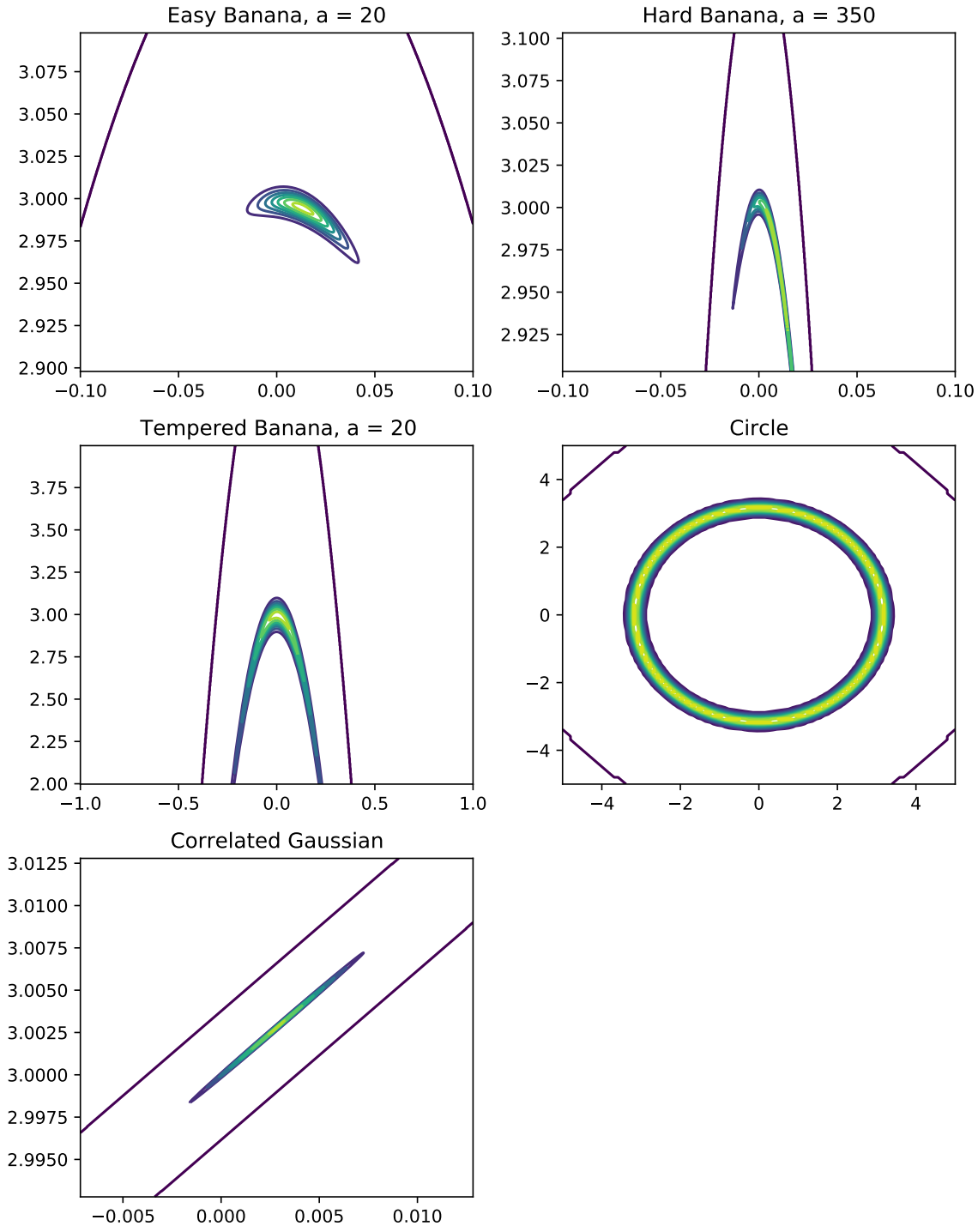

Figure 6.1: Contour plots of the posterior densities of the 2-dimensional models.

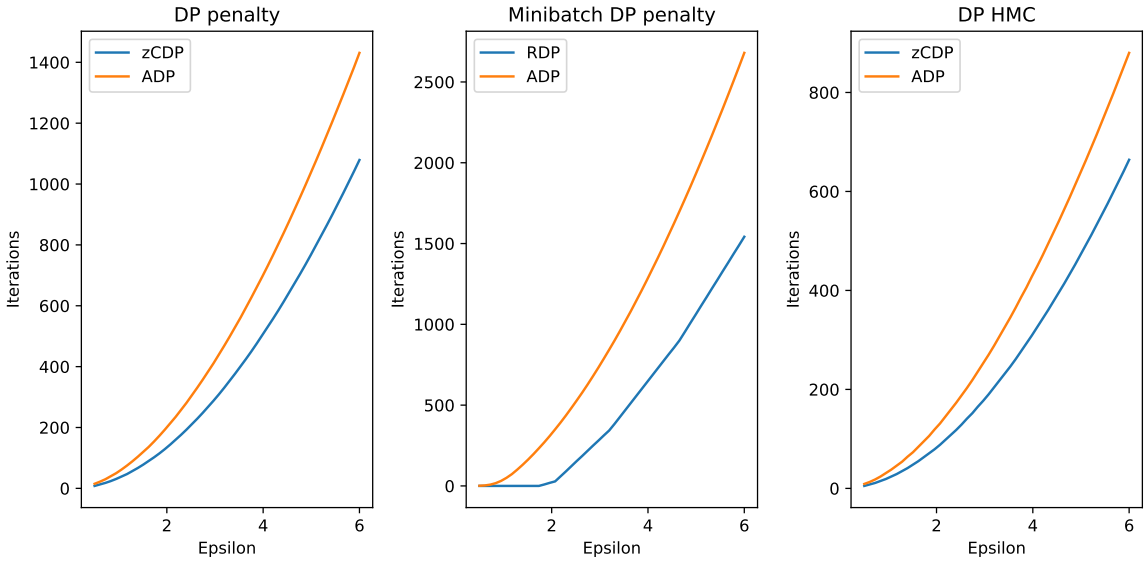

Figure 6.2: Comparing the zCDP or RDP and ADP based privacy accounting methods for the algorithms with multiple accounting methods. The left panel compares zCDP based accounting (Theorem 5) and ADP based accounting (Theorem 6) for the DP penalty algorithm. The middle panel compares the RDP accounting (Theorem 8) and the Fourier accountant. The right panel compares the zCDP (Theorem 10) and ADP (Theorem 11) based accounting. The ADP based methods significantly outperform the other methods in all cases.

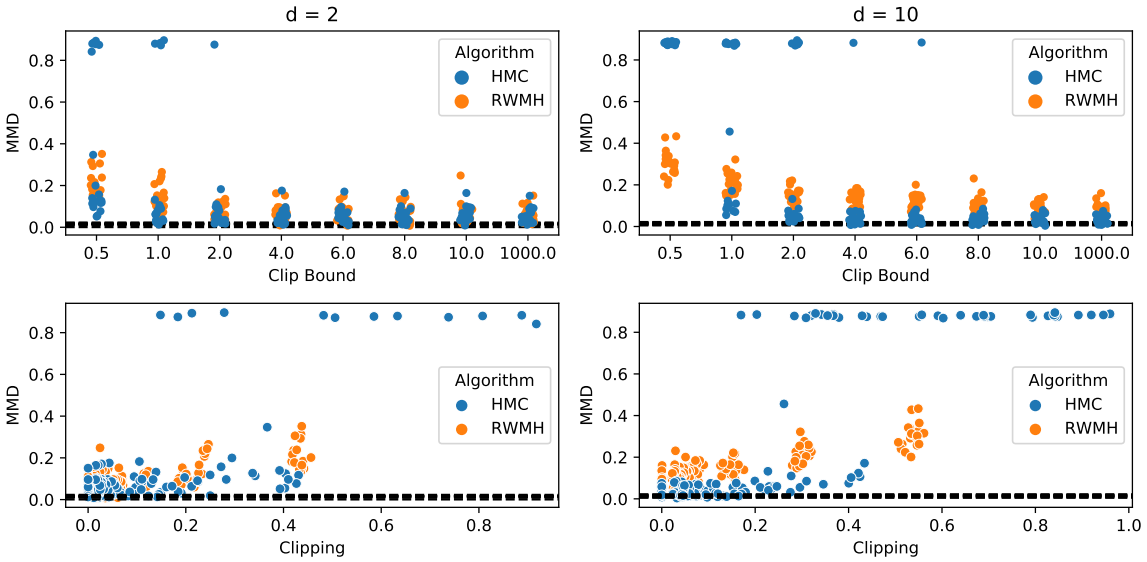

Figure 6.3: The effect of log-likelihood ratio clipping on the posterior of the banana model for random walk Metropolis-Hastings and HMC. The top row shows posterior MMD as a function of the clip bound, and the bottom row shows MMD as a function of the fraction of log-likelihoods that were clipped. The left columns used a 2-dimensional posterior while the right columns had a 10-dimensional posterior. The black lines show the MMDs of ten different samples of the true posterior compared to the reference sample. All of the lines are close the each other and appear as a single line.

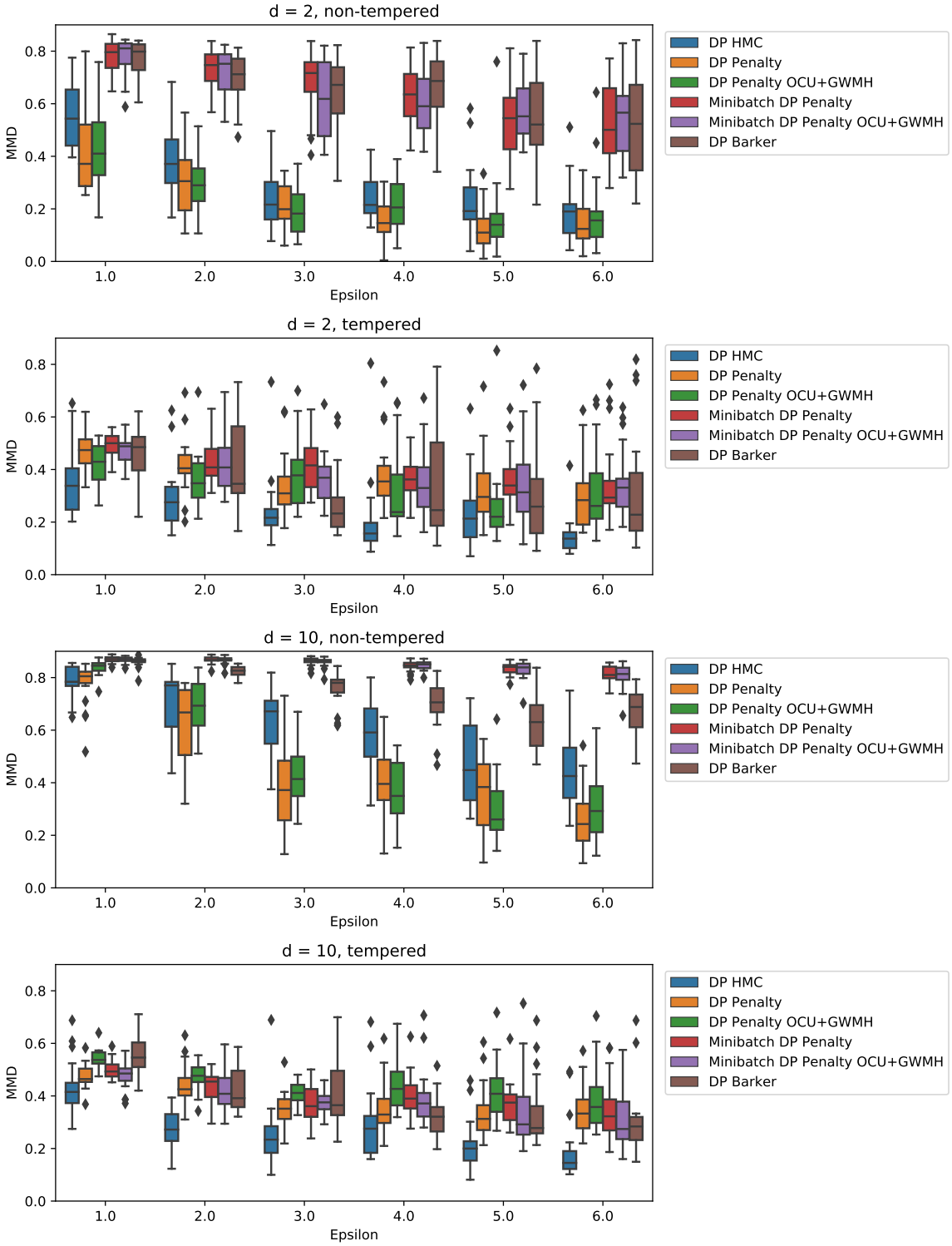

Figure 6.4: MMD as a function of ϵ for the different MCMC algorithms, on flat and tempered banana models with both a low number of dimensions (2) and a high number of dimensions (10).

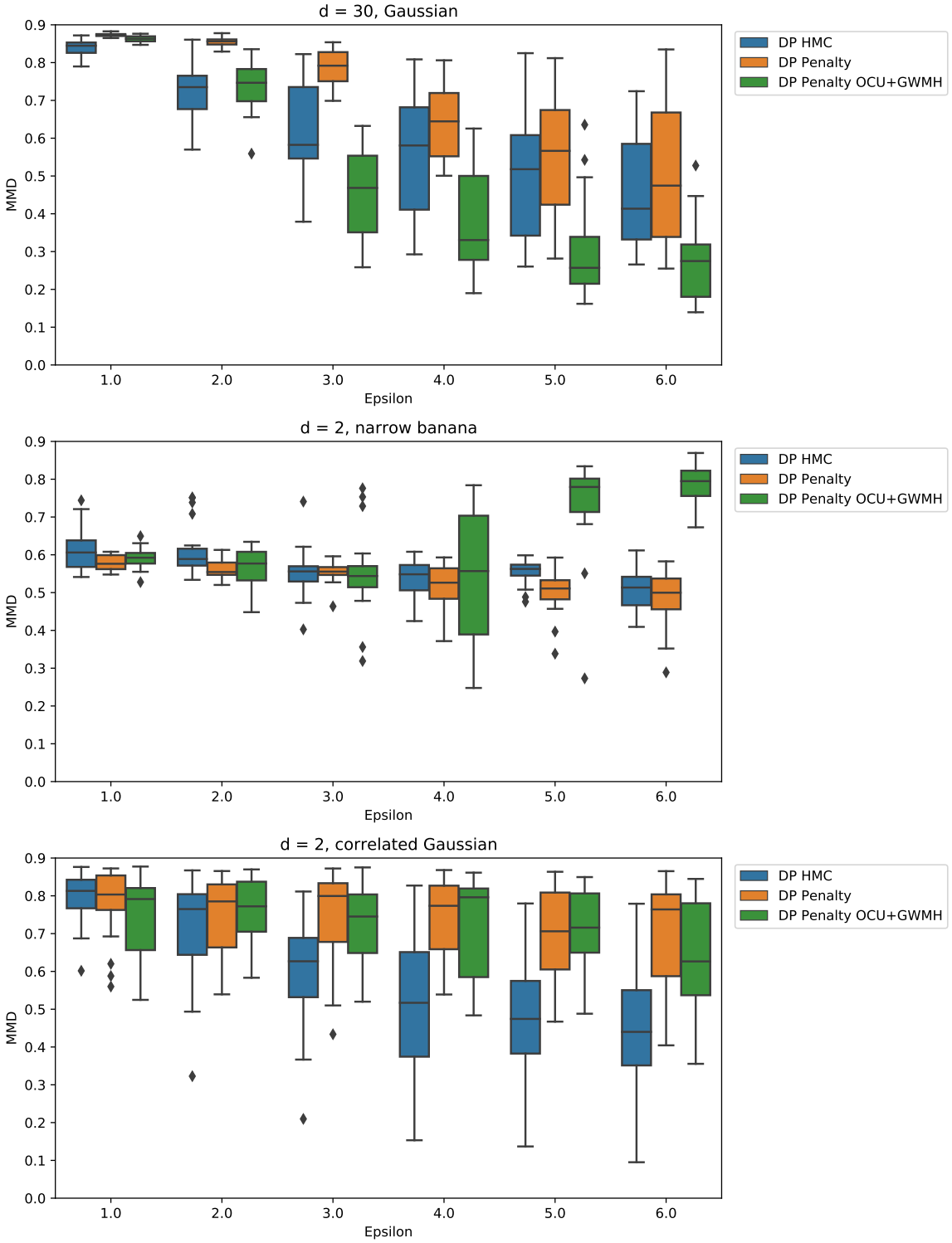

Figure 6.5: MMD for the 30-dimensional Gaussian, hard banana and highly correlated 2-dimensional Gaussian.

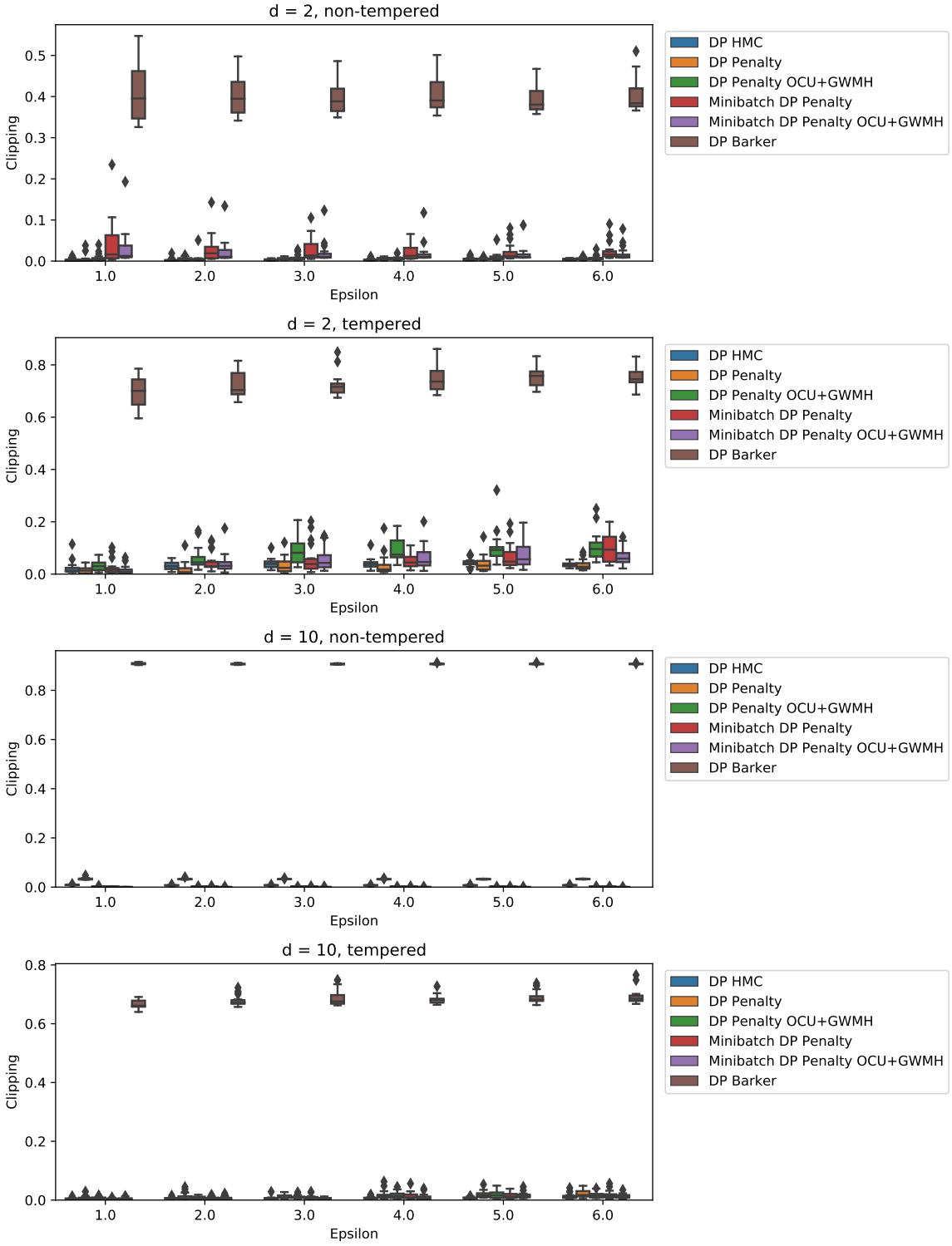

Figure 6.6: Clipping for the easy and tempered banana experiments.

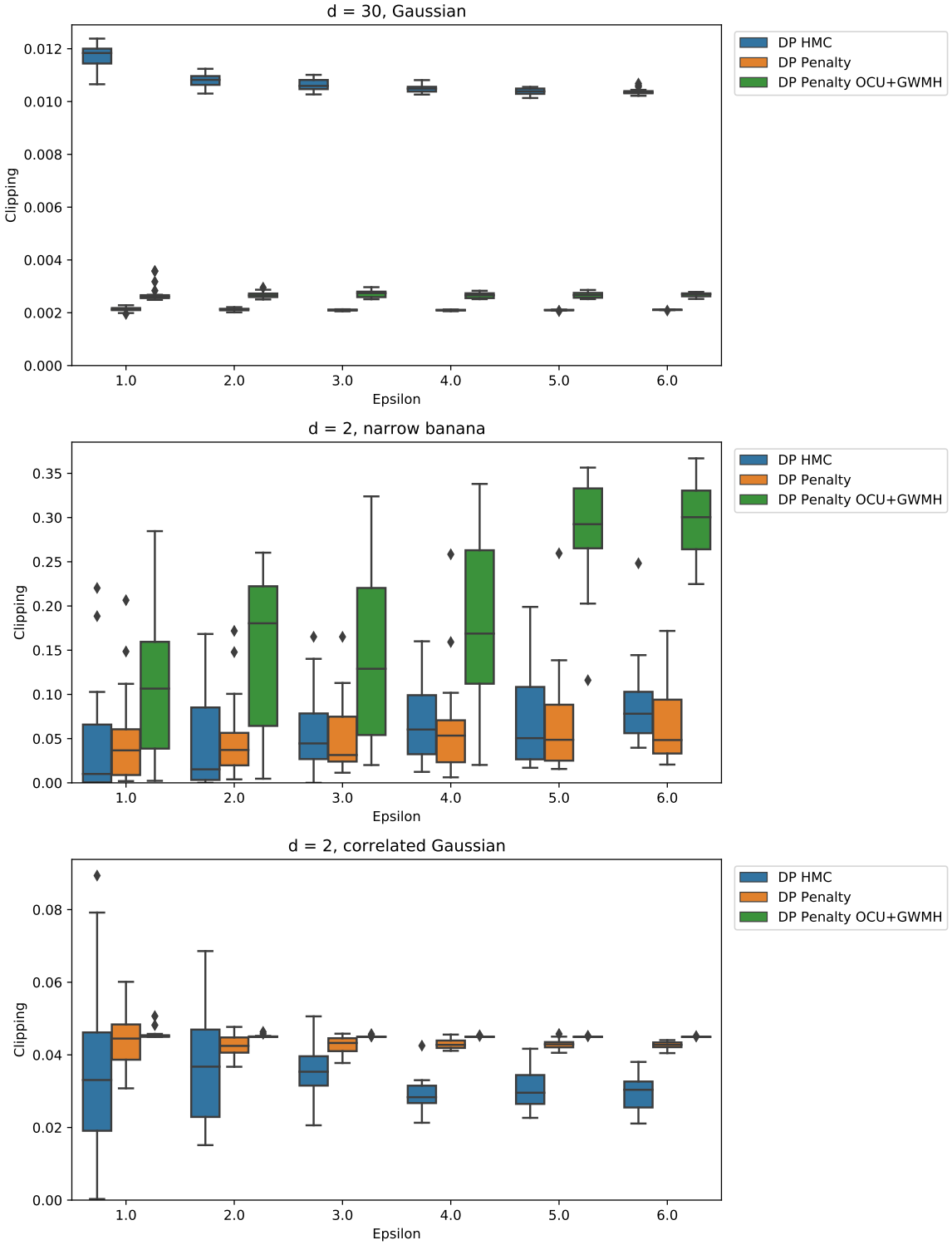

Figure 6.7: Clipping for 30-dimensional Gaussian, hard banana and highly correlated 2-dimensional Gaussian.

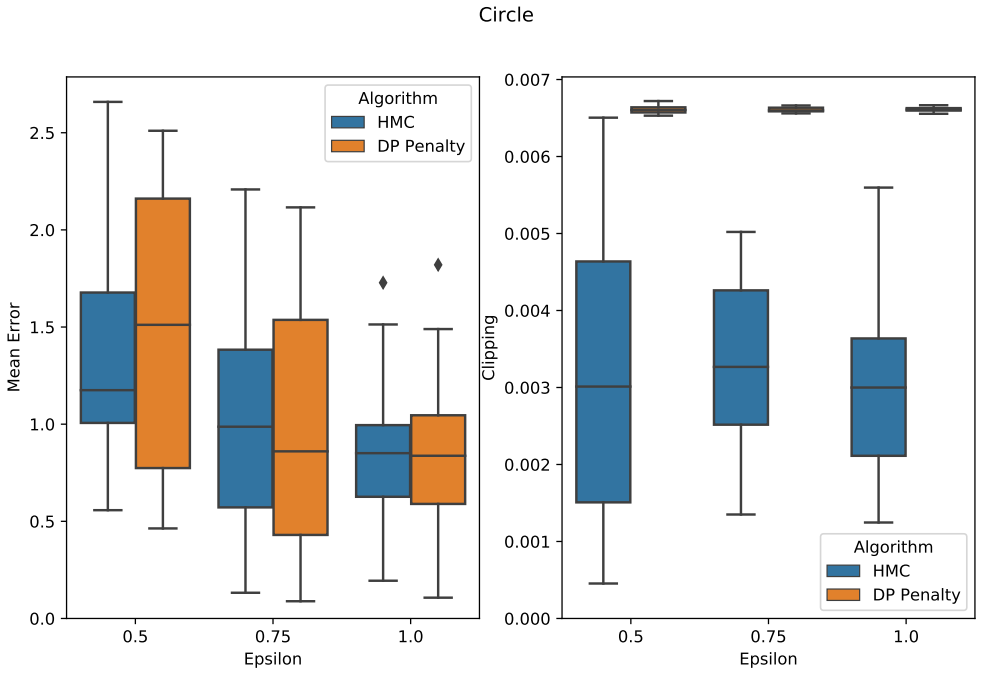

Figure 6.8: Circle mean error on the left and clipping on the right.

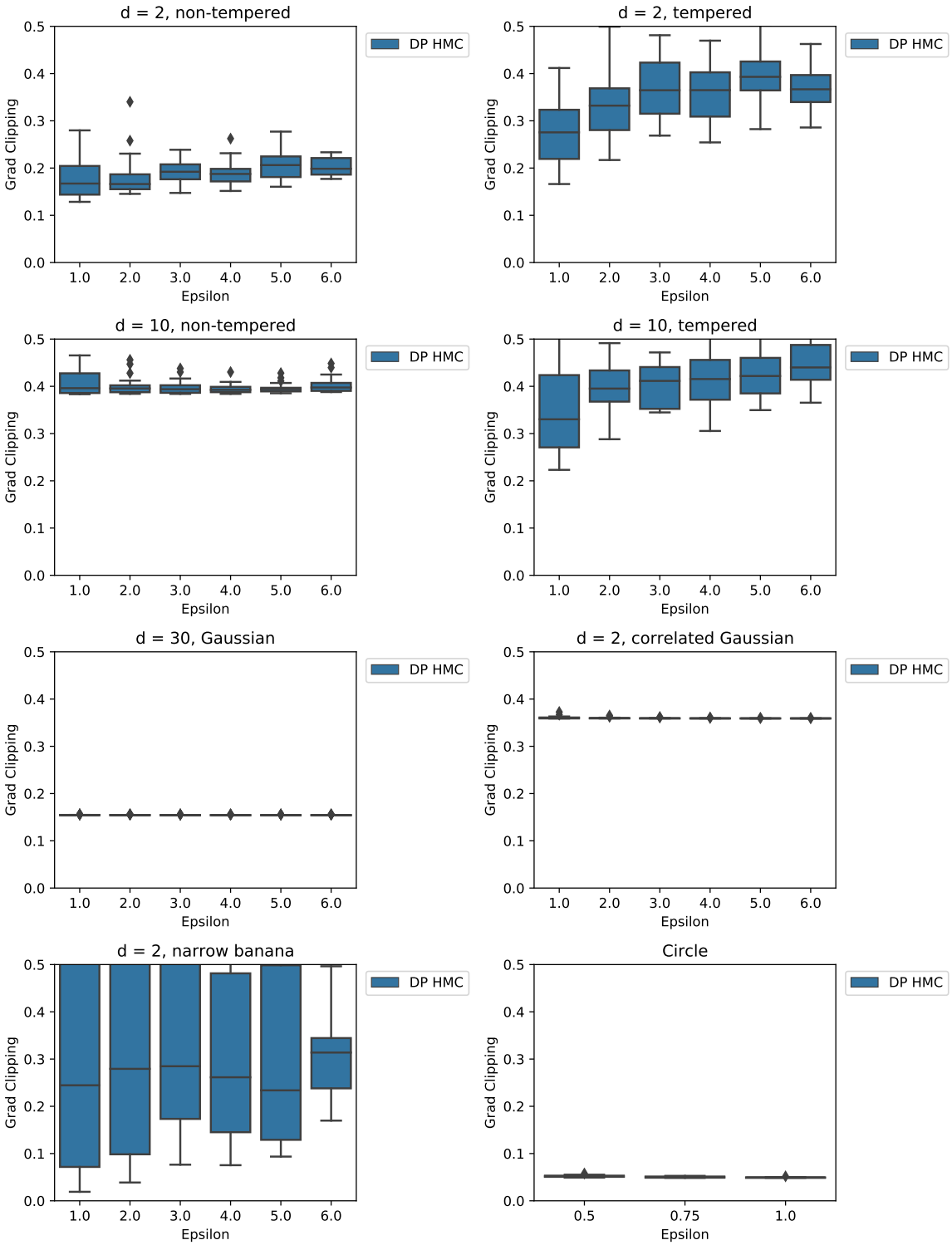

Figure 6.9: Gradient clipping for DP HMC.

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.

[CFG14] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1683–1691. JMLR.org, 2014.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[DKPR87] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography*

Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[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.

[GBR+12] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012.

[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.

[Has70] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. cited By 7759.

[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.

[KJH20] Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2560–2569. PMLR, 2020.

[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.

[MRR+53] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[Nea12] Radford M. Neal. MCMC using Hamiltonian dynamics, 2012.

[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.

[SPCC17] Daniel Seita, Xinlei Pan, Haoyu Chen, and John F. Canny. An efficient minibatch acceptance test for metropolis-hastings. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[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.

[Tu11] Loring W Tu. *Introduction to Manifolds*. Universitext. Springer New York, New York, 2011.

[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.

Appendix A. Proof of Theorem 12

Theorem 12. *Let*

$$\begin{aligned}\theta = (\theta_1, \ldots, \theta_d) &\sim \mathrm{Ban}(0, \sigma_0^2 I, a, b, m)\\ X_1 &\sim \mathcal{N}(\theta_1, \sigma_1^2)\\ X_2 &\sim \mathcal{N}(\theta_2 + a(\theta_1 - m)^2 + b, \sigma_2^2)\\ X_3 &\sim \mathcal{N}(\theta_3, \sigma_3^2)\\ &\vdots\\ X_d &\sim \mathcal{N}(\theta_d, \sigma_d^2)\end{aligned}$$

Given data $x_1, \ldots, x_d \in \mathbb{R}^n$ *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*

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

$$\mu = \left(\frac{Tn\tau_1 \bar{x}_1}{Tn\tau_1 + \tau_0}, \ldots, \frac{Tn\tau_d \bar{x}_d}{Tn\tau_d + \tau_0}\right),$$

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

Proof. Because

$$g^{-1}(y) = (y_1, y_2 + a(y_1 - m)^2 + b, y_3 \ldots, y_d)$$

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

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

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

$$\exp\left(-\frac{1}{2}\left(\left(\frac{x_1 - \mu_1}{\sigma_1}\right)^2 + \left(\frac{x_2 + a(x_1 - m)^2 + b - \mu_2}{\sigma_2}\right)^2 + \sum_{i=3}^{d}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right)\right)$$

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

$$
\begin{aligned}
p(\theta \mid X) &\propto p(X \mid \theta)^T p(\theta) \\
&= p(X_1 \mid \theta_1)^T p(X_2 \mid \theta_1, \theta_2)^T \prod_{i=3}^{d} p(X_i \mid \theta_i)^T p(\theta) \\
&= p(X_1 \mid \theta_1)^T p(X_2 \mid \theta_1, \theta_2)^T \prod_{i=3}^{d} p(X_i \mid \theta_i)^T \\
&\cdot \exp\left(-\frac{1}{2}\left(\tau_0\theta_1^2 + \tau_0(\theta_2 + a(\theta_1 - m)^2 + b)^2 \sum_{i=3}^{d} \tau_0\theta_i^2\right)\right) \\
&= p(X_1 \mid \theta_1)^T p(X_2 \mid \theta_1, \theta_2)^T \exp\left(-\frac{1}{2}\left(\tau_0\theta_1^2 + \tau_0(\theta_2 + a(\theta_1 - m)^2 + b)^2\right)\right) \\
&\cdot \prod_{i=3}^{d} p(X_i \mid \theta_i)^T \exp\left(-\frac{1}{2}\sum_{i=3}^{d} \tau_0\theta_i^2\right)
\end{aligned}
$$

Considering the upper and lower part of the last expression separately

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

and

$$
\begin{aligned}
&\prod_{i=3}^{d} p(X_i \mid \theta_i)^T \cdot \exp\left(-\frac{1}{2}\sum_{i=3}^{d} \tau_0 \theta_i^2\right) \\
&\propto \exp\left(-\frac{1}{2} T \sum_{j=3}^{d} \tau_j \sum_{i=1}^{n} (x_{ij} - \theta_j)^2 - \frac{1}{2}\sum_{j=3}^{d} \tau_0 \theta_j^2\right) \\
&= \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left(T\tau_j \sum_{i=1}^{n}(x_{ij} - \theta_j)^2 + \tau_0\theta_j^2\right)\right) \\
&= \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left(T\tau_j \sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2 + T\tau_j n(\bar{x}_j - \theta_j)^2 + \tau_0\theta_j^2\right)\right) \\
&\propto \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left(T\tau_j n(\bar{x}_j - \theta_j)^2 + \tau_0\theta_j^2\right)\right) \\
&\propto \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left(-2T\tau_j n\bar{x}_j\theta_j + T\tau_j n\theta_j^2 + \tau_0\theta_j^2\right)\right) \\
&\propto \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left((Tn\tau_j + \tau_0)\theta_j^2 - 2Tn\tau_j\bar{x}_j\theta_j + \frac{(Tn\tau_j\bar{x}_j)^2}{Tn\tau_j + \tau_0}\right)\right) \\
&= \exp\left(-\frac{1}{2}\sum_{j=3}^{d}(Tn\tau_j + \tau_0)\left(\theta_j - \frac{Tn\tau_1\bar{x}_i}{Tn\tau_j + \tau_0}\right)^2\right)
\end{aligned}
$$

Multiplying the resulting expression above gives a density proportional to the banana distribution. 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

$$
\mu = \left(\frac{Tn\tau_1\bar{x}_1}{Tn\tau_1 + \tau_0}, \ldots, \frac{Tn\tau_d\bar{x}_d}{Tn\tau_d + \tau_0}\right),
$$

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

Appendix B. Differentiability of the clip-function

This appendix proves that functions of the form $\text{clip} \circ f$ are almost everywhere differentiable for sufficiently well-behaved f, as required by the proof of volume preservation of DP HMC leapfrog iterations (Section 4.2). Lemma 4 gives very general sufficient conditions, and Lemma 5 proves that the models considered in this thesis meet the conditions.

Lemma 3. *Let $d \geq 2$ and $g\colon U \to \mathbb{R}$, where U is an open subset of $\mathbb{R}^d$, be continuously differentiable. Let $S = \{x \in U \mid f(x) = b\}$, $b \in \mathbb{R}$. If for all $x \in S$, $\nabla f(x) \neq 0$, S is a $(d-1)$-dimensional hypersurface.*

Proof. See [Tu11, Section 9.2]. □

Lemma 4. *Let $f\colon \mathbb{R}^d \to \mathbb{R}^d$ be continuously differentiable. If the set of saddle points U of $||f||$ is a null set, the set $A \subset \mathbb{R}^d$ where $\text{clip}_b \circ f$ is not differentiable is a null set.*

Proof.

$$\text{clip}_b(x) = \frac{x}{||x||} \min\{||x||, b\}$$

which means that clip_b is differentiable for all $x \in \mathbb{R}^d$ with $||x|| \neq b$. Consider $\text{clip}_b \circ f$ in a neighbourhood B of point $x_0 \in \mathbb{R}^d$. If $||f(x)|| \leq b$ for all $x \in B$, $\text{clip}_b(f(x)) = f(x)$ in B, which is differentiable. If $||f(x)|| \geq b$ for all $x \in B$, $\text{clip}_b(f(x)) = \frac{f(x)b}{||f(x)||}$ in B, which is also differentiable because $f(x) \neq 0$ when $||f(x)|| \geq b > 0$. In both cases $\text{clip}_b \circ f$ is differentiable at x_0. This means that for all $x_0 \in A$, $||f(x_0)|| = b$, but every neighborhood of x_0 has points x' and x'' such that $||f(x')|| < b$ and $||f(x'')|| > b$. With the differentiability of f, this means that if $\nabla||f(x_0)|| = 0$, x_0 is a saddle point of $||f||$.

Let $S = \{x \in \mathbb{R}^d \mid ||f(x)|| = b \text{ and } \nabla||f(x)|| \neq 0\}$. By Lemma 3, S is a $(d-1)$-dimensional hypersurface, which has zero measure in d-dimensional space. Because $A \cap U^C \subset S$, $A \cap U^C$ is also a null set. Since $A = (A \cap U^C) \cup (A \cap U)$, A is a null set. □

For DP HMC, set $f(\theta) = \nabla U(\theta)$. The conditions of Lemma 4 for U are met if U is twice continuously differentiable and

Lemma 5. *The log-likelihoods of the Gaussian, banana and circle models meet the conditions of Lemma 4.*

Proof. For DP HMC, set $f(\theta) = \nabla U(\theta)$. The conditions of Lemma 4 for U are met if U is twice continuously differentiable and

$$\nabla||\nabla U(x)|| = \frac{2\nabla U(x)}{||\nabla U(x)||} J_{\nabla U}(x) = \frac{2\nabla U(x)}{||\nabla U(x)||} H_U(x) \neq 0$$

almost everywhere, where H_U is the Hessian matrix of U. Because $\nabla U = 0$ and $||\nabla U|| = 0$ at only one point, having $\det H_U(x) \neq 0$ almost everywhere is sufficient.

For the Gaussian log-likelihood

$$U(\theta) = \frac{1}{2}(x-\theta)^T\Sigma^{-1}(x-\theta),$$

so

$$\nabla U(\theta) = \Sigma^{-1}(x-\theta)$$

and $H_U(\theta) = \Sigma^{-1}$.

For the banana log-likelihood, using the notation from Appendix A,

$$U(\theta) = \frac{1}{2}(g^{-1}(x)-\theta)^T\Sigma^{-1}(g^{-1}(x)-\theta),$$

so

$$\nabla U(\theta) = \Sigma^{-1}(g^{-1}(x)-\theta)$$

and $H_U(\theta) = \Sigma^{-1}$.

The circle log-likelihood is

$$U(x,y) = -a(x^2+y^2-r^2)^2,$$

$$\nabla U(x,y) = -4a(x^3+xy^2-xr^2, x^2y+y^3-yr^2),$$

so

$$H_U(x,y) = -4a\begin{bmatrix} 3x^2+y^2-r^2 & 2xy \\ 2xy & 3y^2+x^2-r^2 \end{bmatrix}$$

and

$$\begin{aligned}\det H_U(x,y) &= -4a((3x^2+y^2-r^2)(3y^2+x^2-r^2)-4x^2y^2) \\ &= -4a(9x^2y^2+3x^4-3x^2r^2+3y^4+y^2x^2-y^2r^2-3y^2r^2-x^2r^2+r^4-4x^2y^2) \\ &= -4a(6x^2y^2+3x^4-4x^2r^2+3y^4-4y^2r^2+r^4) \\ &= -4a(6x^2y^2+3(x^4+y^4)-4r^2(x^2+y^2)+r^4).\end{aligned}$$

$$\nabla \det H_U(x,y) = -4a(12xy^2+12x^3-8r^2x, 12yx^2+12y^3-8r^2y),$$

which is zero in the origin and on the circle

$$x^2+y^2 = \frac{2}{3}r^2.$$

Elsewhere, Lemma 3 can be applied to $\det H_U$, which means that the set where $\det H_U(x,y) = 0$ is a 1-dimensional curve. □