Beyond Canonical MCMC: Preconditioning, Adaptivity, and Variational Approximations

Max Harwood Hird

For the degree of Doctor of Philosophy at the Department of Statistical Science, University College London

Declaration

I, Max Harwood Hird confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Since its conception in the middle of the twentieth century Markov chain Monte Carlo (MCMC) has grown into a suite of methods that serve as the de facto algorithmic solutions to a particular set of scientific problems. In this time a selection of variants of MCMC have effectively become canonical through their popularity and use within software packages. These include the Random Walk Metropolis (RWM), the Metropolis Adjusted Langevin Algorithm (MALA), and Hamiltonian Monte Carlo (HMC). It has become increasingly clear that there are certain properties of probability distributions, such as high dimensionality, multimodality, and ill-conditioning, in the face of which these canonical algorithms will perform poorly. In this thesis we explore three methods: preconditioning, adaptivity, and variational approximation, that can enhance the performance of the canonical kernels in order to overcome these obstacles.

Chapter 1 serves as a theoretical and conceptual introduction to these canonical kernels. Chapter 2 introduces the three methods which can be used to enhance the kernels described in chapter 1.

In chapter 3 we examine linear preconditioning. This is the practice of applying a linear transformation to the target distribution to make it easier to sample from. Its success is measured by a quantity known as the condition number, denoted κ. We assert verifiable conditions under which linear preconditioning will change κ and make a given MCMC sampler more efficient. We identify a case in which a commonly used linear preconditioner will cause sampler performance to worsen.

In chapter 4 we propose a novel way to combine a variational approximation to the target distribution with an arbitrary underlying MCMC kernel in order to reduce the variance of the estimators we derive from the MCMC chain. We call our method ‘the occlusion process’. We state the analytic form of the variance of the estimators it produces. We prove that it inherits numerous beneficial properties from the underlying MCMC kernel, such as a Law of Large Numbers, Geometric Ergodicity, and a Central Limit Theorem. We demonstrate empirically the occlusion process’ decorrelation and variance reduction capabilities on two target distributions. The first is a bimodal Gaussian mixture model in 1d and 100d. The second is the Ising model on an arbitrary graph, for which we propose a novel variational distribution.

In chapter 5 we propose a linear preconditioner that is learned and used in an adaptive MCMC algorithm. The preconditioner is conceived so that it can capture correlations between the directions of the target distribution. It is structured so that the resulting adaptive MCMC algorithm operates at a per-iteration computational complexity which is linear in the dimension of the state space. We show that our proposed adaptive algorithm dominates the competing methods in terms of its efficiency per unit time when operating on high-dimensional ill-conditioned target distributions.

Contents

List of Tables, Figures, and Algorithms

- Figure 3.2: $\log$ ESS of 100 runs at 10,000 iterations per run of RWM with dense, diagonal, and zero preconditioning. Each algorithm is started in equilibrium and targets $\mathcal{N}(0, \Sigma_\pi)$ with Σ_π as in 3.5.

- Figure 3.6.2: Boxplots of the medians of the ESSs across configurations of (n, d) with different preconditioners on the Bayesian linear regression with a Hyperbolic prior. The leftmost boxplot in each grouping corresponds to preconditioning with $L = \sigma(X^T X)^{1/2}$ ('A' in the legend), the middle boxplot has $L = \Sigma_\pi^{-1/2}$ ('covariance' in the legend), the rightmost has $L = \mathbf{I}_d$ ('none' in the legend).

- Figure 3.4: Boxplots of the logarithms of the medians of the ESSs across combinations of (d, μ). The ESSs are taken from RWM runs on a binomial regression target with the generalised g-prior. 'covariance' and 'covarianceII' correspond to runs preconditioned with $L = \Sigma_\pi^{-1/2}$ where Σ_π is estimated over 10^4 and 10^5 runs respectively. 'Fisher' and 'FisherII' correspond to runs preconditioned with $L = \mathbb{E}_\pi[\nabla^2 U(\beta)]^{1/2}$ where $\mathbb{E}_\pi[\nabla^2 U(\beta)]$ is estimated over 10^4 and 10^5 runs respectively. 'mode' refers to runs preconditioned with $L = \nabla^2 U(\beta^*)^{1/2}$ where β^* is an estimate of the mode found using preconditioned gradient descent. 'sq_root_Sigma_X' corresponds to runs preconditioned with $L = (n^{-1} X^T X)^{1/2}$.

- Figure 4.1: Three versions of the state space X; the leftmost with the Markov chain $\{X_t\}$ and the middle with the samples $\{Y_t\}$ taken from the target restricted to the regions that the Markov chain visits. We assume that we were only able to successfully sample Y_2 and Y_4, therefore the rightmost picture shows the samples we will use for the occlusion estimator: X_2 and X_4 have been *occluded* by Y_2 and Y_4.

- Figure 4.2: The top line is the estimator constructed using states of the Markov chain $\{X_t\}$. The middle picture is a DAG representing the occlusion process: $\{X_t\}$ is the Markov chain, $\rho(X_t)$ denotes the regions visited by the Markov chain, $\{Y_t\}$ are the samples from the target restricted to those regions, and $\{S_t\}$ indicate which of those samples we were able to successfully produce. The bottom line is the occlusion estimator made up of the samples from the Markov chain, and the successfully produced Y_t's.

- Figure 4.3: A DAG representing the process in which every state X_t in the Markov chain is replaced by a sample from $\pi_{\rho(X_t)}$ in the estimator.

- Figure 4.4: The target density (P in the legend), the variational density (Q in the legend), and the Radon-Nikodym derivative (dP/dQ in the legend) for the bimodal Gaussian example in $d = 1$.

- Figure 4.5: Results from the $d = 1$ case of sampling from (4.10) (left) and the $d = 100$ case (right). The top row shows the first component of the RWM chains, the next row shows the first component of $\{\mathbb{1}\{S_t = 0\}X_t + \mathbb{1}\{S_t = 1\}Y_t\}_{t=1}^{n}$, the next row shows autocorrelation function plots of the first component of the RWM chains, the bottom row shows autocorrelation function plots of the first component of $\{\mathbb{1}\{S_t = 0\}X_t + \mathbb{1}\{S_t = 1\}Y_t\}_{t=1}^{n}$.

- Algorithm 4.2: Metropolis algorithm applied to the Ising model.

- Algorithm 4.3: Wolff algorithm applied to the Ising model.

- Figure 4.6: Left: (V, E), right: $(\tilde{V}, \tilde{E})$. The colours correspond to the V_i's in the original (V, E) that get collapsed to nodes in $(\tilde{V}, \tilde{E})$. Where there exist any edges between two V_i's in (V, E), there is an edge between the corresponding nodes in $(\tilde{V}, \tilde{E})$.

- Figure 4.7: Three graphs comparing the performance of the occlusion process with the Metropolis and Wolff algorithms on the Ising model at a variety of temperatures, for a variety of graph sizes. In every case the horizontal axes show the number of vertices N in the graphs. Bottom: the vertical axes denote the algorithm's estimates of the expected magnetisation. Top left: the vertical axes denote the lag 1 autocorrelation coefficient of the magnetisation over the states produced by the algorithms. Top right: the vertical axes show the number of samples from the π_i's in Algorithm 4.1 divided by the number of states in the Markov chain n. We magnify the estimated magnetisation plot for ease of comprehension, the top two plots then help to explain the phenomena in the bottom plot.

- Algorithm 2.1: Generic adaptive MCMC algorithm.

- Algorithm 5.1: Complete learning step for the proposed adaptive algorithm.

- Algorithm 5.2: Generic multiple chain adaptive algorithm.

axes show the number of samples from the π_i's in Algorithm 4.1 divided by the number of states in the Markov chain n. We magnify the estimated magnetisation plot for ease of comprehension, the top two plots then help to explain the phenomena in the bottom plot.

- Figure 5.1: The action of $H = \mathbf{I}_d - 2nn^T$ on v. Let $n \in \mathbb{R}^d$ be a unit normal to a plane about which we would like to reflect $v \in \mathbb{R}^d$. Adding $-nn^T v$ to v projects it onto the plane and adding another $-nn^T v$ sends it to its reflection.

- Figure 5.2: A visual explanation of why $Q_k e_i = v_i$ for all $i < k$. By construction we have that $Q_k e_i = H\left(Q_{k-1}e_k \leftrightarrow v_k\right) Q_{k-1}e_i$ and so by the properties of Householder matrices if $Q_{k-1}e_i$ is perpendicular to both $Q_{k-1}e_k$ and v_k we have that $Q_k e_i = Q_{k-1}e_i$ which is just v_i by inductive hypothesis. Clearly $Q_{k-1}e_i$ is perpendicular to $Q_{k-1}e_k$ since the two vectors are just canonical basis vectors transformed by a Householder matrix, which is orthogonal. This transformation is shown by the reflection of the blue vectors to the green vectors through the dotted line in the figure. That $Q_{k-1}e_i$ is perpendicular to v_k is evident because $Q_{k-1}e_i = v_i$ by inductive hypothesis, and the fact that v_i is perpendicular to v_k.

Impact Statement

The algorithms discussed, analysed, and developed in this work are in current use both in profit-seeking ventures and in pure scientific research. Their primary role is in inference, the results of which are used to inform decision. Therefore the analysis presented here will affect these decisions by informing practitioners as to the efficiency and therefore the veracity of their inference procedures. The original algorithms which we develop offer alternative inference procedures which may be preferable under the settings we describe due to the increase in performance which they offer. More broadly we offer a tranche of knowledge that may be developed subsequently within academic computational statistics.

Chapter	Content	Dependencies	Collaborators	ArXiv preprint	Submitted to
1	Intro and Review	-	-	-	-
2	Intro and Review	-	-	-	-
3	Original Work	Assumption 24	Dr. Samuel Livingstone	[Hird and Livingstone 2024]	JMLR
4	Original Work	Section 1.2.2.3	Dr. Florian Maire	[Hird and Maire 2024]	JMLR
5	Original Work	-	Dr. Samuel Livingstone	-	-

Table 1: List of chapters along with the type of content contained within them, which previous parts in the text they depend on, collaborators, ArXiv preprints, and the journals which they are in submission to (JMLR stands for the Journal of Machine Learning Research).

Locations of original work, dependencies, and collaborators

See table 1 for the location of original work, the dependencies between the chapters, the collaborators with which the original work was completed, references to the ArXiv preprints corresponding to the original work, and the venue the work has been submitted.

The general development and completion of the work herein was undertaken with the sage guidance and supervision of Dr. Samuel Livingstone.

Notation

A per-chapter breakdown of the notation is collected in appendix 7.1. The notation in later chapters is inherited from the notation in earlier chapters.

Acknowledgements

I would first like to acknowledge the constant support of my supervisor Samuel Livingstone who, over the course of the PhD, has always entertained my thoughts, however incoherent, with serious consideration. I would also like to thank my pseudo-supervisor Florian Maire who has approached our collaborations with a lightness and kindness but also a profound sense of mathematical intuition.

I would like to thank the faculty at UCL statistics for constituting what is probably the best community of statisticians on earth. In particular thanks to Terry Soo for your energetic conversation, Alex Watson for

your quiet kindness, and Codina Cotar for being superlatively supportive (and for your cakes and hats). I would also like to thank Kayvan Sadeghi, Petros Dellaportas, Ioanna Manolopoulou, Omar Rivasplata, Francois-Xavier Briol, Brieuc Lehmann, Jeremias Knoblauch, Giampiero Marra, and Simon Harden for your friendliness and stimulating conversation.

I would like to single out Francois-Xavier, and also George Deligiannidis for applying great discernment and sagacity in examining this thesis, and for making the process of defending it so enjoyable.

From the wider staff at UCL I would like to thank Marina Lewis for your tireless support and Dom for the continued friendship and laughs. The current and former PhD students in UCL stats are too numerous to name, but I would like to thank you all for providing me with a great sense of community, especially after the concept had deteriorated under COVID.

I have an outstanding group of friends who, luckily for me, I could not possibly list here, but special thanks should go to Klara and Katarina Slater, Owen and Matt Mears, Matilda Slight and Jeremy Hatchuel for providing what is effectively the optimal friendship experience.

Thanks also to my assorted non-UCL academic friends: Giorgos Vasdekis for your constant levity, Adrien Corenflos, Raiha Browning, Nico Hernández, Gecia Bravo, Harita Dellaporta, Sebastiano Grazzi, Paolo Ceriani and all the Queen Mary PhD mathematicians for putting up with me, with especial thanks to María Fernanda Pintado Serrano for ~~having so many names~~ hosting me with such kindness and magnanimity. I would also like to thank Sam Power for your unique combination of deep erudition, humour, and considerateness. I would like to thank every single one of Peter Orbanz' PhD students (Hugh Dance, Vince Velkey, Vasco Portilheiro) with special mention going to Kevin Huang, with whom I have greatly enjoyed sharing many mathematical diversions.

Lastly I would like to thank my sister for her sociality and companionship, my uncle for his endless conversation, my grandma for her wit and curiosity, and my parents for their unwavering support and love.

Chapter 1

Introduction

1.1 Probability measures in science

The use of probability measures is commonplace in the practice of modern-day science. This is because non-determinism is an inevitable feature of our interactions with the natural world. Reality may itself be non-deterministic and even if it weren't our measurement devices are noisy and so introduce randomness into the detection of natural phenomena. Randomness can result as a feature of the fact that our physical theories are coarse-grainings of complicated underlying processes. Scientists also use probability measures to represent their credences in theories, and in specific parameter values within these theories. Given a coherent representation of credences in theories and parameter values, policy makers can then make informed actions in the world. In any case, it is imperative that we build the appropriate tools to help scientists learn about probability measures.

1.1.1 What to learn about probability measures

If we are to build tools to learn about probability measures, we must first ask what scientists want to know about the measures. As it happens, many questions about probability measures often have answers which naturally take the form of expectations with respect to those measures. For instance 'What is our credence in theory T given data X?' has the answer $\mathbb{P}(\mathsf{T}\,|X) = \mathbb{E}[\mathbb{1}\{\text{'T is the correct theory of the natural world'}\}\,|X]$. 'What is the probability of event A given theory T?' has the answer $\mathbb{P}(A\,|\mathsf{T}) = \mathbb{E}[\mathbb{1}\{A\}\,|\mathsf{T}]$. 'What is the loss $\mathcal{L}(\theta)$ due to parameter θ given data X and induced credences over theories $\mathsf{T}\,|X$?' has the answer $\mathbb{E}_{\mathsf{T}}[\mathbb{E}_\theta[\mathcal{L}(\theta)\,|\mathsf{T}]\,|X]$[1]. Our tools should then equip us to estimate expectations. In what follows the probability measure which the expectations are taken with respect to will be referred to as the 'target' or 'target distribution'. We will often abuse notation and refer to both a probability measure and its density with respect to the Lebesgue measure by the same name. The notation in this chapter can be found to be defined in section 7.1.1.

1.2 Classical sampling methods

1.2.1 Estimation guarantees from theory

One piece of theory that guarantees we can learn about expectations is a Strong Law of Large Numbers (LLN). Specifically given a probability space $(\mathsf{X}, \mathcal{X}, \pi)$, a collection of independent, identically distributed random variables $\{X_t\}_{t=1}^n \sim \pi^{\otimes n}$ indexed by $n \in \mathbb{N} \backslash \{0\}$, and a function $f \in L^1(\pi)$, we have that

$$\hat{f}_n := \frac{1}{n}\sum_{t=1}^{n} f(X_t) \to \pi(f) \tag{1.1}$$

almost surely, as $n \to \infty$ [Billingsley 1995, Theorem 6.1]. It is readily checked that $\hat{f}_n$ is an unbiased estimate of $\pi(f)$ for all $n \in \mathbb{N} \backslash \{0\}$. Therefore the ability to generate independent samples from π allows one to estimate expectations with respect to it and the errors in our estimates are guaranteed to diminish. A natural question to ask is then 'how many samples should we generate, to achieve a given error in our estimates?'. When $f \in L^2(\pi)$ we can use the Central Limit Theorem (CLT) to help us answer. The CLT states that

$$\sqrt{n}\left(\hat{f}_n - \pi(f)\right) \to \mathcal{N}(0, \sigma^2)$$

[1] In the sense that the answer is $\arg\min_\ell \mathbb{E}_{\theta,\mathsf{T}}\left[(\mathcal{L}(\theta) - \ell)^2\,|X\right]$

in distribution, as $n \to \infty$, where $\sigma^2 := \text{Var}_\pi(f)$ [Billingsley 1995, Theorem 27.1], and so the deviation in $\hat{f}_n$ about $\pi(f)$ decays asymptotically at a rate of $1/\sqrt{n}$. In fact, routine calculations give a variance of $n^{-1}\sigma^2$ for estimates $\hat{f}_n$ for all $n \in \mathbb{N} \setminus \{0\}$. What is crucial in the satisfaction of the conditions on which these results hold is our ability to take independent samples from π.

The variance of $\hat{f}_n$ decomposes into two terms: one depending solely on the number of samples, and the other depending on f and π. Therefore we have two lines of attack to attempt to reduce the variance: we can take more samples from π, or we can adjust f and π (taking into account any biasedness this might cause). Therefore in Section 1.2.2.2 we will describe a method that will allow us to take samples from π, and in Section 1.2.2.3 we describe a method to decrease σ^2 by changing f and π that also preserves unbiasedness. We do not pretend to be exhaustive in our presentation of these methods. What we offer we merely do so to give the reader a notion of the considerations one must take when constructing unbiased estimators using independent samples, and to define techniques that we use in the later chapters.

1.2.2 Monte Carlo methods: sampling and variance reduction

1.2.2.1 Simple Monte Carlo

The act of generating independent samples from π and forming the estimator $\hat{f}_n$ 1.1 is called *simple Monte Carlo*. Simple Monte Carlo is a popular method given the facility to produce independent samples from π. For one thing, small perturbations in f can produce large changes in the analytic form of $\pi(f)$, whereas the simple Monte Carlo method and output remain roughly the same. For another, $\hat{f}_n$ can be calculated in an online fashion from $\hat{f}_{n-1}$ and X_n, hence the method is memory efficient and its estimates can be easily transferred from one source of randomness to another.

A property of simple Monte Carlo which is highly desirable is the independence of the rate at which $\text{Var}\left(\hat{f}_n\right)$ decreases with n on the particular details of X or π. In particular it is independent of the dimension of X. For sufficiently smooth functions f and in low enough dimensions, certain quadrature rules achieve much better error rates, see [Art B. Owen 2023]. However these gains deteriorate rapidly with dimension, and cannot hold for non-smooth functions.

Given these desirable properties, a method to generate independent samples from an arbitrary measure

Algorithm 1.1 The rejection method. **Inputs**: sampleable proposal distribution ν, constant C. **Outputs**: $X \sim \pi$

```
while true
    Sample $X \sim \nu$ and $U \,|X \sim$ Uniform $[0, C\nu\,(X)]$.
    if $U \leq \pi\,(X)$
        return $X$
```

π can be seen to be very valuable. We describe such a method below.

1.2.2.2 The rejection method

The rejection method works by converting one source of randomness into another. It monitors the stream of random variables from the first source of randomness until a particular condition is satisfied. This condition guarantees a sample from π. The first source of randomness is characterised by an easily sampleable probability measure ν such that $\pi \ll \nu$. We call ν the *proposal distribution*. Then, given access to a constant $0 < C < \infty$ such that $\|d\pi/d\nu\|_\infty \leq C$ where $d\pi/d\nu$ is the Radon-Nikodym derivative between π and ν, the method works as in algorithm 1.1. Often the algorithm is stated with $U \sim$ Uniform $[0, 1]$ and the success condition being $CU \leq d\pi/d\nu$, which can be seen to be equivalent to 1.1. We state it as above for the purposes of exposition. If the densities of π and ν are only known in their unnormalised forms $\tilde{\pi}$ and $\tilde{\nu}$, then the densities in algorithm 1.1 can be replaced by their unnormalised counterparts. That $\pi \ll \nu$ is necessary for the proper working of the method is clear from the algorithm since, otherwise, there may be some regions of supp (π) that cannot be accessed by the proposal distribution ν.

That the algorithm produces a sample from π relies on two crucial facts: for the first we must define the restriction of a random variable to a measurable subset:

Definition 1. Say μ is a probability measure on the space (X, $\mathcal{X}$), and that $X \sim \mu$. For all $A \in \mathcal{X}$ such that $\mu\,(A) > 0$, we define the restriction of X to A as the random variable $X\,|_A$ defined by the probability measure $\mu\,|_A$ where $\mu\,|_A\,(B) := \mu\,(A \cap B)\,/\mu\,(A)$.

The first fact is then:

Lemma 2. *Let $A, B \in \mathcal{X}$ with $B \subseteq A$. If $U \sim$* Uniform $\{A\}$ *then $U\,|_B \sim$* Uniform $\{B\}$.

which follows from the definition of the restriction. The second fact is as follows:

Lemma 3. *Let π be a probability measure on* X *and $0 < C < \infty$. If (X_1, U_1) is sampled using $X_1 \sim \pi$ with $U_1 \, |X_1 \sim$* Uniform $[0, C\pi\,(X_1)]$ *and (X_2, U_2) is sampled from* Uniform $\{(x, u) : 0 \leq u \leq C\pi\,(x)\}$ *then $(X_1, U_1) \overset{d}{=} (X_2, U_2)$.*

Lemma 2 is relatively easy to grasp and is the basis for many elementary demonstrations of the Monte Carlo method for estimating, say, the ratio of two volumes. Lemma 3 is more involved. One of the essential pieces of information it conveys is that if (X, U) is sampled uniformly from the area underneath a (scaled) density then X will be distributed according to that density. For a proof of lemma 3 see section 7.2.1.1. The lemmas combine to guarantee the exactness of algorithm 1.1.

Proposition 4. *Algorithm 1.1 outputs a sample from π.*

Lemma 3 has that (X, U) is sampled from Uniform $\{(x, u) : 0 \leq u \leq C\nu\,(x)\}$. The 'If' statement restricts this uniform sample to the area under the density of π. Then Lemma 2 and 3 combine to yield the fact that $X \sim \pi$. Clearly whether the densities of ν or π in algorithm 1.1 are normalised is immaterial to the exactness of the algorithm by the arbitrariness of the constant C in Lemma 3.

What the normalisation of the measures *does* affect however is the expected time to a sample from π. The termination of algorithm 1.1 is conditional on a random event, which is independent between iterations. Therefore the time to termination is geometrically distributed.

Proposition 5. *The expected time to termination of the rejection method with constant $C \geq \|d\tilde{\pi}/d\tilde{\nu}\|_\infty$ is C, where $\tilde{\pi}$ and $\tilde{\nu}$ are the unnormalised forms of the measures π and ν.*

Therefore the user should attempt to find the closest upper bound to $\|d\tilde{\pi}/d\tilde{\nu}\|_\infty$ to use in the algorithm, to minimise the expected time-complexity. Finding a proposal distribution ν such that $\pi \ll \nu$ and for whom $\|d\tilde{\pi}/d\tilde{\nu}\|_\infty$ is finite and bounded away from 0 is not so easy if, say, the tail information of π is not known. Even if we had such a ν that we could sample from, the constant C may be difficult to estimate, or it may be so large as to render the probability of termination by some fixed time negligible. In general the rejection method is 'all-or-nothing' in the sense that its success in a given iteration does not depend on any information from the prior iterations.

Extensions Adaptive rejection methods [Gilks and Wild 1992] have been conceived to iteratively improve ν and thus C. In another case the squeeze method [Marsaglia 1977] reduces the per iteration time-complexity by reducing the probability of an evaluation of the density of π. In any case, we now have a method to generate exact samples from π, and thus we can decrease the variance of $\hat{f}_n$ by adjusting the n-dependent part.

1.2.2.3 Stratified sampling

We now describe a method that decreases the variance of $\hat{f}_n$ by adjusting π and f in such a way that does not introduce bias. Stratified sampling [Cochran 1977, Chapters 5 - 5A] is a technique from classical statistics whereby we 'stratify' the state space into a partition $\{\mathsf{X}_i : i \in [R]\}$. It assumes we can sample from $\pi_i := \pi|_{\mathsf{X}_i}$ for all $i \in [R]$, and that we know the mass of each part under π. We choose the number of samples n_i to sample from π_i for each part. Then, given $\left\{\{Y_{ij}\}_{j=1}^{n_i}\right\}_{i=1}^{R}$ with $Y_{ij} \sim \pi_i$ for all $j \in [n_i]$ and $i \in [R]$, we form the estimator

$$\hat{f}_{\text{strat}} := \sum_{i=1}^{R} \frac{\pi(\mathsf{X}_i)}{n_i} \sum_{j=1}^{n_i} f(Y_{ij})$$

It can be readily checked that $\hat{f}_{\text{strat}}$ is unbiased. It has variance

$$\text{Var}\left(\hat{f}_{\text{strat}}\right) = \sum_{i=1}^{R} \frac{\pi(\mathsf{X}_i)^2}{n_i} \text{Var}_{\pi_i}(f) \tag{1.2}$$

The strategy from here on in is to choose the n_i's in order to achieve a reduced variance from that of the simple Monte Carlo estimator 1.1. Before we describe the first strategy we introduce a mathematical object that naturally occurs in the presence of a partition of the state space.

The resolution The resolution is an adjustment to f that makes it piecewise constant on the parts, but retains its expectation under π. Specifically the resolution is defined as

$$\overrightarrow{\pi} f(x) := \sum_{i=1}^{R} \mathbb{1}\{x \in \mathsf{X}_i\} \pi_i(f)$$

Its variance under π is

$$\mathsf{Var}_{\pi}\left(\overrightarrow{\pi} f\right)=\sum_{i=1}^{R}\pi\left(\mathsf{X}_i\right)\left(\pi_i\left(f\right)-\pi\left(f\right)\right)^2$$

which can be seen as a measure of the variation of f between the parts in the partition. We also define the orthogonal counterpart to the resolution: $\overleftarrow{\pi} f := f-\overrightarrow{\pi} f$. It has variance

$$\mathsf{Var}_{\pi}\left(\overleftarrow{\pi} f\right)=\sum_{i=1}^{R}\pi\left(\mathsf{X}_i\right)\mathsf{Var}_{\pi_i}\left(f\right)$$

which can be seen as a measure of the variation of f within the parts in the partition. It has expectation 0 under π. We call it the orthogonal counterpart because $\mathsf{Cov}_{\pi}\left(\overrightarrow{\pi} f,\overleftarrow{\pi} f\right)=0$. These functions are useful because they appear naturally in the presence of a partition of the state space. For our present purposes, they are useful because they define a decomposition of the variance of $\hat{f}_n$:

$$\mathsf{Var}_{\pi}\left(\hat{f}_n\right)=\frac{1}{n}\left(\mathsf{Var}_{\pi}\left(\overrightarrow{\pi} f\right)+\mathsf{Var}_{\pi}\left(\overleftarrow{\pi} f\right)\right) \tag{1.3}$$

due to the fact that $\mathsf{Var}_{\pi}\left(f\right)=\mathsf{Var}_{\pi}\left(\overrightarrow{\pi} f\right)+\mathsf{Var}_{\pi}\left(\overleftarrow{\pi} f\right)$. With this mathematical machinery to hand we can describe and examine the first allocation strategy.

Proportional allocation Proportional allocation takes the number of samples from π_i in proportion to the π-mass of the associated part. Specifically, fixing $n_{\mathsf{strat}}\in\mathbb{N}\backslash\left\{0\right\}$ it takes $n_i:=\pi\left(\mathsf{X}_i\right)n_{\mathsf{strat}}$ where we ignore effects due to rounding. The variance of the resulting estimator is $n_{\mathsf{strat}}^{-1}\mathsf{Var}_{\pi}\left(\overrightarrow{\pi} f\right)$. From 1.3 we see that $\mathsf{Var}\left(\hat{f}_{\mathsf{strat}}\right)\leq\mathsf{Var}\left(\hat{f}_n\right)$ if $n_{\mathsf{strat}}\geq n$. In fact we are able to state a quantitive relationship between the two variances: $\mathsf{Var}\left(\hat{f}_{\mathsf{strat}}\right)=\left(n/n_{\mathsf{strat}}\right)\left(1-\mathsf{Corr}_{\pi}\left(f,\overrightarrow{\pi} f\right)^2\right)\mathsf{Var}\left(\hat{f}_n\right)$, for proof see section 7.2.1.2. Proportional allocation is not always the best strategy since if f were constant on one of the parts, we would only need to sample once from that part to achieve unbiasedness. Any more samples would not reduce the variance of $\hat{f}_{\mathsf{strat}}$. This fact was recognised in [Neyman 1992] leading to the proposal of an optimal allocation strategy.

Optimal allocation In proportional allocation, the sample numbers only incorporated information about the target weights $\pi\left(\mathsf{X}_i\right)$. However, there exist allocation strategies that take into account information from

f that achieve a reduced variance as a result.

Proposition 6. *The allocation strategy*

$$n_i = \frac{n\pi\left(\mathsf{X}_i\right)\sqrt{\mathsf{Var}_{\pi_i}\left(f\right)}}{\sum_{i=1}^{R}\pi\left(\mathsf{X}_i\right)\sqrt{\mathsf{Var}_{\pi_i}\left(f\right)}}$$

minimises Var $\left(\hat{f}_{\text{strat}}\right)$ *subject to* $n_1+\cdots+n_R = n$ *and* $n_i \geq 0$ *for all* $i \in [R]$ *(ignoring effects due to rounding).*

For a proof see section 7.2.1.3. Since the strategy in Proposition 6 minimises the variance, it must produce an estimator with lower variance compared with proportional allocation. For a treatment that incorporates the cost of sampling per part see [Owen 2013, Section 8.4].

In general we will not know $\mathsf{Var}_{\pi_i}\left(f\right)$. There is also the problem of how to choose the strata: clearly from 1.2 if f is constant across each part, we achieve an estimator with zero variance. Stratified sampling assumes the ability to sample from π_i for all $i \in [R]$. This is a difficult task even if one knew how to sample from π. As a result of this we develop a method in chapter 4 that extends stratified sampling to the case that one cannot sample from π or π_i for all $i \in [R]$. Nor does it rely on the knowledge of the part weights $\pi\left(\mathsf{X}_i\right)$.

1.3 Iterative sampling methods

In many cases simple Monte Carlo is impossible to carry out, whether by rejection sampling or some other technique. This is due to a lack of knowledge about π leading to an inability to sample directly. Distributions over high dimensional state spaces often have highly non-trivial geometry, making it difficult to specify them with a low-dimensional parametrisation. Therefore these distributions are difficult to acquire holistic knowledge about in a way that makes them sampleable. When encountering such challenges, it is common to employ methods that do not directly and independently sample from π. Instead, these approaches construct a sequence of approximating measures that converge to π and can be implemented algorithmically to generate samples $\{X_t\}_{t=1}^{n}$ from these measures.

Using information from previous samples to inform new approximating measures allows us to iteratively improve these approximations. However we incur dependencies between the samples so that $\{X_t\}_{t=1}^{n}$ is no longer sampled from $\pi^{\otimes n}$. A common form of dependence is Markoviannity. An idea to motivate this structure is that using such a loose dependence will cause samples to look effectively independent so long as they are sufficiently far away from each other on the Markov chain. However the number of Markovian samples we will need to collect will then be at most a multiple of the number of independent samples we would have collected in the simple Monte Carlo case. This type of dependency also has the benefit of having a large body of theory to analyse stochastic processes that satisfy it. Markovian processes that are constructed to sample approximately from measures π with the objective of estimating expectations are called Markov chain Monte Carlo (MCMC) methods.

1.3.1 Estimation guarantees from theory

Let $\{X_t\}_{t=1}^{\infty}$ be a Markov chain on X with initial state $X_0 \sim \pi_0$ and kernel K : X $\times \mathcal{X} \to [0, 1]$ such that the nth state in the chain has distribution $\pi_0 K^n$ for all $n \in \mathbb{N}$. We would like to ensure that averages over the chain converge to expectations under π. In the same way as if the samples were independent and all from π we achieve this via Laws of Large Numbers and Central Limit Theorems analogous to those introduced in section 1.2.1 except here the averages are over Markov chains. However we must take the intermediary step to guarantee that $\pi_0 K^n$ tends to π in some formal sense. The way this convergence is commonly defined is in total variation, for which we must define the total variation distance between two probability measures:

Definition 7. For two probability measures π_1 and π_2 on $\mathcal{X}$ their total variation distance is defined as

$$\|\pi_1 - \pi_2\|_{TV} := \sup_{A \in \mathcal{X}} |\pi_1(A) - \pi_2(A)|$$

As suggested by its notational form, the total variation distance is really a norm on the signed measure $\pi_1 - \pi_2$, which we can think of as the pointwise discrepancy between π_1 and π_2. Maximising over this discrepancy gives us the worst possible discrepancy, which is what the total variation distance is.

1.3.1.1 Ergodicity

We can now define the type of convergence of a Markov chain that is necessary for our aims:

Definition 8. Let π be a probability measure on $\mathcal{X}$ and let K : X $\times \mathcal{X} \to [0, 1]$ be the Markov kernel which generates the chain $\{X_t\}_{t=1}^{\infty}$ that has $X_0 \sim \pi_0$ as its initial state. The chain is π-ergodic when

$$\|\pi_0 K^n - \pi\|_{TV} \to 0$$

as $n \to \infty$ for all initial distributions π_0.

We often simply say that the chain is ergodic when it is obvious what the measure π is.

1.3.1.2 Convergence guarantees

We now define conditions under which the chain is π-ergodic and explain their salience one-by-one.

Theorem 9. *[Meyn and R. Tweedie 1993, Theorem 13.3.3] Let $\{X_t\}_{t=1}^{\infty}$ be the Markov chain initialised at $X_0 \sim \pi_0$ generated by the kernel K. Assume that the chain*

1. *is φ-irreducible, for some σ-finite measure φ,*
2. *has an invariant probability measure,*
3. *is Harris recurrent,*
4. *and is aperiodic.*

Then there exists a unique invariant probability measure π such that the chain is π-ergodic.

The conditions can be altered and replaced in various ways, some of which cause the theorem statement to change. We present those in the theorem here because, having explained them, the reader should be left with a sense as to why they cause ergodicity. Furthermore, the chains introduced in the following chapters are assumed to be ergodic unless otherwise stated.

φ-irredubility

Definition. Let φ be a σ-finite measure on $\mathcal{X}$. The Markov chain is φ-irreducible if for all $A \in \mathcal{X}$ such that $\varphi(A) > 0$ and for all $x \in$ X there exists an $n \in \mathbb{N}\backslash \{0\}$ possibly depending on x and A such that

$$\delta_x K^n (A) > 0$$

This property is informative because it determines where the chain can go: namely the support of φ. Clearly the restriction of φ to any non-null subset of its support also defines an irreducibility measure. What is important, however, is an extension of φ which is maximal in some sense, since such a measure will have the full descriptive power as to where the Markov chain can go:

Proposition 10. *[Meyn and R. Tweedie 1993, Proposition 4.2.2] If the Markov chain is φ-irreducible then there exists a probability measure ψ on $\mathcal{X}$ such that*

1. *the Markov chain is ψ-irreducible and*
2. *for any other measure φ' the chain is φ'-irreducible if and only if $\psi \gg \varphi'$.*

In particular 2. implies that $\psi \gg \varphi$. Therefore ψ is somehow maximally descriptive as to where the Markov chain can go. Another crucial guarantee that φ-irreducibility ensures is the existence of so-called ‘small sets’ contained within every set A such that $\psi(A) > 0$ where ψ is the maximal irreducibility measure [Meyn and R. Tweedie 1993, Theorem 5.2.2]. We won’t go into the technical details here, but essentially one can construct a Markov chain that is identical to the original chain in some distributional sense that has the chance to produce an independent sample whenever it enters one of these ‘small’ sets. See [Athreya and Ney 1978; E. Nummelin 1978] for the original expositions of this technique. Therefore if we ensure that these sets are visited infinitely often, the estimator produced by averaging over the Markov chain will share some distributional properties (namely independence) with estimators produced by averaging over independent samples, for which there exist asymptotic results such as LLNs and CLTs.

Invariant probability measure

Definition 11. Let π be a probability measure on $\mathcal{X}$. It is an invariant measure of the Markov chain if $\pi K = \pi$.

It can be checked that if π is an invariant measure of the Markov chain then $\pi K^n = \pi$ for all $n \in \mathbb{N}$. Therefore if we track the distribution of the state of a Markov chain, if it becomes π it will remain π and thus the chain will be stationary from that point onwards. This alone does not ensure the π-ergodicity of the chain. Where the existence of an invariant measure becomes salient is if it is unique in the following sense: say for each initial distribution π_0 we have that the Markov chain converges setwise such that $\pi_0 K^n (A) \to \nu_{\pi_0} (A)$ for all $A \in \mathcal{X}$. Then

$$\begin{aligned}
\nu_{\pi_0} (A) &= \lim_{n \to \infty} \int \pi_0 (dx) \, K^n (x \to A) \\
&= \lim_{n \to \infty} \int \int \pi_0 (dx) \, K^{n-1} (x \to dw) \, K (w \to A) \\
&= \int \nu_{\pi_0} (dw) \, K (w \to A)
\end{aligned}$$

where the third line comes from the fact that the setwise convergence of $\pi_0 K^n$ implies convergence of integrals of bounded measurable functions. Since K is a Markov kernel it is measurable in its first argument and bounded. Therefore if ν_{π_0} is converged to setwise, it is an invariant distribution, and if there is only one invariant distribution it is independent of π_0, which is what we want since we want the Markov chain to be π-ergodic for an arbitrary π.

One property that ensures the existence of an invariant measure is reversibility.

Definition 12. A Markov chain generated by kernel K is π-reversible when for all $A, B \in \mathcal{X}$ we have $\int_A \int_B \pi (dx) \, K (x \to dy) = \int_B \int_A \pi (dx) \, K (x \to dy)$.

Proposition 13. *A Markov chain which is π-reversible has π as an invariant measure.*

Proof. Let A be an arbitrary set in $\mathcal{X}$. Then

$$\begin{aligned}\pi K(A) &= \int_{\mathsf{X}}\int_{A}\pi(dx)\,K(x\to dy)\\ &= \int_{A}\int_{\mathsf{X}}\pi(dx)\,K(x\to dy)\\ &= \int_{A}\pi(dx)=\pi(A)\end{aligned}$$

where the second line comes from π-reversibility. □

If π and $K(x\to .)$ have densities with respect to the Lebesgue measure one can verify π-reversibility by checking that $\pi(x)\,K(x\to y)=\pi(y)\,K(y\to x)$ for all $x,y\in\mathsf{X}$ where we abuse notation to let π and $K(x\to .)$ be densities. Reversibility can therefore be checked locally. This may make it easier to establish than π-invariance since for the latter we need to verify the equation $\pi=\pi K$ which involves an integral even when π and K have densities with respect to the Lebesgue measure.

Harris recurrence First we define recurrence:

Definition 14. A φ-irreducible chain is recurrent if for all $x\in\mathsf{X}$

$$\mathbb{E}\left[\text{Time spent in } A \text{ when the chain starts from } x\right]=\infty$$

for all sets A such that $\psi(A)>0$ where ψ is the maximal measure introduced in Proposition 10.

and then the stronger notion of Harris recurrence:

Definition 15. A φ-irreducible chain is Harris recurrent if for all $x\in\mathsf{X}$

$$\mathbb{P}\left(\text{The Markov chain enters } A \text{ infinitely many times after starting from } x\right)=1$$

for all sets A such that $\psi(A)>0$ where ψ is the maximal measure introduced in Proposition 10.

where we note that Harris recurrence implies recurrence. We introduce these notions simultaneously because morally what the chain needs to be ergodic is recurrence, but Harris recurrence helps prevent edge cases such as non-zero probabilities of escaping to infinity being included. As [Chan and Geyer 1994] note, [Esa Nummelin 1984] uses Harris recurrence to prove that the time for the Markov chain to enter one of the 'small' sets we mention in 1.3.1.2 is almost surely finite for any choice of initial distribution.

Recurrence is essential because it guarantees that the Markov chain will visit the 'small' sets described in the above section on φ-irreducibility infinitely often. It also guarantees the existence of a unique invariant measure:

Theorem 16. *[Meyn and R. Tweedie 1993, Theorem 10.0.1] If a φ-irreducible chain is recurrent then it admits a unique (up to multiplication by a constant) invariant measure.*

Aperiodicity First we must define what it means for the chain to be periodic:

Definition 17. A chain is periodic with period $k \in \mathbb{N}\backslash\{0,1\}$ if there exist disjoint subsets $D_1, \ldots, D_k \in \mathcal{X}$ such that $\delta_x K\left(D_{(i+1) \mod k}\right) = 1$ where $x \in D_i$ for all $i \in [n]$.

A chain is aperiodic if it is not periodic. One simple reason for the chain to be aperiodic is because if it were periodic one can easily construct examples in which it never converges. For instance let the chain generated by K be periodic with period $k \in \mathbb{N}\backslash\{0,1\}$ and let π be any probability measure, let $x \in D_i$ for some $i \in \{0, \ldots, k-1\}$ and $B \in \mathcal{X}$ be such that $B \cap D_i = \varnothing$ and $\pi(B) > 0$. Then whenever $n \mod k = i$

$$\begin{aligned} \|\delta_x K^n - \pi\|_{TV} &= \sup_{A \in \mathcal{X}} |\delta_x K^n(A) - \pi(A)| \\ &\geq |\delta_x K^n(B) - \pi(B)| = |\pi(B)| \end{aligned}$$

since the state distributed according to $\delta_x K^n$ must be in D_i. Therefore the chain cannot be ergodic. Aperiodicity can also ensure the existence of 'small' sets, see [Meyn and R. Tweedie 1993, Proposition 5.4.5].

1.3.1.3 Implications of ergodicity

Ergodicity assures us that the samples from the chain will eventually look like samples from π. We would like to convert this property to one that will allow us to guarantee good estimation properties of the estimators that we form using the chain. Whilst ergodicity doesn't explicitly give a rate of convergence, we can use the following useful property:

Proposition 18. *[Meyn and R. Tweedie 1993, Proposition 13.3.2] If π is an invariant measure for K then*

$$\|\pi_0 K^n - \pi\|_{TV}$$

is non-increasing in n for all initial measures π_0.

So if we want better quality samples from π, all we need to do is wait. Our ultimate goal is to estimate expectations of functions f with respect to π. Ergodicity implies that to do this we can simply initialise many Markov chains, run them for long enough and average f over their final states. However, suppose we only want to use a single Markov chain and average our function over its output. A Markov chain (LLN) allows us to do this. The conditions for such a result are often similar to those necessary for ergodicity:

Proposition 19. *[Meyn and R. Tweedie 1993, Theorem 17.0.1 i)] Let $\{X_t\}_{t=1}^{\infty}$ be the Markov chain initialised at $X_0 \sim \pi_0$ generated by the kernel K. If $\{X_t\}_{t=1}^{\infty}$ is φ-irreducible, π-invariant, and Harris recurrent and if $f \in L^1(\pi)$ then*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} f(X_t) = \pi(f)$$

almost surely.

As we saw in section 1.2.1 a CLT result is more informative than an LLN since it allows us to quantify the error in our estimators. The usual route to establishing a CLT result for Markov chains is to prove that the chain is geometrically ergodic, which is a stronger property than ergodicity:

Definition 20. The Markov chain generated by K is said to be geometrically ergodic when there exists a $\lambda > 0$ such that for all initial states $x \in \mathsf{X}$ there exists a constant $c(x) > 0$ where

$$\|\delta_x K^n - \pi\|_{TV} \leq c(x) \exp(-\lambda n)$$

One may deduce numerous results such as LLNs and concentration inequalities from geometric ergodicity, but here we are primarily interested in establishing a CLT. When we are concerned with averaging a function $f : \mathsf{X} \to \mathbb{R}$ over a geometrically ergodic Markov chain the only fact that we need to verify to establish a CLT is that $f \in L^{2+\varepsilon}(\pi)$ for some $\varepsilon > 0$ [Chan and Geyer 1994, Theorem 2]. If the chain is reversible, we only need to verify that $f \in L^2(\pi)$ [G. Roberts and J. Rosenthal 1997, Corollary 2.1]. Where the error of the estimator is concerned, the latter result has that

$$\sqrt{n}\left(\frac{1}{n}\sum_{t=1}^{n} f(X_t) - \pi(f)\right) \to \mathcal{N}(0, \mathsf{Var}(K, f))$$

in distribution where

$$\mathsf{Var}(K, f) := \mathsf{Var}_\pi(f) + 2\sum_{t=1}^{\infty} \mathsf{Cov}_\pi\left(f(X), K^t f(X)\right) \tag{1.4}$$

Thus we can see the inefficiencies due to using Markovian samples rather than independent ones: if the infinite sum in the definition of $\mathsf{Var}(K, f)$ is positive, the Markov chain estimators will have a higher variance than the Monte Carlo estimators (holding the amount of samples fixed across the two cases). We will call $\mathsf{Cov}_\pi(f(X), K^t f(X))$ and $\mathsf{Corr}_\pi(f(X), K^t f(X))$ the lag t autocovariance and autocorrelation respectively for all $t \in \mathbb{N}\backslash\{0\}$. The form of $\mathsf{Var}(K, f)$ also tells us which chains will produce estimators of a high variance, namely ones whose autocorrelations remain high at all lags. In fact we use $\mathsf{Var}(K, f)$ to define the practical metrics by which we gauge the efficiency of the Markov chain.

1.3.1.4 Markov chain efficiency metrics

Let's assume that $f \in L^2(\pi)$ and that the chain is reversible. This then gives us a CLT result for both the estimator produced by the Markov chain and the Monte Carlo estimator $\hat{f}_n$. We can then compare the asymptotic variances defined in either CLT to give us a measure of efficiency of the Markov chain. This is because we view the Monte Carlo estimator as the 'gold standard' in terms of its asymptotic variance, which itself is due to the fact that $\mathsf{Var}(K, f) \geq \mathsf{Var}_\pi(f)$ under conditions which are mild and commonplace[2]. Under these conditions we can define the effective sample size of the Markov chain:

[2] such as the positivity of the Markov kernel K.

Definition 21. The effective sample size (ESS) of a Markov chain of length $n \in \mathbb{N} \backslash \{0\}$ is defined as

$$n \frac{\mathrm{Var}_{\pi}(f)}{\mathrm{Var}(K, f)}$$

This is so called because it is roughly the amount of samples from the Monte Carlo estimator we would need to take to get an equivalent variance as the Markov chain estimator. One can interpret it as the amount of 'gold standard' samples the Markov chain yields. One should keep in mind that the estimator is dependent on a particular function f, and so is not strictly a fully general measure of health of the Markov chain. It should also be noted that the ESS depends on estimating expectations with respect to π which we can only do well if the Markov chain is efficient. Therefore the ESS should always be taken with a pinch of salt whenever we suspect that the Markov chain is not healthy, by whatever other means we have.

Another estimate of the efficiency of the Markov chain is the lag 1 autocorrelation. A large lag 1 autocorrelation is taken to indicate an inefficient Markov chain. This is because we can make the approximation $\mathrm{Var}(K, f) \approx \mathrm{Var}_{\pi}(f) + 2\mathrm{Cov}_{\pi}(f(X), Kf(X))$. One reason we can make this approximation is that in the case that $\mathrm{Cov}_{\pi}(f(X), Kf(X)) \geq 0$ for all $f \in L^2(\pi)$ we have that $\mathrm{Cov}_{\pi}(f(X), Kf(X)) \geq \mathrm{Cov}_{\pi}\left(f(X), K^2 f(X)\right)$ for all $f \in L^2(\pi)$ (for a proof of this fact see section 7.2.1.4). Using [Haggstrom and J. Rosenthal 2007, Lemma 16] then gives us that $\mathrm{Cov}_{\pi}(f(X), Kf(X)) \geq \mathrm{Cov}_{\pi}(f(X), K^t f(X))$ for all $f \in L^2(\pi)$ and $t \geq 2$. It seems logical that we can use the lag 1 autocorrelation as an indicator of chain efficiency since if the chain doesn't move much in a single step, then it's probably not going to move much over many steps, across many lengths of the chain. Therefore a large lag 1 autocorrelation can be taken as a global measure of the ill-health of the chain (relative to a given estimand f).

1.3.2 Markov chain Monte Carlo

Now that we've examined the conditions under which Markov chains achieve different forms of ergodicity, and the quantities by which we can measure their efficiency, all that remains to do is to construct them. Algorithms that produce samples using Markov chains so as to estimate expectations are called Markov chain Monte Carlo (MCMC) algorithms. In this section we will focus on reversible chains since, as outlined in section 1.3.1.2, their π-invariance is easier to verify.

Algorithm 1.2 Framework for an algorithm in the MH family. **Inputs**: Easily sampleable proposal kernel Q with oracle access to its density q, Initial state $X_0 \sim \pi_0$, Oracle access to the unnormalised density $\tilde{\pi}$ of π **Outputs**: A π-reversible Markov chain $\{X_t\}_{t=1}^{n}$.

```
For $t \in [n]$
    Sample a proposal $Y_t \sim Q\left(X_{t-1} \to .\right)$.
    With probability
        $\alpha\left(X_{t-1} \to Y_t\right) := \min\left\{1, \frac{\tilde{\pi}\left(Y_t\right) q\left(Y_t \to X_{t-1}\right)}{\tilde{\pi}\left(X_{t-1}\right) q\left(X_{t-1} \to Y_t\right)}\right\}$
    set $X_t = Y_t$.
    Otherwise set $X_t = X_{t-1}$.
```

1.3.2.1 The Metropolis-Hastings family

As with many other iterative methods the history of MCMC methods develops in tandem with the history of the modern computer. The first instance of MCMC is [Metropolis et al. 1953] wherein the authors propose an algorithm to be run on the MANIAC computer at Los Alamos National Laboratory in order to simulate the behaviour of hard disks on a two dimensional torus. [Hastings 1970] unified the methods of [Metropolis et al. 1953] and [Barker 1965] and extended their application to uncountable state spaces. It is from these two papers ([Metropolis et al. 1953] and [Hastings 1970]) that the family of Metropolis-Hastings (MH) algorithms gets its name. Algorithms in this family are constructed as follows: given a Markov kernel $Q : \mathsf{X} \times \mathcal{X} \to [0, 1]$ which we call the proposal kernel and oracle access to its density $q : \mathsf{X} \times \mathsf{X} \to \mathbb{R}^{+} \cup \{0\}$ iterate over the steps in algorithm 1.2. Note that the unnormalised density $\tilde{\pi}$ is accessed via its ratio, and therefore the algorithm is equivalent if we replace it with its normalised form. This is one of the aspects of the MH family that make it so popular: that it can work with unnormalised target distributions. We will refer to the unnormalised density $\tilde{\pi}$ as the normalised density π where the two are interchangeable, so as to simplify the notation. The algorithm in 1.2 dictates that the corresponding Markov kernel K of $\{X_t\}_{t=1}^{n}$ has the following form:

$$K\left(x \to A\right) := \int_A q\left(x \to dy\right) \alpha\left(x \to y\right) + \left(1 - \int_{\mathsf{X}} q\left(x \to dy\right) \alpha\left(x \to y\right)\right) \delta_x\left(A\right)$$

It is readily checked that K is π-reversible due to the particular form of α. In fact α can take other forms, see [Barker 1965] for example, although [Peskun 1973] argues that using α as in algorithm 1.2 minimises

Var (K, f) (for a fixed proposal distribution) where it exists. The MH family is then indexed by those proposal distributions that preserve ergodicity. The choice of proposal is crucial to the success of the algorithm. Optimising over the space of Markov kernels $q\left(x \rightarrow .\right)$ is infeasible. To approximate this optimisation task there exist a few canonical instances of proposal distributions $q\left(x \rightarrow .\right)$ whose tuning parameters are then optimised over. We will go over three instances here: random walk Metropolis, the Metropolis adjusted Langevin Algorithm, and Hamiltonian Monte Carlo. We will define these in the case that $\mathsf{X} = \mathbb{R}^d$.

1.3.2.2 Random walk Metropolis

When $q\left(x \rightarrow y\right) = \mathcal{N}\left(y; x, \sigma^2 \mathbf{I}_d\right)$ algorithm 1.2 defines the random walk Metropolis (RWM) algorithm. Note that $q\left(x \rightarrow y\right) = q\left(y \rightarrow x\right)$ and hence α does not require the evaluation of the proposal density. In particular α dictates that proposals that have a higher π density will always be accepted.

The parameter $\sigma^2 \in \mathbb{R}^+$ is called the step size. Current practice (see, for instance, [Michalis Titsias 2023, Algorithm 1]) is to tune the step size based on the results of what is called the 'optimal scaling' literature, see e.g. [Beskos et al. 2013; G. O. Roberts, A. Gelman, and Gilks 1997; Jeffrey S Rosenthal et al. 2011]. These papers look at the behaviour of the RWM Markov chain in the so-called 'diffusion limit' which is where, roughly speaking, the speed of the Markov chain is scaled proportional to d and the step size is scaled proportional to d^{-1}. The authors then observe that the first component of the resulting process as $d \rightarrow \infty$ behaves like a diffusion process. They find that the quantity

$$\bar{\alpha} := \int_{\mathsf{X}} \int_{\mathsf{X}} \pi\left(dx\right) q\left(x \rightarrow dy\right) \alpha\left(x \rightarrow y\right)$$

can be optimised over to maximise the speed of this diffusion process, and that $\bar{\alpha} = 0.234$ achieves approximate optimality. It must be noted that these results are achieved under the assumption that the dimensions of π are independent, or can be made independent under an affine tranformation. In recent works, setting $\sigma^2 \propto d^{-1}$ has been motivated in the non-asymptotic setting [Andrieu, A. Lee, et al. 2024]. In general a step size which is too large will produce many rejection events and a step size which is too small will produce many acceptance events. In both of these cases the autocorrelation in the chain will be large, causing an inflated variance in 1.4.

RWM is a zeroth order algorithm, which means that it does not use the gradient information of π. This effectively puts a cap on the speed with which the algorithm moves which can be a problem for heavy tailed targets, where the Markov chain must make long excursions into the tails so as to properly represent the target mass. The Markov chain can also display diffusive-like behaviour when in the tails of a heavy tailed distribution:

Example 22. Say the target measure π has density $\pi(x) \propto \left(1 + a\|x\|_2^2\right)^{-b}$ for $a, b \in \mathbb{R}^+$. Then the ratio term inside $\alpha(x \to y)$ in algorithm 1.2 is

$$\left(\frac{1 + a\|x\|_2^2}{1 + a\|x + \sigma\xi\|_2^2}\right)^b$$

where $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$. Let $\{x_n\}_{n\geq 1}$ be any sequence such that $\lim_{n\to\infty}\|x_n\|_2 = \infty$ and $\{\xi_n\}_{n\geq 1}$ be a random sequence with $\xi_n \sim \mathcal{N}(0, \mathbf{I}_d)$ independently, for all $n \geq 1$. Then the ratio term tends to one almost surely with respect to the randomness of the sequence of ξ's.

1.3.2.3 The Metropolis adjusted Langevin Algorithm

The Metropolis adjusted Langevin Algorithm (MALA) [Gareth O. Roberts and Richard L. Tweedie 1996] is based on the overdamped Langevin diffusion which is a stochastic differential equation (SDE) conceived in statistical physics to describe the kinematics of a physical system under the influence of a deterministic force and stochastic fluctuations.

$$dX_t = -\nabla U(X_t)\,dt + \sqrt{2}dB_t \tag{1.5}$$

where $B_t \in \mathbb{R}^d$ is the d-dimensional Brownian motion and $U : \mathsf{X} \to \mathbb{R}$ is a differentiable potential function defining the force ∇U on the state of the system X_t. Theorem 1 of [Y.-A. Ma, T. Chen, and Fox 2015] has that $\pi(x) \propto \exp(-U(x))$ is the density of the unique stationary distribution of 1.5 and so setting $U = -\log\pi$ and simulating from it will eventually give us samples from π. If we ignore the Brownian motion equation 1.5 describes the deterministic continuous time dynamics of a gradient descent process on U. Therefore we can interpret the process as a noisy form of gradient descent, where we reach areas of high π density with the drift term and we explore those areas with the Brownian motion.

Exact simulation of a continuous time SDE on a digital computer is usually infeasible. Therefore we need to discretise the Langevin diffusion in order to simulate it. The Euler-Maruyama discretisation gives us a way to do this. It is a generalisation of Euler's method to the stochastic setting. Given times $t_n \leq t_{n+1}$, if we initialise the Langevin diffusion at X_{t_n} and solve for time $X_{t_{n+1}}$ we get

$$X_{t_{n+1}} = X_{t_n} - \int_{t_n}^{t_{n+1}} \nabla U\left(s\right) ds + \sqrt{2} \int_{t_n}^{t_{n+1}} dB_s$$

by definition. The Euler-Maruyama discretisation therefore makes the following approximation to the entire solution of 1.5:

$$X_{t_{n+1}} \approx \tilde{X}_{n+1} \text{ where } \tilde{X}_{n+1} = \tilde{X}_n - \left(t_{n+1} - t_n\right) \nabla U\left(\tilde{X}_n\right) + \sqrt{2}\sqrt{t_{n+1} - t_n}\xi, \text{ with } \tilde{X}_0 = X_{t_0}$$

for all $n \in \{0, \ldots, N-1\}$ where $\xi \sim \mathcal{N}\left(0, \mathbf{I}_d\right)$, X_{t_0} is the initial state of the Langevin diffusion, and $t_0 < t_1 < \cdots < t_N$ for $N \in \mathbb{N}\backslash\{0\}$. Making the approximation over subsequent time intervals of equal length means that we can replace $t_{n+1} - t_n$ with a generic step size $\sigma^2 \in \mathbb{R}^+$. The Euler-Maruyama converges strongly to the solutions of 1.5 in the sense that $\lim_{\sigma^2 \searrow 0} \mathbb{E}\left[\left\|X_{t_N} - \tilde{X}_N\right\|\right] = 0$. In fact, under conditions on the smoothness and growth rate of ∇U, one can establish upper bounds on $\mathbb{E}\left[\left\|X_{t_N} - \tilde{X}_N\right\|\right]$ which are polynomial in σ^2, see [Kloeden and Platen 1992, Theorem 10.2.2]. What is important here is that $\mathbb{E}\left[\left\|X_{t_N} - \tilde{X}_N\right\|\right]$ is non-zero for non-zero step size, which makes the approximation inexact. Therefore we cannot guarantee that π is the distribution of $\tilde{X}_N$ as $N \to \infty$. To rectify this the MALA wraps single steps of the Euler-Maruyama discretisation in accept-reject steps which ensure π-reversibility. It assumes the form of algorithm 1.2 with

$$Y_t = X_{t-1} - \sigma^2 \nabla U\left(X_{t-1}\right) + \sqrt{2\sigma^2}\xi \tag{1.6}$$

where $\xi \sim \mathcal{N}\left(0, \mathbf{I}_d\right)$ and $U \propto -\log \pi$ giving $q\left(x \to y\right) = \mathcal{N}\left(y; x - \sigma^2 \nabla U\left(x\right), 2\sigma^2 \mathbf{I}_d\right)$. Note here that the target density appears in the term $\nabla U \equiv \nabla\left(-\log \pi\right)$ which is unchanged whether π is normalised or not. Note also that $q\left(x \to y\right) \neq q\left(y \to x\right)$ and so $\alpha\left(x \to y\right)$ depends on the density of the proposal. Just like the Euler-Maruyama discretisation, MALA relies on the smoothness of ∇U for its proper functioning. The inclusion of the ∇U term can cause the acceptance probability to be unstable.

Example 23. Say $d = 1$ and $\nabla U(X) = X^3$ with $X_0 = O\left(\sigma^{-2}\right)$ and $\sigma < 1$. This dictates that $U(X) = 4^{-1}X^4$ such that $\pi(X_0) \propto \exp\left(-O\left(\sigma^{-8}\right)\right)$. Then $Y_1 = O\left(\sigma^{-4}\right)$, $\nabla U(Y_1) = O\left(\sigma^{-12}\right)$, and $\pi(Y_0) = \exp\left(-O\left(\sigma^{16}\right)\right)$. The form of the MALA proposal 1.6 dictates that

$$\begin{aligned}\frac{q(Y_1 \to X_0)}{q(X_0 \to Y_1)} &= \exp\left(-\frac{1}{4\sigma^2}\left(\left\|x - y - \sigma^2\nabla U(y)\right\|^2 - \left\|y - x - \sigma^2\nabla U(x)\right\|^2\right)\right) \\ &= \exp\left(-\frac{1}{4\sigma^2}\left(\left\|O\left(\sigma^{-2}\right) - O\left(\sigma^{-4}\right) - O\left(\sigma^{-12}\right)\right\|^2 - \left\|O\left(\sigma^{-4}\right) - O\left(\sigma^{-2}\right) - O\left(\sigma^{-4}\right)\right\|^2\right)\right) \\ &= \exp\left(-\frac{1}{4\sigma^2}\left(O\left(\sigma^{-24}\right) - O\left(\sigma^{-8}\right)\right)\right) \\ &= \exp\left(-O\left(\sigma^{-22}\right)\right)\end{aligned}$$

and we have that $\pi(Y_1)/\pi(X_0) = \exp\left(-O\left(\sigma^{16}\right)\right)$. This means that $\alpha(X_0 \to Y_1)$ will be close to zero for the vast majority of draws of Y_1 and that's if we ignore any numerical instability that might occur in the calculations.

There is also evidence that MALA is less robust to poor tuning of the step size than RWM when the scales of π are heterogeneous across the directions of the state space, see [Livingstone and Zanella 2022] for details. When the acceptance probability in 1.2 is fixed at 1, MALA becomes the Langevin Monte Carlo (LMC) algorithm.

1.3.2.4 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) [Duane et al. 1987; Neal 2011] necessitates the extension of the state space to include a momentum variable $p \in \mathbb{R}^d$. Like MALA, each proposal is the forward simulation of a dynamical process that models the state of a physical system. Taking $x \in \mathbb{R}^d$ to be the position of a particle with mass $m \in \mathbb{R}^+$, Newton's second law of motion dictates that the position of the particle is associated to the potential U in which it moves in the following way: $-\nabla U(x) = m\ddot{x}$. Defining the momentum $p := m\dot{x}$ gives the Hamiltonian dynamics

$$\frac{dp}{dt} = -\nabla U(x), \frac{dx}{dt} = m^{-1}p \tag{1.7}$$

under the Hamiltonian $H\left(x,p\right) := U\left(x\right) + \left(2m\right)^{-1}\left\|p\right\|_2^2$. The Boltzmann-Gibbs distribution defined by this Hamiltonian is $\pi\left(x,p\right) \propto \exp\left(-U\left(x\right) - \left(2m\right)^{-1}\left\|p\right\|_2^2\right)$ which, when marginalised over p, yields $\pi\left(x\right) \propto \exp\left(-U\left(x\right)\right)$. Theorem 1 of [Y.-A. Ma, T. Chen, and Fox 2015] also applies to the dynamics described by equation 1.7 although in this case there is no stochastic component to the motion described, and hence $\pi\left(x,p\right)$ is merely an invariant distribution and not unique. As we know from section 1.3.1.2 this is not enough to guarantee ergodicity. In any case, as with MALA, integrating 1.7 exactly is infeasible, and so we must again consider a discretisation scheme.

The discretisation scheme used in HMC is the leapfrog scheme [Leimkuhler and Reich 2005]. This is defined using the following recursion: given an initial point $\left(X_0, P_0\right)$, a number of iterations N, a step size σ

$$\begin{aligned} P_{n+\frac{1}{2}} &= P_n - \frac{\sigma}{2}\nabla U\left(X_n\right) \\ X_{n+1} &= X_n + \frac{\sigma}{m}P_{n+\frac{1}{2}} \\ P_{n+1} &= P_{n+\frac{1}{2}} - \frac{\sigma}{2}\nabla U\left(X_{n+1}\right) \end{aligned} \tag{1.8}$$

for $n \in \{0, \ldots, N-1\}$. This iterator has a number of desirable properties. The local truncation error, defined as the leading term in the local error $\left(X\left(t_1\right) - X_1, P\left(t_1\right) - P_1\right)^T$ where $\left(X\left(t_1\right), P\left(t_1\right)\right)^T$ is the true solution of the dynamics in 1.7, is order σ^3 whereas for a straightforward Euler discretisation of 1.7 the local truncation error is order σ^2. If we regard 1.8 as three separate transformations $\left(X_n, P_n\right) \mapsto \left(X_n, P_{n+\frac{1}{2}}\right), \left(X_n, P_{n+\frac{1}{2}}\right) \mapsto \left(X_{n+1}, P_{n+\frac{1}{2}}\right), \left(X_{n+1}, P_{n+1}\right) \mapsto \left(X_{n+1}, P_{n+\frac{1}{2}}\right)$ each transformation involves an affine, volume preserving (i.e. determinant $= \pm 1$) transformation of a subset of the full state. Hence the full transition described in 1.8 is volume preserving, and so the transformation $\mathcal{T}\left(X_0, P_0\right) := \left(X_N, P_N\right)$ described by the full trajectory of the integrator is volume preserving. Thus if $\left(X_0, P_0\right) \sim \nu$ then $\left(X_N, P_N\right) \sim \nu\left(\mathcal{T}^{-1}\left(X_N, P_N\right)\right)$ with no volume correction. Note also that if $N = 1$ and $P_0 = \sqrt{m}\xi$ where $\xi \sim \mathcal{N}\left(0, \mathbf{I}_d\right)$, rephrasing 1.8 solely in terms of the X coordinates gives

$$X_1 = X_0 - \frac{1}{2}\frac{\sigma^2}{m}\nabla U\left(X_0\right) + \sqrt{\frac{\sigma^2}{m}}\xi$$

which is just the MALA proposal with step size $m^{-1}\sigma^2$ (up to a rescaling by $\sqrt{2}$).

Since the momentum marginal of the Boltzmann-Gibbs distribution $\pi(x,p) \propto \exp\left(-U(x) - (2m)^{-1}\|p\|_2^2\right)$ is just a Gaussian, each step of HMC starts with a momentum resampling step $P_0 = \sqrt{m}\xi$ where $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$. We then achieve a new point according to the leapfrog integrator, using N steps and a step size of σ^2, giving (X_N, P_N). Now, crucially, we negate the momentum component to ensure reversibility with respect to the Lebesgue measure, yielding $(X_N, -P_N)$. That the leapfrog integrator is reversible and volume-preserving gives us that $q((x,p) \to (x',p')) = q((x',p') \to (x,p))$ and hence we have

$$\alpha((x,p) \to (x',p')) = \min\left\{1, \exp\left(-U(x') - (2m)^{-1}\|p'\|_2^2 + U(x) + (2m)^{-1}\|p\|_2^2\right)\right\}$$

Now all we need is to ensure φ-irreducibility. Having done this, the fact that the method fits into the MH framework in 1.2 ensures π-reversibility which gives ergodicity. However, ensuring φ-irreducibility is non-trivial. Nominally, what we need to ensure is that there exists a momentum draw for each point in the position space that, under the leapfrog dynamics, will transport the position component to an arbitrary set $B \in \mathcal{X}$ which is non-null under φ. One phenomenon which can break irreducibility is when the step size, trajectory length, and target are such that the leapfrog integrator always returns to the same position and momentum, for a given initial position and momentum, see [Livingstone, Betancourt, et al. 2019, Supplementary Material] for an example of this. One way to ensure this never happens is to randomise the trajectory length, as proposed in [Bou-Rabee and Sanz-Serna 2017], such that having $N = 1$ always has a positive probability. We can then note that the $N = 1$ case is equivalent to the MALA proposal 1.6, and the full MALA algorithm is irreducible with respect to the Lebesgue measure under mild and checkable conditions [G. O. Roberts and R. L. Tweedie 1996]. See also [Durmus, E. Moulines, and Saksman 2019] for a proof of irreducibility with a fixed $N \geq 1$ trajectory length. When the acceptance probability in 1.2 is fixed at 1, HMC becomes the unadjusted HMC algorithm.

Chapter 2

Developments to canonical MCMC: preconditioning, variational approximation, and adaptivity

Over the course of their use in the past century, it has become increasingly clear that the canonical Markov kernels described in the previous chapter will fail to work properly on certain classes of target probability distributions. These classes include distributions in high dimension and distributions which are ill-conditioned, whether this is due to anisotropy, multimodality, or some other exotic geometry.

In this chapter we introduce three techniques which are used to enhance the canonical kernels of chapter 1. In section 2.1 we introduce preconditioning, which can be thought of as making an invertible transformation to the target distribution in order to make it easier to sample from. In section 2.2 we introduce variational approximations, which are distributional approximations to the target. When we have a variational approximation to the target distribution, we can use it to enhance the performance of a canonical MCMC sampler. Finally in section 2.3 we introduce adaptive MCMC. Adaptive MCMC is a method which runs alongside and influences an MCMC sampler. It uses the previous states in the sampler to approximate a good preconditioning transformation, which it uses in the subsequent Markov chain step.

For the notation included in this chapter, see section 7.1.2.

2.1 Preconditioning

2.1.1 Introduction to preconditioning

The creation of electronic computers in the first half of the twentieth century gave us two abilities. The first ability was to mathematically encode the process of solving problems. We gained this ability because we were no longer purely solving problems with equipment we did not understand i.e. the human brain, but with computers that we had built. These computers were sufficiently primitive that we could then understand and formalise their operation. The second ability was to rapidly solve problems at multiple different scales. This gave us reliable data as to how the computing machinery, whose influence on the solution was now known precisely, would perform on a variety of problems and how it would scale with the problem size.

Given the first ability we could then theorise about the capability of particular computational resources at solving problems, and given the second ability we could verify the theory we had produced as a result of the first. One important quantity that is often naturally conceived in this process of theorising is a *condition number*. It aims to encode exactly the capability of our computational resources at solving a problem. Another way of viewing it is as encoding the difficulty of a problem given computational resources. It does this by assigning harder problems higher condition numbers. The term 'condition number' was coined in [TURING 1948] although similar quantities are defined in [Von Neumann and Goldstine 1947; Wittmeyer 1936]. If it is sufficiently descriptive we can use it to evaluate new computational methods that have been devised to make the problem easier to solve. These are often called 'preconditioning' methods because their success or failure can be explained by their effects on the condition number.

The problems considered in this thesis are sampling problems. We will see that a condition number is defined for sampling problems, and so is preconditioning. Our primary aim is to evaluate preconditioning methods that can be described using linear transformations on sampling problems whose condition number is finite. We restrict ourselves to the $\mathsf{X} = \mathbb{R}^d$ setting. Linear transformations are the most common form of preconditioning in sampling. That the condition number is finite allows us to describe precisely the effect of these transformations.

2.1.2 The condition number and preconditioning for the problem of sampling

2.1.2.1 Assumptions on the target distribution

In this section and in chapter 3 we assume that the potential of every target distribution we consider will satisfy the following properties:

Assumption 24. *Let $U \in C^2$ be the potential of the target distribution π.*

1. *We assume that U is m-strongly convex i.e. there exists a constant $m > 0$ such that $m\mathbf{I}_d \preceq \nabla^2 U(x)$ for all $x \in \mathbb{R}^d$.*
2. *We assume that U is M-smooth i.e. there exists a constant $M > 0$ such that $\nabla^2 U(x) \preceq M\mathbf{I}_d$ for all $x \in \mathbb{R}^d$.*

Clearly if U is both m-strongly convex and M-smooth then $M \geq m$. These properties are a very common setting in the field of optimisation where U serves as the objective function instead of the potential of the target distribution [Boyd and Vandenberghe 2004; Bubeck 2015]. These properties necessarily hold if we wish to define the condition number for sampling problems, as we shall see. In the foregoing we will assume that, when invoked, the constants m and M are the tightest values they can be i.e. that m is the largest m satisfying assumption 24 1. and that M is the smallest M satisfying assumption 24 2.

2.1.2.2 Features of distributions with m-strongly convex and M-smooth potentials and the algorithms that sample from them

m-strong convexity This property is shared by the potentials of numerous common distributions such as the Gaussian, and the univariate Weibull with parameter $\beta > 2$[1]. More generally, Bayesian posteriors with Gaussian priors whose log-likelihoods are concave in their parameter have strongly convex potentials, where

[1] which is defined to have density $h_\beta(x) := \beta x^{\beta-1}\exp\left(-x^\beta\right)\mathbb{1}\left\{x \in (0,\infty)\right\}$ with respect to the Lebesgue measure on $\mathbb{R}$, see [Saumard and Wellner 2014, Example 2.18].

the constant m is just the inverse of the maximum eigenvalue of the prior covariance. Distributions with m-strongly convex potentials have properties that can be beneficial in statistical contexts, such as unimodality, and the fact that strong convexity is conserved under the marginalisation [Saumard and Wellner 2014]. A potential U which is m-strongly convex can be seen to satisfy the following relation:

$$U\left(x\right)-U\left(y\right)\geq\nabla U\left(y\right)^{T}\left(x-y\right)+\frac{m}{2}\left\Vert x-y\right\Vert _{2}^{2} \tag{2.1}$$

for all $x,y\in\mathbb{R}^{d}$ (in fact this is equivalent to m-strong convexity). Choosing y to be the mode $x^{*}\in\mathbb{R}^{d}$ we can see that $U\left(x\right)-U\left(x^{*}\right)\geq\left(m/2\right)\left\Vert x-x^{*}\right\Vert _{2}^{2}$ and so distributions with m-strongly convex potentials have subgaussian tails

Another equivalent definition of the m-strong convexity of U is the following ‘monotonicity’ property: that $\left(\nabla U\left(x\right)-\nabla U\left(y\right)\right)^{T}\left(x-y\right)\geq m\left\Vert x-y\right\Vert _{2}^{2}$. Setting y to be the mode $x^{*}\in\mathbb{R}^{d}$ and $x\neq x^{*}$ we see that

$$\left(-\nabla U\left(x\right)\right)^{T}\frac{\left(x^{*}-x\right)}{\left\Vert x^{*}-x\right\Vert _{2}}\geq m\left\Vert x^{*}-x\right\Vert _{2} \tag{2.2}$$

Therefore the size of $-\nabla U\left(x\right)$ in the direction from x to the mode is guaranteed to increase when x moves further away from the mode, at a rate dictated by the strong convexity constant m. This means that algorithms such as MALA 1.3.2.3 and HMC 1.3.2.4 that use ∇U move quicker in the tails, when they move at all. This allows them to move from regions of low probability to high probability quickly, which helps them mix. An instance of this behaviour can seen in the fact that the m-strong convexity of U implies a lower bound of m on the spectral gap of the overdamped Langevin diffusion 1.5 [Bakry, Gentil, Ledoux, et al. 2014].

M-smoothness The Gaussian and hyperbolic distributions both have M-smooth potentials. Bayesian posteriors with Gaussian priors whose likelihoods are M-smooth in their potentials have smooth potentials whose constants are $M+\lambda_{\text{min}}^{-1}$-smooth where λ_{min} is the minimum eigenvalue of the prior covariance. Distributions with M-smooth potentials have beneficial statistical properties, although they are less simple than those possessed by distributions with m-smooth potentials. For instance, if $Y\sim\pi$ where π has an M-smooth potential U then $\nabla U\left(Y\right)$ is subgaussian with constant L [Negrea 2022]. An M-smooth potential satisfies a kind of reverse inequality to 2.1:

$$U\left(x\right)-U\left(y\right)\leq\nabla U\left(y\right)^{T}\left(x-y\right)+\frac{M}{2}\left\Vert x-y\right\Vert _{2}^{2} \tag{2.3}$$

for all $x,y\in\mathbb{R}^{d}$. Setting $y=x^{*}$ again and we see that the density whose potential is U can be minorised by scaled Gaussian density. We can use 2.3 to lower bound the average acceptance rate of RWM due to the fact that $\pi\left(y\right)/\pi\left(x\right)=\exp\left(U\left(x\right)-U\left(y\right)\right)$ can be controlled, see [Andrieu, A. Lee, et al. 2024, Corollary 40] for details. We also have that M-smooth potentials satisfy a 'monotonicity' property, which is like the reverse of 2.2:

$$\left(-\nabla U\left(x\right)\right)^{T}\frac{\left(x^{*}-x\right)}{\left\Vert x^{*}-x\right\Vert _{2}}\leq M\left\Vert x^{*}-x\right\Vert _{2}$$

and so algorithms which use $-\nabla U\left(x\right)$ to guide their proposals, like MALA 1.3.2.3 and HMC 1.3.2.4, make small steps toward the mode when close to it. This is desirable since we want to spend large amounts of time in regions of high probability under the target. In general the discretisations of dynamics whose continuous time processes are defined using a smooth ∇U, such as the overdamped Langevin 1.5 and the Hamiltonian dynamics 1.7, will have controllable errors, see [Kloeden and Platen 1992, Theorem 10.2.2] for instance.

2.1.2.3 The condition number

When the potential U of the target distribution satisfies 24 we define the condition number of the target distribution as

$$\kappa:=\frac{M}{m} \tag{2.4}$$

which we can define equivalently as $\kappa=\sup_{x\in\mathbb{R}^{d}}\left\Vert \nabla^{2}U\left(x\right)\right\Vert _{2}\sup_{x\in\mathbb{R}^{d}}\left\Vert \nabla^{2}U\left(x\right)^{-1}\right\Vert _{2}$. The condition number is bounded below by 1. As discussed in section 2.1.1, we would like for it to describe how difficult it is for our sampling algorithms to sample from the target by assigning harder sampling problems higher condition numbers.

The significance of κ in the context of sampling problems is demonstrated by its ubiquity in bounds on quantities which govern the performance of MCMC algorithms, such as the relaxation time 7.1 and the ε-

	Relaxation time	ε-mixing time
Upper bound	$O\left(\kappa d\right)^{\diamondsuit}$[Andrieu, A. Lee, et al. 2024]	$\tilde{O}\left(\kappa d^{\frac{1}{2}}\right)^{\heartsuit}$[Wu, Schmidler, and Y. Chen 2022]
		$\tilde{O}\left(\kappa d \log\frac{1}{\varepsilon}\right)^{\diamondsuit}$[Andrieu, A. Lee, et al. 2024]
		$\tilde{O}\left(\kappa d^{\frac{2}{3}} \log\frac{1}{\varepsilon}\right)^{\spadesuit}$[Y. Chen, Dwivedi, et al. 2020]
Lower bound	$\Omega\left(\frac{\kappa d}{\log d}\right)^{\heartsuit}$ [Y. T. Lee, Shen, and Tian 2021]	$\Omega\left(\frac{\kappa d}{\log^2 d}\right)^{\heartsuit}$ [Y. T. Lee, Shen, and Tian 2021]

Table 2.1: Recently published bounds on the relaxation time and ε-mixing time of various MCMC samplers. The $\tilde{O}$ denotes that the bound excludes poly-logarithmic terms. Superscript $\diamondsuit$ denotes a bound for RWM, superscript $\heartsuit$ for MALA, and superscript $\spadesuit$ for HMC. The bound in [Y. Chen, Dwivedi, et al. 2020] holds under the addition assumptions that $\nabla^2 U$ is L_H-Lipschitz with $L^{2/3} = O\left(M\right)$, that $\kappa = O\left(d^{2/3}\right)$, and that the chain is started from a $\beta = O\left(\exp\left(d^{2/3}\right)\right)$ warm start. See 7.1 for a definition of the relaxation time and 7.2 for a definition of the mixing time.

mixing time 7.2. These two quantities provide some measure of the number of steps to run the MCMC algorithm, since the ε-mixing time tells us how long we should wait for the chain to equilibrate from its initial distribution, and the relaxation time tells us the average time-scale in equilibrium we would need to wait to form good quality estimators. The two quantities are related since, for instance, a small relaxation time will imply a small ε-mixing time if the chain is positive (i.e. the spectrum of the Markov operator is contained in $[0, 1]$) and is well initialised, using e.g. a β-warm start 7.3. See table 2.1 for a selection of bounds on these quantities. Each is polynomial in the dimension and the condition number. The selection is not exhaustive, we merely present it to emphasise the ubiquity of the condition number.

2.1.2.4 Preconditioning

Let's say that the condition number we've defined describes sufficiently well the difficulty of the problems we're trying to solve given the algorithms we're using to solve them. Then we can evaluate adjustments to the problems or algorithms by looking at their effect on the condition number. Adjustments which cause the condition number to go down should then make the problem easier, which is often felt practically by a decrease in time complexity of the algorithms. We call these adjustments 'preconditioning'. Here we focus on adjustments that can be fully described by linear transformations. The matrices that encode these transformations are called 'preconditioners' or 'preconditioning matrices' and the practice of using these preconditioners is called 'linear preconditioning'.

Formally, linear preconditioning is the use of an invertible matrix $L \in \mathbb{R}^{d\times d}$ to take a target distribution π whose potential is m-strongly convex and M-smooth and transform it into $\tilde{\pi}(\tilde{x}) = L\#\pi(\tilde{x}) = \pi\left(L^{-1}\tilde{x}\right)|\det L|^{-1}$ such that the condition number of $\tilde{\pi}$ will be lower than that of π. The corresponding potential of $\tilde{\pi}$ will be $\tilde{U}(\tilde{x}) = U\left(L^{-1}\tilde{x}\right)$. Given the monotonicity of the bounds in table 2.1 on κ, the aim is that the reduction in the condition number will come with a reduction in the time complexity of our sampling algorithm. Since the potential of π is m-strongly convex and M-smooth, the condition number of $\tilde{\pi}$ will exist, be finite, and be defined by

$$\tilde{\kappa} = \sup_{x\in\mathbb{R}^d}\left\|L^{-T}\nabla^2 U(x)L^{-1}\right\|_2 \sup_{x\in\mathbb{R}^d}\left\|L\nabla^2 U(x)^{-1}L^T\right\|_2 \tag{2.5}$$

by a linear change of variables procedure. In our examination of how the condition number changes under a given preconditioner assuming that L is symmetric will aid in notational simplicity, which we can do without a loss of generality:

Proposition 25. *Fix a preconditioner $L \in \mathbb{R}^{d\times d}$ and a target distribution π. Construct a new preconditioner $L^{\dagger} = VDV^T$ where $V \in O(d)$ consists of the right singular vectors of L and $D \in \mathbb{R}^{d\times d}$ is a diagonal matrix containing the singular values of L. Then the condition number $\tilde{\kappa}$ after preconditioning with L and $L^{\dagger}$ is the same.*

For a proof see section 7.2.2.1.

2.1.2.5 Traditional formulation of preconditioning

Often linear preconditioning is encountered as a modification to the proposal distribution of a canonical MCMC algorithm and not as a transformation to the target distribution. For instance, as it is usually encountered, preconditioned MALA 1.6 has proposal

$$Y_t = X_{t-1} - \sigma^2 A\nabla_x U(X_{t-1}) + \sqrt{2\sigma^2}A^{\frac{1}{2}}\xi \tag{2.6}$$

where $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$ and $A \in \mathbb{R}^{d\times d}$ is a positive definite matrix which we call the traditional preconditioner. This is equivalent to making a proposal

$$Y'_t = X'_{t-1} - \sigma^2 \nabla_{x'} U'\left(X'_{t-1}\right) + \sqrt{2\sigma^2}\xi \tag{2.7}$$

where $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$, $U'(x') := U\left(A^{\frac{1}{2}}x'\right)$, and we make the linear transformation $x' = A^{-\frac{1}{2}}x$ to go from the unprimed to the primed variables. We can use this fact to show that the Markov chains generated using proposal 2.6 with target $\pi \propto \exp(-U)$ are isomorphic in the sense of [L. T. Johnson and Geyer 2012, Appendix A] to the Markov chains generated using proposal 2.7 with target $\pi' \propto \exp(-U')$. In fact, the isomorphism holds for most canonical MCMC samplers.

Proposition 26. *Let $\{X_t\}_{t=1}^{\infty}$ be the Markov chain generated by a canonical MCMC sampler with traditional preconditioner $A \in \mathbb{R}^{d\times d}$ targeting $\pi \propto \exp(-U)$. Let $\{X'_t\}_{t=1}^{\infty}$ be the Markov chain generated by a canonical MCMC sampler with traditional preconditioner $\mathbf{I}_d$ targeting $\pi' \propto \exp(-U')$ where $U'(x') := U\left(A^{\frac{1}{2}}x'\right)$. Then the two Markov chains are isomorphic in the sense of [L. T. Johnson and Geyer 2012, Appendix A]. By 'canonical MCMC sampler' we mean any of the following: RWM, MALA, HMC, LMC, unadjusted HMC.*

For a proof see section 7.2.2.2. Isomorphic Markov chains share convergence and stability properties modulo an appropriate transformation, and thus they are basically the same in terms of performance. For instance if a Markov chain is geometrically ergodic with constant λ, all isomorphic chains are also geometrically ergodic with the same constant. This justifies our formulation of preconditioning as a transformation to the target distribution, since it is basically equivalent to the traditional notion of preconditioning.

2.2 Variational approximation

2.2.1 Introduction to variational approximations

MCMC algorithms are tools that practitioners are forced to consider using when the target distribution is sufficiently complex. In these circumstances, variational approximations offer an alternative to MCMC. Variational approximations are distributional surrogates for the target distribution. They are constructed to have properties that allow us to easily estimate expectations that are formed with respect to them. This restriction means that the space of possible approximations will not be the entire space of probability distributions. In fact, the space of possible approximations will be defined by a space of possible parametrisations which is searched over to find the best possible approximation. Concretely this necessitates the introduction of a parameter θ which lives in a parameter space Θ that defines the space of possible distributions over which we search for an approximation. This search is usually conducted by attempting to find $\theta^{*} := \text{argmin}_{\theta\in\Theta} d\left(\nu_{\theta},\pi\right)$ where $d : \mathcal{P}\left(\mathcal{X}\right)\times\mathcal{P}\left(\mathcal{X}\right)\rightarrow[0,\infty)$ is a discrepancy on the space of probability measures on $\mathcal{X}$. This discrepancy is usually the reverse Kullback-Leibler (KL) divergence. The whole process of positing Θ and searching for θ^{*} is called *variational inference* (VI), see [Blei, Kucukelbir, and McAuliffe 2017] for an introductory text and [Ganguly and Earp 2021] for an introduction with applications to machine learning. In the search for a variational approximation there is usually a trade-off between the expressivity of $\left\{\nu_{\theta}\right\}_{\theta\in\Theta}$ and the ease with which we can search through Θ to find θ^{*}.

Often, in the MCMC setting, the target we are trying to approximate is a Bayesian posterior whose density can be written $\pi\left(x\left|y\right.\right)$ where $y \in$ Y is the data. We abuse notation in denoting $\pi\left(.\left|y\right.\right) : \mathcal{X}\rightarrow[0,1]$ as its associated probability measure. In this case, elementary calculations reveal the following relation:

$$KL\left(\nu_{\theta}\left\|\pi\left(.\left|y\right.\right)\right.\right)+\mathbb{E}_{\nu_{\theta}}\left[\log\frac{\pi\left(X,y\right)}{\nu_{\theta}\left(X\right)}\right]=\log\pi\left(y\right) \tag{2.8}$$

where $\nu_{\theta}\left(x\right)$ is the density of the variational approximation at $x \in$ X, $\pi\left(x,y\right)$ is the joint density of the parameters and the data, and $\pi\left(y\right)$ is the marginal density of the data. The second quantity in the sum on the left hand side has acquired a name: the Evidence Lower BOund (ELBO):

$$\text{ELBO}\left(\theta\right) := \mathbb{E}_{\nu_{\theta}}\left[\log\frac{\pi\left(X,y\right)}{\nu_{\theta}\left(X\right)}\right]$$

and that it lower bounds the logarithm of the evidence (i.e. the marginal data density) is evident from 2.8 and the positivity of the KL. The relation 2.8 also reveals that $\text{argmin}_{\theta\in\Theta}KL\left(\nu_{\theta}\left\|\pi\left(.\left|y\right.\right)\right.\right)=\text{argmax}_{\theta\in\Theta}\text{ELBO}\left(\theta\right)$ and for this reason, many VI methods use maximising the ELBO as a surrogate for learning the optimal variational parameter. Decomposing the ELBO into $\mathbb{E}_{\nu_{\theta}}\left[\log\pi\left(X,y\right)\right]+\mathbb{E}_{\nu_{\theta}}\left[-\log\nu_{\theta}\left(X\right)\right]$ reveals that it serves as a regularised objective: in maximising it we want to simultaneously maximise the joint log-density and the entropy $\mathcal{H}(\nu_\theta) := \mathbb{E}_{\nu_\theta}[-\log \nu_\theta(X)]$ of the variational distribution. Maximising the joint log-density is natural to the objectives of VI, whilst maximising the entropy helps maintain the spread of the variational approximation, and ensures it does not collapse to a Dirac mass. A final fact to note is that the ELBO is not convex in θ.

Many common families of distributions have a natural parametrisation. Therefore these families provide for us ready-built parameter spaces over which we can conduct our search. We can also use the product operation to combine distributions so as to form new multivariate distributions to match the dimension of the state space of the target. An example of this approach is the Coordinate Ascent Variational Inference (CAVI) [Bishop 2006]. It imposes a basic structure on the variational approximation: that it can be decomposed along its dimensions. Say $\mathsf{X} = \mathbb{R}^d$ and assume that the parameter space decomposes as follows: $\Theta = \bigotimes_{i=1}^d \Theta_i$. Then CAVI begins by imposing that the density of the approximation is written as

$$\nu_\theta(x) = \prod_{i=1}^{d} \nu_{\theta_i}^{(i)}(x_i) \tag{2.9}$$

with $\nu_\theta^{(i)} : \mathbb{R} \to [0, \infty)$ being the density of a univariate random variable and $\theta_i \in \Theta_i$ for all $i \in [d]$. Given this form, each step of CAVI consists of maximising the ELBO over Θ_i for a given $i \in [d]$. After updating the variable at index i, the ith variational marginal density will have the following form:

$$\nu_{\theta_i}^{(i)}(x_i) \propto \exp\left(\mathbb{E}_{\nu_{\theta_{-i}}^{-(i)}}\left[\log \pi(X_i, X_{-i}, y) \,|X_i = x_i\right]\right) \tag{2.10}$$

where $\nu_{\theta_{-i}}^{-(i)}$ is the measure ν_θ after having marginalised out the variable at the ith index. For a proof see section 7.2.2.3. Note that this update will not necessarily preserve the parametric family of $\nu_{\theta_i}^{(i)}$: see [Blei, Kucukelbir, and McAuliffe 2017, Section 3 Appendix C] for examples that do. The complete CAVI procedure then consists in choosing the parametric families of the $\nu_{\theta_i}^{(i)}$'s and updating using 2.10 for all $i \in [d]$ repeatedly until the ELBO converges. If the updates dictated by 2.10 can be carried out exactly, then the ELBO is guaranteed to converge, since it is upper bounded and maximised coordinate-wise by 2.10. However, under the decomposition assumption in 2.9 on the variational distribution, the ELBO is still non-convex, and so convergence may be to a local maximum. [Bhattacharya, Pati, and Y. Yang 2023] provide exponential convergence under the assumption that Θ is decomposed into the product of 2 spaces, and under control on a specifically derived formulation of correlation of the target measure. [Arnese and Lacker 2024] provide conditions under which CAVI converges to the global maximum of the ELBO, under the assumption of the log-concavity and smoothness of the target measure. [Lavenant and Zanella 2024] provide tight rates of convergence in the case that the index i is chosen at random each step, also under log-concavity and smoothness assumptions.

The structure imposed by 2.9 makes the CAVI updates mathematically nice, and has the possibility of informing the dimension and structure of the parameter space Θ. However, it also precludes the possibility of dependence between any two dimensions of the variational approximation. We'll see in section 3.5 that this independence assumption can lead to pathological examples, even for the simplest target distributions. We note that, as we've formulated it, CAVI needn't necessarily assume that $v_{\theta_i}^{(i)}$ is univariate for all $i \in [d]$, see e.g. the case analysed in [Bhattacharya, Pati, and Y. Yang 2023].

2.2.1.1 Transport based approaches

Often a member of a parametric family of distributions can be viewed as a transformation from a 'source' random variable. Take, for instance, the distribution $\mathcal{N}\left(\mu, \Sigma\right)$ on $\mathbb{R}^d$ which we will attempt to 'target'. Defining our source distribution to be a standard normal, then if ξ is distributed according to the source, $\mu + \Sigma^{1/2}\xi$ will be distributed according to the target. Therefore, if we wanted to sample from the target, we would simply sample from the source, and perform an appropriate transformation, which in this case is affine. Computing expectations with respect to the target can then be done using Monte Carlo.

Transport based approaches attempt to use this framework to approximate a target distribution π: given a source distribution ν on a state space Z that is sufficiently easy to sample from, we attempt to find a diffeomorphic transformation $T : \mathsf{Z} \to \mathsf{X}$ such that $T\#\nu \approx \pi$. Then if we want to evaluate $\mathbb{E}_\pi\left[f\left(X\right)\right]$ we can use $\mathbb{E}_{T\#\nu}\left[f\left(X\right)\right] = \mathbb{E}_\nu\left[f\left(T\left(Z\right)\right)\right]$ as an approximation. Instead of parametrising a variational distribution, we parametrise the transformation using T_θ for $\theta \in \Theta$ and leave the source measure fixed. The construction and parametrisation of the transformation is made so as to render $\left\{T_\theta\#\nu\right\}_{\theta\in\Theta}$ as expressive as possible, whilst ensuring the tractability of the learning procedure. As we stated before, the reverse KL is typically used to define this procedure, which now consists of finding argmin$_{\theta\in\Theta} KL\left(T_\theta\#\nu\,\|\pi\right)$. Elementary calculations on this objective give

$$KL\left(T_\theta\#\nu\,\|\pi\right) = \text{const.} - \mathbb{E}_\nu\left[\log\pi\left(T_\theta\left(Z\right)\right) + \log\left|\det J_{T_\theta}\left(Z\right)\right|\right] \tag{2.11}$$

where the constant is with respect to θ. Often the KL is estimated using Monte Carlo:

$$\begin{aligned} KL\left(T_\theta\#\nu\,\|\pi\right) &\approx \frac{1}{K}\sum_{k=1}^{K}\log\left(\frac{T_\theta\#\nu\left(X_k\right)}{\pi\left(X_k\right)}\right), \qquad X_k \sim T_\theta\#\nu, k\in[K] \\ &= \frac{1}{K}\sum_{k=1}^{K}\log\nu\left(T_\theta^{-1}\left(X_k\right)\right) + \log\left|\det J_{T_\theta^{-1}}\left(X_k\right)\right| + \text{const.} \end{aligned} \tag{2.12}$$

Therefore the transformation needs to be constructed such that we are able to evaluate $\nabla_\theta \log\left|\det J_{T_\theta}\left(Z\right)\right|$ or $\nabla_\theta \log\left|\det J_{T_\theta^{-1}}\left(X_k\right)\right|$, whether we can calculate the expectation in 2.11 exactly or whether we need to use 2.12 to estimate the objective. When we move to the MCMC setting, we will also need to be able to calculate the density of $T_\theta\#\nu$, which will require evaluations of $\left|\det J_{T_\theta}\left(x\right)\right|^{-1}$.

This transformation methodology for VI is undertaken in the fields of measure transport [Kim et al. 2013; Parno and Y. M. Marzouk 2018] and normalising flows [Brofos et al. 2022; M. Hoffman, Sountsov, et al. 2019; Kanwar 2024; Rezende and Mohamed 2015]. The discriminating factor between these two fields is the type of transformation. The motivating structure in the field of measure transport is the Knothe-Rosenblatt map, which is a transformation of the form

$$T\left(z\right) = \begin{pmatrix} T_1\left(z_1\right) \\ T_2\left(z_1, z_2\right) \\ \vdots \\ T_i\left(z_1,\ldots,z_i\right) \\ \vdots \\ T_d\left(z_1,\ldots,z_d\right) \end{pmatrix} \tag{2.13}$$

that is constructible (using explicit information about π) [Villani 2009, Chapter 1], unique given the source and target measure ν and π, and is the limiting solution to a particular formulation of the optimal transport problem [Bonnotte 2013; Carlier, Galichon, and Santambrogio 2010]. Using maps in the form 2.13 has a few practical benefits. One can easily express the determinant of the Jacobian as $\det J_T(z) = \prod_{i=1}^{d} \partial_i T_i(z)$. Inversion of T consists of d inversions of one dimensional maps. Monotonicity of T is guaranteed when $\partial_i T_i(z) > 0$ for all $i \in [d]$, for all $z \in$ Z. If $T\#\nu = \pi$ then the conditional distributions $\pi(.\,|x_1, \ldots, x_i)$ are naturally generated by T_i for all $i \in [d]$, see [Baptista et al. 2024, Theorem 2.4] for details. Having settled on the generic form 2.13 it remains to construct the maps T_i for $i \in [d]$. This is usually done by selecting a basis of parametrisable functions constructed in a way as to aid in the learning process. For instance [Kim et al. 2013; Parno and Y. M. Marzouk 2018] use polynomials, and [Spantini, Baptista, and Y. Marzouk 2022, Appendix A] use maps with elements of the form $T_i(z_1, \ldots, z_i) = u_i(z_1, \ldots, z_{i-1}) + u_i^{(i)}(z_i)$ for whom the condition $\partial_i T_i(z) > 0$ is guaranteed by the monotonicity of $u_i^{(i)}$ for all $i \in [d]$.

As we mentioned before, the main characteristic that separates normalising flows from measure transport is the form of the transformation. Whilst normalising flows are also constructed to have the nice computational properties (easily invertible, triangular, easy to calculate the Jacobian) of the transformations in measure transport, they typically incorporate neural networks and are comprised of compositions of many maps. Many normalising flow methods are devised outside the context of MCMC, and incorporated in methods following their conception, as we shall see in the next section. This is not so with measure transport: usually these techniques are conceived in conjunction with MCMC methods.

2.2.2 Variational approximations within sampling

After having derived an adequate variational approximation it may seem like our job is done since we can sample from it, and therefore compute expectations with respect to it. Indeed, this is the attitude taken by many. However, in practicality, we will often run into instances in which our variational approximation is good but not perfect, or perhaps lacks some key feature of the target distribution, such as correlations in the case of CAVI. Therefore we have fast but imperfect access to the target distribution. In MCMC the situation is the other way around: if we have an MCMC algorithm such that the Markov chain it produces satisfies the assumptions of Theorem 9, then we have ergodicity, and the estimators we derive from the chain will abide by LLNs for appropriate functions f, see section 1.3.1.3 for details. Therefore, absent other guarantees, we know that if we wait long enough, we will get as good an approximation to $\mathbb{E}_\pi\left[f\left(X\right)\right]$ as we want, and we have slow but perfect access to the target distribution.

The prospect of fast, perfect access to the target distribution then motivates the fusion of VI and MCMC. If we have a variational approximation to π, how are we to create these methods? The first way is to embed the variational approximation into a Monte Carlo method. If we have a good variational approximation ν_θ such that we could control $\left\|d\pi/d\nu_\theta\right\|_\infty$ then we could use ν_θ as the proposal distribution in a rejection sampler 1.2.2.2 and be assured of the expected time to achieve a sample, see 5. A variant of this solution constitutes our contribution to the methods which fuse VI and MCMC, see chapter 4 for the details. See also the Neural-Importance Sampler [Müller et al. 2019] and the Boltzmann Generator [Noé et al. 2019] for this idea in the context of Importance Sampling.

If the VI method renders us with a variational approximation in the form $T_\theta\#\nu$ then we can run an MCMC method on the target $T_\theta^{-1}\#\pi$ which, if the VI method worked well, should be approximately equal to ν. Given approximate samples from $T_\theta^{-1}\#\pi$, we can then transform them using T_θ to get approximate samples from π. If ν is a standard Gaussian distribution and $T_\theta^{-1}\#\pi$ is exactly equal to ν then the MCMC method will work well for numerous different reasons. By [Andrieu, A. Lee, et al. 2024, Corollary 39] we can control the average acceptance rate of RWM from below. The Langevin diffusion will have spectral gap of exactly one, and will be separable along the dimensions of the target and therefore simulable in parallel. The condition number 2.4 is also optimal at exactly one. Suffice it to say that if $T_\theta^{-1}\#\pi$ is a standard Gaussian, the situation is ideal, and if $T_\theta^{-1}\#\pi$ is a close approximation to a standard Gaussian, the situation is close to ideal. This particular way of fusing VI with MCMC is instantiated in [M. Hoffman, Sountsov, et al. 2019; Parno and Y. M. Marzouk 2018]. In the language of section 2.1.2.4, T_θ is a nonlinear preconditioner. One final thing to note is that, given that we're using exact methods such as MCMC, we have access to approximate samples from the target π. Therefore we needn't necessarily use the reverse KL, as outlined in section 2.2.1, and we can instead approximate the forward KL, which is in the form of an expectation with respect to π. This is in fact the method proposed in [Parno and Y. M. Marzouk 2018].

Algorithm 2.1 Generic adaptive MCMC algorithm

inputs: Markov kernels $\{K_L : \mathsf{X} \times \mathcal{X} \to [0,1]\}_{L \in \mathbb{R}^{d\times d}}$, learning mechanisms $\left\{A : \mathsf{X}^t \times \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}\right\}_{t \in \mathbb{N}\setminus\{0\}}$, initial state $X_0 \in \mathsf{X}$, initial preconditioner $L_0 \in \mathbb{R}^{d\times d}$, chain length $n \in \mathbb{N}\setminus\{0\}$

outputs: A process $\{X_t\}_{t=1}^n$

```
For $t \in [n]$ do
  1. Sample $X_t \sim K_{L_{t-1}}(X_{t-1} \to .)$
  2. Update the preconditioner:  $L_t = A(X_0, \ldots, X_t, L_{t-1})$
```

2.3 Adaptivity

In chapter 3 we will examine the ways in which a fixed linear preconditioner $L \in \mathbb{R}^{d\times d}$ affects the performance of MCMC algorithms. In the course of our examinations will see examples of preconditioners that are stated in terms of expectations with respect to the target. These expectations include the covariance and the 'Fisher' matrix introduced in [Michalis Titsias 2023] which is simply $\mathbb{E}_\pi\left[\nabla^2(-\log \pi(X))\right]$. Absent any methods to estimate these matrices before the start of the MCMC process, practitioners who wish to use them to precondition will often estimate them using information from the chain. Then with each new estimate we can build a preconditioner which we can use in the next step of the MCMC. This procedure is referred to as 'adaptive MCMC'. See [Andrieu and É. Moulines 2006; Andrieu and Thoms 2008; Laitinen and Vihola 2024; Gareth O. Roberts and Jeffrey S. Rosenthal 2007] for theory of the generic practice of adaptive MCMC, [Haario, Saksman, and Tamminen 2001; Michalis Titsias 2023; Vihola 2012; Wallin and Bolin 2018] for examples of particular adaptive MCMC algorithms. We can view the kernel $K_L : \mathsf{X} \times \mathcal{X} \to [0,1]$ of the MCMC algorithm as having the preconditioner L as a parameter. Each adaptive algorithm will also incorporate a learning function at time n that takes the history of the chain $\{X_t\}_{t=1}^n$ (and possibly some extra information, such as the proposals), and the most recent estimate of the expectation L_{n-1} and output the next estimate of the expectation L_n. This new estimate will then serve as the preconditioner which parametrises the next Markov kernel that generates the subsequent state in the chain. See 2.1 for a generic representation of the process. The final aim is to average $f : \mathsf{X} \to \mathbb{R}$ over $\{X_t\}_{t=1}^n$ to gain an estimate for $\mathbb{E}_\pi[f(X)]$ as we would do with the output of a non-adaptive method.

The Markov kernel $K_{L_{t-1}}$ is usually the kernel of a canonical MCMC algorithm which has been preconditioned, in the sense that the Markov chain generated by a π-invariant K_L is isomorphic to the Markov chain generated by a $L\#\pi$-invariant $K_{\mathbf{I}_d}$, see section 2.1.2.5 for a more detailed exposition of this fact.

2.3.1 The structure of learning mechanisms

The learning mechanism $A\left(X_0, \ldots, X_t, L_{t-1}\right)$ usually falls into one of two categories, depending on the form of the ideal preconditioner. In the first category, the ideal preconditioner takes the form $L = g\left(\mathbb{E}_\pi\left[f\left(X\right)\right]\right)$ where $f : \mathsf{X} \to \mathbb{R}^{d\times d}$ and $g : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$. For instance, in the case of preconditioning with the covariance of the target distribution 3.2.2, we will have that $f\left(x\right) := \left(x - \pi\left(x\right)\right)\left(x - \pi\left(x\right)\right)^T$ and $g\left(A\right) := A^{-1/2}$. Preconditioning with the diagonals of the covariance of the target 3.5 has $f\left(x\right) :=$ diag $\left(\left(x - \pi\left(x\right)\right)\left(x - \pi\left(x\right)\right)^T\right)$ and $g\left(A\right) := A^{-1/2}$[2]. In the case of preconditioning with the Fisher matrix as in [Michalis Titsias 2023] 3.2.1, we will have that $f\left(x\right) := \nabla \log \pi\left(x\right) \nabla \log \pi\left(x\right)^T$ and $g\left(A\right) := A^{1/2}$. In this case we can use the Markov chain to form the following estimator of the expectation: $\hat{f}_n := \left(n - t_0\right)^{-1} \sum_{t=t_0}^{n} f\left(X_t\right)$ where $t_0 \in \mathbb{N}\backslash\left\{0\right\}$ is the time at which we start adapting, and is sometimes chosen as the end of the burn-in. This estimator can be updated online as follows:

$$\hat{f}_n = \hat{f}_{n-1} + \frac{1}{n - t_0}\left(f\left(X_n\right) - \hat{f}_{n-1}\right) \tag{2.14}$$

The final ingredient for the learning mechanism is then to somehow go from $g\left(\hat{f}_{n-1}\right)$ to $g\left(\hat{f}_n\right)$. In the cases where g involves a matrix square root, and f involves an outer product, this can be achieved with a rank-1 Cholesky update. The form of 2.14 dictates that the extended state $\left(X_t, \hat{f}_t\right)$ is Markovian, which is a very helpful fact in establishing theoretical results about the adaptive algorithm. See [Haario, Saksman, and Tamminen 2001; Michalis Titsias 2023] for instances within this category.

In the second category, the ideal preconditioner takes the form of an optimal variational parameter, see section 2.2.1 for an introduction to Variational Inference. For instance, argmin$_{\Sigma>0} KL\left(\pi \,\|\mathcal{N}\left(0, \Sigma\right)\right) =$ Cov$_\pi\left(X\right)$ (where this quantity exists) and we are back to preconditioning with the covariance. The forward KL is expressed as an expectation with respect to the target. This property is shared by many optimisation objectives in adaptive MCMC, see [Brofos et al. 2022; Hirt, M. Titsias, and P. Dellaportas 2021; Song, Zhao, and Ermon 2018], and its presence makes sense since these preconditioners are learned to be optimal

[2] or perhaps $f\left(x\right) := \left(x - \pi\left(x\right)\right)\left(x - \pi\left(x\right)\right)^T$ and $g\left(A\right) :=$ diag $\left(A\right)^{-1/2}$ since the diag operator commutes with expectations.

in the equilibrium regime of the Markov chain. One can then use the samples from the chain to estimate gradients of the objective with respect to the parameter to be used in a gradient-descent type scheme which is run in parallel with the Markov chain:

$$\theta_{t+1} = \theta_t - \gamma_t \hat{g}\left(\theta_t; X_0, \ldots, X_t\right)$$

where $\hat{g} : \Theta \times \mathsf{X}^{t+1} \to \Theta$ is an estimate of the gradient of the objective and $\gamma_t > 0$ is a learning rate. When $\hat{g}$ depends only on θ_t and X_t we have that (X_t, θ_t) is Markovian.

2.3.2 Adaptive MCMC theory

There have been many efforts in the past 25 years to analyse adaptive MCMC algorithms formally. The extent of this analysis has been to establish conditions under which the adaptive mechanism maintains ‘good properties’ of existing non-adaptive algorithms, where by ‘good properties’ we mean the existence of an LLN , a CLT, preservation of ergodicity, or any other such results.

2.3.2.1 Positive theory

[Haario, Saksman, and Tamminen 2001] give a strong LLN for the chain generated with their adaptive mechanism detailed in section 2.3.3. However this LLN is for bounded functions, and targets supported on a compact set. [Saksman and Vihola 2010] extend the results of [Haario, Saksman, and Tamminen 2001] to non-compact supports and unbounded functions by requiring that π is super-exponentially light-tailed. [Atchadé 2010] give conditions on the convergence and stability of the Markov operators K_{L_t} and their invariant distributions, for all $t \in \mathbb{N} \backslash \{0\}$, under which a LLN holds. [Andrieu and É. Moulines 2006] give a decomposition of the summand in the Markov chain estimator resulting in 3 terms which are easier to handle theoretically. One term constitutes the summand of a telescoping sum, and one is a zero mean Martingale. The last describes the innovations due to the learning step and must be controlled. [Laitinen and Vihola 2024] use this decomposition to describe conditions under which the Markov chain estimator satisfies weak and strong LLNs and a CLT. One notable condition is that of simultaneous geometric ergodicity, which assumes that each K_L is geometrically ergodic 1.3.1.3 with a constant that is independent of the initial state, for all preconditioners

L in the parameter space. [Gareth O. Roberts and Jeffrey S. Rosenthal 2007] assert conditions that have come to be known as 'diminishing adaptation' and 'containment' under which they guarantee ergodicity and a weak LLN using coupling arguments. Diminishing adaptation simply requires that the differences between kernels with subsequent adaptive parameters given by $\sup_{x \in \mathsf{X}} \left\| K_{L_{t+1}} \left(x \to .\right) - K_{L_t} \left(x \to .\right) \right\|_{TV}$ converges to zero in probability. Containment requires that the sequence of ε-mixing times of the Markov kernels at each subsequent adaptive parameter is bounded in probability.

2.3.2.2 Counterexamples

The works mentioned above posit conditions under which adaptive MCMC satisfy the basic desiderata for sampling algorithms. However there also exist instructive counterexamples in the literature: pathological cases in which adaptation actively harms the sampling algorithm. [Andrieu and É. Moulines 2006] offer a simple example on the state space $\mathsf{X} = \{1, 2\}$ in which naively allowing the parameters of the transition matrix depend on the latest state X_i alters the invariant distribution of the process, even though it retains Markoviannity and time-homogeneity. Abstractly, if $K_{\theta(X_i)}$ is no longer π-invariant then, even if $X_i \sim \pi$ we would have

$$\begin{aligned} \mathbb{E}\left[f\left(X_{i+1}\right)\right] &= \mathbb{E}_\pi\left[\mathbb{E}\left[f\left(X_{i+1}\right)|X_i\right]\right] \\ &= \int_{\mathsf{X}\times\mathsf{X}} \pi\left(dx\right) K_{\theta(x)}\left(x \to dx'\right) f\left(x'\right) \neq \mathbb{E}_\pi\left[f\left(X\right)\right] \end{aligned}$$

[Andrieu and Thoms 2008] use the same example to discuss the proposed solution to the problem in [Andrieu and É. Moulines 2006] that relies on allowing the adaptive parameter to depend on the next-to-last state X_{i-1} instead of the last state. Let's say that now $K_{\theta(X_{i-1})}$ is π-invariant and that $X_i \sim \pi$. Then for all functions $f : \mathsf{X} \to \mathbb{R}$ which are measurable with respect to the law of the process

$$\begin{aligned} \mathbb{E}\left[f\left(X_{i+1}\right)\right] &= \mathbb{E}\left[\mathbb{E}\left[f\left(X_{i+1}\right)|X_i\right]\right] \\ &= \mathbb{E}_\pi\left[\mathbb{E}\left[f\left(X_{i+1}\right)|X\right]\right] \end{aligned}$$

where the expectations are with respect to the randomness in the process up to time $i + 1$ and, since $\mathbb{E}\left[f\left(X_{i+1}\right)|.\right]$ is a measurable function with respect to the process, we have the final equality. Continuing:

$$\begin{aligned}\mathbb{E}\left[f\left(X_{i+1}\right)\right] &= \mathbb{E}_{\pi}\left[\mathbb{E}\left[f\left(X_{i+1}\right)|X\right]\right] \\ &= \mathbb{E}_{\pi}\left[\mathbb{E}\left[\mathbb{E}\left[f\left(X_{i+1}\right)|X_{i-1}, X\right]|X\right]\right]\end{aligned} \tag{2.15}$$

Now, even though $\mathbb{E}_{\pi}\left[\mathbb{E}\left[f\left(X_{i+1}\right)|X_{i-1}, X\right]\right] = \mathbb{E}_{\pi}\left[f\left(X\right)\right]$ due to the π-invariance of $K_{\theta(X_{i-1})}$, we do not have $\mathbb{E}\left[f\left(X_{i+1}\right)\right] = \mathbb{E}_{\pi}\left[f\left(X\right)\right]$ because the outer two expectations in 2.15 do not commute.

The proposed solution is to gradually allow the adaptive parameter to become less and less dependent on the states in the chain. This is naturally achieved by allowing the parameter to converge in some sense. For instance, the convergence may be that of Diminishing Adaptation discussed in the previous paragraph. [Andrieu and Thoms 2008] discuss a simple case in which allowing the adaptive parameters to converge deterministically to a set of values which render the resulting Markov kernel non-ergodic but still π-invariant would break ergodicity.

[Atchadé 2010] offer a simple, seemingly sensible adaptive accept-reject algorithm on a finite state space, in which the proposal variance increases upon acceptance and decreases upon rejection. The resulting Markov chain has an equilibrium distribution which does not equal that of the kernel with any fixed proposal. Therefore, even if the kernel with any particular proposal is π-invariant, the overall process is not. [Gareth O. Roberts and Jeffrey S. Rosenthal 2007] extend this to a finite but arbitrarily large state space with a target that has a communication barrier i.e. a state over which the Markov chain will have to 'jump over' to pass. Therefore the variance of the proposal kernel needs to be inflated adaptively to a point at which the Markov chain is able to do this. [Gareth O. Roberts and Jeffrey S. Rosenthal 2007] show that if the proposal cannot do this, the Markov process will get stuck and no longer be ergodic. If the proposal can do this, then [Gareth O. Roberts and Jeffrey S. Rosenthal 2007] show that this does not break the ergodicity of the process, so long as the probability with which the proposal variance changes goes to zero with time.

2.3.3 Two example algorithms and their computational complexity

In [Haario, Saksman, and Tamminen 2001] the authors attempt to learn $L = \text{Cov}_\pi\left(X\right)^{-1/2}$. To define the Markov kernel $K_{L_{t-1}}$ on the state space X they use

$$X_t := \mathbb{1}\left\{U_{t-1} \le \alpha\left(X_{t-1} \to Y_{t-1}\right)\right\} Y_t + \mathbb{1}\left\{U_{t-1} > \alpha\left(X_{t-1} \to Y_{t-1}\right)\right\} X_{t-1} \tag{2.16}$$

with $U_{t-1} \sim$ Uniform $[0,1]$ and $Y_{t-1} \sim \mathcal{N}\left(X_{t-1}, \left(L_{t-1}L_{t-1}^T\right)^{-1}\right)$. The acceptance probability α is as defined in 1.2, with the proposal densities cancelling out as is usual in a random walk MCMC method. To define the learning mechanism they choose an initial time $t_0 \in [n]$ at which they begin. They maintain an estimate of the covariance of the target as follows:

$$C_t := \begin{cases} C_0 & t \le t_0 \\ s_d\text{Cov}\left(X_0, \ldots, X_t\right) + s_d\varepsilon\mathbf{I}_d & t > t_0 \end{cases}$$

where Cov $\left(X_0, \ldots, X_t\right)$ is the empirical covariance between the states of the Markov chain, $C_0 \in \mathbb{R}^{d\times d}$ is an initial positive definite estimate, $s_d > 0$ is a step-size, $\varepsilon > 0$ is a small constant to ensure the positive definiteness of estimates C_t. Then the actual preconditioner can be formed as $L_t = C_t^{-1/2}$. The authors use the following recursion:

$$C_t = \frac{t}{t+1}C_{t-1} + \frac{s_d}{t+1}\left(t\bar{X}_{t-1}\bar{X}_{t-1}^T - \left(t+1\right)\bar{X}_t\bar{X}_t^T + X_tX_t^T + \varepsilon\mathbf{I}_d\right) \tag{2.17}$$

for all $t \ge t_0 + 1$ to update their estimate of the covariance, where $\bar{X}_t := \left(t+1\right)^{-1}\sum_{k=0}^t X_k$. Since $\bar{X}_t$ can be updated using $\bar{X}_{t-1}$ and X_t, we have that $\left(X_t, \bar{X}_t, C_t\right)$ is Markovian. Note that to sample from $\mathcal{N}\left(X_{t-1}, \left(L_{t-1}L_{t-1}^T\right)^{-1}\right)$ we do not need $C_{t-1}^{-1/2}$ although we do need to calculate $C_{t-1}^{1/2}$ for some notion of the square root. This can be achieved using three rank-1 updates to $C_{t-2}^{1/2}$, each of which has $O\left(d^2\right)$ computational complexity.

[Michalis Titsias 2023] learns $L = \mathbb{E}_\pi\left[\nabla^2 U\left(X\right)\right]^{1/2}$. The proposed method also uses 2.16 to construct the Markov kernel on X, except here we have that

$$Y_{t-1} \sim \mathcal{N}\left(X_{t-1} + \frac{\sigma^2}{2}\left(\frac{L_{t-1}L_{t-1}^T}{H\left(L_{t-1}L_{t-1}^T\right)}\right)^{-1}\nabla \log \pi\left(X_{t-1}\right), \sigma^2\left(\frac{L_{t-1}L_{t-1}^T}{H\left(L_{t-1}L_{t-1}^T\right)}\right)^{-1}\right)$$

where $H\left(A\right)$ is the harmonic mean of the eigenvalues of a positive definite matrix $A \in \mathbb{R}^{d\times d}$, and the acceptance probability α is as defined in 1.2. This is the usual MALA kernel 1.6, preconditioned with L_{t-1}, which has been regularised by the harmonic mean of its eigenvalues. The quantity $\sigma^2 > 0$ now serves as a step-size. As detailed in [Michalis Titsias 2023, Section 4.1] the matrix LL^T is learned to be the covariance of the increment of the score across the chain, which has been Rao-Blackwellised in the randomness determining the accept-reject mechanism:

$$\begin{aligned} L_tL_t^T &:= \mathsf{Cov}\left(\mathbb{E}\left[\nabla \log \pi\left(X_1\right) - \nabla \log \pi\left(X_0\right)\right], \ldots, \mathbb{E}\left[\nabla \log \pi\left(X_t\right) - \nabla \log \pi\left(X_{t-1}\right)\right]\right) \qquad (2.18)\\ &= \mathsf{Cov}\left(\sqrt{\alpha\left(X_0 \to Y_0\right)}\left(\nabla \log \pi\left(Y_0\right) - \nabla \log \pi\left(X_0\right)\right), \ldots, \sqrt{\alpha\left(X_{t-1} \to Y_{t-1}\right)}\left(\nabla \log \pi\left(Y_{t-1}\right) - \nabla \log \pi\left(X_{t-1}\right)\right)\right) \end{aligned}$$

where the expectations in the first line are with respect to the randomness determining whether the proposals are accepted. Even though the intention was to learn $LL^T = \mathbb{E}_\pi\left[\nabla^2 U\left(X\right)\right]$, [Michalis Titsias 2023] argues that the strategy in 2.18 is superior because the score increment is ‘more centred [than the score] and close to zero’ when the chain is transient. The Rao-Blackwellisation aims to reduce the variance of the resulting estimator. Due to the fact that the learning mechanism must take place in the square root of LL^T, and due to the fact that the square root must be inverted to propose new states, an iteration takes $O\left(d^2\right)$ computational complexity (assuming that the intermediate operations such as the calculation of the score are also $O\left(d^2\right)$) similarly to [Haario, Saksman, and Tamminen 2001].

Chapter 3

Quantifying the effectiveness of linear preconditioning in MCMC

In this chapter we will quantify the effects of linear preconditioners on the condition number κ as defined in section 2.1.2.3. First we show that well-conditioned distributions exist for which κ can be arbitrarily large and yet no linear preconditioner can reduce it. We then impose two sets of extra assumptions under which a linear preconditioner can significantly reduce κ. For the random walk Metropolis we further provide upper and lower bounds on the spectral gap with tight $1/\kappa$ dependence. This allows us to give conditions under which linear preconditioning can provably increase the gap. We then study popular preconditioners such as the covariance, its diagonal approximation, the Hessian at the mode, and the QR decomposition. We show conditions under which each of these reduce κ to near its minimum. We also show that the diagonal approach can in fact *increase* the condition number. This is of interest as diagonal preconditioning is the default choice in well-known software packages. We conclude with a numerical study comparing preconditioners in different models.

The notation in this chapter can be found to be defined in section 7.1.3.

3.1 Unpreconditionable distributions

Methods to adaptively seek a preconditioner are implemented in the MCMC samplers provided by the major software packages. For instance the HMC sampler in the popular statistical modeling platform Stan [Carpenter et al. 2017] offers the ability to infer the target covariance $\Sigma_\pi \in \mathbb{R}^{d\times d}$ giving an estimator $\hat{\Sigma}_\pi \in \mathbb{R}^{d\times d}$ to use as the inverse mass matrix. Proposition 26 has that this is equivalent to linear preconditioning with $L = \hat{\Sigma}_\pi^{-\frac{1}{2}}$. Since computational effort is required to infer $\hat{\Sigma}_\pi$ the idea is that preconditioning with $L = \hat{\Sigma}_\pi^{-\frac{1}{2}}$ is better than doing nothing. Even within the class of models whose potentials satisfy assumption 24 this will not always be the case

Proposition 27. *There exist distributions whose potentials satisfy definition 24 for which any non-orthogonal linear preconditioner will cause the condition number to increase.*

On such distribution has density of the form $\pi(x,y) \propto \exp(-U(x,y))$ where

$$U(x,y) = \frac{m-M}{2}(\cos x + \cos y) + \frac{M+m}{2}\left(\frac{x^2}{2} + \frac{y^2}{2}\right) \tag{3.1}$$

for $x, y \in \mathbb{R}$. For such a target the condition number after preconditioning $\tilde{\kappa}$ is bounded below by $\kappa\left(LL^T\right)\kappa$[1]. The Hessian of U is in the form diag$\{f(x), f(y)\}$ where f ranges freely in $[m, M]$. The lower bound $\kappa\left(LL^T\right)\kappa$ highlights that any preconditioning causes the target to be more ill-conditioned by an amount exactly proportional to $\kappa\left(LL^T\right)$. The full proof of Proposition 27 can be found in 7.2.3.1.

The issue for the potential 3.1 is that every eigenvalue of $\nabla^2 U$ can assume both the value of m and M at given points in $\mathbb{R}^d$. In the following two sections we characterise effective linear preconditioners for two broad classes of models. We do this by establishing upper bounds on the condition number after linear preconditioning under model appropriate assumptions.

Throughout the rest of this chapter $\lambda_i(x)$ denotes the ith largest eigenvalue of $\nabla^2 U(x)$ and $v_i(x)$ the corresponding normalised eigenvector. Since the Hessian is everywhere symmetric its eigenvectors are orthogonal for a fixed $x \in \mathbb{R}^d$. Since L can be assumed to be symmetric we can then denote its ith eigenvalue as σ_i with associated eigenvector v_i.

[1] Here $\kappa(.) : \mathbb{R}^{d\times d} \to \mathbb{R}^+$ is a function on the positive definite matrices which outputs $\kappa(A) := \lambda_{\text{max}}(A)/\lambda_{\text{min}}(A)$ where $\lambda_{\text{max}}(A)$ and $\lambda_{\text{max}}(A)$ are the maximum and minimum eigenvalues of A respectively.

3.2 Linear preconditioning for additive Hessians

We call a Hessian additive if it has the form

$$\nabla^2 U(x) = A + B(x) \tag{3.2}$$

where $A, B(x) \in \mathbb{R}^{d \times d}$ are symmetric. Even though all Hessians assume this form, we will see that those most amenable to linear preconditioning are those for which $B(x)$ varies little across the state space. Examples of models whose potentials have Hessians in this form include: Gaussians ($B(x) \equiv 0$), strongly log-concave mixtures of Gaussians [Dalalyan 2017, Section 6.1], and Bayesian Huberised regressions with strongly log-concave priors [Rosset and Zhu 2004]. The results in this section are presented in generality, but the assumptions under which they hold are particularly appropriate for models with an additive Hessian.

We first present a general result under the following assumptions on the eigenstructure of $\nabla^2 U$ and L.

Assumption 28. *There exists an $\varepsilon > 0$ such that*

$$(1+\varepsilon)^{-1} \leq \frac{\lambda_i(x)}{\sigma_i^2} \leq 1 + \varepsilon$$

for all $i \in [d]$ and $x \in \mathbb{R}^d$.

Assumption 29. *There exists a $\delta > 0$ such that $v_i(x)^T v_i \geq 1 - \left(1 - \sqrt{1-\delta}\right)^2$ for all $i \in [d]$ and $x \in \mathbb{R}^d$.*

Theorem 30. *Let π have a potential U satisfying assumption 24. For a given preconditioner $L \in \mathbb{R}^{d\times d}$ for which assumptions 28 and 29 hold, the condition number after preconditioning satisfies*

$$\tilde{\kappa} \leq (1+\varepsilon)^2 \left(1 + \delta \sqrt{\sum_{i=1}^{d} \sigma_i^2 \sum_{i=1}^{d} \sigma_i^{-2}}\right)^4$$

Proof can be found in section 7.2.3.2. Assumption 28 states that the eigenvalues of $\nabla^2 U(x)$ do not change much over $\mathbb{R}^d$. Assumption 29 implies that $v_i(x)^T v_i \geq 1 - \delta$ for $i \in [d], x, y \in \mathbb{R}^d$ and $v_i(x)^T v_j \leq \delta$ for $i \in [d], x, y \in \mathbb{R}^d$ where $i \neq j$ (see section 7.2.3.3 for details). In the Gaussian case $\varepsilon = \delta = 0$ when $L = \Sigma_\pi^{-\frac{1}{2}}$ and the bound becomes $\tilde{\kappa} \leq 1$ as expected.

Remark 31. In the case of the additive Hessian, the eigenvalue stability inequality [Tao 2012, (1.63)] implies that $\lambda_i (A) - \|B(x)\|_2 \leq \lambda_i (A + B(x)) \leq \lambda_i (A) + \|B(x)\|_2$ for all $i \in [d]$. Therefore choosing $L = A^{\frac{1}{2}}$ gives us that

$$1 - \frac{\|B(x)\|_2}{\lambda_d (x)} \leq \frac{\lambda_i (x)}{\sigma_i^2} \leq 1 + \frac{\|B(x)\|_2}{\lambda_d (x)}$$

for all $i \in [d]$. So if $\|B(x)\|_2$ is small, $L = A^{\frac{1}{2}}$ will yield a smaller ε. If $\|B(x)\|_2$ is large, but $B(x)$ does not exhibit large variations we can simply restate the Hessian as $\nabla^2 U(x) = \tilde{A} + \tilde{B}(x)$ where $\tilde{A} := A + B(x^*)$ and $\tilde{B}(x) := B(x) - B(x^*)$ where $x^* \in \mathbb{R}^d$ is some point in the state space, and use $L = \tilde{A}^{\frac{1}{2}}$.

Remark 32. Note that the quantity $\sqrt{\sum_i \sigma_i^2 \sum_i \sigma_i^{-2}} = \sqrt{\text{Tr}\left(LL^T\right) \text{Tr}\left(\left(LL^T\right)^{-1}\right)}$ is upper bounded by $d\sqrt{\kappa\left(LL^T\right)}$. It can be viewed as an alternative ‘condition number’ of LL^T, similar to that proposed in [Langmore et al. 2020].

The fact that $\sqrt{\sum_i \sigma_i^2 \sum_i \sigma_i^{-2}}$ multiplies δ shows that when sampling from highly anisotropic distributions the penalty for misaligned eigenvectors of LL^T relative to $\nabla^2 U$ is larger. As an example, take $\pi = \mathcal{N}(0, \Sigma_\pi)$ where $\Sigma_\pi \in \mathbb{R}^{2\times 2}$ has eigendecomposition $\Sigma_\pi = Q_\pi D_\pi Q_\pi^T$ with $D_\pi = \text{diag}\{\lambda_1, \lambda_2\}$. We assume that Σ_π is not a multiple of the identity. Construct the preconditioner $L = Q_\pi G D_\pi^{-\frac{1}{2}} G^T Q_\pi^T$ with correct eigenvalues ($\varepsilon = 0$ in assumption 28) but whose eigenvectors have been perturbed by an orthogonal matrix G from those of Σ_π by the angle $\arccos(1-\delta)$. It can be shown that the coefficient of δ^4 in $\tilde{\kappa}$ is $(1/4) \times (l-2)^2$ where $l := \lambda_1 \lambda_2^{-1} + \lambda_1^{-1}\lambda_2$. So the more anisotropic Σ_π is the more we are punished for having misaligned eigenvectors, as stated in the remark above. Figure 3.1 illustrates this fact.

Each plot contains two contours: a blue one representing $\mathcal{N}(0, \Sigma_\pi)$ and an orange one representing $\mathcal{N}\left(0, \left(LL^T\right)^{-1}\right)$. In both cases G perturbs the eigenvectors by $\pi/4$, the angle between the semi-major axes of the contours is shown in the red arrows. In the first case we have $(\lambda_1, \lambda_2) = (2, 1)$, in the second we have $(\lambda_1, \lambda_2) = (50, 1)$ engendering a far smaller ‘overlap’ than in the first. The fact that $\tilde{\kappa} = O\left(\delta^4\right)$ also shows that the δ dependency in Theorem 30 is tight.

Assumptions 28 and 29 require knowledge of each individual eigenvalue and eigenvector of $\nabla^2 U$ across the entire state space, which may not always be available. We next provide more easily verifiable assumptions under which a similar result holds. This is achieved using results from matrix perturbation theory. The eigenvalues of $\nabla^2 U$ can be controlled using only knowledge of the spectral norm by Weyl’s inequality (see

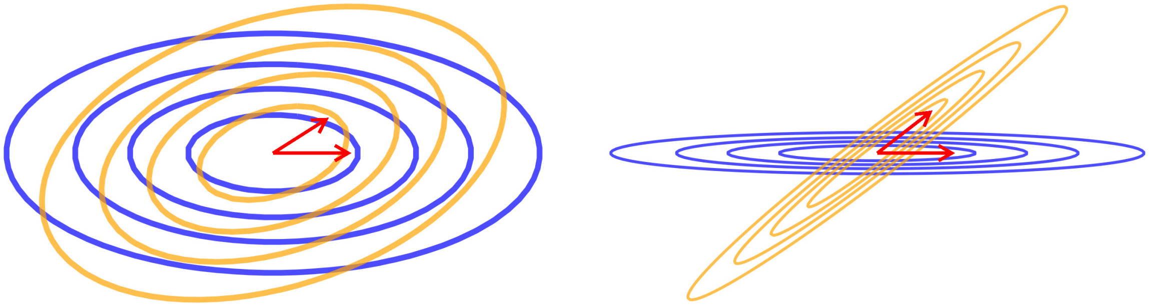

Figure 3.1: Two pairs of contour plots, each representing $\mathcal{N}\left(0,\Sigma_{\pi}\right)$(blue) and $\mathcal{N}\left(0,\left(LL^{T}\right)^{-1}\right)$(orange). The angle between the semi-major axes (red) is the same in either case, but the preconditioner in the right hand plot is worse due to the anisotropy of $\mathcal{N}\left(0,\Sigma_{\pi}\right)$.

assumption 33). Similarly, using the Davis-Kahan theorem (e.g. [Yu, Wang, and Samworth 2015]) eigenvectors can be controlled by the spectral norm provided that an ‘eigengap’ condition holds (see assumption 34). We therefore provide a second result under assumptions 33 and 34 below.

Assumption 33. *There exists an $\varepsilon>0$ such that $\left\|\nabla^{2}U\left(x\right)-LL^{T}\right\|_{2}\leq\sigma_{d}^{2}\varepsilon$ for all $x\in\mathbb{R}^{d}$.*

Assumption 34. *That $\gamma>0$ where*

$$\gamma:=\inf_{i,j\in[d],|i-j|=1}\left|\sigma_{i}^{2}-\sigma_{j}^{2}\right|$$

is the eigengap of LL^{T}.

Theorem 35. *Let π have potential U satisfying assumption 24. For a given preconditioner $L\in\mathbb{R}^{d}$ for which assumptions 33 and 34 hold, the condition number after preconditioning satisfies*

$$\tilde{\kappa}\leq\left(1+\varepsilon\right)^{2}\left(1+\delta\sqrt{\sum_{i=1}^{d}\sigma_{i}^{2}\sum_{i=1}^{d}\sigma_{i}^{-1}}\right)^{4}$$

where $\delta:=1-\left(1-2\gamma^{-1}\sigma_{d}^{-2}\varepsilon\right)^{2}$.

For a proof see section 7.2.3.4. Based on this result, it might be tempting to arbitrarily increase the eigengap γ or the least eigenvalue σ_{d} of L. Note, however, that this may also cause ε to increase.

Remark 36. Section 7.2.3.5 shows that assumption 33 implies 28. Similarly assumption 33 and 34 combined imply assumption 29 [Yu, Wang, and Samworth 2015, Corollary 1]. If the norm $\left\|\nabla^2 U\left(x\right)-LL^T\right\|_2$ is difficult to compute the Frobenius norm can be used to form the upper bound since $\left\|\nabla^2 U\left(x\right)-LL^T\right\|_2 \leq \left\|\nabla^2 U\left(x\right)-LL^T\right\|_F$.

Remark 37. In the case of the additive Hessian an application of the triangle inequality gives

$$\left\|A+B\left(x\right)-LL^T\right\|_2 \leq \min\left\{\left\|A-LL^T\right\|_2+\left\|B\left(x\right)\right\|_2, \left\|B\left(x\right)-LL^T\right\|_2+\left\|A\right\|_2\right\}$$

for all $x\in\mathbb{R}^d$. Therefore assumption 33 suggests choosing $L=A^{\frac{1}{2}}$ if A is 'larger' than $B\left(x\right)$ across the state space and $L=B\left(x^*\right)^{\frac{1}{2}}$ for some $x^*\in\mathbb{R}^d$ if $B\left(x\right)$ is 'larger', but does not exhibit large variations across $\mathbb{R}^d$.

The eigengap assumption 34 is always satisfiable as we are free to choose L. It may not, however, be desirable. We therefore present a final bound on $\tilde{\kappa}$ that requires only assumption 33.

Theorem 38. *Let π have potential U satisfying assumption 24. For a given preconditioner $L\in\mathbb{R}^{d\times d}$ for which assumption 33 holds, the condition number after preconditioning satisfies*

$$\tilde{\kappa}\leq\left(1+\varepsilon\right)\left(1+\frac{\sigma_1^2}{m}\varepsilon\right)$$

For a proof see section 7.2.3.6. In the next two subsections we consider some popular choice of linear preconditioner that can be shown to reduce the condition number when the Hessian of the potential of π is of additive form, as described above.

3.2.1 The Fisher matrix

[Michalis Titsias 2023] suggests using $L\propto\mathcal{I}^{\frac{1}{2}}$ where $\mathcal{I}:=\mathbb{E}_\pi\left[\nabla U\left(x\right)\nabla U\left(x\right)^T\right]$ is called the Fisher matrix. This choice of preconditioner maximises the expected squared jump distance of the LMC. Integration by parts shows that $\mathcal{I}$ can also be written $\mathbb{E}_\pi\left[\nabla^2 U\left(x\right)\right]$, which highlights its relationship with $\nabla^2 U$. Proposition 39 shows that if a choice of L satisfying assumption 33 exists, then the alternative choice of preconditioner $\mathcal{I}^{\frac{1}{2}}$ will also be valid.

Proposition 39. *Let π have potential $U \in C^2$ and assume there exists a preconditioner $L \in \mathbb{R}^{d \times d}$ satisfying assumption 33 for some $\varepsilon > 0$. Then $\left\|\nabla^2 U(x) - \mathcal{I}\right\|_2 \leq 2\sigma_d^2 \varepsilon$ for all $x \in \mathbb{R}^d$ where σ_d^2 is the least eigenvalue of LL^T.*

For a proof see section 7.2.3.7.

Corollary 40. *Consider π with a potential satisfying assumption 24 and an $L \in \mathbb{R}^{d \times d}$ satisfying assumption 33 for some $\varepsilon > 0$. Then choosing the preconditioner $\mathcal{I}^{\frac{1}{2}}$ gives*

$$\tilde{\kappa} \leq (1 + 2\varepsilon)\left(1 + \frac{\sigma_1^2}{m} 2\varepsilon\right)$$

where σ_1^2 is the greatest eigenvalue of LL^T.

The proof is a direct application of Theorem 38.

3.2.2 The target covariance

Preconditioning with an estimate of the target covariance is a popular strategy. To give some intuition for why this strategy is sensible we study the spectral gap of the Ornstein-Uhlenbeck (O-U) process, and show how this changes under preconditioning. The preconditioned O-U process is an instance of the preconditioned Langevin diffusion on a $\mathcal{N}(0, \Sigma_\pi)$ target which is driven by the SDE:

$$dX_t = -\frac{1}{2}\left(LL^T\right)^{-1} \Sigma_\pi^{-1} X_t dt + L^{-1} dB_t$$

where $(B_t)_{t \geq 0}$ is a Brownian motion. Since our practical goal is to simulate this process on a computer, we must take care when interpreting results about its continuous time formulation. We therefore subject ourselves to the condition $\left|\det\left(-L^{-1}L^{-T}\Sigma_\pi^{-1}\right)\right| = 1$, which precludes the choice $L' := s^{-1}L$ for increasingly large $s > 0$, which would arbitrarily increase the rate of convergence of the continuous time process but destabilise the discretised process for any fixed numerical integrator step size. The result below is well known, but can be found, for instance, by modifying results in [Negrea 2022].

Proposition 41. *The preconditioner $L = \Sigma_\pi^{-\frac{1}{2}}$ maximises the spectral gap of the preconditioned O-U process subject to $\left|\det\left(-L^{-1}L^{-T}\Sigma_\pi^{-1}\right)\right| = 1$.*

Here we provide results on how localised the Hessian of the potential is around Σ_π^{-1} in order to apply assumption 33 with $L = \Sigma_\pi^{-\frac{1}{2}}$.

Proposition 42. *Let π be a distribution with potential $U \in C^2$, covariance $\Sigma_\pi \in \mathbb{R}^{d\times d}$, mode $x^* \in \mathbb{R}^d$, and expectation $\mu_\pi \in \mathbb{R}^d$. Assume that there exist positive definite matrices $\Delta_+, \Delta_- \in \mathbb{R}^{d\times d}$ such that $\Delta_- \preceq \nabla^2 U(x) \preceq \Delta_+$ for all $x \in \mathbb{R}^d$ and that $1 - (x^* - \mu_\pi)^T \Delta_+ (x^* - \mu_\pi) > 0$. Then $P_- \preceq \Sigma_\pi^{-1} \preceq P_+$ where*

$$\begin{aligned} P_+ &= c^{-1}\left(\mathbf{I}_d + (1 - \mathrm{tr}\,(D_+))^{-1} D_+\right)\Delta_+ \\ P_- &= c\left(\mathbf{I}_d + (1 - \mathrm{tr}\,(D_-))^{-1} D_-\right)\Delta_- \end{aligned}$$

with $D_\pm = \Delta_\pm (x^* - \mu_\pi)(x^* - \mu_\pi)^T$ and $c := \sqrt{\det\Delta_- \det\Delta_+^{-1}} \leq 1$, and in addition

$$\left\|\nabla^2 U(x) - \Sigma_\pi^{-1}\right\|_2 \leq \max\left\{\|\Delta_+ - P_-\|_2, \|P_+ - \Delta_-\|_2\right\}$$

for all $x \in \mathbb{R}^d$.

For a proof see section 7.2.3.8. Proposition 42 allows us to localise the covariance in terms of the parameters of the target distribution.

One of the intuitions that can be gained from this section is that Hessians which exhibit small variations across the state space are preconditionable. In this scenario we would have $\Delta_+ \approx \Delta_-$ and hence that $c \approx 1$. Proposition 42 then suggests that so long as the distance between the mean and the mode is not too great, $\left\|\nabla^2 U(x) - \Sigma_\pi^{-1}\right\|_2$ is small. In summary, if π is preconditionable, and if the mean is close to the mode, then preconditioning with $L = \Sigma_\pi^{-\frac{1}{2}}$ is sensible.

Recall that a potential with additive Hessian satisfies $\nabla^2 U(x) = A + B(x)$ where $A, B(x) \in \mathbb{R}^{d\times d}$ are symmetric. In this case Δ_- and Δ_+ will be generated by variations in $B(x)$. Therefore, given proposition 42, a tighter localisation of $B(x)$ gives a tighter localisation of $\nabla^2 U(x)$ around the inverse covariance, leading to the following result.

Corollary 43. *Let π be a distribution with potential $U \in C^2$, covariance $\Sigma_\pi \in \mathbb{R}^{d\times d}$, mode $x^* \in \mathbb{R}^d$, and expectation $\mu_\pi \in \mathbb{R}^d$. If the Hessian of U is of the form $\nabla^2 U(x) = A + B(x)$ with $\|B(x)\|_2 \leq \varepsilon$ and $\varepsilon \mathbf{I}_d \prec A$ for some $\varepsilon > 0$, then if $1 - (x^* - \mu_\pi)^T (A + \varepsilon \mathbf{I}_d)(x^* - \mu_\pi) > 0$ it follows that*

$$\left\|\nabla^2 U(x) - \Sigma_\pi^{-1}\right\|_2 \leq \left(c^{-1} + 1\right)\varepsilon + \left(c^{-1} - 1\right)\|A\|_2 + \max\left\{\left\|\tilde{P}_-\right\|_2, \left\|\tilde{P}_+\right\|_2\right\}$$

where

$$\tilde{P}_+ = c^{-1}\left(1 - \mathrm{tr}\left(\tilde{D}_+\right)\right)^{-1}\tilde{D}_+(A + \varepsilon\mathbf{I}_d)$$
$$\tilde{P}_+ = c\left(1 - \mathrm{tr}\left(\tilde{D}_-\right)\right)^{-1}\tilde{D}_-(A - \varepsilon\mathbf{I}_d)$$

with $\tilde{D}_\pm = (A \pm \varepsilon\mathbf{I}_d)(x^* - \mu_\pi)(x^* - \mu_\pi)^T$ and $c := \sqrt{\det(A - \varepsilon\mathbf{I}_d)\det(A + \varepsilon\mathbf{I}_d)^{-1}} \leq 1$.

A proof can be found in 7.2.3.9. In 3.6.2 we look at the difference in performance between the preconditioners $L = A^{\frac{1}{2}}$, $L = \Sigma_\pi^{-\frac{1}{2}}$, and $L = \mathbf{I}_d$.

3.3 Linear preconditioning for multiplicative Hessians

A multiplicative Hessian has the form

$$\nabla^2 U(x) = X^T \Lambda(x) X \tag{3.3}$$

where $X \in \mathbb{R}^{n\times d}$ for $n \geq d$ is a matrix whose rows are usually the rows of some dataset and $\Lambda(x) \in \mathbb{R}^{n\times n}$. An example of a model with a multiplicative Hessian is as follows

$$\pi(\theta) \propto \exp\left(-\sum_{k=1}^{n} l_{y_k}\left(x_k^T\theta\right) - \frac{\lambda}{2}(\theta - \mu)^T X^T \Lambda X(\theta - \mu)\right) \tag{3.4}$$

where $\{(y_k, x_k)\}_{k=1}^n$ are observations with $y_k \in \mathbb{R}$ and $x_k \in \mathbb{R}^d$ for $k \in [n]$, X is a matrix with element in row i and column j equal to the jth element of x_i, and $\Lambda \in \mathbb{R}^{n\times n}$ is positive definite. Here l_{y_k} denotes some loss associated with observation k. In the case that l_{y_k} is a negative log-likelihood, equation 3.4 therefore describes the posterior associated with a typical generalised linear model using the generalised g-prior of

[Hanson, Branscum, and W. O. Johnson 2014].

Without further assumptions we have the following:

Proposition 44. *A distribution π whose potential U has a multiplicative Hessian satisfies*

$$\frac{\sup_{x\in\mathbb{R}^d}\lambda_{n-d+1}\left(\Lambda\left(x\right)\right)}{\inf_{x\in\mathbb{R}^d}\lambda_d\left(\Lambda\left(x\right)\right)\kappa\left(X^TX\right)}\leq\kappa\leq\kappa\left(X^TX\right)\frac{\sup_{x\in\mathbb{R}^d}\lambda_1\left(\Lambda\left(x\right)\right)}{\inf_{x\in\mathbb{R}^d}\lambda_d\left(\Lambda\left(x\right)\right)}$$

The proof in section 7.2.3.10 relies on an extension of Ostrowski's theorem to rectangular matrices [Higham and Cheng 1998, Theorem 3.2]. In many cases $\Lambda\left(x\right)$ will be diagonal and each eigenvalue $\lambda_i\left(\Lambda\left(x\right)\right)$ will range between the same possible values c and C (this is the case for binary logistic regression using the g-prior, for example). In this instance a more precise statement about κ can straightforwardly be made.

Assumption 45. *The Hessian of the potential of π is of multiplicative form with $\Lambda\left(x\right)$ diagonal, and there exists $c, C > 0$ such that $\sup_{x\in\mathbb{R}^d}\lambda_i\left(\Lambda\left(x\right)\right) = C$ and $\inf_{x\in\mathbb{R}^d}\lambda_i\left(\Lambda\left(x\right)\right) = c$ for all $i\in[n]$.*

Proposition 46. *A distribution π for which assumption 45 holds has a condition number*

$$\kappa=\frac{C}{c}\kappa\left(X^TX\right)$$

The choice $L=\left(X^TX\right)^{\frac{1}{2}}$ A natural choice of preconditioner here is $L=\left(X^TX\right)^{\frac{1}{2}}$. Indeed $L=\left(X^TX\right)^{\frac{1}{2}}$ is proportional to the preconditioner suggested by [Dalalyan 2017, Section 6.2]. For this we have the following.

Proposition 47. *Consider a distribution π whose potential has a multiplicative Hessian. Then preconditioning with $L=\left(X^TX\right)^{\frac{1}{2}}$ gives*

$$\frac{\sup_{x\in\mathbb{R}^d}\lambda_{n-d+1}\left(\Lambda\left(x\right)\right)}{\inf_{x\in\mathbb{R}^d}\lambda_d\left(\Lambda\left(x\right)\right)}\leq\tilde{\kappa}\leq\frac{\sup_{x\in\mathbb{R}^d}\lambda_1\left(\Lambda\left(x\right)\right)}{\inf_{x\in\mathbb{R}^d}\lambda_d\left(\Lambda\left(x\right)\right)}$$

For a proof see section 7.2.3.11. Under assumption 45 the upper and lower bounds of proposition 47 are equal giving:

Corollary 48. *Consider a distribution π for which assumption 45 holds. Choosing $L=\left(X^TX\right)^{\frac{1}{2}}$ gives $\tilde{\kappa}=C/c$.*

Hence the condition number is reduced under preconditioning with $L = \left(X^T X\right)^{\frac{1}{2}}$. The level of reduction will be determined by $\kappa\left(X^T X\right)$, which characterises how far from orthogonal $X^T X$ is. In the context of regression or classification problems $\kappa\left(X^T X\right)$ will be smallest under an orthogonal design and larger when there is more collinearity among features of when different features have different variances.

The QR decomposition A popular strategy for regression and classification models in which X is a design matrix is to perform a (reduced) QR decomposition, setting $X = QR$ where $Q \in \mathbb{R}^{n\times d}$ is orthogonal and $R \in \mathbb{R}^{d\times d}$ is upper triangular[2]. In this case the preconditioner is $L = R$, from which is follows by setting $X = Q$ in proposition 44 that

$$\tilde{\kappa} \leq \frac{\sup_{x\in\mathbb{R}^d} \lambda_1\left(\Lambda\left(x\right)\right)}{\inf_{x\in\mathbb{R}^d} \lambda_d\left(\Lambda\left(x\right)\right)}$$

The QR strategy therefore gives the same upper bound as the choice $\left(X^T X\right)^{\frac{1}{2}}$ of the previous section.

The Hessian at the mode Another natural choice of preconditioner is $L = \nabla^2 U\left(x^*\right) = \left(X^T \Lambda\left(x\right) X\right)^{\frac{1}{2}}$ for some $x^* \in \mathbb{R}^d$. For instance x^* could be the mode of π. Here we have the following

Proposition 49. *Consider a distribution π whose potential has a multiplicative Hessian. Then choosing $L = \left(X^T \Lambda\left(x^*\right) X\right)^{\frac{1}{2}}$ gives*

$$\tilde{\kappa} \leq \frac{\sup_{x\in\mathbb{R}^d} \lambda_1\left(\Lambda\left(x^*\right)^{-\frac{1}{2}}\Lambda\left(x\right)\Lambda\left(x^*\right)^{-\frac{1}{2}}\right)}{\inf_{x\in\mathbb{R}^d} \lambda_d\left(\Lambda\left(x^*\right)^{-\frac{1}{2}}\Lambda\left(x\right)\Lambda\left(x^*\right)^{-\frac{1}{2}}\right)} \leq \left(\frac{\sup_{x\in\mathbb{R}^d} \lambda_1\left(\Lambda\left(x\right)\right)}{\inf_{x\in\mathbb{R}^d} \lambda_d\left(\Lambda\left(x\right)\right)}\right)^2$$

A proof of the proposition can be found in 7.2.3.12. Note that the first upper bound is simply the condition number of $\tilde{\pi}$, a measure whose potential has Hessian $\Lambda\left(x\right)$, after preconditioning with $L = \Lambda\left(x^*\right)^{\frac{1}{2}}$. Therefore if $\tilde{\pi}$ is preconditionable with $L = \Lambda\left(x^*\right)^{\frac{1}{2}}$ then π is preconditionable with $L = \left(X^T \Lambda\left(x^*\right) X\right)^{\frac{1}{2}}$. Therefore the strategy of using $L = \left(X^T \Lambda\left(x^*\right) X\right)^{\frac{1}{2}}$ may be preferable to the previous two options in the setting where $\Lambda\left(x\right)$ still has some variation between eigenvalues, but when it does not change much between different values of x, meaning it is an almost constant matrix. In that case $\Lambda\left(x^*\right)^{-\frac{1}{2}}\Lambda\left(x\right)\Lambda\left(x^*\right)^{-\frac{1}{2}}$ should be close to the identity, and so the condition number after preconditioning with the Hessian at the mode will be ≈ 1,

[2] This can be found e.g. in https://mc-stan.org/docs/stan-users-guide/regression.html under the section titled ‘The QR Reparameterization’.

whereas $\sup_{x\in\mathbb{R}^d} \lambda_1 (\Lambda(x)) / \inf_{x\in\mathbb{R}^d} \lambda_d (\Lambda(x))$ might still be much larger than 1. We compare the Hessian at the mode to the choice $L = \left(X^T X\right)^{\frac{1}{2}}$ empirically in 3.6.3.

3.4 Tight condition number dependence of the spectral gap of RWM

[Andrieu, A. Lee, et al. 2024] give upper and lower bounds on the spectral gap of the RWM that are tight in the dimension, but only the lower bound is explicit in its dependence on the condition number. Here we show that if some additional conditions are imposed on U akin to an additive Hessian structure then the dependence on κ can also be made explicit in the upper bound.

Assumption 50. *The potential of π satisfies assumption 24 and there is an $\varepsilon > 0$ such that $\left\|\nabla^2 U(x) - \nabla^2 U(y)\right\|_2 \leq m\varepsilon$ for all $x, y \in \mathbb{R}^d$.*

Remark 51. Assumption 50 will hold when $\nabla^2 U(x) = A + B(x)$ with $\|B(x)\|_2$ suitably small, since $\sup_x \|B(x)\|_x < m\varepsilon/2 \implies \left\|\nabla^2 U(x) - \nabla^2 U(y)\right\|_2 \leq 2 \sup_x \|B(x)\|_x \leq m\varepsilon$ as required. Assumption 50 is therefore simply one way to formalize the notion of 'additive' Hessian structure.

Theorem 52. *Let π have potential $U \in C^2$ satisfying assumption 24 with condition number $\kappa = M/m \geq 1$. If π also satisfies assumption 50 then the spectral gap γ_κ of RWM with proposal variance $\sigma^2 \mathbf{I}_d$ such that $\sigma^2 := \xi / (Md)$ for any $\xi > 0$ satisfies*

$$C\xi \exp(-2\xi) \frac{1}{\kappa}\frac{1}{d} \leq \gamma_\kappa \leq (1 + 2\varepsilon) \frac{\xi}{2}\frac{1}{\kappa}\frac{1}{d}$$

where $C = 1.92 \times 10^{-4}$.

For a proof see 7.2.3.13. Both bounds presented above are $O\left(\kappa^{-1}\right)$, which implies that the relaxation time $1/\gamma_\kappa$ is precisely linear in κ. The lower bound is as originally presented by [Andrieu, A. Lee, et al. 2024, Theorem 1], and the choice $\sigma^2 = \xi/(Md)$ is as recommended in that work to ensure tight $O\left(d^{-1}\right)$ dependence. As the authors remark, the constant C can be made a few orders of magnitude larger.

Theorem 52 can be combined with the condition number results of sections 3.2 and 3.3 to guarantee under the assumptions stated there that the spectral gap increases under appropriate linear preconditioning, as shown in the corollary below

Corollary 53. *Let π has potential $U \in C^2$ satisfying assumption 24 with condition number $\kappa = M/m \geq 1$. Assume that π also satisfies 50 with constant $\varepsilon' > 0$. Using a preconditioner $L \in \mathbb{R}^{d\times d}$ satisfying assumption 33 with constant $\varepsilon > 0$ ensures that the spectral gap γ_κ of RWM with proposal variance $\sigma^2 \mathbf{I}_d$ such that $\sigma^2 := \xi / (Md)$ for any $\xi > 0$ increases under preconditioning whenever*

$$\kappa \geq \frac{1}{2} C^{-1} \exp(2\xi)(1+2\varepsilon')(1+\varepsilon)\left(1+\frac{\sigma_1(L)^2}{m}\varepsilon\right)$$

where $C = 1.972 \times 10^{-4}$, and $\sigma_1(L)$ is the greatest singular value of L.

Corollary 53 uses Theorem 52 along with Theorem 38, which provides an upper bound on the condition number after preconditioning in the additive Hessian setting. This bound leads to a lower bound on the relaxation time, which can be compared with the upper bound from Theorem 52 before preconditioning. This comparison establishes that preconditioning ensures an improvement in relaxation time, provided the initial condition number κ is sufficiently large. A proof can be found in section 7.2.3.14. Note that a similar result could be stated by applying the bounds of Theorem 30 or Theorem 35 in the place of Theorem 38, which would then modify the necessary lower bound on κ in the above.

3.5 Counterproductive diagonal preconditioning

The appeal of diagonal preconditioning is motivated by the promise of reducing the condition number in $O(d)$ computational cost. Therefore choosing $L = \text{diag}\left(\hat{\Sigma}_\pi\right)^{-\frac{1}{2}}$ for some estimate $\hat{\Sigma}_\pi$ of the target covariance has become a common practice since it is viewed as computationally cheap, and it is assumed that it will offer an improvement on no preconditioning at all. The developers of Stan [Carpenter et al. 2017] for instance, offer the option of diagonal preconditioning with the target covariance as using a diagonal mass matrix. Such an option is also offered in the TensorFlow Probability library [Abadi et al. 2016].

In fact, we show here that there are examples of distributions for which diagonal preconditioning in this manner can actually increase the condition number, and therefore be explicitly *worse* than performing no preconditioning at all. This phenomenon can be observed even when the target is Gaussian. Noting that

$\tilde{\kappa} = \left\| \text{diag}\left(\hat{\Sigma}_\pi\right)^{\frac{1}{2}} \Sigma_\pi^{-1} \text{diag}\left(\hat{\Sigma}_\pi\right)^{\frac{1}{2}} \right\|_2 \left\| \text{diag}\left(\hat{\Sigma}_\pi\right)^{\frac{1}{2}} \Sigma_\pi^{-1} \text{diag}\left(\hat{\Sigma}_\pi\right)^{\frac{1}{2}} \right\|_2 = \left\| C_\pi^{-1} \right\|_2 \left\| C_\pi \right\|_2$ where $C_\pi \in \mathbb{R}^{d \times d}$ is the correlation matrix associated with Σ_π, it suffices to find a Σ_π for which $\tilde{\kappa} = \kappa\left(C_\pi\right) > \kappa\left(\Sigma_\pi\right) = \kappa$. The matrix below is such an example.

$$\Sigma_\pi = \begin{pmatrix} 21.5 & 5.7 & 18.7 & 4.5 & 6.9 \\ \star & 2.0 & 4.9 & 1.2 & 2.1 \\ \star & \star & 16.3 & 3.9 & 5.7 \\ \star & \star & \star & 1.4 & 1.4 \\ \star & \star & \star & \star & 2.9 \end{pmatrix} \implies \kappa \approx 4.4 \times 10^3, \tilde{\kappa} \approx 8.1 \times 10^3 \tag{3.5}$$

In the above we have truncated the entries to a single decimal place: see section 7.2.3.15 for the full matrix. The condition number increases by a substantial amount, even though we have perfect knowledge of the target covariance. See section 3.6.1 for an empirical analysis of the random walk Metropolis on a $\mathcal{N}\left(0, \Sigma_\pi\right)$ target (with Σ_π as above) after diagonal and dense preconditioning.

In a practical scenario we would have to expend computational effort to construct diag $\left(\hat{\Sigma}_\pi\right)^{-\frac{1}{2}}$ and so for a target such as the one described here, this effort would be wasted as it actually reduces sample quality. In general we can conjecture that targets with covariance matrices whose associated correlations are far from being diagonally dominant will be least amenable to diagonal preconditioning.

3.6 Preconditioning experiments

3.6.1 Counterproductive diagonal preconditioning

This experiment illustrates the phenomenon described in section 3.5: namely that there exist Gaussian targets in the form $\mathcal{N}(0, \Sigma_\pi)$ such that preconditioning with $L = \text{diag}(\Sigma_\pi)^{-1/2}$ *increases* the condition number.

We compare the performance of three RWM algorithms: one with no preconditioning, one with dense preconditioning ($L = \Sigma_\pi^{-1/2}$), and one with diagonal preconditioning ($L = \text{diag}(\Sigma_\pi)^{-1/2}$). Each chain targets $\mathcal{N}(0, \Sigma_\pi)$ with Σ_π as in 3.5 and is initialised at equilibrium. Proposals take the standard RWM form $X' = X + \sigma L^{-1} \xi$ with $\xi \sim \mathcal{N}(0, \mathbf{I}_5)$ and $\sigma = 2.38/\sqrt{d}$ as recommended by [Gareth O. Roberts and Jeffrey S.

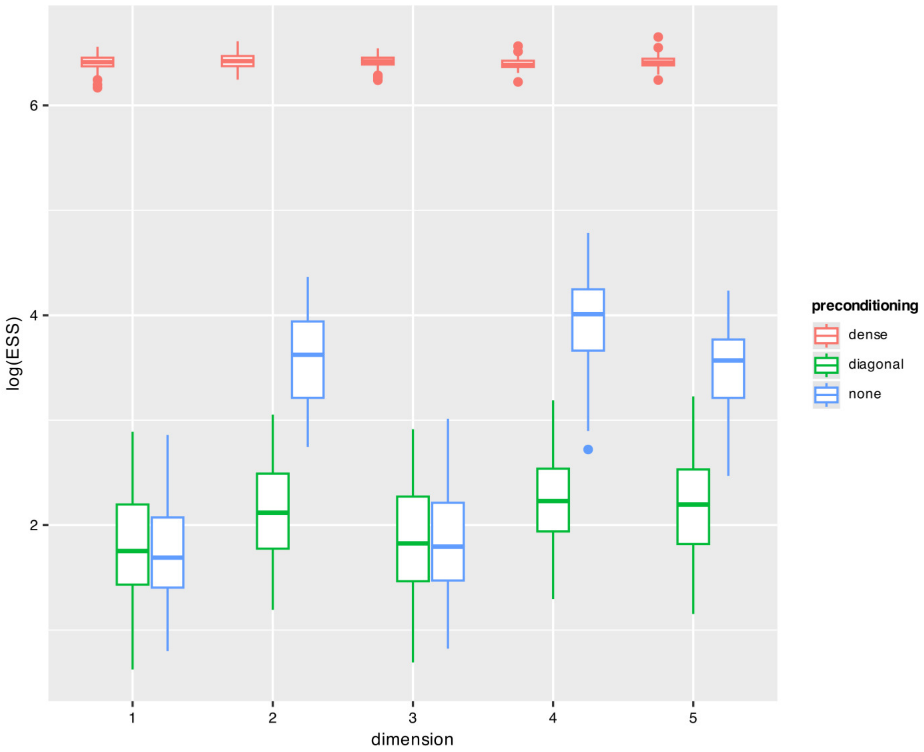

Figure 3.2: $\log$ ESS of 100 runs at 10,000 iterations per run of RWM with dense, diagonal, and zero preconditioning. Each algorithm is started in equilibrium and targets $\mathcal{N}(0, \Sigma_\pi)$ with Σ_π as in 3.5.

Rosenthal 2001]. We run each chain from each algorithm 100 times at 10,000 iterations per chain. For each chain we compute the ESS (see 21 for a definition of this quantity) in each dimension using the `effectiveSize` function from the `coda` package [Plummer et al. 2024].

As can be seen in Figure 3.2 the ESS's of the RWM chain with no preconditioning are clearly larger in dimensions 2, 4, and 5 from the diagonally preconditioned chain. This is despite the diagonal preconditioner being formed with perfect knowledge of the target covariance. As is expected the dense preconditioner performs the best since the target effectively becomes a standard Gaussian.

3.6.2 Preconditioning the additive Hessian

One probabilistic model with additive Hessian structure is a Bayesian regression with hyperbolic prior. It is well known that the Laplace prior $\beta \mapsto 2^{-1}\lambda \exp(-\lambda\|\beta\|_1)$ for $\beta \in \mathbb{R}^d$ and $\lambda > 0$ imposes the same sparsity in the maximum a posteriori estimates as would regularisation with the LASSO [Tibshirani 1996] since it concentrates sharply around $\beta = 0$. More generally, priors with exponential tails can be motivated by results concerning the contraction of regression parameter posteriors around their true values. As demonstrated in the discussion following Theorem 8 in [Castillo, Schmidt-Hieber, and Vaart 2015], heavy-tailed priors such as the Laplace distribution achieve good rates of contraction. Simply using a Laplace prior would violate the M-smoothness assumption due to the behaviour at $\beta = 0$. The hyperbolic prior, however, is a smooth distribution with exponential tails, and so we will use this as a prior for the regression parameters $\beta \in \mathbb{R}^d$. We assume that $Y = X\beta + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$ for $\sigma^2 > 0$ known and $n > d$. We also assume that the columns of $X \in \mathbb{R}^{n\times d}$ are standardised to have variance 1. Because of this it is reasonable to use the same scale in each dimension of the prior. The resulting posterior has a potential of the form

$$U(\beta) = \frac{1}{2\sigma^2}\|Y - X\beta\|^2 + \lambda \sum_{i=1}^{d} \sqrt{1+\beta_i^2}, \tag{3.6}$$

which implies

$$\nabla^2 U(\beta) = \frac{1}{\sigma^2} X^T X + \lambda D(\beta), \tag{3.7}$$

where $D(\beta) = \text{diag}\{(1+\beta_i^2)^{-3/2} : i \in [d]\}$. The Hessian is therefore additive with $A = \sigma^{-2}X^T X$ and $B(\beta) = \lambda D(\beta)$.

Equations (3.6)-(3.7) show that the posterior is a well-conditioned distribution, meaning that U satisfies Assumption 24 with $m = \sigma^{-2}\sigma_d(X^T X)$ and $M = \sigma^{-2}\|X^T X\| + \lambda$. Preconditioning with $L = (\sigma^{-2}X^T X)^{1/2}$ gives $\tilde{\kappa} = 1 + \sigma_d(X^T X)^{-1}\lambda\sigma^2 \leq \kappa$. In this case the distance between LL^T and the Hessian can be bounded using $\|\nabla^2 U(\beta) - LL^T\| \leq \lambda$, so we can also apply the results of Section 3.2.2 to justify the use of the target covariance by setting $\epsilon = \lambda$ in Corollary 43. These results imply that when λ is small then preconditioning with either $L = (\sigma^{-2}X^T X)^{1/2}$ or $L = \Sigma_\pi^{-1/2}$ should improve the efficiency of the sampler, so long as the distance between the mean and the mode is not too large.

In the following experiment we run MALA chains on target distributions with $L = (\sigma^{-2}X^TX)^{1/2}$, $L = \Sigma_\pi^{-1/2}$, and $L = \mathbf{I}_d$. We set $d \in \{2, 5, 10, 20, 100\}$ and $n = \{1, 5, 20\} \times d$ for each value of d. At each combination of n and d we run 15 chains for each preconditioner. Each chain is composed by initialising at $\beta = (X^TX)^{-1}X^TY$ and taking 10^4 samples to equilibrate. We initialise the step size at $d^{-1/6}$ and adapt it along the course of the chain seeking an optimal acceptance rate of 0.574 according to the results of [Gareth O. Roberts and Jeffrey S. Rosenthal 2001]. We then continue the chain with preconditioning and a fixed step size of $d^{-1/6}$ for a further 10^4 samples, over which we measure the ESS of each dimension. To construct $L = \Sigma_\pi^{-1/2}$ we simply use the empirical covariance of the first 10^4 samples.

For the model parameters we set $\lambda = \sqrt{n}/d$ using the lower bound of [Castillo, Schmidt-Hieber, and Vaart 2015]. Every element of X is an independent standard normal random variable, and Y is generated by sampling β_0 from the prior and setting $Y = X\beta_0 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \mathbf{I}_d)$, meaning $\sigma = 1$.

The boxplots in Figure 3.6.2 show the median ESSs for each run. The figure demonstrates that in the $n/d \in \{5, 20\}$ cases preconditioning with $L = \Sigma_\pi^{-1/2}$ is just as good as preconditioning with $L = A^{1/2}$ where $A = \sigma^{-1}X^TX$. For $n/d = 1$ the results are mixed: for instance in the $(d, n) = (100, 100)$ configuration the first 10^4 iterations of the MALA chain mixed poorly, offering a poor estimate of Σ_π. The performance suffered heavily if no preconditioner was applied.

3.6.3 Preconditioning the multiplicative Hessian

To study preconditioning under the multiplicative Hessian structure we consider a Bayesian binomial regression with a generalised g-prior [Sabanés Bové and Held 2010, Section 2.1][Held and Sauter 2017, Section 2.2]. The generalised g-prior is an extension of the classical g-prior to generalised linear models that have dispersion parameters of the form $\phi_i := \phi w_i^{-1}$ for $i \in [n]$ and known weights $w_i \in \mathbb{R}^+$. It is motivated by constructing an 'imaginary sample' of responses $y_0 = h(0)\mathbb{1}_n \in \mathbb{R}^n$ from a generalised linear model with inverse link function $h(.)$ and design matrix $X \in \mathbb{R}^{n\times d}$. Assigning the parameter vector $\beta \in \mathbb{R}^d$ a flat prior, it is observed that as $n \to \infty$ the posterior distribution of β in this construction tends to $\mathcal{N}_d(0, g\phi c(X^TWX)^{-1})$ where $W = \text{diag}\{w_i : i \in [n]\}$, g and ϕ are hyperparameters, and c is a model-specific constant. See [Sabanés Bové and Held 2010, Section 2.1] for a more thorough exposition.

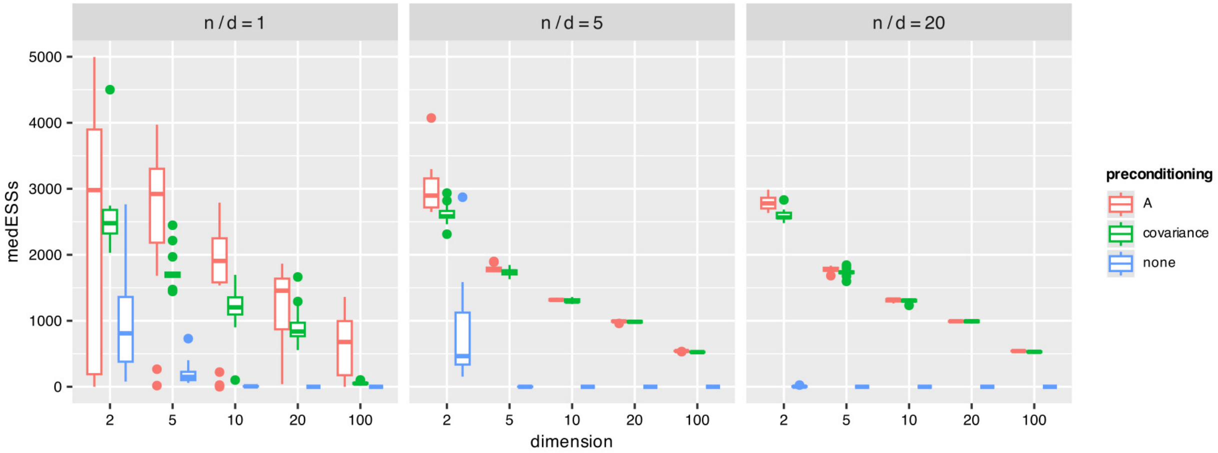

Figure 3.3: Boxplots of the medians of the ESSs across configurations of (n, d) with different preconditioners on the Bayesian linear regression with a Hyperbolic prior. The leftmost boxplot in each grouping corresponds to preconditioning with $L = \sigma(X^T X)^{1/2}$ ('A' in the legend), the middle boxplot has $L = \Sigma_\pi^{-1/2}$ ('covariance' in the legend), the rightmost has $L = \mathbf{I}_d$ ('none' in the legend).

We follow the advice given by [Sabanés Bové and Held 2010, Section 2.1] and [Held and Sauter 2017, Section 2.2] by setting $w_i = m_i$ for all $i \in [n]$. Using a logistic link gives a posterior with potential

$$U(\beta) = \sum_{i=1}^{n} \left(w_i \left((1 - Y_i) X_i^T \beta + \log(1 + \exp(-X_i^T \beta)) \right) \right) + (g\phi c)^{-1} \beta^T X^T W X \beta \tag{3.8}$$

with Hessian $\nabla^2 U(\beta) = X^T \Lambda(\beta) X$, where

$$\Lambda(\beta) := W \text{diag}\{\exp(X_i^T \beta)(1 + \exp(X_i^T \beta))^{-2} + (g\phi c)^{-1} : i \in [n]\}$$

The potential U therefore satisfies Assumption 24 with $M = (0.25 + (g\phi c)^{-1}) w_{\text{max}} \|X^T X\|$ and $m = (g\phi c)^{-1} w_{\text{min}} \sigma_d(X^T X)$, where $w_{\text{max}} := \max_i w_i$ and $w_{\text{min}} := \min_i w_i$. We choose g and ϕ such that $(g\phi c)^{-1} = \lambda n^{-1}$, where $\lambda = 0.01$.

We examine the effectiveness of preconditioning with $L \in \{\Sigma_\pi^{-1/2}, \mathcal{I}^{1/2}, \nabla^2 U(\beta^*)^{1/2}, \mathbf{I}_d, (n^{-1} X^T X)^{1/2}\}$ where Σ_π is the covariance of the posterior, $\mathcal{I}$ is the 'Fisher matrix' of [Michalis Titsias 2023], β^* is the mode, and $L = (n^{-1} X^T X)^{1/2}$ is the preconditioner used in [Dalalyan 2017, Section 6.2]. When $L \in$

$\{\mathbf{I}_d, (n^{-1}X^TX)^{1/2}, \nabla^2 U(\beta^*)^{1/2}\}$ the condition numbers are given by

$$\begin{aligned} L = \mathbf{I}_d &\Rightarrow \tilde{\kappa} = \kappa = \frac{\frac{n}{4}+\lambda}{\lambda}\frac{w_{\textsf{max}}}{w_{\textsf{min}}}\kappa(X^TX) \\ L = (n^{-1}X^TX)^{\frac{1}{2}} &\Rightarrow \tilde{\kappa} = \frac{\frac{n}{4}+\lambda}{\lambda}\frac{w_{\textsf{max}}}{w_{\textsf{min}}} \\ L = \nabla^2 U(\beta^*)^{\frac{1}{2}} &\Rightarrow \tilde{\kappa} = \frac{\frac{n}{4}+\lambda}{\lambda}\frac{n\max_i p_i^*(1-p_i^*)+\lambda}{n\min_i p_i^*(1-p_i^*)+\lambda} \end{aligned}$$

where $p_i^* := (1+\exp(-X_i^T\beta^*))^{-1}$. This suggests that $L = \nabla^2 U(\beta^*)^{1/2}$ offers an increase in efficiency over $L = (n^{-1}X^TX)^{1/2}$ for $w_{\textsf{max}}/w_{\textsf{min}}$ large.

3.6.3.1 Experimental setup and results

We run RWM chains with the preconditioners described above for $d \in \{2, 5, 10, 20\}$ and $n = 5d$. We generate the design matrix $X \in \mathbb{R}^{n\times d}$ with $X = G + M$ where $G_{ij} \sim \mathcal{N}(0,1)$ independently and $M_{ij} = \mu$ for all $i \in [n], j \in [d]$. We set $\mu \in \{0, 5, 50, 200\}$ to arbitrarily worsen the conditioning of the model, as it can be shown that

$$\kappa(X^TX) \geq \frac{\sum_{k=1}^n (G_{k1}+\mu)^2}{\frac{1}{2}\sum_{k=1}^n (G_{k1}-G_{k2})^2}.$$

We set $w_i = i^2$ for $i \in [n]$ and generate the responses using $Y_i = S_i/w_i$ with $S_i \sim \textsf{Bin}(w_i, (1+\exp(-X_i^T\beta_0))^{-1})$ for $\beta_0 \sim \mathcal{N}(0, \mathbf{I}_d)$. We use gradient descent on U which we precondition with $L = (n^{-1}X^TX)^{1/2}$ to find the mode β^*.

We approximate Σ_π and $\mathcal{I}$ in two different ways. We either construct them using ergodic averages generated by unpreconditioned RWM for 10^4 iterations, or we run an $L = \nabla^2 U(\beta^*)^{1/2}$ preconditioned RWM for 10^5 iterations, from which we calculate the same ergodic averages. At each combination of d and μ we run 15 chains for each preconditioner. Each chain is composed by initialising at $\beta \sim \mathcal{N}(0, (n^{-1}X^TX)^{-1})$ and taking 10^4 samples to equilibrate. In each of these initial chains we initialise the step size at $2.38/d^{1/2}$ and adapt it along the course of the trajectory seeking an optimal acceptance rate of 0.234 according to the results of [Gareth O. Roberts and Jeffrey S. Rosenthal 2001]. We then continue the chain with preconditioning and a fixed step size of $2.38/d^{1/2}$ for a further 10^4 samples, over which we measure the ESS of each dimension. The median ESSs in the $\mu \in \{0, 200\}$ cases are plotted in Figure 3.4.

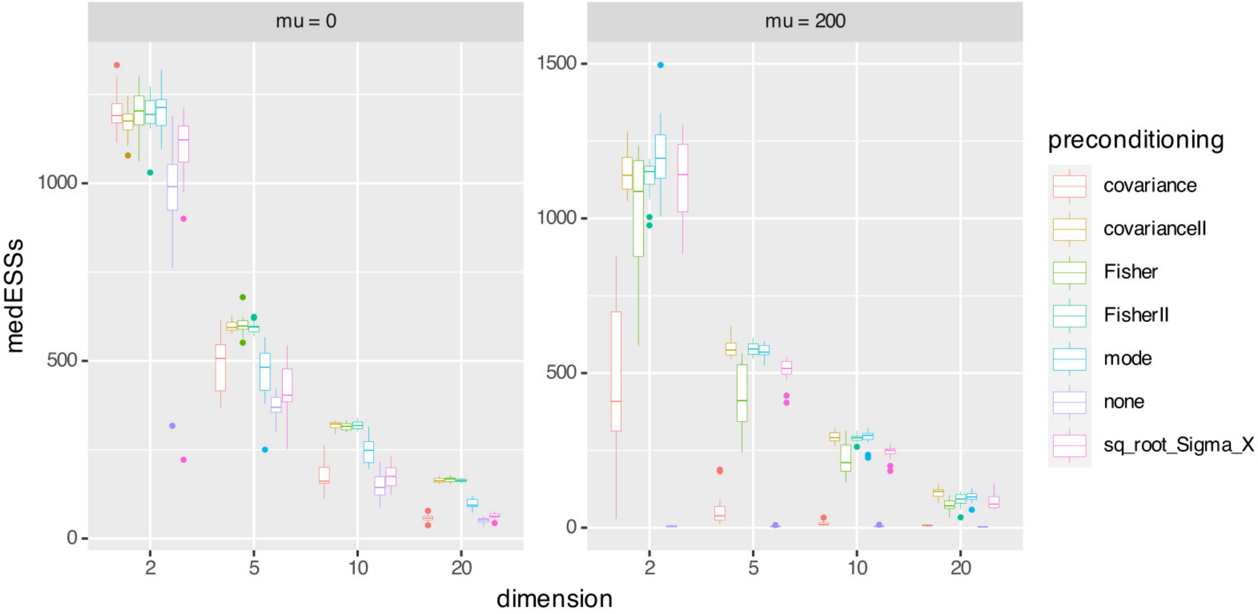

Figure 3.4: Boxplots of the logarithms of the medians of the ESSs across combinations of (d, μ). The ESSs are taken from RWM runs on a binomial regression target with the generalised g-prior. 'covariance' and 'covarianceII' correspond to runs preconditioned with $L = \Sigma_\pi^{-1/2}$ where Σ_π is estimated over 10^4 and 10^5 runs respectively. 'Fisher' and 'FisherII' correspond to runs preconditioned with $L = \mathbb{E}_\pi[\nabla^2 U(\beta)]^{1/2}$ where $\mathbb{E}_\pi[\nabla^2 U(\beta)]$ is estimated over 10^4 and 10^5 runs respectively. 'mode' refers to runs preconditioned with $L = \nabla^2 U(\beta^*)^{1/2}$ where β^* is an estimate of the mode found using preconditioned gradient descent. 'sq_root_Sigma_X' corresponds to runs preconditioned with $L = (n^{-1} X^T X)^{1/2}$.

The preconditioning strategies are detailed in the figure caption. ‘covariance’ and ‘covarianceII’ refer to the runs preconditioned with $L = \Sigma_\pi^{-1/2}$ where the covariance is estimated over 10^4 and 10^5 samples respectively. The same is the case for ‘Fisher’ and ‘FisherII’. ‘sq_root_Sigma_X’ refers to the runs made with $L = (n^{-1}X^TX)^{1/2}$. ‘mode’ refers to preconditioning with $L = \nabla^2 U(\beta^*)^{1/2}$.

As predicted, preconditioning with the Hessian at the mode does offer a benefit over preconditioning with $L = (n^{-1}X^TX)^{1/2}$. Preconditioning with the covariance when it is estimated over a larger, better quality sample (‘covarianceII’) is one of the best performing strategies, whereas preconditioning with the covariance estimated over the smaller sample (‘covariance’) suffers with dimension and ill-conditioning of the model. This is clearly due to the reduction in quality of the covariance estimate. This disparity in performance is contrasted with the difference between the ‘Fisher’ and ‘FisherII’ cases, which is very slight.

3.7 Summary and Extensions

3.7.1 Summary

In all this chapter should serve as an introduction to, and analysis of, linear preconditioning in the context of sampling. In section 2.1.2.1 we give assumptions on the target distribution that are taken to hold throughout the subsequent sections. These are the m-strong convexity and M-smoothness of the potential of the target. In section 2.1.2.2 we outline statistical features of distributions that satisfy these assumptions. We also give practical algorithmic implications that arise as a result of them. These assumptions are important for our purposes because they ensure the existence and finiteness of the condition number $\kappa = M/m$. In section 2.1.2.3 and table 2.1 we present a selection of bounds on relaxation and ε-mixing time that all depend on the condition number. This motivates our study of the way in which preconditioning affects it.

Section 2.1.2.4 contains the main original contributions of the chapter. We define linear preconditioning as a linear pushforward of the target distribution. In section 2.1.2.5 we justify this definition using proposition 26 which states that our definition and the ‘traditional’ definition are the same in that they produce isomorphic Markov chains, which share important convergence and stability properties. In section 3.1 proposition 27 we prove the existence of distributions that can only be made harder to sample from upon linear preconditioning.

This lends intuition as to the kind of assumptions, additional to assumption 24, that must be made to ensure effective preconditioning.

In section 3.2 we provide sets of assumptions, each providing the conditions under which we can control the condition number after preconditioning. These assumptions are most easily verified for distributions whose potentials have additive Hessians, which we define here 3.2. Assumptions 28 and 29 explicitly control the eigenstructure of the Hessian, and imply Theorem 30 which provides control on the the condition number after preconditioning in terms of the variations of the eigenvalues and vectors of the Hessian. Remark 32 provides the intuition that the alignment between the linear preconditioner and the target is crucial, in the case that the target is ill-conditioned. Assumptions 33 and 34 localise the Hessian about LL^T where $L \in \mathbb{R}^{d \times d}$ is the preconditioner, and together they lead to Theorem 35 which also controls the condition number after preconditioning. They provide a relaxation of 28 and 29 in that they do not require explicit knowledge of the spectral information of the Hessian, and that assumption 33 implies assumption 28. We further relax matters by dropping assumption 35 to give Theorem 38. We then look at two preconditioners: $L = \Sigma_\pi^{-\frac{1}{2}}$ (used in Stan, for instance) and $L = \mathbb{E}_\pi \left[\nabla U\left(X\right) \nabla U\left(X\right)^T\right]^{\frac{1}{2}}$ (suggested in [Michalis Titsias 2023]), and examine the ways in which they allow us to satisfy the assumptions referred to above. In section 3.6.2 we test the preconditioners $L = \hat{\Sigma}_\pi^{-\frac{1}{2}}$ and $L = A$ where $A \in \mathbb{R}^{d \times d}$ with the MALA on a Bayesian regression with hyperbolic prior [Castillo, Schmidt-Hieber, and Vaart 2015] whose potential has a Hessian in the additive form $\nabla^2 U\left(x\right) = A + B\left(x\right)$ 3.2. The preconditioners fare similarly well, apart from when the target covariance Σ_π is difficult to estimate such that the preconditioner $L = \hat{\Sigma}_\pi^{-\frac{1}{2}}$ performs poorly.

Section 3.3 examines linear preconditioning in the case that the target distribution has a potential whose Hessian is multiplicative such that $\nabla^2 U\left(x\right) = X^T \Lambda\left(x\right) X$ where $X \in \mathbb{R}^{n \times d}$ and $\Lambda\left(x\right) \in \mathbb{R}^{n \times n}$. This particular structure of Hessian is found in the posterior distributions of the parameters of GLMs which have generalised g-priors [Held and Sauter 2017; Sabanés Bové and Held 2010]. The form of the Hessian suggest natural preconditioners, such as $L = \left(X^T X\right)^{\frac{1}{2}}$ [Dalalyan 2017] and $L = \nabla^2 U\left(x^*\right)^{\frac{1}{2}}$. We provide bounds on the condition number after preconditioning with these in propositions 47 and 49. In section 3.6.3 we compare the performance these two preconditioners, along with $L = \hat{\Sigma}_\pi^{-\frac{1}{2}}$ and $L = \mathbb{E}_\pi \left[\nabla U\left(X\right) \nabla U\left(X\right)^T\right]^{\frac{1}{2}}$, with RWM on a Bayesian binomial regression with generalised g-prior.

One of the places the condition number features is in bounds on the spectral gap (see table 2.1). In the case of RWM, [Andrieu, A. Lee, et al. 2024, Theorem 1] lacks explicitness in the condition number in the upper bound that they posit. In section 3.4 we give use assumption 50 to get an upper and lower bound on the spectral gap of RWM in Theorem 52 that match in terms of their dependence on the condition number. We then use Theorem 38 to state conditions under which this spectral gap increases upon preconditioning.

Practitioners often use diagonal preconditioners to save on compute time. In section 3.5 we provide an example of a Gaussian distribution where preconditioning with the well-used preconditioner $L = \text{diag}\left(\Sigma_\pi\right)^{-\frac{1}{2}}$ can cause the condition number to increase. This fact is demonstrated numerically in section 3.5 where using the $L = \text{diag}\left(\Sigma_\pi\right)^{-\frac{1}{2}}$ preconditioner decreases the ESS compared with using $L = \mathbf{I}_d$.

3.7.2 Extensions

3.7.2.1 The probabilistic perspective

In general the Hessian $\nabla^2 U\left(x\right)$ of the potential of π may be a random variable. For instance, in a Bayesian regression setting the Hessian will be additive 3.2 with $\nabla^2 U\left(x\right) = kX^T X + \Lambda\left(x\right)$ where $k > 0$, $X \in \mathbb{R}^{n \times d}$ is the design matrix and $\Lambda\left(x\right) \in \mathbb{R}^{d \times d}$ is the Hessian of the potential of the prior distribution. An example of this can be seen in section 3.6.2. Equally, in the setting of a Bayesian GLM with a g-prior the Hessian will be multiplicative 3.3 with $X \in \mathbb{R}^{n \times d}$ as the design matrix again, as we see in section 3.6.3. Similarly the preconditioner L may be a random variable: the Fisher matrix 3.2.1 and the target covariance 3.2.2 must be estimated, and the preconditioner for the QR decomposition 3.3 and Hessian at the mode 3.3 cases depend on the design matrix.

This allows us to use the machinery we developed to prove the results in sections 3.2 and 3.3 in a probabilistic setting. For instance, we may be able to say that $\left\|\nabla^2 U\left(x\right) - LL^T\right\|_2 \leq \sigma_d^2 \varepsilon_{\text{prob}}$ holds with high probability (with respect to, say, n and d). We can then use the proof techniques to make statements about say the concentration about a given location of the condition number after preconditioning with L, which is now also a random variable. This engenders a change in perspective to the one explored in the material presented above since we are now examining the behaviour of the typical preconditioned MCMC algorithm whereas in the above material we are looking at a specific instance of preconditioning. Both of these perspectives are useful at different points in the scientific process.

3.7.2.2 Alternative condition numbers and refinements

Alternative, problem specific condition numbers have been defined by various parties. For instance where Σ_π is the covariance of π with spectrum $\sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_d^2$ [Langmore et al. 2020] suggest using

$$\left(\sum_{i=1}^{d}\left(\frac{\sigma_1}{\sigma_i}\right)^4\right)^{\frac{1}{4}}$$

as a condition number. Under specifications on the step size of the algorithm, such a quantity is shown to be proportional to the number of leapfrog steps needed to achieve a stable acceptance rate in HMC.

The condition number as defined in 2.4 encodes the difficulty of sampling from π, but it does not capture additional information we might have about π which might ameliorate the sampling efficiency. For instance, if we knew that there existed positive definite $A_-, A_+ \in \mathbb{R}^{d\times d}$ such that

$$A_- \preceq \nabla^2 U(x) \preceq A_+$$

then we could precondition with $L = A_-^{\frac{1}{2}}$ achieving $\tilde{\kappa} = \lambda_1\left(A_-^{-1}A_+\right)$ over $\kappa = \lambda_1(A_+)/\lambda_d(A_-)$. Therefore defining the condition number as $\lambda_1\left(A_-^{-1}A_+\right)$ encodes the difficulty of sampling *given all the information at hand*. See, for instance, [Safaryan, Hanzely, and Richtárik 2021, Section 2.3], [Saumard and Wellner 2014, Definition 2.9] or [Hillion, O. Johnson, and Saumard 2019, Definition 1] for similarly motivated definitions

3.7.2.3 Nonlinear preconditioning

A natural extension is to broaden the class of preconditioners to include nonlinear transformations. At present nonlinear preconditioning can be seen in the form of normalizing flows [Gabrié, Rotskoff, and Vanden-Eijnden 2022; M. Hoffman, Sountsov, et al. 2019] and measure transport [Parno and Y. M. Marzouk 2018]. Less computationally intensive transformations are considered by [L. T. Johnson and Geyer 2012] and [J. Yang, Łatuszyński, and Gareth O. Roberts 2024] in order to sample from heavy-tailed distributions.

We note that to identify a transformation $g : \mathbb{R}^d \to \mathbb{R}^d$ such that the pushforward of π under g has condition number 1 is to solve the equation

$$U(x) + \log |\det J(g(x))| = \frac{1}{2}\|g(x)\|^2$$

where $J(g(x))$ is the Jacobian of g at $x \in \mathbb{R}^d$. This is an instance of the *Monge-Ampère* equation, which is well studied in optimal transport [Peyré and Cuturi 2019]. Solvers of the Monge-Ampère exist in the literature, see [Benamou, Collino, and Mirebeau 2016; Benamou, Froese, and Oberman 2014]. Contextualising the existing analysis of the Monge-Ampère and its solvers within MCMC is a potentially fruitful line of inquiry.

There exist classes of algorithms that are equivalent to transforming existing sampling algorithms under nonlinear transformations. These include the *Riemannian manifold* algorithms of [Girolami and Calderhead 2011] (see also [Lan et al. 2015; Livingstone 2021; Patterson and Teh 2013]) and the algorithms derived from *mirror descent* [Nemirovskij and Yudin 1983] such as those see in [Chewi et al. 2020; Hsieh et al. 2018; K. S. Zhang et al. 2020]. That the algorithms derived from mirror descent are equivalent to a nonlinearly preconditioned sampling scheme is evident in their construction. For the Riemannian manifold samplers, one can show, for instance, that the Langevin diffusion

$$dY_t = \frac{1}{2}\nabla \log \tilde{\pi}(Y_t)dt + dB_t$$

under diffeomorphism $f(Y) = X$ transforms into the following SDE

$$\begin{aligned} dX_t &= \frac{1}{2}G(X_t)^{-1}\nabla \log \pi(X_t)dt + \Gamma(X_t)dt + G(X_t)^{-\frac{1}{2}}dB_t \\ \Gamma_i(X_t) &= \frac{1}{2}\sum_{j=1}^{d} \frac{\partial}{\partial x_j}\left(G(X_t)^{-1}_{ij}\right) \end{aligned} \tag{3.9}$$

with $G(x)^{-1} = J(g(x))^{-1}J(g(x))^{-T}$ where g is the inverse of f and $\pi(x) = \tilde{\pi}(y)|\det J(f(y))^{-1}|$, see [K. S. Zhang et al. 2020] for a formal statement and proof. [Livingstone and Girolami 2014; Xifara et al. 2014] show that the SDE in 3.9 is the Langevin diffusion on the Riemannian manifold with metric $G(x) \in \mathbb{R}^{d\times d}$, and therefore the same diffusion underlying the Riemannian manifold MALA algorithm of [Girolami and Calderhead 2011] is equivalent to an instance of nonlinear preconditioning. One can make a similar equivalence in the case of Riemannian manifold HMC, whereby we make a nonlinear transformation to the momentum variable used in 1.7, see [M. Hoffman, Sountsov, et al. 2019] for an explanation.

These equivalences provide motivation for further study. For instance if one can identify a g such that the metric $J(g(x))^{-1}J(g(x))^{-T}$ matches that used by [Girolami and Calderhead 2011] one can bypass the computationally costly operations inherent in the Riemannian manifold methods. One can also evaluate the benefits of using Riemannian schemes with arbitrary metrics by evaluating the change in the condition number under transformations which achieve those metrics.

3.7.2.4 Beyond well-conditioned distributions

The condition number as defined in 2.4 is restrictive in the class of models it applies to, namely distributions satisfying Assumption 24. Where Π satisfies M-smoothness and a *Poincaré inequality*: for all $f \in L^1(\Pi)$

$$\mathsf{Var}_{\pi}(f) \leq C_{\mathsf{PI}}\mathbb{E}_{\pi}[\|\nabla f\|^2]$$

with constant $C_{\mathsf{PI}} \geq 0$[S. Zhang et al. 2023, Footnote, Page 3] define it as $\kappa := C_{\mathsf{PI}}M$. They are motivated by its presence in the mixing time bounds they derive for the unadjusted Langevin sampler. An application of the Brascamp-Lieb inequality shows that $C_{\mathsf{PI}} = m^{-1}$ in the case that Π also has an m-strongly convex potential. [Y. Chen and Gatmiry 2023] also derive mixing time bounds under a more general constraint than m-strong convexity. One could alternatively use the quantities involved in their constraints and therefore the mixing time bounds to redefine the condition number. [Altmeyer 2022] constructs a *surrogate posterior* whose potential satisfies Assumption 24 and coincides with the potential of the target posterior on a region in which the target concentrates. Under assumptions, they provide polynomial time mixing bounds for unadjusted Langevin Monte Carlo using the fact that the chain will stay in the aforementioned region for exponentially long with high probability. The ability to identify such behaviour allows one to quantify the conditioning of a posterior whose potential violates Assumption 24.

Chapter 4

The occlusion process: improving sampler performance with parallel computation and variational approximation

Autocorrelations in MCMC chains increase the variance of the estimators they produce. We propose the occlusion process to mitigate this problem. It is a process that sits upon an existing MCMC sampler, and occasionally replaces its samples with ones that are decorrelated from the chain. We show that this process inherits many desirable properties from the underlying MCMC sampler, such as a Law of Large Numbers, convergence in a normed function space, and geometric ergodicity, to name a few. We show how to simulate the occlusion process at no additional time-complexity to the underlying MCMC chain. This requires a threaded computer, and a variational approximation to the target distribution. We demonstrate empirically the occlusion process' decorrelation and variance reduction capabilities on two target distributions. The first is a bimodal Gaussian mixture model in 1d and 100d. The second is the Ising model on an arbitrary graph, for which we propose a novel variational distribution.

The notation in this chapter can be found to be defined in section 7.1.4.

4.1 The occlusion process

As [Grenioux et al. 2023] remark in their assessment of normalising flows in MCMC, one thing that handicaps the methods that implement MCMC on the $T_\theta^{-1}\#\pi$ targets described in section 2.2.2 when compared with the Monte Carlo methods that use VI, is the autocorrelations that exist between the samples in the MCMC chain. It is well known that autocorrelations in the chain result in increased variance in the ensuing estimator. Our proposed algorithm, the *occlusion process*, combines MCMC with a variational approximation to the target with the aim of achieving consistent estimators with reduced variance by decorrelating the samples in the estimator.

Specifically, the occlusion process takes as input a partition of the state space into disjoint regions and is constructed upon an existing MCMC sampler. It monitors the Markov chain produced by the sampler along with the region it is in and, where possible, it produces a sample from π restricted to the region. Upon this event it will use the sample instead of the state from the Markov chain in the estimator, hence *occluding* the Markov chain state from view.

Compare the variance of a functional averaged over the run of a positive Markov chain in a given region, and the variance of that functional averaged over independent draws from π restricted to that region. It is clear that the latter will be smaller than the former. This is the way in which the occlusion process aims to reduce variance.

The process is designed to exploit parallel computation to boost its performance. Given a variational distribution and a threaded computer, the process is straightforward to implement, and its running time is the same as the MCMC sampler it is constructed on by design.

Figures 4.1 and 4.2 offer pictorial representations of the process: in Figure 4.1 we see three versions of the state space X partitioned into $\{\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3, \mathsf{X}_4\}$. The leftmost picture shows the beginning of an MCMC chain visiting regions 4, 3, 1, and 2 in that order. The occlusion process monitors the Markov chain, and attempts to produce samples $Y_1, Y_2, Y_3,$ and Y_4 from the target distribution restricted to the regions the Markov

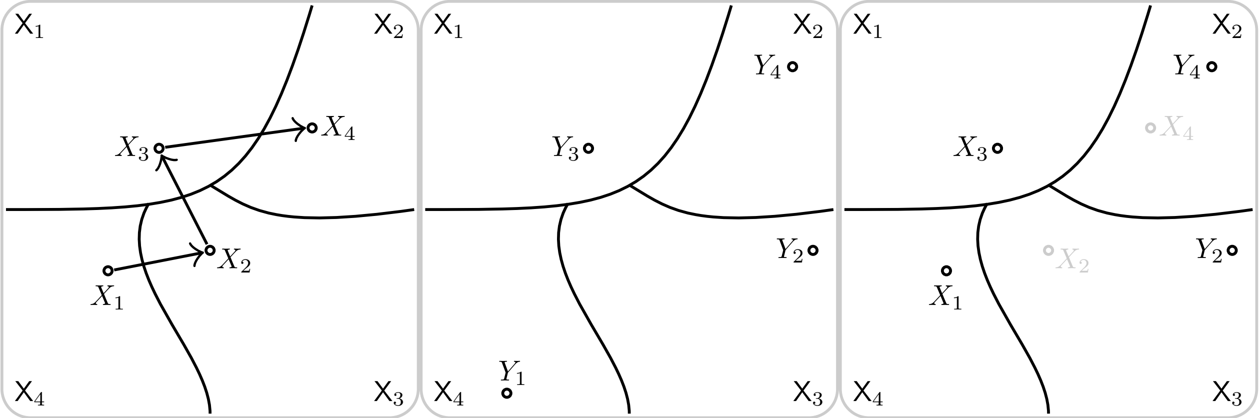

Figure 4.1: Three versions of the state space X; the leftmost with the Markov chain $\{X_t\}$ and the middle with the samples $\{Y_t\}$ taken from the target restricted to the regions that the Markov chain visits. We assume that we were only able to successfully sample Y_2 and Y_4, therefore the rightmost picture shows the samples we will use for the occlusion estimator: X_2 and X_4 have been *occluded* by Y_2 and Y_4.

chain has just visited. These samples are shown in the middle picture. Since we are in the context of MCMC, the target distribution will be such that we will not be able to successfully produce *all* of these samples within some bounded time horizon. Let's say we were only able to produce Y_2 and Y_4 in a reasonable amount of time. The rightmost picture then shows which samples will be used in the estimator: $X_1, Y_2, X_3,$ and Y_4. Samples Y_2 and Y_4 have therefore *occluded* samples X_2 and X_4.

Figure 4.2 has, from top to bottom, the estimator which uses the Markov chain samples, a DAG demonstrating the occlusion process, and the estimator after the occlusions. It relates to the example process depicted in Figure 4.1. The figure contains two additional pieces of information to Figure 4.1: we denote by $\rho(X_t)$ the region that X_t is in, and S_t is an indicator which indicates the successful production of a sample Y_t from the target, restricted to $\mathsf{X}_{\rho(X_t)}$. It demonstrates the following properties of the occlusion process: that $\{S_t\}\perp\{X_t\}\,|\{\rho(X_t)\}$, $\{S_t\}\perp\{Y_t\}\,|\{\rho(X_t)\}$, and $\{Y_t\}\perp\{X_t\}\,|\{\rho(X_t)\}$. In addition we have that $Y_t\perp Y_{t'}\,|\{\rho(X_t)\}$ and $S_t\perp S_{t'}\,|\{\rho(X_t)\}$ for all $t \neq t'$. The figure shows that the occlusion process can be viewed as a hidden Markov model. Note also that at no point do we assume that the chain is regenerative, we simply need it to be invariant to the target distribution π.

The method most similar to the method we devise here is stratified sampling with proportional allocation, see 1.2.2.3 for details. The occlusion process is similar to stratified sampling because the aforementioned

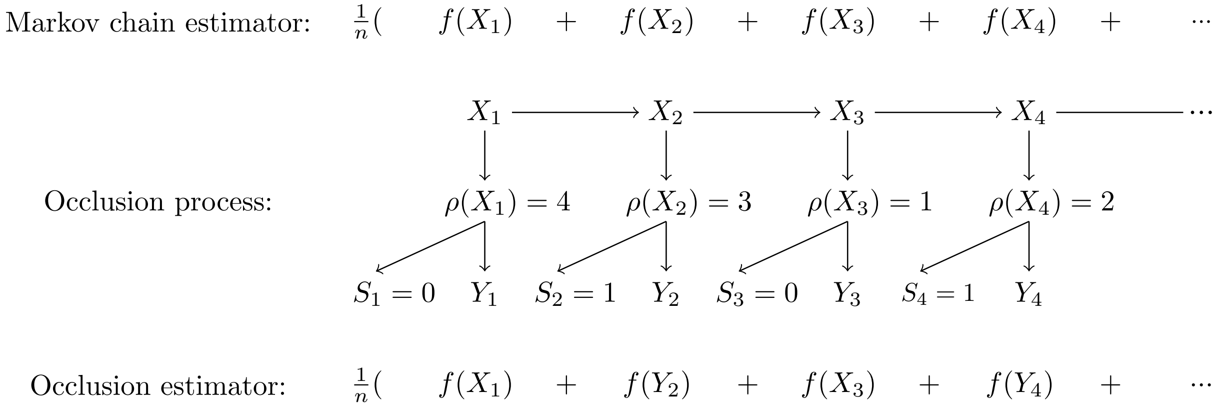
Markov chain estimator: $\frac{1}{n}(\quad f(X_1) + f(X_2) + f(X_3) + f(X_4) + \cdots$
X_1
X_2
X_3
X_4
$\cdots$
Occlusion process:
$\rho(X_1) = 4$
$\rho(X_2) = 3$
$\rho(X_3) = 1$
$\rho(X_4) = 2$
$S_1 = 0$
Y_1
$S_2 = 1$
Y_2
$S_3 = 0$
Y_3
$S_4 = 1$
Y_4
Occlusion estimator: $\frac{1}{n}(\quad f(X_1) + f(Y_2) + f(X_3) + f(Y_4) + \cdots$

Figure 4.2: The top line is the estimator constructed using states of the Markov chain $\{X_t\}$. The middle picture is a DAG representing the occlusion process: $\{X_t\}$ is the Markov chain, $\rho(X_t)$ denotes the regions visited by the Markov chain, $\{Y_t\}$ are the samples from the target restricted to those regions, and $\{S_t\}$ indicate which of those samples we were able to successfully produce. The bottom line is the occlusion estimator made up of the samples from the Markov chain, and the successfully produced Y_t's.

partition acts as a stratification, and the proportional allocation comes from the fact that the number of samples used in the estimator from π restricted to a particular region will be proportional to the length of time the Markov chain spends in that region. What separates our method from stratified sampling is that the conditions for stratified sampling are far more restrictive. Namely, stratified sampling assumes we can sample from π restricted to every region and it assumes that we know the mass of every region under π. Stratified sampling with proportional allocation produces lower variance estimators than plain Monte Carlo 1.2.2.3. One might expect to extend this result to our less restrictive context, although interestingly, we find a counterexample.

Parallel computing allows users to perform more computational operations in a fixed length of time. MCMC is a prima facie serial operation, and hence exploiting parallel threads is an attractive prospect since we are able to do additional computations alongside the Markov chain, which may need to be run for a long time to reach equilibrium. These additional computations may take the form of other Markov chains [M. Hoffman, Radul, and Sountsov 2021; Surjanovic et al. 2023]. In our case we devote the additional computing capacity to the rejection samplers mentioned above.

4.1.1 Related work

It should be noted that the occlusion process shares some similarities with the Kick-Kac samplers conceived in [Douc et al. 2023]. These samplers partition X into two measurable regions $\{\mathsf{X}_1, \mathsf{X}_2\}$, and they attempt to take independent samples from π restricted to X_1. Absent this ability, they instead use a Markov chain which is invariant to π restricted to X_1. Kac's theorem [Kac 1947] dictates that if we average a functional over a π-invariant Markov chain, beginning at one of these independent samples and ending when it re-enters X_1, we get an unbiased estimator of the expectation of the functional with respect to π.

Whilst the setup is very similar to that of the occlusion process, we note the following differences. Firstly the Kick-Kac samplers use two regions, each with a different purpose. The occlusion process uses any number of regions, and they are all treated in the same way. Secondly, to produce samples from π restricted to X_1, the Kick-Kac samplers must either wait for an independent sample, or use a Markov chain which is invariant to it. Therefore the process either has to wait for what is possibly a long time, or introduce autocorrelations. When the occlusion process achieves a sample Y_t from π restricted to a region, it is guaranteed to be independent from all other random variables given $\rho(X_t)$ (see Figure 4.2). The process itself is not fully dependent on these samples, and so can continue regardless of the probability of their production. Thirdly the Kick-Kac samplers rely on regenerations to build their estimators and so they have a random time complexity, whereas the occlusion process does not rely on such conditions and the time complexity is specified before the start of the algorithm.

4.1.2 The process

In the MCMC setting, we will not know how to sample from π nor will we know the weights $\pi(\mathsf{X}_i)$ beforehand, but we can sample from a π-invariant Markov chain. We can still sample from the π_i's, but the computational cost may be random. Specifically, we assume access to a π-invariant Markov chain $\{X_t\}_{t=1}^n$ with $X_1 \sim \pi$. We define the Markov chain estimator

$$\hat{f} := \frac{1}{n}\sum_{t=1}^{n} f(X_t) \tag{4.1}$$

We also assume that we can sample from each π_i during the running of the Markov chain, but the number of samples will depend on our specific resources such as the number of threads we have access to.

$$\begin{tikzcd}
 & X_1 \arrow[r] \arrow[d] & X_2 \arrow[r] \arrow[d] & X_3 \arrow[r] \arrow[d] & X_4 \arrow[r] \arrow[d] & \cdots \\
\text{Ideal process:} & \rho(X_1) = 4 \arrow[d] & \rho(X_2) = 3 \arrow[d] & \rho(X_3) = 4 \arrow[d] & \rho(X_4) = 2 \arrow[d] & \\
 & Y_{41} \sim \pi_4 & Y_{31} \sim \pi_3 & Y_{42} \sim \pi_4 & Y_{21} \sim \pi_2 & \\
\text{Ideal estimator:} & \frac{1}{n}(\ f(Y_{41}) \quad + & f(Y_{31}) \quad + & f(Y_{42}) \quad + & f(Y_{21}) \quad + & \cdots
\end{tikzcd}$$

Figure 4.3: A DAG representing the process in which every state X_t in the Markov chain is replaced by a sample from $\pi_{\rho(X_t)}$ in the estimator.

4.1.2.1 The fully occluded estimator

We first introduce the occluded estimator in the instance in which every sample from the Markov chain is occluded by a sample from π_i. This is, in some sense, an 'ideal' form of the final estimator, since we would use it if we were able to get sufficiently many samples from π_i. We make the conservative assumption that we only use a single sample from π_i for each sample in the Markov chain (in the corresponding region). The estimator is then

$$\hat{f}_{\text{ideal}} := \frac{1}{n}\sum_{i=1}^{R}\sum_{j=1}^{N_i} f(Y_{ij}) \tag{4.2}$$

where $Y_{ij} \sim \pi_i$ for all $j \in [N_i]$ and $N_i = \sum_{t=1}^{n} \mathbb{1}\{X_t \in \mathsf{X}_i\}$ is just the time the Markov chain spends in the ith region, for all $i \in [R]$. We assume that $Y_{ks} \perp Y_{k's'} | \{N_i\}_{i=1}^{R}$ for all possible pairs (k, s) and (k', s'). Therefore the samples Y_{ij} are independent of each other given how many of them we need to collect. We also assume that $Y_{ks} \perp \{X_t\}_{t=1}^{n} | \{N_i\}_{i=1}^{R}$ for all pairs (k, s) and so the samples Y_{ij} are independent of the Markov chain, given how long it spends in each region.

Estimates of the weights $\pi(\mathsf{X}_i)$ are naturally included via the N_i's. The estimator is defined equivalently to the stratified sampling with proportional allocation estimator except that the N_i's are selected using a Markov chain.

Figure 4.3 shows the process in the form of a DAG and the ensuing estimator, along with some sample regions into which the Markov chain states fall.

Before stating proposition 54 we first recall the definition of the resolution $\overrightarrow{\pi} f : \mathsf{X} \to \mathbb{R}$ and its orthogonal counterpart $\overleftarrow{\pi} f : \mathsf{X} \to \mathbb{R}$. The resolution is simply a piecewise constant function over the regions, whose values are determined as the expectations of f with respect to π restricted to the regions:

$$\overrightarrow{\pi} f\,(x) := \sum_{i=1}^{R} \mathbb{1}\left\{x \in \mathsf{X}_i\right\} \pi_i\,(f)$$

for all $x \in \mathsf{X}$. The orthogonal counterpart is defined as $\overleftarrow{\pi} f\,(x) := f\,(x) - \overrightarrow{\pi} f\,(x)$, so called because $\mathsf{Cov}_\pi\,(\overrightarrow{\pi} f, \overleftarrow{\pi} f) = 0$.

Proposition 54. *The estimator $\hat{f}_{\mathsf{ideal}}$ in (4.2) is unbiased. It has variance*

$$\mathsf{Var}(\hat{f}_{\mathsf{ideal}}) = \frac{1}{n}\mathsf{Var}_\pi(\overleftarrow{\pi} f) + \mathsf{Var}\left(\frac{1}{n}\sum_{t=1}^{n} \overrightarrow{\pi} f(X_t)\right) \tag{4.3}$$

Hence $\mathsf{Var}(\hat{f}_{\mathsf{ideal}})$ is just $\mathsf{Var}(\hat{f}_{\mathsf{strat}})$ where the number of samples is dictated by the proportional allocation strategy detailed in 1.2.2.3, plus a penalty paid for the Markovianity, albeit through the resolution $\overrightarrow{\pi} f$. Proof of Proposition 54 can be found in section 7.2.4.1. In the stratified sampling with proportional allocation case we had that the variance of the estimator was smaller than the variance of the Monte Carlo estimator. Interestingly, this is not the case when we compare $\mathsf{Var}(\hat{f}_{\mathsf{ideal}})$ with the variance of the Markov chain estimator.

Fact 55. *There exist distributions π, partitions $\{\mathsf{X}_i : i \in [R]\}$, functions $f \in L^2(\pi)$, and π-invariant Markov chains $\{X_t\}_{t=1}^n$ such that $\mathsf{Var}(\hat{f}_{\mathsf{ideal}}) > \mathsf{Var}(\hat{f})$ for some $n \in \mathbb{N}\backslash\{0\}$.*

The fact is obtained by using the trivial partition $\{\mathsf{X}\}$ which dictates that $\overrightarrow{\pi} f = \pi(f)$ and hence that $\mathsf{Var}(n^{-1}\sum_{t=1}^{n} \overrightarrow{\pi} f(X_t)) = 0$ and $\mathsf{Var}_\pi(\overleftarrow{\pi} f) = \mathsf{Var}_\pi(f)$. In this case $\mathsf{Var}(\hat{f}_{\mathsf{ideal}})$ is simply the variance of the MC estimator. However, there exist scenarios in which the variance of the Markov chain estimator is less than the variance of the MC estimator, for instance, let K be the operator associated with the Markov chain and $f \in L^2(\pi)$ be such that $\langle f, Kf\rangle_\pi < 0$. Then for $n = 2$ we have that

$$\begin{aligned}\mathsf{Var}(n^{-1}\sum_{t=1}^{n} f(X_t)) &= \frac{1}{2}\mathsf{Var}_\pi(f(X)) + \frac{1}{4}\mathsf{Cov}_\pi(f(X), Kf(X)) \\ &= \frac{1}{2}\mathsf{Var}_\pi(f(X)) + \frac{1}{4}\langle f, Kf\rangle_\pi - \frac{1}{4}\mathbb{E}_\pi[f(X)]^2 < \frac{1}{2}\mathsf{Var}_\pi(f(X))\end{aligned} \tag{4.4}$$

See [Neal and Jeffrey S. Rosenthal 2025, Corollary 1] or [Liu, Petros Dellaportas, and M. K. Titsias 2024, Theorem 2] for asymptotic instances of the above phenomenon. It is unknown to the authors whether Fact 55 can be extended to partitions with multiple regions:

Question 56. *Does there exist a distribution π, a partition $\{\mathsf{X}_i : i \in [R]\}$ with $R > 1$, a function $f \in L^2(\pi)$, and a π-invariant Markov chain $\{X_t\}_{t=1}^n$ such that $\mathrm{Var}(\hat{f}_{\text{ideal}}) > \mathrm{Var}(\hat{f})$ for some $n \in \mathbb{N}\backslash\{0\}$?*

[Neal and Jeffrey S. Rosenthal 2025, Corollary 1] uses antithetic Markov chains to produce estimators with lower variances than MC estimators. An antithetic Markov chain is one whose kernel is completely negative (apart from an eigenvalue at 1). The witness (4.4) to Fact 55 also exploits the existence of a negative part of K's spectrum.

Regardless of the sign of the Markov kernel, we are able to establish the following:

Lemma 57. *Let K be the Markov operator associated with the chain $\{X_t\}_{t=1}^n$. We have that*

$$\mathrm{Var}\left(n^{-1}\sum_{t=1}^{n}\overrightarrow{\pi}f(X_t)\right) \leq \mathrm{Var}\left(\hat{f}\right)$$

for all $n \in \mathbb{N}\backslash\{0\}$.

Proof of Lemma 57 can be found in section 7.2.4.2. Clearly then if $\mathrm{Var}_\pi(\overleftarrow{\pi}f)$ is sufficiently small, i.e. if $f \approx \overrightarrow{\pi}f$, we would have that $\mathrm{Var}(\hat{f}_{\text{ideal}}) \leq \mathrm{Var}(\hat{f})$. This mirrors the case in stratified sampling with proportional allocation where we achieve an optimal variance when $f = \overrightarrow{\pi}f$. Interestingly, we also have that $\mathrm{Var}(\hat{f}_{\text{ideal}}) \leq \mathrm{Var}(\hat{f})$ when $f \equiv \overleftarrow{\pi}f + \pi(f)$ and the Markov chain is positive:

Proposition 58. *The following conditions are individually sufficient for $\mathrm{Var}(\hat{f}_{\text{ideal}}) \leq \mathrm{Var}(\hat{f})$:*

1. *$f(x) = \overrightarrow{\pi}f(x)$ for all $x \in \mathsf{X}$.*

2. *$f(x) = \overleftarrow{\pi}f(x) + \pi(f)$ for all $x \in \mathsf{X}$ and that the spectrum of K is positive.*

For a proof, see Section 7.2.4.3. Condition 1. is equivalent to f being piecewise constant over the regions. We state a natural case in which the function requirement of condition 2. holds below:

Example 59. Let f be an odd function, with $f(-x) = -f(x)$ for all $x \in \mathsf{X}$, and let π be even, such that $\pi(A) = \pi(-A)$ for all $A \in \mathcal{X}$. Then if the regions satisfy $\mathsf{X}_i = -\mathsf{X}_i$ for all $i \in [R]$, we have $\overrightarrow{\pi} f \equiv \pi(f) \equiv 0$ and hence that $f \equiv \overleftarrow{\pi} f + \pi(f)$.

4.1.2.2 The occluded estimator

Since we might not have enough samples from each π_i we construct the actual estimator using samples from the Markov chain, and occlude a sample from the chain with a sample from π_i whenever one is drawn. We extend the state space X of the Markov chain to include an indicator variable $s \in \{0, 1\}$ which indicates a sample from π_i and the space of the sample from π_i. We define $\alpha : [R] \to [0, 1]$ such that $\alpha(i)$ is the probability of a sample from π_i for all $i \in [R]$. For all $(x, s, y) \in \mathsf{X} \times \{0, 1\} \times \mathsf{X}$ the occlusion process is then constructed using the Markov kernel $K_{\mathsf{occ}}((x, s, y) \to .) : \mathcal{X} \times \{0, 1\} \times \mathcal{X} \to \mathbb{R}^+$ with

$$K_{\mathsf{occ}}((x, s, y) \to (dx', s', dy')) := K(x \to dx') A\left(s' \,|x'\right) \pi_{\rho(x')}(dy') \tag{4.5}$$

where we define

$$A\left(s \,|x\right) := \alpha(\rho(x)) \mathbb{1}\{s = 1\} + (1 - \alpha(\rho(x)))\, \mathbb{1}\{s = 0\}.$$

We use the estimator

$$\hat{f}_{\mathsf{occ}} := \frac{1}{n} \sum_{t=1}^{n} f_{\mathsf{occ}}(X_t, S_t, Y_t) \tag{4.6}$$

where $f_{\mathsf{occ}}(X_t, S_t, Y_t) := \mathbb{1}\{S_t = 0\} f(X_t) + \mathbb{1}\{S_t = 1\} f(Y_t)$, and so by construction X_t is occluded by Y_t whenever we obtain a successful sample from $\pi_{\rho(X_t)}$. Even though the process described above generates Y_t for all $t \in [n]$, in actuality we will only need to generate Y_t when $S_t = 1$ since it is only then that Y_t is included in the estimator. Note that if $\alpha \equiv 1$ we have that $\hat{f}_{\mathsf{occ}}$ is just $\hat{f}_{\mathsf{ideal}}$ in (4.2). For a pictorial representation, see Figure (4.2).

Since K is π-invariant we have that K_{occ} is π_{occ}-invariant where we define

$$\pi_{\mathsf{occ}}(dx, s, dy) := \pi(dx) \bigg(\alpha(\rho(x)) \mathbb{1}\{s = 1\} + (1 - \alpha(\rho(x)))\, \mathbb{1}\{s = 0\} \bigg) \pi_{\rho(x)}(dy) \tag{4.7}$$

for all $dx \in \mathcal{X}, s \in \{0,1\}, dy \in \mathcal{X}$. Hence the marginal processes of the occlusion process obey the following laws in equilibrium:

- The marginal law of $\{X_t\}$ in equilibrium is π.
- The marginal law of $\{S_t\}$ in equilibrium is defined by the probability mass function $\bar{\alpha}\mathbb{1}\{s=1\} + (1-\bar{\alpha})\mathbb{1}\{s=0\}$ where
$$\bar{\alpha} := \sum_{i=1}^{R} \alpha(i)\pi(\mathsf{X}_i)$$
- The marginal law of $\{Y_t\}$ in equilibrium is π.
- Therefore defining $Z_t := \mathbb{1}\{S_t = 0\}X_t + \mathbb{1}\{S_t = 1\}Y_t$ such that $f_{\text{occ}}(X_t, S_t, Y_t) \equiv f(Z_t)$ we have that the marginal law of $\{Z_t\}$ in equilibrium is π.

Note that $\pi_{\text{occ}}(f_{\text{occ}}) = \pi(f)$ and $\text{Var}_{\pi_{\text{occ}}}(f_{\text{occ}}) = \text{Var}_{\pi}(f)$. For the bias and variance of $\hat{f}_{\text{occ}}$ we have the following:

Proposition 60. *When evaluated on the process $\{(X_t, S_t, Y_t)\}_{t=1}^{n}$ with $(X_1, S_1, Y_1) \sim \pi_{\text{occ}}$, $\hat{f}_{\text{occ}}$ (4.6) is unbiased and has variance*
$$\text{Var}(\hat{f}_{\text{occ}}) = \text{Var}(\hat{f}_{\text{ideal}}) + \frac{1}{n} 2 \sum_{k=1}^{n-1} \frac{n-k}{n} C_k \tag{4.8}$$
where
$$C_k := \text{Cov}_{\pi}(f_a(X), K^k f_a(X)) - \text{Cov}_{\pi}(\overrightarrow{\pi} f_a(X), K^k \overrightarrow{\pi} f_a(X))$$
with $f_a(x) := (1-\alpha(\rho(x)))\, f(x)$ for all $x \in \mathsf{X}$ and $\text{Var}(\hat{f}_{\text{ideal}})$ is as defined in (4.3).

Proof of the above proposition can be found in section 7.2.4.4. Hence when $\alpha \equiv 0$ we have $\text{Var}(\hat{f}_{\text{occ}}) = \text{Var}(\hat{f})$ as expected, and when $\alpha \equiv 1$ we have $\text{Var}(\hat{f}_{\text{occ}}) = \text{Var}(\hat{f}_{\text{ideal}})$. In this latter $\alpha \equiv 1$ case we would have $\text{Var}(\hat{f}_{\text{occ}}) \leq \text{Var}(\hat{f})$ when f satisfies either condition 1. or 2. from Proposition 58. These conditions may seem improbable, but in section 4.1.5.2 we see a practical instantiation of condition 2. in the form outlined in Example 59 where the function f is odd and the target measure π is even. See Figure 4.7 for the concomitant reductions of variance, and Section 4.1.5.2 for a justification of why in this case, we satisfy condition 2.

4.1.3 Inherited theoretical properties of the occlusion process

4.1.3.1 Basic properties

Since the occlusion process $\{(X_t, S_t, Y_t)\}_{t=1}^n$ is derived from an underlying Markov chain $\{X_t\}_{t=1}^n$, we would like the process to inherit the 'good' properties of the Markov chain when they exist. For example we have the following inheritances:

Proposition 61. *If K is π-reversible then K_{occ} is π_{occ}-reversible.*

A proof of Proposition 61 can be found in section 7.2.4.5

Proposition 62. *If f is in $L^2(\pi)$ then f_{occ} is in $L^2(\pi_{\text{occ}})$.*

Proof can be found in section 7.2.4.6.

4.1.3.2 Law of large numbers

The first non-basic result that the occlusion process inherits from the Markov chain is a Law of Large Numbers (LLN). Since no quantitative rates are involved in LLNs we can not directly compare LLNs on the Markov chain with LLNs on the occlusion process but the inheritance result stands nonetheless.

Theorem 63. *The following are equivalent:*

1. *For all probability measures μ on $\mathcal{X}$ and measurable functions $g : \mathsf{X} \to \mathbb{R}$ such that $g \in L^1(\pi)$ we have that*

$$\lim_{n\to\infty} n^{-1} \sum_{t=1}^{n} g(X_t) = \pi(g)$$

   almost surely with $X_1 \sim \mu$.

2. *For all probability measures μ_{occ} on $\mathcal{X} \times 2^{\{0,1\}} \times \mathcal{X}$ and for all measurable functions $g_{\text{occ}} : \mathsf{X} \times \{0,1\} \times \mathsf{X}$ such that $g_{\text{occ}} \in L^1(\pi_{\text{occ}})$ we have that*

$$\lim_{n\to\infty} n^{-1} \sum_{t=1}^{n} g_{\text{occ}}(X_t, S_t, Y_t) = \pi_{\text{occ}}(g_{\text{occ}})$$

   almost surely with $(X_1, S_1, Y_1) \sim \mu_{\text{occ}}$.

The proof in section 7.2.4.7 uses [Douc et al. 2023, Proposition 3.5].

4.1.3.3 Convergence in a normed function space

One way to measure the efficiency of an MCMC algorithm with kernel K is to compare K^t to its equilibrium distribution via their action on measurable functions. The outputs of these actions are themselves functions, so to compare we need to use a norm defined on the appropriate function space.

Definition 64. A Markov chain with kernel K converges to a distribution π in a normed function space $(\mathsf{F}, \|.\|)$ with rate function $r : \mathbb{N}\backslash\{0\} \to \mathbb{R}^+$ when

$$\|K^t f - \pi(f)\| \leq C_f r(t)$$

for all $f \in \mathsf{F}$ and $t \in \mathbb{N}\backslash\{0\}$, where $C_f > 0$ is a constant that depends on f and $r(t) \searrow 0$.

Often we have that $C_f = C\|f - \pi(f)\|$ with $C > 0$. For the occlusion process $\{(X_t, S_t, Y_t)\}_{t=1}^n$ to inherit convergence in a normed space of $\{X_t\}_{t=1}^n$ we need some way to relate the normed spaces that K acts on to the normed spaces that K_{occ} acts on. Given a normed function space $(\mathsf{F}, \|.\|)$ of functions on X, let $\mathsf{F}_{\mathsf{occ}}$ be the vector space of measurable functions of the form $g(x, s, y) = \mathbb{1}\{s = 0\}f(x) + \mathbb{1}\{s = 1\}f(y)$ on $\mathsf{X} \times \{0, 1\} \times \mathsf{X}$ into $\mathbb{R}$ where $f \in (\mathsf{F}, \|.\|)$. These are the functions that K_{occ} will act on to produce the occlusion process. We define the class of normed spaces

$$\mathcal{C}_{\|.\|} := \{(\mathsf{G}, \|.\|_{\mathsf{G}}) : \mathsf{G} \subseteq \mathsf{F}_{\mathsf{occ}}, \|g\|_{\mathsf{G}} = \|g\| \text{ when } g \text{ is a function of its first argument only}\}$$

For instance, if $\|.\|$ is the sup norm on functions from $\mathsf{X} \to \mathbb{R}$ then $\mathcal{C}_{\|.\|}$will contain the normed spaces with the sup norm on functions from $\mathsf{F}_{\mathsf{occ}}$. Equipped with this definition, we have the following inheritance

Theorem 65. *Say the Markov chain $\{X_t\}_{t=1}^n$ converges to π in the normed function space $(\mathsf{F}, \|.\|)$ with rate function $r(t)$ and constant C_f. Then for all normed function spaces $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$ and for all functions $g(x, s, y) = \mathbb{1}\{s = 0\}f(x) + \mathbb{1}\{s = 1\}f(y) \in (\mathsf{G}, \|.\|_{\mathsf{G}})$ we have that*

$$\|K_{\mathsf{occ}}^t g - \pi_{\mathsf{occ}}(g)\|_{\mathsf{G}} \leq C_{f_\alpha} r(t)$$

where

$$f_\alpha(x) := (1 - \alpha(\rho(x)))\, f(x) + \alpha(\rho(x)) \overrightarrow{\pi} f(x)$$

K_{occ} *is as defined in (4.5), and* π_{occ} *is as defined in (4.7)*

The proof relies on the fact that for a given function $g \in (\mathsf{G}, \|.\|_{\mathsf{G}})$ with $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$ we have $K^t_{\text{occ}} g = K^t f_\alpha$, see section 7.2.4.8. Because of this fact, we also inherit lower bounds on the convergence of the occlusion process:

Proposition 66. *Say the convergence of the Markov chain* $\{X_t\}_{t=1}^n$ *to* π *in the normed function space* $(\mathsf{F}, \|.\|)$ *is lower bounded as follows:*

$$\|K^t f - \pi(f)\| \geq C^\dagger_f r^\dagger(t)$$

for some $C^\dagger_f > 0$ *and* $r^\dagger : \mathbb{N}\backslash\{0\} \to \mathbb{R}^+$ *and for all* $f \in (\mathsf{F}, \|.\|)$*. Then for all normed function spaces* $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$ *and for all functions* $g(x, s, y) = \mathbb{1}\{s = 0\} f(x) + \mathbb{1}\{s = 1\} f(y) \in (\mathsf{G}, \|.\|_{\mathsf{G}})$ *we have that*

$$\|K^t_{\text{occ}} g - \pi_{\text{occ}}(g)\|_{\mathsf{G}} \geq C^\dagger_{f_\alpha} r^\dagger(t)$$

where

$$f_\alpha(x) := (1 - \alpha(\rho(x)))\, f(x) + \alpha(\rho(x)) \overrightarrow{\pi} f(x)$$

K_{occ} *is as defined in (4.5), and* π_{occ} *is as defined in (4.7)*

If $C_{f_\alpha} \leq C_f$ or $C^\dagger_{f_\alpha} \leq C^\dagger_f$ we could have better convergence of the occlusion process as compared with the base Markov chain it sits upon. The following example shows such a case.

Example 67. Say $(\mathsf{F}, \|.\|)$ is the space of bounded continuous functions with the sup norm, and that $\{X_t\}_{t=1}^n$ converges to π in $(\mathsf{F}, \|.\|)$ with rate function $r(t)$ and constant $C_f := C\|f - \pi(f)\|$ with $C > 0$. Then the occlusion process $\{(X_t, S_t, Y_t)\}_{t=1}^n$ converges to π_{occ} in all spaces $(\mathsf{G}, \|.\|_{\mathsf{G}})$ in $\mathcal{C}_{\|.\|}$ with rate function $r(t)$ and constant $C_{f_\alpha} := C\|f_\alpha - \pi(f_\alpha)\|$. Note that $C_{f_\alpha} \leq C_f$. For a proof of this fact see 7.2.4.9. The same is true for any similarly defined constants $C^\dagger_{f_\alpha}$ and $C^\dagger_f$ in lower bounds on convergence, for exactly the same reasons.

4.1.3.4 Convergence in a normed measure space

Another way to examine the efficiency of an MCMC algorithm with kernel K is to establish bounds on the distance between μK^t and π in some normed measure space.

Definition 68. A Markov chain with kernel K converges to a distribution π in a normed measure space $(\mathsf{M}, \|.\|)$ with rate function $r(t) : \mathbb{N}\backslash\{0\} \to \mathbb{R}^+$ when

$$\|\mu K^t - \pi\| \leq C_\mu r(t)$$

for all measures $\mu \in (\mathsf{M}, \|.\|)$ and $t \in \mathbb{N}\backslash\{0\}$, where $C_\mu > 0$ is a constant that depends on μ and $r(t) \searrow 0$.

For the above definition to work, we need that measures in $(\mathsf{M}, \|.\|)$ have the same state space as K. The role of the norm in the above definition is to provide the distance between μK^t and π. So given a distance, we could equivalently state the convergence result without a norm. One class of distances we could use is the class of integral probability metrics (IPMs). It is defined using a function space F as follows:

Definition 69. The integral probability metric D_{F} defined by a function space F between measures P and Q is defined as

$$D_{\mathsf{F}}(P, Q) := \sup_{f \in \mathsf{F}} |P(f) - Q(f)|$$

Examples of IPMs include the Wasserstein-1 distance and the total variation distance. As in the previous section, we define $\mathcal{C} := \{\mathsf{G} :$ for all $g \in \mathsf{G}$ we have $g(x, s, y) = \mathbb{1}\{s = 0\} f(x) + \mathbb{1}\{s = 1\} f(y)$ with $f \in \mathsf{F}\}$ as the class of function spaces whose members the occlusion process admits in its estimator. Equipped with these definitions we have the following inheritance result:

Theorem 70. *Say the Markov chain $\{X_t\}_{t=1}^n$ starting at $X_1 \sim \mu$ converges to π in the integral probability metric defined over F with rate function $r(t)$ and constant C_μ. Then the occlusion process $\{(X_t, S_t, Y_t)\}_{t=1}^n$ with $X_1 \sim \mu$ converges to π_{occ} (4.7) in the integral probability metrics defined over the members of $\mathcal{C}$ with rate function $r(t)$ and constant C_μ as soon as*

$$f_\alpha(x) := (1 - \alpha(\rho(x)))\, f(x) + \alpha(\rho(x)) \overrightarrow{\pi} f(x)$$

is in F*.*

A proof of the above is found in section 7.2.4.10.

Example 71. Take $\mathsf{F} = \{f : \mathsf{X} \to [0,1]\}$. Then D_{F} is the total variation distance. That f_α is in F is clear since it is the convex combination of two functions in F. Therefore if $\{X_t\}_{t=1}^n$ converges to π in D_{F} the above result holds and we get convergence in total variation of the occlusion process with the same rate.

That the occlusion process inherits convergence in IPMs is an example of a wider class of inheritance results: we include it here as an example. For a more general result on the inheritance of convergence in normed measure spaces see section 7.3.1 in appendix B.

4.1.3.5 Geometric ergodicity

Geometric ergodicity is a particular kind of convergence in a normed measure space, where μ is a point mass, the rate function is geometric, and the norm is the total variation norm. See 20 for a definition. When a functional $f : \mathsf{X} \to \mathbb{R}$ is averaged over a geometrically ergodic chain, it only takes a small amount of additional work to establish a CLT. For instance [Chan and Geyer 1994, Theorem 2] show that when $f \in L^{2+\varepsilon}(\pi)$ is averaged over a geometrically ergodic chain, for some $\varepsilon > 0$, we are guaranteed a CLT. [G. Roberts and J. Rosenthal 1997, Corollary 2.1] show that the same is true for $f \in L^2(\pi)$ when the chain is π-reversible.

Theorem 72. *When the chain* $\{X_t\}_{t=1}^n$ *generated by* K *is geometrically ergodic, the occlusion process* $\{(X_t, S_t, Y_t)\}_{t=1}^n$ *generated by* K_{occ} *(4.5) is geometrically ergodic.*

This result is also a corollary of Theorem 70, however it also stands on its own, see section 7.2.4.11 for the proof.

Corollary 73. *When the chain generated by* K *is geometrically ergodic and* π*-reversible and when* $f \in L^2(\pi)$*,* $\hat{f}_{\mathsf{occ}}$ *admits the following CLT:*

$$\sqrt{n}\left(\hat{f}_{\mathsf{occ}} - \pi(f)\right) \xrightarrow{d} N(0, \sigma^2_{\mathsf{occ}}) \text{ as } n \to \infty$$

where

$$\sigma^2_{\text{occ}} := \lim_{n\to\infty} n\text{Var}(\hat{f}_{\text{occ}}) = \text{Var}_{\pi_{\text{occ}}}(f_{\text{occ}}) + 2\sum_{k=1}^{\infty} \text{Cov}_{\pi_{\text{occ}}}(f_{\text{occ}}, K^k f_{\text{occ}}) < \infty$$

Proof of the above can be found in section 7.2.4.12. It would be desirable for the occlusion process to inherit a CLT type result from a CLT in the Markov chain $\{X_t\}_{t=1}^{n}$. However to establish such a result would be to establish necessary conditions for a Markov chain CLT which don't currently exist in the literature.

Proposition 74. *When the chain generated by K is geometrically ergodic and π-reversible, and when $f \in L^2(\pi)$ we have that*

$$\begin{aligned}\lim_{n\to\infty} n\text{Var}(\hat{f}_{\text{occ}}) &= \text{Var}_{\pi_{\text{occ}}}(f_{\text{occ}}) + 2\sum_{k=1}^{\infty} \text{Cov}_{\pi_{\text{occ}}}(f_{\text{occ}}, K^k f_{\text{occ}}) \\ &= \text{Var}_{\pi}(f) + 2\sum_{k=1}^{\infty} \Big(\text{Cov}_{\pi}(\overrightarrow{P} f(X), K^k \overset{\rightrightarrows}{\pi} f(X)) + C_k\Big) < \infty\end{aligned}$$

where

$$C_k := \text{Cov}_{\pi}(f_a(X), K^k f_a(X)) - \text{Cov}_{\pi}(\overset{\rightrightarrows}{\pi} f_a(X), K^k \overset{\rightrightarrows}{\pi} f_a(X))$$

with $f_a(x) := (1 - \alpha(\rho(x)))\, f(x)$ *for all* $x \in \mathsf{X}$*.*

Proof of the above proposition may be found in Section 7.2.4.13.

4.1.4 Efficient simulation of the occlusion process

We now describe exactly how one can simulate the occlusion process generated by the kernel in (4.5). We show that given access to a multi-threaded computer and a variational approximation Q we can calculate $\hat{f}_{\text{occ}}$ (4.6) at no additional computational cost to calculating $\hat{f}$. However due to this requirement, we cede control over the values of the $\alpha(i)$'s.

4.1.4.1 Sampling from the π_i's and defining the regions

If we are using MCMC to approximate samples from π, sampling from π restricted to the regions $\mathsf{X}_1, \mathsf{X}_2, ...$ will be a non-trivial task.

The trick we employ is to choose our sampling mechanism, then define the regions such that the samples from π restricted to them is somehow guaranteed. Say we have access to Q: an easy to sample from distributional approximation to π. Let Y be the result of the following sampling mechanism: given an arbitrary constant $C > 0$ and oracle access to the unnormalised Radon-Nikodym derivative $d\tilde{\pi}/d\tilde{Q}$ between π and Q

1. Sample $Y \sim Q$ and $U \sim$ Unif$[0, 1]$ independently of each other.
2. If
$$U \leq \frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y) \text{ and } \frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y) \leq 1$$
output Y, otherwise go to step 1.

Then if we call X_C the region $\{y \in \mathsf{X} : \frac{1}{C}d\tilde{\pi}/d\tilde{Q}(y) \leq 1\}$ we have that Y sampled according to the mechanism above is distributed according to π restricted to X_C, see section 7.2.4.14 for proof. This mechanism has its roots in [Tierney 1994, Section 2.3.4] and it is also used in [Douc et al. 2023].

To form the partition of X we do as follows: take $0 =: C_0 < C_1 < C_2 < \cdots < C_{R-1} < C_R := \infty$ and define $\mathsf{X}_i := \{x : d\tilde{\pi}/d\tilde{Q}(x) \in [C_{i-1}, C_i)\}$. Then to collect samples from the π_i's we simply execute step 1 of the above mechanism, check which X_i Y is in using the Radon-Nikodym derivative, and then use the appropriate C_i in step 2. The particular π_i sampled from will therefore be random and if Y falls in X_R it is automatically rejected. The probability of a sample from π_i in a single iteration is then

$$\pi_{Y\sim Q}\left(U \leq \frac{1}{C_i}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y) \cap Y \in \mathsf{X}_i\right) = \frac{Z_\pi}{Z_Q}\frac{1}{C_i}\pi(\mathsf{X}_\mathsf{i}) \tag{4.9}$$

where Z_π and Z_Q are the normalising constants of $\tilde{\pi}$ and $\tilde{Q}$ respectively.

4.1.4.2 Implementation and computational cost

Strategy for a general target π In general we dedicate a single thread of compute to sampling the Markov chain $\{X_t\}_{t=1}^n$, and the rest of the available threads to sampling from the π_i's. A straightforward way to do this is by having these final threads work in an embarrassingly parallel fashion, repeatedly iterating through the steps detailed in section 4.1.4.1. We stop all the threads upon some condition e.g. $\{X_t\}$ has reached a given length. The fact that the threads work in an embarrassingly parallel fashion minimises communication costs, and therefore maximises the amount of compute going into sampling from the π_i's. It also minimises the amount of programmer time since in general it is far easier to implement embarrassingly parallel code than code in which the threads communicate.

At the end of the procedure we then have the Markov chain $\{X_t\}_{t=1}^n$, the list of regions it visits $\{\rho(X_t)\}_{t=1}^n$, and $\{Y_{ij}\}_{j=1}^{N_i}$ for all $i \in [R]$ where N_i is the number of samples from π_i. We may possibly have $N_i = 0$ for some i, and we may even have $N_1 + \cdots + N_R > n$. We then use some procedure to assign Y_{ij}'s to the X_t's we wish to occlude, which defines the $\alpha(\rho(x))$ term in (4.5).

Define T_i as the amount of time the Markov chain spends in region i and $I_i \subseteq \{X_t\}_{t=1}^n$ the set of states in that region. In our numerical experiments for each $i \in [R]$ we assign $\min\{N_i, T_i\}$ samples from $\{Y_{ij}\}_{j=1}^{N_i}$ to occlude a uniformly sampled size $\min\{N_i, T_i\}$ subset of I_i. Therefore $\alpha(i) := \mathbb{E}[\min\{1, N_i/T_i\}]$ for all $i \in [R]$ where the expectation is over the randomness of the entire process. This is equivalent in distribution to occluding at each step of the process with probability $\min\{1, N_i/T_i\}$ although it is more efficient since it uses as many samples from $\{Y_{ij}\}_{j=1}^{N_i}$ as possible, for each $i \in [R]$.

See Algorithm 4.1 for pseudocode of the whole procedure.

Approaches tailored to particular targets π The nature of the sampling problem may offer more efficient alternatives to the general purpose strategy detailed above. Sampling from $K(x \to .)$ may be slow. For instance π may be a Bayesian posterior distribution whose likelihood evaluations necessitate the solution of a system of differential equations, see [J. Ma 2020]. Or perhaps $K(x \to .)$ is the kernel of a Metropolis-Hastings algorithm whose proposal distribution takes a while to sample from e.g. preconditioned Hamiltonian Monte Carlo in high dimension. In this regime, in the time it takes K to move from a given X_t to X_{t+1} we could concentrate the parallel computational resources to sampling from $\pi_{\rho(X_t)}$. This however incurs some communication costs between threads.

In another scenario, it may be that the regions are defined naturally given the state space. For example it may be that X can be written as the disjoint union of R connected sets. Then we could have a variational distribution Q_i for each region X$_i$.

Algorithm 4.1 Embarrassingly parallel occlusion process

inputs: Chain length n, Initial state X_0, $1 + C_{\text{rej}}$ computational threads, Variational distribution Q, Constants $0 =: C_0 < C_1 < \cdots < C_{R-1} < C_R := \infty$.

outputs: A π-invariant Markov chain $\{X_t\}_{t=1}^{n}$, the regions it visits $\{\rho(X_t)\}_{t=1}^{n}$, sets of samples $\left\{\{Y_{ij}\}_{j=1}^{N_i}\right\}_{i=1}^{R}$ with $Y_{ij} \sim \pi_i$ for all $j \in [N_i]$ and $i \in [R]$.

Markov chain

In thread 1:

```
for $t \in n$ do
    Sample $X_t \sim K(X_{t-1} \to .)$ and store $X_t$ along with its region $\rho(X_t)$.
```

Rejection samplers

In threads $j \in \{2, \ldots, C_{\text{rej}}\}$ concurrently:

```
for $j \in \{2, \ldots, C_{\texttt{rej}}\}$ do
  1. Sample $Y \sim Q$ and $U \sim \texttt{Uniform}[0,1]$ independently.
  2. Determine the constant to use in the rejection sampler $C_{\rho(Y)}$ and determine
     the region $i := \rho(Y)$ that $Y$ is in.
  3. If $UC_i \leq d\tilde{\pi}/d\tilde{Q}(Y)$ then append the sample to $\{Y_{ij}\}_{j=1}^{N_i}$ (such that
     $N_i \leftarrow N_i + 1$), otherwise go to step 1.
```

Postprocessing

inputs: A π-invariant Markov chain $\{X_t\}_{t=1}^{n}$, the regions it visits $\{\rho(X_t)\}_{t=1}^{n}$, sets of samples $\left\{\{Y_{ij}\}_{j=1}^{N_i}\right\}_{i=1}^{R}$ with $Y_{ij} \sim \pi_i$ for all $j \in [N_i]$ and $i \in [R]$.

outputs: The occlusion estimator $\hat{f}_{\text{occ}}$ defined in 4.6

Initialise the indicator sequence $\{S_t\}_{t=1}^{n} \leftarrow \{0\}_{t=1}^{n}$ and the sequence $\{Y_t\}_{t=1}^{n} \leftarrow \{\text{NaN}\}_{t=1}^{n}$

```
for $i \in [R]$ do
  1. Get the set of times the Markov chain $\{X_t\}_{t=1}^{n}$ was in region $i$:
     $\mathcal{T}_i := \{t : X_t \in \mathsf{X}_i, t \in [n]\}$ and the amount of time $T_i := |\mathcal{T}_i|$.
  2. If $N_i \geq T_i$ set $S_t \leftarrow 1$ and $Y_t \leftarrow Y_{it}$ for all $t \in \mathcal{T}_i$.
  3. Else:
     (a) Sample a subset $\mathcal{T}_i' \subseteq \mathcal{T}_i$ uniformly from the subsets of size $N_i$ in $\mathcal{T}_i$.
     (b) Set $S_t \leftarrow 1$ and $Y_t \leftarrow Y_{it}$ for all $t \in \mathcal{T}_i'$.
```

Output

$$\hat{f}_{\text{occ}} = \frac{1}{n} \sum_{t=1}^{n} f_{\text{occ}}(X_t, S_t, Y_t)$$

where $f_{\text{occ}}(X_t, S_t, Y_t) := \mathbb{1}\{S_t = 0\} f(X_t) + \mathbb{1}\{S_t = 1\} f(Y_t)$

4.1.4.3 Choice of regions

The way that the regions $\{\mathsf{X}_i : i \in [R]\}$ influence the statistical properties of the occlusion process is via the probabilities of taking sucessful rejection samples $\alpha(i)$ for $i \in [R]$ and the amount of time the Markov chain $\{X_t\}_{t=1}^n$ spends in each region. As stated at the end of section 4.1.2.2, if the regions are such that $\alpha \equiv 1$ i.e. each π_i is easy to sample from, we would have Var$(\hat{f}_{\text{occ}}) =$ Var$(\hat{f}_{\text{ideal}})$. In the opposite scenario, if the regions are such that $\alpha \equiv 0$ i.e. each π_i is impossibly hard to sample from, we would have Var$(\hat{f}_{\text{occ}}) =$ Var$(\hat{f})$.

It is difficult to tell from the form of (4.8) how exactly Var$(\hat{f}_{\text{occ}})$ will vary with each individual $\alpha(i)$, but as stated below Lemma 57 we should choose the regions to have $f = \overrightarrow{\pi} f$ since such a condition ensures that Var$(\hat{f}_{\text{occ}}) =$ Var$(\hat{f}_{\text{ideal}}) \leq$ Var$(\hat{f})$.

With the general strategy explained in section 4.1.4.2 we assume that we use $C_{\text{rej}} \in \mathbb{N}\backslash\{0\}$ threads for the rejection sampling component, and that each thread can propose a single rejection sample per step in the Markov chain $\{X_t\}_{n=1}^t$. Therefore we have nC_{rej} rejection sample attempts in total and hence $\alpha(i) = \mathbb{E}[\min\{1, N_i/T_i\}]$ where

$$N_i \sim \text{Binomial}\left(nC_{\text{rej}}, \frac{Z_\pi}{Z_Q}\frac{1}{C_i}\pi(\mathsf{X_i})\right)$$

and each T_i will have expectation $n\pi(\mathsf{X}_i)$ for all $i \in [R]$. Therefore each $\alpha(i)$ will depend on the length of the Markov chain n.

4.1.5 Numerical experiments

4.1.5.1 Bimodal Gaussian mixture

As a first demonstration of the behaviour of the occlusion process we look at sampling from a bimodal Gaussian mixture

$$\pi(dx) := (1-p) \times \mathcal{N}(x; 0, \mathbf{I}_d)dx + p \times \mathcal{N}(x; m, \sigma^2\mathbf{I}_d)dx \tag{4.10}$$

for $p \in [0,1]$, $m \in \mathbb{R}^d$, and $\sigma^2 > 0$. Say we are given a variational distribution Q that approximates well the first component of the mixture. If we used such a distribution to do inference, then our estimates would be biased by the fact that the distribution ignores the second component. For a σ^2 which is sufficiently small the Radon-Nikodym derivative $d\pi/dQ(x)$ will have a large upper bound. This would be prohibitive against

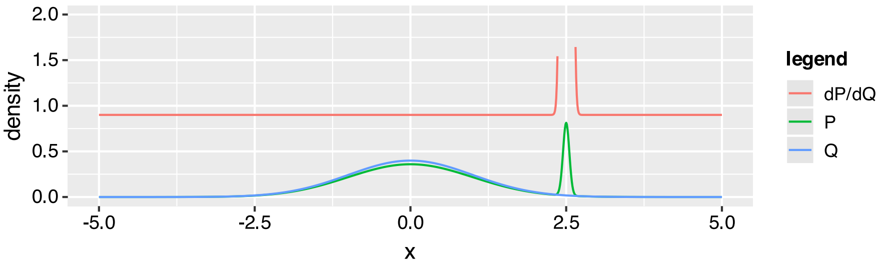

Figure 4.4: The target density (P in the legend), the variational density (Q in the legend), and the Radon-Nikodym derivative (dP/dQ in the legend) for the bimodal Gaussian example in $d = 1$.

doing vanilla rejection sampling with Q as the proposal distribution. Therefore to resolve these issues, we use the occlusion process so as to take advantage of the variational distribution and the unbiasedness of an underlying Markov chain targeting π.

Experiment setup We perform two experiments, one with $d = 1$ and the other with $d = 100$. The mean m of the second component has all its elements set to zero, apart from its first which we set to 2.5. We set $\sigma^2 = 0.05$ and $p = 0.1$. We set the variational distribution to be the Laplace approximation where we find the mode using gradient descent on the negative log-density of the target. In every case this resulted in Q being a standard normal. For a plot of the target density, variational density, and the Radon-Nikodym derivative in $d = 1$ see Figure 4.4. Note that the Radon-Nikodym derivative is mostly flat apart from around $x = 2.5$ where it becomes large.

We use $R = 2$ regions, defining the first region as $\{x : d\pi/dQ(x) \leq 1\}$ and the second region as the complement of the first. From Figure 4.4 we can see that the first region encompasses most of the state space, apart from a small area around the mean of the second component of the target. The number of total threads we use is 7, and we employ them in the fashion according to Section 4.1.4.2, thereby having $C_{\text{rej}} = 6$ threads for the rejection samplers, and one thread for the underlying Markov chain. For this underlying chain we use RWM, for which we set the step size to $2.38/\sqrt{d}$ as recommended in [Gareth O. Roberts and Jeffrey S. Rosenthal 2001]. In each case we run the sampler for 20 seconds.

Experiment results In Figure 4.5 we show the results of the two experiments, with the $d = 1$ results on the left and the $d = 100$ results on the right. The top row shows the first component of the RWM chains $\{X_t\}_{t=1}^n$, the row below shows the first component of the occluded chain $\{\mathbb{1}\{S_t = 0\}X_t + \mathbb{1}\{S_t = 1\}Y_t\}_{t=1}^n$. In the titles of these plots is the proportion of total states in the chain that were occluded: in the notation of Section 4.1.4.2 this is $(N_1 + \cdots + N_R)/n$. The bottom four plots show the autocorrelation functions of the two processes. The top row has the autocorrelation function of the first component of the RWM chain, and the bottom row has the autocorrelation function of the first component of the occluded chain.

The main impression given by the figure is that the addition of the occlusions has the effect of decorrelating the process. This is evinced in the traceplots most clearly in the $d = 100$ case, where the RWM chain is clearly displaying the highly autocorrelated behaviour of a diffusion, whereas the states in the occluded process are decorrelated from each other. This is further evinced by the autocorrelation function plots, where in every case at every lag the autocorrelation of the RWM chain is greater than that of the occlusion process. Again this is markedly so in the $d = 100$ case.

We note that although the use of occlusion decorrelates the underlying process, in this case it does not debias it. See, for instance, the $d = 100$ plots on the right hand side, where the RWM chain should spend 10% of its time in the second component of the target whereas the traceplots (and inspection of the output) shows that it spends 0%. Therefore the occluded chain also spends no time in the second component, due to the fact that by definition it only ever visits regions that the RWM also visits.

4.1.5.2 The Ising model

The Ising model describes the behaviour of magnetic spins in $\sigma_i \in \{-1, 1\}$ at nodes in a graph (V, E). Classically the graph in question is a regular cubic lattice although the model has been generalised to arbitrary graphs [Bresler 2015; Delgado 2015; Mosseri 2015]. We assume that no external magnetic field is present, and that the graph is finite with $|V| = N$ vertices. A single state $\sigma \in \{-1, 1\}^N$ describes the spin configuration over all vertices, and its potential energy can be written as $U(\sigma) = -\sum_{i=1}^N \sum_{j \in S_i} J_{ij}\sigma_i\sigma_j$, where S_i is the set of i's neighbours, and $J_{ij} \in \mathbb{R}$ describes the interaction strength between vertices i and

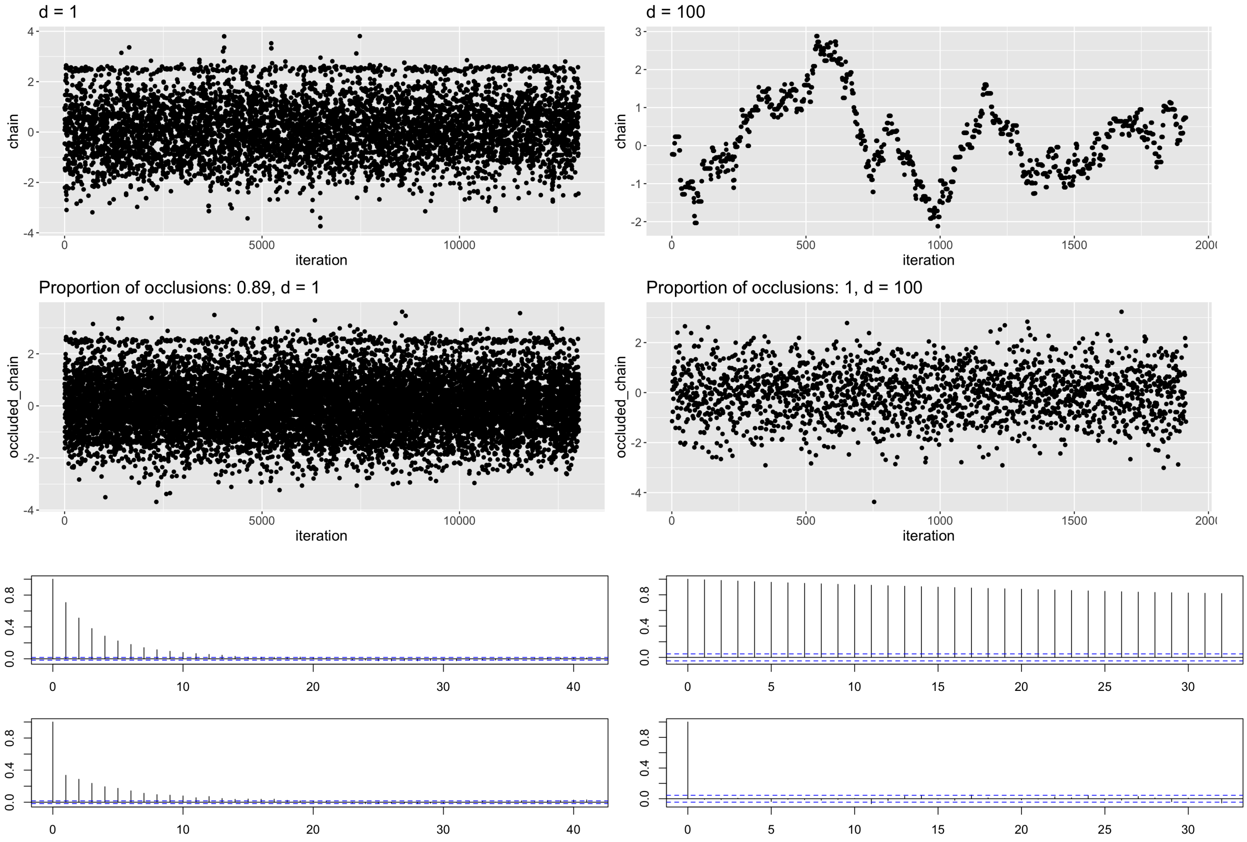

Figure 4.5: Results from the $d = 1$ case of sampling from (4.10) (left) and the $d = 100$ case (right). The top row shows the first component of the RWM chains, the next row shows the first component of $\{\mathbb{1}\{S_t = 0\}X_t + \mathbb{1}\{S_t = 1\}Y_t\}_{t=1}^n$, the next row shows autocorrelation function plots of the first component of the RWM chains, the bottom row shows autocorrelation function plots of the first component of $\{\mathbb{1}\{S_t = 0\}X_t + \mathbb{1}\{S_t = 1\}Y_t\}_{t=1}^n$.

j. Here π is the distribution over spin configurations and is defined with

$$\pi(A) := \sum_{\sigma \in A} \frac{\exp(-\beta U(\sigma))}{Z(\beta)} \tag{4.11}$$

for all subsets $A \subseteq \{-1, 1\}^N$, where $\beta \in [0, \infty]$ is the inverse temperature and $Z(\beta) := \sum_{\sigma \in \{-1,1\}^N} \exp(-\beta U(\sigma))$ is the so called 'partition function' i.e. the normalising constant. We assume that $J_{ij} > 0$ for all $i, j \in [N]$ and hence that the system is ferromagnetic. For such a model neighbouring spins align since doing so decreases the potential energy of the system. We also assume that $J_{ij} = 1$ for all $(i, j) \in E$. Note that for all $A \subseteq \{-1, 1\}^N$ we have that $\pi(A) = \pi(-A)$. Therefore any sampler on π should spend an equal amount of time in A and $-A$ in the large sample limit.

If the temperature β^{-1} is very low, the strength of alignment between neighbouring spins is very high. Therefore in this regime, if the graph is sufficiently well connected, the sampling algorithm should spend most of its time in configurations where a large proportion of the spins are aligned. It should spend about half the time with the overwhelming majority of the spins aligned in the positive direction, and half of the time with the overwhelming majority of the spins aligned in the negative direction.

If the temperature is very high, the strength of alignment is greatly reduced. Even if the the graph is well-connected a sample from the model at high temperature will have a significant proportion of both -1 and 1 spins.

A quantity of interest in the Ising model is the *average magnetisation*. It is defined as

$$M := \mathbb{E}_\pi \left[\frac{1}{N} \sum_{i=1}^{N} \sigma_i \right] \tag{4.12}$$

and describes the average spin across the entire graph measured in a typical configuration. Since the expression in the expectation is an odd function of σ and π is even, we are able to exactly calculate that $M = 0$ for all graphs, at all temperatures. This allows us to objectively measure the output of our sampling algorithms, and of the occlusion process.

Algorithm 4.2 Metropolis algorithm applied to the Ising model

inputs: Chain length n, Initial state $\sigma^{(0)} \in \{-1,1\}^N$, matrix of interaction strengths $J \in \mathbb{R}^{N\times N}$, inverse temperature $\beta \in [0,\infty]$.
outputs: A π-invariant Markov chain $\{\sigma^{(t)}\}_{t=1}^{n}$, where π is as in 4.11.

```
for $t \in [n]$ do
  1. Sample a vertex $I \sim \texttt{Uniform}[N]$.
  2. Let $\sigma^{(t)}_{\texttt{prop}}$ be $\sigma^{(t-1)}$ with a sign change in the $I$th vertex.
  3. With probability
       $$\alpha\left(\sigma^{(t-1)} \to \sigma^{(t)}_{\texttt{prop}}\right) := \min\left\{1, \exp\left(-\beta U\left(\sigma^{(t)}_{\texttt{prop}}\right) + \beta U\left(\sigma^{(t-1)}\right)\right)\right\}$$
     set $\sigma^{(t)} = \sigma^{(t)}_{\texttt{prop}}$ otherwise set $\sigma^{(t)} = \sigma^{(t-1)}$.
```

Sampling from the Ising model: the Metropolis algorithm The state space has 2^N configurations rendering $Z(\beta)$ very difficult to calculate for large N. For this reason practitioners use MCMC methods to sample from the Ising model. One such method is the Metropolis algorithm [Metropolis et al. 1953]. Originally conceived to sample from the two-dimensional hard-disk model, the Metropolis algorithm can also be applied to the Ising model. It selects a vertex at random and proposes switching the sign of the spin at that vertex. The switch is accepted with the usual Metropolis acceptance probability. For an exact description of the process see Algorithm 4.2.

Say we apply the Metropolis algorithm to the low temperature setting: when it has reached equilibrium it will likely be in a region of the state space with most of the spins aligned. After this point most of the proposal spin flips are rejected because they are in conflict with the spins of their neighbours. Therefore the chain will spend the vast majority of time in this region of the state space, instead of only half the time. Also any estimate we derive from the chain will have a high variance due to the size of the autocorrelations.

In the high temperature regime the strength of alignment between neighbouring spin sites will only be weak. Therefore the probability of accepting the individual spin flips proposed by the Metropolis algorithm will be higher and the Markov chain generated will exhibit lower autocorrelations than in the low temperature regime.

Algorithm 4.3 Wolff algorithm applied to the Ising model.

inputs: Chain length n, Initial state $\sigma^{(0)} \in \{-1,1\}^N$, matrix of interaction strengths $J \in \mathbb{R}^{N\times N}$, inverse temperature $\beta \in [0,\infty]$, aligned neighbour function AN
outputs: A π-invariant Markov chain $\{\sigma^{(t)}\}_{t=1}^{n}$, where π is as in 4.11

```
for $t \in [n]$ do
  1. Sample a vertex $I \sim \mathrm{Uniform}[N]$.
  2. Initialise the cluster $C = \{I\}$, a set of visited vertices $\mathcal{V} = \{I\}$, a set of
     aligned neighbours which have not been previously visited $\mathcal{N} = \mathsf{AN}(\{I\}) \setminus \mathcal{V}$,
     and a boolean ‘neighbours’ that is true if $\mathcal{N}$ is nonempty and false
     otherwise.
  3. while ‘neighbours’ do
     for $j \in \mathcal{N}$ do
         With probability $1 - \exp(-2\beta J)$ append $j$ to $C$ and append $j$ to $\mathcal{V}$.
     Set $\mathcal{N} = \mathsf{AN}(C) \setminus \mathcal{V}$ and if $\mathcal{N} = \varnothing$ set ‘neighbours’ to false
  4. Set $\sigma^{(t)} = \sigma^{(t-1)}$ and flip all the signs of the spins at the vertices in the
     cluster $C$ in $\sigma^{(t)}$.
```

Sampling from the Ising model: the Wolff algorithm In order to remedy this problem algorithms have been conceived that switch the signs of spins at a cluster of vertices in the graph in a single step of the Markov chain. This is how the Wolff algorithm works [Wolff 1989]. Specifically we select a vertex at random as the starting vertex in a cluster. We then add its neighbours that have a similar sign to the cluster each with probability $1 - \exp(-2\beta J)$. We add their aligned neighbours (so long as they have not been considered before) in a similar fashion, and their neighbours, etc. etc. until there are no unvisited aligned neighbours to consider. Once we have built a cluster in this way, we deterministically flip all of the spins inside the cluster. In order to formally describe the algorithm in 4.3, we define the aligned neighbour function AN : $\mathcal{P}(V) \to \mathcal{P}(V)$ which takes a subset of vertices of all the same sign spin, and outputs all those neighbours of the subset which are aligned.

In an above paragraph we explained why the Metropolis algorithm would work poorly in the low temperature setting. Let's say we use the Wolff algorithm instead. After reaching equilibrium, if the graph is sufficiently connected, most of the spins will be aligned. The value of the β parameter will dictate that the cluster inclusion probability $1 - \exp(-2\beta J)$ will be high. The number of aligned visitors will also be high. Therefore the Wolff algorithm will build large clusters whose spins it will flip deterministically, allowing it to traverse large distances in the state space in a single step.

In the high temperature regime the strength of alignment between spins is weaker. Hence in equilibrium the size of aligned clusters will generally be smaller, and so the individual moves generated by the Wolff algorithm will involve a smaller amount of spin flips. Therefore in the high temperature regime we expect the Wolff algorithm to exhibit higher autocorrelation than in the low temperature regime.

An efficiently simulable variational approximation to the Ising model In order to simulate the occlusion process described in Algorithm 4.1 we need access to a variational approximation to the Ising model which we can easily sample from. To do this we form a partition of $V = \bigcup_{i=1}^{k} V_i$ such that k is small, and define a new graph $(\tilde{V}, \tilde{E})$ where $|\tilde{V}| = k$ and $(i', j') \in \tilde{E}$ when there exists an edge in the original graph $(i, j) \in E$ such that $i \in V_{i'}$ and $j \in V_{j'}$. The node clusters V_i therefore form the nodes in $\tilde{V}$. See Figure 4.6 for an example of two graphs: (V, E) and $(\tilde{V}, \tilde{E})$.

We extend the state space to include $\mu \in \{-(1-\epsilon), 1-\epsilon\}^k$ for a given $\epsilon > 0$ such that each μ_i serves as the mean of the spins in V_i. We let μ behave according to the dynamics of an Ising model on $(\tilde{V}, \tilde{E})$ and sample the spins within the clusters V_i independently with mean μ_i. Therefore we dictate that

$$\tilde{U}(\mu) := -\sum_{i=1}^{k} \sum_{j:(i,j)\in\tilde{E}} \tilde{J}_{ij} \mu_i \mu_j$$

where each $\tilde{J}_{ij} \in \mathbb{R}$ possibly depends on the number of edges between subgraphs i and j. Since k is small we can calculate the normalising constant

$$\tilde{Z}(\tilde{\beta}) := \sum_{\mu \in \{-(1-\epsilon), 1-\epsilon\}^k} \exp(-\tilde{\beta} \tilde{U}(\mu))$$

where $\tilde{\beta} \in [0, \infty]$ is an inverse temperature hyperparameter. As mentioned before, we sample the spins within each subgraph independently conditional on their means, which are given by μ. We choose a Rademacher distribution such that

$$q(\sigma \,|\mu) := \prod_{i=1}^{k} \prod_{s=1}^{|V_i|} \left(\frac{1+\mu_i}{2}\right)^{\frac{1+\sigma_s^{(i)}}{2}} \left(\frac{1-\mu_i}{2}\right)^{\frac{1-\sigma_s^{(i)}}{2}}$$

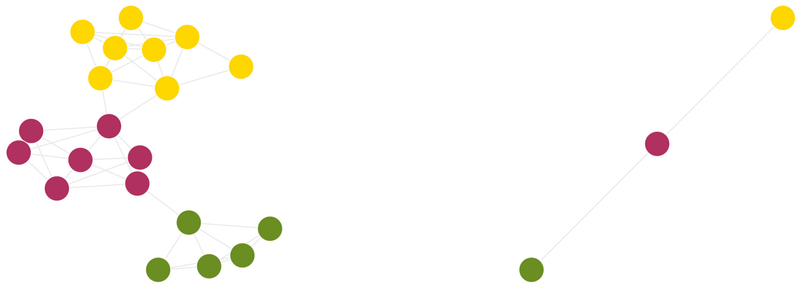

Figure 4.6: Left: (V, E), right: $(\tilde{V}, \tilde{E})$. The colours correspond to the V_i's in the original (V, E) that get collapsed to nodes in $(\tilde{V}, \tilde{E})$. Where there exist any edges between two V_i's in (V, E), there is an edge between the corresponding nodes in $(\tilde{V}, \tilde{E})$.

where $\sigma_s^{(i)} \in \{-1, 1\}$ is the sth spin in the ith cluster. Therefore the full variational distribution has density

$$q(\sigma) := \sum_{\mu \in \{-(1-\epsilon), 1-\epsilon\}^k} q(\sigma \,|\mu) \tilde{Z}(\tilde{\beta})^{-1} \exp(-\tilde{\beta}\tilde{U}(\mu))$$

To sample from Q we simply sample μ according to the Ising distribution defined by $\tilde{U}$ and then sample σ conditionally on μ. To do this we need to take an exact sample of μ. For this we can simply calculate the probabilities of all points in $\{-(1-\epsilon), 1-\epsilon\}^k$ since we have access to $\tilde{Z}(\tilde{\beta})$.

Experiment setup In order to test the occlusion process in a full range of settings, we record its performance on the Ising model at low and high temperatures on graphs of different sizes and connectivities, and we compare it with the Metropolis and the Wolff algorithm.

To generate the underlying graph we sample from an N-vertex stochastic block model [C. Lee and Wilkinson 2019] with k communities, where the community sizes are sampled uniformly from size k partitions of $\{1, ..., N\}$. For the matrix of edge probabilities $A \in \mathbb{R}^{k \times k}$ we have $A_{ii} = 0.8$ for all $i \in [k]$ and $A_{ij} = 0.01$ for all $i \neq j$ and $i, j \in [k]$. The left side of Figure 4.6 shows such a graph with $k = 3$ communities and $N = 20$ vertices.

As stated in the introduction of the model, we set $J_{ij} = J = 1$ for all $i, j \in [n]$. We set $\beta = 1$ to emulate a low temperature setting, and $\beta = 0.01$ to emulate a high temperature setting. For the variational approximation defined in Section 4.1.5.2 we define the clusters V_i as the communities of the underlying graph. In practice we will not have access to the actual latent cluster structure of the graph, and therefore some clustering algorithm must be used to find these. We set $\tilde{J}_{ij}$ to the number of connections between clusters V_i and V_j for all $i, j \in [k]$ and $\tilde{\beta} = 0.5 \times \beta$ in all temperature settings. We set $\epsilon = 0.1$ in the $\beta = 1$ low temperature setting and $\epsilon = 0.9$ in the $\beta = 0.01$ high temperature setting.

For the number of regions we set $R = 3$. To choose the constants $C_1 < C_2$ which define the regions (see Section 4.1.4.1) we run the Wolff algorithm for 20 seconds from a random $\sigma^{(0)} \sim \text{Uniform}\{-1, 1\}^N$ initialisation. We set C_1 to be the median from $\{d\pi/dQ(\sigma^{(t)})\}_{t\geq 1}$ and C_2 to be the maximum.

To initialise the Markov chains generated by the Metropolis and Wolff algorithms we use the final state in a Markov chain generated by the Swendsen-Wang algorithm. In [Huber 1999, Theorems 1 and 2] one can find conditions under which the Swendsen-Wang chain couples with probability $\geq 1/2$ for a specified chain length. We therefore use these results to run a Swendsen-Wang chain until the probability of being uncoupled is ≤ 0.0001 for each model we choose to sample from.

For each pair (k, N) with $k \in \{2, 5, 10\}$ and $N \in \{20, 50, 100\}$ we run 15 replications of the Metropolis and Wolff algorithms for 20 seconds each. Each time we use one of these algorithms, we also run the occlusion process defined in Algorithm 4.1 so that we can compare their results. Specifically, we compare their estimates of the expected magnetisation as defined in (4.12).

Experiment results In Figure 4.7 we show three graphs all from the same set of results which are generated according to the setup described above. Here we display the results in the $k = 5$ case. For the $k \in \{2, 10\}$ cases, refer to Appendix 7.3.2. Each graph has four subgraphs across the various temperatures and algorithms we use to sample. The ‘normal’ in the legend refers to data from the non-occluded Markov chain. The bottom graph compares the estimates of the expected magnetisation between the occluded process and the Metropolis and Wolff algorithms. The true value of the expected magnetisation is 0. The top left graph shows the lag 1 autocorrelation coefficient of the magnetisation over the course of the Markov chains. The top right graph shows the number of occluded states as a proportion of the total number of states from the Markov chain sampling algorithms. In the notation of Section 4.1.4.2 this is $(N_1 + \cdots + N_R)/n$.

As expected the Wolff algorithm excels in the low temperature setting whereas the Metropolis algorithm fares very poorly due to the fact that in each instance it never strays from its initial position. In the high temperature setting the variance of the estimates from the Wolff algorithm are lower in general than those from the Metropolis chain, but the difference is not as stark as in the low temperature setting. An explanation as to why the Wolff algorithm does so well in the low temperature setting is offered by the graphs of the lag 1 autocorrelation coefficients (top right). One can clearly see that some of these coefficients are close to -1, explaining why the variance of the resulting estimates is very low.

As concerns the performance of the occluded chain versus the Metropolis and Wolff samplers: we can see that in the high temperature setting, in every case the variance of the occluded estimator is much lower than those of the Markov chains. The fact that the lag 1 autocorrelation coefficients of the occluded process are so close to zero, combined with the high proportions of occlusion events as seen in the top level of the top right, explain this fact.

In the low temperature setting the story is more mixed. If we compare the performance of the occluded estimator with the Metropolis algorithm, we can see that for $N = 20$ and 50 the occluded estimator offers a reduction in variance. This is explained by the corresponding reduction in the lag 1 autocorrelation coefficients and the high occlusion proportions. However the low occlusion proportion in the $N = 100$ case means that the occluded chain and the Metropolis chain are basically the same, and so the occluded estimator offers no increase in performance. In this case the Markov chain does not move, and so the lag 1 estimator breaks, as seen in the graph in the top left.

Comparing the occluded estimator with the Wolff algorithm in the low temperature regime shows that in the $N = 50$ case using the occluded estimator actually decreases the performance by increasing the bias and the variance. This is explained by the fact that the occlusion events tend to *de*correlate subsequent states in the process, whereas the Wolff algorithm produces *anti*correlated states. Hence the decorrelations are undesirable in this case. Again, as with the Metropolis algorithm, the number of occlusions in the $N = 100$ case is very few, and there is negligible difference between the performance of the occlusion process and the Wolff algorithm.

We would like to note that we have not tuned the parameters of the variational distribution for any of the models presented here. Therefore for a better tuned variational distribution, we would expect the occlusion proportions to be higher, and so at least in the low temperature $N = 100$ case achieve a reduced lower variance of the occluded estimator compared to the Metropolis algorithm.

Satisfaction of the theoretical conditions for variance reduction Here we claim that, in the high temperature setting, we satisfy condition 2. from Proposition 58, such that the variance reductions seen in Figure 4.7 verify the theory.

Clearly the measure defining the Ising model (4.11) is even in its arguments and the magnetisation (4.12) is odd. That the regions satisfy $\mathsf{X}_i = -\mathsf{X}_i$ follow from the fact that the Radon-Nikodym derivative is such that $d\pi/dQ(\sigma) = d\pi/dQ(-\sigma)$ for all $\sigma \in \{-1, 1\}^N$. These three facts satisfy the conditions outlined in Example 59 such that $f \equiv \overleftarrow{\pi} f + \pi(f)$. That the Markov kernel is positive when applied to the magnetisation functional is evidenced by the lag-1 autocorrelation coefficients in the high temperature settings of Figure 4.7. The last criterion for variance reduction we need is for $\hat{\mu}_{\text{occ}} = \hat{\mu}_{\text{ideal}}$. This is evidenced by the occlusion proportion graphs in Figure 4.7, which imply that $\alpha \equiv 1$ at high temperature.

Overall this suggests that in the high-temperature setting, condition 2. of Proposition 58 is satisfied, and so the theory suggests that $\text{Var}(\hat{f}_{\text{occ}}) \leq \text{Var}(\hat{f})$. This is verified by the variance reductions seen at high-temperature in Figure 4.7.

4.1.6 Discussion

4.1.6.1 Summary

We have introduced the occlusion process which sits on top of an existing Markov kernel K. It produces unbiased estimates of expectations under the equilibrium distribution π of K. The occlusion process is designed to reduce the variance of the estimates produced solely by K at no additional compute-time cost. It uses a variational distribution Q to do this.

We define the process in the Markov kernel (4.5), showing that it is unbiased and stating its variance in Proposition 60. In Section 4.1.3 we show that it inherits properties such as an LLN (Theorem 63), convergence in a normed function space (Theorem 65), and geometric ergodicity (Theorem 72) from K. In Section 4.1.4 we explain how to simulate the occlusion process at no additional computational cost in terms of wall-

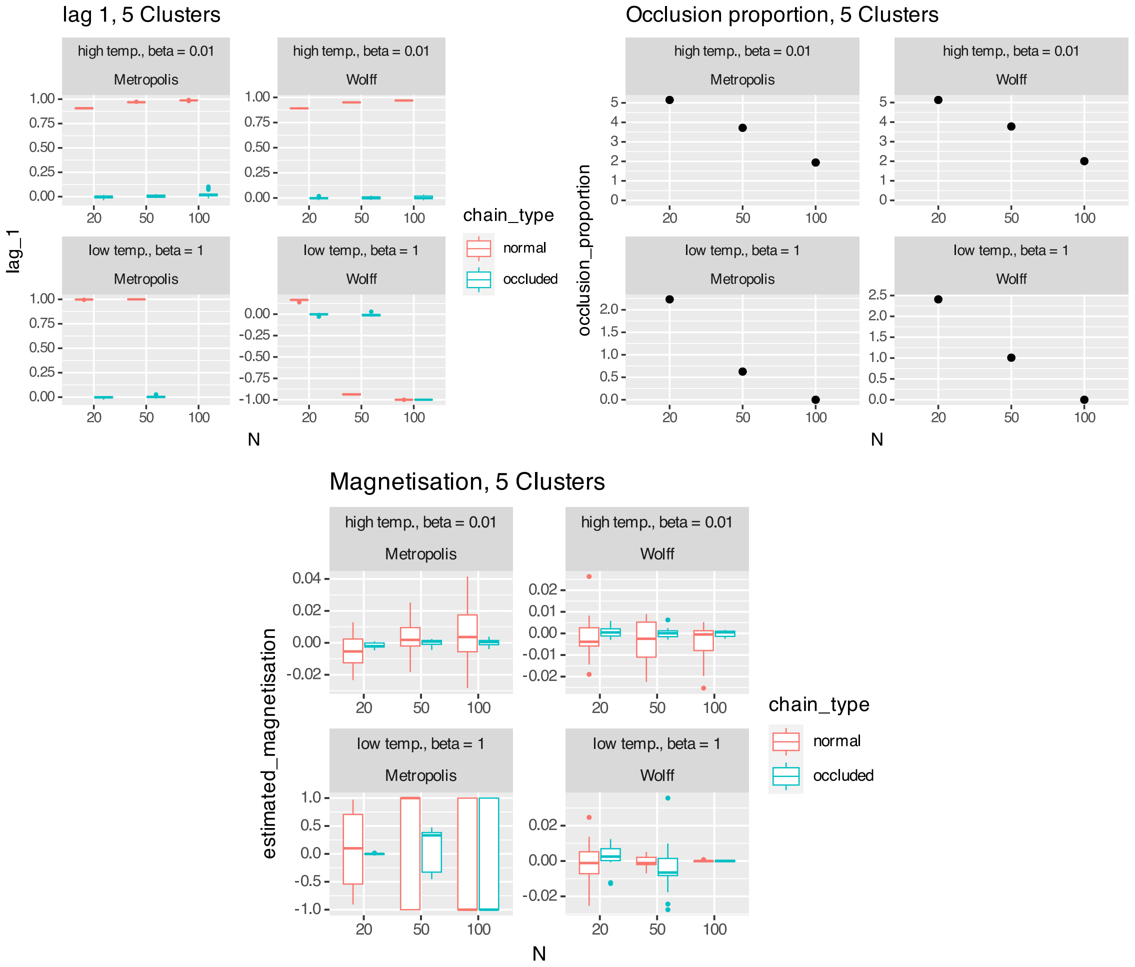

Figure 4.7: Three graphs comparing the performance of the occlusion process with the Metropolis and Wolff algorithms on the Ising model at a variety of temperatures, for a variety of graph sizes. In every case the horizontal axes show the number of vertices N in the graphs. Bottom: the vertical axes denote the algorithm's estimates of the expected magnetisation. Top left: the vertical axes denote the lag 1 autocorrelation coefficient of the magnetisation over the states produced by the algorithms. Top right: the vertical axes show the number of samples from the π_i's in Algorithm 4.1 divided by the number of states in the Markov chain n. We magnify the estimated magnetisation plot for ease of comprehension, the top two plots then help to explain the phenomena in the bottom plot.

clock time. In Section 4.1.5 we present two numerical experiments so as to inspect the empirical properties of the occlusion process, and to compare its performance against the Markov chains generated by K. The first experiment in Section 4.1.5.1 compares the ability of K with the occlusion process to sample from a bimodal Gaussian mixture, for which we have a good approximation Q of one of the mixture components. The results show that the occlusion process is able to effectively decorrelate subsequent steps in the Markov chain generated by K. In Section 4.1.5.2 we define the Ising model on an arbitrary graph, and introduce the Metropolis (Algorithm 4.2) and Wolff (Algorithm 4.3) algorithms which are used to form Markov chains with which we form estimators for this model. Since the occlusion process uses a variational distribution Q of the Ising model we propose one in Section 4.1.5.2. We run the Metropolis and Wolff at various temperatures on various graphs, and show that the occlusion process produces estimators of reduced variance, apart from in the low temperature case with the Wolff algorithm. This is because, in such a setting the Wolff algorithm produces states which are anticorrelated, whereas the occlusion process produces states which are decorrelated. In Section 4.1.5.2 we argue that in the high temperature setting, we satisfy the conditions for variance reduction, which explains the variance dominance in the experimental results.

4.1.6.2 Extensions

There are numerous interesting extensions to the work presented here. On the theoretical side, it would be interesting if we could establish conditions under which the variance of the occluded estimator 4.6 is dominated by the variance of the estimator produced by the Markov kernel K (where K is positive). In the numerical experiments we clearly see variance dominance in multiple cases, so it would be useful to theoretically establish this phenomenon. In general it would be interesting to find the minimum variance unbiased estimator of an expectation μ under π given an array of samples $\{X_t\}_{t=1}^{n}$ from a π-invariant Markov chain and R collections of samples $\{\{Y_{ij}\}_{j=1}^{N_i}\}_{i=1}^{R}$ with $Y_{ij} \sim \pi_i$ for all $j \in [N_i]$ and $i \in [R]$. For example: in the embarrassingly parallel process outlined in Algorithm 4.1 the Y_{ij} samples occlude uniformly at random the appropriate X_t samples. Is there some allocation scheme that is less random, achieves a lower variance, but is still provably unbiased? We might also not want to throw away the samples X_t from the Markov chain upon occlusion. Is there some estimator then that uses all the samples from the Markov chain, as well as the Y_{ij} samples, in an unbiased way?

On the practical side of things there is a plethora of ways in which we could enhance the occlusion process. One way is to learn the optimal tuning parameters of the variational distribution Q online as the process runs. Another way is to use the samples from the π_i's to inform the Markov chain generated by K as in the equi-energy sampler [Kou, Zhou, and Wong 2006]. We could also use enhanced versions of the rejection samplers in Algorithm 4.1 such as the squeeze method [Marsaglia 1977].

Chapter 5

High dimensional adaptive MCMC with reduced computation complexity

We propose an adaptive MCMC method that learns a linear preconditioner which is dense in terms of its off-diagonal elements but sparse in terms of its parametrisation. Due to this sparsity we achieve a computational complexity of $O(d)$ compared with $O\left(d^2\right)$ for existing preconditioners that can also capture correlations in the target. Diagonal preconditioning also has $O(d)$ computational complexity, but is known to fail in the case that the target distribution is highly correlated, see section 3.5. Our preconditioner is constructed using eigeninformation about the target covariance which we infer using online Principal Components Analysis on the MCMC chain. It is composed of a diagonal matrix and a product of carefully chosen Householder reflections. On various numerical tests we show that it outcompetes diagonal preconditioning in terms of absolute performance, and that it outcompetes the traditional dense preconditioning in terms of time normalised performance

The notation in this chapter can be found to be defined in section 7.1.5.

5.1 Motivation

When applied to the problem of Bayesian inference, modern MCMC methods must adapt to accommodate the forms of the data that shape the posterior. When the data is high-dimensional, in the sense that each data point lives in a high-dimensional space, the parameter for which we construct the posterior is often also high-dimensional. In the case of Generalised Linear Models, the dependency of the posterior distribution on the data $X \in \mathbb{R}^{n \times d}$ enters in the form $X\theta$ where $\theta \in \mathbb{R}^d$. Therefore if the data is high dimensional in the sense that d is large we will have to construct MCMC algorithms to operate within high dimensional spaces.

In chapter 3 we examined the effect of linear preconditioners on the condition number of the target distribution. One of the preconditioners we focused on was constructed using the target covariance Σ_π. We showed that the O-U process preconditioned with Σ_π had an optimal spectral gap. We also showed that the Hessian of the potential of the target could be localised around Σ_π^{-1} in such a way that we could use the theory we developed to estimate the effect of using a covariance-based preconditioner on the condition number. Therefore in this chapter we develop an adaptive algorithm which learns spectral information about the target covariance, in order to use this information to construct a preconditioner.

In a high-dimensional context learning and using a fully dense preconditioner, such as the covariance, in the manner of [Haario, Saksman, and Tamminen 2001] and 2.17 is infeasible due to the $O\left(d^2\right)$ computational complexity of both the learning update, and the way in which the preconditioner is used in the Markov kernel. This is only partially solved by using the diagonals of the target covariance as we demonstrated in 3.5, since doing so is inadvisable on target distributions that have large correlations between many dimensions. Therefore we seek to construct a preconditioner that is dense in terms of its off diagonal elements but sparse in terms of its parametrisation in the hopes that both the learning step and the Markov kernel sampling step in algorithm 2.1 can be executed in $o\left(d^2\right)$ computational complexity. This is a difficult problem since the preconditioner construction must give a $o\left(d^2\right)$ complexity of, say, matrix-vector multiplication but we can't just arbitrarily constrain the parameterisation without also taking into account how it's going to be updated in the learning step.

The method we propose learns the $m \in [d]$ eigenvectors of the target covariance associated with the top m eigenvalues. It also learns the target scales along these directions, which are just the top m eigenvalues of the target covariance, along with some additional scale information. It uses all of this eigeninformation to construct a preconditioner that aims to reduce $\|\Sigma_\pi\|_2$ by isotropising the top m eigenvalues. The preconditioner is constructed such that the matrix multiplication, matrix inversion, and matrix square root operations are $O\left(m^2d\right)$. That the learning step also incurs $O\left(m^2d\right)$ operations gives an $O\left(m^2d\right)$ total iteration complexity of the adaptive algorithm.

5.2 Existing practices

One possible sparse parameterisation of a dense matrix is the 'diagonal plus low rank' form: $D + v_1v_1^T + \cdots + v_mv_m^T \in \mathbb{R}^{d\times d}$ where $m \ll d$ and $D \in \mathbb{R}^{d\times d}$ is a diagonal matrix. This parametrisation takes $(m+1)\,d$ scalar parameters. With this form, matrix vector multiplication is $O\left(md\right)$ and calculations using its inverse are made much simpler using the Woodbury formula. We will often need to take the Cholesky factor of a positive definite $\Sigma \in \mathbb{R}^{d\times d}$ to sample from $\mathcal{N}\left(0, \Sigma\right)$. However, if Σ is in the diagonal plus low rank form we have that $\sqrt{D}\xi + V\zeta \sim \mathcal{N}\left(0, D + VV^T\right)$ where $\xi \sim \mathcal{N}\left(0, \mathbf{I}_d\right)$, $\zeta \sim \mathcal{N}\left(0, \mathbf{I}_m\right)$ and $V \in \mathbb{R}^{d\times m}$ has v_i as its ith column, for all $i \in [m]$ [Miller, Foti, and Adams 2017; Ong, Nott, and Smith 2018].

The use of preconditioning matrices with this structure has history in numerical linear algebra, with a prominent example being the Limited-Memory BFGS (L-BFGS) Quasi-Newton method for Optimisation; see e.g. [Morales and Nocedal 2001] and [Helmberg 2024]. In sampling [Yichuan Zhang and Sutton 2011] construct a diagonal plus low rank estimate of the inverse Hessian of the potential using L-BFGS. They maintain the Cholesky factor of their estimate using a mechanism that takes $O\left(m^2d\right)$. Their approach is fundamentally different from ours since the Hessian of the potential varies across the state space whereas we aim to estimate the eigeninformation from a single matrix (the target covariance) which is fixed across the state space. Therefore in optimality our preconditioner is fixed whereas theirs varies. Since they use the inverse Hessian of the target potential to provide the covariance matrix for their proposal distributions, their method is specifically designed for cases where this potential is strictly convex. It is unclear how their method would perform when this condition is violated, whereas the only assumption we require for our method is for the target covariance to exist and be elementwise finite.

[Langmore et al. 2020] define a new condition number for HMC that uses the spectrum $\lambda_1^2 \geq \lambda_2^2 \geq \cdots \geq$

λ_d^2 of the target covariance in the following way:

$$\kappa_{\text{HMC}} := \left(\sum_{i=1}^{d} \left(\frac{\lambda_1}{\lambda_i} \right)^4 \right)^{1/4}$$

Under certain assumptions on the step-size, [Langmore et al. 2020] show that this condition number is proportional to the number of leapfrog steps needed to achieve a stable acceptance rate. They note that κ_{HMC} is inflated when the target covariance has a few large eigenvalues and many small ones. They then learn argmin$KL\left(\mathcal{N}\left(0, LL^T\right) \| \pi\right)$ with $L \in \mathbb{R}^{d\times d}$ restricted to be diagonal plus low rank. They do this is the case that $\pi = \mathcal{N}\left(0, L_\pi L_\pi^T\right)$ where $L_\pi \in \mathbb{R}^{d\times d}$ is a circulant matrix such that $L_\pi L_\pi^T$ has a few large eigenvalues and many small. They found that their preconditioner did a better job at reducing κ_{HMC} when m (the number of terms in the low rank expansion) exceeded the number of large eigenvalues in the circulant matrix. Since they use the reverse KL, they can learn their preconditioner off-line i.e. with no information from an MCMC algorithm.

In sampling the other notable work that exploits a sparsely parametrised preconditioner is [Wallin and Bolin 2018]. The authors note that the sparsity structure of the precision matrix of the target can be detected via partial correlation information. The authors use conditional dependence as a proxy for partial correlation and pose a pre-processing routine to detect the presence of a conditional independence between the target dimensions (but not necessarily a dependence). They then calculate the estimated sparsity structure of the Cholesky factor of the precision, using a reordering of the dimensions to make the factor sparser if desired. Then, given a sparsity structure they propose routines to calculate the Cholesky factor of $(\Sigma_{n+1})^{-1}$ from $(\Sigma_n)^{-1}$ and X_n where Σ_n is a chain based estimate of the target covariance and X_n is the latest state in the chain. They conceive the adaptive algorithms paMHRW (precision adaptive Metropolis-Hastings Random Walk) and paMALA (precision adaptive MALA) which are RWM and MALA with precision adaptive elements. Define $\{A_j\}_{j=1}^d$ with A_j the set of estimated non-zero indices of the jth column of the Cholesky factor of the target precision. Then one iteration of paMALA has $O\left(d + \sum_{j=1}^d |A_j|^2\right)$ computational complexity. Therefore this complexity is $O(d)$ at best and $O\left(d^3\right)$ at worst.

5.3 An eigen-informed, sparsely parametrised preconditioner

We propose a preconditioner that is constructed using information from the target covariance $\Sigma_\pi \in \mathbb{R}^{d\times d}$, specifically the top $m \in [d]$ eigenvalues and their associated eigenvectors. We show that this preconditioner is beneficial in the sense that it reduces the operator norm of the target covariance as motivated in section 5.1. We then construct a parametrisation for which matrix-vector multiplication is $O\left(m^2 d\right)$ and matrix inversion and Cholesky factorisation are easily computable.

A word on notation In chapter 3 a 'preconditioner' was a matrix $L \in \mathbb{R}^{d\times d}$ that described the transformation $Y = LX$ where $X \sim \pi$, whose desired function was to make it easier to sample from $\mathcal{L}(Y)$. That is still the desired function here, but instead we call L^{-1} the 'preconditioner', where L is as introduced in chapter 3. One can think of the preconditioners in chapter 3 as preconditioning the target, and the preconditioners here as preconditioning the algorithm, and we can switch between these ways of thinking with proposition 26. It is more fitting to talk about the algorithm-based formulation of the preconditioner here since we are developing an algorithm.

5.3.1 The optimal preconditioner

When constructing a sparsely parametrised preconditioner, we need a parameter space from which we can reconstruct a full preconditioner and in which we can plausibly learn. Fixing $m \in [d]$ we choose the spaces of m eigenvectors and m eigenvalues in which to learn, and from which we can fill in the rest of the eigeninformation to construct the full preconditioner. Define $\left\{\lambda_i^{(\pi)} : i \in [d]\right\}$ as the set of eigenvalues of the target covariance (in descending order) and $\left\{v_i^{(\pi)} : i \in [d]\right\}$ the corresponding normalised eigenvectors. We attempt to learn the preconditioner $L = QD \in \mathbb{R}^{d\times d}$ where $D = \text{diag}\left\{\sqrt{\lambda_1^{(\pi)}}, \sqrt{\lambda_2^{(\pi)}}, \ldots, \sqrt{\lambda_m^{(\pi)}}, 1, \ldots, 1\right\}$ is a diagonal matrix whose first m terms are the square roots of the top m eigenvalues of the target covariance. The matrix $Q \in O(d)$ has as its first m columns the associated ordered eigenvectors $\left\{v_1^{(\pi)}, v_2^{(\pi)}, \ldots, v_m^{(\pi)}\right\}$ and the remaining $d - m$ columns are chosen arbitrarily to make Q orthogonal. This preconditioner makes the MCMC algorithms it preconditions more efficient in the following sense:

Proposition 75. *Assume that the covariance $\Sigma_\pi := \text{Cov}_\pi(X)$ of the target distribution exists and is element-wise finite. Let $\tilde{\pi} \in \mathcal{P}(\mathcal{X})$ be the target distribution π after preconditioning with $L = QD$ with corresponding covariance $\Sigma_{\tilde{\pi}}$. Then $\|\Sigma_{\tilde{\pi}}\|_2 = \max\left\{\lambda_{m+1}^{(\pi)}, 1\right\}$.*

For a proof see section 7.2.5.1. If preconditioned with $L = QD$ the target will become more isotropic in general due to the fact that the scales along the directions $\left\{v_1^{(\pi)}, v_2^{(\pi)}, \ldots, v_m^{(\pi)}\right\}$ will all be 1 after preconditioning.

5.3.2 Construction and computation with Householder matrices

Construction Even though we have stated the optimal preconditioner, we must still provide a parameterisation of the preconditioner that can be learned and used within a Markov step in $o\left(d^2\right)$ computational complexity. Computations with the diagonal component D of the preconditioner are straightforward and clearly $o\left(d^2\right)$, and so we will focus on the construction of the orthogonal component Q here. Therefore assume that we have m orthonormal vectors $v_i \in \mathbb{R}^d$ for $i \in [m]$ which will serve as our estimates of the top m eigenvectors introduced in section 5.3.1. Given $v, w \in \mathbb{R}^d$ we define the Householder matrix $H(v \leftrightarrow w) \in O(d)$ with

$$H(v \leftrightarrow w) := \mathbf{I}_d - 2\frac{(v-w)(v-w)^T}{\|v-w\|_2^2}$$

We construct the orthogonal component of our preconditioner as $Q := Q_m$ where we define Q_m iteratively:

$$\begin{aligned} Q_1 &:= H(e_1 \leftrightarrow v_1) \\ \text{for } k \in \{2, \ldots, m\}, Q_k &:= H(Q_{k-1}e_k \leftrightarrow v_k) Q_{k-1} \end{aligned} \tag{5.1}$$

where $e_i \in \mathbb{R}^d$ is the ith canonical basis vector of $\mathbb{R}^d$. Therefore $Q_1, \ldots, Q_m$ can be seen as a series of Householder matrices which are notable for their ability to transform an arbitrary vector into another, and vice versa. Specifically $H(v \leftrightarrow w)$ is an orthogonal matrix with determinant -1 (because it is a reflection) such that $H(v \leftrightarrow w)\,v = w$ and $H(v \leftrightarrow w)\,w = v$. If u is perpendicular to both v and w then $H(v \leftrightarrow w)\,u = u$ because u is in the plane of reflection. See figure 5.1 for a visual representation of the action of a

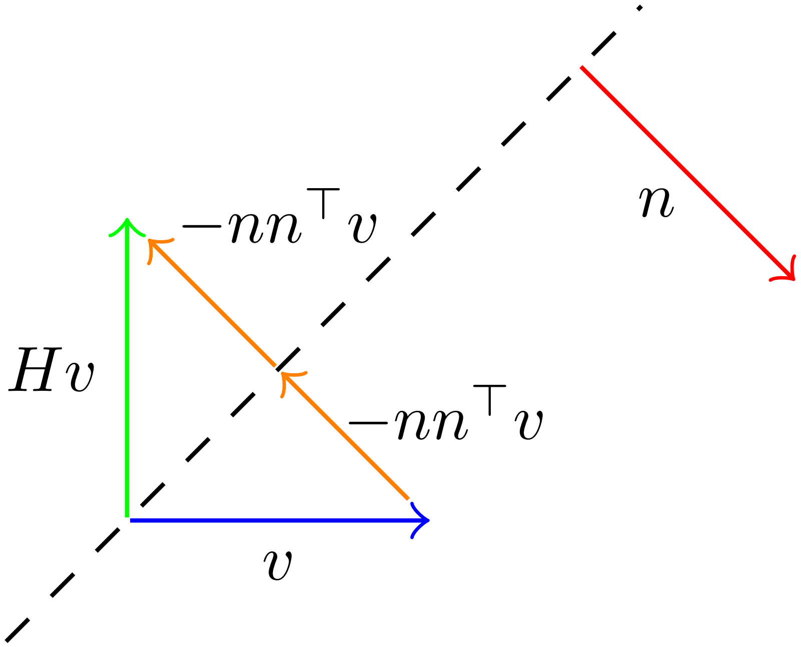

Figure 5.1: The action of $H = \mathbf{I}_d - 2nn^T$ on v. Let $n \in \mathbb{R}^d$ be a unit normal to a plane about which we would like to reflect $v \in \mathbb{R}^d$. Adding $-nn^Tv$ to v projects it onto the plane and adding another $-nn^Tv$ sends it to its reflection.

Householder matrix. A given Householder matrix $H \in O(d)$ will also have the nice properties that $H = H^T$ and $H^2 = \mathbf{I}_d$.

The idea behind the orthogonal component of the preconditioner is that it acts on e_k in the following way:

$$\begin{aligned} Q_m e_k &= H(Q_{m-1}e_m \leftrightarrow v_m)\dots H(Q_k e_{k+1} \leftrightarrow v_{k+1}) H(Q_{k-1}e_k \leftrightarrow v_k) Q_{k-1}e_k \\ &= H(Q_{m-1}e_m \leftrightarrow v_m)\dots H(Q_k e_{k+1} \leftrightarrow v_{k+1}) v_k \end{aligned}$$

The remaining Householder matrices are constructed such that $H(Q_{m-1}e_m \leftrightarrow v_m)\dots H(Q_k e_{k+1} \leftrightarrow v_{k+1}) v_k = v_k$.

Proposition 76. *Fix $m \leq d$. Given a set of m orthonormal vectors $\{v_1, \dots, v_m\}$ in $\mathbb{R}^d$ which constructs $Q = Q_m$ as in 5.1 we have that $Qe_i = v_i$ for all $i \in [m]$.*

For a proof, see section 7.2.5.2. The proposition implies that we have a way of building a $Q \in O(d)$

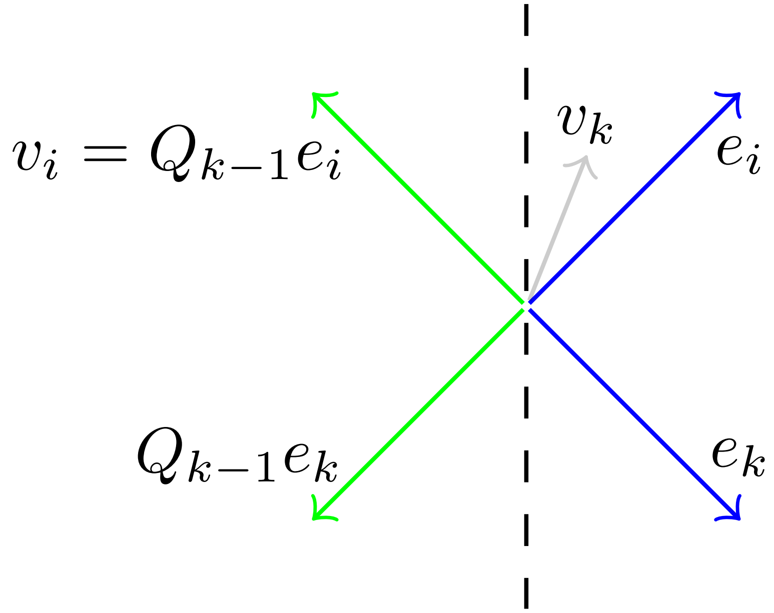

Figure 5.2: A visual explanation of why $Q_k e_i = v_i$ for all $i < k$. By construction we have that $Q_k e_i = H\left(Q_{k-1}e_k \leftrightarrow v_k\right)Q_{k-1}e_i$ and so by the properties of Householder matrices if $Q_{k-1}e_i$ is perpendicular to both $Q_{k-1}e_k$ and v_k we have that $Q_k e_i = Q_{k-1}e_i$ which is just v_i by inductive hypothesis. Clearly $Q_{k-1}e_i$ is perpendicular to $Q_{k-1}e_k$ since the two vectors are just canonical basis vectors transformed by a Householder matrix, which is orthogonal. This transformation is shown by the reflection of the blue vectors to the green vectors through the dotted line in the figure. That $Q_{k-1}e_i$ is perpendicular to v_k is evident because $Q_{k-1}e_i = v_i$ by inductive hypothesis, and the fact that v_i is perpendicular to v_k.

whose first columns are $\{v_1, ..., v_m\}$. In other words, we have a way of extending a set of m orthonormal vectors to an orthonormal basis of $\mathbb{R}^d$. Note that every matrix Q_k for $k \in [m]$ in the construction of Q_m has the property that $Q_k e_1 = v_1, ..., Q_k e_k = v_k$ by construction. It is because of this fact that proposition 76 holds true. That $Q_k e_k = v_k$ is obviously true. That $Q_k e_i = v_i$ for $i < k$ is true is explained in figure 5.2. This means that the first k columns of Q_k are the first k columns of Q_j for $j \in \{k+1, ..., m\}$.

5.3.2.1 Computation

The full preconditioner is then $L = QD$ with Q constructed as in the section above, and D a diagonal matrix. Matrix-vector multiplication with a diagonal matrix is $O\left(d\right)$ and Matrix-vector multiplication with a Householder matrix is $O\left(d\right)$. However each Q_k contains Q_{k-1} for all $k \in \{2, \ldots, m\}$ and so to calculate $Q_k v$

we must calculate $Q_{k-1}v$ and then $e_k^T Q_{k-1}\left(Q_{k-1}v\right)$. For this reason, matrix vector multiplication with L is $O\left(m^2 d\right)$.

In practice we find that simply constructing Q_m, storing it as a matrix, and executing matrix-vector multiplication in the usual way is faster than the bespoke matrix-vector routines we wrote to exploit its construction. This is because in built matrix-vector multiplication routines are very well optimised. Therefore we use normal matrix-vector multiplication in our numerical experiments in section 5.3.6. Clearly if the dimension is high enough our bespoke $O\left(m^2 d\right)$ matrix-vector multiplications will be faster than the in built routines (which are $O\left(d^2\right)$).

Calculating the inverse $L^{-1} = D^{-1}Q^T$ is particularly simple since Q is the composition of Householder matrices, which are symmetric. Since we use MALA as the base MCMC kernel, this inverse needs calculating at every accept-reject step. This is why our algorithm achieves a lower time per iteration than those which use a dense preconditioner, even though we do dense matrix-vector multiplication. Sampling from $\mathcal{N}\left(0, LL^T\right)$ can be achieved by multiplying $\xi \sim \mathcal{N}\left(0, \mathbf{I}_d\right)$ by L, and so has the same computational complexity as matrix-vector multiplication.

5.3.3 Learning with online principal components analysis

5.3.3.1 Eigenvectors

Principal components analysis (PCA) is a method for learning the orthogonal directions across which data vary the most. Given a dataset $X \in \mathbb{R}^{n\times d}$ whose rows are identically distributed data points that have been centred, and a number of directions $m \in [d]$ we seek the optimal V^* which satisfies

$$\text{argmax}\left\{\text{tr}\left(V^T \Sigma_\pi V\right) : V \in \mathbb{R}^{d\times m}, V^T V = \mathbf{I}_m\right\}$$

where the empirical covariance $\hat{\Sigma} \propto X^T X$ is used to approximate Σ_π. Say $\hat{\Sigma}$ has eigendecomposition $\hat{Q}\hat{D}\hat{Q}^T$ where the diagonal elements of $\hat{D}$ are in decreasing order. Then the maximum is achieved when V^* contains the the first m columns of $\hat{Q}$. Therefore PCA is equivalent to learning the eigenvectors associated with the top m eigenvalues of the empirical covariance. Since PCA can be interpreted as a simple dimensionality reduction subroutine, it can be used on data with any kind of dependency structure. However, establishing guarantees for its output must take into account this dependency, which is achieved in [M. Chen et al. 2018; Kumar and Sarkar 2023] for Markovian data. We choose Oja's algorithm [Oja 1984] to use for the learning step in our MCMC scheme because it is analysed in both [M. Chen et al. 2018] and [Kumar and Sarkar 2023]. We find that the only other competing online PCA method, CCIPCA [Weng, Yilu Zhang, and Hwang 2003], dominates the performance of Oja's algorithm in the $m = 1$ case although it becomes numerically unstable in the $m > 1$ case with Markovian data.

Oja's algorithm is projected gradient descent on the objective tr $\left(V^T \Sigma_\pi V\right)$. The gradient of this objective is $\Sigma_\pi V$ and the projection is onto the set $O\left(d, m\right) := \left\{V \in \mathbb{R}^{d \times m}, V^T V = \mathbf{I}_m\right\}$. The gradient is estimated using $\left(X_t - \mu_t\right)\left(X_t - \mu_t\right)^T$ where X_t is the latest state in the Markov chain and $\mu_t \in \mathbb{R}^d$ is a running estimate of the mean of π. The full update abides by the following recursion:

$$V_t = \Pi_{O(d,m)}\left(V_{t-1} + \gamma_t\left(X_t - \mu_t\right)\left(X_t - \mu_t\right)^T V_{t-1}\right)$$

where $\Pi_{O(d,m)}$ describes a projection onto $O\left(d, m\right)$ which we implement using a Gram-Schmidt orthogonalisation and $\gamma_t > 0$ is a learning rate. The authors of [Cardot and Degras 2018] recommend using $\gamma_t = ct^{-\alpha}$ where $c \in (0, \infty)$ and $\alpha \in (0.5, 1]$. In all of our numerical examples we use $c = 1$ and $\alpha = 0.7$. After having made the update, the m columns serve as the eigenvectors from which we construct the orthogonal component of the preconditioner as described in section 5.3.2. For the computational complexity: the product $\left(X_t - \mu_t\right)^T V$ is $O\left(md\right)$ and the projection with Gram-Schmidt is $O\left(m^2 d\right)$.

5.3.3.2 Eigenvalues

Let $Q\left(V\right)$ denote the orthogonal matrix constructed in section 5.3.2 where v_i is just the ith column of V for all $i \in [m]$. Then if V contained the eigenvectors corresponding to the top eigenvalues of the target covariance, then

$$\begin{aligned} Q\left(V\right)^{T}\Sigma_{\pi}Q\left(V\right) &= \begin{pmatrix} V^{T} \\ W^{T} \end{pmatrix}\begin{pmatrix} V & W_{\pi} \end{pmatrix}\text{diag}\left\{\lambda_{i}^{\pi}:i\in[d]\right\}\begin{pmatrix} V^{T} \\ W_{\pi}^{T} \end{pmatrix}\begin{pmatrix} V & W \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I}_{m} & 0 \\ 0 & W^{T}W_{\pi} \end{pmatrix}\text{diag}\left\{\lambda_{i}^{\pi}:i\in[d]\right\}\begin{pmatrix} \mathbf{I}_{m} & 0 \\ 0 & W_{\pi}^{T}W \end{pmatrix} \end{aligned}$$

where $W\in\mathbb{R}^{d\times(d-m)}$ is the remaining $d-m$ columns of $Q\left(V\right)$, and the second equality holds because the columns of V are mutually orthonormal, and the columns of W are orthogonal to the columns V which are also orthogonal to the columns of W_{π}. Therefore, from the final expression, applying $Q\left(V\right)^{T}$ 'uncovers' the top m eigenvalues of the target covariance so if we scale the output of our MCMC algorithm by $Q\left(V\right)^{T}$ the first m marginal variances that result will be the top m eigenvalues. This is how we propose to learn these eigenvalues: defining $\tilde{\mu}_{t}:=Q\left(V_{t}\right)^{T}\mu_{t}$ as the transformation of our current estimate of the mean of π and $\tilde{X}_{t}:=Q\left(V_{t}\right)^{T}X_{t}$ as the transformation of the current Markov chain state we set

$$D_{t}=\left(D_{t-1}^{2}+\gamma_{t}\left(\text{diag}\left\{\left(\tilde{X}_{t}-\tilde{\mu}_{t}\right)_{i}^{2}:i\in[d]\right\}-D_{t-1}^{2}\right)\right)^{1/2}$$

where $\gamma_{t}>0$ is a learning rate. Then the final $d-m$ diagonal elements of D_{t} can be set to 1 in accordance with the description of the optimal preconditioner in section 5.3.1.

5.3.4 Implementation

The actual implementation has two additional adaptive steps apart from those described in section 5.3.3. The first is a standard update to the adaptive mean μ_{t}. The second is the learning of an optimal global scale parameter $\sigma>0$ for the proposal distribution of the Markov kernel. This is motivated by the optimal scaling literature [Beskos et al. 2013; G. O. Roberts, A. Gelman, and Gilks 1997; Jeffrey S Rosenthal et al. 2011], a line of research which establishes that for suitable high-dimensional model problems, the convergence behaviour of many MCMC algorithms can be described in terms of a limiting univariate Markov process (often a diffusion), with the value of σ informing the 'speed' at which this limiting process traces out its path.

In this setting, σ is selected to optimise some property of the limiting process. Various results suggest that the optimal value of σ can be characterised in terms of the average acceptance probability of the chain at equilibrium, independently of the target distribution. Tuning σ to control the acceptance rate also has motivations in non-asymptotic analysis, where one might instead care particularly about the worst-case acceptance rate out of any state x; see e.g. [Andrieu, A. Lee, et al. 2024]. Therefore we implement a learning step for $\sigma > 0$ based on matching an optimal acceptance rate $\alpha^* \in (0, 1]$.

The exact implementation can be found in algorithm 5.1

The overall computational complexity of the learning step is bottlenecked by Gram-Schmidt and is hence $O\left(m^2 d\right)$.

5.3.5 Implementation details

5.3.5.1 Adaptive variants

Along with the learning step described in algorithm 5.1, we also propose a variant in which step 5 (c) is excluded. The purpose of doing this is that the final $d - m$ diagonal elements of D_t will contain information about the marginal scales of the target covariance (along the directions of the final $d - m$ columns of $Q\left(V_t\right)$). Therefore step 5 (c) is possibly throwing away this information, even though using this information may harm the algorithm, as explained in 3.5. We call the variant that includes step 5 (c) ‘eigen_identity’ and the variant that excludes it ‘eigen’.

5.3.5.2 Competing adaptive schemes

Before we describe the adaptive schemes against which we compare ours, we note that in every scheme shown here we also initialise a global scale parameter at $\sigma_0 = 0.5 \times d^{-1/4}$ and adapt at each step according to the rule shown in step 4. of algorithm 5.1. Given that we use MALA as our underlying Markov kernel, we set $\alpha^* := 0.574$ in all circumstances according to [Gareth O. Roberts and Jeffrey S. Rosenthal 2001, Theorem 3 ii)]. Each adaptive scheme (including ours) updates its adaptive parameters at every step in the Markov chain.

Algorithm 5.1 Complete learning step for the proposed adaptive algorithm

```
inputs: learning rate $\gamma_t > 0$, latest state from the Markov chain $X_t \in \mathbb{R}^d$, latest acceptance probability $\alpha_t \in [0,1]$, optimal acceptance rate $\alpha^* \in (0,1]$, latest values of the adaptive parameters $\mu_{t-1} \in \mathsf{X}$, $V_{t-1} \in O(d,m)$, $D_{t-1} \in \mathbb{R}^{d \times d}$, $\sigma_{t-1} > 0$, mechanism for constructing the orthogonal component of the preconditioner $Q : O(d,m) \to O(d)$, Gram-Schmidt projection operator $\Pi_{O(d,m)} : \mathbb{R}^{d \times m} \to O(m,d)$, learning rate coefficient for Oja's algorithm $c \in (0,\infty)$, general learning rate tuning parameter $\alpha \in (0.5,1]$.
outputs: updated values of the adaptive parameters $\mu_t \in \mathbb{R}^d$, $V_t \in \mathbb{R}^{d \times m}$, $D_t \in \mathbb{R}^{d \times d}$

  1. Set the general learning rate: $\gamma_t = t^{-\alpha}$
  2. Learn the adaptive mean:
         $\mu_t = \mu_{t-1} + \gamma_t (X_t - \mu_{t-1})$
  3. Learn the adaptive eigenvector information:
      (a) Update the eigenvector matrix:
             $\tilde{V}_t = V_{t-1} + \gamma_t (X_t - \mu_t)(X_t - \mu_t)^T V_{t-1}$
      (b) Orthogonalise using Gram-Schmidt: $V_t = \Pi_{O(d,m)}\left(\tilde{V}_t\right)$
  4. Learn the adaptive global scale:
         $\log \sigma_t = \log \sigma_{t-1} + \gamma_t (\alpha_t - \alpha^*)$
  5. Learn the adaptive diagonal information
      (a) Project the location information along the new eigenvectors $V_t$: $\tilde{\mu}_t := Q(V_t)^T \mu_t$, $Q(V_t)^T X_t$
      (b) Learn the marginal variances along the new eigenvectors $V_t$:
             $D_t = \left(D_{t-1}^2 + \gamma_t \left(\text{diag}\left\{\left(\tilde{X}_t - \tilde{\mu}_t\right)_i^2 : i \in [d]\right\} - D_{t-1}\right)\right)^{1/2}$
      (c) Set $(D_t)_{ii} = 1$ for $i \in \{m+1,\dots,d\}$ in accordance with the form of the optimal preconditioner 5.3.1.
```

The first adaptive algorithm we compare ours to is one where we do nothing. We initialise at $L_0 = \mathbf{I}_d$ and do not perform any updates whatsoever. We call this adaptive scheme 'none' as it does nothing.

The second adaptive algorithm we compare ours to is the diagonal adaptive scheme. This scheme initialises at $L_0 = \mathbf{I}_d$ and upon each new state produced by the Markov kernel, attempts to learn the marginal standard deviations of the target, which are then used as the diagonal elements of L_t. Specifically, given a new state $X_t \in \mathbb{R}^d$ and a learning rate $\gamma_t > 0$ we set

$$L_t = \left(L_{t-1}^2 + \gamma_t \left(\text{diag}\left\{(X_t - \mu_t)_i^2 : i \in [d]\right\} - L_{t-1}^2\right)\right)^{1/2}$$

where $\mu_t \in \mathbb{R}^d$ is a running estimate of the mean of π. This learning step is equivalent to that described in algorithm 5.1 holding V_t such that $Q(V_t) = \mathbf{I}_d$ and skipping step 5 (c). Matrix multiplication, inversion, and square root are $O(d)$ with this preconditioner, and so it provides the fastest per-iteration algorithm of all those presented here (apart from 'none'). We refer to this algorithm as 'diagonal'.

The final adaptive algorithm we compare to is the dense adaptive scheme described in [Andrieu and Thoms 2008, Algorithm 4]. This scheme initialises at $L_0 = \mathbf{I}_d$ and upon each new state produced by the Markov kernel, attempts to learn the target covariance. The preconditioner L_t is then set to a matrix square root of the estimate of the target covariance. Specifically, given a new state $X_t \in \mathbb{R}^d$ and a learning rate $\gamma_t > 0$ we set

$$L_t = \left(L_{t-1}L_{t-1}^T + \gamma_t \left((X_t - \mu_t)(X_t - \mu_t)^T - L_{t-1}L_{t-1}^T\right)\right)^{1/2} \tag{5.2}$$

Matrix multiplication and inversion are $O\left(d^2\right)$ with this preconditioner, and so it provides the slowest per-iteration algorithm of all those presented here. We often found that the matrix inside the outer set of brackets in 5.2 would become non-positive definite. Therefore we also needed to detect its minimal eigenvalue so as to perturb it into positive definiteness. This operation takes $O\left(d^3\right)$.

5.3.5.3 Multiple chains

For each adaptive scheme we run multiple chains whose information we average over and use in the learning step. Mathematically this can be described by the following generic scheme in Algorithm 5.2.

Algorithm 5.2 Generic multiple chain adaptive algorithm

inputs: Markov kernels $\{K_L : \mathsf{X} \times \mathcal{X} \to [0,1]\}_{L \in \mathbb{R}^{d\times d}}$, learning increments $\left\{A : \mathsf{X} \times \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}\right\}_{t\in\mathbb{N}\setminus\{0\}}$, number of chains $k \in \mathbb{N}\setminus\{0\}$, initial state $\left(X_0^{(1)}, \dots, X_0^{(k)}\right) \in \mathsf{X}^k$, initial preconditioner $L_0 \in \mathbb{R}^{d\times d}$, chain length $n \in \mathbb{N}\setminus\{0\}$, learning rate $\{\gamma_t\}_{t\in\mathbb{N}\setminus\{0\}}$.

outputs: A process $\left\{\left(X_t^{(1)}, \dots, X_t^{(k)}\right)\right\}_{t=1}^{n}$

```
For $t \in [n]$ do
  1. Sample $\left(X_t^{(1)}, \dots, X_t^{(k)}\right) \sim \bigotimes_{i=1}^{k} K_{L_{t-1}}\left(X_{t-1}^{(i)} \to .\right)$
  2. Update the preconditioner:  $L_t = L_t + \gamma_t \frac{1}{k} \sum_{i=1}^{k} A\left(X_t^{(i)}, L_t\right)$
```

To give a concrete example: instead of using step 2. in algorithm 5.1 with a single chain, with multiple chains step 2. would take the form

$$\mu_t = \mu_{t-1} + \gamma_t \frac{1}{k} \sum_{i=1}^{k} \left(X_t^{(i)} - \mu_{t-1}\right)$$

Using multiple chains has been proposed in many different forms, see e.g. [Goodman and Weare 2010; Jacob, O'Leary, and Atchadé 2020] for general methods and [Craiu, J. Rosenthal, and C. Yang 2009; Riou-Durand et al. 2023; Schär, Habeck, and Rudolf 2024] for methods in adaptive MCMC and [Margossian and Andrew Gelman 2024; Sountsov, Carroll, and M. D. Hoffman 2024] for a discussion of the generic benefits of using many chains. The general idea is that MCMC methods are efficient for local exploration of target distributions, but lack the ability to acquire and exploit global information. Therefore using many chains allows global information to be gained and shared between chains, even though each individual chain is localised.

5.3.5.4 Algorithmic setup

The Markov kernel alongside all the adaptive methods shown here is the MALA kernel preconditioned with the matrix L_t.

For the 'none' adaptive scheme we have $L_{t-1} = \mathbf{I}_d$, for the 'diagonal' adaptive scheme L_{t-1} is a diagonal matrix containing the marginal standard deviations, for the 'dense' adaptive scheme L_{t-1} is the

Algorithm 5.3 Preconditioned MALA Markov kernel

inputs: previous state $X_{t-1} \in \mathbb{R}^d$, adaptive parameters $L_{t-1} \in \mathbb{R}^{d \times d}$, $\sigma_{t-1} > 0$, target density $\pi : \mathbb{R}^d \to [0, \infty)$ (normalised or unnormalised)
outputs: subsequent state $X_t \in \mathbb{R}^d$

1. Propose a new point:

$$Y_t = X_{t-1} + \frac{\sigma_{t-1}^2}{2} L_{t-1} L_{t-1}^T \nabla_x \log \pi \left(X_{t-1}\right) + \sigma_{t-1} L_{t-1} \xi$$

   where $\xi \sim \mathcal{N}\left(0, \mathbf{I}_d\right)$.

2. Accept the proposed point with probability

$$\alpha\left(X_{t-1} \to Y_t\right) = \min\left\{1, \frac{\pi\left(Y_t\right) \mathcal{N}\left(X_{t-1}; Y_t + \frac{\sigma_{t-1}^2}{2} L_{t-1} L_{t-1}^T \nabla_x \log \pi\left(Y_t\right), \sigma_{t-1}^2 L_{t-1} L_{t-1}^T\right)}{\pi\left(X_{t-1}\right) \mathcal{N}\left(Y_t; X_{t-1} + \frac{\sigma_{t-1}^2}{2} L_{t-1} L_{t-1}^T \nabla_x \log \pi\left(X_{t-1}\right), \sigma_{t-1}^2 L_{t-1} L_{t-1}^T\right)}\right\}$$

Cholesky factor of an estimate of the target covariance, for the ‘eigen’ and ‘eigen_identity’ schemes $L_{t-1} = Q\left(V_{t-1}\right) D_{t-1}$.

5.3.6 Numerical Experiments

Before introducing our numerical experiments we note that in every case we sum the ESSs for each chain dimension across all k chains described in section 5.3.5.3.

5.3.6.1 Ill-conditioned Gaussian

Here we compare our proposed adaptive schemes with the schemes described in 5.3.5.2 on an ill-conditioned Gaussian target with a dense covariance.

Experimental set-up The target has the form $\pi = \mathcal{N}\left((5, \ldots, 5)^T, \Sigma_\pi\right)$ where Σ_π has K significant eigenvalues sampled from $\mathcal{N}(100, 0.01)$ and $d - K$ smaller eigenvalues at 0.1. This gives the target a condition number of ≈ 1000. The first eigenvector of Σ_π is the all ones vector, and the rest of the eigenvectors are determined using the `svd` function in R.

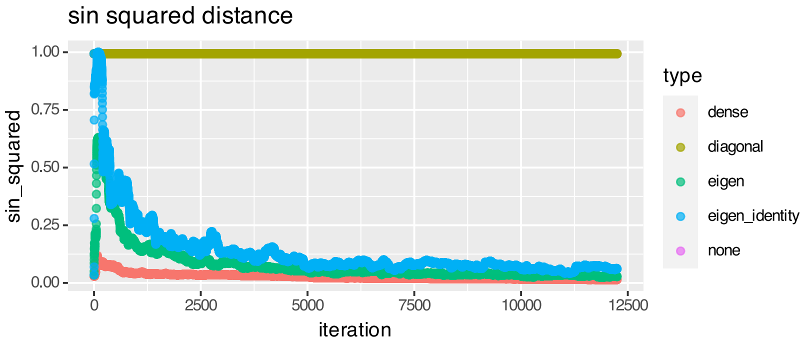

Figure 5.3: $\sin^2$ distance between the preconditioners' leading eigenvector and the leading eigenvector of the target covariance over a single MCMC chain in $d = 150$ with an ill-conditioned Gaussian target.

We use a $\gamma_t = (t+1)^{-0.7}$ learning rate, and a value of $c = 1$ for Oja's algorithm in the 'eigen' and 'eigen_identity' cases. We initialise the global scale at $\sigma_0 = 0.5d^{-1/4}$. We initialise all matrix valued adaptive parameters (i.e. D_0, L_0, $Q(V_0)$) at the identity.

Results First we demonstrate the adaptive algorithms' ability to learn their adaptive parameters. In figure 5.3, we look at the $\sin^2$ distance between the eigenvector associated with the top eigenvalue of the adaptive preconditioners, and the eigenvector associated with the top eigenvalue of the target covariance. Here we use a $d = 150$ dimensional $\pi = \mathcal{N}\left((5, \dots, 5)^T, \Sigma_\pi\right)$ target, but with a single $K = 1$ significant eigenvalue. We do this because if $K > 1$ the top eigenvectors of the preconditioners switch between the top eigenvectors of the target covariance, making the results difficult to read. We observe a single run for $1000\sqrt{d}$ iterations with $k = 2$ chains, where the 'eigen' and 'eigen_identity' algorithms learn $m = 3$ top eigenvectors.

What the plot contains but does not show is the $\sin^2$ distance between the top eigenvector of the 'none' preconditioner and the top eigenvector of the target. This is because it is hidden behind the 'diagonal' points: the $\sin^2$ distances for both of these preconditioners is fixed near 1 because they remain diagonal for the entire MCMC chain. In the 'dense', 'eigen', and 'eigen_identity' cases the $\sin^2$ distances drop close to zero over the course of the run. This shows that they are learning the top eigenvector of the target covariance. Note that the 'eigen_identity' performs slightly worse than the 'eigen' algorithm: this is possibly due to the fact that it ignores the additional marginal variance information, as explained in section 5.3.5.1

We now increase the number of significant eigenvalues in the target covariance to $K = 3$. The algo-

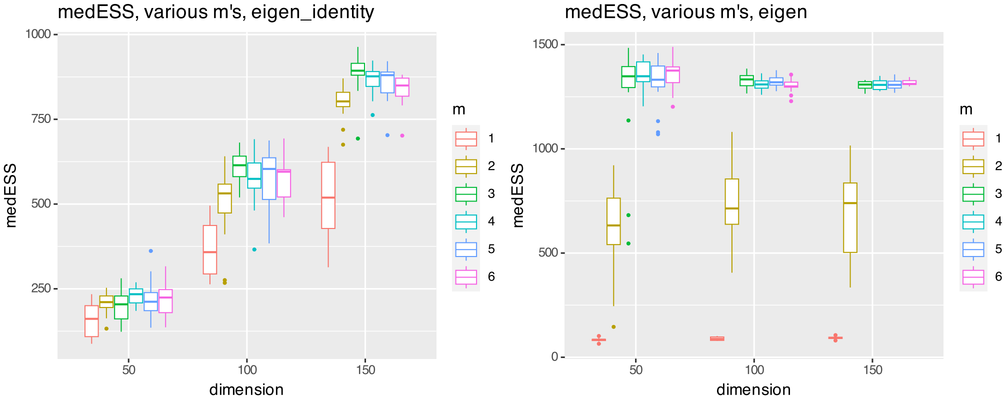

Figure 5.4: The median ESSs across the dimensions of each individual chain of the 'eigen_identity' and 'eigen' adaptive algorithms. The results are shown from various m's and dimensions on a $\mathcal{N}\left((5,\ldots,5)^T, \Sigma_\pi\right)$ where Σ_π is ill conditioned and dense.

rithms are in dimension $d \in \{50, 100, 150\}$ and we run the 'eigen' and 'eigen_identity' algorithms (i.e. our proposed algorithms) with $m \in \{1, 2, 3, 4, 5, 6\}$. The idea is that these algorithms should increase in performance when m increases from 1 to 3 and stabilise from 4 to 6. We collect the results from 15 chains for each $(d, m, \text{adaptive scheme})$ combination. For each algorithmic run we use $k = 2$ chains, each started in equilibrium and run for $1000\sqrt{d}$ iterations. In figure 5.4 we show the raw median ESSs for the 'eigen' and 'eigen_identity' schemes, across dimensions and across m values.

The broad takeaway from these plots is that increasing m (the number of eigenvectors we attempt to learn) does genuinely improve absolute performance up to $m = 3$ as expected. This effect is seen most strikingly in the 'eigen' case, pointing toward its superior ability to learn the eigenvectors shown in 5.3 (note the different y-axes in each plot). As we note in 3.1, the worse the conditioning of the target, the worse we will be punished for having misaligned eigenvectors between the target and the preconditioner [1]. Once the top three eigenvalues of the target are made to equal 1 by these preconditioners, the condition number will still be at least 10 by the existence of the remaining $d - 3$ eigenvalues at 0.1. The 'eigen' scheme has the ability to reduce this condition number, but the 'eigen_identity' scheme does not.

We note that the targets actually get easier to sample from as the dimension increases (in a non-time

[1] Strictly speaking this is about the alignment of the eigenvectors of the preconditioner and the eigenvectors of the Hessian of the target potential. However, in this case the target is normal and so the Hessian is just the precision of the target, which has the same eigenvectors as the covariance.

normalised sense): this is an artifact of using the `svd` function to construct the target covariance. This is corroborated by the raw ESS performances across the preconditioning schemes in figure 5.5. Therefore the fact that the ‘eigen’ preconditioner is doing equally well across all dimensions suggests that it is achieving minimality in the target condition number (with respect to its inherent construction) in all cases.

We now display the medians of the ESSs across the dimensions of the chains for all adaptive schemes. We fix $m = 3$ in the ‘eigen’ and ‘eigen_identity’ cases since 5.4 suggests that doing so will achieve the best time-normalised performance. We show both the absolute performance and the time-normalised performance in figure 5.5.

The raw results show that ‘dense’ and ‘eigen’ are the most competitive adaptive schemes, owing to the fact that they contain the most target information with which they can isotropise. Only the ‘eigen’ and ‘dense’ preconditioners have the ability to shrink the condition number of the target down past 10. Second is the ‘eigen_identity’ scheme that has the ability to shrink the target condition number from 1000 to 10. Third is the diagonal scheme, whose effect on the condition number of the target is unknown, but certainly reduced by the fact that Σ_π is dense. Last is ‘none’ which does not alter the target.

The time-normalised results show that the effect of the per-iteration time-complexity outweighs the fact that the target is getting easier to sample from with dimension. The ‘eigen’ strategy wins out due to its competitive raw performance and its $O\left(m^2 d\right)$ per-iteration time-complexity. Interestingly in the $d = 150$ case the ‘dense’ strategy is doing slightly worse than the ‘diagonal’ strategy.

5.3.6.2 Bayesian logistic regression with synthetic data

Here we compare the adaptive algorithms on a Bayesian logistic regression posterior with a classical g prior [Agliari and Parisetti 1988] with synthetic data.

Experimental set-up For this posterior, the likelihood is logistic and the prior is normal giving the following potential:

$$U\left(\beta\right) \propto \sum_{i=1}^{n} \left(\left(1 - Y_i\right) X_i^T \beta + \log\left(1 + \exp\left(-X_i^T \beta\right)\right)\right) + \frac{\lambda}{2n} \beta^T X^T X \beta \tag{5.3}$$

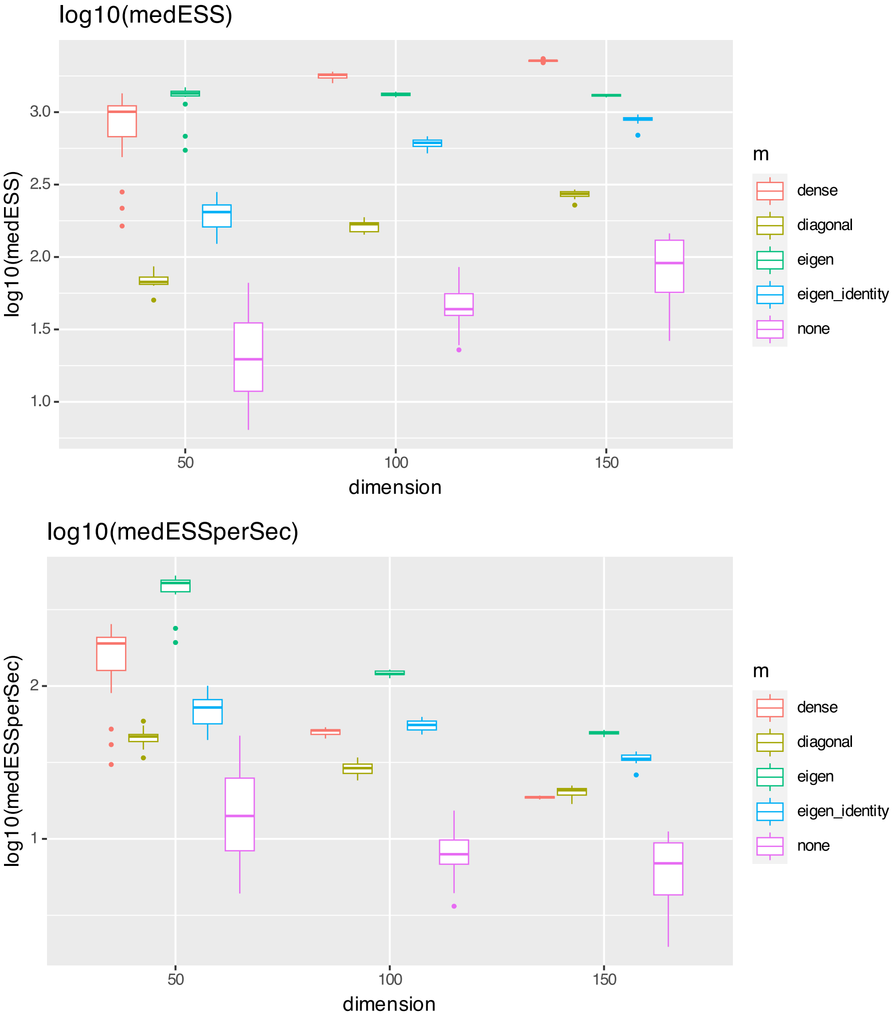

Figure 5.5: Two plots showing the raw and time-normalised log-transformed median ESSs across the dimensions of the chains. The results are shown in various dimensions on a $\mathcal{N}\left((5,\ldots,5)^T,\Sigma_\pi\right)$ where Σ_π is ill conditioned and dense. The logarithms are base 10.

where Y_i are the response variables which we sample from Bernoulli$\left(\left(1+\exp\left(-X_i^T\beta\right)\right)^{-1}\right)$ for $i \in [n]$, $X_i \in \mathbb{R}^d$ are the rows of the data matrix $X \in \mathbb{R}^{d\times n}$, and $\lambda > 0$ is chosen to determine the strength of the prior. The data matrix is of the form UDV^T where $U \in O(d)$ and $V \in O(n)$ are sampled from the Haar measure and $D \in \mathbb{R}^{d\times n}$ is a diagonal matrix with 3 of the diagonal elements sampled from $\mathcal{N}\left(1, 10^{-6}\right)$ and the rest set to $\sqrt{1000}$. The Hessian of the potential is then $\nabla^2 U(\beta) = X^T\Lambda(\beta)X$ where

$$\Lambda(\beta) := \text{diag}\left\{\exp\left(X_i^T\beta\right)\left(1+\exp\left(X_i^T\beta\right)\right)^{-2} + \frac{\lambda}{n} : i \in [n]\right\}$$

The diagonal elements of D are chosen in an attempt to make 3 of the eigenvalues of the target covariance 'significant' (in the sense that they are much larger than the rest). The condition number of the posterior is

$$\kappa = \kappa\left(X^TX\right)\frac{\frac{1}{4}n+\lambda}{\lambda} \approx 1000\frac{\frac{1}{4}n+\lambda}{\lambda}$$

We set $\lambda = 0.01$ and $n = d$ for all $d \in \{50, 100, 150, 200, 300\}$. For each algorithmic run we use $k = 2$ chains and run for $1000\sqrt{d}$ iterations. For each combination of dimension and adaptive scheme we use 15 runs, initialising at the mode which we find using preconditioned gradient descent on U.

For the adaptive schemes we use a learning rate of $\gamma_t = (t+1)^{-0.7}$ and $c = 1$ for Oja's algorithm in 'eigen' and 'eigen_identity'. We initialise the global scale at $\sigma_0 = 0.5d^{-1/4}$. We initialise all matrix valued adaptive parameters (i.e. D_0, L_0, $Q(V_0)$) at the identity. We use $m = 3$ for both the 'eigen' and 'eigen_identity' adaptive schemes.

Results In figure 5.6 we show the log-transformed medians of the raw ESSs and the log-transformed medians of the time-normalised ESSs across the dimensions of each Markov chain.

The first thing to note is that the raw performance of the 'dense', 'eigen', and 'eigen_identity' algorithms increases with dimension. The exact reason why this occurs is not fully clear although we suspect that it is an artifact of the way in which we construct the data matrix X. As in the Gaussian example in section 5.3.6.1 the 'eigen_identity' scheme lags behind the 'eigen' scheme in terms of its performance. However in this case the differences in performance are much less than in the Gaussian case. The 'eigen_identity' and the 'eigen' scheme both remove the effect of the leading eigenvalue of X^TX on the condition number by

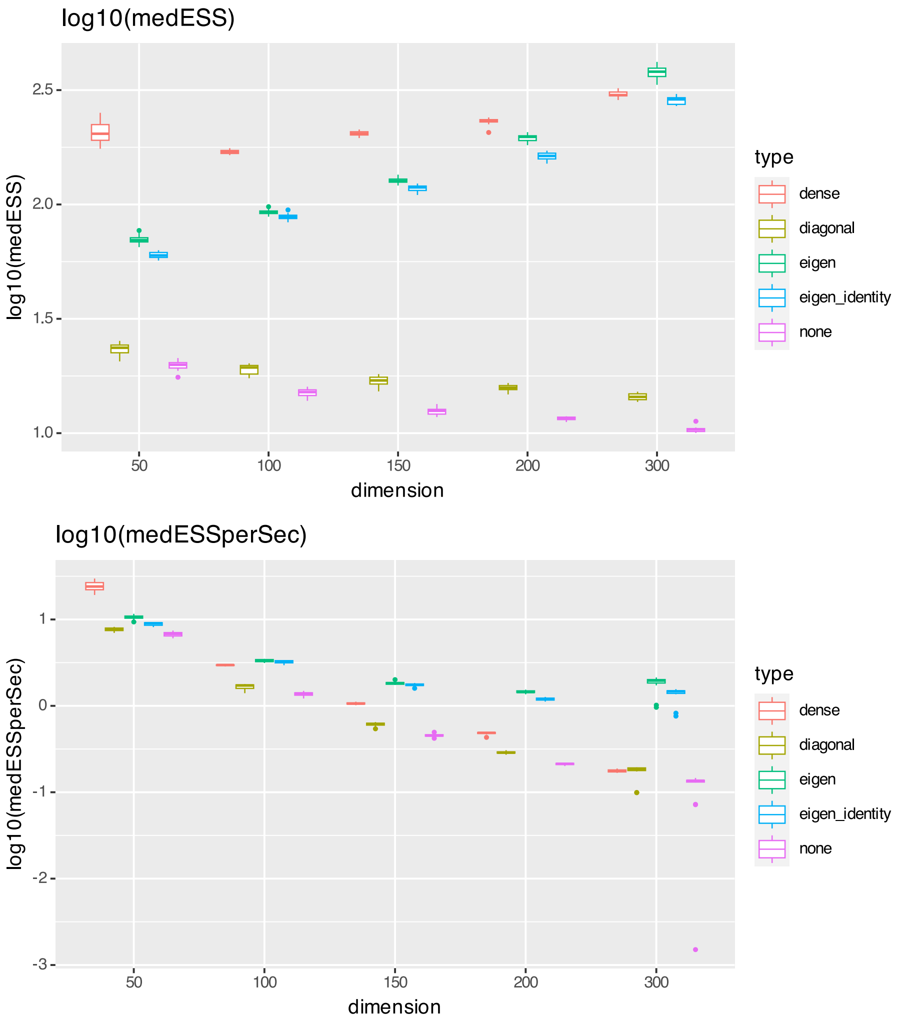

Figure 5.6: Two plots showing the raw and time-normalised log-transformed median ESSs across the dimensions of the chains. The results are shown in various dimensions on a Bayesian logistic regression posterior with a g prior, whose potential is as defined in 5.3.

setting it to 1. However in this case the remaining eigenvalues of $X^T X$ are at 1 as well, and so the influence of $\kappa \left(X^T X\right)$ on the condition number cannot be further reduced by the 'eigen' scheme, meaning that it loses its advantage over the 'eigen_identity' scheme.

The next thing to note is that the 'diagonal' scheme does not do much better than the 'none' scheme. This indicates that the target covariance is dense, and has sufficiently many significant off diagonal elements. Despite the fact that the raw median ESSs are increasing for the 'dense', 'eigen', and 'eigen_identity' schemes, the computational complexity of the 'dense' scheme causes its time-normalised performance to decay sufficiently rapidly that it is dominated by the 'eigen' and 'eigen_identity' schemes in all dimensions higher than 100. In fact, in $d = 300$, the time-normalised performance of the 'dense' scheme is dominated by that of the 'diagonal' scheme.

5.3.7 Summary and extensions

5.3.7.1 Summary

In section 5 we introduce a sparsely parametrised preconditioner that uses the m top eigenvalues of the target covariance, and their associated eigenvectors. We show that this preconditioner can be learned and used over the course of an MCMC algorithm and is competitive when compared with standard existing preconditioners, such as those which learn the full target covariance and those which learn the diagonal of the target covariance. This is due to two properties: i) the way in which it is parametrised allows it to be dense with respect to its off-diagonal elements and ii) its learning step and the operations which involve it in the Markov kernel can be executed in $O\left(m^2 d\right)$ time-complexity.

In section 5.3.1 we show that the optimal parametrisation that is constructed with the full knowledge of the top m eigenvalues and vectors of the target covariance does indeed isotropise the target, in that it reduces the 2-norm of its covariance. In section 5.3.2 we show how to construct the preconditioner using Householder matrices in such a way as to allow computation in $O\left(m^2 d\right)$ complexity. In 5.3.3 we introduce the mechanism by which we propose to learn the top m eigenvalues and vectors of the target covariance using a form of online PCA called 'Oja's algorithm' [Oja 1984]. In 5.1 we detail the implementation of the full learning mechanism of the preconditioner. In section 5.3.5.1 we introduce a variant of our preconditioner which extracts additional diagonal information.

In section 5.3.6.1 we test our adaptive schemes on a Gaussian target whose covariance is dense, and has 3 eigenvalues which are significantly larger than the rest. Figure 5.3 demonstrates that our adaptive schemes do in fact learn the top eigenvector. Figure 5.4 shows that our schemes improve with increasing m until $m \geq 3$, which matched our expectations. Figure 5.5 demonstrates that the time-normalised performance of our schemes dominate those of their competitors in higher dimensions. Section 5.3.6.2 shows the performance of the various adaptive schemes in sampling from a Bayesian logistic regression posterior with a g prior, whose data has been synthesised in an attempt to make 3 of the eigenvalues of the target covariance 1000 times greater than the rest. We again see that our adaptive schemes dominate the others in high dimension because of their ability to isotropise the target coupled with the fact that they operate at reduced per-iteration computational complexity.

5.3.7.2 Extensions

One obvious extension of the work here is to test our preconditioners on target distributions that are not constructed so as to have the correct eigenstructure in their covariances. It would also be interesting to see how our adaptive schemes fared outside the log-concave target case. [Langmore et al. 2020] define a condition number for sampling with HMC. They remark that a target covariance with a few large eigenvalues and many small ones renders a large condition number. This happens to be the case in which our preconditioner is best suited. A natural extension would be then to test our preconditioner alongside HMC, instead of MALA. We have left the selection of m (the number of top eigenvalues and eigenvectors of the target covariance we wish to learn) unexamined. Is there a natural way in which this could be done? For instance: we may initialise m at a large value and then gradually reduce it as we learn the eigenstructure of the target covariance as the MCMC algorithm continues to sample. A simplified, $m = 1$ version of our scheme is used within a larger adaptive scheme in [Riou-Durand et al. 2023, Section 3.2]. If we used our scheme instead, would this benefit the performance of the larger adaptive scheme?

The introduction of different Markov kernels points towards the broader question of how the per sample ESS affects the performance of adaptive MCMC algorithms. For instance there exist contexts in which the per sample ESS of (unpreconditioned) HMC is higher than that of (unpreconditioned) RWM. What does this mean for adaptive MCMC schemes, especially those that learn every iteration?

Chapter 6

Conclusion

We conclude by elaborating in broad terms on what we feel to be the overall contributions of the thesis. We also offer a view of the natural research program that extends from the work contained here.

6.1 Overall contributions

In all we hope to have given the reader a sense of what is both required and beneficial when one wishes to enhance canonical MCMC methods in the face of challenging target distributions, in light of new theory and the new technologies provided by adjacent fields, such as Variational Inference.

- In chapter 1 we provide the theoretical groundwork necessary to understand when algorithms based on Markov chains, such as MCMC, satisfy basic desiderata such as π-ergodicity 8. In section 1.3.2 we define the canonical MCMC algorithms of Random Walk Metropolis, Metropolis Adjusted Langevin Algorithm, and Hamiltonian Monte Carlo, so that we can define their enhancements in chapter 2.
- In chapter 2 we looked at three methods for enhancing these canonical algorithms: preconditioning in section 2.1, variational approximation in section 2.2, and adaptivity in section 2.3. In section 2.1 we framed the analysis of preconditioning through the lens of the condition number 2.4 since it is those target distributions whose condition number is finite that are most amenable to linear preconditioning:

the technique studied in chapter 3. We introduce Variational Inference in section 2.2, noting its complementary benefits with respect to MCMC at the start of section 2.2.2 i.e. that it is fast, but biased whereas MCMC is slow but (asymptotically) unbiased. This suggests synthesising the two methods, as we do in chapter 4. Preconditioners must be learned, often using information from an MCMC algorithm. This is the practice of adaptivity, which we introduce in 2.3. After recounting some cautionary tales in section 2.3.2.2 we present two existing adaptive algorithms in section 2.3.3, namely of [Haario, Saksman, and Tamminen 2001] and [Michalis Titsias 2023], noting their exact computational complexity, to be compared with the complexity of the scheme we construct in chapter 5.

- Chapter 3 offers the first rigourous examination of the effects of linear preconditioning on sampler performance. Each new linear preconditioner proposed in the literature comes with its own justification, which are often heuristic. We therefore provide a united framework under which to assess their effectiveness by looking at how the condition number 2.4 changes after linear preconditioning. We examine how popular preconditioners, such as those derived from the target covariance [Haario, Saksman, and Tamminen 2001] and the 'Fisher matrix' of [Michalis Titsias 2023], fit within this framework, and therefore serve their intended purpose of improving sampling efficiency. In section 3.4 we assert a condition on the potential on the target, namely that the spectrum of its Hessian does not vary too much, that allows us to achieve a tight dependence on the condition number of the spectral gap of Random Walk Metropolis, building on the theory developed in [Andrieu, A. Lee, et al. 2024]. Often the heuristic explanations for the success of preconditioners do not extend to rigourous justifications for the benefits they seem to provide. This is shown in 3.5 where we find that a popular preconditioner constructed solely from the marginal variances of the target will hinder algorithmic performance, in certain cases.
- As we mentioned earlier in the conclusion, it is our contention that Variational Inference is a potentially very powerful tool to use in conjunction with MCMC due to the complementary properties of the two methods. In chapter 4 we provide one such solution. Given a Markov chain and a variational approximation to the target, use a single thread of compute to generate the Markov chain, and the other threads to sample from the target restricted to pre-defined regions of the state space using a restricted

rejection sampler. One then combines the resulting samples into an estimator, see the theoretical construction in section 4.1.2.2 and implementation details in section 4.1.4. This construction is fully general: one can use any Markov chain, and any variational approximation, and we prove in section 4.1.3 that the resulting process inherits many of the efficiencies of the underlying chain. The samples used in the estimator we construct are decorrelated from one another due to the fact that they are from both the Markov chain and from the target restricted to specific regions of the state space. We show this fact empirically in section 4.1.5 on a bimodal Gaussian mixture target, and on the Ising model.

- Although lacking in theoretical guarantees, adaptive MCMC is one of the most established methods of enhancing MCMC algorithms. One can realise the extraordinary gains in efficiency when the adaptive procedure finds a good preconditioner, see, for example, the experiments in section 3.6. Adaptive algorithms must be developed to suit the computational constraints inherent to targets in high dimensions. In chapter 5 we provide an adaptive mechanism that learns the top $m \in \mathbb{N}$ eigenvalues of the target covariance along with their associated eigenvectors, see section 5.3.3. We use this eigeninformation in a sparse parametrisation of a preconditioner that captures correlation and marginal variance information about the target distribution, see section 5.3.2. This information can be learned and used within a Markov kernel in $O\left(m^2 d\right)$ computational operations per iteration (where d is the dimension of the state space), see sections 5.3.2 and 5.3.3. If learned properly such a preconditioner isotropises the target distribution, see proposition 75. We provide numerical experiments in section 5.3.6 to show that the speed of our adaptive scheme combines with its ability to learn correlations in the target leading to an advantage over slower $O\left(d^2\right)$ schemes, such as in [Haario, Saksman, and Tamminen 2001], and schemes that only incorporate marginal variance information.

6.2 Future research directions

There are many interesting ways in which the work presented here can be developed.

- **Extending the analysis of preconditioners to the nonlinear case**: Preconditioners are conceived to ameliorate pathologies in the target distribution that are not adequately overcome by the sampling algorithms we use. For instance, anisotropy in the target is addressed by linear preconditioning. However there exist pathologies such as heavy-tailedness and multimodality that are impossible to fix with a linear transformation (applied either to the target distribution or the algorithm). Therefore the need to use nonlinear transformations arises, see [Gabrié, Rotskoff, and Vanden-Eijnden 2022; Girolami and Calderhead 2011; L. T. Johnson and Geyer 2012; Parno and Y. M. Marzouk 2018] for examples. Can we hope to offer a unified framework under which we can assess the effectiveness of these preconditioners, as we do for linear preconditioners in chapter 3? We showed in proposition 26 that a given linear preconditioner can be viewed as a linear transformation to the target distribution, or to the sampling algorithm. The same is not true for a given nonlinear preconditioner, see [B. J. Zhang, Y. M. Marzouk, and Spiliopoulos 2024, Remark 3.3]. Therefore it is a design choice whether to precondition the target or to precondition the sampler. What circumstances should lead one to be chosen over another?

- **Altering the definition of the condition number**: It is clear that the condition number, as defined in 2.4, is a very harsh measure of the difficulty with which we can sample from a given target distribution. For instance, a target can be perfectly well-conditioned (i.e. isotropic) within the vast amount of the target mass, but if it loses strong log-concavity or smoothness in a single point in the tails the condition number in 2.4 will describe it as impossible to sample from. We should therefore wish to define a condition number which is finite under a much larger class of distributions than those that are strongly log-concave and smooth. This should ideally be done by stating desiderata for an ideal condition number (i.e. one that is sufficiently descriptive of the difficulty of the sampling problem, given the computational resources we have to solve it). One can then construct a better condition number using invented quantities, or quantities that exist in the theory of Markov processes, and assess whether it satisfies these desiderata.

- **Constructing preconditioners**: A mode of analysis for general preconditioners, as proposed in the first bullet, and an alternative definition of the condition number, as proposed in the second, could, if successful, be used to construct new preconditioners. For an example, see [Cui, Tong, and Zahm 2024], who use an extension of a Poincaré inequality [Bobkov 1999] to construct a nonlinear preconditioner using a Stein kernel [Fathi 2019]. If one has difficulty learning and using linear preconditioners in high dimension, as addressed by chapter 5, then one will certainly have difficulty learning and using a nonlinear preconditioner. Therefore ample motivation exists to create computationally lightweight nonlinear preconditioners.

- **Further exploiting Variational Inference and MCMC**: In our view the combination of Variational Inference and MCMC has not been explored to its fullest extent. For instance an MCMC chain can allow one to calculate gradients of the forward Kullback-Leibler divergence, as opposed to the reverse, as is usually done in Variational Inference. Therefore if a Markov kernel uses a variational approximation, like is the case with the Occlusion Process in chapter 4, one can iteratively improve the variational approximation and use it to enhance the Markov kernel, as is done with a preconditioner in adaptive MCMC.

Chapter 7

Appendices

7.1 Appendix A: Notation

7.1.1 Notation for chapter 1

- Say $\pi : \mathcal{X} \to [0,1]$ is a probability measure and $n \in \mathbb{N} \backslash \{0\}$. Then $\pi^{\otimes n} : \mathcal{X}^n \to [0,1]$ is the n-fold product of π defined such that $\pi^{\otimes n}((A_1, \ldots, A_n)) := \pi(A_1) \cdots \pi(A_n)$ for all $(A_1, \ldots, A_n) \in \mathcal{X}^n$.

- The support of a probability measure $\pi : \mathcal{X} \to [0,1]$, denoted supp (π), is defined as the smallest closed subset $S \in$ X such that $\pi(S) = 1$.

- We use $\mathcal{N}(\mu, \sigma^2)$ to denote the Gaussian distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. We use $\mathcal{N}(\mu, \Sigma)$ to denote the Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d \times d}$ for some $d \in \mathbb{N} \backslash \{0\}$. We denote by $\mathcal{N}(x; \mu, \sigma^2)$ and $\mathcal{N}(x; \mu, \Sigma)$ their respective densities.

- We define $\|f\|_\infty := \sup_{x \in \mathsf{dom}(f)} |f(x)|$ as the sup-norm for all f in an appropriate function space. We denote the Euclidean norm by $\|.\|_2 : \mathbb{R}^d \to [0, \infty]$. We also use $\|.\|_2 : \mathbb{R}^{d \times d} \to [0, \infty]$ to mean the induced operator norm defined by

$$\|A\|_2 := \sup_{v \in \mathbb{R}^d, \|v\|_2 = 1} \|Av\|_2$$

for all $A \in \mathbb{R}^{d\times d}$.

- Given $k \in \mathbb{N}\backslash\{0\}$ we define $[k] := \{1, \ldots, k\}$.
- For a Markov kernel $K : \mathsf{X} \times \mathcal{X} \to [0,1]$ we define its action on a measure $\nu : \mathcal{X} \to [0,1]$ by

  $$\nu K(A) := \int_{\mathsf{X}} \nu(dx) K(x \to A)$$

  for all $A \in \mathcal{X}$. We define its action on a function $f : \mathsf{X} \to \mathbb{R}$ by

  $$Kf(x) := \int_{\mathsf{X}} K(x \to dx') f(x')$$

  For all $n \in \mathbb{N}\backslash\{0\}$ we define the Markov kernel $K^n : \mathsf{X} \times \mathcal{X} \to [0,1]$ iteratively using

  $$K^n(x \to A) = \int_{\mathsf{X}} K^{n-1}(x \to dx') K(x' \to A)$$

  for all $x \in \mathsf{X}$ and $A \in \mathcal{X}$, with $K^1 := K$.
- We define $\delta_x : \mathcal{X} \to [0,1]$ as the Dirac measure at $x \in \mathsf{X}$.
- For a measure $\pi : \mathcal{X} \to [0,1]$ we define $L^k(\pi)$ for $k \in (0,\infty)$ as the set $\left\{f : \mathsf{X} \to \mathbb{R} : \pi\left(|f|^k\right) < \infty\right\}$ and $L_0^k(\pi)$ as the set $\left\{f : \mathsf{X} \to \mathbb{R} : \pi\left(|f|^k\right) < \infty, \pi(f) = 0\right\}$. The space $L^2(\pi)$ is naturally equipped with an inner product $\langle ., . \rangle_\pi : L^2(\pi) \times L^2(\pi) \to \mathbb{R}$ which is defined using

  $$\langle f, g \rangle_\pi := \int_{\mathsf{X}} \pi(dx) f(x) g(x)$$

  for all $f, g \in L^2(\pi)$. This then defines a norm $\|.\|_2 : L^2(\pi) \to [0,\infty)$ with $\|f\|_2 := \sqrt{\langle f, f \rangle_\pi}$ (we omit the π dependency of the norm where obvious).

7.1.2 Notation for chapter 2

- We overload the diag function as follows: diag(A) is the diagonal matrix that shares its diagonal with $A \in \mathbb{R}^{d\times d}$ and diag$\{f(i) : i \in [d]\} \in \mathbb{R}^{d\times d}$ is the diagonal matrix whose (i,i)th element is $f(i)$.

- We denote by $GL_d(\mathbb{R})$ the set of invertible $d \times d$ matrices over $\mathbb{R}$.

- We use the $\preceq$ and $\succeq$ relations to define a partial ordering on the set of symmetric matrices in the following way: $A \preceq B$ (resp. $A \succeq B$) if and only if $B - A$ (resp. $A - B$) is positive semidefinite. This ordering is also known as the Loewner order. The relations $\prec$ and $\succ$ are defined similarly, replacing the semidefiniteness condition with definiteness.

- We define $\|.\|_2 : \mathbb{R}^{d\times d} \to [0, \infty]$ as the matrix 2-norm that returns the largest singular value of the enclosed matrix.

- For two real-valued functions $f(n)$ and $g(n)$ we say $f(n) = O(g(n))$ if there exists a universal constant $K > 0$ such that $|f(n)| \leq Kg(n)$ for all n sufficiently large.

- We say $f(n) = \Omega(g(n))$ if there exists a universal constant $K > 0$ such that $|f(n)| \geq Kg(n)$ for all n sufficiently large.

- We say $f(n) = \tilde{O}(g(n))$ if $f(n)\log^{-q}(n) = O(g(n))$ for some $q \in \mathbb{N}$ and define $\tilde{\Omega}$ analogously.

- We denote by I the identity map on $L^2(\pi)$ for some probability measure π. For a given operator $K : L^2(\pi) \to L^2(\pi)$ we define the Dirichlet form $\mathcal{E}(K, f) := \langle (I - K) f, f\rangle$ for all $f \in L^2(\pi)$ where the inner product is that which is defined in $L^2(\pi)$. When K is π-reversible we define its right spectral gap (we we often refer to as simply its spectral gap) as

$$\gamma := \inf_{f\in L_0^2(\pi), f\neq 0} \frac{\mathcal{E}(K, f)}{\mathsf{Var}_\pi(f)} \tag{7.1}$$

  The relaxation time is simply the inverse of the spectral gap γ^{-1}.

- Given a distance metric $\mathcal{D}$ on the space of probability measures and an initial measure ν_0 the ε-mixing time of K starting from ν_0 is defined for all $\varepsilon > 0$ as

$$t(\epsilon, \nu_0) := \inf\{n \in \mathbb{N} : \mathcal{D}(\nu_0 K^n, \pi) \leq \varepsilon\} \tag{7.2}$$

- We say that a given Markov chain initialised according to the probability measure ν_0 has a β-warm start if there exists a constant $\beta \in \mathbb{R}$ such that

$$\sup_{A \in \mathcal{X}} \frac{\nu_0 (A)}{\pi (A)} \leq \beta \tag{7.3}$$

.

- When $T : \mathsf{X} \to \mathsf{Y}$ is a bimeasurable diffeomorphism we define the pushforward of a probability measure $\pi : \mathcal{X} \to [0,1]$ through T as the probability measure $\tilde{\pi} : \mathcal{Y} \to [0,1]$ defined with $\tilde{\pi}(A) := \pi\left(T^{-1}(A)\right)$ which has density

$$\tilde{\pi}(y) := \pi\left(T^{-1}(y)\right) \left|\det J_{T^{-1}}(y)\right|$$

where $J_{T^{-1}} : \mathsf{Y} \to \mathbb{R}^{\mathsf{dim(X)} \times \mathsf{dim(Y)}}$ is the Jacobian of T^{-1}.

- We define $\mathcal{P}(\mathcal{X})$ as the set of probability measures on the σ-field $\mathcal{X}$.

7.1.3 Notation for chapter 3

- For all $n \in \mathbb{N} \backslash \{0\}$ the we define $O(n)$ as the set of $n \times n$ orthogonal matrices over the reals.

- For a function $g \in C^2\left(\mathbb{R}^d\right)$ we define $\nabla_y g(y) \in \mathbb{R}^d$ and $\nabla_y^2 g(y)$ elementwise as

$$\left(\nabla_y g(y)\right)_i := \frac{\partial}{\partial y_i} g(y) \quad \left(\nabla_y^2 g(y)\right)_{ij} := \frac{\partial^2}{\partial y_i \partial y_j} g(y)$$

for $i, j \in [d]$. We will often drop the y subscript where the variable we are differentiating with respect to is obvious.

- For a given symmetric matrix A we let $\lambda_i(A)$ be its ith largest eigenvalue. We define its spectral condition number as

$$\kappa(A) := \frac{\max_{i \in [d]} |\lambda_i(A)|}{\min_{i \in [d]} |\lambda_i(A)|}$$

- We define the Frobenius norm $\|.\|_F : \mathbb{R}^{d \times d} \to [0, \infty)$ with $\|A\|_F := \sqrt{\mathsf{tr}\left(A^T A\right)}$ for all $A \in \mathbb{R}^{d \times d}$.

7.1.4 Notation for chapter 4

- Say that the state space X is partitioned into $\{\mathsf{X}_i : i \in [R]\}$ for $R \in \mathbb{N}\setminus\{0\}$ such that $\pi(\mathsf{X}_i) > 0$ for a target measure π for all $i \in [R]$. We define π_i to be π restricted to X_i for all $i \in [R]$. Given a function $f \in L^2(\pi)$ we define $\mu_i := \pi_i(f)$ and $\sigma_i^2 := \mathrm{Var}_{\pi_i}(f)$.

- Given a size R partition of X we define the function

$$\rho(x) := \sum_{i=1}^{R} i \mathbb{1}\{x \in \mathsf{X}_i\}$$

that simply outputs which part $x \in \mathsf{X}$ is in.

- Given three random variables X, Y, and Z we denote by $X \perp Y | Z$ the independence of X and Y given Z.

- For a set A we denote by $\mathcal{P}(A)$ its power set.

- For a measurable set $A \in \mathcal{X}$ we define $-A := \{-a : a \in A\}$.

- We denote by $\mathbb{R}^+$ the positive reals.

7.1.5 Notation for chapter 5

- For two real-valued functions $f(n)$ and $g(n)$ we say $f(n) = o(g(n))$ if for all constants $K > 0$ we have $|f(n)| \le Kg(n)$ for all n sufficiently large.

- When Y is a random variable, $\mathcal{L}(Y)$ is the probability distribution that it is sampled from.

- For $d, m \in \mathbb{N}\setminus\{0\}$ we define the set $O(d, m) := \left\{V \in \mathbb{R}^{d\times m} : V^T V = \mathbf{I}_m\right\}$.

- For two vectors $v, w \in \mathbb{R}^d$ we define their $\sin^2$ distance to be $1 - \left(v^T w\right)^2$.

7.2 Appendix B: Proofs

7.2.1 Proofs from chapter 1

7.2.1.1 Proof of lemma 3

Let λ_{X} be the Lebesgue measure on X and λ be the Lebesgue measure on $\mathbb{R}$. Then for all $A \times B \in \mathsf{X} \times \mathcal{B}(\mathbb{R})$

$$\begin{aligned}
\mathbb{P}\left((X_1, U_1) \in A \times B\right) &= \int_A \pi(dx)\, \mathbb{P}\left(U_1 \in B \,|X_1 = x\right) \\
&= \int_A \pi(dx)\, \frac{\lambda\left(B \cap [0, C\pi(x)]\right)}{C\pi(x)} \\
&= \int_A \lambda_{\mathsf{X}}(dx)\, \frac{\lambda\left(B \cap [0, C\pi(x)]\right)}{C}
\end{aligned}$$

For (X_2, U_2) we first calculate the normalising constant of Uniform $\{(x, u) : 0 \leq u \leq C\pi(x)\}$:

$$\begin{aligned}
\int_{0 \leq u \leq C\pi(x)} \lambda_{\mathsf{X}}(dx)\, \lambda(du) &= \int_{x \in \mathsf{X}} \left(\int_0^{C\pi(x)} \lambda(du)\right) \lambda_{\mathsf{X}}(dx) \\
&= \int_{x \in \mathsf{X}} C\pi(x)\, \lambda_{\mathsf{X}}(dx) \\
&= C
\end{aligned}$$

Therefore we have that

$$\begin{aligned}
\mathbb{P}\left((X_2, U_2) \in A \times B\right) &= \int_{0 \leq u \leq C\pi(x)} \mathbb{1}\{x \in A\}\, \mathbb{1}\{u \in B\}\, \frac{\lambda_{\mathsf{X}}(dx)\, \lambda(du)}{C} \\
&= \int_{x \in \mathsf{X}} \lambda_{\mathsf{X}}(dx)\, \mathbb{1}\{x \in A\} \left(\int_0^{c\pi(x)} \mathbb{1}\{u \in B\}\, \lambda(du)\right) C^{-1} \\
&= \int_A \lambda_{\mathsf{X}}(dx)\, \frac{\lambda\left(B \cap [0, C\pi(x)]\right)}{C}
\end{aligned}$$

as required.

7.2.1.2 Proof of the relation between Var $\left(\hat{f}_{\textbf{strat}}\right)$ and Var $\left(\hat{f}_n\right)$

Firstly note that in the proportional allocation case we have Var $\left(\hat{f}_{\text{strat}}\right) = n_{\text{strat}}^{-1}\text{Var}_{\pi}\left(\overleftarrow{\pi} f\right)$. Equally we have that Var $\left(\hat{f}_n\right) = n^{-1}\text{Var}_{\pi}\left(f\right)$. Therefore the expression we wish to show becomes $\text{Var}_{\pi}\left(\overleftarrow{\pi} f\right) = \left(1 - \text{Corr}_{\pi}\left(f, \overrightarrow{\pi} f\right)^2\right)\text{Var}_{\pi}\left(f\right)$. First we evaluate the covariance between f and $\overrightarrow{\pi} f$:

$$\begin{aligned}
\text{Cov}_{\pi}\left(f, \overrightarrow{\pi} f\right) &= \int_{\mathsf{X}} \pi\left(dx\right)\left(f\left(x\right) - \pi\left(f\right)\right)\left(\overrightarrow{\pi} f\left(x\right) - \pi\left(f\right)\right) \\
&= \sum_{i=1}^{R}\left(\pi_i\left(f\right) - \pi\left(f\right)\right)\int_{\mathsf{X}_i} \pi\left(dx\right)\left(f\left(x\right) - \pi\left(f\right)\right) \\
&= \sum_{i=1}^{R}\left(\pi_i\left(f\right) - \pi\left(f\right)\right)^2 \pi\left(\mathsf{X}_i\right) \\
&= \text{Var}_{\pi}\left(\overrightarrow{\pi} f\right)
\end{aligned}$$

The right hand of the expression is then:

$$\begin{aligned}
\left(1 - \text{Corr}_{\pi}\left(f, \overrightarrow{\pi} f\right)^2\right)\text{Var}_{\pi}\left(f\right) &= \left(1 - \frac{\text{Cov}_{\pi}\left(f, \overrightarrow{\pi} f\right)^2}{\text{Var}_{\pi}\left(f\right)\text{Var}_{\pi}\left(\overrightarrow{\pi} f\right)}\right)\text{Var}_{\pi}\left(f\right) \\
&= \text{Var}_{\pi}\left(f\right) - \text{Var}_{\pi}\left(\overrightarrow{\pi} f\right)
\end{aligned}$$

which is clearly equal to $\text{Var}_{\pi}\left(\overleftarrow{\pi} f\right)$ by 1.3.

7.2.1.3 Proof of proposition 6

Let $\gamma_i := \pi\left(\mathsf{X}_i\right)\sqrt{\text{Var}_{\pi_i}\left(f\right)}$. The we need to minimise $f\left(n_1, \ldots, n_R\right) = \text{Var}\left(\hat{f}_{\text{strat}}\right) = \sum_{i=1}^{R} n_i \gamma_i^2$ subject to the constraint that $g\left(n_1, \ldots, n_R\right) = n_1 + \cdots + n_R = n$. Setting the gradient of the Lagrangian of this problem to zero gives $-\gamma_i^2 n_i^{-2} = \lambda$ for all $i \in [R]$. Multiplying each equation by n_i and adding them all gives

$$\lambda = -\frac{1}{n}\sum_{i=1}\frac{\gamma_i^2}{n_i}$$

Subbing this value of λ into each equation $-\gamma_i^2 n_i^{-2} = \lambda$ gives the required values of n_i for all $i \in [R]$.

7.2.1.4 Proof that $\mathbf{Cov}_{\pi}\left(f\left(X\right),Kf\left(X\right)\right)\geq\mathbf{Cov}_{\pi}\left(f\left(X\right),K^{2}f\left(X\right)\right)$

Defining $g_0 := g - \pi\left(g\right)$ for all $g \in L^2\left(\pi\right)$ we note that $\mathsf{Cov}_{\pi}\left(f\left(X\right),Kf\left(X\right)\right) = \left\langle f_0, \left(Kf_0\right)\right\rangle_{\pi} = \left\langle f_0, Kf_0\right\rangle_{\pi}$ where the final equality is due to the fact that $\pi\left(Kf\right) = \pi\left(f\right)$ when K is π-reversible. Similarly $\mathsf{Cov}_{\pi}\left(f\left(X\right),K^{2}f\left(X\right)\right) = \left\langle f_0, K^2 f_0\right\rangle_{\pi}$. Next we note that

$$\begin{aligned}\left\langle f_0, Kf_0\right\rangle_{\pi} - \left\langle f_0, K^2 f_0\right\rangle_{\pi} &= \left\langle f_0, \left(\mathsf{Id} - K\right)Kf_0\right\rangle_{\pi}\\ &= \left\langle\left(\mathsf{Id} - K\right)f_0, Kf_0\right\rangle_{\pi}\\ &= \frac{1}{4}\left(\left\|f_0\right\|_2^2 - \left\|\left(\mathsf{Id} - 2K\right)f_0\right\|_2^2\right)\end{aligned}$$

where we use the polarisation identity in the final line. That $\mathsf{Cov}_{\pi}\left(f\left(X\right),Kf\left(X\right)\right) \geq 0$ dictates that the numerical range of the Markov operator K on $L_0^2\left(\pi\right)$ is contained entirely within $\left[0,1\right]$. Therefore [Shapiro 2003] dictates that the spectrum of K is contained entirely in $\left[0,1\right]$. This means that the spectrum of $\mathsf{Id} - 2K$ is contained in $\left[-1,1\right]$, and therefore so is the spectrum of $\left(\mathsf{Id} - 2K\right)^2$. Finally, we have that

$$\begin{aligned}\frac{\left\|\left(\mathsf{Id} - 2K\right)f_0\right\|_2^2}{\left\|f_0\right\|_2^2} &= \left\langle\frac{f_0}{\left\|f_0\right\|_2}, \left(\mathsf{Id} - 2K\right)^2\frac{f_0}{\left\|f_0\right\|_2}\right\rangle\\ &\leq \left\|\left(\mathsf{Id} - 2K\right)^2\right\| \leq 1\end{aligned}$$

7.2.2 Proofs from chapter 2

7.2.2.1 Proof of proposition 25

Let $L^{\dagger} \in \mathbb{R}^{d\times d}$ be as in the proposition statement. It is invertible with inverse $\left(L^{\dagger}\right)^{-1} = VD^{-1}V^{T}$. If we precondition with $L = UDV^{T}$ where $U \in O\left(d\right)$ contains the left singular vectors of L, the first operator norm in 2.5 is

$$\begin{aligned}\left\|L^{-T}\nabla^2 U\left(x\right)L^{-1}\right\|_2 &= \left\|UD^{-1}V^T\nabla^2 U\left(x\right)VD^{-1}U^T\right\|_2\\ &= \left\|D^{-1}V^T\nabla^2 U\left(x\right)VD^{-1}\right\|_2\\ &= \left\|VD^{-1}V^T\nabla^2 U\left(x\right)VD^{-1}V^T\right\|_2\\ &= \left\|\left(L^\dagger\right)^{-T}\nabla^2 U\left(x\right)\left(L^\dagger\right)^{-1}\right\|_2\end{aligned}$$

and the second operator norm in 2.5 is

$$\begin{aligned}\left\|L\nabla^2 U\left(x\right)^{-1}L^T\right\|_2 &= \left\|UDV^T\nabla^2 U\left(x\right)^{-1}V^TDU\right\|_2\\ &= \left\|DV^T\nabla^2 U\left(x\right)^{-1}V^TD\right\|_2\\ &= \left\|VDV^T\nabla^2 U\left(x\right)^{-1}V^TDV\right\|_2\\ &= \left\|L^\dagger\nabla^2 U\left(x\right)^{-1}\left(L^\dagger\right)^T\right\|_2\end{aligned}$$

7.2.2.2 Proof of proposition 26

We first define the notion of isomorphism between Markov chains.

Definition 77. [L. T. Johnson and Geyer 2012, Appendix A] Markov chains on state spaces $(\mathsf{X},\mathcal{X})$ and $(\mathsf{X}',\mathcal{X}')$ are isomorphic if there is an invertible bimeasurable mapping $h:\mathsf{X}\to\mathsf{X}'$ such that the corresponding initial distributions π_0 and π_0' and the transition probability kernels K and K' satisfy $\pi_0=\pi_0'\circ h$ and

$$K\left(x\to A\right)=K'\left(h\left(x\right)\to h\left(A\right)\right)$$

for all $x\in\mathsf{X}$ and $A\in\mathcal{X}$.

An invertible mapping h is bimeasurable if h and h^{-1} are measurable. Now we prove a lemma guaranteeing isomorphism between generic accept-reject chains. Define the kernel $K:\mathsf{X}\times\mathcal{X}\to[0,1]$ of an accept-reject chain as follows:

$$K\left(x\to A\right):=\int_{A}q\left(x\to dy\right)\alpha\left(x\to y\right)+\left(1-\int_{\mathsf{X}}q\left(x\to dy\right)\alpha\left(x\to y\right)\right)\delta_{x}\left(A\right)$$

where $q:\mathsf{X}\times\mathcal{X}\to[0,1]$ is a Markov kernel defining the 'proposal' and $\alpha:\mathsf{X}\times\mathsf{X}\to[0,1]$ is an acceptance probability, and we define $K':\mathsf{X}'\times\mathcal{X}'\to[0,1]$ similarly using $q':\mathsf{X}'\times\mathcal{X}'\to[0,1]$ and $\alpha':\mathsf{X}'\times\mathsf{X}'\to[0,1]$.

Lemma 78. *Let $T^{-1}:\mathsf{X}\to\mathsf{X}'$ be an invertible bimeasurable map with Jacobian $J\left(T^{-1}\right)(y)$ for all $y\in\mathsf{X}$. If $q\left(x\to dy\right)=\left|\det J\left(T^{-1}\right)(y)\right|q'\left(T^{-1}\left(x\right)\to T^{-1}\left(dy\right)\right)$ and $\alpha\left(x\to y\right)=\alpha'\left(T^{-1}\left(x\right)\to T^{-1}\left(y\right)\right)$ for all $x,y\in\mathsf{X}$ then the chains generated by K and K' are isomorphic.*

Proof. We examine the first term in $K\left(x\to A\right)$:

$$\int_{A}q\left(x\to dy\right)\alpha\left(x\to y\right)=\int_{A}\left|\det J\left(T^{-1}\right)(y)\right|q'\left(T^{-1}\left(x\right)\to T^{-1}\left(y\right)\right)\alpha'\left(T^{-1}\left(x\right)\to T^{-1}\left(y\right)\right)$$

for all $x\in\mathsf{X}$ and $A\in\mathcal{X}$, and make the change of variables $z=T^{-1}\left(y\right)$

$$\begin{aligned}\int_{A}q\left(x\to dy\right)\alpha\left(x\to y\right)&=\int_{A}\left|\det J\left(T^{-1}\right)(y)\right|\left|\det J\left(T\right)(z)\right|q'\left(T^{-1}\left(x\right)\to dz\right)\alpha'\left(T^{-1}\left(x\right)\to z\right)\\&=\int_{T^{-1}(A)}q'\left(T^{-1}\left(x\right)\to dz\right)\alpha'\left(T^{-1}\left(x\right)\to z\right)\end{aligned}$$

This statement is true for all $A\in\mathcal{X}$ and hence it is true for $A=\mathsf{X}$. Noting that $\delta_{x}\left(A\right)=\mathbb{1}\left\{x\in A\right\}=\mathbb{1}\left\{T^{-1}\left(x\right)\in T^{-1}\left(A\right)\right\}=\delta_{T^{-1}(x)}\left(T^{-1}\left(A\right)\right)$ and $T^{-1}\left(\mathsf{X}\right)=\mathsf{X}'$ gives

$$\begin{aligned}K\left(x\to A\right)&:=\int_{A}q\left(x\to dy\right)\alpha\left(x\to y\right)+\left(1-\int_{\mathsf{X}}q\left(x\to dy\right)\alpha\left(x\to y\right)\right)\delta_{x}\left(A\right)\\&=\int_{T^{-1}(A)}q'\left(T^{-1}\left(x\right)\to dz\right)\alpha'\left(T^{-1}\left(x\right)\to z\right)+\left(1-\int_{\mathsf{X}'}q'\left(T^{-1}\left(x\right)\to dz\right)\alpha'\left(T^{-1}\left(x\right)\to z\right)\right)\delta_{T^{-1}(x)}\left(T^{-1}\left(A\right)\right)\\&=K'\left(T^{-1}\left(x\right)\to T^{-1}\left(A\right)\right)\end{aligned}$$

and so the Markov chains generated by K and K' are isomorphic. □

We're dealing with linear transformations, and so we verify the conditions from the lemma above with

$T = A^{\frac{1}{2}}$. For Metropolised Markov chains the acceptance probability usually depends on the proposal distribution, and so we first verify the condition from the lemma for the proposal. Handily, in the case of the unadjusted Markov chains such as LMC and unadjusted HMC, the proposal distribution matches the Metropolised Markov chains, and the acceptance probability is identically 1. This means proving the condition for the proposal distributions immediately gives an isomorphism in the case of the unadjusted chains.

RWM In the case of RWM the proposal distribution with traditional preconditioner $A \in \mathbb{R}^{d\times d}$ is given by $Y = X + \sqrt{\sigma^2}A^{\frac{1}{2}}\xi$ with $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$ and so has proposal density

$$\begin{aligned} q(x \to y) &= (2\pi)^{-\frac{d}{2}} \det(A)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}(y-x)^T A^{-1}(y-x)\right) \\ &= \det\left(A^{-\frac{1}{2}}\right)(2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2\sigma^2}\left\|A^{-\frac{1}{2}}y - A^{-\frac{1}{2}}x\right\|_2^2\right) \\ &= \det\left(A^{-\frac{1}{2}}\right) q'\left(A^{-\frac{1}{2}}x \to A^{-\frac{1}{2}}y\right) \end{aligned}$$

for all $x, y \in \mathbb{R}^d$, where $q' : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is the proposal density of the RWM proposal with traditional preconditioner $\mathbf{I}_d$. The acceptance probability of RWM with traditional preconditioner $A \in \mathbb{R}^{d\times d}$ is given by $\alpha(x \to y) = \min\{1, \pi(y)/\pi(x)\} = \min\{1, \exp(-U(y) + U(x))\}$. Here we have $U(x) = U'\left(A^{-\frac{1}{2}}x\right)$ by definition of the potential U' and so $\alpha(x \to y) = \alpha'\left(A^{-\frac{1}{2}}x \to A^{-\frac{1}{2}}y\right)$ since the normalising constants of the π' terms cancel out. Therefore the conditions of Lemma 78 hold and we have our isomorphism for RWM.

MALA and LMC For MALA the proposal distribution with traditional preconditioner $A \in \mathbb{R}^{d\times d}$ is given by $Y = X - \sigma^2 A\nabla_x U(X) + \sqrt{2\sigma^2}A^{\frac{1}{2}}\xi$ with $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$. Therefore the proposal density is

$$\begin{aligned} q(x \to y) &= (2\pi)^{-\frac{d}{2}} \det(A)^{-\frac{1}{2}} \exp\left(-\frac{1}{4\sigma^2}\left(y - x + \sigma^2 A\nabla_x U(x)\right)^T A^{-1}\left(y - x + \sigma^2 A\nabla_x U(x)\right)\right) \\ &= \det\left(A^{-\frac{1}{2}}\right)(2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{4\sigma^2}\left\|A^{-\frac{1}{2}}y - A^{-\frac{1}{2}}x + \sigma^2\left(A^{\frac{1}{2}}\right)^T \nabla_x U(x)\right\|\right) \end{aligned}$$

for all $x, y \in \mathbb{R}^d$. Noting that $\left(A^{\frac{1}{2}}\right)^T \nabla_x = \nabla_{A^{-\frac{1}{2}}x}$ and $U(x) = U'\left(A^{-\frac{1}{2}}x\right)$ gives that $q(x \to y) = \det\left(A^{-\frac{1}{2}}\right) q'\left(A^{-\frac{1}{2}}x \to A^{-\frac{1}{2}}y\right)$ and so we have the isomorphism for LMC. For the acceptance probability we have that $q(y \to x)/q(x \to y) = q'\left(A^{-\frac{1}{2}}y \to A^{-\frac{1}{2}}x\right)/q'\left(A^{-\frac{1}{2}}x \to A^{-\frac{1}{2}}y\right)$ and $\pi(y)/\pi(x) = \pi'\left(A^{-\frac{1}{2}}y\right)/\pi'\left(A^{-\frac{1}{2}}x\right)$ as in the RWM case. This gives us that $\alpha(x \to y) = \alpha'\left(A^{-\frac{1}{2}}x \to A^{-\frac{1}{2}}y\right)$ and so we have an isomorphism in the MALA case too.

HMC and Unadjusted HMC Hamiltonian dynamics are defined on $\mathbb{R}^{2d}$ since we incorporate a momentum coordinate $p \in \mathbb{R}^d$. In this case linear transformations $T^{-1} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ are defined by

$$T^{-1}\begin{pmatrix} x \\ p \end{pmatrix} = \begin{pmatrix} A^{-\frac{1}{2}} & 0_{d\times d} \\ 0_{d\times d} & \left(A^{\frac{1}{2}}\right)^T \end{pmatrix}\begin{pmatrix} x \\ p \end{pmatrix}$$

and so the determinant of the Jacobian is always 1. The reason for this particular form of T^{-1} is that the momentum naturally lives in the cotangent spaces of the Riemannian manifold in which the position $x \in \mathbb{R}^d$ resides. This is because the Hamiltonian flow is defined over the cotangent bundle, see [Betancourt 2018] for more details. According to [Hirt, M. Titsias, and P. Dellaportas 2021] the map defined by undergoing N leapfrog steps with step size $\sigma^2 > 0$ and traditional preconditioner $A \in \mathbb{R}^{d\times d}$ is

$$\begin{aligned} y &= \mathcal{T}_{N,x}(\xi) = x - \frac{N\sigma^2}{2}A\nabla_x U(x) + L\sqrt{\sigma^2}A^{\frac{1}{2}}\xi - \sigma^2 A \Xi_{N,x}(\xi) \\ q &= \mathcal{W}_{N,x}(\xi) = \left(A^{-\frac{1}{2}}\right)^T \xi - \frac{\sqrt{\sigma^2}}{2}\left(\nabla_x U(x) + \nabla_x U \circ \mathcal{T}_{N,x}(\xi)\right) - \sqrt{\sigma^2}\sum_{i=1}^{N-1}\nabla_x U \circ \mathcal{T}_{i,x}(\xi) \end{aligned} \tag{7.4}$$

where $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$ and $\Xi_{N,x}(\xi) = \sum_{i=1}^{N-1}(L-i)\nabla_x U \circ \mathcal{T}_{i,x}(\xi)$. Pre-multiplying the first equation by $A^{-\frac{1}{2}}$ and the second by $\left(A^{\frac{1}{2}}\right)^T$ gives

$$\begin{aligned} A^{-\frac{1}{2}}y &= A^{-\frac{1}{2}}\mathcal{T}_{N,x}(\xi) = A^{-\frac{1}{2}}x - \frac{N\sigma^2}{2}\left(A^{\frac{1}{2}}\right)^T\nabla_x U(x) + L\sqrt{\sigma^2}\xi - \sigma^2\sum_{i=1}^{N-1}(L-i)\left(A^{\frac{1}{2}}\right)^T\nabla_x U \circ \mathcal{T}_{i,x}(\xi) \\ \left(A^{\frac{1}{2}}\right)^T q &= \left(A^{\frac{1}{2}}\right)^T \mathcal{W}_{N,x}(\xi) = \xi - \frac{\sqrt{\sigma^2}}{2}\left(\left(A^{\frac{1}{2}}\right)^T\nabla_x U(x) + \left(A^{\frac{1}{2}}\right)^T\nabla_x U \circ \mathcal{T}_{N,x}(\xi)\right) - \sqrt{\sigma^2}\sum_{i=1}^{N-1}\left(A^{\frac{1}{2}}\right)^T\nabla_x U \circ \mathcal{T}_{i,x}(\xi) \end{aligned}$$

noting that $\left(A^{\frac{1}{2}}\right)^T \nabla_x = \nabla_{A^{-\frac{1}{2}}x}$ and $U\left(x\right) = U'\left(A^{-\frac{1}{2}}x\right)$ gives

$$A^{-\frac{1}{2}}y = A^{-\frac{1}{2}}\mathcal{T}_{N,x}\left(\xi\right) = A^{-\frac{1}{2}}x - \frac{N\sigma^2}{2}\nabla_{A^{-\frac{1}{2}}x}U'\left(A^{-\frac{1}{2}}x\right) + L\sqrt{\sigma^2}\xi - \sigma^2\sum_{i=1}^{N-1}\left(L-i\right)\nabla_{A^{-\frac{1}{2}}x}U'\circ A^{-\frac{1}{2}}\mathcal{T}_{i,x}\left(\xi\right)$$

$$\left(A^{\frac{1}{2}}\right)^T q = \left(A^{\frac{1}{2}}\right)^T \mathcal{W}_{N,x}\left(\xi\right) = \xi - \frac{\sqrt{\sigma^2}}{2}\left(\nabla_{A^{-\frac{1}{2}}x}U'\left(A^{-\frac{1}{2}}x\right) + \nabla_{A^{-\frac{1}{2}}x}U'\circ A^{-\frac{1}{2}}\mathcal{T}_{N,x}\left(\xi\right)\right) - \sqrt{\sigma^2}\sum_{i=1}^{N-1}\nabla_{A^{-\frac{1}{2}}x}U'\circ A^{-\frac{1}{2}}\mathcal{T}_{i,x}\left(\xi\right)$$

We can read off from the equations above that if $A^{-\frac{1}{2}}\mathcal{T}_{i,x}\left(\xi\right) = \mathcal{T}'_{i,A^{-\frac{1}{2}}x}\left(\xi\right)$ for all $i \in [N-1]$, where $\mathcal{T}'$ is simply the map $\mathcal{T}$ with U replaced by U' and A replaced by $\mathbf{I}_d$, then $A^{-\frac{1}{2}}\mathcal{T}_{N,x}\left(\xi\right) = \mathcal{T}'_{N,A^{-\frac{1}{2}}x}\left(\xi\right)$ and $\left(A^{\frac{1}{2}}\right)^T \mathcal{W}_{N,x}\left(\xi\right) = \mathcal{W}'_{N,A^{-\frac{1}{2}}x}\left(\xi\right)$, where $\mathcal{W}'$ is defined similarly to $\mathcal{T}'$. The case with $N = 1$ has been handled, since this is just the MALA proposal. Therefore the statement is true for all $N \in \mathbb{N}\setminus\left\{0\right\}$ by induction.

A final fact to note is that the proposal density $q\left(\left(x,p\right) \to \left(y,q\right)\right)$ is independent of p because the momentum is resampled before any leapfrog steps. We then have

$$\begin{aligned}
q\left(\left(x,p\right) \to \left(y,q\right)\right) &= q\left(x \to \left(y,q\right)\right) \\
&= \int_{\mathbb{R}^d} \mathbb{1}\left\{\left(y,q\right) = \left(\mathcal{T}_{N,x}\left(\xi\right), \mathcal{W}_{N,x}\left(\xi\right)\right)\right\}\mathcal{N}\left(d\xi; 0, \mathbf{I}_d\right) \\
&= \int_{\mathbb{R}^d} \mathbb{1}\left\{\left(A^{-\frac{1}{2}}y, \left(A^{\frac{1}{2}}\right)^T q\right) = \left(A^{-\frac{1}{2}}\mathcal{T}_{N,x}\left(\xi\right), \left(A^{\frac{1}{2}}\right)^T \mathcal{W}_{N,x}\left(\xi\right)\right)\right\}\mathcal{N}\left(d\xi; 0, \mathbf{I}_d\right) \\
&= \int_{\mathbb{R}^d} \mathbb{1}\left\{\left(A^{-\frac{1}{2}}y, \left(A^{\frac{1}{2}}\right)^T q\right) = \left(\mathcal{T}'_{N,A^{-\frac{1}{2}}x}\left(\xi\right), \mathcal{W}'_{N,A^{-\frac{1}{2}}x}\left(\xi\right)\right)\right\}\mathcal{N}\left(d\xi; 0, \mathbf{I}_d\right) \\
&= q'\left(A^{-\frac{1}{2}}x \to \left(A^{-\frac{1}{2}}y, \left(A^{\frac{1}{2}}\right)^T q\right)\right) \\
&= q'\left(\left(A^{-\frac{1}{2}}x, \left(A^{\frac{1}{2}}\right)^T p\right) \to \left(A^{-\frac{1}{2}}y, \left(A^{\frac{1}{2}}\right)^T q\right)\right)
\end{aligned}$$

where the final line comes from the independence of q' from the initial momentum. Therefore the condition on the proposal distribution from Lemma 78 holds, and the isomorphism holds of unadjusted HMC. The last thing we need to check is the condition from Lemma 78 on the acceptance probability. Here we have

$$\begin{aligned}\alpha\left((x,p)\rightarrow(y,q)\right) &= \min\left\{1,\exp\left(-H\left(y,q\right)+H\left(x,p\right)\right)\right\}\\ &= \min\left\{1,\exp\left(-U\left(y\right)-\frac{1}{2}q^TAq+U\left(x\right)+\frac{1}{2}p^TAp\right)\right\}\\ &= \min\left\{1,\exp\left(-U'\left(A^{-\frac{1}{2}}y\right)-\frac{1}{2}\left\|\left(A^{\frac{1}{2}}\right)^Tq\right\|_2^2+U'\left(A^{-\frac{1}{2}}x\right)+\frac{1}{2}\left\|\left(A^{\frac{1}{2}}\right)^Tp\right\|_2^2\right)\right\}\\ &= \alpha'\left(\left(A^{-\frac{1}{2}}x,\left(A^{\frac{1}{2}}\right)^Tp\right)\rightarrow\left(A^{-\frac{1}{2}}y,\left(A^{\frac{1}{2}}\right)^Tq\right)\right)\end{aligned}$$

and so the isomorphism holds of HMC.

7.2.2.3 Proof of the form of the ELBO maximising marginal in CAVI ascent

We reproduce the proof from [Blei, Kucukelbir, and McAuliffe 2017]. Let's say we are maximising the ELBO with respect to the ith marginal variational component. Therefore we state the ELBO as a function solely of θ_i:

$$\begin{aligned}\text{ELBO}\left(\theta_i\right) &= \mathbb{E}_{\nu_\theta}\left[\log\pi\left(X,y\right)\right]-\mathbb{E}_{\nu_\theta}\left[\log\nu_\theta\left(X\right)\right]\\ &= \mathbb{E}_{\nu_\theta}\left[\log\pi\left(X,y\right)\right]-\mathbb{E}_{\nu_{\theta_i}^{(i)}}\left[\log\nu_{\theta_i}^{(i)}\left(X_i\right)\right]+\text{const.}\\ &= \mathbb{E}_{\nu_{\theta_i}^{(i)}}\left[\mathbb{E}_{\nu_{\theta_{-i}}^{-(i)}}\left[\log\pi\left(X_i,X_{-i},y\right)|X_i\right]\right]-\mathbb{E}_{\nu_{\theta_i}^{(i)}}\left[\log\nu_{\theta_i}^{(i)}\left(X_i\right)\right]+\text{const.}\end{aligned}$$

where the second line uses the decomposition of the variational density, and the final line uses the law of total expectation. The collection of the first two terms on the right hand side is the negative KL between $\nu_{\theta_i}^{(i)}$ and the ELBO maximising marginal, as stated in 2.10.

7.2.3 Proofs from chapter 3

7.2.3.1 Proof of Proposition 27

The Hessian of the model with potential

$$U(x,y) = \frac{m-M}{2}(\cos x + \cos y) + \frac{M+m}{2}\left(\frac{x^2}{2} + \frac{y^2}{2}\right)$$

is in the form $\nabla^2 U(x,y) = \text{diag}\{f(x), f(y)\}$ where $f(x) := (1/2)(M-m)\cos x + (1/2)(M+m) \in [m, M]$. As detailed in Proposition 25, the condition number is ignorant as to whether the preconditioner L is symmetric or not, so we assume it is. Therefore we can perform an eigendecomposition $L = QDQ^T$ where $D = \text{diag}\{\lambda_1, \lambda_2\}$ is the matrix of eigenvalues (not necessarily ordered) and, since we are in two dimensions, Q can be represented as the two dimensional Givens matrix

$$Q = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The matrix enclosed by the first operator norm in (2.5) has trace and determinant

$$\begin{aligned} \text{Tr}(x,y) &:= \text{Tr}(L^{-T}\nabla^2 U(x,y)L^{-1}) = c^2(\lambda_1^{-2} f(x) + \lambda_2^{-2} f(y)) + s^2(\lambda_2^{-2} f(x) + \lambda_1^{-2} f(y)) \\ \text{Det}(x,y) &:= \text{Det}(L^{-T}\nabla^2 U(x,y)L^{-1}) = \lambda_1^{-2}\lambda_2^{-2} f(x) f(y) \end{aligned}$$

where we have abbreviated $c := \cos\theta$, $s := \sin\theta$ for notational simplicity. The matrix enclosed by the second operator norm in (2.5) has trace and determinant

$$\begin{aligned} \text{Tr}^*(x^*,y^*) &:= \text{Tr}(L\nabla^2 U(x^*,y^*)^{-1}L^T) = c^2(\lambda_1^2 f(x^*)^{-1} + \lambda_2^2 f(y^*)^{-1}) + s^2(\lambda_2^{-2} f(x^*)^{-1} + \lambda_1^{-2} f(x^*)^{-1}) \\ \text{Det}^*(x^*,y^*) &:= \text{Det}(L\nabla^2 U(x^*,y^*)^{-1}L^T) = \lambda_1^2\lambda_2^2 f(x^*)^{-1} f(y^*)^{-1} \end{aligned}$$

Using the fact that the operator norm of a positive definite matrix is simply the largest eigenvalue, we are able to lower bound

$$\tilde{\kappa} \geq \frac{1}{2}\left(\mathsf{Tr}(x,y) + \sqrt{\mathsf{Tr}(x,y)^2 - 4\mathsf{Det}(x,y)}\right)\frac{1}{2}\left(\mathsf{Tr}^*(x^*,y^*) + \sqrt{\mathsf{Tr}^*(x^*,y^*)^2 - 4\mathsf{Det}^*(x^*,y^*)}\right)$$

Choosing (x,y) such that $f(x) = f(y) = M$ and (x^*,y^*) such that $f(x^*) = f(y^*) = m$ we have

$$\begin{aligned}\tilde{\kappa} &\geq \frac{1}{2}\left((\lambda_1^{-2} + \lambda_2^{-2})M + \left|\lambda_1^{-2} - \lambda_2^{-2}\right| M\right)\frac{1}{2}\left((\lambda_1^2 + \lambda_2^2)m^{-1} + \left|\lambda_1^2 - \lambda_2^2\right| m^{-1}\right)\\ &= \max\{\lambda_1^{-2}, \lambda_2^{-2}\}\max\{\lambda_1^2, \lambda_2^2\}\frac{M}{m}\\ &= \kappa(LL^T)\kappa\end{aligned}$$

Therefore $\tilde{\kappa} > \kappa$ for non-orthogonal L.

7.2.3.2 Proof of Theorem 30

Perform the eigendecomposition $\nabla^2 U(x) = O_x D_x O_x^T$ for $O_x \in O(d)$ with columns $v_i(x)$ and $D_x \in \mathbb{R}^{d\times d}$ diagonal with elements $\lambda_i(x)$. Perform the eigendecomposition $L = V\Sigma V^T$ where V has columns $v_i \in \mathbb{R}^d$ for $i \in [d]$ and $\Sigma := \mathsf{diag}\{\sigma_1, ..., \sigma_d\}$. Defining $\mathcal{E}_x := V^T O_x - \mathbf{I}_d$, Assumption 29 guarantees that the elements of $\mathcal{E}_x$ are at most δ in absolute value. Inspecting the first term in the definition of $\tilde{\kappa}$, we have that

$$\begin{aligned}\|L^{-T}\nabla^2 U(x) L^{-1}\| &= \|\Sigma^{-1}(\mathcal{E}_x + \mathbf{I}_d)D_x(\mathcal{E}_x + \mathbf{I}_d)^T\Sigma^{-1}\|\\ &\leq \|\Sigma^{-1}\mathcal{E}_x D_x \mathcal{E}_x^T \Sigma^{-1}\| + 2\|\Sigma^{-1}\mathcal{E}_x D_x \Sigma^{-1}\| + \|\Sigma^{-1} D_x \Sigma^{-1}\|\\ &\leq \|\Sigma^{-1}\mathcal{E}_x D_x \mathcal{E}_x^T \Sigma^{-1}\| + 2\|\Sigma^{-1}\mathcal{E}_x D_x \Sigma^{-1}\| + (1+\varepsilon)\end{aligned}$$

where the second line is due to the triangle inequality of the matrix 2-norm, the last line due to Assumption 28. Inspecting the norm in the second term in the above:

$$\begin{aligned}
\|\Sigma^{-1}\mathcal{E}_x D_x \Sigma^{-1}\|^2 &= \sup_{\|v\|=1} \sum_{k=1}^{d} \left(\sum_{s=1}^{d} \frac{\lambda_s(x)}{\sigma_s \sigma_k} (\mathcal{E}_x)_{ks} v_s \right)^2 \\
&\leq \delta^2 \sup_{\|v\|=1} \sum_{k=1}^{d} \left(\sum_{s=1}^{d} \frac{\lambda_s(x)}{\sigma_s \sigma_k} v_s \right)^2 \\
&= \delta^2 \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \sup_{\|v\|=1} \left(\sum_{s=1}^{d} \frac{\lambda_s(x)}{\sigma_s} v_s \right)^2 \\
&\leq \delta^2 (1+\varepsilon)^2 \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \sup_{\|v\|=1} \left(\sum_{s=1}^{d} \sigma_s v_s \right)^2 \\
&= \delta^2 (1+\varepsilon)^2 \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \sum_{s=1}^{d} \sigma_s^2
\end{aligned}$$

where the second line comes from Assumption 29 and the fourth line comes from Assumption 28. Looking at the first term now:

$$\begin{aligned}
\|\Sigma^{-1}\mathcal{E}_x D_x \mathcal{E}_x^T \Sigma^{-1}\| &= \|\Sigma^{-1}\mathcal{E}_x D_x^{\frac{1}{2}}\|^2 \\
&= \sup_{\|v\|=1} \sum_{k=1}^{d} \left(\sum_{s=1}^{d} \frac{\sqrt{\lambda_s(x)}}{\sigma_k} (\mathcal{E}_x)_{ks} v_s \right)^2 \\
&\leq \delta^2 \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \sup_{\|v\|=1} \left(\sum_{s=1}^{d} \sqrt{\lambda_s(x)} v_s \right)^2 \\
&\leq \delta^2 (1+\varepsilon) \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \sup_{\|v\|=1} \left(\sum_{s=1}^{d} \sigma_s v_s \right)^2 \\
&= \delta^2 (1+\varepsilon) \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \sum_{s=1}^{d} \sigma_s^2
\end{aligned}$$

where the third line comes from Assumption 29 and the fourth line comes from Assumption 28. Putting the terms together yields

$$\|L^{-T}\nabla^2 U(x) L^{-1}\| \leq (1+\varepsilon) \left(1 + \delta \sqrt{\sum_{i=1}^{d} \sigma_i^2 \sum_{i=1}^{d} \sigma_i^{-2}} \right)^2$$

Now we follow the same procedure for $\|L\nabla^2 U(x) L^T\|$:

$$\begin{aligned}\|L\nabla^2 U(x) L^T\| &\leq \|\Sigma\mathcal{E}_x D_x^{-\frac{1}{2}}\|^2 + 2\|\Sigma\mathcal{E}_x D_x^{-1}\Sigma\| + \|\Sigma D_x^{-1}\Sigma\| \\ &\leq \|\Sigma\mathcal{E}_x D_x^{-\frac{1}{2}}\|^2 + 2\|\Sigma\mathcal{E}_x D_x^{-1}\Sigma\| + (1+\varepsilon)\end{aligned}$$

starting with the second term:

$$\begin{aligned}\|\Sigma\mathcal{E}_x D_x^{-1}\Sigma\|^2 &= \sup_{\|v\|=1}\sum_{k=1}^{d}\left(\sum_{s=1}^{d}\frac{\sigma_s\sigma_k}{\lambda_s(x)}(\mathcal{E}_x)_{ks}v_s\right)^2 \\ &\leq \delta^2\sum_{k=1}^{d}\sigma_k^2\sup_{\|v\|=1}\left(\sum_{s=1}^{d}\frac{\sigma_s}{\lambda_s(x)}v_s\right)^2 \\ &\leq \delta^2(1+\varepsilon)^2\sum_{k=1}^{d}\sigma_k^2\sup_{\|v\|=1}\left(\sum_{s=1}^{d}\frac{1}{\sigma_s}v_s\right)^2 \\ &\leq \delta^2(1+\varepsilon)^2\sum_{k=1}^{d}\sigma_k^2\sum_{s=1}^{d}\frac{1}{\sigma_s^2}\end{aligned}$$

and the first term:

$$\begin{aligned}\|\Sigma\mathcal{E}_x D_x^{-\frac{1}{2}}\|^2 &= \sup_{\|v\|=1}\sum_{k=1}^{d}\left(\sum_{s=1}^{d}\frac{\sigma_k}{\sqrt{\lambda_s(x)}}(\mathcal{E}_x)_{ks}v_s\right)^2 \\ &\leq \delta^2\sum_{k=1}^{d}\sigma_k^2\sup_{\|v\|=1}\left(\sum_{s=1}^{d}\frac{1}{\sqrt{\lambda_s(x)}}v_s\right)^2 \\ &\leq \delta^2(1+\varepsilon)\sum_{k=1}^{d}\sigma_k^2\sup_{\|v\|=1}\left(\sum_{s=1}^{d}\frac{1}{\sigma_s}v_s\right)^2 \\ &= \delta^2(1+\varepsilon)\sum_{k=1}^{d}\sigma_k^2\sum_{s=1}^{d}\frac{1}{\sigma_s^2}\end{aligned}$$

from which follows

$$\tilde{\kappa} \leq (1+\varepsilon)^2\left(1+\delta\sqrt{\sum_{i=1}^{d}\sigma_i^2\sum_{i=1}^{d}\sigma_i^{-2}}\right)^4$$

7.2.3.3 Implications from assumption 29

From the statement of Assumption 29 it is immediate that $v_i(x)^T v_i \geq 1 - \delta$. Note that the assumption implies the following bound $\|v_i(x) - v_i\| \leq \sqrt{2}\left(1 - \sqrt{1-\delta}\right)$. For $i, j \in [d]$ such that $i \neq j$, the reverse triangle inequality gives us that

$$\begin{aligned}\|v_i(x) - v_j\| &\geq \|v_j - v_i\| - \|v_i - v_i(x)\| \\ &\geq \sqrt{2} - \sqrt{2}\left(1 - \sqrt{1-\delta}\right)\end{aligned}$$

and so

$$\sqrt{2(1 - \langle v_i(x), v_j\rangle} \geq \sqrt{2} - \sqrt{2}\left(1 - \sqrt{1-\delta}\right)$$

hence $v_i(x)^T v_j \leq \delta$ as required.

7.2.3.4 Proof of Theorem 35

Based on the intuition gained from Proposition 25 we can assume that L is symmetric, and so its left and right singular vectors are simply its eigenvectors. Using [Yu, Wang, and Samworth 2015] with $\hat{\Sigma} = \nabla^2 U(x)$ and $\Sigma = LL^T$ we have that $\|v_i(x) - v_i\| \leq 2^{\frac{3}{2}}\gamma^{-1}\|\nabla^2 U(x) - LL^T\|$. Rearranging, the Assumption 33 gives $\langle v_i(x), v_i\rangle \geq 1 - 4\gamma^{-2}\sigma_d^{-4}\varepsilon^2$. From 7.2.3.5, Assumption 33 gives us Assumption 28 with the same ε, and hence we can apply Theorem 30 with $\delta = 1 - (1 - 2\gamma^{-1}\sigma_d^{-2}\varepsilon)^2$.

7.2.3.5 Proof that 33 implies 28

Weyl's inequality implies that

$$\frac{\lambda_i(\nabla^2 U(x))}{\sigma_i^2} \leq \frac{\lambda_i(LL^T) + \lambda_1(\nabla^2 U(x) - LL^T)}{\sigma_i^2} \leq 1 + \frac{\|\nabla^2 U(x) - LL^T\|}{\sigma_i^2}$$

and so $\|\nabla^2 U(x) - LL^T\| \leq \sigma_d^2\varepsilon$ implies Assumption 28 with the same ε.

7.2.3.6 Proof of Theorem 38

Using Proposition 25 we assume that L is symmetric. For the first supremum in the definition of $\tilde{\kappa}$ note that $\|\nabla^2 U(x) - L^2\| = \|L^T(L^{-T}\nabla^2 U(x)L^{-1} - \mathbf{I}_d)L\|$. Using the fact that $\sigma_i(BA) \leq \|B\|\sigma_i(A)$ and $\sigma_i(AC) \leq \sigma_i(A)\|C\|$ for matrices A, B, C of appropriate sizes and all $i \in [d]$ [Tao 2012, Exercise 1.3.24] we have

$$\begin{aligned}
\|L^T(L^{-T}\nabla^2 U(x)L^{-1} - \mathbf{I}_d)L\| &\geq \frac{\sigma_1(L^T(L^{-T}\nabla^2 U(x)L^{-1} - \mathbf{I}_d)LL^{-1})}{\sigma_1(L^{-1})} \\
&= \sigma_d(L)\|L^T(L^{-T}\nabla^2 U(x)L^{-1} - \mathbf{I}_d)\| \\
&\geq \sigma_d(L)\frac{\sigma_1(L^{-T}L^T(L^{-T}\nabla^2 U(x)L^{-1} - \mathbf{I}_d))}{\sigma_1(L^{-T})} \\
&= \sigma_d(L)^2\|L^{-T}\nabla^2 U(x)L^{-1} - \mathbf{I}_d\|
\end{aligned}$$

Therefore we can bound $\|L^{-T}\nabla^2 U(x)L^{-1} - \mathbf{I}_d\| \leq \varepsilon$ using Assumption 33. Using the reverse triangle inequality $\|L^{-T}\nabla^2 U(x)L^{-1} - \mathbf{I}_d\| \geq |\|L^{-T}\nabla^2 U(x)L^{-1}\| - 1|$ we get $\|L^{-T}\nabla^2 U(x)L^{-1}\| \leq 1 + \varepsilon$.

For the second supremum in the definition of $\tilde{\kappa}$ we use the same technique as the first supremum, first noting that $\|\nabla^2 U(x)^{-1} - L^{-2}\| \leq \|\nabla^2 U(x)^{-1}\|\|L^{-2}\|\|\nabla^2 U(x) - L^2\| \leq m^{-1}\varepsilon$. Employing the technique from before:

$$\begin{aligned}
\|\nabla^2 U(x)^{-1} - L^{-2}\| &= \|L^{-1}(L\nabla^2 U(x)^{-1}L^T - \mathbf{I}_d)L^{-T}\| \\
&\geq \sigma_d(L^{-1})^2\|L\nabla^2 U(x)^{-1}L^T - \mathbf{I}_d\|
\end{aligned}$$

and hence $\|L\nabla^2 U(x)^{-1}L^T - \mathbf{I}_d\| \leq \sigma_1(L)^2 m^{-1}\varepsilon$. Using the reverse triangle inequality again gives $\|L\nabla^2 U(x)^{-1}L^T\| \leq 1 + \sigma_1(L)^2 m^{-1}\varepsilon$.

7.2.3.7 Proof of proposition 39

The assumption 33 has that $\left\|\nabla^2 U\left(x\right) - LL^T\right\|_2 \leq \sigma_d^2 \varepsilon$ for all $x \in \mathbb{R}^d$. Therefore

$$\begin{aligned}\left\|\nabla^2 U\left(x\right) - \mathcal{I}\right\|_2 &= \left\|\nabla^2 U\left(x\right) - LL^T + LL^T - \mathcal{I}\right\|_2 \\ &\leq \sigma_d^2 \varepsilon + \left\|\mathbb{E}_\pi\left[\nabla^2 U\left(X\right) - LL^T\right]\right\|_2 \\ &\leq \sigma_d^2 \varepsilon + \mathbb{E}_\pi\left[\left\|\nabla^2 U\left(X\right) - LL^T\right\|_2\right] \\ &\leq 2\sigma_d^2 \varepsilon\end{aligned}$$

where the penultimate line is due to Jensen's inequality.

7.2.3.8 Proof of proposition 42

We assume WLOG that $U(x^*) = 0$. Taylor's theorem with integral remainder has that

$$U(x) = \int_0^1 (1-t)(x - x^*)^T \nabla^2 U(x^* + t(x - x^*))(x - x^*) dt$$

(since $U(x^*) = 0$, $\nabla U(x^*) = 0$) from which we can deduce

$$\frac{1}{2}(x - x^*)^T \Delta_-(x - x^*) \leq U(x) \leq \frac{1}{2}(x - x^*)^T \Delta_+(x - x^*)$$

and hence

$$\exp\left(-\frac{1}{2}(x - x^*)^T \Delta_+(x - x^*)\right) \leq \exp(-U(x)) \leq \exp\left(-\frac{1}{2}(x - x^*)^T \Delta_-(x - x^*)\right)$$

with

$$\frac{Z_{\Delta_+}}{Z}\frac{1}{Z_{\Delta_+}}\exp\left(-\frac{1}{2}(x - x^*)^T \Delta_+(x - x^*)\right) \leq \frac{1}{Z}\exp(-U(x)) \leq \frac{Z_{\Delta_-}}{Z}\frac{1}{Z_{\Delta_-}}\exp\left(-\frac{1}{2}(x - x^*)^T \Delta_-(x - x^*)\right)$$

where $Z_A := \sqrt{(2\pi)^d \det A^{-1}}$. For an arbitrary $v \in \mathbb{R}^d$ we have that

$$\begin{aligned} v^T\Sigma_\pi v &= \frac{1}{Z}\int (v^T(x-\mu_\pi))^2 \exp(-U(x))dx \\ &\leq \frac{Z_{\Delta_-}}{Z}\mathbb{E}_{\mathcal{N}(x^*,(\Delta_-)^{-1})}[(v^T(X-\mu_\pi))^2] \\ &\leq \frac{Z_{\Delta_-}}{Z}\left(\mathbb{E}_{\mathcal{N}(x^*,(\Delta_-)^{-1})}[(v^T(X-x^*))^2] - (v^T(x^*-\mu_\pi))^2\right) \\ &\leq \frac{Z_{\Delta_-}}{Z_{\Delta_+}}\left(v^T(\Delta_-)^{-1}v - v^T(x^*-\mu_\pi)(x^*-\mu_\pi)^T v\right) \end{aligned}$$

where the last inequality follows from the fact that $Z_{\Delta_+} \leq Z \leq Z_{\Delta_-}$. We can construct a similar lower bound to give

$$c((\Delta_+)^{-1} - (x^*-\mu_\pi)(x^*-\mu_\pi)^T) \leq \Sigma_\pi \leq c^{-1}((\Delta_-)^{-1} - (x^*-\mu_\pi)(x^*-\mu_\pi)^T)$$

defining $c := (Z_{\Delta_+}/Z_{\Delta_-}) = \sqrt{\det\Delta_- \det\Delta_+^{-1}} \leq 1$. This gives $P_- \leq \Sigma_\pi^{-1} \leq P_+$ where

$$\begin{aligned} P_+ &:= c^{-1}\left((\Delta_+)^{-1} - (x^*-\mu_\pi)(x^*-\mu_\pi)^T\right)^{-1} \\ P^- &:= c\left((\Delta_-)^{-1} - (x^*-\mu_\pi)(x^*-\mu_\pi)^T\right)^{-1} \end{aligned}$$

and hence

$$\begin{aligned} P_+ &= c^{-1}\Big(\Delta_+ + \left(1-(x^*-\mu_\pi)^T\Delta_+(x^*-\mu_\pi)\right)^{-1}\Delta_+(x^*-\mu_\pi)(x^*-\mu_\pi)^T\Delta_+\Big) \\ P_- &= c\Big(\Delta_- + \left(1-(x^*-\mu_\pi)^T\Delta_-(x^*-\mu_\pi)\right)^{-1}\Delta_-(x^*-\mu_\pi)(x^*-\mu_\pi)^T\Delta_-\Big) \end{aligned}$$

using the Woodbury identity. The fact that $(x^*-\mu_\pi)^T\Delta_\pm(x^*-\mu_\pi) = \mathrm{Tr}(D_\pm)$ gives the result.

We have that $\|\nabla^2 U(x) - \Sigma_\pi^{-1}\| := \sup_v \left|v^T\nabla^2 U(x)v - v^T\Sigma_\pi^{-1}v\right|$. Say the quantity inside the absolute value is positive. Then $v^T\nabla^2 U(x)v - v^T\Sigma_\pi^{-1}v \leq v^T\nabla^2 U(x)v - v^T P_- v$. Now say the quantity is negative, giving us $v^T\Sigma_\pi^{-1}v - v^T\nabla^2U(x)v \le v^TP_+v - v^T\nabla^2U(x)v$. In sum this gives

$$\begin{aligned}\|\nabla^2U(x) - \Sigma_\pi^{-1}\| &\le \sup_{v:\|v\|=1} \max\left\{v^T\nabla^2U(x)v - v^TP_-v, v^TP_+v - v^T\nabla^2U(x)v\right\}\\ &\le \sup_{v:\|v\|=1} \max\left\{v^T\Delta_+v - v^TP_-v, v^TP_+v - v^T\Delta_-v\right\}\\ &\le \max\left\{\|\Delta_+ - P_-\|, \|P_+ - \Delta_-\|\right\}\end{aligned}$$

7.2.3.9 Proof of corollary 43

From proposition 42 we have that $P_- \le \Sigma_\pi^{-1} \le P_+$where

$$\begin{aligned}P_+ &= c^{-1}\left(A + \epsilon\mathbf{I}_d + \left(1 - (x^* - \mu_\pi)^T(A + \epsilon\mathbf{I}_d)(x^* - \mu_\pi)\right)^{-1}(A + \epsilon\mathbf{I}_d)(x^* - \mu_\pi)(x^* - \mu_\pi)^T(A + \epsilon\mathbf{I}_d)\right)\\ P_- &= c\left(A - \epsilon\mathbf{I}_d + \left(1 - (x^* - \mu_\pi)^T(A - \epsilon\mathbf{I}_d)(x^* - \mu_\pi)\right)^{-1}(A - \epsilon\mathbf{I}_d)(x^* - \mu_\pi)(x^* - \mu_\pi)^T(A - \epsilon\mathbf{I}_d)\right)\end{aligned}$$

since $\Delta_- = A - \epsilon\mathbf{I}_d$ and $\Delta_+ = A + \epsilon\mathbf{I}_d$. The bounds stated at the end of the proposition give

$$\begin{aligned}\|\nabla^2U(x) - \Sigma_\pi^{-1}\| &\le \max\left\{\|\Delta_+ - P_-\|, \|P_+ - \Delta_-\|\right\}\\ &= \max\left\{\|(1-c)A + (1+c)\epsilon\mathbf{I}_d - c\tilde{P}_-\|, \|(c^{-1}-1)A + (c^{-1}+1)\epsilon\mathbf{I}_d - c^{-1}\tilde{P}_+\|\right\}\\ &\le (c^{-1}-1)\|A\| + (c^{-1}+1)\epsilon + \max\left\{c\|\tilde{P}_-\|, c^{-1}\|\tilde{P}_+\|\right\}\end{aligned}$$

where in the final line we use the triangle inequality.

7.2.3.10 Proof of proposition 44

Applying the non-rectangular form of Ostrowski's theorem [Higham and Cheng 1998, Theorem 3.2] gives for any $x \in \mathbb{R}^d$

$$\lambda_d(B^TB)\lambda_{n-d+1}(\Lambda(x)) \le \lambda_1(B^T\Lambda(x)B) \le \lambda_1(B^TB)\lambda_1(\Lambda(x)),$$

and similarly

$$\lambda_d(B^TB)\lambda_n(\Lambda(x)) \le \lambda_d(B^T\Lambda(x)B) \le \lambda_1(B^TB)\lambda_d(\Lambda(x))$$

Since $\kappa := \sup_{x \in \mathbb{R}^d} \lambda_1(B^T \Lambda(x) B) / \inf_{x \in \mathbb{R}^d} \lambda_d(B^T \Lambda(x) B)$ then applying the upper/lower bound to $\lambda_1(B^T \Lambda(x) B)$ and the lower/upper bound to $\lambda_d(B^T \Lambda(x) B)$ point-wise gives the upper/lower bound on κ as desired.

7.2.3.11 Proof of proposition 47

Setting $\tilde{X}^T = (X^T X)^{-1/2} X^T$ and applying proposition 44 gives the result, noting that $\tilde{X}^T \tilde{X} = \mathbf{I}_d$.

7.2.3.12 Proof of proposition 49

First note that the preconditioned Hessian can be written

$$L^{-T} \nabla^2 U(x) L^{-1} = (X^T \Lambda(x^*) X)^{1/2} X^T \Lambda(x^*)^{1/2} \Lambda(x^*)^{-1/2} \Lambda(x) \Lambda(x^*)^{-1/2} \Lambda(x^*)^{1/2} X (X^T \Lambda(x^*) X)^{1/2}$$

Setting $\tilde{X}^T := (X^T \Lambda(x^*) X)^{1/2} X^T \Lambda(x^*)^{1/2}$ and then applying the upper bound of Proposition 44 to the matrix $\tilde{X}^T \Lambda(x^*)^{-1/2} \Lambda(x) \Lambda(x^*)^{-1/2} \tilde{X}$ gives the first inequality. The second follows from applying the same bound again to $\Lambda(x^*)^{-1/2} \Lambda(x) \Lambda(x^*)^{-1/2}$ and noting that

$$\kappa(\Lambda(x^*)) \leq \frac{\sup_{x \in \mathbb{R}^d} \lambda_1(\Lambda(x))}{\inf_{x \in \mathbb{R}^d} \lambda_d(\Lambda(x))}$$

7.2.3.13 Proof of Theorem 52

If we take $\sigma^2 = \xi/(Md)$, then [Andrieu, A. Lee, et al. 2024, Theorem 1] implies that the spectral gap γ_κ of the RWM algorithm on a target with a m-strongly convex, M-smooth potential is bounded as follows:

$$C \xi \exp(-2\xi) \frac{1}{\kappa} \frac{1}{d} \leq \gamma_\kappa \leq \frac{\xi}{2} \frac{1}{d}$$

We will modify the proof of [Andrieu, A. Lee, et al. 2024, Lemma 47] so that the upper bound on γ_κ subsequently depends on κ. The spectral gap of the RWM algorithm on a target π with kernel P is defined as $\gamma_k := \inf_{f \in L^2_0(\pi)} (\mathcal{E}(P, f)/\text{Var}_\pi(f))$ where $\mathcal{E}$ is the Dirichlet form associated with π. Define $g(x) := \langle v_{\text{max}}, x - \mathbb{E}_\pi[X] \rangle$ where $v_{\text{max}} \in \mathbb{R}^d$ is the eigenvector associated with the greatest eigenvalue of

$E_\pi \left(\nabla^2 U(X)\right)^{-1}$. The Cramér-Rao inequality gives that

$$\begin{aligned}
\mathsf{Var}_\pi(g(X)) &\geq v_{\mathsf{max}}^T \mathbb{E}_\pi \left[\nabla^2 U(X)\right]^{-1} v_{\mathsf{max}} \\
&= \lambda_1 \left(\mathbb{E}_\pi \left[\nabla^2 U(X)\right]^{-1}\right) \|v_{\mathsf{max}}\|^2 \\
&= \lambda_d \left(\mathbb{E}_\pi \left[\nabla^2 U(X)\right]\right)^{-1} \|v_{\mathsf{max}}\|^2
\end{aligned}$$

The second equality comes from the fact that $\mathbb{E}_\pi \left[\nabla^2 U(X)\right]^{-1}$ is positive definite, since it is the inverse of the expectation of a matrix that is itself positive definite.

Say $v \in \mathbb{R}^d$ is the eigenvector associated with the smallest eigenvalue $\lambda_d(y)$ of $\nabla^2 U(y)$ for a given $y \in \mathbb{R}^d$. Then

$$\begin{aligned}
\lambda_d \left(\mathbb{E}_\pi \left[\nabla^2 U(X)\right]\right) &= \inf_{\|v\|=1} v^T \mathbb{E}_\pi \left[\nabla^2 U(X)\right] v \\
&\leq v^T \mathbb{E}_\pi \left[\nabla^2 U(X)\right] v \\
&= \mathbb{E}_\pi \left[v^T \left(\nabla^2 U(X) - \nabla^2 U(y) + \nabla^2 U(y)\right) v\right] \\
&\leq \sup_{x \in \mathbb{R}^d} \|\nabla^2 U(x) - \nabla^2 U(y)\| + \lambda_d(y) \\
&\leq m(1+2\varepsilon)
\end{aligned}$$

where in the final line we use Assumption 50, and the fact that Assumption 50 implies

$$\frac{\lambda_d(y)}{\lambda_d(x)} \leq 1+$$

for all $x, y \in \mathbb{R}^d$ (see **??**) and hence $\lambda_d(y) \leq (1+\varepsilon)\lambda_d(x) \leq (1+\varepsilon)m$. Therefore $\mathsf{Var}_\pi(g(X)) \geq m^{-1}(1+2\varepsilon)^{-1}\|v_{\mathsf{max}}\|^2$. Upper bounding the Dirichlet form in the same way as [Andrieu, A. Lee, et al. 2024, Lemma 47]gives $\mathcal{E}(P,g) \leq (1/2)\sigma^2\|v_{\mathsf{max}}\|^2$, and so

$$\gamma_k = \inf_{f \in L_0^2(\pi)} \frac{\mathcal{E}(P,f)}{\mathsf{Var}_\pi(f)} \leq \frac{\mathcal{E}(P,g)}{\mathsf{Var}_\pi(g)} \leq \frac{\frac{1}{2}\sigma^2\|v_{\mathsf{max}}\|^2}{m^{-1}(1+2\varepsilon)^{-1}\|v_{\mathsf{max}}\|^2} = \frac{1}{2}\xi\kappa^{-1}d^{-1}(1+2\varepsilon)$$

7.2.3.14 Proof of corollary 53

The lower bound for the spectral gap post-preconditioning is $\gamma_{\tilde{\kappa}} \geq C\xi\exp(-2\xi)\tilde{\kappa}^{-1}d^{-1}$. The target satisfies Assumption 33 so we can use Theorem 38 to modify the bound: $\gamma_{\tilde{\kappa}} \geq C\xi\exp(-2\xi)(1+\varepsilon)^{-1}\left(1+m^{-1}\sigma_1(L)^2\varepsilon\right)^{-1}d^{-1}$. So, applying the upper bound found in Theorem 52 to the spectral gap before preconditioning, we see that a condition number κ such that

$$\frac{1}{2}\xi\kappa^{-1}d^{-1}(1+2\varepsilon') \leq C\xi\exp(-2\xi)(1+\varepsilon)^{-1}\left(1+\frac{\sigma_1(L)^2}{m}\varepsilon\right)^{-1}d^{-1}$$

guarantees that $\gamma_{\tilde{\kappa}} \geq \gamma_\kappa$ and so increases the spectral gap.

7.2.3.15 Full Σ_π matrix from section 3.5

The full matrix from equation 3.5 in section 3.5 is as follows:

$$\Sigma_\pi = \begin{pmatrix} 21.548973 & 5.678587 & 18.667787 & 4.463119 & 6.855300 \\ \star & 2.028958 & 4.863393 & 1.208146 & 2.109502 \\ \star & \star & 16.261735 & 3.926604 & 5.726388 \\ \star & \star & \star & 1.405213 & 1.409477 \\ \star & \star & \star & \star & 2.905902 \end{pmatrix}$$

7.2.4 Proofs from chapter 4

7.2.4.1 Proof of proposition 54

To prove the unbiasedness we have that

$$\begin{aligned}\mathbb{E}[\hat{f}_{\text{ideal}}] &= \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{R}\mathbb{E}\left[\sum_{j=1}^{N_i} f(Y_{ij})\,\big|\{N_i\}_{i=1}^{R}\right]\right]\\ &= \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{R} N_i\mu_i\right]\\ &= \frac{1}{n}\sum_{i=1}^{R}\mathbb{E}[N_i]\mu_i = \frac{1}{n}\sum_{i=1}^{R} n\pi(\mathsf{X}_i)\mu_i\end{aligned}$$

and the result follows. For the variance we use the law of total variance:

$$\text{Var}(\hat{f}_{\text{ideal}}) = \mathbb{E}\left[\text{Var}\left(\hat{f}_{\text{ideal}}\,|\{\rho(X_t)\}_{t=1}^{n}\right)\right] + \text{Var}\left(\mathbb{E}\left[\hat{f}_{\text{ideal}}\,|\{\rho(X_t)\}_{t=1}^{n}\right]\right)$$

where $\rho(X_t)$ is simply the region that X_t is in. Inspecting the second term on the right hand side:

$$\mathbb{E}\left[\hat{f}_{\text{ideal}}\,|\{\rho(X_t)\}_{t=1}^{n}\right] = \frac{1}{n}\sum_{i=1}^{R}\sum_{j=1}^{N_i}\mathbb{E}[f(Y_{ij})\,|\{\rho(X_t)\}_{t=1}^{n}] = \frac{1}{n}\sum_{i=1}^{R}\mu_i N_i$$

and so

$$\begin{aligned}\text{Var}\left(\mathbb{E}\left[\hat{f}_{\text{ideal}}\,|\{\rho(X_t)\}_{t=1}^{n}\right]\right) &= \text{Var}\left(\frac{1}{n}\sum_{i=1}^{R}\mu_i N_i\right)\\ &= \frac{1}{n^2}\sum_{i,i'}^{R}\mu_i\mu_{i'}\text{Cov}\left(N_i, N_{i'}\right)\\ &= \frac{1}{n^2}\sum_{i,i'}^{R}\mu_i\mu_{i'}\sum_{t,t'}^{n}\text{Cov}\left(\mathbb{1}\{X_t\in\mathsf{X}_i\}, \mathbb{1}\{X_t'\in\mathsf{X}_{i'}\}\right)\end{aligned}$$

We have that

$$\sum_{t,t'}^{n} \mathsf{Cov}\left(\mathbb{1}\{X_t \in \mathsf{X}_i\}, \mathbb{1}\{X'_t \in \mathsf{X}_{i'}\}\right) =$$
$$n\mathsf{Cov}_\pi(\mathbb{1}\{X \in \mathsf{X}_i\},\mathbb{1}\{X \in \mathsf{X}_{i'}\}) + 2\sum_{k=1}^{n-1}(n-k)\mathsf{Cov}_\pi\left(K^k\mathbb{1}\{X \in \mathsf{X}_i\}, \mathbb{1}\{X \in \mathsf{X}_{i'}\}\right)$$

where K is the Markov operator of $\{X_t\}_{t=1}^{n}$. Since $\mathsf{Cov}_\pi\left(\mathbb{1}\{X \in \mathsf{X}_i\}, \mathbb{1}\{X \in \mathsf{X}_{i'}\}\right) = \mathbb{1}\{i = i'\}\pi(\mathsf{X}_i) - \pi(\mathsf{X}_i)\pi(\mathsf{X}_j)$ we have that

$$\frac{1}{n^2}\sum_{i,i'}^{R}\mu_i\mu_{i'}\mathsf{Cov}_\pi\left(\mathbb{1}\{X \in \mathsf{X}_i\}, \mathbb{1}\{X \in \mathsf{X}_{i'}\}\right) = \frac{1}{n}\left(\sum_{i=1}^{R}\pi(\mathsf{X}_i)\mu_i^2 - \mu^2\right) = \frac{1}{n}\mathsf{Var}_\pi(\overrightarrow{\pi} f)$$

which gives that

$$\begin{aligned}\mathsf{Var}\left(\mathbb{E}\left[\hat{f}_{\mathsf{ideal}}\,|\{\rho(X_t)\}_{t=1}^{n}\right]\right) &= \frac{1}{n}\mathsf{Var}_\pi(\overrightarrow{\pi} f) + \frac{1}{n}2\sum_{k=1}^{n-1}\frac{n-k}{n}\mathsf{Cov}_\pi\left(K^k\overrightarrow{\pi} f(X), \overrightarrow{\pi} f(X)\right)\\ &= \mathsf{Var}\left(\frac{1}{n}\sum_{t=1}^{n}\overrightarrow{\pi} f(X_t)\right)\end{aligned}$$

after we absorb the sums involving i and i' into $\mathsf{Cov}_\pi\left(K^k\mathbb{1}\{X \in \mathsf{X}_i\}, \mathbb{1}\{X \in \mathsf{X}_{i'}\}\right)$. Finally we inspect the first term on the right hand side:

$$\begin{aligned}\mathbb{E}\left[\mathsf{Var}\left(\hat{f}_{\mathsf{ideal}}\,|\{\rho(X_t)\}_{t=1}^{n}\right)\right] &= \mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^{R}\sum_{j=1}^{N_i}\mathsf{Var}\left(f(Y_{ij})\,|\{\rho(X_t)\}_{t=1}^{n}\right)\right]\\ &= \mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^{R}N_i\sigma_i^2\right]\\ &= \frac{1}{n^2}\sum_{i=1}^{R}\mathbb{E}[N_i]\sigma_i^2 = \frac{1}{n}\mathsf{Var}_\pi(\overleftarrow{\pi} f(X))\end{aligned}$$

from which the full result follows.

7.2.4.2 Proof of lemma 57

First note that $\mathbb{E}[f(X_t)\,|\{\rho(X_t)\}_{t=1}^n] = \mu_{\rho(X_t)} = \overrightarrow{\pi}f(X_t)$. Second, by the law of total variance we have that

$$\begin{aligned}\text{Var}\left(\frac{1}{n}\sum_{t=1}^{n} f(X_t)\right) &= \mathbb{E}\left[\text{Var}\left(\frac{1}{n}\sum_{t=1}^{n} f(X_t)\,|\{\rho(X_t)\}_{t=1}^n\right)\right] + \text{Var}\left(\mathbb{E}\left[\frac{1}{n}\sum_{t=1}^{n} f(X_t)\,|\{\rho(X_t)\}_{t=1}^n\right]\right)\\ &= \mathbb{E}\left[\text{Var}\left(\frac{1}{n}\sum_{t=1}^{n} f(X_t)\,|\{\rho(X_t)\}_{t=1}^n\right)\right] + \text{Var}\left(\frac{1}{n}\sum_{t=1}^{n} \mathbb{E}[f(X_t)\,|\{\rho(X_t)\}_{t=1}^n]\right)\\ &= \mathbb{E}\left[\text{Var}\left(\frac{1}{n}\sum_{t=1}^{n} f(X_t)\,|\{\rho(X_t)\}_{t=1}^n\right)\right] + \text{Var}\left(\frac{1}{n}\sum_{t=1}^{n} \overrightarrow{\pi}f(X_t)\right)\end{aligned}$$

and the result follows.

7.2.4.3 Proof of proposition 58

That the condition 1. entails $\text{Var}(\hat{f}_{\text{ideal}}) \leq \text{Var}(n^{-1}\sum_{t=1}^n f(X_t))$ follows from Proposition 54, Lemma 57 and the fact that $f \equiv \overrightarrow{\pi}f$ implies $\overleftarrow{\pi}f \equiv 0$.

For condition 2. the fact that $f \equiv \overleftarrow{\pi}f + \pi(f)$ means that $\overrightarrow{\pi}f \equiv \pi(f)$. Therefore $\text{Var}(\hat{f}_{\text{ideal}}) = n^{-1}\text{Var}_\pi(\overleftarrow{\pi}f)$. Compare this with $\text{Var}(n^{-1}\sum_{t=1}^n f(X_t))$:

$$\text{Var}\left(n^{-1}\sum_{t=1}^{n} f(X_t)\right) := \frac{1}{n}\text{Var}_\pi(\overleftarrow{\pi}f) + \frac{1}{n}2\sum_{k=1}^{n-1}\frac{n-k}{n}\text{Cov}_\pi(\overleftarrow{\pi}f(X), K^k\overleftarrow{\pi}f(X))$$

which clearly dominates $\text{Var}(\hat{\mu}_{\text{ideal}}) = n^{-1}\text{Var}_\pi(\overleftarrow{\pi}f)$ when K is positive.

7.2.4.4 Proof of proposition 60

For the unbiasedness of $\hat{f}_{\text{occ}}$ we have the following:

$$\mathbb{E}[\hat{f}_{\text{occ}}] = \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}[\mathbb{1}\{S_t = 0\}f(X_t)] + \mathbb{E}[\mathbb{1}\{S_t = 1\}f(Y_t)]$$

Inspecting the first term in the summand:

$$\begin{aligned}
\mathbb{E}[\mathbb{1}\{S_t = 0\}f(X_t)] &= \mathbb{E}[\mathbb{E}[\mathbb{1}\{S_t = 0\}f(X_t)\,|\{X_t\}_{t=1}^n]] \\
&= \mathbb{E}[f(X_t)\mathbb{E}[\mathbb{1}\{S_t = 0\}\,|\{X_t\}_{t=1}^n]] \\
&= \mathbb{E}[(1 - \alpha(\rho(X_t)))f(X_t)] \\
&= \sum_{i=1}^{R} \pi(\mathsf{X}_i)(1 - \alpha(i))\pi_i(f)
\end{aligned}$$

Inspecting the second term:

$$\begin{aligned}
\mathbb{E}[\mathbb{1}\{S_t = 1\}f(Y_t)] &= \mathbb{E}[\mathbb{E}[\mathbb{1}\{S_t = 1\}f(Y_t)\,|\{X_t\}_{t=1}^n]] \\
&= \mathbb{E}[f(Y_t)\mathbb{E}[\mathbb{1}\{S_t = 1\}\,|\{X_t\}_{t=1}^n]] \\
&= \mathbb{E}[\alpha(\rho(X_t))f(Y_t)] \\
&= \sum_{i=1}^{R} \pi(\mathsf{X}_i)\alpha(i)\pi_i(f)
\end{aligned}$$

Incorporating the summands into the sum gives the desired answer.

As for the variance, we make the following decomposition:

$$\mathsf{Var}(\hat{f}_{\mathsf{occ}}) = \mathbb{E}\left[\mathsf{Var}\left(\hat{f}_{\mathsf{occ}}\,|\{\rho(X_t)\}_{t=1}^n\right)\right] + \mathsf{Var}\left(\mathbb{E}\left[\hat{f}_{\mathsf{occ}}\,|\{\rho(X_t)\}_{t=1}^n\right]\right)$$

Working with the expectation on the left of the sum:

$$\mathbb{E}\left[\hat{f}_{\mathsf{occ}}\,|\{\rho(X_t)\}_{t=1}^n\right] = \frac{1}{n}\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\{S_t = 0\}f(X_t)\,|\{\rho(X_t)\}_{t=1}^n\right] + \mathbb{E}\left[\mathbb{1}\{S_t = 1\}f(Y_t)\,|\{\rho(X_t)\}_{t=1}^n\right]$$

Inspecting the first term in the summand:

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}\{S_t = 0\}f(X_t)\,|\{\rho(X_t)\}_{t=1}^n\right] &= \mathbb{E}\left[\mathbb{1}\{S_t = 0\}\,|\{\rho(X_t)\}_{t=1}^n\right]\mathbb{E}\left[f(X_t)\,|\{\rho(X_t)\}_{t=1}^n\right] \\
&= (1 - \alpha(\rho(X_t)))\,\mu_{\rho(X_t)}
\end{aligned}$$

where the first line comes from the conditional independence $\{S_t\} \perp \{X_t\} \,|\{\rho(X_t)\}$. Inspecting the second term:

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}\{S_t = 1\} f(Y_t) \,|\{\rho(X_t)\}_{t=1}^{n}\right] &= \mathbb{E}\left[\mathbb{1}\{S_t = 1\} \,|\{\rho(X_t)\}_{t=1}^{n}\right] \mathbb{E}\left[f(Y_t) \,|\{\rho(X_t)\}_{t=1}^{n}\right] \\
&= \alpha(\rho(X_t)) \mu_{\rho(X_t)}
\end{aligned}$$

where the first line comes from the conditional independence $\{S_t\} \perp \{Y_t\} \,|\{\rho(X_t)\}$. Combining the two terms gives

$$\mathbb{E}\left[\hat{f}_{\text{occ}} \,|\{\rho(X_t)\}_{t=1}^{n}\right] = \frac{1}{n} \sum_{t=1}^{n} \mu_{\rho(X_t)} = \frac{1}{n} \sum_{t=1}^{n} \overrightarrow{\pi} f(X_t)$$

and hence

$$\text{Var}\left(\mathbb{E}\left[\hat{f}_{\text{occ}} \,|\{\rho(X_t)\}_{t=1}^{n}\right]\right) = \text{Var}\left(\frac{1}{n} \sum_{t=1}^{n} \overrightarrow{\pi} f(X_t)\right)$$

Now we work with the first term in the variance decomposition:

$$\begin{aligned}
\text{Var}\left(\hat{f}_{\text{occ}} \,|\{\rho(X_t)\}_{t=1}^{n}\right) = \frac{1}{n^2} \sum_{t=1}^{n} &\text{Var}\left(f_{\text{occ}}(X_t, S_t, Y_t) \,|\{\rho(X_t)\}_{t=1}^{n}\right) \\
&+ \frac{1}{n^2} \sum_{t \neq t'} \text{Cov}\left(f_{\text{occ}}(X_t, S_t, Y_t), f_{\text{occ}}(X_t', S_t', Y_t') \,|\{\rho(X_t)\}_{t=1}^{n}\right)
\end{aligned}$$

Inspecting the summand in the first sum:

$$\begin{aligned}
\text{Var}\left(f_{\text{occ}}(X_t, S_t, Y_t) \,|\{\rho(X_t)\}_{t=1}^{n}\right) &= \text{Var}\left(\mathbb{1}\{S_t = 0\} f(X_t) + \mathbb{1}\{S_t = 1\} f(Y_t) \,|\{\rho(X_t)\}_{t=1}^{n}\right) \\
&= \mathbb{E}\left[\mathbb{1}\{S_t = 0\} f(X_t)^2 + \mathbb{1}\{S_t = 1\} f(Y_t)^2 \,|\{\rho(X_t)\}_{t=1}^{n}\right] \\
&\quad - \mathbb{E}\left[\mathbb{1}\{S_t = 0\} f(X_t) + \mathbb{1}\{S_t = 1\} f(Y_t) \,|\{\rho(X_t)\}_{t=1}^{n}\right]^2 \\
&= (1 - \alpha(\rho(X_t)))\, \mathbb{E}[f(X_t)^2 \,|\{\rho(X_t)\}_{t=1}^{n}] + \alpha(\rho(X_t)) \mathbb{E}[f(Y_t)^2 \,|\{\rho(X_t)\}_{t=1}^{n}] \\
&\quad - \left((1 - \alpha(\rho(X_t)))\, \mu_{\rho(X_t)} + \alpha(\rho(X_t)) \mu_{\rho(X_t)}\right)^2 \\
&= (1 - \alpha(\rho(X_t)))\, (\sigma_{\rho(X_t)}^2 + \mu_{\rho(X_t)}^2) + \alpha(\rho(X_t))(\sigma_{\rho(X_t)}^2 + \mu_{\rho(X_t)}^2) - \mu_{\rho(X_t)}^2 \\
&= \sigma_{\rho(X_t)}^2
\end{aligned}$$

where the third equality comes from the conditional independences $\{S_t\} \perp \{X_t\} \,|\{\rho(X_t)\}$ and $\{S_t\} \perp \{Y_t\} \,|\{\rho(X_t)\}$. Inspecting the summand in the second sum:

$$\begin{aligned}
\text{Cov}\left(f_{\text{occ}}, f'_{\text{occ}} \,|\{\rho(X_t)\}_{t=1}^n\right) &= \text{Cov}\left(\mathbb{1}\{S_t = 0\} f(X_t), \mathbb{1}\{S'_t = 0\} f(X'_t) \,|\{\rho(X_t)\}_{t=1}^n\right) \\
&= \mathbb{E}\left[\mathbb{1}\{S_t = 0\} f(X_t) \mathbb{1}\{S'_t = 0\} f(X'_t) \,|\{\rho(X_t)\}_{t=1}^n\right] \\
&\quad - \mathbb{E}\left[\mathbb{1}\{S_t = 0\} f(X_t) \,|\{\rho(X_t)\}_{t=1}^n\right] \mathbb{E}\left[\mathbb{1}\{S'_t = 0\} f(X'_t) \,|\{\rho(X_t)\}_{t=1}^n\right] \\
&= (1 - \alpha(\rho(X_t)))\,(1 - \alpha(\rho(X'_t)))\, \mathbb{E}\left[f(X_t) f(X'_t) \,|\{\rho(X_t)\}_{t=1}^n\right] \\
&\quad - (1 - \alpha(\rho(X_t)))\,(1 - \alpha(\rho(X'_t)))\, \mu_{\rho(X_t)} \mu_{\rho(X'_t)} \\
&= (1 - \alpha(\rho(X_t)))\,(1 - \alpha(\rho(X'_t)))\, \text{Cov}\left(f(X_t), f(X'_t) \,|\{\rho(X_t)\}_{t=1}^n\right) \\
&= \text{Cov}\left(f_a(X_t), f_a(X'_t) \,|\{\rho(X_t)\}_{t=1}^n\right)
\end{aligned}$$

where we abbreviate $f_{\text{occ}} := f_{\text{occ}}(X_t, S_t, Y_t)$ and $f'_{\text{occ}} := f_{\text{occ}}(X'_t, S'_t, Y'_t)$where the third equality comes from the conditional independences $\{S_t\} \perp \{X_t\} \,|\{\rho(X_t)\}$ and $S_i \perp S_j \,|\{\rho(X_t)\}$ for all $i, j \in [n]$. Using the law of total covariance we have

$$\begin{aligned}
\mathbb{E}\left[\text{Cov}\left(f_a(X_t), f_a(X'_t) \,|\{\rho(X_t)\}_{t=1}^n\right)\right] &= \text{Cov}\left(f_a(X_t), f_a(X'_t)\right) \\
&\quad - \text{Cov}\left(\mathbb{E}\left[f_a(X_t) \,|\{\rho(X_t)\}_{t=1}^n\right], \mathbb{E}\left[f_a(X'_t) \,|\{\rho(X_t)\}_{t=1}^n\right]\right) \\
&= \text{Cov}\left(f_a(X_t), f_a(X'_t)\right) - \text{Cov}\left(\overrightarrow{\pi} f_a(X_t), \overrightarrow{\pi} f_a(X'_t)\right)
\end{aligned}$$

Combining everything gives

$$\mathbb{E}\left[\text{Var}\left(\hat{f}_{\text{occ}} \,|\{\rho(X_t)\}_{t=1}^n\right)\right] = \frac{1}{n}\mathbb{E}[\sigma^2_{\rho(X_t)}] + \frac{1}{n^2}\sum_{t \neq t'} \text{Cov}\left(f_a(X_t), f_a(X'_t)\right) - \text{Cov}\left(\overrightarrow{\pi} f_a(X_t), \overrightarrow{\pi} f_a(X'_t)\right)$$

We have that $\mathbb{E}[\sigma^2_{\rho(X_t)}] = \text{Var}_\pi(\overleftarrow{\pi} f(X))$ which gives the desired expression, when combined with Var $\left(\mathbb{E}\left[\hat{f}_{\text{occ}} \,|\{\rho(X_t)\}_{t=1}^n\right]\right)$ and the results of Proposition 54.

7.2.4.5 Proof of proposition 61

For π_{occ}-reversibility we need that

$$\pi_{\mathsf{occ}}(dx, s, dy)K_{\mathsf{occ}}((dx, s, dy) \to (dx', s', dy')) = \pi_{\mathsf{occ}}(dx', s', dy')K_{\mathsf{occ}}((dx', s', dy') \to (dx, s, dy))$$

for all $dx, dx', dy, dy' \in \mathcal{X}$ and $s, s' \in \{0, 1\}$. For notational simplicity we define

$$A\left(s\,|x\,\right) := \alpha(\rho(x))\mathbb{1}\{s = 1\} + (1 - \alpha(\rho(x)))\,\mathbb{1}\{s = 0\}$$

i.e. the probability mass function of s given x. We have

$$\begin{aligned}
\pi_{\mathsf{occ}}(dx, s, dy)K_{\mathsf{occ}}((dx, s, dy) \to (dx', s', dy')) &= \pi(dx)A\left(s\,|x\,\right)\pi_{\rho(x)}(dy)K(x \to dx')A\left(s'\,|x'\,\right)\pi_{\rho(x')}(dy') \\
&= \pi(dx)K(x \to dx')A\left(s'\,|x'\,\right)\pi_{\rho(x')}(dy')A\left(s\,|x\,\right)\pi_{\rho(x)}(dy) \\
&= \pi(dx')K(x' \to dx)A\left(s'\,|x'\,\right)\pi_{\rho(x')}(dy')A\left(s\,|x\,\right)\pi_{\rho(x)}(dy) \\
&= \pi(dx')A\left(s'\,|x'\,\right)\pi_{\rho(x')}(dy')K(x' \to dx)A\left(s\,|x\,\right)\pi_{\rho(x)}(dy) \\
&= \pi_{\mathsf{occ}}(dx', s', dy')K_{\mathsf{occ}}((dx', s', dy') \to (dx, s, dy))
\end{aligned}$$

where the third equality comes from the π-reversibility of K.

7.2.4.6 Proof of proposition 62

We directly observe

$$\begin{aligned}
\int_{\mathsf{X}} \sum_{s=0}^{1} \int_{\mathsf{X}} f_{\mathsf{occ}}(x, s, y)^2 \pi_{\mathsf{occ}}(dx, s, dy) &= \int_{\mathsf{X}} \int_{\mathsf{X}} f(y)^2\,(1 - \alpha(\rho(x)))\,\pi(dx)\pi_{\rho(x)}(dy) \\
&\quad + \int_{\mathsf{X}} \int_{\mathsf{X}} f(x)^2 \alpha(\rho(x))\pi(dx)\pi_{\rho(x)}(dy) \\
&= \sum_{i=1}^{R} \pi(\mathsf{X}_i)\,(1 - \alpha(i))\,\pi_i(f^2) + \pi(\mathsf{X}_i)\alpha(i)\pi_i(f^2) \\
&= \pi(f^2) < \infty
\end{aligned}$$

where the last line is due to the fact that $f \in L^2(\pi)$.

7.2.4.7 Proof of Theorem 63

1. $\Rightarrow$ 2. Let K be the kernel of the Markov chain $\{X_t\}_{t=1}^n$. [Douc et al. 2023, Proposition 3.5] says the LLN stated in statement 1. is equivalent to the fact that for all functions $h : \mathsf{X} \to \mathbb{R}$ if $Kh \equiv h$ then h is constant. Let $h_{\text{occ}} : \mathsf{X} \times \{0,1\} \times \mathsf{X} \to \mathbb{R}$ be such that $K_{\text{occ}} h_{\text{occ}} \equiv h_{\text{occ}}$. Then for all $(x,s,y) \in \mathsf{X} \times \{0,1\} \times \mathsf{X}$

$$\int_{\mathsf{X}} \sum_{s'=0}^{1} \int_{\mathsf{X}} h_{\text{occ}}(x',s',y') K_{\text{occ}}((x,s,y) \to (dx',s',dy')) = h_{\text{occ}}(x,s,y)$$

The fact that $K_{\text{occ}}((x,s,y) \to (dx',s',dy'))$ is independent of s and y means that h_{occ} is a function of x only. Therefore the above equation is equivalent to

$$\begin{aligned} \int_{\mathsf{X}} \sum_{s'=0}^{1} \int_{\mathsf{X}} h_{\text{occ}}(x') K_{\text{occ}}((x,s,y) \to (dx',s',dy')) &= h_{\text{occ}}(x) \\ \Rightarrow K h_{\text{occ}}(x) &= h_{\text{occ}}(x) \end{aligned}$$

and so h_{occ} is constant by hypothesis. Therefore [Douc et al. 2023, Proposition 3.5] asserts an LLN for the occlusion process.

2. $\Rightarrow$ 1. Assuming statement 2. means that $K_{\text{occ}} h_{\text{occ}} \equiv h_{\text{occ}}$ implies h_{occ} is a constant. Here we use the fact that applying K_{occ} to a function of only its first variable is the same as applying K to it. Thus we have that for any $h : \mathsf{X} \to \mathbb{R}$

$$\begin{aligned} Kh \equiv h &\Rightarrow K_{\text{occ}} h \equiv h \\ &\Rightarrow h \text{ is constant} \end{aligned}$$

and we are done.

7.2.4.8 Proof of Theorem 65

Let $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$ and let $g(x,s,y) = \mathbb{1}\{s=0\}f(x) + \mathbb{1}\{s=1\}f(y) \in (\mathsf{G}, \|.\|_{\mathsf{G}})$ as in the theorem statement. First note that $\pi_{\mathsf{occ}}(g) = \pi(f_\alpha)$. Then we have that

$$\begin{aligned}
K_{\mathsf{occ}}g(x,s,y) &= \int_{\mathsf{X}} \sum_{s'=0}^{1} \int_{\mathsf{X}} (\mathbb{1}\{s'=0\}f(x') + \mathbb{1}\{s'=1\}f(y'))K_{\mathsf{occ}}((x,s,y) \to (dx', s', dy')) \\
&= \int_{\mathsf{X}} \int_{\mathsf{X}} (1 - \alpha(\rho(x')))\, f(x')K(x \to dx')\pi_{\rho(x')}(dy') \\
&\qquad + \int_{\mathsf{X}} \int_{\mathsf{X}} \alpha(\rho(x'))f(y')K(x \to dx')\pi_{\rho(x')}(dy') \\
&= \int_{\mathsf{X}} (1 - \alpha(\rho(x')))\, f(x')K(x \to dx') + \int_{\mathsf{X}} \alpha(\rho(x'))\pi_{\rho(x')}(f)K(x \to dx') \\
&= Kf_\alpha(x)
\end{aligned}$$

Since $K_{\mathsf{occ}} = K$ when acting on functions who only depend on x we have that $K_{\mathsf{occ}}^t g = K^t f_\alpha$ for $t \in \mathbb{N}\backslash\{0\}$. Therefore

$$\begin{aligned}
\|K_{\mathsf{occ}}^t g - \pi_{\mathsf{occ}}(g)\|_{\mathsf{G}} &= \|K^t f_\alpha - \pi(f_\alpha)\| \\
&\leq C_{f_\alpha} r(t)
\end{aligned}$$

where the first line is due to the fact that $\|g\|_{\mathsf{G}} = \|g\|$ for all functions g solely of their first argument and the second line is by hypothesis.

7.2.4.9 Proof that $C_{f_\alpha} \leq C_f$ in example 67

Note that $C_f := C\|f - \pi(f)\|$ with $C > 0$ where $\|.\|$ is the sup norm. Define $f_0 := f - \pi(f)$. Then

$$\begin{aligned}
C_{f_\alpha} &= C\|f_\alpha - \pi(f_\alpha)\| \\
&= C \sup_{x \in \mathsf{X}} \left|(1 - \alpha(\rho(x)))\, f(x) + \alpha(\rho(x)) \overrightarrow{\pi} f(x) - \pi(f)\right| \\
&= C \sup_{x \in \mathsf{X}} \left|(1 - \alpha(\rho(x)))\, f_0(x) + \alpha(\rho(x)) \overrightarrow{\pi} f_0(x)\right| \\
&\leq C \sup_{x \in \mathsf{X}} (1 - \alpha(\rho(x)))\, |f_0(x)| + \alpha(\rho(x))\, |\overrightarrow{\pi} f_0(x)| \\
&\leq C\|f_0\|
\end{aligned}$$

where the second equality comes from the fact that $\pi(f_\alpha) = \pi(f)$, the third equality comes from the fact that $\pi(\overrightarrow{P} f) = \pi(f)$, the first inequality comes from Jensen's inequality, and the final inequality comes from the fact that $|f_0(x)|$ and $|\overrightarrow{\pi} f_0(x)|$ are upper bounded by $\|f_0\|$.

7.2.4.10 Proof of Theorem 70

Say that ν is the distribution of the first state (X_1, S_1, Y_1) of the occlusion process, such that μ is the marginal of its first component. Then the result follows from the fact that $\nu K^t_{\mathsf{occ}}(g) = \mu K^t(f_\alpha)$ and $\pi_{\mathsf{occ}}(g) = \pi(f_\alpha)$ for all $g \in \mathsf{G}$ and $\mathsf{G} \in \mathcal{C}$. These results are derived in 7.2.4.8. In particular, we have that

$$\begin{aligned}
D_{\mathsf{G}}(\nu K^t_{\mathsf{occ}}, \pi_{\mathsf{occ}}) &= \sup_{g \in \mathsf{G}} \left|\nu K^t_{\mathsf{occ}}(g) - \pi_{\mathsf{occ}}(g)\right| \\
&= \sup_{f \in \mathsf{F}} \left|\mu K^t(f_\alpha) - \pi(f_\alpha)\right| \\
&\leq C_\mu r(t)
\end{aligned}$$

where the second line comes from the fact that as f ranges over F, g ranges over a subset of G. The final line is by hypothesis.

7.2.4.11 Proof of Theorem 72

[Gallegos-Herrada, Ledvinka, and J. Rosenthal 2023, Theorem 1 viii)] has that the geometric ergodicity of a Markov kernel K is equivalent to the existence of a π-almost everywhere measurable function $V : \mathsf{X} \to [1, \infty]$, a small set $S \in \mathcal{X}$ and constants $\lambda < 1$ and $b < \infty$ with

$$KV(x) \leq \lambda V(x) + b\mathbb{1}\{x \in S\}$$

Defining $V_{\mathsf{occ}}(x, s, y) := V(x)$ and $S_{\mathsf{occ}} := S \times \{0, 1\} \times \mathsf{X}$ we prove that V_{occ} and S_{occ} satisfy an inequality such as the above with K_{occ}. Firstly

$$\begin{aligned}
K_{\mathsf{occ}}V_{\mathsf{occ}}(x, s, y) &= \int_{\mathsf{X}} \sum_{s'=0}^{1} \int_{\mathsf{X}} K_{\mathsf{occ}}((x, s, y) \to (dx', s', dy'))V_{\mathsf{occ}}(x', s', y') \\
&= \int_{\mathsf{X}} \sum_{s'=0}^{1} \int_{\mathsf{X}} K(x \to dx')\left((1 - \alpha(\rho(x')))\,\mathbb{1}\{s' = 0\} + \alpha(\rho(x'))\mathbb{1}\{s' = 1\}\right)\pi_{\rho(x')}(dy')V(x) \\
&= \int_{\mathsf{X}} \int_{\mathsf{X}} K(x \to dx')V(x)\pi_{\rho(x')}(dy') \\
&= \int_{\mathsf{X}} K(x \to dx')V(x) = KV(x)
\end{aligned}$$

Hence $K_{\mathsf{occ}}V_{\mathsf{occ}}(x, s, y) \leq \lambda V(x) + b\mathbb{1}\{x \in S\}$. Noting that $V(x) \equiv V_{\mathsf{occ}}(x, s, y)$ and $\mathbb{1}\{x \in S\} \equiv \mathbb{1}\{(x, s, y) \in S_{\mathsf{occ}}\}$ along with the fact that V_{occ} is π_{occ}-almost everywhere measurable gives the result.

7.2.4.12 Proof of corollary 73

Since K is geometrically ergodic and π-reversible Proposition 61 and Theorem 72 imply that K_{occ} is geometrically ergodic and π_{occ}-reversible. Similarly Proposition 62 has that $f_{\mathsf{occ}} \in L^2(\pi_{\mathsf{occ}})$. Therefore [G. Roberts and J. Rosenthal 1997, Corollary 2.1] implies that $\hat{f}_{\mathsf{occ}}$ admits the CLT as in the statement of the corollary.

For the finiteness of the asymptotic variance, without loss of generality we work with $f_0 := f_{\mathsf{occ}} - \pi(f)$ since

$$\lim_{n\to\infty} n\mathsf{Var}(n^{-1}\sum_{t=1}^{n} f_{\mathsf{occ}}(X_t, S_t, Y_t)) = \lim_{n\to\infty} n\mathsf{Var}(n^{-1}\sum_{t=1}^{n} f_0(X_t, S_t, Y_t))$$

Expanding the variance then gives

$$\lim_{n\to\infty} n\text{Var}(n^{-1}\sum_{t=1}^{n} f_0(X_t, S_t, Y_t)) = \|f_0\|^2_{\pi_{\text{occ}}} + 2\lim_{n\to\infty}\sum_{k=1}^{n-1}\frac{n-k}{n}\langle f_0, K^k f_0\rangle_{\pi_{\text{occ}}}$$

The sum in the limit is a Cesàro sum and so converges to $\sum_{k=1}^{\infty}\langle f_0, K^k f_0\rangle_{\pi_{\text{occ}}}$ if $\sum_{k=1}^{\infty}|\langle f_0, K^k f_0\rangle_{\pi_{\text{occ}}}| < \infty$. Geometric ergodicity along with reversibility implies a spectral gap [Gallegos-Herrada, Ledvinka, and J. Rosenthal 2023, Theorem 1 xxx)] which means that $|\langle f_0, K^k f_0\rangle_{\pi_{\text{occ}}}| \leq \lambda^k \|f_0\|^2_{\pi_{\text{occ}}}$ for some $\lambda \in [0, 1)$. This gives absolute convergence and hence the convergence of the Cesàro sum. Noting that $\|f_0\|^2_{\pi_{\text{occ}}} = \text{Var}_{\pi_{\text{occ}}}(f)$ and $\langle f_0, K^k f_0\rangle_{\pi_{\text{occ}}} = \text{Cov}_{\pi_{\text{occ}}}(f_{\text{occ}}, K^k f_{\text{occ}})$ completes the proof.

7.2.4.13 Proof of proposition 74

From Proposition 60 we have that

$$\lim_{n\to\infty} n\text{Var}(\hat{f}_{\text{occ}}) = \lim_{n\to\infty} n\text{Var}(\hat{f}_{\text{ideal}}) + 2\lim_{n\to\infty}\sum_{k=1}^{n-1}\frac{n-k}{n}C_k \tag{7.5}$$

where

$$C_k := \text{Cov}_\pi(f_a(X), K^k f_a(X)) - \text{Cov}_\pi(\overrightarrow{\pi} f_a(X), K^k \overrightarrow{\pi} f_a(X))$$

and $f_a(x) := (1 - \alpha(\rho(x)))f(x)$, so as long as the two limits on the right hand side of (7.5) converge the proposition statement will hold: we will see that they do.

For the first limit, Proposition 54 dictates that

$$\lim_{n\to\infty} n\text{Var}(\hat{f}_{\text{ideal}}) = \text{Var}_\pi(\overleftarrow{\pi} f) + \lim_{n\to\infty}\text{Var}\left(\frac{1}{n}\sum_{t=1}^{n}\overrightarrow{\pi} f(X_t)\right)$$

Geometric ergodicity and reversibility implies a spectral gap [Gallegos-Herrada, Ledvinka, and J. Rosenthal 2023, Theorem 1 xxx)], which itself implies that the above limit on the right hand side of the equation can be expressed as

$$\lim_{n\to\infty}\text{Var}\left(\frac{1}{n}\sum_{t=1}^{n}\overrightarrow{\pi} f(X_t)\right) = \text{Var}_\pi(\overrightarrow{\pi} f) + 2\sum_{k=1}^{\infty}\text{Cov}_\pi(\overrightarrow{\pi} f(X), K^k \overrightarrow{\pi} f(X))$$

Now to address the second limit on the right hand side of the equation (7.5). The expression inside the limit is a Cesàro sum, and hence converges when $\sum_{k=1}^{\infty} |C_k| < \infty$. The existence of a spectral gap for K implies the existence of a $\lambda \in [0, 1)$ such that

$$\begin{aligned} |C_k| &= |\langle f_a + \overrightarrow{\pi} f_a, K^k(f_a - \overrightarrow{\pi} f_a)\rangle| \\ &\leq \lambda^k |\langle f_a + \overrightarrow{\pi} f_a, f_a - \overrightarrow{\pi} f_a\rangle| \end{aligned}$$

and hence the sum converges absolutely.

7.2.4.14 Proof that Y is sampled from π restricted to X_C

We have that

$$\begin{aligned} \mathbb{P}\left(Y \in A \middle| U \leq \frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y) \cap Y \in \mathsf{X}_C\right) &= \frac{\mathbb{E}_{Y,U}\left[\mathbb{1}\{Y \in A\}\mathbb{1}\{U \leq \frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y)\}\mathbb{1}\{Y \in \mathsf{X}_C\}\right]}{\mathbb{E}_{Y,U}\left[\mathbb{1}\{U \leq \frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y)\}\mathbb{1}\{Y \in \mathsf{X}_C\}\right]} \\ &= \frac{\mathbb{E}_{Y}\left[\mathbb{1}\{Y \in A\}\mathbb{E}_U\left[\mathbb{1}\{U \leq \frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y)\}\right]\mathbb{1}\{Y \in \mathsf{X}_C\}\right]}{\mathbb{E}_{Y}\left[\mathbb{E}_U\left[\mathbb{1}\{U \leq \frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y)\}\right]\mathbb{1}\{Y \in \mathsf{X}_C\}\right]} \\ &= \frac{\mathbb{E}_{Y}\left[\mathbb{1}\{Y \in A\}\frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y)\mathbb{1}\{Y \in \mathsf{X}_C\}\right]}{\mathbb{E}_{Y}\left[\frac{1}{C}\frac{d\tilde{\pi}}{d\tilde{Q}}(Y)\mathbb{1}\{Y \in \mathsf{X}_C\}\right]} \\ &= \frac{\mathbb{E}_{Y\sim\pi}\left[\mathbb{1}\{Y \in A\}\mathbb{1}\{Y \in \mathsf{X}_C\}\right]}{\mathbb{E}_{Y\sim\pi}\left[\mathbb{1}\{Y \in \mathsf{X}_C\}\right]} = \mathbb{P}_{Y\sim\pi}(Y \in A \,|Y \in \mathsf{X}_C\,) \end{aligned}$$

7.2.4.15 Proof of Theorem 80

First note that for all $B = B_x \times B_s \times B_y \in \mathcal{X} \times 2^{\{0,1\}} \times \mathcal{X}$ we have

$$\begin{aligned}
\nu K_{\mathsf{occ}}(B) &= \int_{\mathsf{X}} \sum_{s=0}^{1} \int_{\mathsf{X}} \nu(dx, s, dy) K_{\mathsf{occ}}((x, s, y) \to B) \\
&= \int_{\mathsf{X}} \sum_{s=0}^{1} \int_{\mathsf{X}} \mu(dx) q(s, dy\,|x\,) K_{\mathsf{occ}}((x, s, y) \to B) \\
&= \int_{x \in \mathsf{X}} \mu(dx) K_{\mathsf{occ}}((x, s, y) \to B) \\
&= \int_{x \in \mathsf{X}} \mu(dx) \int_{x' \in B_x} K(x \to dx')\,(\alpha(\rho(x'))\mathbb{1}\{1 \in B_s\} + (1 - \alpha(\rho(x')))\,\mathbb{1}\{0 \in B_s\})\,\pi_{\rho(x')}(B_y)
\end{aligned}$$

for $\mu \in (\mathsf{M}, \|.\|_{*,\mathsf{M}})$ where the second equality is by hypothesis and the third equality comes from the fact that $K_{\mathsf{occ}}((x, s, y) \to B)$ is independent of s and y. Continuing, we have

$$\begin{aligned}
\nu K_{\mathsf{occ}}(B) &= \sum_{i=1}^{R} (\alpha(i)\mathbb{1}\{1 \in B_s\} + (1 - \alpha(i))\,\mathbb{1}\{0 \in B_s\})\,\pi_i(B_y) \int_{x \in \mathsf{X}} \mu(dx) K(x \to B_x \cap \mathsf{X}_i) \\
&= \sum_{i=1}^{R} (\alpha(i)\mathbb{1}\{1 \in B_s\} + (1 - \alpha(i))\,\mathbb{1}\{0 \in B_s\})\,\pi_i(B_y) \mu K(B_x \cap \mathsf{X}_i)
\end{aligned}$$

By a similar argument we have that

$$\nu K^t_{\mathsf{occ}}(B) = \sum_{i=1}^{R} (\alpha(i)\mathbb{1}\{1 \in B_s\} + (1 - \alpha(i))\,\mathbb{1}\{0 \in B_s\})\,\pi_i(B_y) \mu K^t(B_x \cap \mathsf{X}_i)$$

for all $t \in \mathbb{N}\backslash\{0\}$.

Dual norms are defined as suprema of a dual object as evaluated across the primal space. Therefore we need to inspect how νK^t_{occ} acts on a function g in the normed function space $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$.

$$\begin{aligned}
\nu K^t_{\mathsf{occ}}(g) &= \sum_{i=1}^{R} \int_{\mathsf{X}} \sum_{s=0}^{1} \int_{\mathsf{X}} g(x,s,y) \left(\alpha(i)\mathbb{1}\{s=1\} + (1-\alpha(i))\,\mathbb{1}\{s=0\}\right) \pi_i(dy) \mu K^t(dx) \mathbb{1}\{x \in \mathsf{X}_i\} \\
&= \sum_{i=1}^{R} \int_{\mathsf{X}} \int_{\mathsf{X}} f(y)\alpha(i)\pi_i(dy)\mu K^t(dx)\mathbb{1}\{x \in \mathsf{X}_i\} + \int_{\mathsf{X}} \int_{\mathsf{X}} f(x)\,(1-\alpha(i))\,\pi_i(dy)\mu K^t(dx)\mathbb{1}\{x \in \mathsf{X}_i\} \\
&= \sum_{i=1}^{R} \alpha(i)\mu K^t(\pi_i(f)\mathbb{1}\{. \in \mathsf{X}_i\}) + (1-\alpha(i))\,\mu K^t(f\mathbb{1}\{. \in \mathsf{X}_i\}) \\
&= \mu K^t \left(\sum_{i=1}^{R} \alpha(i)\pi_i(f)\mathbb{1}\{. \in \mathsf{X}_i\} + (1-\alpha(i))\, f\mathbb{1}\{. \in \mathsf{X}_i\} \right) \\
&= \mu K^t(f_\alpha)
\end{aligned}$$

Equally, we have that $\pi_{\mathsf{occ}}(g) = \pi(f_\alpha)$. Therefore the dual norm has the following form:

$$\begin{aligned}
\|\nu K^t_{\mathsf{occ}} - \pi_{\mathsf{occ}}\|_{*,\mathsf{N}} &= \sup_{g \in (\mathsf{G}, \|.\|_{\mathsf{G}})} \frac{|(\nu K^t_{\mathsf{occ}} - \pi_{\mathsf{occ}})\,(g)|}{\|g\|_{\mathsf{G}}} \\
&= \sup_{f \in (\mathsf{F}, \|.\|)} \frac{|(\mu K^t - \pi)\,(f_\alpha)|}{\|g\|_{\mathsf{G}}}
\end{aligned}$$

Where the second equality comes from the fact that as f ranges over $(\mathsf{F}, \|.\|)$, g ranges over $(\mathsf{G}, \|.\|_{\mathsf{G}})$. To place an upper bound on the numerator within the supremum, we have that $|(\mu K^t - \pi)\,(f_\alpha)| = \|\mu K^t - \pi\|_{\mathsf{M},*} d(f_\alpha, \ker(\mu K^t - \pi))$ where $d(f,S) := \inf_{h \in S} \|f - h\|$ for all $f \in (\mathsf{F}, \|.\|)$ and $S \subseteq \mathsf{F}$ [Hashimoto, Nakamura, and Oharu 1986, Lemma 1.1]. Since $0 \in \ker(\mu K^t - \pi)$ we have that $|(\mu K^t - \pi)\,(f_\alpha)| \leq \|\mu K^t - \pi\|_{\mathsf{M},*} \|f_\alpha\|$ and hence

$$\begin{aligned}
\|\nu K^t_{\mathsf{occ}} - \pi_{\mathsf{occ}}\|_{*,\mathsf{N}} &\leq \left(\sup_{f \in (\mathsf{F}, \|.\|)} \frac{\|f_\alpha\|}{\|g\|_{\mathsf{G}}} \right) \|\mu K^t - \pi\|_{\mathsf{M},*} \\
&\leq \left(\sup_{f \in (\mathsf{F}, \|.\|)} \frac{\|f_\alpha\|}{\|g\|_{\mathsf{G}}} \right) C_\mu r(t)
\end{aligned}$$

where the final inequality is by hypothesis.

7.2.5 Proofs from chapter 5

7.2.5.1 Proof of proposition 75

Let $L = QD \in \mathbb{R}^{d\times d}$ be the optimal preconditioner and $\Sigma_\pi \in \mathbb{R}^{d\times d}$ be the covariance of π. This covariance eigendecomposes as $\Sigma_\pi = Q_\pi D_\pi Q_\pi^T$ where $Q_\pi = (V_\pi\, W_\pi) \in O\left(d\right)$ with $V_\pi \in \mathbb{R}^{d\times m}$ having columns $\left\{v_i^{(\pi)} : i \in [m]\right\}$ and $D_\pi = \text{diag}\left\{\lambda_i^{(\pi)} : i \in [d]\right\}$. Since $\Sigma_{\tilde{\pi}} = \left(QD\right)^{-1} \Sigma_\pi \left(QD\right)^{-T} = \left(QD\right)^{-1} \Sigma_\pi^{1/2} \left(\left(QD\right)^{-1} \Sigma_\pi^{1/2}\right)^T$ where $\Sigma_\pi^{1/2} := Q_\pi D_\pi^{1/2}$ we will work with $\left(QD\right)^{-1} \Sigma_\pi^{1/2}$ because the manipulations we do will simply be mirrored in $\left(\left(QD\right)^{-1} \Sigma_\pi^{1/2}\right)^T$. We have that

$$\left(QD\right)^{-1} \Sigma_\pi^{1/2} = \begin{pmatrix} \sigma_\pi^{-1/2} & 0 \\ 0 & \mathbf{I}_{d-m} \end{pmatrix} \begin{pmatrix} V_\pi^T \\ W^T \end{pmatrix} \begin{pmatrix} V_\pi & W_\pi \end{pmatrix} D_\pi^{1/2}$$

where $\sigma_\pi := \text{diag}\left\{\lambda_i^{(\pi)} : i \in [m]\right\}$ and $W \in \mathbb{R}^{d\times(d-m)}$ has orthonormal columns, which are all orthogonal to the columns of V_π. Continuing:

$$\begin{aligned}
\left(QD\right)^{-1} \Sigma_\pi^{1/2} &= \begin{pmatrix} \sigma_\pi^{-1/2} & 0 \\ 0 & \mathbf{I}_{d-m} \end{pmatrix} \begin{pmatrix} V_\pi^T \\ W^T \end{pmatrix} \begin{pmatrix} V_\pi & W_\pi \end{pmatrix} D_\pi^{1/2} \\
&= \begin{pmatrix} \sigma_\pi^{-1/2} & 0 \\ 0 & \mathbf{I}_{d-m} \end{pmatrix} \begin{pmatrix} V_\pi^T V_\pi & V_\pi^T W_\pi \\ W^T V_\pi & W^T W_\pi \end{pmatrix} D_\pi^{1/2} \\
&= \begin{pmatrix} \sigma_\pi^{-1/2} & 0 \\ 0 & \mathbf{I}_{d-m} \end{pmatrix} \begin{pmatrix} \mathbf{I}_m & 0 \\ 0 & W^T W_\pi \end{pmatrix} D_\pi^{1/2} \\
&= \begin{pmatrix} \mathbf{I}_m & 0 \\ 0 & W^T W_\pi \end{pmatrix} \begin{pmatrix} \sigma_\pi^{-1/2} & 0 \\ 0 & \mathbf{I}_{d-m} \end{pmatrix} D_\pi^{1/2} \\
&= \begin{pmatrix} \mathbf{I}_m & 0 \\ 0 & W^T W_\pi \end{pmatrix} \begin{pmatrix} \mathbf{I}_m & 0 \\ 0 & \tilde{\sigma}_\pi^{1/2} \end{pmatrix}
\end{aligned}$$

where $\tilde{\sigma}_\pi := \text{diag}\left\{\lambda_i^{(\pi)} : i \in \{m+1, \ldots, d\}\right\}$, and the third equality comes from the fact that the columns of V_π are mutually orthogonal, and the columns of W are orthogonal to the columns V_π which are also orthogonal to the columns of W_π. Putting this all together, we have that $\|\Sigma_{\tilde{\pi}}\|_{\mathsf{op}} = \left\| (QD)^{-1} \Sigma_\pi (QD)^{-T} \right\|_{\mathsf{op}} = \max\left\{\lambda^{(\pi)}_{m+1}, 1\right\}$.

7.2.5.2 Proof of proposition 76

We proceed by induction on m.

Base Case, $m = 1$: Here the set of orthonormal vectors is $\{v_1\}$. It is easily checked that $Q_1 e_1 = v_1$.

Assume the hypothesis for $m = k$.

Inductive Step, $m = k + 1$: Again, it is easily checked that $Q_{k+1} e_{k+1} = v_{k+1}$. Let $n < k + 1$. Then

$$\begin{aligned} Q_{k+1} e_n &= \left(I_d - 2 \frac{(Q_k e_{k+1} - v_{k+1})(Q_k e_{k+1} - v_{k+1})^T}{\|Q_k e_{k+1} - v_{k+1}\|^2} \right) Q_k e_n \\ &= Q_k e_n - 2 \frac{(Q_k e_{k+1} - v_{k+1})^T Q_k e_n}{\|Q_k e_{k+1} - v_{k+1}\|^2} (Q_k e_{k+1} - v_{k+1}) \\ &= v_n - 2 \frac{e_{k+1}^T Q_k^T Q_k e_n - v_{k+1}^T v_n}{\|Q_k e_{k+1} - v_{k+1}\|^2} (Q_k e_{k+1} - v_{k+1}) \\ &= v_n \end{aligned}$$

where $Q_k e_n = v_n$ is true by hypothesis, and $e_{k+1}^T Q_k^T Q_k e_n = 0$ due to the fact that $Q_k \in O(d)$.

7.3 Appendix C: Miscellanea

7.3.1 Inherited convergence of the occlusion process in a generic normed measure space

As stated in Section 4.1.3.4 the inherited convergence of the occlusion process in IPMs was just an example of a more general phenomenon. Here we detail a result which applies to a more general class of normed measure spaces. Normed measure spaces are often dual to normed function spaces. Therefore the following inheritance is stated relative to two normed function spaces: $(\mathsf{F}, \|.\|)$ and $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$ where $\mathcal{C}_{\|.\|}$ is defined as in Section 4.1.3.3.

Theorem 79. *Say the Markov chain $\{X_t\}_{t=1}^{n}$ converges to π in the normed measure space $(\mathsf{M}, \|.\|_{*,\mathsf{M}})$ dual to the normed function space $(\mathsf{F}, \|.\|)$ with rate function $r(t)$ and constant C_μ. Consider all normed measure spaces $(\mathsf{N}, \|.\|_{*,\mathsf{N}})$ dual to the normed function spaces $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$. Then for all measures $\nu \in (\mathsf{N}, \|.\|_{*,\mathsf{N}})$ such that $\nu(dx, s, dy) = \mu(dx)q(s, dy\,|x\,)$ with $\mu \in (\mathsf{M}, \|.\|_{*,\mathsf{M}})$ for all $(dx, s, dy) \in \mathcal{X} \times \{0, 1\} \times \mathcal{X}$ we have that*

$$\|\nu K_{\mathsf{occ}}^{t} - \pi_{\mathsf{occ}}\|_{*,\mathsf{N}} \leq \left(\sup_{f \in (\mathsf{F}, \|.\|)} \frac{\|f_\alpha\|}{\|g\|_{\mathsf{G}}} \right) C_\mu r(t)$$

where

$$f_\alpha(x) := (1 - \alpha(\rho(x)))\, f(x) + \alpha(\rho(x))\, \overrightarrow{\pi} f(x)$$

and

$$g(x, s, y) := \mathbb{1}\{s = 0\} f(x) + \mathbb{1}\{s = 1\} f(y)$$

Theorem 80. *Say the Markov chain $\{X_t\}_{t=1}^{n}$ converges to π in the normed measure space $(\mathsf{M}, \|.\|_{*,\mathsf{M}})$ dual to the normed function space $(\mathsf{F}, \|.\|)$ with rate function $r(t)$ and constant C_μ. Consider all normed measure spaces $(\mathsf{N}, \|.\|_{*,\mathsf{N}})$ dual to the normed function spaces $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$. Then for all measures $\nu \in (\mathsf{N}, \|.\|_{*,\mathsf{N}})$ such that $\nu(dx, s, dy) = \mu(dx)q(s, dy\,|x\,)$ with $\mu \in (\mathsf{M}, \|.\|_{*,\mathsf{M}})$ for all $(dx, s, dy) \in \mathcal{X} \times \{0, 1\} \times \mathcal{X}$ we have that*

$$\|\nu K_{\mathsf{occ}}^{t} - \pi_{\mathsf{occ}}\|_{*,\mathsf{N}} \leq \left(\sup_{f \in (\mathsf{F}, \|.\|)} \frac{\|f_\alpha\|}{\|g\|_{\mathsf{G}}} \right) C_\mu r(t)$$

where

$$f_\alpha(x) := (1 - \alpha(\rho(x)))\, f(x) + \alpha(\rho(x))\, \overrightarrow{\pi} f(x)$$

and

$$g(x, s, y) := \mathbb{1}\{s = 0\} f(x) + \mathbb{1}\{s = 1\} f(y)$$

For a proof see section 7.2.4.15. Say X is Hausdorff and compact, and $(\mathsf{F}, \|.\|)$ is the space of continuous functions on X with the sup norm. Then $(\mathsf{M}, \|.\|_{*,\mathsf{M}})$ is the space of regular countably additive measures with $\|\mu\|_{*,\mathsf{M}} := \sup_{f \in (\mathsf{F}, \|.\|)} \|f\|^{-1} |\mu(f)|$. So for a given $(\mathsf{G}, \|.\|_{\mathsf{G}}) \in \mathcal{C}_{\|.\|}$ we have that $g \in (\mathsf{G}, \|.\|_{\mathsf{G}})$ when there exists an $f \in (\mathsf{F}, \|.\|)$ such that $g(x,s,y) = \mathbb{1}\{s=0\}f(x) + \mathbb{1}\{s=1\}f(y)$. In this case

$$\begin{aligned}
\|g\|_{\mathsf{G}} &= \sup_{(x,s,y)\in\mathsf{X}\times\{0,1\}\times\mathsf{X}} |g(x,s,y)| \\
&= \sup_{(x,s,y)\in\mathsf{X}\times\{0,1\}\times\mathsf{X}} |\mathbb{1}\{s=0\}f(x) + \mathbb{1}\{s=1\}f(y)| \\
&= \max\{\sup_{x\in\mathsf{X}} |f(x)|, \sup_{y\in\mathsf{X}} |f(y)|\} \\
&= \sup_{x\in\mathsf{X}} |f(x)| = \|f\|
\end{aligned}$$

where in the third equality we have split into the cases $s = 0$ and $s = 1$. To work out the new constant for the convergence in $(\mathsf{N}, \|.\|_{*,\mathsf{N}})$ of the occlusion process $\{(X_t, S_t, Y_t)\}_{t=1}^{n}$ we have

$$\begin{aligned}
\sup_{f\in(\mathsf{F},\|.\|)} \frac{\|f_\alpha\|}{\|g\|_{\mathsf{G}}} &= \sup_{f\in(\mathsf{F},\|.\|)} \frac{\|(1-\alpha(\rho(.)))f + \alpha(\rho(.))\overrightarrow{\pi}f\|}{\|f\|} \\
&\leq \sup_{f\in(\mathsf{F},\|.\|)} \frac{\|(1-\alpha(\rho(.)))f\| + \|\alpha(\rho(.))\overrightarrow{\pi}f\|}{\|f\|} \\
&\leq \sup_{f\in(\mathsf{F},\|.\|)} \frac{\|f\| + \|\overrightarrow{\pi}f\|}{\|f\|} \leq 2
\end{aligned}$$

where the final inequality is due to the fact that $\|\overrightarrow{\pi}f\| \leq \|f\|$.

7.3.2 Additional results for the Ising experiment in Section 4.1.5.2

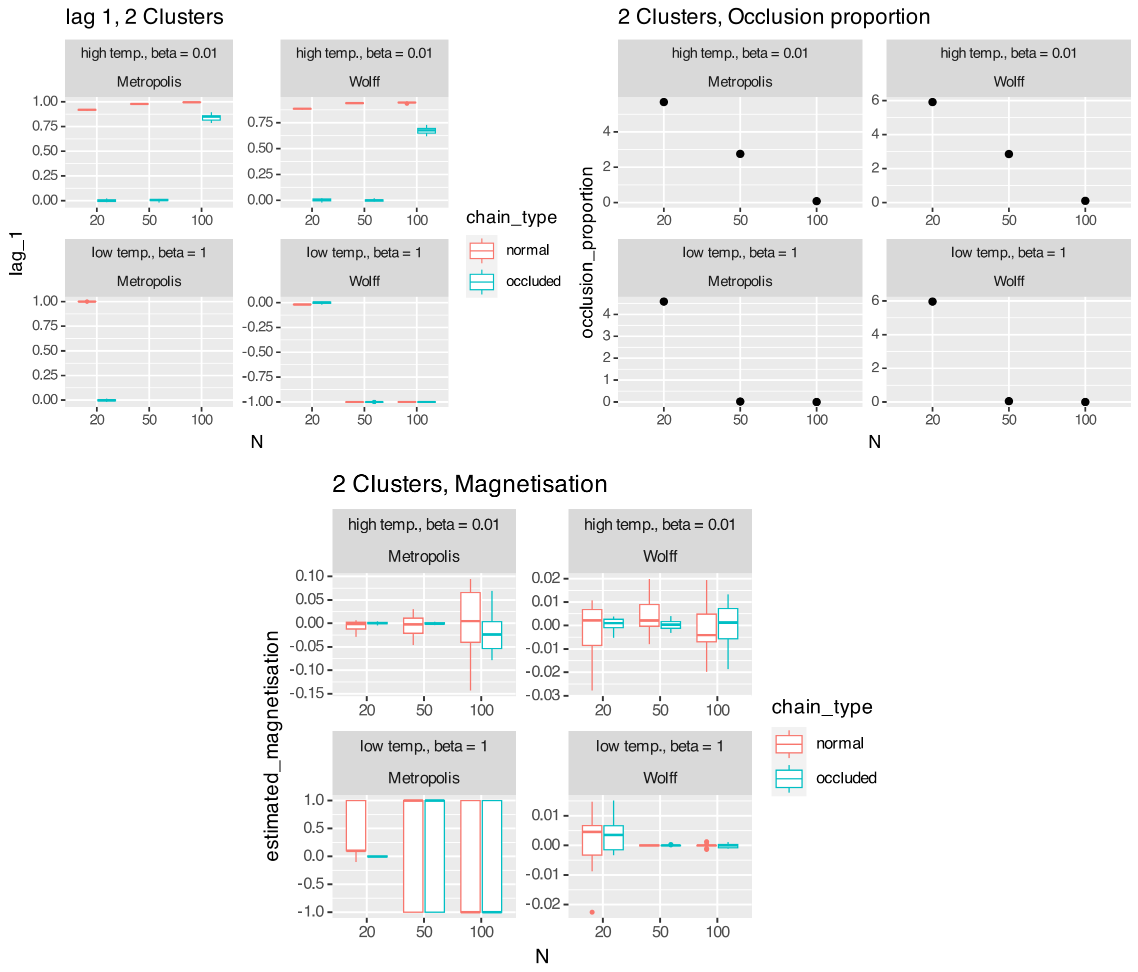

Figure 7.1: Three graphs comparing the performance of the occlusion process with the Metropolis and Wolff algorithms on the Ising model at a variety of temperatures, for a variety of graph sizes, each generated using the stochastic block model with 2 communities. In every case the horizontal axes show the number of vertices N in the graphs. Bottom: the vertical axes denote the algorithm's estimates of the expected magnetisation. Top left: the vertical axes denote the lag 1 autocorrelation coefficient of the magnetisation over the states produced by the algorithms. Top right: the vertical axes show the number of samples from the π_i's in Algorithm 4.1 divided by the number of states in the Markov chain n. We magnify the estimated magnetisation plot for ease of comprehension, the top two plots then help to explain the phenomena in the bottom plot.

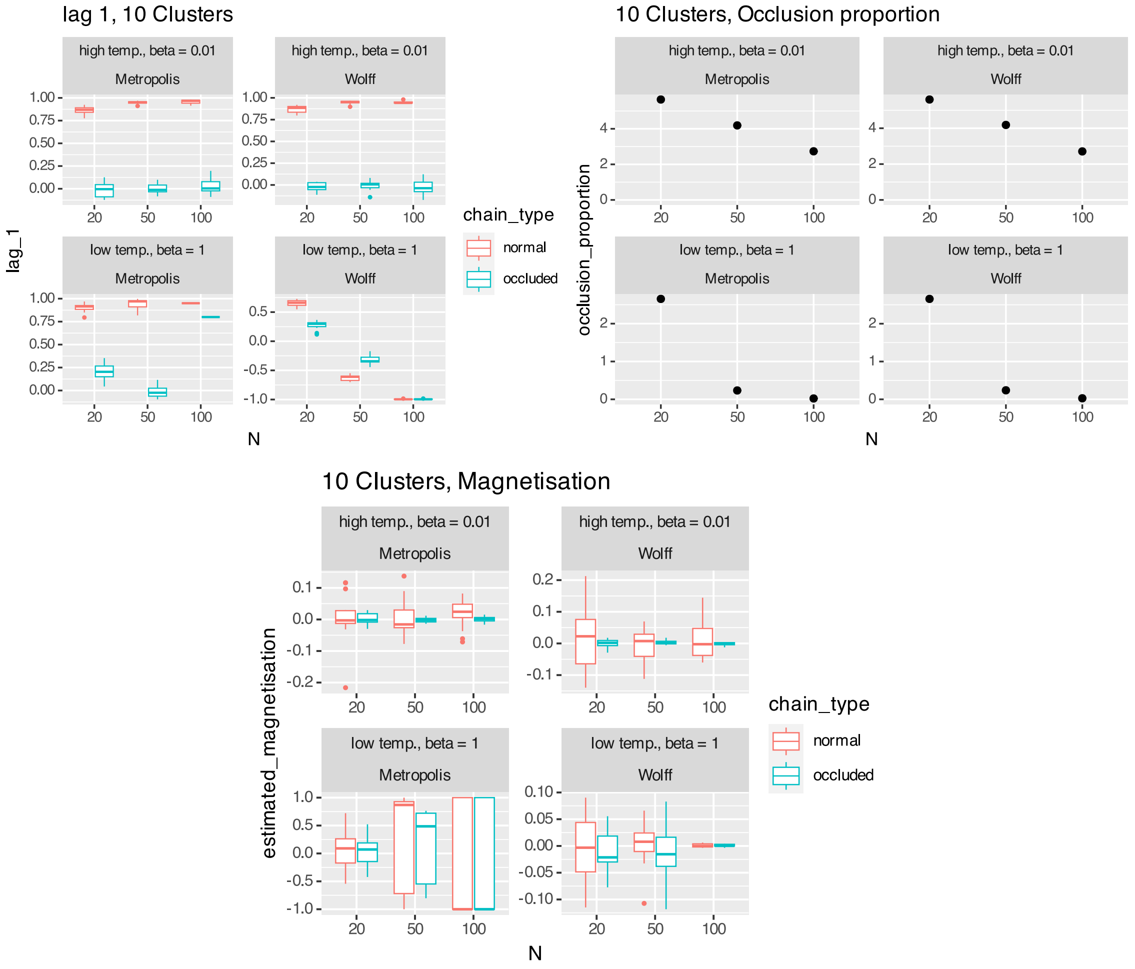

Figure 7.2: Three graphs comparing the performance of the occlusion process with the Metropolis and Wolff algorithms on the Ising model at a variety of temperatures, for a variety of graph sizes, each generated using the stochastic block model with 10 communities. In every case the horizontal axes show the number of vertices N in the graphs. Bottom: the vertical axes denote the algorithm's estimates of the expected magnetisation. Top left: the vertical axes denote the lag 1 autocorrelation coefficient of the magnetisation over the states produced by the algorithms. Top right: the vertical axes show the number of samples from the π_i's in Algorithm 4.1 divided by the number of states in the Markov chain n. We magnify the estimated magnetisation plot for ease of comprehension, the top two plots then help to explain the phenomena in the bottom plot.

Bibliography

[1] Martín Abadi et al. “TensorFlow: a system for large-scale machine learning”. In: *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*. OSDI’16. USA: USENIX Association, Nov. 2016, pp. 265–283. ISBN: 978-1-931971-33-1.

[2] A. Agliari and C. Calvi Parisetti. “A-g Reference Informative Prior: A Note on Zellner’s g Prior”. In: *Journal of the Royal Statistical Society. Series D (The Statistician)* 37.3 (1988). Publisher: [Royal Statistical Society, Wiley], pp. 271–275. ISSN: 0039-0526. DOI: 10.2307/2348164.

[3] Randolf Altmeyer. “Polynomial time guarantees for sampling based posterior inference in high-dimensional generalised linear models”. In: *arXiv preprint arXiv:2208.13296* (2022). tex.date-added: 2025-01-06 13:14:31 +0000 tex.date-modified: 2025-01-06 13:14:31 +0000.

[4] Christophe Andrieu, Anthony Lee, et al. “Explicit convergence bounds for Metropolis Markov chains: Isoperimetry, spectral gaps and profiles”. In: *The Annals of Applied Probability* 34.4 (Aug. 2024). Publisher: Institute of Mathematical Statistics, pp. 4022–4071. ISSN: 1050-5164, 2168-8737. DOI: 10.1214/24-AAP2058.

[5] Christophe Andrieu and Éric Moulines. “On the ergodicity properties of some adaptive MCMC algorithms”. In: *The Annals of Applied Probability* 16.3 (Aug. 2006). Publisher: Institute of Mathematical Statistics, pp. 1462–1505. ISSN: 1050-5164, 2168-8737. DOI: 10.1214/105051606000000286.

[6] Christophe Andrieu and Johannes Thoms. “A tutorial on adaptive MCMC”. en. In: *Statistics and Computing* 18.4 (Dec. 2008), pp. 343–373. ISSN: 1573-1375. DOI: 10.1007/s11222-008-9110-y.

[7] Manuel Arnese and Daniel Lacker. *Convergence of coordinate ascent variational inference for log-concave measures via optimal transport*. arXiv:2404.08792 [stat]. Apr. 2024. DOI: 10.48550/arXiv.2404.08792.

[8] Art B. Owen. *Practical Quasi-Monte Carlo Integration*. 2023.

[9] Yves F. Atchadé. "A cautionary tale on the efficiency of some adaptive Monte Carlo schemes". In: *The Annals of Applied Probability* 20.3 (June 2010). Publisher: Institute of Mathematical Statistics, pp. 841–868. ISSN: 1050-5164, 2168-8737. DOI: 10.1214/09-AAP636.

[10] K. B. Athreya and P. Ney. "A New Approach to the Limit Theory of Recurrent Markov Chains". In: *Transactions of the American Mathematical Society* 245 (1978). Publisher: American Mathematical Society, pp. 493–501. ISSN: 0002-9947. DOI: 10.2307/1998882.

[11] Dominique Bakry, Ivan Gentil, Michel Ledoux, et al. *Analysis and geometry of Markov diffusion operators*. Vol. 103. tex.date-added: 2024-12-20 12:48:16 +0000 tex.date-modified: 2024-12-20 12:48:16 +0000. Springer, 2014.

[12] Ricardo Baptista et al. "Conditional Sampling with Monotone GANs: From Generative Models to Likelihood-Free Inference". In: *SIAM/ASA Journal on Uncertainty Quantification* 12.3 (Sept. 2024). Publisher: Society for Industrial and Applied Mathematics, pp. 868–900. DOI: 10.1137/23M1581546.

[13] A. A. Barker. "Monte Carlo Calculations of the Radial Distribution Functions for a Proton?Electron Plasma". en. In: *Australian Journal of Physics* 18.2 (1965). Publisher: CSIRO PUBLISHING, pp. 119–134. ISSN: 1446-5582. DOI: 10.1071/ph650119.

[14] Jean-David Benamou, Francis Collino, and Jean-Marie Mirebeau. "Monotone and Consistent discretization of the Monge-Ampere operator". en. In: *Mathematics of Computation* 85.302 (2016), p. 2743. DOI: 10.1090/mcom/3080.

[15] Jean-David Benamou, Brittany D. Froese, and Adam M. Oberman. "Numerical solution of the Optimal Transportation problem using the Monge–Ampère equation". In: *Journal of Computational Physics* 260.1 (Mar. 2014). Publisher: Elsevier, pp. 107–126. DOI: 10.1016/j.jcp.2013.12.015.

[16] Alexandros Beskos et al. “Optimal tuning of the hybrid Monte Carlo algorithm”. In: *Bernoulli* 19.5A (2013). Publisher: International Statistical Institute (ISI) and Bernoulli Society for Mathematical Statistics and Probability, pp. 1501–1534. ISSN: 1350-7265.

[17] Michael Betancourt. *A Conceptual Introduction to Hamiltonian Monte Carlo*. arXiv:1701.02434 [stat]. July 2018. DOI: `10.48550/arXiv.1701.02434`.

[18] Anirban Bhattacharya, Debdeep Pati, and Yun Yang. *On the Convergence of Coordinate Ascent Variational Inference*. arXiv:2306.01122 [stat]. June 2023. DOI: `10.48550/arXiv.2306.01122`.

[19] Patrick Billingsley. *Measure and Probability*. John Wiley & Sons: New York, 1995.

[20] Christopher Bishop. *Pattern Recognition and Machine Learning*. en. 2006.

[21] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112.518 (Apr. 2017). arXiv:1601.00670 [stat], pp. 859–877. ISSN: 0162-1459, 1537-274X. DOI: `10.1080/01621459.2017.1285773`.

[22] S. G. Bobkov. “Isoperimetric and Analytic Inequalities for Log-Concave Probability Measures”. In: *The Annals of Probability* 27.4 (1999). Publisher: Institute of Mathematical Statistics, pp. 1903–1921. ISSN: 0091-1798.

[23] Nicolas Bonnotte. “From Knothe’s Rearrangement to Brenier’s Optimal Transport Map”. In: *SIAM Journal on Mathematical Analysis* 45.1 (Jan. 2013). Publisher: Society for Industrial and Applied Mathematics, pp. 64–87. ISSN: 0036-1410. DOI: `10.1137/120874850`.

[24] Nawaf Bou-Rabee and Jesús María Sanz-Serna. “Randomized Hamiltonian Monte Carlo”. In: *The Annals of Applied Probability* 27.4 (2017). Publisher: Institute of Mathematical Statistics, pp. 2159–2194. ISSN: 1050-5164.

[25] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. en. ISBN: 9780511804441 Publisher: Cambridge University Press. Mar. 2004. DOI: `10.1017/CBO9780511804441`.

[26] Guy Bresler. “Efficiently Learning Ising Models on Arbitrary Graphs”. In: *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*. STOC ’15. New York, NY, USA: Associ-

ation for Computing Machinery, June 2015, pp. 771–782. ISBN: 978-1-4503-3536-2. DOI: 10.1145/2746539.2746631.

[27] James Brofos et al. “Adaptation of the Independent Metropolis-Hastings Sampler with Normalizing Flow Proposals”. en. In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, May 2022, pp. 5949–5986.

[28] Sébastien Bubeck. *Convex Optimization: Algorithms and Complexity*. arXiv:1405.4980 [math]. Nov. 2015. DOI: 10.48550/arXiv.1405.4980.

[29] Hervé Cardot and David Degras. “Online Principal Component Analysis in High Dimension: Which Algorithm to Choose?” en. In: *International Statistical Review* 86.1 (2018). _eprint: https://onlinelibrary.wiley.com/doi/pdf pp. 29–50. ISSN: 1751-5823. DOI: 10.1111/insr.12220.

[30] G. Carlier, A. Galichon, and F. Santambrogio. “From Knothe’s Transport to Brenier’s Map and a Continuation Method for Optimal Transport”. In: *SIAM Journal on Mathematical Analysis* 41.6 (Jan. 2010). Publisher: Society for Industrial and Applied Mathematics, pp. 2554–2576. ISSN: 0036-1410. DOI: 10.1137/080740647.

[31] Bob Carpenter et al. “Stan: A Probabilistic Programming Language”. en. In: *Journal of Statistical Software* 76 (Jan. 2017), pp. 1–32. ISSN: 1548-7660. DOI: 10.18637/jss.v076.i01.

[32] Ismaël Castillo, Johannes Schmidt-Hieber, and Aad van der Vaart. “Bayesian Linear Regression with Sparse Priors”. In: *The Annals of Statistics* 43.5 (2015). Publisher: Institute of Mathematical Statistics, pp. 1986–2018. ISSN: 0090-5364.

[33] Kung Sik Chan and Charles J. Geyer. “Discussion: Markov Chains for Exploring Posterior Distributions”. In: *The Annals of Statistics* 22.4 (1994). Publisher: Institute of Mathematical Statistics, pp. 1747–1758. ISSN: 0090-5364.

[34] Minshuo Chen et al. *Dimensionality Reduction for Stationary Time Series via Stochastic Nonconvex Optimization*. arXiv:1803.02312 [cs]. Oct. 2018. DOI: 10.48550/arXiv.1803.02312.

[35] Yuansi Chen, Raaz Dwivedi, et al. “Fast mixing of metropolized hamiltonian monte carlo: Benefits of multi-step gradients”. In: *Journal of Machine Learning Research* 21.92 (2020). tex.date-added: 2024-12-20 11:39:58 +0000 tex.date-modified: 2024-12-20 11:39:58 +0000, pp. 1–72.

[36] Yuansi Chen and Khashayar Gatmiry. “A simple proof of the mixing of metropolis-adjusted langevin algorithm under smoothness and isoperimetry”. In: *arXiv preprint arXiv:2304.04095* (2023). tex.date-added: 2025-01-06 14:52:00 +0000 tex.date-modified: 2025-01-06 14:52:00 +0000.

[37] Sinho Chewi et al. “Exponential ergodicity of mirror-Langevin diffusions”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 19573–19585.

[38] William G. Cochran. *Sampling techniques*. eng. 3d ed. Wiley series in probability and mathematical statistics. OCLC: 2799031. New York, NY: Wiley, 1977. ISBN: 978-0-471-16240-7.

[39] Radu V. Craiu, Jeffrey Rosenthal, and Chao Yang. “Learn From Thy Neighbor: Parallel-Chain and Regional Adaptive MCMC”. In: *Journal of the American Statistical Association* 104.488 (Dec. 2009). Publisher: ASA Website _eprint: https://doi.org/10.1198/jasa.2009.tm08393, pp. 1454–1466. ISSN: 0162-1459. DOI: 10.1198/jasa.2009.tm08393.

[40] Tiangang Cui, Xin Tong, and Olivier Zahm. *Optimal Riemannian metric for Poincar{é} inequalities and how to ideally precondition Langevin dymanics*. arXiv:2404.02554 [math]. Apr. 2024. DOI: 10.48550/arXiv.2404.02554.

[41] Arnak S. Dalalyan. “Theoretical guarantees for approximate sampling from smooth and log-concave densities”. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 79.3 (2017). Publisher: [Royal Statistical Society, Wiley], pp. 651–676. ISSN: 1369-7412.

[42] Francisco Delgado. “Algebraic and group structure for bipartite anisotropic Ising model on a non-local basis”. In: *International Journal of Quantum Information* 13.07 (Oct. 2015). Publisher: World Scientific Publishing Co., p. 1550055. ISSN: 0219-7499. DOI: 10.1142/S0219749915500550.

[43] Randal Douc et al. *Boost your favorite Markov Chain Monte Carlo sampler using Kac’s theorem: the Kick-Kac teleportation algorithm*. arXiv:2201.05002 [stat]. May 2023. DOI: 10.48550/arXiv.2201.05002.

[44] Simon Duane et al. “Hybrid Monte Carlo”. In: *Physics Letters B* 195.2 (Sept. 1987), pp. 216–222. ISSN: 0370-2693. DOI: 10.1016/0370-2693(87)91197-X.

[45] Alain Durmus, Eric Moulines, and Eero Saksman. *On the convergence of Hamiltonian Monte Carlo*. arXiv:1705.00166 [stat]. May 2019. DOI: 10.48550/arXiv.1705.00166.

[46] Max Fathi. “Stein kernels and moment maps”. In: *The Annals of Probability* 47.4 (July 2019). Publisher: Institute of Mathematical Statistics, pp. 2172–2185. ISSN: 0091-1798, 2168-894X. DOI: 10.1214/18-AOP1305.

[47] Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. “Adaptive Monte Carlo augmented with normalizing flows”. In: *Proceedings of the National Academy of Sciences* 119.10 (Mar. 2022). Publisher: Proceedings of the National Academy of Sciences, e2109420119. DOI: 10.1073/pnas.2109420119.

[48] Marco Gallegos-Herrada, David Ledvinka, and Jeffrey Rosenthal. “Equivalences of Geometric Ergodicity of Markov Chains”. In: *Journal of Theoretical Probability* 37 (May 2023), pp. 1–27. DOI: 10.1007/s10959-023-01240-1.

[49] Ankush Ganguly and Samuel W. F. Earp. *An Introduction to Variational Inference*. arXiv:2108.13083 [cs]. Nov. 2021. DOI: 10.48550/arXiv.2108.13083.

[50] W. R. Gilks and P. Wild. “Adaptive Rejection Sampling for Gibbs Sampling”. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 41.2 (1992). Publisher: [Royal Statistical Society, Oxford University Press], pp. 337–348. ISSN: 0035-9254. DOI: 10.2307/2347565.

[51] Mark Girolami and Ben Calderhead. “Riemann manifold langevin and hamiltonian monte carlo methods”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 73.2 (2011). Publisher: Oxford University Press tex.date-added: 2024-12-20 11:52:39 +0000 tex.date-modified: 2024-12-20 11:52:39 +0000, pp. 123–214.

[52] Jonathan Goodman and Jonathan Weare. “Ensemble samplers with affine invariance”. In: *Communications in Applied Mathematics and Computational Science* 5.1 (Jan. 2010). Publisher: MSP, pp. 65–80. ISSN: 1559-3940, 2157-5452. DOI: 10.2140/camcos.2010.5.65.

[53] Louis Grenioux et al. “On Sampling with Approximate Transport Maps”. en. In: *Proceedings of the 40th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2023, pp. 11698–11733.

[54] Heikki Haario, Eero Saksman, and Johanna Tamminen. “An Adaptive Metropolis Algorithm”. In: *Bernoulli* 7.2 (2001). Publisher: International Statistical Institute (ISI) and Bernoulli Society for Mathematical Statistics and Probability, pp. 223–242. ISSN: 1350-7265. DOI: 10.2307/3318737.

[55] Olle Haggstrom and Jeffrey Rosenthal. “On Variance Conditions for Markov Chain CLTs”. In: *Electronic Communications in Probability* 12.none (Jan. 2007). Publisher: Institute of Mathematical Statistics and Bernoulli Society, pp. 454–464. ISSN: 1083-589X, 1083-589X. DOI: 10.1214/ECP.v12-1336.

[56] Timothy E. Hanson, Adam J. Branscum, and Wesley O. Johnson. “Informative g-Priors for Logistic Regression”. In: *Bayesian Analysis* 9.3 (Sept. 2014). Publisher: International Society for Bayesian Analysis, pp. 597–612. ISSN: 1936-0975, 1931-6690. DOI: 10.1214/14-BA868.

[57] Kazuo Hashimoto, Gen Nakamura, and Shinnosuke Oharu. “Riesz’s lemma and orthogonality in normed spaces”. In: *Hiroshima Mathematical Journal* 16.2 (Jan. 1986). Publisher: Hiroshima University, Mathematics Program, pp. 279–304. ISSN: 0018-2079, 2758-9641. DOI: 10.32917/hmj/1206130429.

[58] W. K. Hastings. “Monte Carlo Sampling Methods Using Markov Chains and Their Applications”. In: *Biometrika* 57.1 (1970). Publisher: [Oxford University Press, Biometrika Trust], pp. 97–109. ISSN: 0006-3444. DOI: 10.2307/2334940.

[59] Leonhard Held and Rafael Sauter. “Adaptive Prior Weighting in Generalized Regression”. In: *Biometrics* 73.1 (2017). Publisher: [Wiley, International Biometric Society], pp. 242–251. ISSN: 0006-341X.

[60] Christoph Helmberg. “A preconditioned iterative interior point approach to the conic bundle subproblem”. In: *Mathematical Programming* 205.1 (May 2024), pp. 559–615. ISSN: 1436-4646. DOI: 10.1007/s10107-023-01986-w.

[61] Nicholas J. Higham and Sheung Hun Cheng. “Modifying the inertia of matrices arising in optimization”. In: *Linear Algebra and its Applications*. Proceedings of the Sixth Conference of the International

Linear Algebra Society 275-276 (May 1998), pp. 261–279. ISSN: 0024-3795. DOI: 10.1016/S0024-3795(97)10015-5.

[62] Erwan Hillion, Oliver Johnson, and Adrien Saumard. “An extremal property of the normal distribution, with a discrete analog”. In: *Statistics & Probability Letters* 145 (Feb. 2019), pp. 181–186. ISSN: 0167-7152. DOI: 10.1016/j.spl.2018.08.018.

[63] Max Hird and Samuel Livingstone. *Quantifying the effectiveness of linear preconditioning in Markov chain Monte Carlo*. arXiv:2312.04898 [stat]. Dec. 2024. DOI: 10.48550/arXiv.2312.04898.

[64] Max Hird and Florian Maire. *The occlusion process: improving sampler performance with parallel computation and variational approximation*. arXiv:2411.11983 [stat]. Nov. 2024. DOI: 10.48550/arXiv.2411.11983.

[65] M. Hirt, M. Titsias, and P. Dellaportas. *Gradient-based adaptive HMC*. eng. Proceedings paper. Conference Name: NeurIPS 2021 Thirty-fifth Conference on Neural Information Processing Systems Meeting Name: NeurIPS 2021 Thirty-fifth Conference on Neural Information Processing Systems Place: Online Publisher: NeurIPS Volume: 35. Dec. 2021.

[66] Matthew Hoffman, Alexey Radul, and Pavel Sountsov. “An Adaptive-MCMC Scheme for Setting Trajectory Lengths in Hamiltonian Monte Carlo”. en. In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Mar. 2021, pp. 3907–3915.

[67] Matthew Hoffman, Pavel Sountsov, et al. *NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport*. arXiv:1903.03704 [stat]. Mar. 2019. DOI: 10.48550/arXiv.1903.03704.

[68] Ya-Ping Hsieh et al. “Mirrored Langevin Dynamics”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[69] Mark Huber. “Efficient Exact Sampling from the Ising Model Using Swendsen-Wang.” In: *Soda*. 1999, pp. 921–922.

[70] Pierre E. Jacob, John O’Leary, and Yves F. Atchadé. “Unbiased Markov Chain Monte Carlo Methods with Couplings”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 82.3 (July 2020), pp. 543–600. ISSN: 1369-7412. DOI: 10.1111/rssb.12336.

[71] Leif T. Johnson and Charles J. Geyer. “Variable transformation to obtain geometric ergodicity in the random-walk Metropolis algorithm”. In: *The Annals of Statistics* 40.6 (Dec. 2012). arXiv:1302.6741 [math]. ISSN: 0090-5364. DOI: 10.1214/12-AOS1048.

[72] M. Kac. “On the notion of recurrence in discrete stochastic processes”. en. In: *Bulletin of the American Mathematical Society* 53.10 (1947), pp. 1002–1010. ISSN: 0002-9904, 1936-881X. DOI: 10.1090/S0002-9904-1947-08927-8.

[73] Gurtej Kanwar. “Flow-based sampling for lattice field theories”. In: *arXiv preprint arXiv:2401.01297* (2024). tex.date-added: 2024-12-20 11:53:26 +0000 tex.date-modified: 2024-12-20 11:53:26 +0000.

[74] Sanggyun Kim et al. “Efficient Bayesian inference methods via convex optimization and optimal transport”. In: *2013 IEEE International Symposium on Information Theory*. ISSN: 2157-8117. July 2013, pp. 2259–2263. DOI: 10.1109/ISIT.2013.6620628.

[75] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. en. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992. ISBN: 978-3-642-08107-1 978-3-662-12616-5. DOI: 10.1007/978-3-662-12616-5.

[76] S. C. Kou, Qing Zhou, and Wing Hung Wong. “Equi-energy sampler with applications in statistical inference and statistical mechanics”. In: *The Annals of Statistics* 34.4 (Aug. 2006). Publisher: Institute of Mathematical Statistics, pp. 1581–1619. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/009053606000000515.

[77] Syamantak Kumar and Purnamrita Sarkar. *Streaming PCA for Markovian Data*. arXiv:2305.02456 [math]. June 2023. DOI: 10.48550/arXiv.2305.02456.

[78] Pietari Laitinen and Matti Vihola. *An invitation to adaptive Markov chain Monte Carlo convergence theory*. arXiv:2408.14903. Aug. 2024. DOI: 10.48550/arXiv.2408.14903.

[79] Shiwei Lan et al. “Markov Chain Monte Carlo from Lagrangian Dynamics”. In: *Journal of computational and graphical statistics : a joint publication of American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of North America* 24.2 (Apr. 2015), pp. 357–378. ISSN: 1061-8600. DOI: 10.1080/10618600.2014.902764.

[80] Ian Langmore et al. *A Condition Number for Hamiltonian Monte Carlo*. arXiv:1905.09813. Feb. 2020. DOI: 10.48550/arXiv.1905.09813.

[81] Hugo Lavenant and Giacomo Zanella. "Convergence Rate of Random Scan Coordinate Ascent Variational Inference Under Log-Concavity". In: *SIAM Journal on Optimization* 34.4 (Dec. 2024). Publisher: Society for Industrial and Applied Mathematics, pp. 3750–3761. ISSN: 1052-6234. DOI: 10.1137/24M1670627.

[82] Clement Lee and Darren J. Wilkinson. "A review of stochastic block models and extensions for graph clustering". en. In: *Applied Network Science* 4.1 (Dec. 2019). Number: 1 Publisher: SpringerOpen, pp. 1–50. ISSN: 2364-8228. DOI: 10.1007/s41109-019-0232-2.

[83] Yin Tat Lee, Ruoqi Shen, and Kevin Tian. "Lower Bounds on Metropolized Sampling Methods for Well-Conditioned Distributions". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 18812–18824.

[84] Benedict Leimkuhler and Sebastian Reich. *Simulating Hamiltonian Dynamics*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge: Cambridge University Press, 2005. ISBN: 978-0-521-77290-7. DOI: 10.1017/CBO9780511614118.

[85] Siran Liu, Petros Dellaportas, and Michalis K. Titsias. *Variance Reduction for the Independent Metropolis Sampler*. arXiv:2406.17699 [math]. Oct. 2024. DOI: 10.48550/arXiv.2406.17699.

[86] Samuel Livingstone. "Geometric Ergodicity of the Random Walk Metropolis with Position-Dependent Proposal Covariance". en. In: *Mathematics* 9.4 (Jan. 2021). Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, p. 341. ISSN: 2227-7390. DOI: 10.3390/math9040341.

[87] Samuel Livingstone, Michael Betancourt, et al. "On the geometric ergodicity of Hamiltonian Monte Carlo". In: *Bernoulli* 25.4A (2019). Publisher: [Bernoulli Society for Mathematical Statistics and Probability, International Statistical Institute (ISI)], pp. 3109–3138. ISSN: 1350-7265.

[88] Samuel Livingstone and Mark Girolami. "Information-Geometric Markov Chain Monte Carlo Methods Using Diffusions". en. In: *Entropy* 16.6 (June 2014). Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, pp. 3074–3102. ISSN: 1099-4300. DOI: 10.3390/e16063074.

[89] Samuel Livingstone and Giacomo Zanella. “The Barker Proposal: Combining Robustness and Efficiency in Gradient-Based MCMC”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 84.2 (Apr. 2022), pp. 496–523. ISSN: 1369-7412. DOI: 10.1111/rssb.12482.

[90] Yi-An Ma, Tianqi Chen, and Emily Fox. “A Complete Recipe for Stochastic Gradient MCMC”. In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015.

[91] Junling Ma. “Estimating epidemic exponential growth rate and basic reproduction number”. In: *Infectious Disease Modelling* 5 (Jan. 2020), pp. 129–141. ISSN: 2468-0427. DOI: 10.1016/j.idm.2019.12.009.

[92] Charles C. Margossian and Andrew Gelman. *For how many iterations should we run Markov chain Monte Carlo?* arXiv:2311.02726 [stat]. Feb. 2024. DOI: 10.48550/arXiv.2311.02726.

[93] George Marsaglia. “The squeeze method for generating gamma variates”. In: *Computers & Mathematics with Applications* 3.4 (Jan. 1977), pp. 321–325. ISSN: 0898-1221. DOI: 10.1016/0898-1221(77)90089-X.

[94] Nicholas Metropolis et al. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092. ISSN: 0021-9606. DOI: 10.1063/1.1699114.

[95] Sean Meyn and R. Tweedie. *Markov Chains and Stochastic Stability*. Vol. 92. Journal Abbreviation: Journal of the American Statistical Association Publication Title: Journal of the American Statistical Association. Jan. 1993. DOI: 10.2307/2965732.

[96] Andrew C. Miller, Nicholas J. Foti, and Ryan P. Adams. “Variational Boosting: Iteratively Refining Posterior Approximations”. en. In: *Proceedings of the 34th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2017, pp. 2420–2429.

[97] José Luis Morales and Jorge Nocedal. “Algorithm 809: PREQN: Fortran 77 subroutines for preconditioning the conjugate gradient method”. In: *ACM Trans. Math. Softw.* 27.1 (Mar. 2001), pp. 83–91. ISSN: 0098-3500. DOI: 10.1145/382043.382343.

[98] Rémy Mosseri. “Ising-like models on arbitrary graphs: The Hadamard way”. In: *Physical Review E* 91.1 (Jan. 2015). Publisher: American Physical Society, p. 012142. DOI: 10.1103/PhysRevE.91.012142.

[99] Thomas Müller et al. *Neural Importance Sampling*. arXiv:1808.03856 [cs]. Sept. 2019. DOI: 10.48550/arXiv.1808.03856.

[100] Radford M. Neal. *MCMC using Hamiltonian dynamics*. arXiv:1206.1901 [stat]. May 2011. DOI: 10.1201/b10905.

[101] Radford M. Neal and Jeffrey S. Rosenthal. “Efficiency of reversible MCMC methods: elementary derivations and applications to composite methods”. en. In: *Journal of Applied Probability* 62.1 (Mar. 2025), pp. 188–208. ISSN: 0021-9002, 1475-6072. DOI: 10.1017/jpr.2024.48.

[102] Jeffrey Negrea. “Approximations and scaling limits of Markov chains with applications to MCMC and approximate inference”. PhD Thesis. University of Toronto (Canada), 2022.

[103] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. “Problem complexity and method efficiency in optimization”. In: (1983). Publisher: Wiley-Interscience tex.date-added: 2024-12-20 11:35:17 +0000 tex.date-modified: 2024-12-20 11:35:17 +0000.

[104] Jerzy Neyman. “On the Two Different Aspects of the Representative Method: the Method of Stratified Sampling and the Method of Purposive Selection”. en. In: *Breakthroughs in Statistics: Methodology and Distribution*. Ed. by Samuel Kotz and Norman L. Johnson. New York, NY: Springer, 1992, pp. 123–150. ISBN: 978-1-4612-4380-9. DOI: 10.1007/978-1-4612-4380-9_12.

[105] Frank Noé et al. “Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning”. In: *Science* 365.6457 (Sept. 2019). Publisher: American Association for the Advancement of Science, eaaw1147. DOI: 10.1126/science.aaw1147.

[106] E. Nummelin. “A splitting technique for Harris recurrent Markov chains”. en. In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 43.4 (Dec. 1978), pp. 309–318. ISSN: 1432-2064. DOI: 10.1007/BF00534764.

[107] Esa Nummelin. *General Irreducible Markov Chains and Non-Negative Operators*. Cambridge Tracts in Mathematics. Cambridge: Cambridge University Press, 1984. ISBN: 978-0-521-60494-9. DOI: 10.1017/CBO9780511526237.

[108] Erkki Oja. "Subspace methods of pattern recognition". en. In: *Signal Processing*. Vol. 7. ISSN: 01651684 Issue: 1 Journal Abbreviation: Signal Processing. Sept. 1984, p. 79. DOI: 10.1016/0165-1684(84)90028-8.

[109] Victor M.-H. Ong, David J. Nott, and Michael S. Smith. "Gaussian Variational Approximation With a Factor Covariance Structure". In: *Journal of Computational and Graphical Statistics* 27.3 (July 2018). Publisher: ASA Website _eprint: https://doi.org/10.1080/10618600.2017.1390472, pp. 465–478. ISSN: 1061-8600. DOI: 10.1080/10618600.2017.1390472.

[110] Art B. Owen. *Monte Carlo theory, methods and examples*. artowen.su.domains/mc/, 2013.

[111] Matthew D. Parno and Youssef M. Marzouk. "Transport Map Accelerated Markov Chain Monte Carlo". In: *SIAM/ASA Journal on Uncertainty Quantification* 6.2 (Jan. 2018). Publisher: Society for Industrial and Applied Mathematics, pp. 645–682. DOI: 10.1137/17M1134640.

[112] Sam Patterson and Yee Whye Teh. "Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex". In: *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., 2013.

[113] P. H. Peskun. "Optimum Monte-Carlo Sampling Using Markov Chains". In: *Biometrika* 60.3 (1973). Publisher: [Oxford University Press, Biometrika Trust], pp. 607–612. ISSN: 0006-3444. DOI: 10.2307/2335011.

[114] Gabriel Peyré and Marco Cuturi. "Computational Optimal Transport: With Applications to Data Science". English. In: *Foundations and Trends® in Machine Learning* 11.5-6 (Feb. 2019). Publisher: Now Publishers, Inc., pp. 355–607. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000073.

[115] Martyn Plummer et al. *coda: Output Analysis and Diagnostics for MCMC*. Jan. 2024.

[116] Danilo Rezende and Shakir Mohamed. “Variational Inference with Normalizing Flows”. en. In: *Proceedings of the 32nd International Conference on Machine Learning*. ISSN: 1938-7228. PMLR, June 2015, pp. 1530–1538.

[117] Lionel Riou-Durand et al. “Adaptive Tuning for Metropolis Adjusted Langevin Trajectories”. en. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2023, pp. 8102–8116.

[118] G. O. Roberts, A. Gelman, and W. R. Gilks. “Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms”. In: *The Annals of Applied Probability* 7.1 (1997). Publisher: Institute of Mathematical Statistics, pp. 110–120. ISSN: 1050-5164.

[119] G. O. Roberts and R. L. Tweedie. “Geometric Convergence and Central Limit Theorems for Multidimensional Hastings and Metropolis Algorithms”. In: *Biometrika* 83.1 (1996). Publisher: [Oxford University Press, Biometrika Trust], pp. 95–110. ISSN: 0006-3444.

[120] Gareth Roberts and Jeffrey Rosenthal. “Geometric Ergodicity and Hybrid Markov Chains”. In: *Electronic Communications in Probability* 2.none (Jan. 1997). Publisher: Institute of Mathematical Statistics and Bernoulli Society, pp. 13–25. ISSN: 1083-589X, 1083-589X. DOI: 10.1214/ECP.v2-981.

[121] Gareth O. Roberts and Jeffrey S. Rosenthal. “Coupling and Ergodicity of Adaptive Markov Chain Monte Carlo Algorithms”. In: *Journal of Applied Probability* 44.2 (2007). Publisher: Applied Probability Trust, pp. 458–475. ISSN: 0021-9002.

[122] Gareth O. Roberts and Jeffrey S. Rosenthal. “Optimal scaling for various Metropolis-Hastings algorithms”. In: *Statistical Science* 16.4 (Nov. 2001). Publisher: Institute of Mathematical Statistics, pp. 351–367. ISSN: 0883-4237, 2168-8745. DOI: 10.1214/ss/1015346320.

[123] Gareth O. Roberts and Richard L. Tweedie. “Exponential convergence of Langevin distributions and their discrete approximations”. In: *Bernoulli* 2.4 (Dec. 1996). Publisher: Bernoulli Society for Mathematical Statistics and Probability, pp. 341–363. ISSN: 1350-7265.

[124] Jeffrey S Rosenthal et al. “Optimal proposal distributions and adaptive MCMC”. In: *Handbook of Markov Chain Monte Carlo* 4.10.1201 (2011). Publisher: Chapman & Hall/CRC Boca Raton, FL tex.date-added: 2025-01-06 13:59:03 +0000 tex.date-modified: 2025-01-06 13:59:03 +0000.

[125] Saharon Rosset and Ji Zhu. "[Least Angle Regression]: Discussion". In: *The Annals of Statistics* 32.2 (2004). Publisher: Institute of Mathematical Statistics, pp. 469–475. ISSN: 0090-5364.

[126] Daniel Sabanés Bové and Leonhard Held. "Hyper-g Priors for Generalized Linear Models". In: *Bayesian Analysis* 6 (Aug. 2010). DOI: 10.1214/11-BA615.

[127] Mher Safaryan, Filip Hanzely, and Peter Richtárik. *Smoothness Matrices Beat Smoothness Constants: Better Communication Compression Techniques for Distributed Optimization*. arXiv:2102.07245 [cs]. Feb. 2021. DOI: 10.48550/arXiv.2102.07245.

[128] Eero Saksman and Matti Vihola. "On the ergodicity of the adaptive Metropolis algorithm on unbounded domains". In: *The Annals of Applied Probability* 20.6 (Dec. 2010). Publisher: Institute of Mathematical Statistics, pp. 2178–2203. ISSN: 1050-5164, 2168-8737. DOI: 10.1214/10-AAP682.

[129] Adrien Saumard and Jon A. Wellner. *Log-concavity and strong log-concavity: a review*. arXiv:1404.5886 [math]. Apr. 2014. DOI: 10.48550/arXiv.1404.5886.

[130] Philip Schär, Michael Habeck, and Daniel Rudolf. *Parallel Affine Transformation Tuning of Markov Chain Monte Carlo*. arXiv:2401.16567 [stat]. May 2024. DOI: 10.48550/arXiv.2401.16567.

[131] Joel H Shapiro. "NOTES ON THE NUMERICAL RANGE". en. In: (2003).

[132] Jiaming Song, Shengjia Zhao, and Stefano Ermon. *A-NICE-MC: Adversarial Training for MCMC*. arXiv:1706.07561 [stat]. Mar. 2018. DOI: 10.48550/arXiv.1706.07561.

[133] Pavel Sountsov, Colin Carroll, and Matthew D. Hoffman. *Running Markov Chain Monte Carlo on Modern Hardware and Software*. arXiv:2411.04260 [stat]. Nov. 2024. DOI: 10.48550/arXiv.2411.04260.

[134] Alessio Spantini, Ricardo Baptista, and Youssef Marzouk. "Coupling Techniques for Nonlinear Ensemble Filtering". In: *SIAM Review* 64.4 (Nov. 2022). Publisher: Society for Industrial and Applied Mathematics, pp. 921–953. ISSN: 0036-1445. DOI: 10.1137/20M1312204.

[135] Nikola Surjanovic et al. *Pigeons.jl: Distributed Sampling From Intractable Distributions*. arXiv:2308.09769 [stat]. Aug. 2023. DOI: 10.48550/arXiv.2308.09769.

[136] Terence Tao. *Topics in random matrix theory*. Vol. 132. American Mathematical Soc., 2012.

[137] Robert Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996). Publisher: [Royal Statistical Society, Oxford University Press], pp. 267–288. ISSN: 0035-9246.

[138] Luke Tierney. “Markov Chains for Exploring Posterior Distributions”. In: *The Annals of Statistics* 22.4 (1994). Publisher: Institute of Mathematical Statistics, pp. 1701–1728. ISSN: 0090-5364.

[139] Michalis Titsias. “Optimal Preconditioning and Fisher Adaptive Langevin Sampling”. en. In: *Advances in Neural Information Processing Systems* 36 (Dec. 2023), pp. 29449–29460.

[140] A. M. TURING. “ROUNDING-OFF ERRORS IN MATRIX PROCESSES”. In: *The Quarterly Journal of Mechanics and Applied Mathematics* 1.1 (Jan. 1948), pp. 287–308. ISSN: 0033-5614. DOI: 10.1093/qjmam/1.1.287.

[141] Matti Vihola. “Robust adaptive Metropolis algorithm with coerced acceptance rate”. en. In: *Statistics and Computing* 22.5 (Sept. 2012), pp. 997–1008. ISSN: 1573-1375. DOI: 10.1007/s11222-011-9269-5.

[142] Cédric Villani. *Optimal Transport*. Ed. by M. Berger et al. Vol. 338. Grundlehren der mathematischen Wissenschaften. Berlin, Heidelberg: Springer, 2009. ISBN: 978-3-540-71049-3 978-3-540-71050-9. DOI: 10.1007/978-3-540-71050-9.

[143] John Von Neumann and H. H. Goldstine. “Numerical inverting of matrices of high order”. en. In: *Bulletin of the American Mathematical Society* 53.11 (1947), pp. 1021–1099. ISSN: 0273-0979, 1088-9485. DOI: 10.1090/S0002-9904-1947-08909-6.

[144] Jonas Wallin and David Bolin. “Efficient Adaptive MCMC Through Precision Estimation”. In: *Journal of Computational and Graphical Statistics* 27.4 (Oct. 2018). Publisher: ASA Website _eprint: https://doi.org/10.1080/10618600.2018.1459303, pp. 887–897. ISSN: 1061-8600. DOI: 10.1080/10618600.2018.1459303.

[145] Juyang Weng, Yilu Zhang, and Wey-Shiuan Hwang. “A Fast Algorithm for Incremental Principal Component Analysis”. en. In: *Intelligent Data Engineering and Automated Learning*. Ed. by Jiming Liu, Yiu-ming Cheung, and Hujun Yin. Berlin, Heidelberg: Springer, 2003, pp. 876–881. ISBN: 978-3-540-45080-1. DOI: 10.1007/978-3-540-45080-1_122.

[146] Helmut Wittmeyer. “Einfluß der Änderung einer Matrix auf die Lösung des zugehörigen Gleichungssystems, sowie auf die charakteristischen Zahlen und die Eigenvektoren”. en. In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 16.5 (1936). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/zamm.19360160504, pp. 287–300. ISSN: 1521-4001. DOI: 10.1002/zamm.19360160504.

[147] Ulli Wolff. “Collective Monte Carlo Updating for Spin Systems”. In: *Physical Review Letters* 62.4 (Jan. 1989). Publisher: American Physical Society, pp. 361–364. DOI: 10.1103/PhysRevLett.62.361.

[148] Keru Wu, Scott Schmidler, and Yuansi Chen. “Minimax Mixing Time of the Metropolis-Adjusted Langevin Algorithm for Log-Concave Sampling”. In: *Journal of Machine Learning Research* 23.270 (2022), pp. 1–63. ISSN: 1533-7928.

[149] T. Xifara et al. “Langevin diffusions and the Metropolis-adjusted Langevin algorithm”. In: *Statistics & Probability Letters* 91 (Aug. 2014), pp. 14–19. ISSN: 0167-7152. DOI: 10.1016/j.spl.2014.04.002.

[150] Jun Yang, Krzysztof Łatuszyński, and Gareth O. Roberts. “Stereographic Markov chain Monte Carlo”. In: *The Annals of Statistics* 52.6 (Dec. 2024). Publisher: Institute of Mathematical Statistics, pp. 2692–2713. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/24-AOS2426.

[151] Y. Yu, T. Wang, and R. J. Samworth. “A useful variant of the Davis—Kahan theorem for statisticians”. In: *Biometrika* 102.2 (2015). Publisher: [Oxford University Press, Biometrika Trust], pp. 315–323. ISSN: 0006-3444.

[152] Benjamin J. Zhang, Youssef M. Marzouk, and Konstantinos Spiliopoulos. “Transport map unadjusted Langevin algorithms: Learning and discretizing perturbed samplers”. en. In: *Foundations of Data Science* (Nov. 2024). Publisher: Foundations of Data Science, pp. 0–0. DOI: 10.3934/fods.2024047.

[153] Kelvin Shuangjian Zhang et al. “Wasserstein Control of Mirror Langevin Monte Carlo”. en. In: *Proceedings of Thirty Third Conference on Learning Theory*. ISSN: 2640-3498. PMLR, July 2020, pp. 3814–3841.

[154] Shunshi Zhang et al. “Improved Discretization Analysis for Underdamped Langevin Monte Carlo”. en. In: *Proceedings of Thirty Sixth Conference on Learning Theory*. ISSN: 2640-3498. PMLR, July 2023, pp. 36–71.

[155] Yichuan Zhang and Charles Sutton. “Quasi-Newton Methods for Markov Chain Monte Carlo”. In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011.