Approximate inference methods in probabilistic machine learning and Bayesian statistics

Marcel Andre Hirt

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Statistical Science

University College London

July 7, 2021

I, Marcel Andre Hirt, 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 work.

Abstract

This thesis develops new methods for efficient approximate inference in probabilistic models. Such models are routinely used in different fields, yet they remain computationally challenging as they involve high-dimensional integrals. We propose different approximate inference approaches addressing some challenges in probabilistic machine learning and Bayesian statistics. First, we present a Bayesian framework for genome-wide inference of DNA methylation levels and devise an efficient particle filtering and smoothing algorithm that can be used to identify differentially methylated regions between case and control groups. Second, we present a scalable inference approach for state space models by combining variational methods with sequential Monte Carlo sampling. The method is applied to self-exciting point process models that allow for flexible dynamics in the latent intensity function. Third, a new variational density motivated by copulas is developed. This new variational family can be beneficial compared with Gaussian approximations, as illustrated on examples with Bayesian neural networks. Lastly, we make some progress in a gradient-based adaptation of Hamiltonian Monte Carlo samplers by maximizing an approximation of the proposal entropy.

Impact Statement

The work in this thesis is to some degree methodological in nature and addresses inferences in probabilistic models so that any potential broader impact will be derived in fields where such methods are already employed. A significant part of this work has focused on state space models, which are routinely used in different disciplines such as epidemiology, genomics, finance, reinforcement learning and speech recognition.

DNA methylation is an important epigenetic mark that has been studied extensively for its regulatory role in biological processes and diseases. Whole genome bisulfite sequencing (WGBS) allows for genome-wide measurements of methylation up to single-base resolution, yet poses challenges to identify significantly different methylation patterns across distinct biological conditions. Using novel inferences approaches developed in this thesis, differentially methylated positions for different phenotypes can be detected. The identification of differentially methylated genomic regions can constitute a valuable contribution for life scientists that can be followed-up with further analyses.

Research presented in this work can be seen as a bridge between various communities of statistics and machine learning. Tools from different communities have been considered and the developed methods can be useful for practitioners depending on their requirements in terms of accuracy and computational costs.

Acknowledgements

I am extremely grateful to my primary supervisor Petros Dellaportas for his encouragement, humour, genuine support and advice. Without his excellent supervision, I would not have made it this far.

I would also like to extend my gratitude to my secondary supervisor Alex Beskos for his patience and guidance that made me enjoy my research.

I would like to thank my collaborators: Alain Durmus, Axel Finke and Michalis Titsias for the many fruitful and helpful discussions and thorough contributions for this thesis. Thanks also to Simone Ecker and Stephan Beck.

I very much appreciate the feedback from Theo Damoulas and Samuel Livingstone during my PhD viva and from Jim Griffin during my PhD upgrade.

On a personal note, thanks to the staff and students in the Department of Statistical Science at UCL, particularly to Ain, Anna, Marco and Xiaochen that made this experience more enjoyable.

Finally, I want to express my greatest gratitude to the limitless love and support of my family. This is for you.

Contents

List of Figures

List of Tables

Symbols

$\mathcal{B}(\mathcal{X})$	Borel σ-field of $\mathcal{X}$
C^k	set of k-times continuously differentiable functions
δ_x	Dirac measure at x
δ_{ij}	Kronecker delta
$\mathsf{D}f$	Jacobian of function f
KL	KL divergence
$L^p(\mu)$	set of measureable functions with μ-integrable p-th absolute value
$\mathcal{P}(\mathcal{X})$	space of probability measures on $\mathcal{X}$
$\circ$	function composition
∇f	Gradient of function f
$\nabla^2 f$	Hessian of function f
$\lfloor x \rfloor$	integer part of x

List of Publications and Working Papers

- Hirt, M. and Dellaportas, P. (2019). Scalable Bayesian learning for state space models using variational inference with SMC samplers. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 76–86.
- Hirt, M., Dellaportas, P., and Durmus, A. (2019). Copula-like variational inference. In Advances in Neural Information Processing Systems, pages 2959–2971.
- Hirt, M., Titsias, M., and Dellaportas, P. (2021). Entropy-based adaptive Hamiltonian Monte Carlo. Submitted.

Chapter 1

Introduction

1.1 Motivation

Probabilistic (or statistical) machine learning aims to infer plausible models to explain observed data, thereby using probability theory to represent uncertainties (Ghahramani, 2015). Bayesian machine learning allows for transforming prior probability distributions into posterior distributions after observing data and is used routinely in scientific data analysis and artificial intelligence. However, such an approach is computationally challenging as it involves a Bayesian model average, *i.e.* a marginalisation or integration of all variables that are not of direct interest. Since these computations cannot be performed exactly for most models, approximate inference methods such as Markov Chain Monte Carlo (MCMC), Sequential Monte Carlo (SMC) or variational inference are therefore commonly used instead. Although MCMC methods such as the Metropolis-Hastings algorithm (Hastings, 1970) are often considered a gold standard for Bayesian inference, the computational cost of applying such methods are generally prohibitive for large data sets, as each iteration requires a sweep of all the data points, but see for instance (Bardenet et al., 2017; Johndrow et al., 2020) for discussions on subsampling-based approaches with control variates. Furthermore, MCMC methods with a random walk proposal can be very slow to converge for models with many parameters. In contrast, proposals that use some gradient information can be more efficient, albeit automated tuning of such methods has been an ongoing challenge. Variational methods (Jordan et al., 1999) can lead to a computationally cheaper approximation for big data applications that can have frequentist constistency guarantees as the number of observations goes to infinity (Wang and Blei, 2019; Knoblauch et al., 2019), although such methods tend to exhibit some bias that is often not well

understood for a finite number of observations. There has been much research effort trying to combine efficiency and accuracy for Bayesian inference, particularly for high-dimensional models that arise for instance in Bayesian neural networks.
In many domains, time series data are often modelled using state space models. SMC (Gordon et al., 1993) is a popular method to perform inference for such probabilistic models, because it can scale well to data sets with many observations. However, parameter inference in such state space models can be challenging. The performance of such methods can be very sensitive as to how such particles are being proposed and SMC be computationally demanding in high dimensions because they require sampling many particles.

The rest of this thesis is structured as follows. Section 1.2 contains standard material, in the form of a brief review of different approximate inference methods. An outline of the contributions is given in Section 1.3. The first chapter of this thesis is a case study in Bayesian statistics that hopefully motivates some of the developments in the subsequent chapters and illustrates challenges for approximate inference methods. In Chapter 2, we aim to infer latent regime states in DNA methylation data. This large epigenetic data set can be described using state space models, which allows us to introduce particle filtering methods. We illustrate that choosing good proposal functions are important for such methods to perform well, for instance for learning model parameters or inferring the distribution of the latent paths. The filtering and smoothing algorithms suggested in Chapter 2 make explicit use of the discrete state space in a novel change point model. In Chapter 3, we want to learn proposal functions for continuous state spaces and we do not only want to perform point-estimation for the static parameters, but also infer the posterior distribution of the static parameters. We achieve this using a combination of SMC and variational inference. While Chapter 2 and 3 consider inference in state space models, the methods developed in Chapter 4 and 5 can be applied to general probabilistic models under some smoothness assumptions. Chapter 4 develops a new variational family that allows for some flexibility and that is not prohibitively expensive for high-dimensional models. Chapter 5 presents an approach to automatically adapt some hyperparameters of Hamiltonian Monte Carlo, which belongs to the class of MCMC algorithms. Chapter 6 concludes with some future work projects.

1.2 Approximate Bayesian inference methods

1.2.1 Bayesian Inference and Monte Carlo Methods

This chapter introduces computational methods for Bayesian inference and probabilistic machine learning. We introduce briefly different Monte Carlo methods: Sequential Monte Carlo in section 1.2.2, variational inference in section 1.2.3 and Markov Chain Monte Carlo in section 1.2.4.

Let $x \in \mathcal{X} \subset \mathbb{R}^d$ denote the unknown parameters of a probabilistic model that are supported on some parameter space $\mathcal{X}$. The Bayesian paradigm posits a prior distribution $\pi_0 \in \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ denotes the space of all probability measures on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, with $\mathcal{B}(\mathcal{X})$ being the Borel σ-field on $\mathcal{X}$. Suppose that π_0 has a positive density that we also write as π_0 with respect to some σ-finite-dominating measure denoted $\mathrm{d}x$. Assume we observe a data set $y \in \mathcal{Y}$ in some space $\mathcal{Y} \subset \mathbb{R}^k$ and that the statistical model gives rise to a measurable function $L\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, $(x, y) \to \mathbb{R}, \mapsto L(y|x)$, called the likelihood function of the model. Bayes' rule yields the posterior distribution, i.e. the conditional distribution of the parameters given the data, as

$$\pi(\mathrm{d}x) = \frac{\pi_0(\mathrm{d}x)L(y|x)}{\mathsf{Z}}. \tag{1.1}$$

Here, the normalizing constant $\mathsf{Z} := \int_{\mathcal{X}} \pi_0(dx)L(y|x) < \infty$ represents the marginal likelihood of the data set y. For most statistical models, the normalising constant Z in (1.1) is intractable, so resorting to numerical approximation is generally necessary to make inferences about

$$\pi(h) := \int_{\mathcal{X}} h(x)\pi(\mathrm{d}x) \tag{1.2}$$

for some function $h\colon \mathcal{X} \to \mathbb{R}$ of interest.

A central approach for estimating the quantity $\pi(f)$ defined in (1.2) relies on Monte Carlo methods using an estimator of the form

$$\hat{\pi}(h) = \frac{1}{K}\sum_{k=1}^{K} h(X^k) \tag{1.3}$$

for some sequence of random variables $(X^k)_{k=1}^K$. For example, if $(X^k)_{k=1}^K$ are iid with distribution π, then this estimator is consistent for $h \in L(\pi)$ as $K \to \infty$ by the strong law of large numbers; while for $h \in L^2(\pi)$, a central limit theorem argument applies to its asymptotic error.

1.2.2 Importance sampling and Sequential Monte Carlo

Importance sampling (Kahn and Marshall, 1953; Geweke, 1989) allows for using a Monte Carlo estimator with samples from a proposal distribution $q \in \mathcal{P}(\mathcal{X})$ different from π, if $\pi \ll q$, i.e. $q(A) = 0$ implies $\pi(A)$ for any $A \in \mathcal{B}(\mathbb{R}^d)$. Suppose that the proposal distribution has a density q with respect to the measure $\mathrm{d}x$. Then the Radon-Nikodym derivative can be written as $\frac{\mathrm{d}\pi}{\mathrm{d}q}(x) = \frac{w(x)}{\mathsf{Z}}$, where $w(x) = \frac{\gamma(x)}{q(x)}$ and $\gamma(x) = \mathsf{Z} \cdot \pi(x)$ is the unnormalised target distribution. So for Bayesian inference with the target being the posterior, we have $\gamma(x) = \pi_0(x)L(y|x)$. Hence, using $\mathsf{Z} = q(w)$, we have $\pi(h) = \frac{q(hw)}{q(w)}$, which motivates the Monte Carlo estimator

$$\hat{\pi}_{IS}(f) = \sum_{k=1}^{K} W^k h(X^k)$$

where $W^k = w(X^k)/\sum_{k=1}^{K} w(X^k)$ and $(X_k)_{k=1}^{K}$ are iid according to q. We call the weighted random measure $\hat{\pi}_{IS} = \sum_{k=1}^{K} W^k(X^k)$ the particle approximation of π. Furthermore, $\hat{\mathsf{Z}}_K = \frac{1}{K}\sum_{k=1}^{K} w(X^k)$ is an unbiased estimator of Z. Moreover, it holds that,

$$\frac{Var(\hat{\mathsf{Z}}_K)}{\mathsf{Z}^2} = \frac{1}{K}\left(q\left(\frac{\mathrm{d}\pi}{\mathrm{d}q}\right)^2 - 1\right),$$

cf. Doucet and Johansen (2009). Observe first that for $q = \pi$, importance sampling yields a zero variance estimator of the normalizing constant. Second, the variance of the estimator is characterised by the χ^2-divergence $\chi^2(\pi||q)$ between the proposal q and the target π, defined as

$$\chi^2(\pi||q) = q\left(\left(\frac{\mathrm{d}\pi}{\mathrm{d}q} - 1\right)^2\right)$$

that satisfies

$$\chi^2(\pi||q) + 1 = \left(q\left(\frac{\mathrm{d}\pi}{\mathrm{d}q}\right)^2 - 1\right) \geqslant \exp\left(\mathrm{KL}(\pi||q)\right),$$

where $\mathrm{KL}(\pi||q)$ is the Kullback-Leibler divergence between absolutely continuous measures π and q,

$$\mathrm{KL}(\pi||q) = \pi\left(\log\frac{\mathrm{d}\pi}{\mathrm{d}q}\right) \tag{1.4}$$

that plays an important role in the forthcoming chapters. We refer to Doucet and Johansen (2009); Owen (2013) for a more detailed introduction and to Agapiou

et al. (2017); Chatterjee et al. (2018) for details regarding computational costs of importance sampling.

Let us now consider a sequence of densities $(\pi_n)_{n=0}^M$, where for any $n \in \{0, \dots, M\}$, π_n is a density on the space $(\mathcal{X}^{n+1}, \mathcal{B}(\mathcal{X})^{\otimes n+1})$ with respect to $\mathrm{d}x_{0:n}$ that has the representation $\pi_n(x_{0:n}) = \gamma_n(x_{0:n})/\mathsf{Z}_n$ for a normalizing constant Z_n and positive integrable function γ_n. Here, we used the notation $x_{0:n} = (x_0, \dots, x_n)$. In sequential importance sampling, the proposal distribution has a density function q_n with an auto-regressive structure in the form of

$$q_n(x_{0:n}) = q_{n-1}(x_{0:n-1}) M_n(x_n | x_{0:n-1})$$

for $n > 1$ and $q_0(x_0) = M_0(x_0)$, where M_0 is a density on $\mathcal{X}$ and a series of transition kernels with associated densities $M_n(x_n|x_{0:n-1})$ with respect to $\mathrm{d}x$. This allows for computing importance weights recursively. Indeed, let

$$\alpha_n(x_{0:n}) = \frac{\gamma_n(x_{0:n})}{\gamma_{n-1}(x_{0:n-1}) M_n(x_n|x_{0:n-1})}$$

be the incremental importance weight function. Then,

$$w_n(x_{0:n}) = \frac{\gamma_n(x_{0:n})}{q_n(x_{0:n})} = w_0(x_0) \prod_{t=1}^{n} \alpha_t(x_{0:t}).$$

One obtains a particle approximation of $\pi_n(\mathrm{d}x_{0:n})$ given by

$$\hat{\pi}_{SIS}(\mathrm{d}x_{0:n}) = \sum_{k=1}^{K} W_n^k \delta_{X_{0:n}^k}(\mathrm{d}x_{0:n}) \quad , \quad W_n^k = \frac{w_n(X_{0:n}^k)}{\sum_{l=1}^K w_n(X_{0:n}^l)}$$

by sampling for any $k \in \{1, \dots K\} =: [K]$, first $X_0^k \sim M_0(\cdot)$ and iteratively $X_t^k \sim M_t(\cdot|X_{0:t-1}^k)$ for any $t \in [n]$. However, for increasing n, this approximation uses effectively a single particle with index $k^\star$ say satisfying $W_n^{k^\star} \approx 1$. This phenomenon is called weight degeneracy and can be mitigated by a resampling step. One possibility to do so is to first sample an ancestor variable A_{n-1}^k representing the parent of particle $X_{0:n}^k$ according to a categorical distribution on $[K]$ with probabilities W_{n-1}. We set $W_{n-1}^k = \frac{1}{K}$ and after this resampling step, one extends the path of each particle by sampling from the transition kernel $X_n^k \sim M_n(\cdot|X_{0:n-1}^{A_{n-1}^k})$ and updates the latent path via $X_{0:n}^k = (X_{0:n-1}^{A_{n-1}^k}, X_n^k)$. The resulting algorithm is called particle filtering or a Sequential Monte Carlo (SMC) algorithm and it yields an unbiased and strongly consistent estimator of the normalization constant Z_n given by

$$\hat{\mathsf{Z}}_n = \prod_{m=0}^{n} \prod_{k=1}^{K} w_m(X_m^k), \tag{1.5}$$

cf. Del Moral (1996) and see for instance more detailed treatments given in (Doucet and Johansen, 2009; Cappé et al., 2006; Douc et al., 2014; Naesseth et al., 2019).

1.2.3 Variational Inference

Variational methods (Jordan et al., 1999; Wainwright and Jordan, 2008; Blei et al., 2017) aim at approximating a density π on $\mathcal{X} \subset \mathbb{R}^d$ by first postulating a variational family Q of densities and then commonly find a good approximation $q^\star$ belonging to Q by minimizing the KL divergence with respect to π over Q, *i.e.* $q^\star \approx \arg\min_{q \in \mathrm{Q}} \mathrm{KL}(q||\pi)$. Assuming that both π and q are densities with respect to a σ-finite measure $\mathrm{d}x$, recall that the KL divergence (1.4) can be written as

$$\mathrm{KL}(q||\pi) = -\int_{\mathcal{X}} q(x) \log \frac{\pi(x)}{q(x)} \mathrm{d}x = -\mathbb{E}_{q(x)} \left[\log \gamma(x) - \log q(x)\right] + \log \mathsf{Z}, \tag{1.6}$$

where the normalising constant $\mathsf{Z} = \int_{\mathcal{X}} \gamma(x) \mathrm{d}x < \infty$ does not depend on q and $\gamma = \mathsf{Z} \cdot \pi$. Hence, minimizing $q \mapsto \mathrm{KL}(q|\pi)$ is equivalent to maximizing the variational lower bound $q \mapsto \log \mathsf{Z} - \mathrm{KL}(q|\pi) = \mathcal{L}(q)$. If the target density π is the posterior density given in (1.1), then

$$\mathcal{L}(q) = \mathbb{E}_{q(x)} \left[\log \pi_0(x) + \log L(y|x) - \log q(x)\right], \tag{1.7}$$

which is called Evidence Lower Bound (ELBO). Due to the non-negativity of the KL-divergence, $\mathcal{L}(q)$ is a lower bound on the log evidence $\log \mathsf{Z}$ for any $q \in \mathrm{Q}$.

A convenient family Q is the mean-field variational family that posits that any $q \in \mathrm{Q}$ is of the form $q(x_1, \ldots, x_d) = \prod_{i=1}^d q_i(x_i)$ for densities q_i on $\mathbb{R}$. In this case, one can optimize the variational bound using a coordinate ascent method, by iteratively maximizing each factor of the variational density, while holding the other factors fixed, cf. Attias (2000); Winn and Bishop (2005). To derive the optimal factors, one can show that

$$\mathcal{L}(q) = -\mathrm{KL}(q_j||q_j^\star) + \text{const.}$$

where constant terms do not depend on q_j and the distribution $q_j^\star$ satisfies

$$\log q_j^*(x_j) = \int_{\mathbb{R}^{d-1}} \log \gamma(x) q_{-j}(x_{-j}) \mathrm{d}x_{-j} + \text{const.} \tag{1.8}$$

where $x_{-j} = (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots x_d)$ and $q_{-j}(x_{-j}) = \prod_{i=1, i \neq j}^d q_i(x_i)$. Computing $q_j^\star$ thus requires evaluating the integral in (1.8) and evaluating the normalising constant of $q_j^\star$. While this is possible for approximate Bayesian inference where the complete

conditional is in the exponential family, computing the optimal variational factor is infeasible for a general function γ. Furthermore, the independence assumption in the mean-field family could be too restrictive, resulting in a loose variational bound.

An alternative is to assume that any $q_\xi \in \mathrm{Q}$ is indexed by a variational parameter $\xi \in \Xi$ and then find a good approximation $q_{\xi^\star}$ with $\xi^\star \approx \arg\min_{\xi\in\Xi} \mathrm{KL}(q_\xi||\pi)$, where the minimization is performed using stochastic gradient descent, assuming that $\xi \mapsto q_\xi(x)$ is differentiable for any $x \in \mathcal{X}$. To be more concrete, we can write the gradient of the variational bound as

$$\begin{aligned}\nabla_\xi \mathcal{L}(\xi) &= \nabla_\xi \int_{\mathcal{X}} q_\xi(x) \left(\log \gamma(x) - \log q_\xi(x)\right) \mathrm{d}x \\ &= \int_{\mathcal{X}} q_\xi(x) \nabla_\xi \log q_\xi(x) \left(\log \gamma(x) - \log q_\xi(x)\right) \mathrm{d}x.\end{aligned}$$

This can be seen by using the log-derivative trick $\nabla_\xi q_\xi(x) = q_\xi(x)\nabla_\xi \log q_\xi(x)$ and that the score function has expectation of zero, $\int_{\mathcal{X}} q_\xi(x)\nabla_\xi \log q_\xi(x)\mathrm{d}x = 0$. This gives rise to an unbiased and consistent Monte Carlo estimator of $\mathcal{L}(\xi)$ given by

$$\widehat{\nabla_\xi \mathcal{L}}(\xi) = \frac{1}{K}\sum_{i=1}^{K} \nabla_\xi \log q_\xi(X_i) \left(\log \gamma(X_i) - \log q_\xi(X_i)\right)$$

for iid $X_i \sim q_\xi$. Using a sequence of step sizes $(\alpha_n) > 0$ satisfying $\sum_n \alpha_n = \infty$ and $\sum_n \alpha_n^2 < \infty$, cf. Robbins and Monro (1951), a stochastic gradient algorithm $\xi_{n+1} = \xi_n + \alpha_n \widehat{\nabla_\xi \mathcal{L}}(\xi_n)$ can be used to obtain $\xi^\star$. Such a score gradient estimator, see also Ranganath et al. (2014), allows applying variational inference to general models without requiring conjugacy assumptions, and can be applied for $\mathrm{d}x$ being the Lebesgue or the counting measure, and thus allows for approximate posterior inference over latent variables that can be discrete or continuous. However, it tends to yield gradient estimators with a high variance. Different variance reduction techniques such as control variates (Tucker et al., 2017; Grathwohl et al., 2017) are often used with such estimators.

Suppose now that additionally $x \mapsto \gamma(x)$ and $x \mapsto q_\xi(x)$ are differentiable. Further, assume that there is a density p independent of ξ and a function $f\colon (x, \xi) \mapsto f_\xi(x)$ being differentiable so that the pushforward of p by f_ξ is q_ξ, *i.e.* for a random

variable H with density p, the density of $f_\xi(H)$ is q_ξ. Then

$$\begin{aligned}\nabla_\xi \mathcal{L}(\xi) &= \nabla_\xi \int_{\mathcal{X}} p(h) \left(\log \gamma(f_\xi(h)) - \log q_\xi(f_\xi(h))\right) \mathrm{d}h \\ &= \int_{\mathcal{X}} p(h) \nabla_x \left(\log \gamma(x) - \log q_\xi(x)\right) \Big|_{x=f_\xi(h)} \nabla_\xi f_\xi(h) \mathrm{d}h.\end{aligned}$$

This is commonly referred to as the reparameterization trick, see Kingma and Welling (2014); Rezende et al. (2014); Titsias and Lázaro-Gredilla (2014) and the resulting Monte Carlo estimator tends to have lower variance compared to the score gradient estimator.

1.2.4 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) methods are a popular alternative approach to sample from a target density π on $\mathcal{X}$ and can be used to make inferences about $\pi(h)$ in (1.2) for some function $h\colon \mathcal{X} \to \mathbb{R}$ that can have some theoretical guarantees. A review of MCMC methods is outside the scope of this work; we refer for instance to Tierney (1994); Roberts et al. (2004) or textbook treatments (Robert and Casella, 2013).

Recall that a Metropolis-Hastings kernel P on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ can be constructed for a given proposal kernel Q in the form

$$P(A|x) = \int_A \alpha(x,y) Q(\mathrm{d}y|x) + \delta_x(A) \int_{\mathcal{X}} (1 - \alpha(x,y)) Q(\mathrm{d}y|x) \tag{1.9}$$

for any $x \in \mathcal{X}$, $A \in \mathcal{B}(\mathcal{X})$ and measurable $\alpha\colon \mathcal{X} \times \mathcal{X} \to [0,1]$. Assume that Q admits a density q with respect to $\mathrm{d}x$, *i.e.* $Q(\mathrm{d}y|x) = q(y|x)\mathrm{d}y$, and that the target distribution π has a density still denoted by π with respect to $\mathrm{d}x$. The standard choice of the acceptance rate is

$$\alpha(x,y) = \begin{cases} \min\left(1, \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}\right) & \text{if } \pi(x)q(y|x) > 0 \\ 1 & \text{otherwise.} \end{cases}$$

which is optimal for minimizing the asymptotic variance of sample path averages among alternative choices that yield π as the invariant distribution of the Markov chain. To sample from this Markov chain at time n, one generates a candidate value $Y_n \sim Q(\cdot|X_{n-1})$. Then, with probability $\alpha(X_{n-1}, Y_n)$, one accepts this proposal and sets $X_n = Y_n$. Otherwise, the proposal is rejected and one sets $X_n = X_{n-1}$. It can be shown that the kernel P of the Markov chain $(X_n)_n$ is reversible with respect to

π, *i.e.* the detailed balance condition $\pi(\mathrm{d}y)P(\mathrm{d}x|y) = \pi(\mathrm{d}x)P(\mathrm{d}y|x)$ holds. This implies that π is an invariant distribution for P, which means for any $A \in \mathcal{B}(\mathcal{X})$, we have $\int_{\mathcal{X}} \pi(\mathrm{d}x)P(A|x) = \pi(A)$.

One choice for sampling from the proposal kernel $Y_n \sim Q(\cdot|X_{n-1})$ is

$$Y_n = X_n + \sqrt{h}Z_n$$

for $h > 0$ and iid $(Z_n)_n \sim \mathcal{N}(0, \mathrm{I})$, which yields a Random-Walk sampler. Another choice for differentiable $\pi(x) \propto e^{-U(x)}$ in the case $\mathcal{X} = \mathbb{R}^d$ is to propose

$$Y_n = X_{n-1} - h\nabla U(X_{n-1}) + \sqrt{2h}Z_n, \tag{1.10}$$

which gives rise to a MALA sampler (Roberts et al., 1996; Roberts and Stramer, 2002). Such a proposal corresponds to an Euler-Maruyama or Milstein scheme for the overdamped Langevin diffusion $(X_t)_{t>0}$ that solves the SDE

$$\mathrm{d}X_t = -\nabla U(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

with B_t being a d-dimensional Brownian motion. Instead of proposing approximately from Langevin dynamics, one can also be guided by Hamiltonian dynamics (Duane et al., 1987; Neal, 2011; Betancourt, 2017), that is by $(X_t, P_t)_{t>0}$ evolving on the phase space $\mathbb{R}^{2d}$ according to the differential equations

$$\mathrm{d}X_t = \frac{\partial H(X_t, P_t)}{\partial p}\mathrm{d}t = M^{-1}P_t\mathrm{d}t \tag{1.11}$$

and

$$\mathrm{d}P_t = -\frac{\partial H(X_t, P_t)}{\partial x}\mathrm{d}t = -\nabla U(X_t)\mathrm{d}t. \tag{1.12}$$

with the Hamiltonian function

$$H(x, p) = U(x) + \frac{1}{2}p^\top M^{-1}p$$

for some positive definite mass matrix $M \in \mathbb{R}^{d\times d}$. Numerical approximations such as the leapfrog integrator (Hairer et al., 2003) are routinely used in place of the proposing from the exact HMC dynamics (1.11)-(1.12), in conjunction with a momentum refreshment $P_0 \sim \mathcal{N}(0, M)$ and a Metropolis-Hastings acceptance step.

A π-invariant Markov chain X can be used to make inferences about $\pi(h)$ in (1.2)

using the Monte Carlo approximation (1.3), particularly if a central limit argument applies in the sense that

$$n^{-1/2}\sum_{i=1}^{n} h(X_i) \xrightarrow{d} \mathcal{N}(\pi(h), \sigma_h^2),$$

where $\sigma_h^2 \in [0, \infty)$ is the asymptotic variance. For $h \in L^2(\pi)$ with $\pi(h) = 0$, let $\gamma_k = \mathbb{E}[h(X_0)h(X_k)]$ be the lag-k autocovariance of X and suppose that $\lim_{n\to\infty} \frac{1}{n}\mathrm{Var}(\sum_{i=1}^{n} h(X_i)) < \infty$. Then,

$$\sigma_h^2 = \lim_{n\to\infty} \frac{1}{n}\mathrm{Var}\left[\sum_{i=1}^{n} h(X_i)\right] = \gamma_0 + 2\sum_{k=1}^{\infty}\gamma_k,$$

see Haggstrom et al. (2007). It can be argued that the Markov chain is more efficient if σ_h^2 is small, and this criteria is often assessed by computing the effective sample size N_{eff} that satisfies $\mathrm{Var}(\frac{1}{N}\sum_{i=1}^{N} h(X_i)) = \mathsf{N}_{\text{eff}}^{-1}\mathrm{Var}(X_1)$ and which can be expressed for stationary Markov chain X using

$$\mathsf{N}_{\text{eff}} = \frac{N}{1 + 2\sum_{k=1}^{N-1}\left(\frac{N-k}{N}\right)\mathrm{Cov}(h(X_1), h(X_{k+1}))}.$$

1.3 Outline and contributions

The thesis is divided into four parts according to the different topics that have been investigated.

1.3.1 Bayesian inference of DNA methylation levels

Chapter 2 develops a novel methylation change-point model that allows for efficient inference of latent methylation regimes. The underlying probabilistic model is a state space model. These probabilistic models consist of two processes: A latent process $(X_n)_{\geqslant 0}$ on $\mathcal{X}$ and an observable process $(Y_n)_{n\geqslant 0}$ on $\mathcal{Y}$ that follow the transition dynamics

$$X_n|(\theta, X_{n-1} = x_{n-1}, Y_{n-1} = y_{n-1}) \sim f_\theta(\cdot|x_{n-1}, y_{n-1}), \tag{1.13}$$

with a Markov transition density f_θ, $X_0 \sim f_\theta$ sampled from an initial distribution f_θ and

$$Y_n|(\theta, X_{0:n} = x_{0:n}, Y_{0:n-1} = y_{0:n-1}) \sim g_\theta(\cdot|x_n) \tag{1.14}$$

for a conditional emission density g_θ. Assume that we are given M observations $y_{0:M}$. For some function $h\colon \mathcal{X}^{M+1} \to \mathbb{R}$ of the hidden states and fixed static parameter

$\theta \in \Theta$, the evaluation of

$$\begin{aligned}&\mathbb{E}_\theta\left[h(X_{0:M})|y_{0:M}\right] \\ =&p_\theta(y_{0:M})^{-1}\int\cdots\int h(x_{0:M})g_\theta(x_0)f_\theta(x_0)\mathrm{d}x_0\prod_{\ell=0}^{M-1}f_\theta(x_{\ell+1}|x_\ell)g_\theta(y_\ell)\mathrm{d}x_{\ell+1}\end{aligned} \tag{1.15}$$

with

$$p_\theta(y_{0:M}) = \int\cdots\int g_\theta(x_0)f_\theta(x_0)\mathrm{d}x_0\prod_{\ell=0}^{M-1}f_\theta(x_{\ell+1}|x_\ell)g_\theta(y_\ell)\mathrm{d}x_{\ell+1} \tag{1.16}$$

the observed data likelihood, is generally intractable and requires Monte Carlo approximations. Our methods will be applied to real Whole genome bisulfite sequencing (WGBS) data, necessitating approximate inference approaches that can be applied to millions of time steps M. The proposed inference approach can be applied in an online fashion in the sense that the computational costs grow at most linearly with the time steps M. For the setting with a single-group, the latent process at the t-step, $X_t = (D_t, R_t) \in \mathbb{N} \times \{1, \cdots, R\}$, consists of the distance D_t to the most recent change point, while R_t denotes the current regime out of R possibilities. The state transition density is

$$\begin{aligned}f_{\theta,t+1}(x_{t+1}|x_t) \coloneqq\ &\rho_{\theta,t+1}(x_t)\delta_1(d_{t+1})P_\theta(r_{t+1}|r_t) \\ &+ (1-\rho_{\theta,t+1}(x_t))\delta_{d_t+1}(d_{t+1})\delta_{r_t}(r_{t+1})\end{aligned}$$

where $\rho_{\theta,t+1}(x_t) \in [0,1]$ is the probability of occurrence of a change point, after having spent d_t time steps in regime r_t. Furthermore, $P_\theta(r_{t+1}|r_t)$ is the r_{t+1}-th entry in the r_t-th row of an $R \times R$ matrix of transition probabilities of the regimes. The observations $y_t \coloneqq y_{t,1:S}$, with $y_{t,s}$ taking values in $\mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\}$ are the number of methylated reads, while $n_{t,s}$ denotes the number of methylated and non-methylated reads at the t-th CpG site associated with the s-th sample. The observation density is then assumed to be

$$g_{\theta,t}(x_t) = \prod_{s=1}^{S}\mathrm{BetaBinom}(y_{t,s}; n_{t,s}, \alpha_{r_t}, \beta_{r_t}),$$

where α_r and β_r are parameters of the corresponding Beta distribution that characterise each regime $r \in \{1, \ldots, R\}$. Similar change-point models have been suggested before, see Adams and MacKay (2007); Fearnhead and Liu (2007a); Caron et al. (2012); Yildirim et al. (2013), but have not been used in a two group multiple testing

problem under dependence. Our interest lies in particular in estimating the marginal posterior probabilities of the regimes $\mathbb{P}(R_t = r|y_{0:T}, \theta)$ given the observed counts of methylated reads $y_{1:T}$. With $h(x_{0:T}) = h_t(x_t) = \mathbf{1}\{r_t = r\}$, we therefore approximate (1.15) with an adaptive-lag approximation,

$$\psi_{t|T,r} = \mathbb{E}_\theta[h_{t,r}(X_t)|y_{0:T}] \approx \mathbb{E}_\theta[h_{t,r}(X_t)|y_{1:(t+\Delta)\wedge T}] \tag{1.17}$$

where we stop updating the expectation after Δ steps, see Alenlöv and Olsson (2019). Although we are considering a finite state space $\mathcal{X}$, exact computations of the filter distributions are not practical for large T, so that we use a particle approximation in (1.17). However, the particle filter used here is different from those considered in Chapter 3 for continuous state spaces that sample the proposed particles randomly from a learned proposal kernel. Instead, the proposed algorithms rely on discrete particles filters (Fearnhead, 1998; Fearnhead and Clifford, 2003; Fearnhead and Liu, 2007b) that explore that state space systematically, probing any possible next state. We consider a frequentist approach to estimate the static parameter θ by performing recursive maximum likelihood estimation that can be implemented online. We then extend the single-group model to a two-group model that includes a latent variable for each site with values in $\{0, 1\}$ to indicate if the methylation states of two phenotypes are in the same regime or not. Simulation studies are performed to asses the performance of both the single-group and two group model and we find that using optimal resampling techniques (Fearnhead and Clifford, 2003) tend to improve the performance compared to an unbiased resampling scheme. The proposed model is then applied to identify differentially methylated positions (DMPs) for an adult and a newborn donor on a chromosome-wide scale.

This chapter is based on joint ongoing work with Axel Finke, Alex Beskos, Petros Dellaportas, Simone Ecker and Stephan Beck. Axel Finke has developed and implemented the filtering algorithms for the single-group case and suggested the discrete particle filter algorithm for the two-group model.

1.3.2 Combining Sequential Monte Carlo and variational inference

In Chapter 3, we develop an approximate fully Bayesian inference approach for generic state space models with dynamics (1.13) and (1.14). The frequentist parameter estimation approach aims to maximize the log-likelihood function $\theta \mapsto \log p_\theta(y_{0:M})$ from (1.16). Using particle approximations of the score function $\nabla_\theta \log p_\theta(y_{0:M})$,

different approaches have been suggested to approximately maximize the log-likelihood using expectation-maximization (Dempster et al., 1977) or gradient-based approaches, see for instance Kantas et al. (2015); Poyiadjis et al. (2011); Del Moral et al. (2015); Olsson and Alenlöv (2020).

We suggest an alternative approach building up on previous work (Maddison et al., 2017; Naesseth et al., 2018; Le et al., 2018) that consider point estimating θ only in a variational EM setting. Denoting by $q_\phi(\cdot|\theta)$ the density of all variables generated by a particle filter, such approaches consist in maximizing a lower bound on the log-likelihood

$$\mathbb{E}_{q_\phi(\cdot|\theta)}[\hat{Z}_M^\theta] \leqslant \log p_\theta(y_{0:M})$$

where $\hat{Z}_M^\theta$ is an unbiased estimator of $p_\theta(y_{0:M})$ obtained from the particle filter algorithm. We consider a fully Bayesian approach and assume that θ has a prior density π_0. Particle MCMC algorithms (Andrieu et al., 2010) yield 'exact approximations' in such models and can be seen as an MCMC sampler that operates on an extended space - they leave invariant an extended target density $\tilde{\pi}(\theta, x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l)$. This extended density has the marginal density $\pi(\theta, x) = p(\theta, x_{0:M}|y_{0:M})$, *i.e.* the posterior distribution of the static parameter and the latent path given observations $y_{0:M}$. The extended target is a density of all K latent paths $x_{0:M}^{1:K}$ from running a particle filter, as well as ancestor variables $a_{0:M-1}^{1:K}$ and a final particle index l. We propose a variational distribution $q_{\psi,\phi}$ of these variable from which one can sample by first sampling $\theta \sim q_\psi$ from some variational density q_ψ and then running a particle filter assuming θ is the static parameter and using proposals parameterised by ϕ with output $(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l)$. We then propose to maximize the variational bound $\mathcal{L}(\psi, \phi)$ that satisfies

$$\mathcal{L}(\psi, \phi) = -\mathrm{KL}(q_{\psi,\phi}||\tilde{\pi}) + \log p(y_{0:M})$$

and that can be evaluated using

$$\mathcal{L}(\psi, \phi) = \mathbb{E}_{q_\psi(\theta)}\left[\mathbb{E}_{q_\phi(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l|\theta)}\left[\log \hat{Z}_M^{\theta,\phi}\right] + \log \frac{p(\theta)}{q_\psi(\theta)}\right], \tag{1.18}$$

where $\hat{Z}_M^{\theta,\phi}$ is an unbiased estimator of the normalising constant $\mathsf{Z}_M^\theta = p_\theta(y_{0:M})$, cf. (1.5), conditional on θ being the static parameter and ϕ parametrising the proposal transition kernels. If we choose $q_\psi(\theta)$ be a delta function and neglect the last summand in (1.18), we recover the previously considered variational EM settings optimizing

$p_\theta(y_{0:M})$ over θ. In contrast, our approach can be seen as an approximate inference alternative to particle MCMC approaches.

We apply the proposed inference approach to linear Gaussian state space models, a multivariate stochastic volatility model and a point process model. The proposed approach was motivated by the development of new inferences approaches for non-linear Hawkes point processes with latent intensity dynamics driven by a piecewise-deterministic continuous-time Markov process (Davis, 1984).

Chapter 3 is based on joint work with Petros Dellaportas (Hirt and Dellaportas, 2019).

1.3.3 A new variational density motivated by copulas

The marginal variational distribution of the static parameter and a single latent path that we propose in Chapter 3 approximates the target density more closely as measured by the KL-divergence if the number K of particles grows. For instance, we illustrate that isotropic proposals can yield a variational approximation where there is significant correlation in the latent states. This be seen as a version of Importance Weighted Auto-Encoders (IWEA), see Burda et al. (2015); Cremer et al. (2017) that is commonly used for variational EM over a static parameter. While such approaches allow for more flexible variational families using sampling-importance-resampling, the resulting (marginal) variational densities are implicit, lacking an explicit density function. We introduce in Chapter 4 a new variational density that instead allows for sampling and log-density evaluation with complexity $\mathcal{O}(d \log d)$ in the dimension d of the latent parameter. This new variational density is based on a new density c_θ with parameter θ on the hypercube $[0,1]^d$. Samples $V \sim c_\theta$ can be obtained by first sampling $\tilde{U}$ from a Beta-Liouville distribution, cf. Fang (2017). To obtain a random variable V on the hypercube from a random variable $\tilde{U}$ supported within the simplex only, we then set $V = \tilde{U}/U^*$, where $U^* = \max_{i\in\{1,\dots,d\}} \tilde{U}_i$. The marginal laws of c_θ are non-uniform, in contrast to any copula density with uniform marginals on the hypercube. We call c_θ a copula-like density and note that any density function c on the hypercube defines a density on $\mathbb{R}^d$ via

$$q(x) = c\left(F_1(x_1), \dots, F_d(x_d)\right) \prod_{i=1}^{d} f_i(x_i) \,, \tag{1.19}$$

for $x \in \mathbb{R}^d$ with respect to the Lebesgue measure. Here, $(f_1, \ldots, f_d)$ is d-dimensional vector of density functions with parameter ϕ and F_i is the cumulative distribution function associated with f_i, so for any $x_i \in \mathbb{R}$, $F_i(x_i) = \int_{-\infty}^{x_i} f_i(t_i)\mathrm{d}t_i$. Previous variational densities constructed using a Gaussian (Han et al., 2016) copula-density c or a vine-copula (Tran et al., 2015) density c in (1.19) have a computational cost of order d^2. Our contribution is that the introduced copula-like density c_θ has a computational cost of order d. Furthermore, to allow for a more flexible dependence structure, we introduce an orthonormal transformation using a product of $\frac{1}{2}d\log d$ Givens rotations arranged in a FFT-style butterfly-architecture, that has not been considered previously among the many proposed bijective transformations, also known as normalising flows (Rezende and Mohamed, 2015; Papamakarios et al., 2019), used in variational inference such as orthonormal projections (Tomczak and Welling, 2016). We illustrate our approach for a logistic regression model and for standard regression and classification benchmarks using Bayesian Neural Networks.

Chapter 4 is based on joint work with Petros Dellaportas and Alain Durmus (Hirt et al., 2019).

1.3.4 Gradient-based adaptive HMC

In chapter 5, we want to sample from a target distribution $\pi(q) \propto e^{-U(q)}$ on $\mathbb{R}^d$ by learning a mass matrix M_θ so that HMC with a leap-frog integrator that approximates the dynamics (1.11) - (1.12) with $M = M_\theta$ has (i) a high average log-acceptance rate in the Metropolis-Hastings step and (ii) is based on a proposal kernel with a high entropy. HMC with a leap-frog integrator with L steps and step-size h usually resamples the momentum $P_0 \sim \mathcal{N}(0, M_\theta)$ and then proposes deterministically from the current state q the next state $\mathcal{T}_{\theta,q}(C_\theta^\top P_0)$ where $C_\theta C_\theta^\top = M_\theta^{-1}$ and

$$\mathcal{T}_{\theta,q}(v) = q - \frac{Lh^2}{2} M_\theta^{-1}\nabla U(q) + LhC_\theta v - h^2 M_\theta^{-1}\Xi_{\theta,q}(v)$$

with $\Xi_{\theta,q}(v)$ a sum of weighted energy gradients along the trajectory. In the MALA case $L = 1$, cf. also (1.10) without preconditioning and a rescaled step-size h, it holds that $\Xi_{\theta,q}(v) = 0$ for all $v \in \mathbb{R}^d$ and $\mathcal{T}_{\theta,q}$ is a linear function, which allows for a tractable expression for the entropy of the proposal kernel in the form of $\mathcal{H}_\theta = d\log h + \log|\det C_\theta| + \text{const}$. Titsias and Dellaportas (2019) have learned the

mass or preconditioning matrix M_θ in the MALA case by maximizing

$$\mathcal{L}(\theta) = \int \pi(q)\nu(v) \left[\log \alpha_\theta(q, \mathcal{T}_{\theta,q}(v)) + \beta \mathcal{H}_\theta\right] \mathrm{d}v\mathrm{d}q.$$

Here, ν is a standard Gaussian density, $\alpha_\theta(q, q')$ is the acceptance rate for MALA when proposing from state q to state q' and $\beta > 0$ is a hyper-parameter that trades off the desire to have high log-acceptance rates and proposing large moves in all dimensions. We aim to optimize an analogous *generalised speed measure* objective also for HMC with $L > 1$. The entropy of the proposal kernel for the current position q and normalised momentum $v = C_\theta^\top P_0$ can be estimated using Monte-Carlo by evaluating the log determinant of the Jacobian $\mathsf{D}\mathcal{T}_{\theta,q}(v)$, and so requires $\mathcal{O}(d^3)$ operations. We suggest an approximation thereof that seems to work reasonably well if the Hessian function $\nabla^2 U(q)$ does not vary too much across the state space with complexity $\mathcal{O}(d^2)$ for a general mass matrix and $\mathcal{O}(d)$ for a diagonal mass matrix. The suggested approach to adapt the Markov chain differs from some previous work that are motivated by maximizing some form of expected squared jumping distance (Pasarica and Gelman, 2010)

$$\int \pi(q)\nu(v) \left[\alpha_\theta(q, \mathcal{T}_{\theta,q}(v)) \left\|q - \mathcal{T}_{\theta,q}(v)\right\|_2^2\right] \mathrm{d}v\mathrm{d}q.$$

A popular example is the "no-U-turn sampler" (NUTS) in Hoffman and Gelman (2014) that increases L until the leap-frog integrator would decrease the distance between the initial and the proposed state by making a U-turn.

Numerical experiments to sample from high-dimensional or ill-conditioned Gaussian targets suggest that the new adaptation approach can yield better effective sample sizes per computation time compared to a NUTS implementation and that choosing $L > 1$ can be beneficial. Simulations using a logistic regression model with different data sets show better effective sample sizes per computation time compared to NUTS or MALA for some data sets only. The gradient-based adaptation approach can also be used to learn a mass matrix for a log-Gaussian Cox point process that performs similar to Riemann-Manifold MALA and HMC (Girolami and Calderhead, 2011).

Chapter 5 is based on joint work with Michalis K. Titsias and Petros Dellaportas (Hirt et al., 2021).

Chapter 2

Motivating case study in Bayesian statistics: Genome-Wide Inference of DNA Methylation Levels

2.1 Introduction

DNA methylation in mammals is an epigenetic modification of DNA that adds a methyl group at position C5 in the context of cytosine-guanine dinucleotides (CpGs) (Bird, 2002) which is associated with organismal development, aging and progression of human diseases such as cancer (Robertson, 2005). High-throughput sequencing techniques provide high-resolution methylation profiles on a genome-wide scale, with WGBS becoming a gold-standard technique for methylation studies (Bock, 2012), due to its single-base coverage and high accuracy. The diploid human epigenome has more than 10^7 CpG sites within its (more than 10^8) cytosines, making a statistical analysis of such data sets extremely challenging undertaking, particularly in the context of epigenome-wide association studies (EWAS) that aim to link epigenetic variations to particular phenotypes for example in cases and controls studies (Rakyan et al., 2011). For illustration, DNA *hypomethylation*, i.e. relative undermethylation within promoter regions, has been shown to inactivate certain tumor-suppressor genes, while global DNA overmethylation or *hypermethylation* can induce genomic instability, thus contributing to cell transformation (Kulis and Esteller, 2010).

A large number of approaches have been suggested to detect differentially methylated positions or regions (DMPs, DMRs) between case and control groups – see the

review in Shafi et al. (2018) – but they often fail to: (i) allow for flexible methylation patterns that are characterised not just by the mean methylation level; (ii) take into account the spatial correlation that can change abruptly; (iii) work with a single replicate and missing reads; or (iv) allow for scalable inference on a genome-wide scale. A commonly used approach suggested in Hansen et al. (2012) smooths the methylation values and then tests for group differences using t-tests for each site, but cannot for instance (amongst other shortcomings) handle missing reads. Different Beta-Binomial models have been suggested (Feng et al., 2014; Park et al., 2014; Sun et al., 2014), but they tend to allow for only limited spatial dependence, as do most approaches relying on logistic regression (Akalin et al., 2012) or established statistical tests (Stockwell et al., 2014). More realistic models of DNA methylation have been suggested based on an one-dimensional Ising model (Jenkinson et al., 2017, 2018) or a latent Gaussian field model (Rackham et al., 2017), however, due to high computational complexity, estimating such models requires partitioning of the genome in small sub-regions or approximate inference techniques. Our approach relies on a hidden Markov model that – unlike previous approaches in such a direction (Yu and Sun, 2016; Sun and Yu, 2016; Shokoohi et al., 2019) – can distinguish regimes that differ also in the variability of the methylation levels. We combine an online maximum likelihood estimation approach with an online Bayesian approach for retrieval of information about methylation patterns. Our framework can carry out full Bayesian inference on a chromosome-wide scale, an attribute regarded as a great challenge for previous approaches.

2.2 Single-Group Methylome Change Point Model

2.2.1 Data and Likelihood Conditional on Change Points

Consider $S \geq 1$ available epigenetic samples and let $t = 1, \ldots, T$ be the numbering of the CpG sites in some chromosome. Typically, $T = \mathcal{O}(10^6)$. Observations are denoted $(y_t)_{t \geqslant 1}$, where $y_t \coloneqq y_{t,1:S}$. Here, $y_{t,s}$ takes values in $\mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\}$ and denotes the number of methylated reads at the t-th CpG site associated with the s-th sample. We assume that the corresponding total number of reads, $n_{t,s} \in \mathbb{N}_0$, is known; we write $n_t \coloneqq n_{t,1:S}$. We will make use of the notation $[n] \coloneqq \{k \in \mathbb{N} \mid k \leqslant n\}$. Taking

into account the variability in the reads, we can assume that

$$p(y_{1:T}|\pi_{1:T}, x_{1:T}, \theta) = \prod_{t=1}^{T}\prod_{s=1}^{S} p(y_{t,s}|\pi_{t,s}, \theta),$$

where

$$p(y_{t,s}|\pi_{t,s}, \theta) \coloneqq \text{Binom}(y_{t,s}; n_{t,s}, \pi_{t,s}).$$

Here, $\text{Binom}(x, n, \pi)$ denotes the probability mass function of a Binomial distribution with size parameter n and probability π evaluated at x. Furthermore, $\pi_{t,s}$ denotes the proportion of methylated alleles over all cells at CpG site t for sample s and, as usual, we write $\pi_t \coloneqq \pi_{t,1:S}$. A-priori, we model these probabilities as

$$p(\pi_{1:T}|x_{1:T}, \theta) = \prod_{t=1}^{T}\prod_{s=1}^{S} p(\pi_{t,s}|x_t, \theta),$$

where

$$p(\pi_{t,s}|x_t, \theta) \coloneqq \text{Beta}(\pi_{t,s}; \alpha_{r_t}, \beta_{r_t})$$

where $\text{Beta}(x; \alpha, \beta)$ denotes the probability mass function of a Beta distribution with parameters $\alpha, \beta > 0$, evaluated at x. Furthermore, the random variable r_t is a component of x_t (more details on this below) and takes values in $[R]$, where $R \in \mathbb{N}$ denotes the number of different *regimes*. Based on empirical evidence and expert-knowledge elicitation[1], we envisage six different model regimes for the complete genome. These are summarised in Table 2.1. To facilitate interpretation, we present these regime-specific parameters not in the standard parametrisation (α_r, β_r) for a Beta law but in the form of its mean μ_r and standard deviation σ_r. The standard parameters can then be recovered via the relationship

$$\alpha_r = \mu_r\nu_r, \quad \beta_r = (1-\mu_r)\nu_r, \quad \text{where} \quad \nu_r \coloneqq \frac{\mu_r(1-\mu_r)}{\sigma_r^2} - 1,$$

for $\sigma_r < \sqrt{\mu_r(1-\mu_r)}$.

[1]The number of regimes and the regime-specific parameters can be changed and adjusted to accommodate particular biological research questions, different technical aspects and sequencing techniques. By varying the mean and standard deviation parameter, different methylation patterns can be described using a Beta-Binomial density - for instance also bistable regimes with bimodal densities as considered in Jenkinson et al. (2017, 2018) - while being relatively parsimonious.

Table 2.1: The $R = 6$ regimes in the single-group scenario.

Regime r	Description	Mean μ_r	Std dev. σ_r
1	very large mean, small std dev.	0.95	0.05
2	very small mean, small std dev.	0.05	0.05
3	large mean, moderate std dev.	0.8	0.10
4	small mean, moderate std dev.	0.2	0.10
5	mean around 0.5, moderate std dev.	0.50	0.10
6	'chaotic' (i.e. uniform on $[0, 1]$)	0.50	$1/\sqrt{12}$

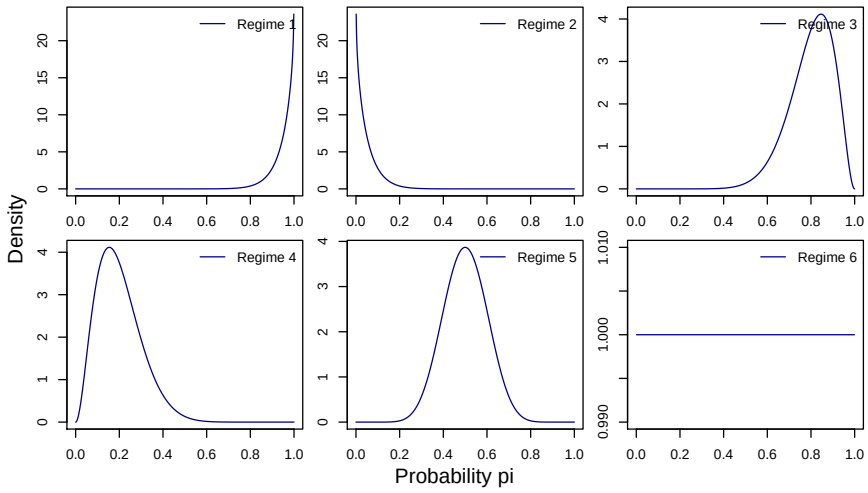


Figure 2.1: Densities of the Beta distribution for regimes 1–6, under the hyper-parameter choices shown in Table 2.1.

Assume that $x_{1:T}$ determines a sequence of *change points* over the specified regimes, such that given x_t it is known that the model is in regime r_t at CpG site $t \in [T]$ – the regime r_t at site t is taken to be the same over all samples $s \in [S]$ in the single group setting we have adopted for the moment. We can analytically integrate out all the probabilities π_t to obtain the product of Beta-Binomial likelihoods

$$p(y_{1:T}|x_{1:T}, \theta) = \int p(y_{1:T}|\pi_{1:T}, x_{1:T}, \theta) p(\pi_{1:T}|x_{1:T}, \theta) \, \mathrm{d}\pi_{1:T} = \prod_{t=1}^{T} g_{\theta,t}(x_t),$$

where, on the right-hand-side we have dropped the observations from the expression

for the quantities $g_{\theta,t}(x_t)$, the latter determined analytically as

$$\begin{aligned} g_{\theta,t}(x_t) &\coloneqq \prod_{s=1}^{S} \int_{[0,1]} \mathrm{Binom}(y_{t,s}; n_{t,s}, \pi)\, \mathrm{Beta}(\pi; \alpha_{r_t}, \beta_{r_t})\, \mathrm{d}\pi \\ &= \prod_{s=1}^{S} \mathrm{BetaBinom}(y_{t,s}; n_{t,s}, \alpha_{r_t}, \beta_{r_t}). \end{aligned}$$

Here, $\mathrm{BetaBinom}(y; n, \alpha, \beta)$ denotes the density of a Beta-Binomial distribution, i.e.

$$\begin{aligned} &\mathrm{BetaBinom}(y; n, \alpha, \beta) \\ &\quad\coloneqq \binom{n}{y} \frac{\mathrm{B}(y+\alpha, n-y+\beta)}{\mathrm{B}(\alpha, \beta)} \\ &\quad= \frac{\Gamma(n+1)}{\Gamma(y+1)\Gamma(n-y+1)} \frac{\Gamma(y+\alpha)\Gamma(n-y+\beta)}{\Gamma(n+\alpha+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}, \end{aligned}$$

where $\mathrm{B}(\,\cdot\,,\,\cdot\,)$ and $\Gamma(\,\cdot\,)$ denote the Beta and Gamma functions, respectively.

The likelihood function is properly defined in the case $n_{t,s} = 0$ for all $s \in [S]$, that is when no reads at site t are available for all samples. The change point model thereby naturally allows for imputing the latent methylation state, although we leave an empirical comparison against alternative imputation methods (such as Ernst and Kellis 2015; Angermueller et al. 2017) for future work.

2.2.2 Change Point Model

We follow the setup of Fearnhead and Liu (2007a); Caron et al. (2012); Yildirim et al. (2013) and adopt a Markov model structure for the latent sequence $x_{1:T}$ – this will later facilitate the use of computationally effective particle filtering algorithms for the model fitting; for a similar modelling approach allowing message passing computations at the point of calibration, see Adams and MacKay (2007). Let $(x_t)_{t\in\mathbb{N}}$ be a latent Markov chain taking values in

$$\mathcal{X} \coloneqq \mathbb{N} \times [R].$$

If we write $x_t = (d_t, r_t)$ then, at CpG site t over the chromosome, d_t denotes the distance to the most recent change point, whereas $r_t \in [R]$ indicates the regime.

A-priori, Markov chain $(x_t)_{t\in\mathbb{N}}$ is assigned an initial distribution $f_{\theta,1}(x_1) \coloneqq \delta_1(d_1)\nu_\theta(r_1)$ – for some distribution ν_θ on $[R]$ – and transition kernels

$$\begin{aligned} f_{\theta,t+1}(x_{t+1}|x_t) \coloneqq\ & \rho_{\theta,t+1}(x_t)\delta_1(d_{t+1})P_\theta(r_{t+1}|r_t) \\ & + (1-\rho_{\theta,t+1}(x_t))\delta_{d_t+1}(d_{t+1})\delta_{r_t}(r_{t+1}). \end{aligned}$$

Here, given a current state $x_t = (d_t, r_t)$:

(i) $\rho_{\theta,t+1}(x_t) \in [0,1]$ is the probability of occurrence of a change point, after having spent d_t time steps in regime r_t. Following Caron et al. (2012), we select probability mass function h_{θ,r_t} with support on $\mathbb{N}$ (or on some subset thereof) such that $h_{\theta,r_t}(d_t)$ constitutes the prior probability over the sojourn time, d_t, in the r_t-th regime. Thus,

$$\rho_{\theta,t+1}(x_t) \coloneqq \frac{h_{\theta,r_t}(d_t)}{1 - H_{\theta,r_t}(d_t - 1)},$$

where $H_{\theta,r_t}(l) \coloneqq \sum_{k=1}^{l} h_{\theta,r_t}(k)$ denotes the cumulative distribution function associated with h_{θ,r_t}.

(ii) $P_\theta(r_{t+1}|r_t)$ is the r_{t+1}-th entry in the r_t-th row of an $R \times R$ matrix of transition probabilities determining the new generative model at the occurrence of a change point. To rule out trivial change points, we assume that this matrix has zeros on its diagonal.

Conditional on the model parameters θ, the posterior distribution of interest is then given by

$$p(x_{1:T}|y_{1:T},\theta) \propto \prod_{t=1}^{T} \gamma_{\theta,t}(x_{t-1},x_t),$$

where $\gamma_{\theta,t}(x_{t-1},x_t) \coloneqq f_{\theta,t}(x_t|x_{t-1})g_{\theta,t}(x_t)$.

2.2.3 Model Parameters

To ensure that the matrix P_θ is stochastic, we parametrise it as a function of model parameters $\theta_{1:((R-1)R)}$ (which are to be estimated from the data) in the following way which also avoids optimisation on the $(R-2)$-simplex, for $r \in [R]$,

$$P_\theta(r'|r) \coloneqq \begin{cases} \dfrac{\exp(\theta_{(R-1)(r-1)+r'})}{\sum_{i=1}^{R-1}\exp(\theta_{(R-1)(r-1)+i})}, & \text{if } r' < r, \\ 0, & \text{if } r' = r, \\ \dfrac{\exp(\theta_{(R-1)(r-1)+r'-1})}{\sum_{i=1}^{R-1}\exp(\theta_{(R-1)(r-1)+i})}, & \text{if } r' > r. \end{cases}$$

For any regime $r \in [R]$, we specify the prior distribution over the number and location of the change points as follows. Following Caron et al. (2012), we take $h_{\theta,r}(d) \coloneqq \text{Neg-Bin}(d - u_r; \kappa_r, \omega_r)\,\mathbf{1}\{d \geqslant u_r\}$ to be the probability mass function of a shifted Negative-Binomial distribution, where $u_r \geqslant 1$ is the shift, and

$$\text{Neg-Bin}(z;\kappa,\omega) \coloneqq \frac{\Gamma(z+\kappa)}{\Gamma(\kappa)\Gamma(z+1)}\omega^z(1-\omega)^\kappa,$$

for $z \in \mathbb{N} \cup \{0\}$. For any $r \in [R]$, we assume that the shifts $u_r \geqslant 1$ are known and parametrise the Negative-Binomial distributions as follows. The regime-specific 'success probability' parameter is given by

$$\omega_r \coloneqq \text{logit}^{-1}(\theta_{R(R-1)+r}) \in (0,1).$$

We follow Caron et al. (2012) and set $\kappa_r = 2$ for this regime-specific parameter, for all $r \in [R]$ in our experiments. Thus , the unknown parameter vector becomes

$$\theta \coloneqq \theta_{1:R^2} \in \mathbb{R}^{R^2}.$$

We note in passing that κ_r can alternatively also be estimated from the data.

2.2.4 Sequential Monte Carlo Algorithm

2.2.4.1 Filtering Algorithm

The filtering algorithm is outlined in Algorithm 1, where we use the notation $x_t^n = (d_t^n, r_t^n)$ for the n-th particle at time t. Furthermore, let $q_{\theta,1}$ represent some joint proposal distribution (on $\mathcal{X}^N$) for the particles drawn at the initial step; its n-th marginal distribution is denoted $q_{\theta,1}^n$. We recall here the expression $\gamma_{\theta,t}(x_{t-1}, x_t) = f_{\theta,t}(x_t|x_{t-1})g_{\theta,t}(x_t)$. We also write $W_{t-1}^n \coloneqq w_{t-1}^n / \sum_{m=1}^N w_{t-1}^m$, for any $n \in [N]$. We remark that our presentation differs from that in Yildirim et al. (2013) where the algorithm targets the one-step-ahead predictive distributions, whereas the algorithm here targets the filtering distributions. In the following, we write

$$M \coloneqq N - R.$$

Algorithm 1 (Particle Filter for Change-Point Models)**.**

1. *At time* $t = 1$*:*

   (a) *Sample* $x_1^{1:N} \sim q_{\theta,1}$.

   (b) *For for* $n \in [N]$*, set* $w_1^n \coloneqq \dfrac{\gamma_{\theta,1}(x_1^n)}{\sum_{m=1}^N q_{\theta,1}^m(x_1^n)}$.

2. *At time* t*,* $t > 1$*:*

   (a) ***Option I*** (Standard Resampling). *Sample* $a_{t-1}^{1:M} \in [N]^M$ *according to the weights* $W_{t-1}^{1:N}$ *via systematic, stratified, residual or multinomial resampling.*

   Option II (Optimal Finite-State Resampling). *Without loss of generality, assume here that the particles and weights are ordered so that* $W_{t-1}^1 \geqslant \ldots \geqslant W_{t-1}^N$*. Use Fearnhead (1998, Algorithm 5.2) to find* $C_{t-1} > 0$ *such that*

$$\sum_{n=1}^{N} [1 \wedge C_{t-1} W_{t-1}^n] = M,$$

   and set $K \coloneqq \max\{n \in [N] \mid C_{t-1} W_{t-1}^n \geqslant 1\}$*,* $L \coloneqq M - K$ *and* $I \coloneqq N - K$*. Set* $a_{t-1}^k \coloneqq k$*, for* $k \in [K]$*.*

   Generate ancestor indices $b^{1:L} \in [I]^L$ *via systematic or stratified resampling based on the weights* $V_{t-1}^i \coloneqq C_{t-1} W_{t-1}^{K+i} / L$*, for* $i \in [I]$*; set* $a_{t-1}^{K+l} \coloneqq K + b^l$*, for* $l \in [L]$*.*

   (b) *For* $n \in [N]$*, set*

$$x_t^n \coloneqq \begin{cases} (d_{t-1}^{a_{t-1}^n} + 1, r_{t-1}^{a_{t-1}^n}), & \text{for } n \leqslant M, \\ (1, n - M), & \text{for } n > M. \end{cases}$$

   (c) *For* $n \in [M]$*, set*

$$w_t^n \coloneqq \begin{cases} \left[\dfrac{1}{M} \displaystyle\sum_{m=1}^{N} w_{t-1}^m\right] \gamma_{\theta,t}(x_{t-1}^{a_{t-1}^n}, x_t^n), & \text{for } \textbf{\textit{Option I}}, \\ \dfrac{\gamma_{\theta,t}(x_{t-1}^{a_{t-1}^n}, x_t^n)}{1 \wedge C_{t-1} W_{t-1}^{a_{t-1}^n}} w_{t-1}^{a_{t-1}^n}, & \text{for } \textbf{\textit{Option II}}. \end{cases}$$

   For $n \in \{M+1, \ldots, N\}$*, set*

$$w_t^n \coloneqq \sum_{m=1}^{N} \gamma_{\theta,t}(x_{t-1}^m, x_t^n) w_{t-1}^m.$$

2.2.4.2 Parameter Estimation

We now describe the additional steps needed to update the parameters θ via online stochastic gradient-ascent steps. Write

$$\begin{aligned}\phi_{\theta,t}(x_{t-1}, x_t) &:= \nabla_\theta \log \gamma_{\theta,t}(x_{t-1}, x_t) \\ &= \nabla_\theta \log f_{\theta,t}(x_t|x_{t-1}) + \nabla_\theta \log g_{\theta,t}(x_t),\end{aligned}$$

with the usual convention that any quantity with site-subscript 0 is to be ignored from the notation. The required gradient expressions can be found in Appendix 2.7.5. With $(\eta_t)_{t\in\mathbb{N}}$ denoting some suitable step-size sequence for gradient-ascent algorithms, the parameters can be estimated online using Algorithm 2, which can be extended using adaptive preconditioning such as Adam (Kingma and Ba, 2014); also, it is possible to apply the gradient update step 2c only every ℓ-th step with $\ell \in \mathbb{N}$.

Algorithm 2 (parameter updates)**.** *Choose some initial value θ at the start of Algorithm 1.*

1. *At the end of Step 1 of Algorithm 1:*
   (a) *Set $\Phi_1^n := \phi_{\theta,1}(x_1^n)$, for $n \in [N]$.*
   (b) *Set $\nabla_1 := \sum_{n=1}^N W_1^n \Phi_1^n$.*
2. *At the end of Step 2 of Algorithm 1:*
   (a) *For $n \in [M]$, set*

$$\Phi_t^n := \Phi_{t-1}^{a_{t-1}^n} + \phi_{t,\theta}(x_{t-1}^{a_{t-1}^n}, x_t^n).$$

   For $n \in \{M+1, \ldots, N\}$, set

$$\Phi_t^n := \sum_{m=1}^N \frac{w_{t-1}^m f_{\theta,t}(x_t^n|x_{t-1}^m)}{\sum_{l=1}^N w_{t-1}^l f_{\theta,t}(x_t^n|x_{t-1}^l)} [\Phi_{t-1}^m + \phi_{t,\theta}(x_{t-1}^m, x_t^n)].$$

   (b) *Set $\nabla_t := \sum_{n=1}^N W_t^n \Phi_t^n$.*
   (c) *Set $\theta \leftarrow \theta + \eta_t(\nabla_t - \nabla_{t-1})$.*

2.2.4.3 Adaptive-Lag Smoothing for the Regime Indicators

Recall that $x_t = (d_t, r_t)$. Our main interest is in computing the marginal posterior probabilities of the regimes, $p(r_t = r|y_{1:T}, \theta)$, for $t \in [T]$, i.e. we want to compute expectations with respect to the marginal smoothing distributions $p(x_t|y_{1:T}, \theta)$, of the following form

$$\psi_{t|T,r} \coloneqq \mathbb{E}_{x_t \sim p(x_t|y_{1:T},\theta)} \left[\psi_{t,r}(x_t)\right],$$

for functionals $\psi_{t,r}(x_t) \coloneqq \mathbf{1}\{r_t = r\}$, for all $r \in [R]$ and all $t \in [T]$. Assuming that the model has sufficient forgetting properties, i.e. the model is such that $p(x_t|y_{1:T}, \theta) \approx p(x_t|y_{1:(t+\Delta)\wedge T}, \theta)$ for some sufficiently large lag $\Delta \geqslant 0$, these expectations can be approximated by

$$\psi_{t|(t+\Delta)\wedge T,r} \coloneqq \mathbb{E}_{x_t \sim p(x_t|y_{1:t+\Delta\wedge T},\theta)} \left[\psi_{t,r}(x_t)\right],$$

without introducing too much bias. However, while the bias decreases with Δ, the computational cost of computing (or, at least approximating) these expectations increases with Δ. Thus, a sensible choice of the lag Δ is crucial. However, manually tuning Δ is typically difficult.

Here, we propose to tune Δ automatically (and separately for each expectation of interest) via the adaptive-lag smoother from Alenlöv and Olsson (2019). Loosely speaking, this approach exploits that $\mathrm{var}_{x_t \sim p(x_t|y_{1:t+\Delta},\theta)}[\psi_{t,r}(x_t)] \downarrow 0$ as $\Delta \uparrow \infty$. This motivates approximating $\psi_{t|T,r}$ by $\psi_{t|(t+\Delta_{t,r}(\varepsilon))\wedge T,r}$, where

$$\Delta_{t,r}(\varepsilon) \coloneqq \min\left\{k \geqslant 0 \,\middle|\, \mathrm{var}_{x_t \sim p(x_t|y_{1:t+k\wedge T},\theta)}[\psi_{t,r}(x_t)] < \varepsilon\right\},$$

for some threshold $\varepsilon > 0$ chosen by the user. Below we summarise the particle-filter approximation based of this idea.

It should be clear that we can use this algorithm to estimate smoothed functionals other than the posterior probabilities of the regimes. Therefore, in the algorithm given below, we assume that we are interested in a family of Q such test functions at each position, denoted $\{\psi_{t,q}\}_{q \in [Q]}$. In other words, the marginal posterior probabilities of the R regimes can be approximated by taking $Q \coloneqq R$ and $\psi_{t,q}(x_t) \coloneqq \mathbf{1}\{r_t = q\}$ in Algorithm 3.

Algorithm 3 (Smoothing the Regime Indicators)**.**

1. *At the end of Step 1 of Algorithm 1:*

   (a) *Set* $\mathcal{S} \leftarrow \{(1, q) \mid q \in [Q]\}$.

   (b) *Set* $\Psi^n_{1|1,q} := \psi_{1,q}(x^n_1)$, *for* $n \in [N]$ *and* $q \in [Q]$.

2. *At the end of Step 2 of Algorithm 1:*

   (a) *Set* $\mathcal{S} \leftarrow \mathcal{S} \cup \{(t, q) \mid q \in [Q]\}$.

   (b) *For any* $(s, q) \in \mathcal{S}$ *and* $n \in [N]$, *set*

$$\Psi^n_{s|t,q} := \begin{cases} \psi_{t,q}(x^n_t), & \text{if } s = t, \\ \Psi^{a^n_{t-1}}_{s|t-1,q}, & \text{if } s < t \text{ and } n \leqslant M, \\ \sum_{m=1}^{N} \dfrac{w^m_{t-1} f_{\theta,t}(x^n_t | x^m_{t-1})}{\sum_{l=1}^{N} w^l_{t-1} f_{\theta,t}(x^n_t | x^l_{t-1})} \Psi^m_{s|t-1,q}, & \text{if } s < t \text{ and } n > M. \end{cases}$$

   (c) *For any* $(s, q) \in \mathcal{S}$*: If*

$$\sum_{n=1}^{N} W^n_t \left(\Psi^n_{s|t,q} - \sum_{m=1}^{N} W^m_t \Psi^m_{s|t,q} \right)^2 < \varepsilon,$$

   set $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(s, q)\}$ *and return* $\sum_{n=1}^{N} W^n_t \Psi^n_{s|t,q}$ *as an estimate of* $\psi_{s|T,q}$.

2.3 Case-Control Methylome Change Point Model

2.3.1 Notation and Setup

In this section, we describe an extended change-point model for the case/control scenario. That is, we now have observations for two groups: *controls* and *cases*. The observations in each group will be modelled as a change-point model, i.e. we now have two change-point models: one for the control group and one for the case group. However, these two models *are not independent*. While the marginal dynamics of the control group are described by the above single group model, the methylation states for the case group become dependent on the methylation signal of the control group.

We shall use the convention that the 'bar'-accented quantities are associated with the 'control' group whereas 'tilde'-accented quantities are associated with the 'case' group. That is, for the respective groups,

(i) $\bar{y}_t \coloneqq \bar{y}_{t,1:\bar{S}}$ and $\tilde{y}_t \coloneqq \tilde{y}_{t,1:\tilde{S}}$ are the observed numbers of methylated reads,

(ii) $\bar{n}_t \coloneqq \bar{n}_{t,1:\bar{S}}$ and $\tilde{n}_t \coloneqq \tilde{n}_{t,1:\tilde{S}}$ are the total number of reads,

(iii) $\bar{\pi}_t \coloneqq \bar{\pi}_{t,1:\bar{S}}$ and $\tilde{\pi}_t \coloneqq \tilde{\pi}_{t,1:\tilde{S}}$ are the 'methylation probabilities',

(iv) $\bar{x}_t \coloneqq (\bar{d}_t, \bar{r}_t)$ and $\tilde{x}_t \coloneqq (\tilde{d}_t, \tilde{r}_t)$ are the change-point-model states.

To keep the notation concise, we also write

$$\begin{aligned} y_t &\coloneqq (\bar{y}_t, \tilde{y}_t), \\ n_t &\coloneqq (\bar{n}_t, \tilde{n}_t), \\ \pi_t &\coloneqq (\bar{\pi}_t, \tilde{\pi}_t), \\ x_t &\coloneqq (z_t, \bar{x}_t, \tilde{x}_t) = (z_t, \bar{d}_t, \bar{r}_t, \tilde{d}_t, \tilde{r}_t), \end{aligned}$$

where z_t denotes an additional binary latent variable which governs the dependence between the two groups.

2.3.2 Data and Likelihood Conditional on Change Points

As before, we use a Beta-Binomial model, i.e. we assume that

$$\begin{aligned} &p(y_{1:T}|\pi_{1:T}, x_{1:T}, \theta) \\ &\quad = \prod_{t=1}^{T} \Bigg[\prod_{s=1}^{\bar{S}} \mathrm{Binom}(\bar{y}_{t,s}; \bar{n}_{t,s}, \bar{\pi}_{t,s}) \Bigg] \Bigg[\prod_{s=1}^{\tilde{S}} \mathrm{Binom}(\tilde{y}_{t,s}; \tilde{n}_{t,s}, \tilde{\pi}_{t,s}) \Bigg]. \end{aligned}$$

The main difference with the single-group scenario is that probabilities $\bar{\pi}_{t,s}$ and $\tilde{\pi}_{t,s}$ are now not necessarily *identically* distributed conditionally on the latent regimes/change points. That is, we model

$$p(\pi_{1:T}|x_{1:T}, \theta) \coloneqq \prod_{t=1}^{T} \Bigg[\prod_{s=1}^{\bar{S}} \mathrm{Beta}(\bar{\pi}_{t,s}; \alpha_{\bar{r}_t}, \beta_{\bar{r}_t}) \Bigg] \Bigg[\prod_{s=1}^{\tilde{S}} \mathrm{Beta}(\tilde{\pi}_{t,s}; \alpha_{\tilde{r}_t}, \beta_{\tilde{r}_t}) \Bigg],$$

As in the single-group scenario, we can analytically integrate out all the probabilities $\pi_{1:T}$ to obtain the product of Beta-Binomial likelihoods

$$p(y_{1:T}|x_{1:T}, \theta) = \int p(y_{1:T}|\pi_{1:T}, x_{1:T}, \theta) p(\pi_{1:T}|x_{1:T}, \theta) \, \mathrm{d}\pi_{1:T} = \prod_{t=1}^{T} g_{\theta,t}(x_t),$$

where, writing

$$\begin{aligned} \bar{g}_{\theta,t}(\bar{x}_t) &\coloneqq \prod_{s=1}^{\bar{S}} \mathrm{BetaBinom}(\bar{y}_{t,s}; \bar{n}_{t,s}, \alpha_{\bar{r}_t}, \beta_{\bar{r}_t}), \\ \tilde{g}_{\theta,t}(\tilde{x}_t) &\coloneqq \prod_{s=1}^{\tilde{S}} \mathrm{BetaBinom}(\tilde{y}_{t,s}; \tilde{n}_{t,s}, \alpha_{\tilde{r}_t}, \beta_{\tilde{r}_t}). \end{aligned}$$

we have set

$$g_{\theta,t}(x_t) \coloneqq \bar{g}_{\theta,t}(\bar{x}_t)\tilde{g}_{\theta,t}(\tilde{x}_t).$$

2.3.3 Change Point Model

Recall that each state of the latent Markov chain $(x_t)_{t\in\mathbb{N}}$ now takes the form

$$x_t = (z_t, \bar{x}_t, \tilde{x}_t) = (z_t, \bar{d}_t, \bar{r}_t, \tilde{d}_t, \tilde{r}_t) \in \mathcal{X} \coloneqq \{0,1\} \times (\mathbb{N} \times [R])^2,$$

where the interpretation of $\bar{x}_t = (\bar{d}_t, \bar{r}_t)$ and $\tilde{x}_t = (\tilde{d}_t, \tilde{r}_t)$ is exactly as in a standard change-point model. In particular, at each site t, the regime indicators $\bar{r}_t$ and $\tilde{r}_t$ take values in $[R]$. The binary latent variable z_t governs the dependence structure between the two groups, i.e. $z_t = 1$ indicates that the two groups are *merged*, whereas $z_t = 0$ indicates that they are *split*. More precisely, a-priori, the Markov chain $(x_t)_{t\in\mathbb{N}}$ has some initial distribution

$$f_{\theta,1}(x_1) = Q_{\theta,1}(z_1)\bar{f}_{\theta,1}(\bar{x}_1)\big[\mathbf{1}\{z_1 = 1\}\delta_{\bar{x}_1}(\tilde{x}_1) + \mathbf{1}\{z_1 = 0\}\tilde{f}_{\theta,1}(\tilde{x}_1|\bar{r}_1)\big],$$

on $\mathcal{X}$ and transition kernels motivated in more detail below in the form of

$$\begin{aligned} f_{\theta,t+1}(x_{t+1}|x_t) \coloneqq\ & \bar{f}_{\theta,t+1}(\bar{x}_{t+1}|\bar{x}_t)Q_{\theta,t+1}(z_{t+1}|x_t) \\ & \times \big[\mathbf{1}\{z_{t+1} = 1\}\delta_{\bar{x}_t}(\tilde{x}_t) \\ & \quad + \mathbf{1}\{(z_t, z_{t+1}) = (0,0) \text{ and } \bar{r}_{t+1} \neq \tilde{r}_t\}\tilde{f}_{\theta,t+1}(\tilde{x}_{t+1}|\tilde{x}_t; \bar{r}_{t+1}) \\ & \quad + \mathbf{1}\{(z_t, z_{t+1}) = (0,0) \text{ and } \bar{r}_{t+1} = \tilde{r}_t\}\tilde{f}'_{\theta,t+1}(\tilde{x}_{t+1}|\tilde{x}_t; \bar{r}_{t+1}) \\ & \quad + \mathbf{1}\{(z_t, z_{t+1}) = (1,0) \text{ and } \bar{d}_{t+1} \neq 1\}\tilde{f}_{\theta,1}(\tilde{x}_{t+1}|\bar{r}_{t+1}) \\ & \quad + \mathbf{1}\{(z_t, z_{t+1}) = (1,0) \text{ and } \bar{d}_{t+1} = 1\}\tilde{f}_{\theta,t+1}(\tilde{x}_{t+1}|\tilde{x}_t; \bar{r}_{t+1})\big]. \end{aligned}$$

Here, we have defined some of the quantities used above as follows.

(i) The distribution $Q_{\theta,1}(z_1)$ and the transition kernel $Q_{\theta,t+1}(z_{t+1}|x_t)$ have support $\{0,1\}$ and govern the distribution of the latent binary variable.

(ii) The initial distributions for the two groups, $\bar{f}_{\theta,1}(\bar{x}_1) \coloneqq \delta_1(\bar{d}_1)\bar{\nu}_\theta(\bar{r}_1)$ and $\tilde{f}_{\theta,1}(\tilde{x}_1|\bar{r}_1) \coloneqq \delta_1(\tilde{d}_1)\tilde{\nu}_\theta(\tilde{r}_1|\bar{r}_1)$ are similarly specified as with the single-group scenario. The only non-standard aspect here is that for the case group we enforce that its regime avoids the regime of the control group, i.e. $\tilde{\nu}_\theta(\tilde{r}_1|\bar{r}_1) = 0$ for $\tilde{r}_1 = \bar{r}_1$.

(iii) The transition kernels

$$\begin{aligned}
\bar{f}_{\theta,t+1}(\bar{x}_{t+1}|\bar{x}_t) &\coloneqq \big[\bar{\rho}_{\theta,t+1}(\bar{x}_t)\delta_1(\bar{d}_{t+1})\bar{P}_\theta(\bar{r}_{t+1}|\bar{r}_t) \\
&\qquad + (1-\bar{\rho}_{\theta,t+1}(\bar{x}_t))\delta_{\bar{d}_t+1}(\bar{d}_{t+1})\delta_{\bar{r}_t}(\bar{r}_{t+1})\big], \\
\tilde{f}_{\theta,t+1}(\tilde{x}_{t+1}|\tilde{x}_t;\bar{r}_{t+1}) &\coloneqq \big[\tilde{\rho}_{\theta,t+1}(\tilde{x}_t)\delta_1(\tilde{d}_{t+1})\tilde{P}_\theta(\tilde{r}_{t+1}|\tilde{r}_t;\bar{r}_{t+1}) \\
&\qquad + (1-\tilde{\rho}_{\theta,t+1}(\tilde{x}_t))\delta_{\tilde{d}_t+1}(\tilde{d}_{t+1})\delta_{\tilde{r}_t}(\tilde{r}_{t+1})\big], \\
\tilde{f}'_{\theta,t+1}(\tilde{x}_{t+1}|\tilde{x}_t;\bar{r}_{t+1}) &\coloneqq \delta_1(\bar{d}_{t+1})\tilde{P}_\theta(\tilde{r}_{t+1}|\tilde{r}_t;\bar{r}_{t+1}),
\end{aligned}$$

are again similar to the single-group scenario. The only non-standard aspect is that for the case group, we enforce that its regime avoids the regime of the control group, i.e. $\tilde{P}_\theta(\tilde{r}_{t+1}|\tilde{r}_t;\bar{r}_{t+1}) = 0$ for $\tilde{r}_{t+1} = \bar{r}_{t+1}$.

The dynamics of the full state transition described by the kernel $f_{\theta,t+1}(x_{t+1}|x_t)$ can be clarified by noting first that the marginal dynamics of the control group $\bar{f}_{\theta,t+1}(\bar{x}_{t+1}|\bar{x}_t)$ coincide with the single-group model. Second, the transition dynamics of the split indicator $Q_{\theta,t+1}(z_{t+1}|x_t)$ depend potentially on the previous states of the case and control group. Third, conditional on the next state of the control group and the next latent indicator variable, the dynamics of the case group can be motivated by looking at the following different configurations.

(a) The two groups are merged at the next site: The case states then coincide with the control states.

(b) The two groups have been split, remain split and the control group regime does not jump into the previous regime of the case group: The case states then evolve according to the change-point transition kernel $\tilde{f}_{\theta,t+1}(\tilde{x}_{t+1}|\tilde{x}_t;\bar{r}_{t+1})$ with change-point probability function $\tilde{\rho}_{\theta,t+1}$ and a regime transition matrix $\tilde{P}_\theta$ that avoids the regime of next control group.

(c) The two group have been split, remain split, but the control group regime jumps onto the the previous regime of the case group: The case stats then automatically changes the regime according to the regime transition matrix $\tilde{P}_\theta$.

(d) The groups have previously been merged and become split next with no change point occurring in the control group: The case group then has a change point sampled from the initial distribution $\tilde{f}_{\theta,1}$.

(e) The groups have previously been merged and become split next with a change point occurring in the control group: The case state then evolves according to the change-point transition kernel $\tilde{f}_{\theta,t+1}(\tilde{x}_{t+1}|\tilde{x}_t;\bar{r}_{t+1})$ as in configuration (a).

Conditionally on the model parameters θ, the posterior distribution of interest is then given by

$$p(x_{1:T}|y_{1:T},\theta) \propto \prod_{t=1}^{T} \gamma_{\theta,t}(x_{t-1},x_t),$$

where $\gamma_{\theta,t}(x_{t-1},x_t) \coloneqq f_{\theta,t}(x_t|x_{t-1})g_{\theta,t}(x_t)$.

2.3.4 Model Parameters

The transition kernel of the control group is modelled as in the single-group case described in Subsection 2.2.2, i.e. $\bar{f}_{\theta,t}$ corresponds to $f_{\theta,t}$ therein with the model parameters as in Subsection 2.2.3.

Different choices for the transition kernel of the latent variable z_t are possible. As a first example, we consider a shifted Geometric distribution for the duration between change points of z_t. Suppose that

$$Q_{\theta,t+1}(z_{t+1}|x_t) \coloneqq \begin{cases} q_{z_t+1,z_{t+1}+1}, & \text{if } \tilde{d}_t \wedge \bar{d}_t \geqslant u, \\ \delta_{z_t}(z_{t+1}), & \text{otherwise,} \end{cases}$$

where $q_{i,j}$ denotes the (i,j) -element of a 2-by-2 row-stochastic matrix

$$\begin{bmatrix} 1-q_{\text{MERGE}} & q_{\text{MERGE}} \\ q_{\text{SPLIT}} & 1-q_{\text{SPLIT}} \end{bmatrix},$$

and $u \geqslant 0$ is the minimum distance between jumps of the chain $(z_t)_{t\geqslant 1}$. Although a more flexible model parametrisation for the latent dynamics of the case group is possible in principle that could also be estimated online, we just fix the regime transition probability for the case group via

$$\tilde{P}_\theta(\tilde{r}_{t+1}|\tilde{r}_t;\bar{r}_{t+1}) \coloneqq \frac{1}{R-2}\,\mathbf{1}\{\tilde{r}_{t+1} \in [R] \setminus \{\tilde{r}_t,\bar{r}_{t+1}\}\},$$

and similarly assume for the regime transition density of the split moves that

$$\tilde{\nu}_\theta(\tilde{r}_{t+1}|\bar{r}_{t+1}) \coloneqq \frac{1}{R-1}\,\mathbf{1}\{\tilde{r}_{t+1} \in [R] \setminus \{\bar{r}_{t+1}\}\}.$$

The probability $\tilde{\rho}_{\theta,t+1}(\tilde{x}_t)$ of a change point in the case group occurring under the scenario $(z_t,z_{t+1}) = (0,0)$ and $\bar{r}_{t+1} \neq \tilde{r}_t$ is modelled analogously to Subsection 2.2.2,

i.e. as the hazard function of a probability mass function $\tilde{h}_{\theta,\tilde{r}_t}$, with $\tilde{h}_{\theta,\tilde{r}}$ a shifted Negative-Binomial function $\tilde{h}_{\theta,r}(\tilde{d}) = \text{Neg-Bin}(\tilde{d} - \tilde{u}_r; \tilde{\kappa}_r, \tilde{\omega}_r)$ as in Subsection 2.2.3.

Since splits are in practice likely to occur only very infrequently, it would be difficult to estimate any of the parameters mentioned in this section from the data. Therefore, we fix $\tilde{\kappa}_r = \tilde{\kappa} = 2$ and assume that $\tilde{\omega}_r = \tilde{\omega} \in (0,1)$ and $\tilde{u}_r = \tilde{u} \geqslant 0$, where $\tilde{u} \geqslant 0$ and $\tilde{\omega} > 0$, along with the probabilities $q_{\text{SPLIT}}, q_{\text{MERGE}} \in (0,1)$, are specified by the user.

2.3.5 Sequential Monte Carlo Algorithm

2.3.5.1 Filtering Algorithm

A particle filtering algorithm that makes use of the discrete latent state space is described in detail in Appendix 2.7.3 and outlined in Algorithm 5. Similarly to the filtering algorithm for the single-group model, such a discrete particle filter (Fearnhead, 1998; Fearnhead and Clifford, 2003; Whiteley et al., 2010) does not rely on random proposals but explores the state space systematically for any possibility. In the two-group model, there are at most $I = 2R + R^2$ possibilities how each state can evolve with each possibility probed in the particle filter. To remain computationally feasible also for many CpG sites, the particle system is pruned down by resampling M particle lineages before the next systematic exploration step. Both unbiased or optimal resampling schemes can be used as before. For large T, the total number of particles becomes $N = MI$ and the filter has a linear complexity $\mathcal{O}(N)$.

2.3.5.2 Inferences on the methylation signal at CpG sites or genomic regions

We are often interested in making inferences on the latent states in entire genomic regions, say from site s to t, or just at a single CpG site with $s = t$. For a given test function $\psi_{s:t,q}$ on $\mathcal{X}^{t-s}$, we therefore aim to compute the expectation with respect to the joint distribution of all latent states in the region given the observations at sites 1 to T of the whole chromosome, that is $\mathbb{E}_{x_{s:t} \sim p(x_{s:t}|y_{1:T},\theta)}[\psi_{s:t,q}(x_{s:t})]$. For some distribution ν on $\mathcal{X}$, consider the backward kernel $B^{\nu}_{t,\theta}(\mathrm{d}x_t|x_{t+1}) \propto \nu(\nabla x_t) f_{\theta,t+1}(x_{t+1}|x_t)$ that allows to express expectations over regions recursively using the decomposition

$$\mathbb{E}_{x_{s:t} \sim p(x_{s:t}|y_{1:T},\theta)}\left[\psi_{s:t,q}(x_{s:t})\right] = \int \tilde{\pi}_T(\mathrm{d}x_T) \prod_{\ell=s}^{T-1} \int B^{\tilde{\pi}_\ell}_{\ell,\theta}(\mathrm{d}x_\ell|x_{\ell+1}) \psi_{s:t,q}(x_{s:t}),$$

where $\tilde{\pi}_\ell(x_\ell) = p(x_\ell|y_{1:T},\theta)$ is the (marginal) filter distribution. This expectation is generally intractable. The Forward Filter Backward Simulation algorithm (Godsill

et al., 2004) uses an approximation $\hat{\pi}(x_\ell)$ of the marginal filter distributions $\tilde{\pi}_\ell(x_\ell)$ as obtained by a particle filter. After running the particle filter, one first samples $\hat{x}_T \sim \hat{\pi}_T(x_T)$ and then recursively generates a backward trajectory $\hat{x}_\ell \sim B_{\ell,\theta}^{\hat{\pi}_\ell}(\hat{x}_\ell|\hat{x}_{\ell+1})$ for ℓ from $T-1$ to s, as shown in Algorithm 4 where K such trajectories are simulated, potentially in parallel. The complexity of the backward simulation algorithm is $\mathcal{O}(NK)$. While K can be chosen arbitrarily, a rule-of-thumb (Lindsten and Schön, 2013) can be $K < N$.

In the case/control scenario, it is of particular interest to approximate expectations for test functions $\psi_{t,q}(x_t) = \mathbf{1}\{z_t = 1\}$ under the posterior. Indeed, this expectation is the probability $\psi_{t|T,q} \coloneqq \mathbb{E}[\psi_{t,q}(x_t)|y_{1:T}, \theta] = p(z_t = 1|y_{1:T}, \theta)$, i.e. the probability of the two groups being in the same regime at CpG position t. Other test functions such as $\psi_{t,q}(x_t) = \mathbf{1}\{\bar{r}_t = r\}$ may also be of interest (as in the single-group scenario).

Algorithm 4 (Estimation of test functions using backward simulation)**.**

1. *For* $k \in [K]$*:*
   (a) *Sample* $b_T^k \sim Cat(w_T)$ *and set* $\hat{x}_T = x_T^{b_T^k}$.
   (b) *For* $t = T-1$ `to` s*:*
      i. *Compute* $\hat{w}_{t,k}^i = \frac{w_t^i f_{\theta,t+1}(\hat{x}_{t+1}^k|x_t^i)}{\sum_{j=1}^{N_t} w_t^j f_{\theta,t+1}(\hat{x}_{t+1}^k|x_t^j)}$ *for all* $i \in [N_t]$.
      ii. *Sample* $b_t^k \sim Cat(\hat{w}_{t,k})$.
      iii. *Set* $\hat{x}_t^k = x_t^{b_t^k}$.
2. `Return` $\frac{1}{K}\sum_{k=1}^K \psi_{s:t,q}(\hat{x}_{s:t}^k)$.

2.3.6 Identification of DMPs or DMRs and FDR control

Identification of differentially methylated positions (DMPs) is a multiple testing problem. The observed methylation signals are spatially correlated and accounting for such spatial dependencies can allow for more efficient multiple testing procedures. While multiple testing approaches designed for the independent setting can be valid to control the false discovery rate (FDR) also under dependence (Benjamini and Yekutieli, 2001), such procedures can be overly conservative. In this article, the dependence structure is described by a parametric hidden Markov model. Such an approach can allow for optimal procedures under very idealised assumptions that

minimize the false non-discovery rate, i.e. maximize some form of multiple testing power, subject to a constraint on the FDR (Sun and Tony Cai, 2009). Such idealised procedures require knowledge of the model parameters and can be approximated by a data-driven procedure using consistent estimates of the generative parameters and the latent states. Although such guarantees do not necessarily hold for the approach suggested here - some model parameters in the two-group model are not estimated for instance - it can still perform competitively compared to different methods for detecting DMPs.

Our approach is to classify CpG site t to be a DMP if $z_t = 0$, i.e. the latent methylation state is in different regimes for the case and control group by testing the null hypotheses $\mathsf{H}^0_{t,c} : z_t = 1$ against the alternatives $\mathsf{H}^1_{t,c} : z_t = 0$ for all sites $t \in T_c$ and chromosomes $c \in [C]$ simultaneously. A CpG-site at position t that is classified as differentially methylated is considered a true positive if $z_t = 0$ and a false positive if $z_t = 1$. DMPs can then be identified in a compound decision theoretic framework subject to a control on the posterior expected False Discovery Rate (FDR), see for instance Müller et al. (2004); Müller et al. (2007); Cui et al. (2015).

Suppose that $\hat{p}_{t,c}$ is an estimate of $p(z_t = 1|y_{1:T}, \theta)$ at position t of chromosome c, which is also known as the local index of significance (Sun and Tony Cai, 2009), assuming θ is the true parameter vector of chromosome c. Denote the ordered estimates of $\{\hat{p}_{t,c}\}_{t\in[T_c],c\in[C]}$ by $\hat{p}_{(1)} \leqslant \ldots \leqslant \hat{p}_{(T')}$ with $T' = \sum_{c=1}^{C} T_c$ and write the corresponding null hypotheses as $\mathsf{H}^0_{(1)}, \ldots, \mathsf{H}^0_{(T')}$. We consider a decision rule of the form $\delta(y) = \{\mathbf{1}\{\hat{p}_{t,c} \leqslant \lambda\}\}_{t\in[T],c\in[C]}$ for some threshold λ which has optimality guarantees in an oracle setting. The threshold λ depends on observations y from all chromosomes and is chosen to control the FDR genome-wide at some level $\alpha > 0$, see also Wei et al. (2009), as follows. Let

$$\hat{Q}_s = \frac{1}{s} \sum_{t=1}^{s} \hat{p}_{(t)}$$

be an approximation of the marginal FDR $\frac{\mathbb{E}\, V_\lambda}{\mathbb{E}\, R_\lambda}$ where $R_\lambda = \sum_{t=1}^{T'} \mathbf{1}\{\hat{p}_{(t)} < \lambda\}$ and $V_\lambda = \sum_{t=1}^{T'} \mathbf{1}\{\hat{p}_{(t)} < \lambda\}\, \mathbf{1}\{z_t = 1\}$ is the number of rejections and false positive results, respectively. We then compute

$$k = \max\{s \in [T'] \colon \hat{Q}_s \leqslant \alpha\}$$

and reject all $\mathsf{H}^0_{(s)}$ for $s \in [k]$. The test statistic $\hat{p}_{(s)}$ can be computed using trajectories from the backward sampling algorithm with the test function $\psi_t(x_t) = \mathbf{1}\{z_t = 1\}$.

Instead of testing individual locations, and possibly reporting them in terms of clusters if the locations are next to each other, it is also possible to test hypotheses over regions or clusters. We assume that such regions of interest are known a-priori, say particular genes or obtained after some preliminary CpG-wise analysis. Testing for a global null hypothesis, i.e. against the alternative that at least one CpG site in a region is differentially methylated, may not support strong scientific conclusions. Conversely, tests against the alternative that all CpG sites of a region are differentially methylated can be difficult to reject. A compromise can be to test the partial conjunction hypotheses (Benjamini and Heller, 2008) that the proportion of differentially methylated sites is below some tolerance level γ, $\mathsf{H}^0_k : \pi_k \leqslant \gamma$ versus $\mathsf{H}^1_k : \pi_k > \gamma$ simultaneously for any region R_k from the regions of interest $\{R_1, \ldots, R_K\}$ and

$$\pi_k = \frac{\sum_{t=1}^T \mathbf{1}\{t \in R_k\}\, \mathbf{1}\{z_t = 0\}}{\sum_{t=1}^T \mathbf{1}\{t \in R_k\}}.$$

We follow the procedure from Sun et al. (2015) that aims to minimize the missed cluster rate (MCR) while controlling the false cluster rate (FCR). Let $\vartheta_k = \mathbf{1}\{\pi_k > \gamma\}$ and associate a weight ω_k to each region R_k. Assume that $\hat{p}_k$ is an approximation to the test statistics $p(\vartheta_k = 0|y, \theta)$ with their ordered statistics written as $\hat{p}_{(1)} \leqslant \ldots \leqslant \hat{p}_{(K)}$ with the corresponding hypotheses $\mathsf{H}^0_{(k)}$ and weights $\omega_{(k)}$. We compute

$$r = \max\left\{ j \in [k] \colon \frac{\sum_{k=1}^j \omega_{(k)} \hat{p}_{(k)}}{\sum_{k=1}^j \omega_{(k)}} \leqslant \alpha \right\}$$

and reject all $\mathsf{H}^0_{(k)}$ for $k \in [r]$. Approximating the test statistics requires an approximation to the joint law of the latent states in a region. Indeed, for region R_k with start s^k_1 and end location $s^k_{T_k}$, we can approximate $p(\vartheta_k = 0|y, \theta) = \mathbb{E}\left[\mathbf{1}\{\pi_k \leqslant \gamma\}|y, \theta\right]$ by taking expectations with respect to the law of the latent states in the region of the function

$$\psi_{s^k_1 : s^k_{T_k}}(x_{s^t_1 : s^k_{T_k}}) = \mathbf{1}\left\{ \frac{1}{T_k} \sum_{s=s^k_1}^{s^k_{T_k}} \mathbf{1}\{z_s = 0\} \leqslant \gamma \right\}$$

using the backward simulation algorithm.

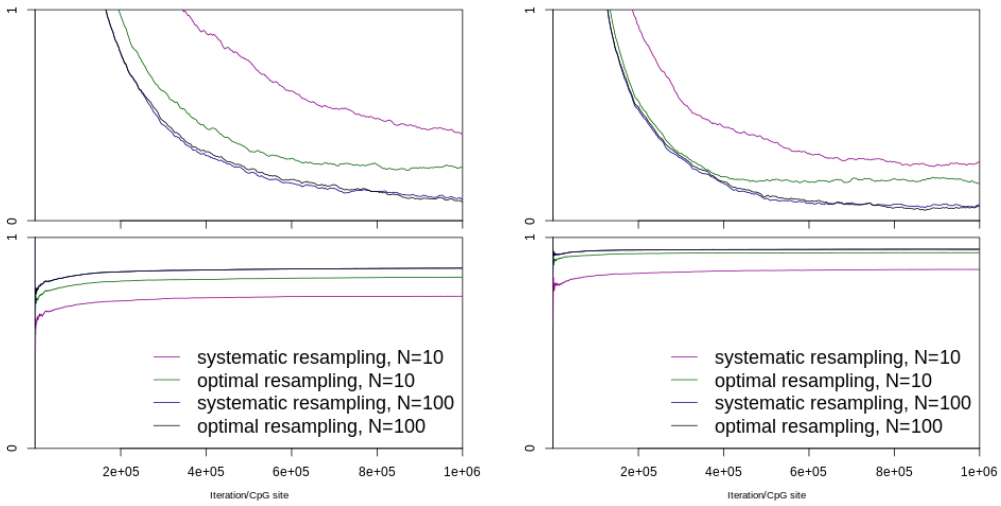
2.4 Simulation studies

2.4.1 Regime & Parameter Inference for Single Group Model

We make use of a simulation study to evaluate the performance of the single-group model, analysing the parameter estimation as well as the inference of the latent regimes in the case of both low and high read depths. We start by simulating two types of data sets, with the first one using an average read depth of 10, whereas the second one uses an average read depth of 100. To generate the model parameters used for simulating the data, we draw ω_r from a uniform distribution on the interval $[0.1, 0.9]$ and P_{ij} from a Dirichlet distribution with concentration parameter $\frac{2}{3}$ for $i \neq j$. For each setting, one million CpG sites are simulated and we repeat the simulation to create 20 replicates. We consider varying the number of particles N from $N \in \{10, 100\}$ and compare the systematic resampling with the optimal resampling scheme. We use Adam (Kingma and Ba, 2014) as an adaptive stochastic gradient algorithm for the parameter estimation with the default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and consider learning rate values in $\{0.05, 0.01, 0.002\}$. The evolution of the average error of the estimated parameters relative to the simulated ones is shown in Figure 2.2 for a single data set with a read depth of either 10 or 100. It illustrates the superior performance of the optimal resampling scheme when only a small number of particles such as $N = 10$ is used, whereas the performance of both schemes is similar for $N = 100$ particles. Figure 2.2 also shows the posterior probability of the true regimes, indicating that a higher read depth yields a higher estimated probability of the true regimes and that the optimal resampling scheme performs better for $N = 10$ particles, see also Table 2.2 for average values across all data sets. Averages over all data sets of the L_1-errors for both the regime transition and the parameter ω governing the sojourn times are given in Tables 2.6 and 2.7 for an average read depth of 10. Table 2.5 shows the estimated probability of the true regime state across all data sets for an average read depth of 100.

2.4.2 Particle Algorithms in Two-Group Models

We assess the performance of the particle algorithm in the case-control model by generate 10 data sets of $10,000$ CpG sites with average read depth of 100 using the regimes from Figure 2.1. For each data set, we perform the filtering algorithms with either the unbiased or optimal resampling schemes while varying the number



(a) Average read depth of 10.

(b) Average read depth of 100.

Figure 2.2: Average L_1-error during optimization (top) and the cumulative mean of the probability of the true regime state (bottom) using ADAM with a step size of 0.01 applied after every 200 iterations using one simulated data set with average read depth of 10 in 2.2a and average read depth of 100 in 2.2b.

$N \in \{1200, 2400, 4800, 9600, 19200\}$ of particles and using the true static parameters. Estimates of the log-likelihood from the filtering algorithm for the first two data sets is shown in Figure 2.3 for 10 different replications. The optimal resampling scheme with $N \geqslant 9600$ yields large and low-variance estimates of the log-likelihood. To assess the classification of DMPs, we have estimates the posterior probabilities of the latent split states using $K = 100$ backward trajectories. The resulting area under the curve (AUC) averaged over the 10 data sets in Table is shown in 2.3, which indicates better performance for the optimal resampling scheme with $N \geqslant 9600$.

2.5 DNA Methylation and aging

We analyse two methylomes from blood samples (CD14-positive, CD16-negative classical monocytes) for a female *adult* donor (age 60-65) and a female *newborn* donor as studied in Libertini et al. (2016a,b) with accession code EGAD00001001261 [2] from the European Genome-Archive (EGA) . Counts of unmethylated and methylated cytosine in CpG context for chromosomes 1 to 22 are obtained using GemBS (Merkel et al., 2019). The adult samples contain 23.740 million sites with a median coverage of

[2] http://dcc.blueprint-epigenome.eu/#/datasets/EGAD00001001261

Table 2.2: Average estimated posterior probability of the true regimes for 20 replicates with average read depth of 10.

frequency of gradient update	1			200		
learning rate	0.05	0.01	0.002	0.05	0.01	0.002
systematic resampling with N=10	0.575	0.661	0.690	0.702	0.696	0.664
optimal resampling with N=10	0.768	0.798	0.801	0.803	0.797	0.766
systematic resampling with N=100	0.835	0.842	0.843	0.843	0.838	0.811
optimal resampling with N=100	0.840	0.844	0.844	0.844	0.838	0.811

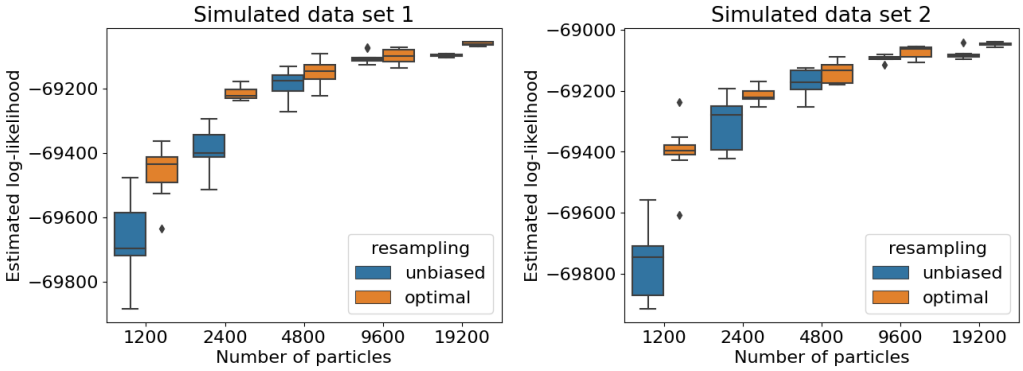
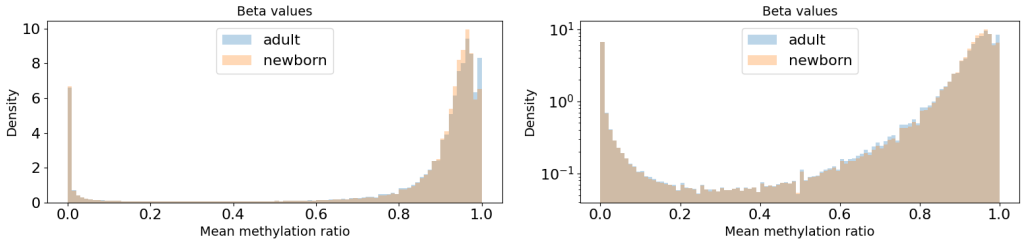


Figure 2.3: Log-likelihood estimates from the particle filter with different resampling scheme on two simulated data sets.

91 reads, while the newborn samples have 23.957 million sites and a median coverage of 101 reads. 23.264 million sites have reads for both the adult and newborn samples. We use the GRCh38 reference genome to get the positions of 27.853 million CpG sites so that missing values in at least one group need to be imputed for 16.5% of all CpG sites. The empirical distribution of the beta-values $\{\bar{y}_{t,s}/\bar{n}_{t,s}\}$ and $\{\tilde{y}_{t,s}/\tilde{n}_{t,s}\}$ for all sites t and samples $s = 1$ where reads are available, that is if $n_{t,s} > 0$, are shown in Figure 2.4.

Table 2.3: Area under the curve for different resampling schemes on simulated data.

Algorithm	unbiased				optimal			
	N=2400	N=4800	N=9600	N=19200	N=2400	N=4800	N=9600	N=19200
AUC	0.894	0.937	0.961	0.973	0.916	0.947	0.966	**0.974**



(a) Distribution of beta-values for the adult and newborn samples.

(b) Distribution of beta-values for the adult and newborn samples on a log-scale.

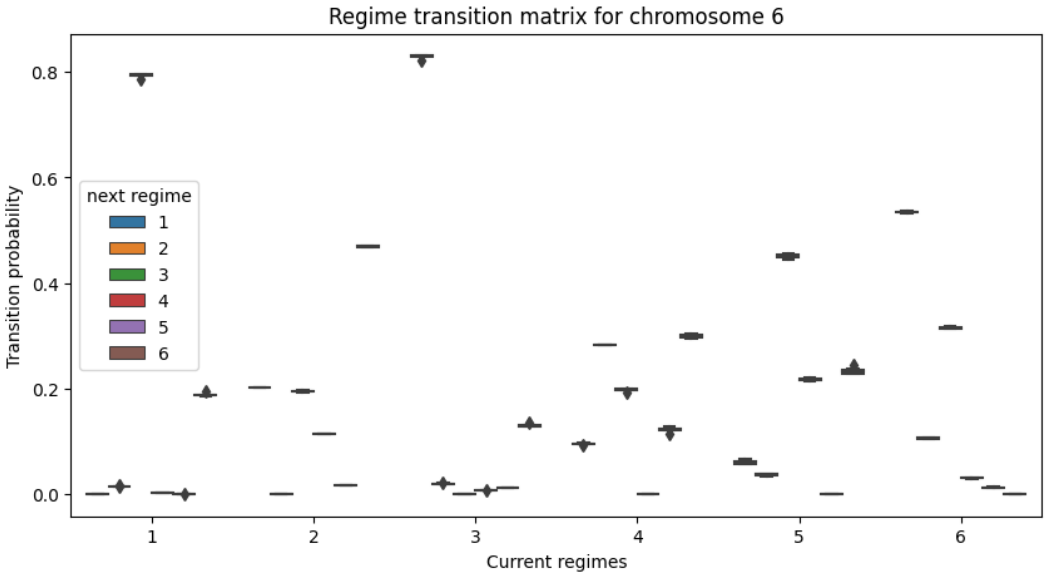
Figure 2.4: Distribution of beta-values for the aging data set.

2.5.1 Estimating model parameters

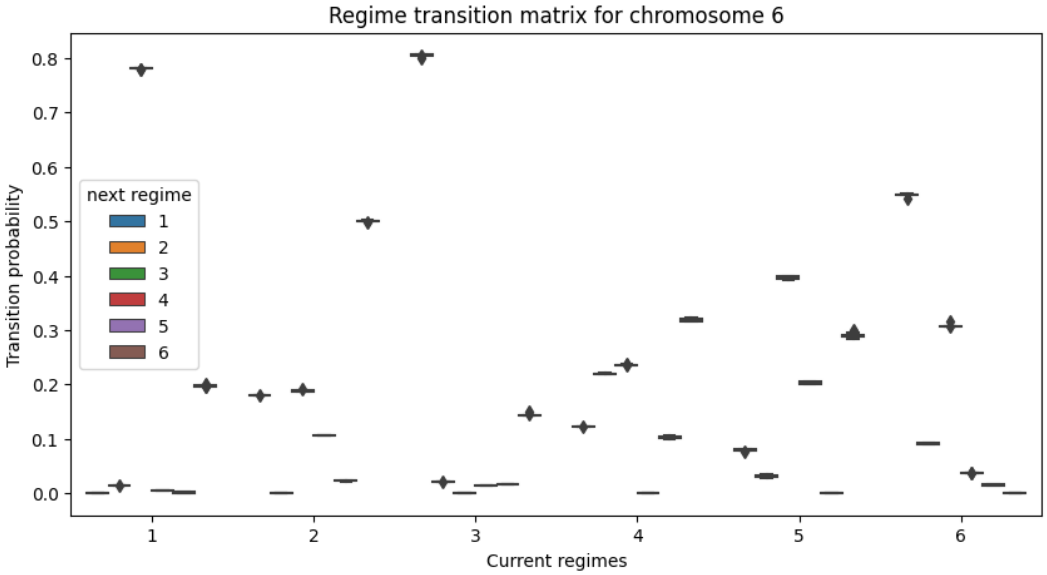
We perform online maximum likelihood estimation of the static parameters of the single-group model for both the adult and newborn samples, separately for each chromosome. The Beta distribution for each regime are chosen according to Figure 2.1. We use $N = 100$ particles as well as the optimal resampling scheme. In order to reduce the variability due to random initialisation of the parameters, it was beneficial to run the maximum likelihood estimation twice over each chromosome using a learning rate of 0.01 for ADAM with parameters updated every 200 steps that decays geometrically with exponent 0.1. To illustrate the robustness of the estimated parameters, we repeat the estimation over 10 replications. Figure 2.5a shows estimates for the regime transition matrix in chromosome 6 for the adult sample using $N = 100$ particles with the estimated duration parameters shown in Figure 2.6a. There is only a small variability in the parameter estimates for different replications. The corresponding estimates in the single-group model applied to the newborn samples are shown in Figures 2.5b and 2.6b, which appear similar and thus seem to support the modeling assumption of both groups evolving jointly with the same parameters as long as they are not in different methylation regimes. The distribution of each estimated parameter across different chromosomes for a single replication is shown in Figures 2.7-2.8.

2.5.2 Differentially methylated positions and regions

We apply the suggested algorithms for the ELOV2 gene, which has been found in previous work (Garagnani et al., 2012; Florath et al., 2014; Park et al., 2016; Goel et al., 2017) to contain differentially methylated CpG sites. The hyper-parameters



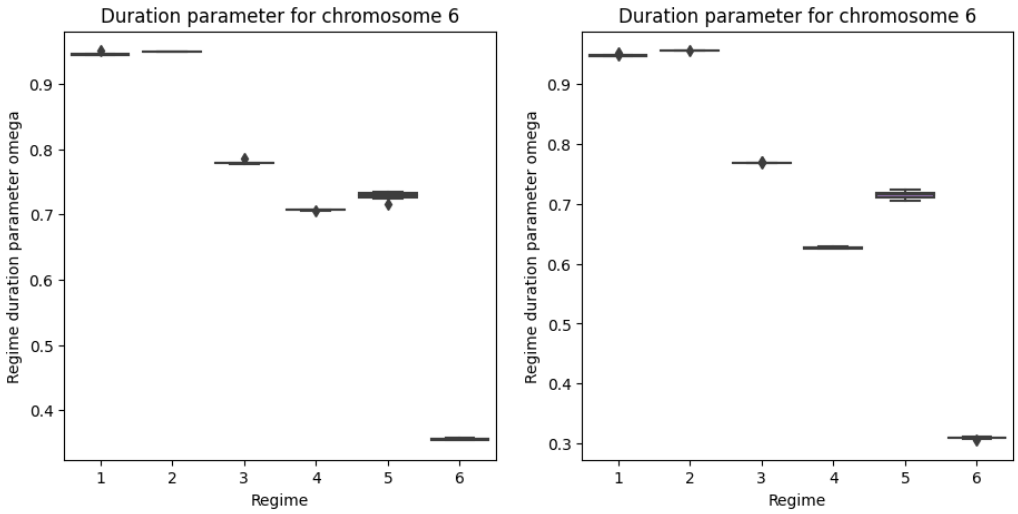
(a) Regime transition parameters for the adult samples in chromosome 6.



(b) Regime transition parameters for the newborn samples in chromosome 6.

Figure 2.5: Boxplot of the regime transition parameters of the single-group model for chromosome 6 for the aging data set using 10 replications.

for the two-group model are set to $\bar{u}_r = \tilde{u}_r = 5$, $\tilde{\omega}_r = 0.8$, $q_{\text{MERGE}} = 0.1$ and $q_{\text{SPLIT}} \in \{0.002, 0.01, 0.05\}$. The optimal filtering algorithm is run with $M = 200$ and $K = 25$ trajectories have been sampled using backward sampling. This procedure was repeated 100 times and yields estimates of the split and regime probabilities at each CpG site as shown in Figures 2.9, 2.10 and 2.11 for a small genomic region



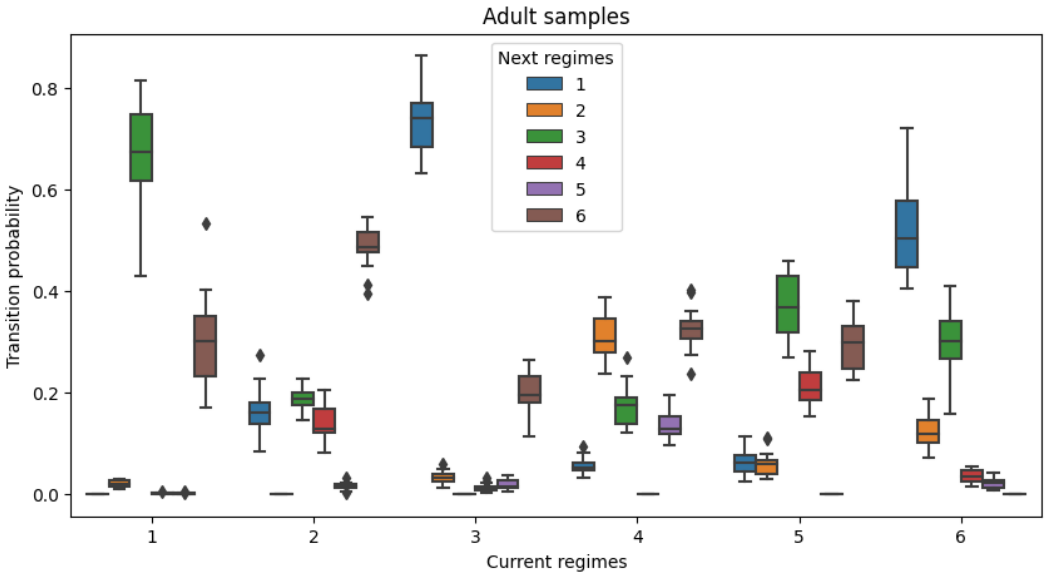
(a) Regime duration parameter for the adult samples.

(b) Regime duration parameter for the case samples.

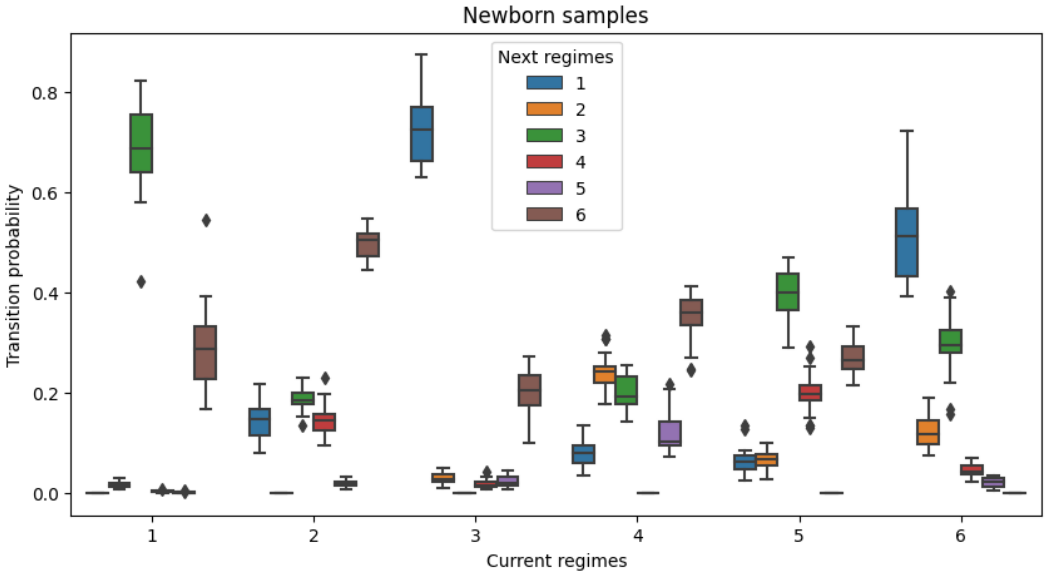
Figure 2.6: Boxplot of the regime duration parameters ω_r of the single-group model for chromosome 6 of the aging data set.

for the different a-priori split probabilities. We observe that for all three choices of q_{SPLIT}, the algorithm detects a high probability of the groups being split around the same region, although the choice of q_{SPLIT} impacts the local index of significance and consequently the results from testing for differential methylation.

The proposed algorithm for detecting DMPs scales linear in the number of CpG sites and the number of particles. One possibility can be to run the filtering and backward sampling algorithm for the entire chromosome, but the computation time can then be in days. An alternative that can be favourable in a High-Performance Computing environment is to split the chromosomes in overlapping regions and apply the filter and smoothing algorithm independently on each region. This can yield an embarrassingly parallel approach and we expect that any bias due to such a subsampling strategy can be made negligible by including an additional buffer of CpG sites at the start and end of each region. We have chosen here the latter approach using $M = 200$ ancestors and $K = 25$ trajectories on regions of 10000 CpG sites and a buffer of 500 at the start and end of the region. The algorithms were repeated 20 times to arrive at the Monte-Carlo estimates $\hat{p}_{t,c}$ of the probability of site t in chromosome c being in the same regime for both the adult and the newborn



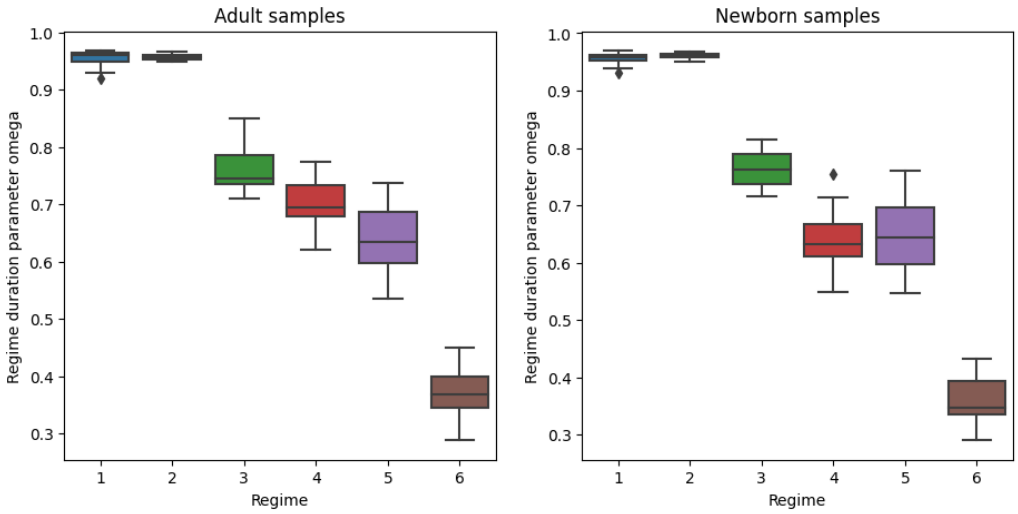
(a) Regime transition parameters for the adult samples across different chromosomes.



(b) Regime transition parameters for the newborn samples across different chromosomes.

Figure 2.7: Boxplot of the regime transition parameters of the single-group model across different chromosomes.

sample. Depending on the choice of the false discovery threshold, the number of detected differentially methylated positions using (2.3.6) are shown in Table 2.4. Our analysis differs from Libertini et al. (2016a). First, their approach to segment blocks of comethylation (COMETs) does not recover DMPs. Second, their analysis using MethylSeekR (Burger et al., 2013) allows for less flexible regimes (*i.e.* unmethylated,



(a) Regime duration parameter for the adult samples.

(b) Regime duration parameter for the newborn samples.

Figure 2.8: Boxplot of the regime duration parameters ω_r of the single-group model across different chromosomes of the aging data set.

low-methylated, partially-methylated) that tend to stretch over very long regions. However, it is still ongoing work whether these differences are biologically meaningful.

Table 2.4: Number of differentially methylated positions for the aging data set.

False discovery threshold	0.1%	0.5%	1%
Number of DMPs	13728	16882	18456

2.6 Discussion

The proposed change point model for DNA methylation as introduced in this article can be adapted in different ways. For instance, the regime transition does not depend on the CpG-density as the distance is measured only in CpG sites. The minimum duration requirement between change points can be easily modified to also include non-CpG positions. Furthermore, different transition kernels can be assumed in CpG-islands and in regions of low CpG-density, respectively.

The introduced two-group model can be applied for general case control studies. It is however possible to adapt it specifically for twin-pairs data by assuming that the probability parameters $\tilde{\pi}_{t,s}$ and $\bar{\pi}_{t,s}$ coincide whenever $\tilde{r}_t = \bar{r}_t$ in order to reduce

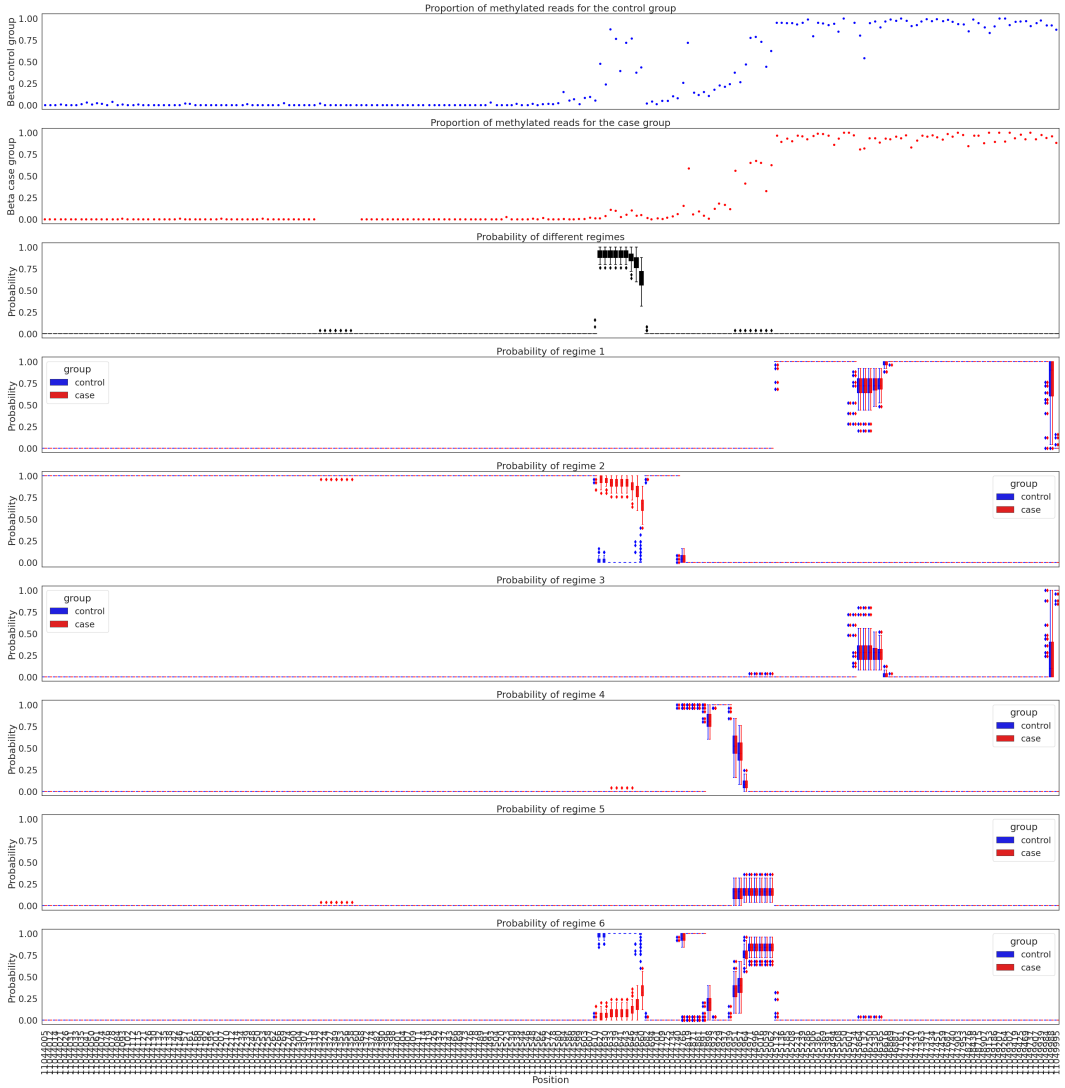


Figure 2.9: Results from the filter and backward sampling algorithm for a region in chromosome 6 with $q_{\text{SPLIT}} = 1\%$. The first two plots show the proportion of methylated reads relative to total reads for each sample s with $\bar{n}_{t,s} > 0$ or $\tilde{n}_{t,s} > 0$, that is $\bar{y}_{t,s}/\bar{n}_{t,s}$ or $\tilde{y}_{t,s}/\tilde{n}_{t,s}$. The third plot shows the estimated posterior probability of the two groups being split, *i.e.* $p(z_t = 0|y_{1:T}, \theta)$. The remaining plots show estimates of the posterior regime probabilities for each group, that is $p(\bar{r}_t = r|y_{1:T}, \theta)$ for the control group and $p(\tilde{r}_t = r|y_{1:T}, \theta)$ for the case group with $r \in [6]$.

the observation variance within each twin pair. The same algorithm can be applied for such a choice using a potential function g_θ.

The suggested standard Bayesian approach is most efficient if the model coincides with the true data generating mechanism. However, it might not be robust under

model misspecification, for instance in the presence of outliers. Alternative belief updates have been suggested using for instance a Tsallis-score (Jewson et al., 2018; Knoblauch et al., 2018; Boustati et al., 2020) that downweights observations with a small likelihood which decreases the influence of observations in the tails. The same algorithms can also be used in such a generalised Bayesian setting (Bissiri et al., 2016), but with an adjusted potential function that can be evaluated exactly with not too much additional computational costs if the number of read depths are not too high.

2.7 Appendix

2.7.1 Sequential Monte Carlo methods a.k.a. particle filters

In this section, we describe *marginal* PF which typically require $\mathcal{O}(N^2)$ operations except in the special case of simple change-point models whose structure permits a $\mathcal{O}(N)$ implementation. However, some of the notation (e.g. related to the target distribution) and concepts (e.g. the resampling schemes) introduced in this section are also used by standard PF.

2.7.1.1 Generic target distribution (i.e. the model)

To simplify the presentation, we will drop θ from the notation and use the convention that any variable with subscript 0 is to be ignored from the notation. We assume that the latent variable at time t takes values in some space $\mathcal{X}_t$. For some $T \in \mathbb{N}$ and $t \in [T]$, assume that we are interested in approximating target distributions of the form

$$\pi_t(x_{1:t}) \coloneqq \frac{1}{\mathcal{Z}_t} \prod_{s=1}^{t} \gamma_s(x_{s-1:s}),$$

with (typically intractable) normalising constants

$$\mathcal{Z}_t \coloneqq \int \left[\prod_{s=1}^{t} \gamma_s(x_{s-1:s}) \right] \mathrm{d}x_{1:s}.$$

We refer to $\pi_t(x_{1:t})$ as the *filter* at time t. We also sometimes refer to $\tilde{\pi}_t(x_t) = \int_{\prod_{s=1}^{t-1} \mathcal{X}_s} \pi_t(x_{1:t}) \, \mathrm{d}x_{1:t-1}$ as the *marginal filter* at time t.

Example 1. *In the context of state space models with transitions* $p(x_t|x_{1:t-1}) = p(x_t|x_{t-1}) =: f_t(x_t|x_{t-1})$ *and observation densities* $p(y_t|x_{1:t}, y_{1:t-1}) = p(y_t|x_t) =:$

$g_t(x_t)$, we have

$$\gamma_t(x_{t-1:t}) \coloneqq f_t(x_t|x_{t-1})g_t(x_t).$$

In this case, the filter at time t is $\pi_t(x_{1:t}) = p(x_{1:t}|y_{1:t})$ and the marginal filter at time t is $\tilde{\pi}_t(x_t) = p(x_t|y_{1:t})$.

2.7.1.2 Resampling schemes

Resampling can be used both by marginal and standard PF. Therefore, before turning to these algorithms, we now briefly describe two kinds of resampling schemes. Assume that we are given a weighted sample with N components: $(x^n, w^n)_{n\in[N]}$, which we wish to resample to yield another weighted sample with M components: $(y^m, u^m)_{m\in[M]}$. Throughout, the weights w^n and u^m are not necessarily normalised (i.e. they may not sum to 1) and we introduce the notation $W^n \coloneqq w^n / \sum_{l\in[N]} w^l$. Resampling draws indices $a^{1:M}$ from some distribution $\rho(a^{1:M})$ on $[N]^M$ and sets

$$y^m \coloneqq x^{a^m}, \quad \text{and} \quad u^m \coloneqq \frac{w^m}{\sum_{l=1}^M \rho^l(m)},$$

where ρ^m denotes the marginal distribution of the mth component under the joint law ρ.

- **Option I: Standard Resampling Schemes.** We recall that the resampling scheme associated with the joint law $\rho(a^{1:M})$ was termed *unbiased*[3] in Andrieu et al. (2010) if, for $a \in [N]$,

  $$\sum_{m=1}^M \rho^m(a) = MW^a,$$

  where ρ^m denotes the marginal distribution of the mth component under the joint law ρ. Commonly used resampling schemes which are unbiased are *multinomial, stratified,* and *systematic* resampling. For such schemes,

  $$u^m = \frac{1}{M}\sum_{n=1}^N w^n.$$

[3]We stress that not having an unbiased resampling scheme does *not* induce bias, e.g. in the estimate of the normalising constant, as long as the post-resampling weights are suitably modified to take the resampling step into account. That is, if we use resampling schemes that are not unbiased in this sense, then the particles are not evenly weighted after resampling. To avoid confusion, we call these *standard* resampling schemes here.

- **Option II: Optimal Finite-State Resampling Scheme.** Standard resampling schemes are not optimal for discrete state-space applications because variance reductions can be achieved by taking greater care to avoid duplicate particles post-resampling (at the cost of not having evenly-weighted particles post resampling) in such scenarios. Instead, we may use the optimal finite-state resampling scheme proposed by Fearnhead (1998); Fearnhead and Clifford (2003).

  Without loss of generality, also assume that the particles and weights are ordered such that $W^1 \geqslant \ldots \geqslant W^N$. The optimal finite-state resampling scheme proceeds as follows. First, we use Fearnhead (1998, Algorithm 5.2) to solve

$$\sum_{n=1}^{N} [1 \wedge CW^n] = M,$$

  for $C > 0$. Write $K \coloneqq \max\{n \in [N] \mid CW^n \geqslant 1\}$, $L \coloneqq M - K$ and $I \coloneqq N - K$. The joint distribution of the indices under the resampling scheme is then given by

$$\rho(a^{1:M}) \coloneqq \left[\prod_{n=1}^{K} \delta_n(a_{t-1}^n)\right] \sum_{b^{1:L} \in [I]^L} \varrho(b^{1:L}) \prod_{l=1}^{L} \delta_{K+b^l}(^{K+l}).$$

  Here, $\varrho_{t-1}(b^{1:L})$ denotes some standard resampling scheme (see Option I) which draws L ancestor indices, each of which takes values in $[I]$, based on the (self-normalised) weights $V^{1:I}$,

$$V^i \coloneqq \frac{W^{K+i}}{\sum_{j=1}^{I} W^{K+j}} = \frac{CW^{K+i}}{L},$$

  for $i \in [I]$. Here, the identity on the r.h.s. follows from the definition of C and L. Hence, by definition of standard resampling schemes (i.e. the unbiasedness of the resampling associated with the law $\varrho(a^{1:L})$):

$$\begin{aligned}\sum_{m=1}^{M} \rho^m(a) &= \begin{cases} 1, & \text{if } a \in [K], \\ LV^{a-K}, & \text{if } a \in [N] \setminus [K], \end{cases} \\ &= 1 \wedge CW^a.\end{aligned}$$

  In summary, we thus obtain the post-resampling weights

$$u^m = \frac{w^m}{1 \wedge CW^m}.$$

2.7.2 Generic Marginal Particle Filter

In this section, we describe a generic 'marginal' PF. Such particle filters were introduced in Klass et al. (2005); Lin et al. (2005) but, as shown in Finke et al. (2016), they are based on ideas which go back at least as far as Neal (2003); Neal et al. (2004). In contrast to conventional PF, these algorithms can be justified as working on an extended space which does *not* include ancestor indices a_{t-1}^n, i.e. these algorithms do not maintain a collection of particle lineages at any time steps (although particle lineages can be constructed via backward-sampling type recursions). Marginal PF are used much less frequently than standard PF because the former usually have a computational complexity which is quadratic in the number of particles whereas it is only linear for the latter.

2.7.2.1 Extended Proposal Distribution

We hereafter write $\mathbf{x}_t \coloneqq x_t^{1:N_t}$, where N_t is the total number of particles used at time t. The following is the law over all random variables generated by the PF (plus a collection of particle indices $k_{1:T} \in \prod_{t=1}^T [N_t]$ which index a single particle path):

$$\bar{q}_T(\mathbf{x}_{1:T}, k_{1:T}) \coloneqq \left[\prod_{t=1}^T q_t(\mathbf{x}_t|\mathbf{x}_{1:t-1})\right] \frac{\prod_{t=1}^T v_t^{k_{t-1},k_t}(\mathbf{x}_{1:t})}{\widehat{\mathcal{Z}}_T(\mathbf{x}_{1:T})},$$

where

$$\widehat{\mathcal{Z}}_T(\mathbf{x}_{1:T}) \coloneqq \sum_{k_T \in [N_T]} w_T^{k_T}(\mathbf{x}_{1:T}) = \sum_{k_1 \in [N_1]} \cdots \sum_{k_T \in [N_T]} \prod_{t=1}^T v_t^{k_{t-1},k_t}(\mathbf{x}_{t-1:t}),$$

and where for any $t \in [T]$ and any $(k_{t-1}, k_t) \in [N_{t-1}] \times [N_t]$:

$$\begin{aligned}
w_t^{k_t}(\mathbf{x}_{1:t}) &\coloneqq \sum_{k_{t-1} \in [N_{t-1}]} w_{t-1}^{k_{t-1}}(\mathbf{x}_{1:t-1}) v_t^{k_{t-1},k_t}(\mathbf{x}_{1:t}), \\
v_t^{k_{t-1},k_t}(\mathbf{x}_{1:t}) &\coloneqq \frac{\gamma_t(x_{t-1:t}^{k_{t-1:t}}) \lambda_t(k_t|x_t^{k_t}, \mathbf{x}_{1:t-1})}{q_t^{k_t}(x_t^{k_t}|\mathbf{x}_{1:t-1})}, \\
\lambda_t(k_t|x_t, \mathbf{x}_{1:t-1}) &\coloneqq \frac{q_t^{k_t}(x_t|\mathbf{x}_{1:t-1})}{\sum_{n=1}^{N_t} q_t^n(x_t|\mathbf{x}_{1:t-1})}, \\
q_t(\mathbf{x}_t|\mathbf{x}_{1:t-1}) &\coloneqq q_t^{k_t}(x_t^{k_t}|\mathbf{x}_{1:t-1}) q_t^{-k_t}(\mathbf{x}_t^{-k_t}|x_t^{k_t}, \mathbf{x}_{1:t-1}).
\end{aligned}$$

Here, $q_t^n(x_t|\mathbf{x}_{1:t-1})$ denotes the nth marginal under the joint proposal for all the N_t particles generated at time t, $q_t(\mathbf{x}_t|\mathbf{x}_{1:t-1})$; and $q_t^{-n}(\mathbf{x}_t^{-n}|x_t, \mathbf{x}_{1:t-1})$ denotes the associated full conditional distribution for all the remaining particles $\mathbf{x}_t^{-n} \coloneqq (x_t^1, \ldots, x_t^{n-1}, x_t^{n+1}, \ldots, x_t^{N_t})$. Furthermore, $\lambda_t(k_t|x_t, \mathbf{x}_{1:t-1})$ is a distribution

over k_t which will be used in the extended target distribution below. Such laws were first introduced in the as yet unpublished work Lee et al. nd in order to justify the use of non-exchangeable resampling schemes in standard PF. Here, this distribution allows us to use a proposal for the N_t particles drawn at time t which is non-exchangeable in the sense that $q_t^n(x_t|\mathbf{x}_{1:t-1}) \neq q_t^m(x_t|\mathbf{x}_{1:t-1})$, for some $m \neq n$, without jeopardising unbiasedness.

2.7.2.2 Extended Target Distribution

The extended target distribution is given by $\bar{\pi}_T(\mathbf{x}_{1:T}, k_{1:T}) \coloneqq \bar{\gamma}_T(\mathbf{x}_{1:T}, k_{1:T})/\mathcal{Z}_T$ with

$$\bar{\gamma}_T(\mathbf{x}_{1:T}, k_{1:T}) \coloneqq \prod_{t=1}^{T} \gamma_t(x_{t-1:t}^{k_{t-1:t}}) \lambda_t(k_t|x_t^{k_t}, \mathbf{x}_{1:t-1}) q_t^{-k_t}(\mathbf{x}_t^{-k_t}|x_t^{k_t}, \mathbf{x}_{1:t-1}).$$

It can be seen that $\bar{\pi}_T(\mathbf{x}_{1:T}, k_{1:T})$ admits $\pi_T(x_{1:T})$ as a marginal. That is, if $(\mathbf{x}_{1:T}, k_{1:T}) \sim \bar{\pi}_T$ then $x_{1:T}^{k_{1:T}} \sim \pi_T$.

2.7.2.3 Radon-Nikodym Derivative

If the proposal kernels $q_t(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ are chosen such that $\bar{\gamma}_T \ll \bar{q}_T$, the following Radon-Nikodym derivative is well defined and constitutes the usual (and by construction unbiased) marginal-particle filter estimate of the normalising constant of the target distribution:

$$\frac{\bar{\gamma}_T(\mathbf{x}_{1:T}, k_{1:T})}{\bar{q}_T(\mathbf{x}_{1:T}, k_{1:T})} = \widehat{\mathcal{Z}}_T(\mathbf{x}_{1:T}).$$

2.7.3 Special case of an $\mathcal{O}(N)$ (standard) particle filter for the case/control-group scenario

2.7.3.1 Model-specific quantities

Recall that in the change-point model for the case/control scenario, the state space is given by $\mathcal{X} \coloneqq \{0, 1\} \times (\mathbb{N} \times [R])^2$, i.e. each state takes the form $x_t = (z_t, \bar{d}_t, \bar{r}_t, \tilde{d}_t, \tilde{r}_t) \in \mathcal{X}$, where $z_t = 1$ indicates that the two groups are merged whereas $z_t = 0$ indicates that they are split. At time 1, the model requires that $\bar{d}_t = \tilde{d}_t = 1$. This implies the following.

- At time 1, the state x_1 can only take one out of (at most) R^2 distinct values with positive probability under the model. These values are labelled $\tilde{\chi}_1, \ldots, \tilde{\chi}_{R^2}$, where

$$\tilde{\chi}_i \coloneqq (\mathbf{1}\{r = s\}, 1, r, 1, s), \quad \text{for } i = (r-1)R + s \text{ and } r, s \in [R].$$

- At time $t > 1$, conditional on the ancestor particle x_{t-1}, the particle x_t can only take one out of (at most) $I \coloneqq 2R + R^2$ distinct values with positive probability under the model. These values are labelled $\chi_1(x_{t-1}), \ldots, \chi_{2R+R^2}(x_{t-1})$, where

$$\chi_i(x_{t-1}) \coloneqq \begin{cases} (z_{t-1}, \bar{d}_{t-1} + 1, \bar{r}_{t-1}, \tilde{d}_{t-1} + 1, \tilde{r}_{t-1}), & \text{for } i = 1, \\ (1, \bar{d}_{t-1} + 1, \bar{r}_{t-1}, \bar{d}_{t-1} + 1, \bar{r}_{t-1}), & \text{for } i = 2, \\ (0, 1, i-1, \tilde{d}_{t-1} + 1, \tilde{r}_{t-1}), & \text{for } 2 < i \leqslant \tilde{r}_{t-1} + 1, \\ (0, 1, i, \tilde{d}_{t-1} + 1, \tilde{r}_{t-1}), & \text{for } \tilde{r}_{t-1} + 1 < i \leqslant R + 1, \\ (0, \bar{d}_{t-1} + 1, \bar{r}_{t-1}, 1, i - R), & \text{for } R + 1 < i \leqslant R + \bar{r}_{t-1}, \\ (0, \bar{d}_{t-1} + 1, \bar{r}_{t-1}, 1, i - R + 1), & \text{for } R + \bar{r}_{t-1} < i \leqslant 2R, \\ \tilde{\chi}_{i-2R}, & \text{for } 2R < i \leqslant I. \end{cases}$$

In summary, any state sequence up to time t, $x_{1:t}$, can only take one out of (at most) $R^2 I^{t-1}$ distinct values with positive probability under the model. Hence, in principle, filtering and marginal-likelihood computation could be performed analytically in this model by averaging over all $\mathcal{O}(R^2 I^{t-1})$ possible state sequences that have positive probability under the model. Unfortunately, the cost of such exact calculations grows quadratically in t and so is only feasible if t is very small.

2.7.3.2 Algorithm-specific quantities

We set an upper bound for the number of particles at time t to be

$$N_t \coloneqq \begin{cases} R^2, & \text{if } t = 1, \\ (M \wedge N_{t-1})I, & \text{for } t > 1, \end{cases}$$

where we recall that $I \coloneqq 2R + R^2$ and where $M \in \mathbb{N}$ which denotes the maximum number of 'resampled' particles and is a tuning parameter which allows us to control the computational cost: We have at most $N_t \leqslant MI$ particles which means that the algorithm can be implemented in $\mathcal{O}(MI)$ operations per time step (or $\mathcal{O}(MI \log(MI))$ operations if we use optimal finite-state resampling which requires additional sorting of the weights).

It is possible that some of the proposed particles have a weight of zero. Let N'_{t-1} be the number of particles with non-zero weights at step t. The motivation for this

specification is that, depending on the value of M, we can perform filtering exactly over the first few time steps before we need to start pruning the particle system from N_{t-1} particle lineages down to M particle lineages.

More precisely, if $N'_{t-1} \leqslant M$, we simply extend each of the N'_{t-1} existing particle lineages in all I possible directions. If $N'_{t-1} > M$, we need to prune the number of particle lineages from N'_{t-1} down to M before again extending each existing particle lineages in all I possible directions. If $t \geqslant 1$ is such that $N'_{t-1} \leqslant M$, then the normalising-constant 'estimate' produced by the algorithm below, $\widehat{\mathcal{Z}}_t \coloneqq \sum_{n=1}^{N'_t} w_t^n$, constitutes an exact evaluation of the normalising constant and the same is true for the 'approximations' of expectations under the time-t filter. Approximations are only induced once $N'_{t-1} > M$ because then the total number of possible distinct state sequences up to time t is so large that can no longer keep track of all of them and need to start pruning the number of particle trajectories from N'_{t-1} down to M.

Algorithm 5 (standard particle filter for case/control scenario)**.**

1. *At time* $t = 1$*, set* $N_1 \coloneqq R^2$ *and*

   (a) *for* $n \in [N_1]$*, set* $x_1^n \coloneqq \tilde{\chi}_n$*,*

   (b) *for* $n \in [N_1]$*, set* $w_1^n \coloneqq \gamma_1(x_1^n)$*.*

2. *At time* t*,* $t > 1$*,*

   - *set* $N'_{t-1} \coloneqq \#\{n \in [N_{t-1}] \mid w_{t-1}^n > 0\}$*,*
   - *set* $N_t \coloneqq (N'_{t-1} \wedge M)I$*,*
   - *if* $N'_{t-1} \leqslant M$*,*

     (a) *set* $b^{1:N'_{t-1}} \coloneqq (\rho_{t-1}(1), \ldots, \rho_{t-1}(N'_{t-1}))$*, where* $\rho_{t-1} \colon [N'_{t-1}] \to [N_{t-1}]$ *maps each index of the particles after removing zero weights to the index of the original particle.*

     (b) *for* $n \in [N_t]$*, set*

$$\begin{aligned} a_{t-1}^n &\coloneqq b^{\lceil n/I \rceil}, \\ x_t^n &\coloneqq \chi_{((n-1) \bmod I)+1}(x_{t-1}^{a_{t-1}^n}), \\ w_t^n &\coloneqq w_{t-1}^{a_{t-1}^n} \gamma_t(x_{t-1}^{a_{t-1}^n}, x_t^n); \end{aligned}$$

   - *else, if* $N'_{t-1} > M$*,*

     (a) *sample* $b^{1:M} \in [N'_{t-1}]^M$ *via a standard resampling scheme (**Option I**) or optimal finite-state resampling (**Option II**) according to the weights* $W_{t-1}^{1:N'_{t-1}}$*, where*

$$W_{t-1}^n \coloneqq \frac{w_{t-1}^n}{\sum_{l=1}^{N'_{t-1}} w_{t-1}^l},$$

     (b) *for* $n \in [N_t]$*, set*

$$\begin{aligned} a_{t-1}^n &\coloneqq b^{\lceil n/I \rceil}, \\ x_t^n &\coloneqq \chi_{((n-1) \bmod I)+1}(x_{t-1}^{a_{t-1}^n}), \\ w_t^n &\coloneqq \begin{cases} \left[\dfrac{1}{M} \displaystyle\sum_{k=1}^{N'_{t-1}} w_{t-1}^k\right] \gamma_t(x_{t-1}^{a_{t-1}^n}, x_t^n), & \text{for } \textbf{\textit{Option I}}, \\ \dfrac{w_{t-1}^{a_{t-1}^n} \gamma_t(x_{t-1}^{a_{t-1}^n}, x_t^n)}{1 \wedge C_{t-1} W_{t-1}^{a_{t-1}^n}}, & \text{for } \textbf{\textit{Option II}}. \end{cases} \end{aligned}$$

2.7.4 Special case of a marginal particle filter: Simple Change-Point model (e.g. Single-Group Scenario)

In this section, we derive the PF for simple change-point models (Caron et al., 2012; Yildirim et al., 2013) as a special case of the generic marginal PF stated above. This is useful because it provides a simple way of deriving the weight updates for this particular PF. It also immediately proves that the PF for change-point models yields unbiased estimates of the marginal likelihood.

The change-point model in this section is assumed to belong to the class described in Section 2.2. We thus assume that there is only one sequence of change points. In this model, the marginal filters are supported on a finite subset of $\mathcal{X}$ so that the marginal filters and the marginal likelihood up to time T can be calculated analytically via standard recursions for finite-state hidden Markov models. However, since the support of the marginal filters grows linearly with T, the implementation of these exact recursions requires $\mathcal{O}(T^2)$ operations which can quickly become prohibitive. In contrast, the PF for change-point models is able to approximate the filters and marginal likelihood in $\mathcal{O}(T)$ operations.

Recall that we mentioned above that marginal PF typically have a computational complexity which is quadratic in the number of particles. We stress that due to the particular structure of the state space in the context of simple change-point models, the algorithm described here has a computational complexity which is only linear in the number of particles.

2.7.4.1 Change-Point Model-Specific Quantities

Hereafter, we use a constant number of particles, i.e. we set $N_t \coloneqq N$. Write $\mathbf{x}_t \coloneqq x_t^{1:N}$ as well as

$$\chi_1(d, r) \coloneqq (d + 1, r),$$
$$\chi(r) \coloneqq (1, r),$$

for $d \in \mathbb{N}$ and for $r \in [R]$. With this notation, we choose some initial proposal distribution $q_1(\mathbf{x}_1)$ on $\mathcal{X}^N$ and, for $t > 1$, we specify the proposal kernels as

$$q_t(\mathbf{x}_t|\mathbf{x}_{1:t-1}) \coloneqq \left[\sum_{a_{t-1}^{1:M} \in [N]^M} \rho_{t-1}(a_{t-1}^{1:M}|\mathbf{x}_{1:t-1}) \prod_{m=1}^{M} \delta_{\chi_1(x_{t-1}^{a_{t-1}^m})}(x_t^m)\right] \times \prod_{r=1}^{R} \delta_{\chi(r)}(x_t^{M+r}),$$

where $\rho_{t-1}(a_{t-1}^{1:M}|\mathbf{x}_{1:t-1})$ is the joint distribution of the ancestor indices $a_{t-1}^{1:M} \in [N]^M$ which are used for 'resampling' (although, being a 'marginal' PF, the algorithm does not actually subsample particle lineages; instead, we work directly with mixture proposals, i.e. the ancestor indices are integrated out). Further below, we will specify this distribution for the two potential choices of resampling scheme which were termed Options I and II in Algorithm 1. This implies that the marginal proposal kernel for the nth particle is given by

$$q_t^n(x_t|\mathbf{x}_{1:t-1}) = \begin{cases} \sum_{a_{t-1} \in [N]} \rho_{t-1}^n(a_{t-1}|\mathbf{x}_{1:t-1}) \delta_{\chi_1(x_{t-1}^{a_{t-1}})}(x_t), & \text{if } n \in [M], \\ \delta_{\chi(n-M)}(x_t), & \text{if } n \in [N] \setminus [M], \end{cases}$$

where $\rho_{t-1}^m(a_{t-1}|\mathbf{x}_{1:t-1})$ denotes the m-th marginal of $\rho_{t-1}(a_{t-1}^{1:M}|\mathbf{x}_{1:t-1})$. In particular, we thus have $\lambda_1(n|x_1^n) = \text{Unif}_{[N]}(n)$ and, for $t > 1$,

$$\lambda_t(n|x_t, \mathbf{x}_{1:t-1}) = \begin{cases} \frac{\sum_{a_{t-1} \in [N]} \rho_{t-1}^n(a_{t-1}|\mathbf{x}_{1:t-1}) \mathbf{1}\{\chi_1(x_{t-1}^{a_{t-1}}) = x_t\}}{\sum_{l=1}^{M} \sum_{a_{t-1} \in [N]} \rho_{t-1}^l(a_{t-1}|\mathbf{x}_{1:t-1}) \mathbf{1}\{\chi_1(x_{t-1}^{a_{t-1}}) = x_t\}}, & \text{if } n \in [M], \\ \delta_{\chi^{-1}(x_t)+M}(n), & \text{if } n \in [N] \setminus [M]. \end{cases}$$

Putting all of these definitions together then gives $v_1^n(\mathbf{x}_1) = \gamma_1(x_1^n)/\sum_{m=1}^{N} q_1^m(x_1^n)$ and

$$v_t^{m,n}(\mathbf{x}_{1:t}) = \begin{cases} \frac{\gamma_t(x_{t-1}^m, x_t^n)}{\sum_{l=1}^{M} \sum_{a_{t-1} \in [N]} \rho_{t-1}^l(a_{t-1}|\mathbf{x}_{1:t-1}) \mathbf{1}\{\chi_1(x_{t-1}^{a_{t-1}}) = x_t^n\}}, & \text{if } n \in [M], \\ \gamma_t(x_{t-1}^m, x_t^n), & \text{if } n \in [N] \setminus [M]. \end{cases}$$

2.7.4.2 Resampling Schemes and Associated Weight Updates.

We now derive the importance weights implied the two different families of resampling schemes which were referred to as Options I and II in Algorithm 1.

- **Option I: Standard Resampling Schemes.** For such schemes, we have

$$v_t^{m,n}(\mathbf{x}_{1:t}) = \begin{cases} \dfrac{\gamma_t(x_{t-1}^m, x_t^n)}{M \sum_{a_{t-1}=1}^N W_{t-1}^{a_{t-1}}(\mathbf{x}_{1:t-1})\, \mathbf{1}\{\chi_1(x_{t-1}^{a_{t-1}}) = x_t^n\}}, & \text{if } n \in [M], \\ \gamma_t(x_{t-1}^m, x_t^n), & \text{if } n \in [N] \setminus [M]. \end{cases}$$

Hence, $w_1^n(\mathbf{x}_1) = v_1^n(\mathbf{x}_1)$ and, for $t > 1$,

$$w_t^n(\mathbf{x}_{1:t}) = \begin{cases} \left[\dfrac{1}{M} \displaystyle\sum_{m=1}^N w_{t-1}^m(\mathbf{x}_{1:t-1})\right] \gamma_t(\chi_1^{-1}(x_t^n), x_t^n), & \text{if } n \in [M], \\ \displaystyle\sum_{m=1}^N w_{t-1}^m(\mathbf{x}_{1:t-1}) \gamma_t(x_{t-1}^m, x_t^n), & \text{if } n \in [N] \setminus [M]. \end{cases}$$

Note that within Algorithm 1, $\chi_1^{-1}(x_t^n) = x_{t-1}^{a_{t-1}^n}$.

- **Option II: Optimal Finite-State Resampling Scheme.** For such schemes, we obtain the incremental weight

$$v_t^{m,n}(\mathbf{x}_{1:t}) = \begin{cases} \dfrac{\gamma_t(x_{t-1}^m, x_t^n)}{\sum_{a_{t-1}=1}^N [1 \wedge C_{t-1} W_{t-1}^{a_{t-1}}(\mathbf{x}_{1:t-1})]\, \mathbf{1}\{\chi_1(x_{t-1}^{a_{t-1}}) = x_t^n\}}, & \text{if } n \in [M], \\ \gamma_t(x_{t-1}^m, x_t^n), & \text{if } n \in [N] \setminus [M]. \end{cases}$$

Hence, $w_1^n(\mathbf{x}_1) = v_1^n(\mathbf{x}_1)$ and, for $t > 1$,

$$w_t^n(\mathbf{x}_{1:t}) = \begin{cases} \dfrac{\gamma_t(x_{t-1}^{a(n)}, x_t^n) w_{t-1}^{a(n)}(\mathbf{x}_{1:t})}{1 \wedge C_{t-1} W_{t-1}^{a(n)}(\mathbf{x}_{1:t-1})}, & \text{if } n \in [M], \\ \displaystyle\sum_{m=1}^N w_{t-1}^m(\mathbf{x}_{1:t-1}) \gamma_t(x_{t-1}^m, x_t^n), & \text{if } n \in [N] \setminus [M], \end{cases}$$

where $a(n) \in [N]$ is a particle index which satisfies $\chi_1(x_{t-1}^{a(n)}) = x_t^n$. Within Algorithm 1, this index is again readily available since we can always take $a(n) = a_{t-1}^n$.[4]

2.7.5 Gradient Calculations for the single-group model

Parameters governing the transition matrix. For any $(r, r') \in [R]^2$ and $j \in [R]$, the gradient for the parameters governing the j-th row of the transition matrix P_θ is

[4] Index $a(n)$ is not necessarily unique. Yet, this is inconsequential as for any $(i, j) \in [N]^2$, if $\chi_1(x_{t-1}^i) = \chi_1(x_{t-1}^j) = x_t^n$ then (a) $x_{t-1}^i = x_{t-1}^j$, (b) $w_{t-1}^i(\mathbf{x}_{1:t-1}) = w_{t-1}^j(\mathbf{x}_{1:t-1})$.

given by

$$\nabla_{\theta_{(R-1)(j-1):(R-1)j}} \log P_\theta(r'|r) = \begin{cases} [\iota_{r'}^R - (P_\theta(1|r), \ldots, P_\theta(R|r))^\top]_{-r}, & \text{if } j = r, \\ 0 \in \mathbb{R}^{R-1}, & \text{if } j \neq r, \end{cases}$$

where $\iota_r^R \in \mathbb{R}^R$ denotes a vector filled with 0's except for the r-th element that is equal to 1; furthermore, operation $[\,\cdot\,]_{-r}$ removes the r-th element from a vector, i.e. $[x_{1:R}]_{-r} = (x_{1:(r-1)}, x_{(r+1):R})$

Parameters governing the sojourn time. For the conditional change-point probabilities, we have

$$\nabla_\theta \log(1 - \rho_{\theta,t+1}(x_t)) = -\frac{\rho_{\theta,t+1}(x_t)}{1 - \rho_{\theta,t+1}(x_t)} \nabla_\theta \log \rho_{\theta,t+1}(x_t),$$

where

$$\begin{aligned} \nabla_\theta \log \rho_{\theta,t+1}(x_t) &= \nabla_\theta \log \frac{h_{\theta,r_t}(d_t)}{1 - H_{\theta,r_t}(d_t - 1)} \\ &= \nabla_\theta \log h_{\theta,r_t}(d_t) + \frac{\nabla_\theta H_{\theta,r_t}(d_t - 1)}{1 - H_{\theta,r_t}(d_t - 1)} \\ &= \nabla_\theta \log h_{\theta,r_t}(d_t) + \frac{\sum_{i=1}^{d_t-1} h_{\theta,r_t}(i) \nabla_\theta \log h_{\theta,r_t}(i)}{1 - H_{\theta,r_t}(d_t - 1)}. \end{aligned}$$

Thus, one only needs to calculate gradients of the form $\nabla_\theta \log h_{\theta,r_t}(i)$. For the regime-specific 'success probability' parameters, for any $r' \in [R]$, we have

$$\frac{\partial}{\partial \theta_{R(R-1)+r'}} \log h_{\theta,r}(i) \Big|_{\omega_{r'} = \text{logit}^{-1}(\theta_{R(R-1)+r'})} = \begin{cases} i - u_r - \omega_r(i - u_r + \kappa_r), & r' = r \text{ and } i \geqslant u_r, \\ 0, & \text{otherwise.} \end{cases}$$

2.7.6 Supplementary material for the single-group simulation studies

2.7.7 Supplementary material for the aging data set

Table 2.5: Average estimated posterior probability of the true regimes for 20 replicates with average read depth of 100.

frequency of gradient update	1			200		
learning rate	0.05	0.01	0.002	0.05	0.01	0.002
systematic resampling with N=10	0.764	0.822	0.834	0.842	0.842	0.824
optimal resampling with N=10	0.917	0.924	0.926	0.927	0.926	0.916
systematic resampling with N=100	0.932	0.937	0.938	0.939	0.938	0.930
optimal resampling with N=100	0.939	0.940	0.940	0.940	0.939	0.931

Table 2.6: Average L_1 error of the regime transition matrix for simulated data with average read depth of 10.

frequency of gradient update	1			200		
learning rate	0.05	0.01	0.002	0.05	0.01	0.002
systematic resampling with N=10	0.181	0.109	0.089	0.075	0.069	0.081
optimal resampling with N=10	0.085	0.055	0.051	0.048	0.044	0.051
systematic resampling with N=100	0.062	0.044	0.039	0.038	0.035	0.039
optimal resampling with N=100	0.059	0.042	0.038	0.038	0.035	0.039

Table 2.7: Average L_1 error of the regime duration parameter ω for simulated data with average read depth of 10.

frequency of gradient update	1			200		
learning rate	0.05	0.01	0.002	0.05	0.01	0.002
systematic resampling with N=10	0.123	0.065	0.051	0.047	0.046	0.060
optimal resampling with N=10	0.071	0.033	0.023	0.030	0.024	0.034
systematic resampling with N=100	0.047	0.019	0.008	0.014	0.007	0.009
optimal resampling with N=100	0.042	0.019	0.008	0.014	0.006	0.008

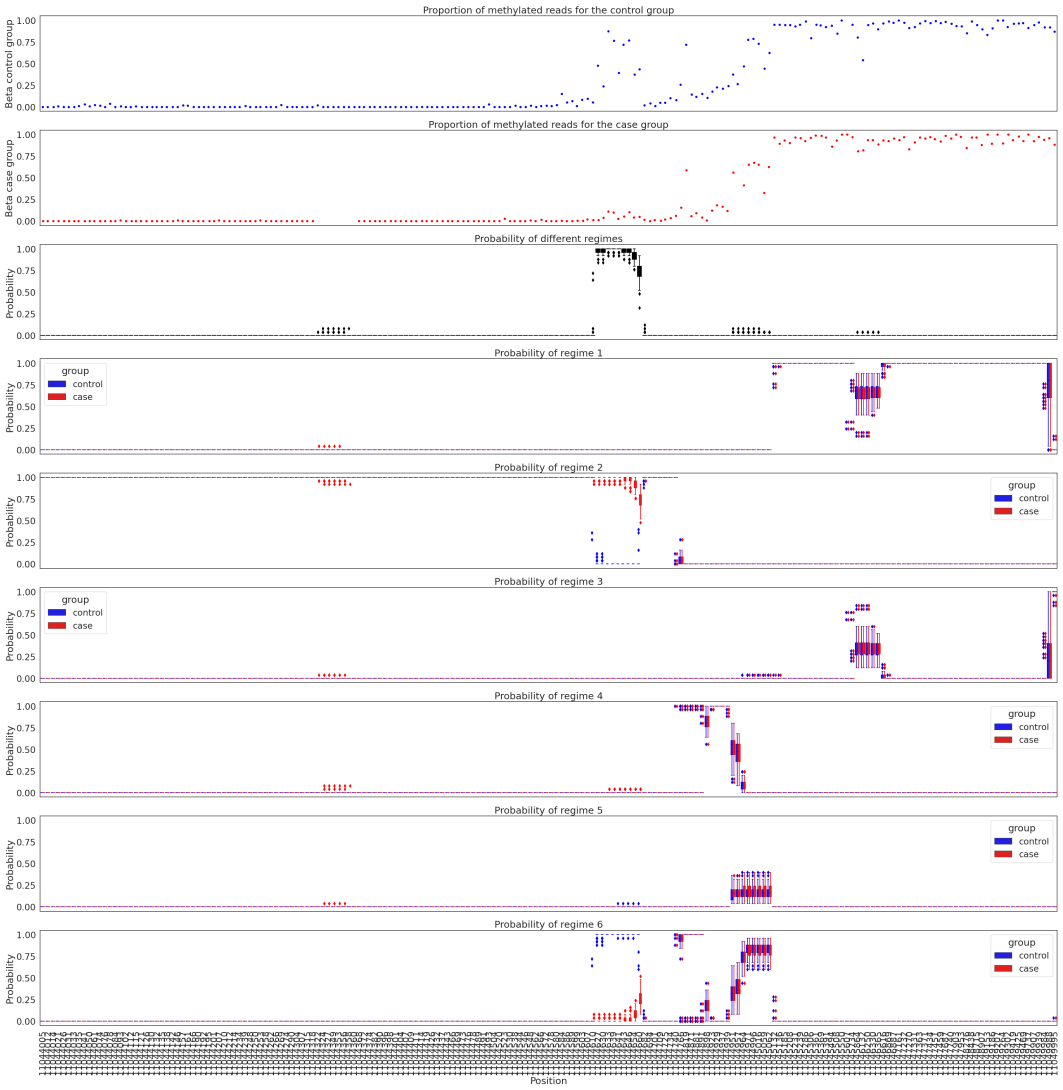


Figure 2.10: Results from the filter and backward sampling algorithm for a region in chromosome 6 with $q_{\text{SPLIT}} = 5\%$. The first two plots show the proportion of methylated reads relative to total reads for each sample s with $\bar{n}_{t,s} > 0$ or $\tilde{n}_{t,s} > 0$, that is $\bar{y}_{t,s}/\bar{n}_{t,s}$ or $\tilde{y}_{t,s}/\tilde{n}_{t,s}$. The third plot shows the estimated posterior probability of the two groups being split, *i.e.* $p(z_t = 0|y_{1:T}, \theta)$. The remaining plots show estimates of the posterior regime probabilities for each group, that is $p(\bar{r}_t = r|y_{1:T}, \theta)$ for the control group and $p(\tilde{r}_t = r|y_{1:T}, \theta)$ for the case group with $r \in [6]$.

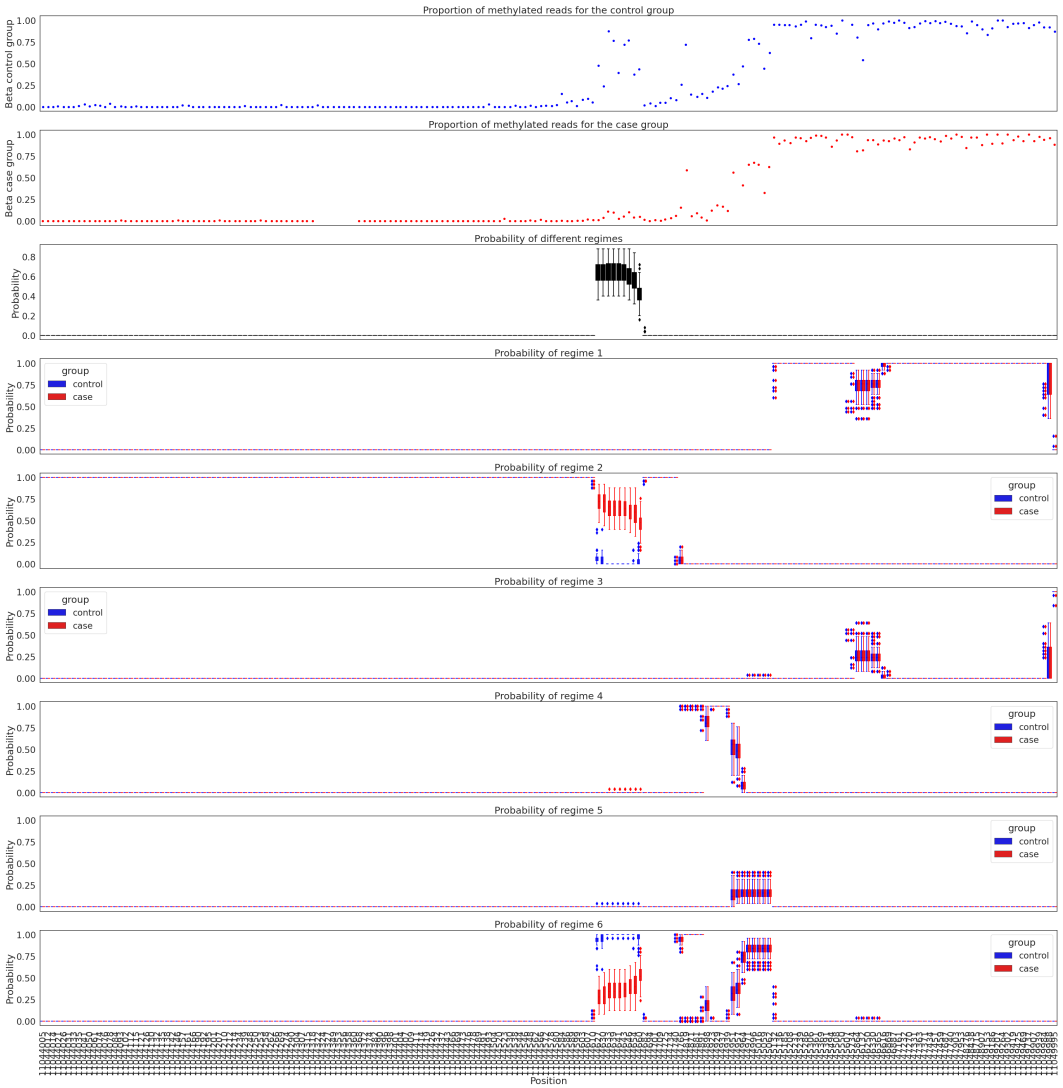


Figure 2.11: Results from the filter and backward sampling algorithm for a region in chromosome 6 with $q_{\text{SPLIT}} = 0.2\%$. The first two plots show the proportion of methylated reads relative to total reads for each sample s with $\bar{n}_{t,s} > 0$ or $\tilde{n}_{t,s} > 0$, that is $\bar{y}_{t,s}/\bar{n}_{t,s}$ or $\tilde{y}_{t,s}/\tilde{n}_{t,s}$. The third plot shows the estimated posterior probability of the two groups being split, *i.e.* $p(z_t = 0|y_{1:T}, \theta)$. The remaining plots show estimates of the posterior regime probabilities for each group, that is $p(\bar{r}_t = r|y_{1:T}, \theta)$ for the control group and $p(\tilde{r}_t = r|y_{1:T}, \theta)$ for the case group with $r \in [6]$.

Chapter 3

Scalable Bayesian Learning for State Space Models using Variational Inference with SMC Samplers

3.1 Introduction

We deal with generic state-space models (SSM) which may be nonlinear and non-Gaussian. Inference for this important and popular family of statistical models presents tremendous challenges that has prohibited their widespread applicability. The key difficulty is that inference on the latent process of the model depends crucially on unknown static parameters that need to be also estimated. While MCMC samplers are unsatisfactory because they both fail to produce high dimensional, efficiently-mixing Markov chains and because they are inappropriate for on-line inference, sequential Monte Carlo (SMC) methods (Kantas et al., 2015) provide the tools to construct successful viable implementation strategies. In particular, particle MCMC (Andrieu et al., 2010) utilises SMC to build generic efficient MCMC algorithms that provide inferences for both static parameters and latent paths. We provide a scalable alternative to these methods via an approximation that combines SMC and variational inference.

We introduce a new variational distribution that unlike recent strand of literature (Maddison et al., 2017; Naesseth et al., 2018; Le et al., 2018) performs variational inference also on the static parameters of the SSM. This is essential for various reasons. First, when there is dependency between static and dynamic parameters posterior inference may be inaccurate if the joint posterior density is approximated

by conditioning on fixed values of static parameters. Second, inferring the static parameter is often the primary problem of interest: for example, for biochemical networks and models involving Lotka Voltera equations, we are not interested in the population of the species per se, but we want to infer some chemical rate constants (such as reaction rates or predation/growth rates), which are parameters of the transition density; in neuroscience, Bayesian decoding of neural spike trains is often made via a state-space representation of point processes in which inference for static parameters is of great importance. Finally, for complex dynamic systems it is often advisable to improve model compression or interpretability by encouraging sparsity and such operations may require inference for the posterior densities of the static parameters.

Our approach differs from Maddison et al. (2017); Naesseth et al. (2018); Le et al. (2018) that do not include the static parameters in a joint variational density. While we introduce a variational density of both the static parameters and the latent path as marginals on an extended space by resorting to sequential Monte Carlo sampling, joint variational densities for the static parameters and one latent path have been considered before for instance in Tan and Nott (2018); Quiroz et al. (2018) using Gaussian variational families that rely on different restrictions for the covariance matrix such as a factor structure or a sparse Cholesky decomposition of the precision matrix.

Sampling from the new variational distribution involves running a SMC algorithm which yields an unbiased estimate of the likelihood for a fixed static parameter value. Importantly, we show that the SMC algorithm constructs a computational graph that allows for optimisation of the variational bound using stochastic gradient descent. We provide some empirical evidence that variational inference on static parameters can give better predictive performance, either out-of sample in the linear Gaussian state space model or in-sample for predictive distributions in a multivariate stochastic volatility model. We also illustrate our method by modelling fairly general intensity functions in a multivariate Hawkes process model.

3.2 Background

Let us begin by introducing the standard inference problem in a generic SSM, followed by a review of the SMC approach to sample from a sequence of distributions arising in

such probabilistic structures. SSMs are characterized by a latent Markov state process $\{X_n\}_{n\geqslant 0}$ on $\mathbb{R}^{d_x}$ and an observable process $\{Y_n\}_{n\geqslant 0}$ on $\mathbb{R}^{d_y}$. We follow the standard convention of using capital letters for random variables and the corresponding lower case letter to denote their values. The dynamics of the latent states is determined, conditional on a static parameter vector $\theta \in \Theta$, by a transition probability density

$$X_n|(\theta, X_{n-1} = x_{n-1}, Y_{n-1} = y_{n-1}) \sim f_\theta(\cdot|x_{n-1}, y_{n-1}),$$

along with an initial density $X_0 \sim f_\theta(\cdot)$. The observations are assumed to be conditionally iid given the states with density given by

$$Y_n|(\theta, X_{0:n} = x_{0:n}, Y_{0:n-1} = y_{0:n-1}) \sim g_\theta(\cdot|x_n),$$

for any $n \geqslant 0$ with the generic notation $x_{0:n} = (x_0, ..., x_n)$.

We consider a Bayesian framework and assume θ has a prior density $p(\theta)$. Consequently, for observed data $y_{0:M}$, we perform inference using the posterior density

$$\pi(\theta, x_{0:M}) \coloneqq p(\theta, x_{0:M}|y_{0:M}) \propto p(\theta)p_\theta(x_{0:M}, y_{0:M}), \tag{3.1}$$

where the joint density of the latent states and observations given a fixed static parameter value θ writes as

$$\begin{aligned} p_\theta(x_{0:M}, y_{0:M}) &= \gamma_\theta(x_{0:M}) \\ \coloneqq & f_\theta(x_0) \prod_{n=1}^{M} f_\theta(x_n|x_{n-1}, y_{n-1}) \prod_{n=0}^{M} g_\theta(y_n|x_n). \end{aligned} \tag{3.2}$$

The posterior density $p(\theta, x_{0:M}|y_{0:M})$ is in general intractable, as is

$$p_\theta(x_{0:M}|y_{0:M}) = \frac{\gamma_\theta(x_{0:M})}{p_\theta(y_{0:M})}, \tag{3.3}$$

where $p_\theta(y_{0:M}) = \int p_\theta(x_{0:M}, y_{0:M})dx_{0:M}$. However, an SMC algorithm can be used to approximate (3.3). A brief review of how this sampling algorithm proceeds is as follows and further details can be found in Doucet et al. (2000); Doucet and Johansen (2009).

SMC methods approximate $p_\theta(x_{0:n}|y_{0:n})$ using a set of K weighted random samples $X_{0:n}^{1:K} = (X_{0:n}^1, ..., X_{0:n}^K)$, also called particles, having positive weights $W_n = W_n^{1:K}$, so that $p_\theta(x_{0:n}|y_{0:n}) \approx \hat{p}_\theta(x_{0:n}|y_{0:n}) = \sum_{k=1}^{K} W_n^k \delta_{X_{0:n}^k}(x_{0:n})$. Here, δ denotes the Dirac delta function. To do so, one starts at $n = 0$ by sampling X_0^k from an importance

density $M_0^\phi(\cdot|y_0)$, parametrized with ϕ, where ϕ can depend on the static parameters θ. For any $n \geqslant 1$, we first resample an ancestor variable A_{n-1}^k that represents the 'parent' of particle $X_{0:n}^k$ according to $A_{n-1}^k \sim r(\cdot|W_{n-1})$, where r is a categorical distribution on $\{1, ..., K\}$ with probabilities W_{n-1}. We then set $W_{n-1} = \frac{1}{K}$ and proceed by extending the path of each particle by sampling from a transition kernel $X_n^k \sim M_n^\phi(\cdot|y_n, X_{0:n-1}^{A_{n-1}^k})$. This yields an updated latent path $X_{0:n}^k = (X_{0:n-1}^{A_{n-1}^k}, X_n^k)$ for which we compute the incremental importance weight

$$\alpha_n(X_{0:n}^k) = \frac{\gamma_\theta(X_{0:n}^k)}{\gamma_\theta(X_{0:n-1}^k) M_n^\phi(X_n^k|y_n, X_{0:n-1}^{A_{n-1}^k})}.$$

We set $w_n(X_{0:n}^k) = W_{n-1}^k \alpha_n(X_{0:n}^k)$ as well as $W_n^k = \frac{w_n(X_{0:n}^k)}{\sum_l w_n(X_{0:n}^l)}$ and define

$$\hat{Z}_n^{\theta,\phi} := \prod_{m=0}^{n} \sum_{k=1}^{K} w_m(X_{0:m}^k),$$

which is an unbiased and strongly consistent estimator of $p_\theta(y_{0:n})$, see Del Moral (1996). A pseudo-code (Algorithm 1) for this standard SMC sampler can be found in Appendix 3.8.1. It is possible to perform the resampling step only if some condition on W_{n-1} is satisfied, see Algorithm 1. For simplicity, we assume that the particles are resampled at every step. The density of all variables generated by this SMC sampler for a fixed static parameter value θ is given by

$$q_\phi(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l|\theta) = w_M^l \prod_{k=1}^{K} M_0^\phi(x_0^k|y_0)$$
$$\cdot \prod_{n=1}^{M} \prod_{k=1}^{K} r(a_{n-1}^k|w_{n-1}) M_n^\phi(x_n^k|y_n, x_{0:n-1}^{a_{n-1}^k}),$$

where l is a final particle index drawn from a categorical distribution with weights W_M. Since $\hat{Z}_n^{\theta,\phi}$ is unbiased, we have

$$\mathbb{E}_{q_\phi(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l|\theta)} \left[\hat{Z}_M^{\theta,\phi} \right] = p_\theta(y_{0:M}). \tag{3.4}$$

3.3 Variational bounds for state space models using SMC samplers

Variational inference (Jordan et al., 1999; Wainwright and Jordan, 2008; Blei et al., 2017) allows Bayesian inference to scale to large data sets (Hoffman et al., 2013)

and is applicable to a wide range of models (Ranganath et al., 2014; Kucukelbir et al., 2017). It generally postulates a family of approximating distributions with variational parameters that minimize some divergence, most commonly the KL divergence, between the approximating distribution and the posterior. The quality of the approximation hinges on the expressiveness of the variational family.

Let $q_\psi(\theta)$ be a distribution on Θ with variational parameters ψ. We aim to approximate the posterior density $p(\theta, x_{0:M}|y_{0:M})$ in (3.1) as an appropriate marginal density of a variational distribution on an extended space. This extended variational distribution describes auxiliary variables from an SMC sampler and is of the form

$$q_{\psi,\phi}(\theta, x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l) := q_\psi(\theta) q_\phi(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l|\theta), \tag{3.5}$$

defined precisely below. Note that sampling from the extended variational distribution (3.5) just means sampling $\theta \sim q_\psi(\theta)$ and then generate all auxiliary random variables by running a particle filter using the sampled value θ as the static parameter.

We introduce the proposed variational bound first as a lower bound on $\log p(y_{0:M}) - \mathrm{KL}(q_\psi(\theta)||p(\theta|y_{0:M}))$. We then show that optimizing the proposed bound means minimizing the KL-divergence between the extended variational distribution (3.5) and an extended target density that resembles closely the density targeted in particle MCMC methods.

We can write $p(\theta|y_{0:M}) = p(\theta)p_\theta(y_{0:M})/p(y_{0:M})$. Hence, using the fact that the likelihood estimator is unbiased (3.4) and due to Jensen's inequality,

$$\begin{aligned}
&- \mathrm{KL}(q_\psi(\theta)||p(\theta|y_{0:M})) + \log p(y_{0:M}) \\
=&\mathbb{E}_{q_\psi(\theta)}\left[\log p_\theta(y_{0:M}) + \log p(\theta) - \log q_\psi(\theta)\right] \\
=&\mathbb{E}_{q_\psi(\theta)}\left[\log \mathbb{E}_{q_\phi(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l|\theta)}\left[\hat{Z}_M^{\theta,\phi}\right] + \log \frac{p(\theta)}{q_\psi(\theta)}\right] \\
\geqslant&\mathbb{E}_{q_\psi(\theta)}\left[\mathbb{E}_{q_\phi(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l|\theta)}\left[\log \hat{Z}_M^{\theta,\phi}\right] + \log \frac{p(\theta)}{q_\psi(\theta)}\right] \\
=:&\mathcal{L}(\psi, \phi).
\end{aligned}$$

In particular, $\mathcal{L}(\psi, \phi)$ is a lower bound on $p(y_{0:M}) - \mathrm{KL}(q_\psi(\theta)||p(\theta|y_{0:M}))$. This motivates maximizing $\mathcal{L}(\psi, \phi)$ with respect to ψ in order to learn an approximation of the posterior distribution of the static parameter.

Remark 6 (Inference for multiple independent time series). *Instead of considering one latent process $\{X\}$ and observable process $\{Y\}$, we can also consider*

S independent latent processes $\{X^s\}_{s=1,...,S}$ with corresponding observable processes $\{Y^s\}_{s=1,...,S}$ described by the same static parameter θ. We obtain a lower bound on $p(y_{0:M}) - KL(q_\psi(\theta)||p(\theta|y^1_{0:M}, ..., y^S_{0:M}))$ given by

$$\mathbb{E}_{q_\psi(\theta)}\left[\mathbb{E}_{\prod_s q_\phi(x^{s,1:K}_{0:M}, a^{s,1:K}_{0:M-1}, l^s|\theta)}\left[\sum_{s=1}^{S} \log \hat{Z}^{\theta,\phi}_{M,s}\right] + \log \frac{p(\theta)}{q_\psi(\theta)}\right],$$

where $\hat{Z}^{\theta,\phi}_{M,s}$ is the estimator of $p_\theta(y^s_{0:M})$. Note that we can obtain an unbiased estimate of this bound by sampling an element $s \in \{1, ..., S\}$ and using $S \cdot \log \hat{Z}^{\theta,\phi}_{M,s}$ as an estimate of $\sum_{s'=1}^{S} \log \hat{Z}^{\theta,\phi}_{M,s'}$, thereby allowing our method to scale to a large number of independent time series. For ease of exposition, we formulate our results for a single time series only.

Next, we show that the variational bound can be represented as the difference between the log-evidence and the KL divergence between the variational distribution and an extended target density. More concretely, following Andrieu et al. (2010), we consider a target density on the extended space $\Theta \times \mathcal{X}$, $\mathcal{X} := (\mathbb{R}^{d_x})^{(M+1)K} \times \{1, ..., K\}^{MK+1}$,

$$\tilde{\pi}(\theta, x^{1:K}_{0:M}, a^{1:K}_{0:M-1}, l) := \frac{\pi(\theta, x^l_{0:M})}{K^{M+1}} \frac{q_\phi(x^{1:K}_{0:M}, a^{1:K}_{0:M-1}, l|\theta)}{M^\phi_0(x^{b^l_0}_0|y_0) \prod_{n=1}^{M} r(b^l_{n-1}|w_{n-1}) M^\phi_n(x^{b^l_n}_n|y_n, x^{b^l_{n-1}}_{0:n-1})}.$$

Here, we have defined $b^l_M = l$ and $b^l_n = a^{b^l_{n+1}}_n$ for $n = M-1, ..., 1$, i.e. b^l_n is the index that the ancestor of particle $X^l_{0:M}$ at generation n had. It follows, using $r(b^l_n|w_{n-1}) = w^{b^l_{n-1}}_{n-1}$, that the ratio between the extended target density and the variational distribution is given by

$$\begin{aligned}&\frac{\tilde{\pi}(\theta, x^{1:K}_{0:M}, a^{1:K}_{0:M-1}, l)}{q_{\phi,\psi}(\theta, x^{1:K}_{0:M}, a^{1:K}_{0:M-1}, l)} \\ =&\frac{K^{-(M+1)} p(\theta) p_\theta(x^l_{0:M}, y_{0:M})/p(y_{0:M})}{q_\psi(\theta) W^l_M M^\phi_0(x^{b^l_0}_0|y_0) \prod_{n=1}^{M} W^{b^l_{n-1}}_{n-1} M^\phi_n(x^{b^l_n}_n|y_n, x^{b^l_{n-1}}_{0:n-1})}.\end{aligned} \tag{3.6}$$

Proposition 7 (KL divergence in extended space). *It holds that*

$$\mathcal{L}(\psi, \phi) = -KL(q_{\psi,\phi}||\tilde{\pi}) + \log p(y_{0:M}).$$

The proof can be found in Appendix 3.8.2. Recall that we have introduced $\mathcal{L}(\psi, \phi)$ so that its maximisation pushes the variational approximation of the static parameter θ closer to its true posterior as measured by the KL divergence. The above

proposition shows that this objective also minimizes the KL divergence between densities on an extended space that includes multiple latent paths. To elucidate further the relation between the variational distribution of a single latent path and its posterior, we need to introduce a further distribution. Consider the density under $\tilde{\pi}$ of the variables generated by a SMC algorithm conditional on a fixed latent path $(x^l_{0:M}, b^l_{0:M-1})$. This is known as a conditional SMC algorithm (Andrieu et al., 2010), with distribution given by

$$\begin{aligned}&\tilde{\pi}_{\text{CSMC}}(x_{0:M}^{\neg b^l_{0:M}}, a_{0:M-1}^{\neg b^l_{0:M-1}} | \theta, x^l_{0:M}, b^l_{0:M}) \\ =&\frac{q_\phi(x^{1:K}_{0:M}, a^{1:K}_{0:M-1}, l|\theta)}{W^l_M M^\phi_0(X_0^{b^l_0}|y_0)\prod_{n=1}^{M} r(b^l_{n-1}|W_{n-1}) M^\phi_n(x_n^{b^l_n}|y_n, x_{0:n-1}^{b^l_{n-1}})},\end{aligned}$$

where $\neg b^l_{0:M}$ are the indices of all particles that are not equal to $b^l_{0:M}$. We obtain the following corollary proved in Appendix 3.8.3.

Corollary 8 (Marginal KL divergence and marginal ELBO). *The KL divergence in the extended space is an upper bound on the KL divergence between the marginal variational approximation and the posterior, with the gap between bounds being*

$$\begin{aligned}&KL\left(q_{\psi,\phi}(\theta, x^{1:K}_{0:M}, a^{1:K}_{0:M-1}, l)||\tilde{\pi}(\theta, x^{1:K}_{0:M}, a^{1:K}_{0:M-1}, l)\right) - KL\left(q_{\psi,\phi}(\theta, x_{0:M})||\pi(\theta, x^l_{0:M})\right) \\ =&\mathbb{E}_{q_{\psi,\phi}(\theta, x^l_{0:M}, b^l_{0:M})}\Big[\\ &\quad KL(q_\phi(x_{0:M}^{\neg b^l_{0:M}}, a_{0:M-1}^{\neg b^l_{0:M-1}})|\theta, x^l_{0:M}, b^l_{0:M})||\tilde{\pi}_{CSMC}(x_{0:M}^{\neg b^l_{0:M}}, a_{0:M-1}^{\neg b^l_{0:M-1}}|\theta, x^l_{0:M}, b^l_{0:M}))\Big].\end{aligned}$$

Particularly, $\mathcal{L}$ is a lower bound compared to the standard ELBO using the marginal $q_{\psi,\phi}(\theta, x_{0:M})$ with $x^l_{0:M} = x_{0:M}$ as the variational distribution:

$$\mathcal{L}(\psi, \phi) \leqslant -KL\left(q_{\psi,\phi}(\theta, x_{0:M})||\pi(\theta, x_{0:M})\right) + \log p(y_{0:M}).$$

The proposed surrogate objective resembles variational bounds with auxiliary variables (Salimans et al., 2015; Maaløe et al., 2016; Ranganath et al., 2016) where the gap between the two bounds is expressed by the KL-divergence between the variational approximation of the auxiliary variable given the latent variable of interest and a so-called reverse model. Here, this reverse model is specified by the conditional

SMC algorithm. The above corollary implies that the variational bound is looser than the standard ELBO with the auxiliary variables integrated out. This marginal variational distribution cannot in general be evaluated analytically. However, we can obtain unbiased estimates of it by computing the log-likelihood estimate under a conditional SMC algorithm, resembling a particle Gibbs update. This constitutes an extension of Proposition 1 in Naesseth et al. (2018). We present a proof in Appendix 3.8.4.

Proposition 9 (Marginal variational distribution). *We have*

$$q_{\psi,\phi}(\theta, x^l_{0:M}, b^l_{0:M}) = q_\psi(\theta)\gamma_\theta(x^l_{0:M}) \cdot \mathbb{E}_{\tilde{\pi}_{CSMC}(x^{\neg b^l_{0:M}}_{0:M}, a^{\neg b^l_{0:M}}_{0:M-1} | \theta, x^l_{0:M})} \left[\left(\hat{Z}^{\theta,\phi}_M \right)^{-1} \right]$$

and there exists $c(\theta, \phi) < \infty$ *so that*

$$KL(q_{\psi,\phi}(\theta, x_{0:M}) || p(\theta, x_{0:M}|y_{0:M}) \leqslant \mathbb{E}_{q_\psi(\theta)} \left[\frac{c(\theta,\phi)}{K} \right] + KL(q_\psi(\theta)||p(\theta|y_{0:M})).$$

The last inequality in Proposition 9 is a straightforward extension of an analogous result in the EM setting (Naesseth et al., 2018). It implies that, for fixed variational parameters ψ and ϕ, the approximation becomes more accurate for increasing the number of particles K. The constant $c(\theta, \phi)$ is of order M and we refer to the experiments in Naesseth et al. (2018) that illustrate that one can achieve a good approximation of $p_\theta(x_{0:M}|y_{0:M})$ in a simple model even for $M \to \infty$ by setting $K \propto M$. The dependence of the constant $c(\theta, \phi)$ on the dimension d_x of the latent state or of the static parameter appears more involved. We refer to Huggins et al. (2019) for details on bounding the divergence between SMC approximations and the posterior distribution that might yield more explicit bounds for specific models.

Sampling from the variational distribution can be seen as an extension of visualizing the expected importance weighted approximation in Importance Weighted Auto-Encoders (Cremer et al., 2017). Since this distribution can be high-dimensional, the preceding proposition gives an alternative to kernel-density estimation.

Lastly, from a different angle, the variational objective can be seen as a sequential variational-autoencoding (VAE) bound. Indeed, as a consequence of Proposition 7 and equation (3.6), we obtain immediately the following result. We elaborate on it further in the next section.

Corollary 10 (Sequential VAE representation). *The variational bound can be written as*

$$\begin{aligned}&\mathcal{L}(\psi,\phi)\\ =&\mathbb{E}_{q_\psi(\theta)}\left[\mathbb{E}_{q_\phi(x_{0:M}^{1:K},a_{0:M-1}^{1:K},l|\theta)}\left[\sum_{n=0}^{M}\log g_\theta(y_n|x_n^{b_n^l})-\log W_n^{b_n^l}+\log\frac{f_\theta(x_n^{b_n^l}|x_{n-1}^{b_{n-1}^l},y_{n-1})}{M^\phi(x_n^{b_n^l}|y_n,x_{0:n}^{b_{n-1}^l})}\right]\right]\\ &-(M+1)\log K-KL(q_\psi(\theta)||p(\theta)).\end{aligned}$$

3.4 Related Work

The representation in Corollary 10 allows us to contrast the variational bound to previously considered sequential VAE frameworks (Chung et al., 2015; Archer et al., 2015; Fraccaro et al., 2016; Krishnan et al., 2017; Goyal et al., 2017). The introduced bound contains the cross-entropy between the proposal distribution and the likelihood common to sequential VAE bounds. However, this reconstruction error is only evaluated for surviving particles. Similarly, while a sequential VAE framework includes a KL-divergence between the proposal distribution and the prior transition probability, the log-ratio of these two densities is only evaluated for a surviving path. Most work using sequential VAEs have considered observation and state transition models parametrised by neural networks, and given the high-dimensionality of the static parameters, have confined their analysis to variational EM inferences. This is also the case for the approaches in Maddison et al. (2017); Naesseth et al. (2018); Le et al. (2018), to which this work is most closely related. They have demonstrated that resampling increases the variational bound compared to a sequential IWAE (Burda et al., 2015) approach. Rainforth et al. (2018) demonstrated that increasing the number of particles leads to a worse signal to noise ratio of the gradient estimate of the proposal parameters in an IWAE setting. Le et al. (2018) suggested to use fewer particles without resampling for calculating the proposal gradient. A possible approach left for future work would be to consider a different resampling threshold for the proposal gradients. Finally, the objective in this work differs from adaptive SMC approaches optimizing the reverse KL-divergence (or χ^2-divergence) between the posterior and the proposal, cf. Cornebise et al. (2008); Gu et al. (2015).

3.5 Optimization of the variational bound

The gradient of the variational bound is given by

$$\nabla_{\psi,\phi}\mathcal{L}(\psi,\phi) = \nabla_{\psi,\phi}\mathbb{E}_{q_\psi(\theta)}\left[\mathbb{E}_{q_\phi(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l|\theta)}\left[\log \hat{Z}_M^{\theta,\phi}\right]\right] + \nabla_\psi \mathbb{E}_{q_\psi(\theta)}\left[\log \frac{p(\theta)}{q_\psi(\theta)}\right]. \tag{3.7}$$

We focus on the gradient of the first expectation and note that the gradient of the second expectation can be estimated by standard (black-box) approaches in variational inference, depending of course on the chosen variational approximation. If for instance the variational distribution over the static parameters is continuously reparametrisable, one can use standard low-variance reparametrised gradients (Kingma and Welling, 2014; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014). This is the gradient estimator that we use in our experiments in combination with mean-field variational families. We assume that the proposals $X_n^k \sim M_n^\phi(\cdot|y_n, x_{0:n-1}^{a_{n-1}^k})$ are reparametrisable, i.e. there exists a differentiable deterministic function h_ϕ such that $X_n^k = h_\phi(X_{0:n-1}^{A_{n-1}^k}, \epsilon_n^k)$, with $\epsilon_n^k \sim p(\cdot)$ continuous and independent of ϕ. Similarly, we assume that the variational distribution of the static parameters is reparametrisable, i.e. there exists a differentiable deterministic function h_ψ such that $\theta = h_\psi(\eta)$, with $\eta \sim p(\cdot)$ continuous and independent of ψ. We abbreviate $\boldsymbol{\epsilon} = \epsilon_{0:M}^{1:K}$, $\boldsymbol{x} = x_{0:M}^{1:K}$ and $\boldsymbol{a} = a_{0:M-1}^{1:K}$. Using the product rule, observe that the first gradient in (3.7) is

$$\begin{aligned}
&\nabla_{\psi,\phi}\int p(\eta)p(\boldsymbol{\epsilon})q_\phi(\boldsymbol{a}|\theta,\boldsymbol{x})\cdot\log\hat{Z}_M^{\theta,\phi}d(\eta,\boldsymbol{a},\boldsymbol{\epsilon})\Big|_{\theta=h_\psi(\eta),\boldsymbol{x}=h_\phi(\boldsymbol{\epsilon})}\\
&=\int p(\eta)p(\boldsymbol{\epsilon})\nabla_{\psi,\phi}q_\phi(\boldsymbol{a}|\theta,\boldsymbol{x})\cdot\log\hat{Z}_M^{\theta,\phi}d(\eta,\boldsymbol{a},\boldsymbol{\epsilon})\Big|_{\theta=h_\psi(\eta),\boldsymbol{x}=h_\phi(\boldsymbol{\epsilon})}\\
&=\mathbb{E}_{p(\eta)p(\boldsymbol{\epsilon})q_\phi(\boldsymbol{a}|h_\psi(\eta),h_\phi(\boldsymbol{\epsilon}))}\left[\nabla_{\psi,\phi}\log\hat{Z}_M^{h_\psi(\eta),\phi}+\nabla_{\psi,\phi}\log q_\phi(\boldsymbol{a}|h_\psi(\eta),h_\phi(\boldsymbol{\epsilon}))\log\hat{Z}_M^{h_\psi(\eta),\phi}\right].
\end{aligned}$$

Analogously to Maddison et al. (2017); Le et al. (2018); Naesseth et al. (2018) in a variational EM framework, we have also ignored the second summand in the gradient due to its high variance in our experiments.[1] We take Monte Carlo samples of the expectation above and optimize the bound using Adam (Kingma and Ba, 2014). It is also possible to use natural gradients (Amari, 1998), see Appendix 3.8.5.

[1] A differentiable particle filtering algorithm has been suggested recently in Corenflos et al. (2021) using optimal transport ideas which allows for consistent gradient estimates.

3.6 Experiments

3.6.1 Linear Gaussian state space models

Regularisation in a high-dimensional model. We illustrate potential benefits of a fully Bayesian approach in a standard linear Gaussian state space model

$$f_\theta(x_n|x_{n-1}) = \mathcal{N}(Ax_{n-1}, \Sigma_x), \tag{3.8}$$

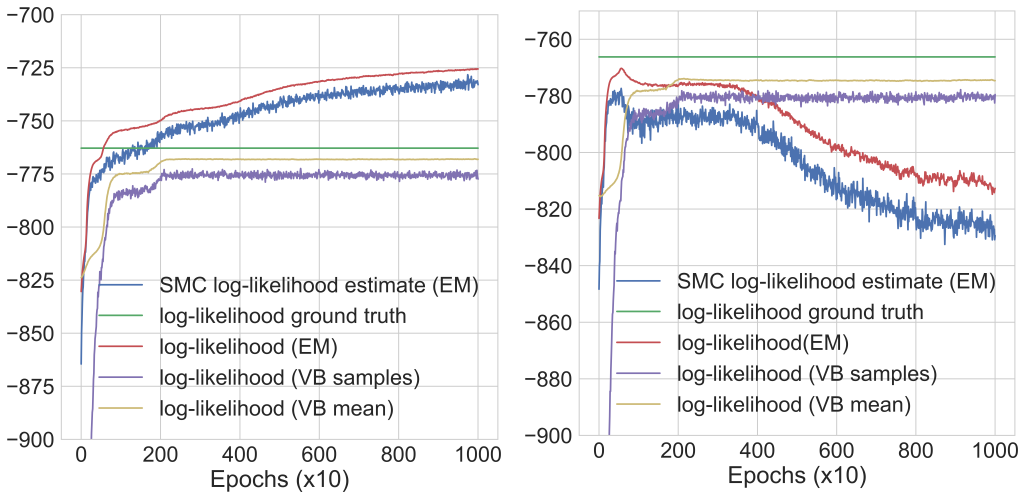
$$g_\theta(y_n|x_n) = \mathcal{N}(Bx_n, \Sigma_y), \tag{3.9}$$

with initial state distribution $X_0 \sim \mathcal{N}(A^0, \Sigma_x^0)$ and parameters $A, \Sigma_x, \Sigma_x^0 \in \mathbb{R}^{d_x \times d_x}$, $A^0 \in \mathbb{R}^{d_x}$, $B \in \mathbb{R}^{d_x \times d_y}$, and $C, \Sigma_y \in \mathbb{R}^{d_x \times d_y}$. Naesseth et al. (2018) have shown in a linear Gaussian model that learning the proposal yields a higher variational lower bound compared to proposing from the prior, and the variational bound is close to the true log-marginal likelihood for both sparse and dense emission matrices B. However, an EM approach might easily over-fit, unless one employs some regularisation, such as stopping early if the variational bound decreases on some test set. We demonstrate this effect by re-examining one of the experiments in Naesseth et al. (2018), setting $(d_x, d_y) = (10, 3)$, $M = 10$ and assume that Σ_x, Σ_x^0 and Σ_y are all identity matrices. Furthermore, $A^0 = 0$ and $(A_{ij}) = \alpha^{|i-j|+1}$ with $\alpha = 0.42$, and B has randomly generated elements with $B_{ij} \sim \mathcal{N}(0, 1)$. We assume that the proposal density is

$$M_{n+1}^\phi(x_{n+1}|x_n, y_{n+1}) = \mathcal{N}(x_{n+1}|A_\phi x_n + B_\phi y_{n+1}, \Sigma_\phi),$$

and $M_0^\phi(x_0|y_0) = \mathcal{N}(x_0|A_\phi^0 + B_\phi y_0, \Sigma_\phi^0)$, with Σ_ϕ and Σ_ϕ^0 diagonal matrices. We perform both a variational EM approach and a fully approximate Bayesian approach over the static parameters using $K = 4$ particles. In the latter case, we place Normal priors $B_{ij} \sim \mathcal{N}(0, 10)$ and $A_{ij} \sim \mathcal{N}(0, 1)$. Furthermore, we suppose that a priori Σ_y is diagonal with variances drawn independently from an Inverse Gamma distribution with shape and scale parameters of 0.01 each. A mean-field approximation for the static parameters is assumed. We suppose that the variational distribution over each element of A and B is a normal distribution and the approximation over the diagonal elements of Σ_y is log-normal. For identifiability reasons, we assume that Σ_x, Σ_x^0 and A^0 are known. We compare the EM and VB approach in terms of log-likelihoods on out-of-sample data assuming training and testing on 10 iid sequences. Figure 3.1 shows that in contrast to the VB approach, the EM approach attains a higher

log-likelihood on the training data with a lower log-likelihood on the test set as the training progresses.



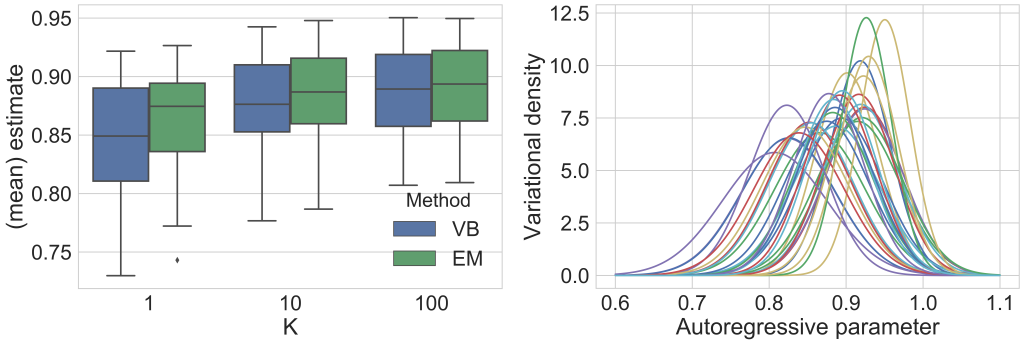
(a) Log-likelihood on training data.

(b) Log-likelihood on testing data.

Figure 3.1: Log-likelihood for linear Gaussian state space models. Log-likelihood values are computed using Kalman filtering. The static parameters used in the VB case are the mean of the variational distribution (VB mean) or the samples from the variational distribution (VB samples) as they are drawn during training.

Approximation bias in a low-dimensional model. Variational approximations for the latent path can yield biased estimates of the static parameters, see Turner and Sahani (2011). We illustrate that this bias decreases for increasing K in a two-dimensional linear Gaussian model, both in an EM and VB setting. We therefore consider inference in a linear Gaussian state space model (3.8-3.9) with two-dimensional latent states and one-dimensional observations. The state transition matrix is assumed to be determined by the autoregressive parameter λ with $A = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$. We consider inference over λ as the static parameter and fix $B = (1, 1)$ with Σ_x and Σ_y being identity matrices. We simulate 30 realisations of length $M = 100$ each using $\lambda = 0.9$. Inference is performed with different initialisations and learning rates over the simulated datasets. It has been documented in such a linear Gaussian model, see Turner and Sahani (2011), that Gaussian variational approximations of the latent path that factorise over the state components underestimate λ. We observe the same

effect in Figure 3.2a when using just $K = 1$ particle. However, increasing the number of particles used during inference reduces this bias. Furthermore, we find that point estimates of the static parameters show some variation over different simulations, while an approximate Bayesian approach can be argued to better account for this uncertainty. The variational distributions for θ for each of the simulations using $K = 100$ particles is shown in Figure 3.2b, confirming that they all put significant mass on the ground truth. Let us remark that these experiments also complement those in Le et al. (2018), where it is illustrated that increasing K improves learning point estimates of the static parameters in a Gaussian model with a one-dimensional latent state. Indeed, as shown next, the marginal variational distribution, cf. Proposition 9, allows not just for dependencies in the latent states across time, but also across different state dimensions, even if they are independent under the proposal.



(a) Point estimate of the autoregressive parameter λ in the EM case or the variational mean in the VB case over 30 simulations for particles $K \in \{1, 10, 100\}$ particles.

(b) Variational distribution of the autoregressive parameter λ using $K = 100$ for each of the 30 simulations.

Figure 3.2: Inference on the autoregressive parameter λ over 30 simulations of length $M = 100$. Ground truth values are $\lambda = 0.9$.

Marginal variational distribution in a low-dimensional model. In an additional experiment, we evaluate if the variational approximation from Proposition 9 of the latent path matches the distribution of its true posterior. We consider the above state space model over 2 time steps as in Turner and Sahani (2011). Note that for given static parameters, the posterior is Gaussian. Indeed, for $\boldsymbol{x} = (x_0^{(0)}, x_0^{(1)}, x_1^{(0)}, x_1^{(1)})$,

where $x_n^{(i)}$ denotes dimension i of x_n, we have $p(\boldsymbol{x}|y_{0:1}, \lambda) = \mathcal{N}(\mu_{x|y}, \Sigma_{x|y})$ with

$$\Sigma_{x|y}^{-1} = \begin{pmatrix} 2 & 1 & -\lambda & 0 \\ 1 & 2 & 0 & -\lambda \\ -\lambda & 0 & 2 & 1 \\ 0 & -\lambda & 1 & 2 \end{pmatrix}, \mu_{x|y} = \Sigma_{x|y} \begin{pmatrix} y_0 \\ y_0 \\ y_1 \\ y_1 \end{pmatrix},$$

assuming $X_0 \sim \mathcal{N}(0, \frac{1}{1-\lambda^2} I)$ is drawn from its stationary distribution. We visualise the posterior distribution along with the marginal variational distribution

$$q_\phi(x_{0:M}^l|\theta) = \gamma_\theta(x_{0:M}^l)\mathbb{E}_{\tilde{\pi}_{\text{CSMC}}(x_{0:M}^{\neg b_{0:M}^l}, a_{0:M-1}^{\neg b_{0:M-1}^l}|\theta, x_{0:M}^l)}\left[\left(\hat{Z}_M^{\theta,\phi}\right)^{-1}\right]$$

in Figure 3.3 using $K = 100$ particles and 50 samples for the expectation. We find that the approximation mirrors the true posterior. In particular, it accounts for explaining-away between different dimensions of the latent state, although we have used isotropic proposals.

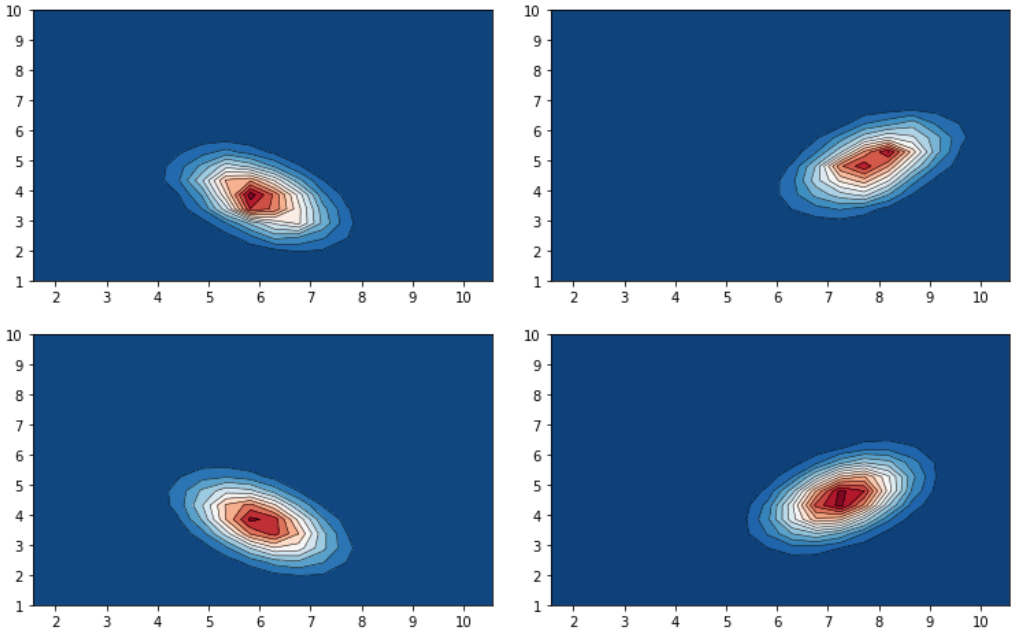
3.6.2 Stochastic volatility models

To show that our method allows inference of latent states and static parameters of higher dimensions, we consider a multivariate stochastic volatility model,

$$f_\theta(x_n|x_{n-1}) = \mathcal{N}(\mu + \text{diag}(a)(x_{n-1} - \mu), \Sigma_x),$$
$$g_\theta(y_n|x_n) = \mathcal{N}(0, \exp(\text{diag}(x_n)),$$

where $X_0 \sim \mathcal{N}(\mu, \Sigma_x^0)$ with $x_n, y_n, \mu, a \in \mathbb{R}^D$, and covariance matrix $\Sigma_x \in \mathbb{R}^{D\times D}$, $\theta = (\mu, a, \Sigma_x, \Sigma_x^0)$. This model has been considered in Guarniero et al. (2017) using particle MCMC methods under the restriction that Σ_x is band-diagonal to reduce the number of parameters. It is also more general than that entertained in Naesseth et al. (2018) with Σ_x assumed diagonal, see also Chib et al. (2009) for a review on stochastic volatility models. We consider a fully Bayesian treatment as in Guarniero et al. (2017), applied to the same data set of 90 monthly returns (9/2008 to 2/2016) of 20 exchange rates with respect to the US dollar as reported by the Federal Reserve System. The specification of the prior and variational forms of the static parameters are explained in Appendix 3.8.6. We consider proposals of the form

$$M_\phi(x_{n+1}|y_{n+1}, x_n) = \mathcal{N}(\mu + \text{diag}(a)(x_n - \mu), \Sigma^\phi),$$



(a) Joint distribution of the latent states at the second time step. Top: variational approximation, bottom: true posterior.

(b) Joint distribution of the first state component at the first and second time step. Top: variational approximation, bottom: true posterior.

Figure 3.3: Two-dimensional contour plots of the distribution of the latent path over two time steps and two state components. Function arguments are set to the ground truth state values as simulated if they are not shown.

where Σ^{ϕ} is diagonal and using $K = 50$ particles. Densities of the variational approximation that correspond to the GBP exchange rate can be found in Appendix 3.8.6, Figure 3.4, which are largely similar to those obtained in (Guarniero et al., 2017). Furthermore, we approximate the one- and two-step predictive distributions

$$p(y_{m+p}|y_{0:m}) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{K} W_m^{k,s}\delta_{X_{m+p}^{k,s}} p_{\theta_s}(y_{m+p}|X_{m+p}^{k,s})$$

for $p \in \{1,2\}$,where $\theta_1,...,\theta_S \sim q_\psi(\theta)$, $\sum_{k=1}^{K} W_m^{k,s}\delta_{X_m^k}$ is the approximation of $p_{\theta_s}(x_m|y_{0:m})$ by the particle filter and $X_n^s \sim p_{\theta_s}(x_n^{k,s}|X_{n-1}^s, Y_{n-1}^{k,s})$ with $Y_n^s \sim p_{\theta_s}(y_n^{k,s}|X_n^{k,s})$ for $n = m+1,...,m+p$ simulated from the generative model. The predictive distributions are evaluated using a log scoring rule (Gneiting and Raftery, 2007; Geweke and Amisano, 2010) to arrive at the predictive log-likelihoods in Table 3.1. The full variational approach attains higher predictive log-likelihoods.

Table 3.1: Average p-step predictive log-likelihoods per observation for the stochastic volatility model with different number of particles K and number of samples S from the variational distribution. In the EM case, we run S particle filters with the same optimal static values, whereas we use S particle filters using a sample of the static parameters from the variational distribution in the VB case. Mean estimates with standard deviation in parentheses based on 100 replicates.

	$(S, K) = (4, 50)$	
Method	$p = 1$	$p = 2$
EM	9.697 (0.008)	9.716 (0.008)
VB	9.707 (0.011)	9.728 (0.015)
	$(S, K) = (20, 100)$	
Method	$p = 1$	$p = 2$
EM	9.690 (0.003)	9.713 (0.003)
VB	9.701 (0.004)	9.727 (0.005)

3.6.3 Non-linear stochastic Hawkes processes

There has been an increasing interest in modelling asynchronous sequential data using point processes in various domains, including social networks (Linderman and Adams, 2014; Wang et al., 2017), finance (Bacry et al., 2015), and electronic health (Lian et al., 2015). Recent work (Du et al., 2016; Mei and Eisner, 2017; Xiao et al., 2017b,a) have advocated the use of neural networks in a black-box treatment of point process dynamics.

We illustrate that our approach allows scalable probabilistic inference for continuous-time event data $\{T_n, C_n\}_{n>0}$, $T_n < T_{n+1}$, where T_n is the time when the n-th event occurs and $C_n \in \{1, ..., D\}$ is an additional discrete mark associated with the event. We consider describing such a realisation as a D-variate point process with intensities $\lambda_t = h_\theta(\mu + \sum_{b=1}^{B} \Xi_t^b)$, driven by B continuous time processes

$$\Xi_t^b = \sum_{n \geqslant 1} \beta_b A_n^b e^{-\beta_b (t - T_n)} 1_{[0,t)}(T_n), \quad t > 0,$$

and a non-negative monotone function h_θ. Moreover, $\mu, A_n \in \mathbb{R}^D$ and $\beta^b > 0$. Importantly, we allow A_n^b to depend on C_n, and the i-th component of A_n^b describes by how

much the n-th event excites, if $(A_n^b)^i > 0$, or inhibits, if $(A_n^b)^i < 0$, subsequent events of type i. It is possible to view the dynamics as a discrete-time SSM; the essential idea being that Ξ^b is piecewise-deterministic between events, see Appendix 3.8.7 for details along with related work on Hawkes point processes (Hawkes, 1971a). Let us define the discrete-time latent process $X_{n+1} = (Z_n, A_n)$ with $Z_n = \Xi_{T_n}$, $A_n = \text{vec}(A_n^1, ..., A_n^B)$. Standard theory about point processes, see Daley and Vere-Jones (2003), implies that the observation density is given by $g_\theta(t_n, c_n|z_{n-1}) = \lambda_{t_n}^{c_n} \exp\left(-\sum_{i=1}^D \int_{t_{n-1}}^{t_n} \lambda_s^i ds\right)$, where our model specification yields λ_s as a deterministic function between T_{n-1} and T_n given Z_{n-1}. Similar to Mei and Eisner (2017), we set $h_\theta(y) = \nu \,\text{softplus}(y/\nu) = \nu \log(1 + \exp(y/\nu))$ as a scaled softplus function with ν a static parameter. Next, we specify the dynamics of A_n. We take the arguable most simple model, assuming $f_\theta(a_n|a_{n-1}, z_{n-1}, c_n) = \mathcal{N}(\sum_d \alpha_d \delta_{c_n d}, \sum_d \sigma_d^2 \delta_{c_n d})$ with $\alpha_1, ..., \alpha_D \in \mathbb{R}^{BD}$ and $\sigma_1^2, ..., \sigma_D^2$ positive diagonal matrices, while remarking in passing that our approach allows readily for extensions that could include temporal dynamics between successive intensity jumps or intensity jumps instantaneously correlated across different marks and time scales. Due to the piecewise deterministic decay of Ξ, note that $Z_n^b|Z_{n-1}^b, A_n^b = e^{-\beta^b(T_n - T_{n-1})} Z_{n-1}^b + \beta^b A_n^b$, so the state transition of the process X is fully specified.

We apply our model to 20 days of high-frequency financial data for the BUND futures contract. The data is available as part of the tick library (Bacry et al., 2017) with 4 event types: (i) mid-price up moves, (ii) mid-price down moves, (iii) buyer-initiated trades leaving the mid-price unchanged and (iv) seller-initiated trades not changing the mid. We train our model on 15 days and evaluate how well it predicts the type of the next event on out of sample data from the remaining 5 days.

Table 3.2 reports better predictive performance of the proposed model in comparison with two benchmark models. First, a linear Hawkes process model estimated using maximum likelihood. Second, to illustrate that improved predictions might not be just explained due to inhibitory effects, we also compare against a non-linear Hawkes model. The latter can be seen, and has been implemented, as a limiting case of our generative model letting $\sigma_d^2 \to 0$, with inference thus performed using stochastic gradient descent of the negative log-likelihood. Predictions are Monte Carlo samples of the next event realisation from the generative model. Further details including

Table 3.2: Prediction metric for different Hawkes process models on the test set of around 206k events. The stochastic Hawkes model is trained with 20 particles and uses $K \in \{20, 80\}$ particles during testing.

Method	Error rate next mark
Linear Hawkes	43.3 %
Non-linear Hawkes	40.9 %
Non-linear stochastic Hawkes ($K = 20$)	40.0%
Non-linear stochastic Hawkes ($K = 80$)	39.3%

assumptions on the variational distributions and the predictive performance using a smaller training set are given in Appendix 3.8.8.

3.7 Conclusion

This paper has explored an inference approach that merges the scalability of variational methods with SMC sampling. We would like to emphasize that our approach is completely complementary to many recent advances in variational inference that can be used to parametrize $q_\psi(\theta)$. For instance, one can consider more expressive variational families (Rezende and Mohamed, 2015; Kingma et al., 2016; Salimans et al., 2015; Maaløe et al., 2016; Ranganath et al., 2016). Similarly, our Bayesian approach naturally allows us to incorporate prior knowledge. For instance, one could place sparsity-inducing priors and impose corresponding variational approximations (Ingraham and Marks, 2017; Ghosh and Doshi-Velez, 2017; Louizos et al., 2017). Applying such variational approximations to more expressive autoregressive models would be an interesting avenue to explore in future work.

3.8 Appendix

3.8.1 SMC algorithm

3.8.2 Proof of Proposition 7

Consider an SMC algorithm with K particles targeting

$$\pi_\theta(x_{0:M}) \coloneqq \gamma(\theta, x_{0:M})/\gamma_M(\theta),$$

where $\gamma(\theta, x_{0:M}) = p(\theta, x_{0:M}, y_{0:M})$ is related to the posterior via $\pi(\theta, x_{0:M}) = \gamma(\theta, x_{0:M})/Z_M$. Z_M is a normalising constant independent of θ that represents the marginal likelihood $Z_M = p(y_{0:M})$. Furthermore, $\gamma_M(\theta) = \int \gamma(\theta, x_{0:M}) dx_{0:M} = p(\theta)p_\theta(y_{0:M})$. We denote the likelihood estimator of this SMC algorithm as $\tilde{Z}_M^{\theta,\phi}$. Following analogous arguments as in Andrieu et al. (2010), we have from the definition of the importance weights

$$\begin{aligned}
&\frac{\tilde{\pi}(\theta, x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l)}{q_{\phi,\psi}(\theta, x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l)} \\
=&\frac{\pi(\theta, x_{0:M}^l)K^{-(M+1)}}{q_\psi W_M^l M_0^\phi(x_0^{b_0^l}|y_0)\prod_{n=1}^M W_{n-1}^{b_{n-1}^l} M_n^\phi(x_n^{b_n^l}|y_n, x_{0:n-1}^{b_{n-1}^l})} \\
=&\frac{\pi(\theta, x_{0:M}^l)K^{-(M+1)}}{q_\psi(\theta) M_0^\phi(x_0^{b_0^l}|y_0)\prod_{n=1}^M M_n^\phi(x_n^{b_n^l}|y_n, x_{0:n-1}^{b_{n-1}^l})} \\
&\cdot \frac{\prod_{n=0}^M \left(\sum_{k=1}^K w_k(x_{0:M}^k)\right)}{\prod_{n=0}^M w_n(X_{0:M}^{b_n^l})} \\
=&\frac{\pi(\theta, x_{0:M}^l)\tilde{Z}_M^{\theta,\phi}}{q_\psi(\theta)\gamma(\theta, x_{0:M}^l)} \\
=&\frac{\tilde{Z}_M^{\theta,\phi}}{q_\psi(\theta)p(y_{0:M})}.
\end{aligned}$$

Note that $\tilde{Z}^{\theta,\phi} = p(\theta)\hat{Z}^{\theta,\phi}$, where $\hat{Z}^{\phi,\theta}$ is the SMC likelihood estimator in the main paper targeting a density proportional to $p_\theta(x_{0:M}, y_{0:M})$, whilst $\tilde{Z}^{\theta,\phi}$ targets a density proportional to $p(\theta)p_\theta(x_{0:M}, y_{0:M})$. Consequently,

$$\begin{aligned}
\mathrm{KL}(q_{\psi,\phi}||\tilde{\pi}) &= -\mathbb{E}_{q_{\psi,\phi}}\left[\log \frac{\tilde{Z}_M^{\theta,\phi}}{q_\psi(\theta)}\right] + \log p(y_{0:M}) \\
&= -\mathcal{L}(\psi, \phi) + \log p(y_{0:M}),
\end{aligned}$$

which concludes the proof.

Algorithm 1 Sampling from $q_\phi(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l|\theta)$ via an SMC sampler

Input: observations $y_{0:M}$, prior density p_θ, initial density $f_\theta(x_0)$, state transition density $f_\theta(x_{n+1}|x_n, y_n)$, observation density $g_\theta(y_n|x_n)$, proposal densities $M_n^\phi(x_n|y_n, x_{0:n-1})$, resampling criteria and static parameter value θ.

Output: $(X_{0:M}^{1:K}, A_{0:M-1}^{1:K}, L) \sim q_\phi(\cdot|\theta)$.

for $k = 1...K$ **do**

  Sample $X_0^k \sim M_0^\phi(\cdot|y_0)$.

  Set $\alpha_0(X_0^k) = \frac{g_\theta(y_0|X_0^k) f_\theta(X_0^k|y_0)}{M_0^\phi(X_0^k)}$.

  Set $w_0(X_{0:n}^k) = \alpha_0(X_{0:n}^k)/K$.

  Set $W_0^k \propto w_0(X_0^k)$.

end for

for $n = 2...M$ **do**

  if resampling criteria satisfied **then**

    for $k = 1...K$ **do**

      Sample $A_{n-1}^k \sim r(\cdot|W_{n-1})$.

    end for

    Set $W_{n-1} = (\frac{1}{K}, ..., \frac{1}{K})$.

  else

    Set $A_{n-1} = (1, ..., K)$.

  end if

  for $k = 1...K$ **do**

    Sample $X_n^k \sim M_n^\phi(\cdot|y_n, X_{0:n-1}^{A_{n-1}^k})$.

    Set $X_{0:n}^k = (X_{0:n-1}^k, X_n^k)$.

    Set $\alpha_n(X_{0:n}^k) = \frac{g_\theta(y_n|X_n^k) f_\theta(X_n^k|X_{n-1}^{A_{n-1}^k}, y_{n-1})}{M_n^\phi(X_n^k|y_n, X_{0:n-1}^{A_{n-1}^k})}$.

    Set $w_n(X_{0:n}^k) = W_{n-1}^k \alpha_n(X_{0:n}^k)$.

    Set $W_n^k \propto w_n(X_{0:n}^k)$.

  end for

  Sample $L = l$ with probability W_M^l

end for

3.8.3 Proof of Corollary 8

Observe that we can write

$$
\begin{aligned}
&\mathrm{KL}\left(q_{\psi,\phi}(\theta, x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l)||\tilde{\pi}(\theta, x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l)\right) \\
&= \mathbb{E}_{q_{\psi,\phi}(\theta, x_{0:M}^{l}, b_{0:M}^{l})}\Bigg[\mathbb{E}_{q_\phi(x_{0:M}^{\neg b_{0:M}^{l}}, a_{0:M-1}^{\neg b_{0:M-1}^{l}})|\theta, x_{0:M}^{l}, b_{0:M}^{l})}\Bigg[\\
&\qquad \log q_{\psi,\phi}(\theta, x_{0:M}^{l}, b_{0:M}^{l}) \\
&\qquad + \log q_\phi(x_{0:M}^{\neg b_{0:M}^{l}}, a_{0:M-1}^{\neg b_{0:M-1}^{l}}|\theta, x_{0:M}^{l}, b_{0:M}^{l})\Bigg] \\
&\qquad - \log \tilde{\pi}(\theta, x_{0:M}^{l}, b_{0:M}^{l}) \\
&\qquad - \log \tilde{\pi}_{\mathrm{CSMC}}(x_{0:M}^{\neg b_{0:M}^{l}}, a_{0:M-1}^{\neg b_{0:M-1}^{l}}|\theta, x_{0:M}^{l}, b_{0:M}^{l})\Bigg] \\
&= \mathrm{KL}(q_{\psi,\phi}(\theta, x_{0:M}^{l})||\pi(\theta, x_{0:M}^{l})) \\
&\quad + \mathbb{E}_{q_{\psi,\phi}(\theta, x_{0:M}^{l}, b_{0:M}^{l})}\Bigg[\\
&\qquad \mathrm{KL}(q_\phi(x_{0:M}^{\neg b_{0:M}^{l}}, a_{0:M-1}^{\neg b_{0:M-1}^{l}})|\theta, x_{0:M}^{l}, b_{0:M}^{l})\Big|\Big| \\
&\qquad\quad \tilde{\pi}_{\mathrm{CSMC}}(x_{0:M}^{\neg b_{0:M}^{l}}, a_{0:M-1}^{\neg b_{0:M-1}^{l}}|\theta, x_{0:M}^{l}, b_{0:M}^{l}))\Bigg].
\end{aligned}
$$

3.8.4 Proof of Proposition 9

We can write the extended target distribution as

$$
\tilde{\pi}(x_{0:M}^{1:K}, a_{0:M-1}^{1:K}, l) = \frac{\pi(\theta, x_{0:M}^{l})}{K^{M+1}} \tilde{\pi}_{\mathrm{CSMC}}(x_{0:M}^{\neg b_{0:M}^{l}}, a_{0:M-1}^{\neg b_{0:M-1}^{l}}|\theta, x_{0:M}^{l}, b_{0:M}^{l}).
$$

This follows from the fact that $x_{0:M}^{l} = (x_0^{b_0^l}, ..., x_M^{b_M^l})$ and that $b_{0:M}|x_{0:M}^{l}, \theta$ is uniformly distributed on $\{1, ..., K\}^{M+1}$. Hence, $\frac{\pi(\theta, x_{0:M}^{l})}{K^{-(M+1)}}$ is the marginal density $\tilde{\pi}(\theta, x_{0:M}^{l}, b_{0:M}^{l})$. Moreover, the variational approximation of the static parameter θ and latent states $x_{0:M}^{l}$, obtained as the marginal of the extended variational distribution, is given by,

following similar arguments as in Naesseth et al. (2018),

$$
\begin{aligned}
q_{\psi,\phi}(\theta, x^l_{0:M}) &= \frac{q_{\psi,\phi}(\theta, x^l_{0:M}, b^l_{0:M})}{q_{\psi,\phi}(b^l_{0:M}|\theta, x^l_{0:M})} \\
&= \frac{1}{K^{-(M+1)}} \int q_{\psi,\phi}(\theta, x^l_{0:M}, a^l_{0:M-1}, x^{\neg b^l}_{0:M}, a^{\neg b^l}_{0:M-1}) \\
&\qquad d(x^{\neg b^l}_{0:M}, a^{\neg b^l}_{0:M-1}) \\
&= K^{M+1} \int q_\psi(\theta) \frac{w^l_M(x^{b^l}_{0:M})}{\sum_{l'} w^{l'}_M(x^{l'}_{0:M})} \prod_{k=1}^{K} M^\phi_0(x^k_0|y_0) \\
&\qquad \cdot \prod_{n=1}^{M} \frac{w^k_{n-1}(x^{b^k_{n-1}}_{0:n})}{\sum_{l'} w^{l'}_{n-1}(x^{b^{l'}_{n-1}}_{0:n-1})} M^\phi_n(x^k_n|y_n, x^{a^k_{n-1}}_{0:n-1}) \\
&\qquad d(x^{\neg b^l}_{0:M}, a^{\neg b^l}_{0:M-1}) \\
&= \int q_\psi(\theta) \left(\prod_{n=1}^{M} \frac{\gamma_\theta(x^l_{0:n})}{\gamma_\theta(x^l_{0:n-1}) \sum_{l'} w^{l'}_n((x^{l'}_{0:n}))} \right) \\
&\quad \cdot \prod_{k:k\neq b^l_0} M^\phi_0(x^k_0|y_0) \\
&\quad \cdot \prod_{n=1}^{M} \prod_{k:k\neq b^l_n} W^k_{n-1} M^\phi_n(x^k_n|y_n, x^{a^k_{n-1}}_{n-1}) d(x^{\neg b^l}_{0:M}, a^{\neg b^l}_{0:M-1}) \\
&= q_\psi(\theta)\gamma_\theta(x^l_{0:M}) \\
&\quad \cdot \mathbb{E}_{\tilde{\pi}_{\mathrm{CSMC}}(x^{\neg b^l}_{0:M}, a^{\neg b^l}_{0:M-1}|\theta, x^l_{0:M})} \left[\left(\hat{Z}^{\theta,\phi}_M \right)^{-1} \right]
\end{aligned}
$$

3.8.5 Natural gradients

We have also experimented with optimizing the variational distribution over the static parameters using natural gradients (Amari, 1998; Martens, 2014) to take into account the Riemannian geometry of the approximating distributions, as explored previously for variational approximations, see for instance Honkela et al. (2010); Hoffman et al. (2013). Recall that we are optimizing over the space of probability distributions $q_\psi(\cdot)$ with parameter ψ, for which we can consider a possible metric given by the Fisher information

$$
I(\psi) = \mathbb{E}_{q_\psi(\theta)} \left[\nabla_\psi \log q_\psi(\theta) \left(\nabla_\psi \log q_\psi(\theta) \right)^T \right] = -\mathbb{E}_{q_\psi(\theta)} \left[H_{\log q_\psi}(\theta) \right],
$$

The last equation assumes that q_ψ is twice differentiable and $H_{\log q_\psi}(\theta) = \left(\frac{\partial^2 \log q_\psi(\theta)}{\partial \psi_i \partial \psi_j} \right)_{ij}$ denotes the Hessian. This induces an inner product $\langle \psi_1, \psi_2 \rangle_{\psi_0} =$

$\psi_1^T F(\psi_0)\psi_2$ locally around ψ_0, hence gives rise to a norm $||\cdot||_{\psi_0}$. The Fisher information matrix is connected to the KL divergence, since the distance in the induced metric is given approximately by the square root of twice the KL-divergence:

$$\mathrm{KL}(q_{\psi_1}||q_{\psi_2}) = \frac{1}{2}(\psi_2 - \psi_1)I(\psi_1)(\psi_2 - \psi_1)^T + O((\psi_2 - \psi_1)^3),$$

This follows from a second order Taylor expansion and from using the fact that $\mathbb{E}_{q_\psi}[\nabla_\psi \log q_\psi] = 0$. Recall that the natural gradient of a function $\mathcal{L}(\psi)$ is defined by

$$\tilde{\nabla}_\psi \mathcal{L}(\psi) = I(\psi)^{-1}\nabla_\psi \mathcal{L}(\psi)$$

and one can show that under mild assumptions (Martens, 2014),

$$\sqrt{2}\frac{\tilde{\nabla}_\psi \mathcal{L}(\psi)}{||\tilde{\nabla}_\psi \mathcal{L}(\psi)||_\psi} = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\mathrm{argmax}_{d:\mathrm{KL}(q_{\psi+d}||q_\psi)\leqslant\epsilon^2}\mathcal{L}(\psi + d).$$

Thus the natural gradient is the steepest ascent direction with the distance measured by the KL-divergence. The natural gradient ascent does not depend on the parametrisation of q_ψ as a consequence of the invariance of the KL-divergence with respect to reparametrisations.

For mean-field approximations, computing the inverse of the Fisher information matrix simplifies, as the Fisher information has a block-diagonal structure in this case. We consider both normal and log-normal factors. For a univariate Gaussian distribution $q_{\mu,v}$ with mean μ and variance $\exp(v)^2$ parametrized by the logarithm of the standard deviation v, we obtain $\nabla_{\mu,v}\log q_{\mu,v}(\theta) = (e^{-2v}(\theta-\mu), e^{-2v}(\theta-\mu)^2 - 1)^T$. Consequently,

$$I(\mu, v) = \begin{pmatrix} e^{-2v} & 0 \\ 0 & 2 \end{pmatrix}.$$

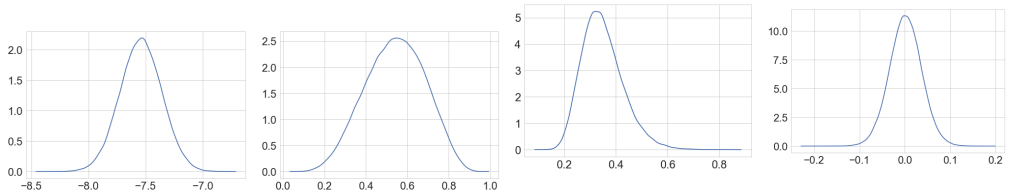
For a log-normal distribution $q_{a,b}(\theta)$, parametrized so that $\log\theta \sim \mathcal{N}(a, \exp(b)^2)$, we have $\nabla_{a,b}\log q_{a,b}(\theta) = (e^{-2b}(\log(\theta) - a), e^{-2b}(\log(\theta) - a)^2 - 1)^T$ and we arrive at the same form for the Fisher information

$$I(a, b) = \begin{pmatrix} e^{-2b} & 0 \\ 0 & 2 \end{pmatrix}.$$

3.8.6 Priors and variational approximations for the stochastic volatility model

Compared to Guarniero et al. (2017), we choose a different structure of Σ_x to guarantee its positive-definiteness, along with slightly different priors. We model

Σ_x with its unique Cholesky factorisation (Dellaportas and Pourahmadi, 2012), i.e. $\Sigma_x = LL^T$ with L a lower triangular matrix having positive values on its diagonal. We set Σ_x^0 as the stationary covariance of the latent state. Independent priors are placed for $a_i \sim U(0,1)$ and $\mu_i \sim \mathcal{N}(0,10)$ as well as $L_{ij} \sim \mathcal{N}(0,10)$, for $i < j$ and $\log L_{ii} \sim \mathcal{N}(0,10)$. We assume a mean-field variational approximation with normal factors for μ and for the entries of L below the diagonal and log-normal factors for its diagonal. Furthermore, a_i is assumed to be the sigmoid transform sigm: $x \mapsto 1/(1+e^{-x})$ of normally distributed variational factors. We initialized the mean of L with a diagonal matrix having entries 0.2 and the mean of μ_i with the logarithm of the standard deviation of the ith component of the time series. Densities of the variational approximation for parameters corresponding to the GBP exchange rate are given in Figure 3.4.



(a) Mean reversion level μ of the log volatility related to the Pound Sterling.

(b) Autoregressive coefficient a of the log volatility related to the Pound Sterling.

(c) Variance part of Σ_x for the error term of the log volatility related to the Pound Sterling.

(d) Covariance part of Σ_x for the error term of the log volatilities related to the Pound Sterling and Euro.

Figure 3.4: Density estimates for the parameters related to the Pound Sterling in the multivariate stochastic volatility model.

3.8.7 Hawkes point processes and state space models

In contrast to linear Hawkes processes (Hawkes, 1971a,b), we also allow for negative excitations, as explored previously for instance in Brémaud and Massoulié (1996); Bowsher et al. (2007); Duarte et al. (2016). The values of A^b and β^b are commonly assumed to be fixed through time, while time-varying μ have been considered in various settings. Stochastic time-varying excitations have been analysed in a probabilistic setting in Brémaud and Massoulié (2002); Dassios and Zhao (2011). Moreover, Ricci (2014) considered frequentist inference of the excitation model parameters from a matrix-valued categorical distribution, while Lee et al. (2016) performed MCMC

with excitations evolving according to an Ito process in the one-dimensional case. However, scalable Bayesian inference for non-linear stochastic Hawkes processes has been missing, with previous variational inference schemes (Linderman and Adams, 2015) having been restricted to linear Hawkes processes due to their resilience on the branching structure of linear Hawkes processes. SMC methods for shot-noise Cox processes has been considered in Whiteley et al. (2011); Martin et al. (2013) for on-line filtering and Finke et al. (2014) for static-parameter inference. While we expect such methods to scale poorly to models with many parameters and observations, we borrow their idea of describing the dynamics of the point process using piecewise-deterministic processes (Davis, 1984), which enables us to employ the proposed inference approach for discrete-time state space models.

More concretely, since Ξ_t^b follows deterministic dynamics between two events, we can write $\Xi_t^b = F_b(t, T_n, \Xi_{T_n}^b)$ for $t \in [T_n, T_{n+1})$ with the deterministic function $F_b(t, s, z^b) = e^{-\beta_b(t-s)} z^b$. Whenever an event of type C_n occurs at time T_n, the process Ξ^b jumps with size $\Delta\Xi_{T_n}^b = \beta_b A_n^b$. The process $Z_n^b = \Xi_{T_n}^b$, $n > 0$, satisfies $\Xi_t^b = F_b(t, T_n, Z_n^b)$ for $t \in [T_n, T_{n+1})$. Note that we scale each A_n^b with the diagonal matrix β_b. This ensures that the triggering kernel functions $s \mapsto \beta^b e^{-\beta^b s}$ have L_0 norm of one for any b.

3.8.8 Inference and predictions details for Hawkes process models

We place the following priors for the dynamics of A: For any $d \in \{1, ..., D\}$, $\alpha_d \sim \otimes_{i=1}^{DB} \mathcal{N}(0, 10)$ and consider mean-field variational approximations having the same forms. Furthermore, a priori, suppose that $\mu \sim \otimes_{i=1}^{D} \mathrm{Ga}(0.01, 0.01)$, $\mathrm{diag}(\sigma_d^2) \sim \otimes_{i=1}^{DB} \mathrm{Ga}(0.01, 0.01)$ and $\beta_b - \beta_{b-1} \sim \mathcal{LN}(0, 1)$, $b \in \{1, ..., B\}, \beta_0 = 0$, all with a log-normal variational approximation. Eventually, for the softmax scale parameter, a priori $\nu \sim U(0, 1)$ with a variational approximation as the sigmoid transform of a normal factor. The proposal function used is

$$M_\phi(a_n, z_n | a_{n-1}, z_{n-1}, t_{n+1}, c_{n+1}, t_n, c_n) = h_\phi(a_n | c_n) f_\theta(z_n | z_{n-1}, a_{n-1}, t_n, c_n), \quad (3.10)$$

with $h_\phi(a_n | c_n) = \mathcal{N}(\sum_d \tilde{\alpha}_d \delta_{c_n d}, \sum_d \tilde{\sigma}_d^2 \delta_{c_n d})$, $\tilde{\alpha_d} \in \mathbb{R}^{BD}$, $\tilde{\sigma_d}$ positive diagonal matrices and where f_θ describes the determinsitic decay of Z_n according to the prior transition density.

Let us also mention that the observation density contains a one-dimenisonal intractable integral. We apply Gaussian quadrature to evaluate the integral after transforming

the quadrature points to better cover the interval immediately after an event where the intensity function is varying more quickly, see Appendix 3.8.9 for details. We initialised the variational parameters so that the variational distribution of α is largely concentrated around the maximum likelihood estimates in a linear Hawkes model and the variational distribution of ν concentrated around 0. The values of β_b are commonly fixed in a maximum likelihood estimation setting to guarantee concavity of the log-likelihood. We have chosen $B = 5$ with $(\log \beta_1, \log(\beta_2 - \beta_1), ..., \log(\beta_5 - \beta_4)) = (-1, 1, 3, 5, 7)$ fixed. This allows event interactions across various time scales, ranging from $\beta_1 \approx 0.36$ to $\beta_5 \approx 1268$.

We have also split the events in subsamples of length $M = 100$ each and used the particles from the previous event-batch as the initial particles for the subsequent event-batch. We used $K = 20$ particles and performed optimisation with Adam (Kingma and Ba, 2014) and step size 0.0001. Similar performance was observed either using standard or natural gradients for the considered hyperparameters and reported results correspond to optimisation with standard gradients only.

Regarding inference for the benchmark models, maximum likelihood estimation for the linear Hawkes model was performed using the tick library (Bacry et al., 2017), with the fixed time scales $\beta_1, ..., \beta_5$ given above. Parameters for the non-linear Hawkes model were estimated using a limiting case of the generative model with very small σ_d, $K = 1$, and proposing the single particle according to the generative model, hence particularly with small variances σ_d. Concretely, we consider

$$f_\theta(a_n | a_{n-1}, z_{n-1}, c_n) = h_\phi(a_n | c_n) = \mathcal{N}\left(\sum_d \alpha_d \delta_{c_n d}, \sum_d \sigma_d \delta_{c_n d}\right),$$

recalling h_ϕ from the definition (3.10) of the proposal function and where for all $d \in \{1, ..., D\}$,

$$\sigma_d = \epsilon \begin{pmatrix} \beta_1^{-1} & & & & & & \\ & \ddots & & & & & \\ & & \beta_1^{-1} & & & & \\ & & & \ddots & & & \\ & & & & \beta_B^{-1} & & \\ & & & & & \ddots & \\ & & & & & & \beta_B^{-1} \end{pmatrix},$$

$\epsilon = 0.0001$. Stochastic gradient descent then yields point estimates over $\alpha_1, ..., \alpha_D$, decay parameters $\beta_1, ..., \beta_B$, softmax scale parameter ν and the background intensity parameter μ. Initial parameters have similary been set to the maximum likelihood estimates from the linear Hawkes model. We used Adam (Kingma and Ba, 2014) with step sizes 0.0001 and 0.0005, with the reported result corresponding to the best performing step size for the considered metric in Table 3.2.

For the prediction of the next mark c_{m+1} given the observations $t_{1:m}, c_{1:m}$, we can sample $\theta_1, ..., \theta_S \sim q_\psi(\theta)$ and run a particle filter that yields

$$\sum_{k=1}^{K} W_m^{k,s} \delta_{(Z_{0:m-1}^{k,s}, A_{0:m-1}^{k,s})}(z_{0:m-1}^s, a_{0:m-1}^s)$$

as an approximation of $p_{\theta_s}(z_{0:m-1}^s, \alpha_{0:m-1}^s | t_{1:m}, c_{1:m})$. Set

$$\hat{Z}_m^{b,k,s} = e^{-\beta_b (t_m - t_{m-1})} Z_{m-1}^{b,k,s} + A_m^{b,k,s},$$

with $A_m^{k,s} \sim f_{\theta_s}(\cdot | c_m)$ sampled from the prior transition density. We then sample 10 realisations

$$t_{m+1}^{k,s,j}, c_{m+1}^{k,s,j} \sim g_{\theta^s}(t_{m+1}, c_{m+1} | \hat{Z}_m^{k,s}), \quad j = 1, ..., 10,$$

using the standard thinning algorithm for point processes, see for instance Ogata (1981); Daley and Vere-Jones (2003); Bowsher et al. (2007). In the stochastic Hawkes process model, we have chosen $S = 4$ and $K = 20$. To account for a similar computational budget for the benchmark models, we sample $10 \cdot 4 \cdot 20$ event realisations in these cases instead. For predicting the next mark c_{m+1}, we use the sampled mark that occurred most often within $\{c_{m+1}^{k,s,j}\}_{k,s,j}$, where the count associated with $c_{m+1}^{k,s,j}$ is weighted by $W_m^{k,s}$. Notice that we do not condition on the observed t_{m+1} for predicting c_{m+1} and the dependence of $c_{m+1}^{k,s,j}$ on $t_{m+1}^{k,s,j}$ is accounted for via the thinning procedure. In the stochastic Hawkes process model, we have also run predictions using $K = 80$ particles, using the same model trained with $K = 20$ particles.

In order to show how the different models generalize if less data is available, we have trained the different models on either the first 100 or 1000 events of one day and evaluated how well the model performs on predicting the first 10000 events on another day. We have repeated this procedure for 10 days and found that a fully Bayesian treatment is beneficial when trained on 100 events. The fully variational

approach has an error rate of 65%, whilst the same stochastic Hawkes process model using a point estimate of the static parameters has an error rate of 70%. The two approaches yield similar results when trained on 1000 events with an error rate of below 50%, whereas a benchmark non-linear Hawkes model without latent intensity dynamics has an error rate of 65%. Although a fully Bayesian treatment might not be necessary if one imposes a parsimonious model for the evolution of the latent intensity, we hope that this example encourages further point process models that allow for online Bayesian updating as we feel that intensity excitations with latent dynamics have been underexplored for Hawkes process models.

3.8.9 Gaussian quadrature of the intensity function

We approximate the integral of the intensity function with Gaussian quadrature, see for instance Süli and Mayers (2003) for details. Let $p_1, ..., p_n$ be orthogonal polynomials in $L^2[a, b]$ equipped with the scalar product $\langle f, g\rangle = \int_a^b f(t)g(t)dt$, $f, g \in L^2[a, b]$ with p_k having degree k. Note that p_k can be constructed recursively by Gram-Schmidt-orthogonalization. Furthermore, let $t_1, ..., t_n$ be the roots of p_n and consider the Lagrange polynomials for $i = 1, ..., n$,

$$L_i(t) = \prod_{j=1, j\neq i}^{n} \frac{t - t_j}{t_i - t_j},$$

which satisfy $L_i(t_k) = \delta_{ik}, k = 1, ..., n$. Define

$$w_i = \int_a^b L_i(t)dt$$

as well as the Gaussian quadrature

$$I_n(f) = \sum_{i=1}^{n} w_i f(t_i).$$

Then $I_n(p) = \int_a^b p(t)dt$ for polynomials p of degree up to $2n - 1$. We are interested in evaluating $\int_{T_{min}}^{T_{max}} \lambda^i(t)dt$ for fixed T_{min} and T_{max}. Here, T_{max} is the time of the next event and we have fixed T_{min} to the previous event plus one microsecond. The lowest resolution of the event timestamps for the considered dataset is one microsecond. Assume there is a function g such that $\lambda(t) = g(e^t)$. We can write

$$\int_{T_{min}}^{T_{max}} \lambda(t)dt = \int_{\log T_{min}}^{\log T_{max}} g(e^{\tilde{t}})e^{\tilde{t}}d\tilde{t}.$$

This motivates the following change of variables that has also been considered in Bacry et al. (2016) for solving an integral equation involving the kernel function of a

Hawkes process. Suppose that $t_1 ... t_n$ are the quadrature point with weights $w_1, ... w_n$ on $[\log T_{min}, \log T_{max}]$. The transformed quadrature scheme is then

$$(\tilde{t}_n, \tilde{w}_n) = (e^{t_n}, w_n e^{t_n}).$$

We used 50 quadrature points in our experiments.

Chapter 4

Copula-like Variational Inference

4.1 Introduction

Variational inference (Jordan et al., 1999; Wainwright and Jordan, 2008; Blei et al., 2017) aims at performing Bayesian inference by approximating an intractable posterior density π with respect to the Lebesgue measure on $\mathbb{R}^d$, based on a family of distributions which can be easily sampled from. More precisely, this kind of inference posits some variational family Q of densities $(q_\xi)_{\xi\in\Xi}$ with respect to the Lebesgue measure and intends to find a good approximation $q_{\xi^\star}$ belonging to Q by minimizing the Kullback-Leibler (KL) with respect to π over Q, *i.e.* $\xi^\star \approx \arg\min_{\xi\in\Xi} \mathrm{KL}(q_\xi|\pi)$. Further, suppose that $\pi(x) = \mathrm{e}^{-U(x)}/\mathrm{Z}$ with $U\colon \mathbb{R}^d \to \mathbb{R}$ measurable and $\mathrm{Z} = \int_{\mathbb{R}^d} \mathrm{e}^{-U(x)}\mathrm{d}x < \infty$ is an unknown normalising constant. Then, for any $\xi \in \Xi$,

$$\mathrm{KL}(q_\xi|\pi) = -\int_{\mathbb{R}^d} q_\xi(x) \log \frac{\pi(x)}{q_\xi(x)}\mathrm{d}x = -\mathbb{E}_{q_\xi(x)}\left[-U(x) - \log q_\xi(x)\right] + \log \mathrm{Z}\,. \quad (4.1)$$

Since Z does not depend on q_ξ, minimizing $\xi \mapsto \mathrm{KL}(q_\xi|\pi)$ is equivalent to maximizing $\xi \mapsto \log \mathrm{Z} - \mathrm{KL}(q_\xi|\pi)$. A standard example is Bayesian inference over latent variables x having a prior density π_0 for a given likelihood function $L(y^{1:n}|x)$ and n observations $y^{1:n} = (y^1, \ldots, y^n)$. The target density is the posterior $p(x|y^{1:n})$ with $U(x) = -\log \pi_0(x) - \log L(y^{1:n}|x)$ and the objective that is commonly maximized,

$$\mathcal{L}(\xi) = \mathbb{E}_{q_\xi(x)}\left[\log \pi_0(x) + \log L(y^{1:n}|x) - \log q_\xi(x)\right] \quad (4.2)$$

is called a variational lower bound or ELBO. One of the main features of variational inference methods is their ability to be scaled to large datasets using stochastic approximation methods (Hoffman et al., 2013) and applied to non-conjugate models

by using Monte Carlo estimators of the gradient (Ranganath et al., 2014; Kingma and Welling, 2014; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014; Kucukelbir et al., 2017). However, the approximation quality hinges on the expressiveness of the distributions in Q and restrictive assumptions on the variational family that allow for efficient computations such as mean-field families, tend to be too restrictive to recover the target distribution. Constructing an approximation family Q that is both flexible to closely approximate the density of interest and at the same time computationally efficient has been an ongoing challenge. Much effort has been dedicated to find flexible and rich enough variational approximations, for instance by assuming a Gaussian approximation with different types of covariance matrices. For example, full-rank covariance matrices have been considered in Barber and Bishop (1998); Jaakkola and Jordan (1997); Titsias and Lázaro-Gredilla (2014) and low-rank perturbations of diagonal matrices in Barber and Bishop (1998); Miller et al. (2017); Ong et al. (2018); Mishkin et al. (2018). Furthermore, covariance matrices with a Kronecker structure have been proposed in Louizos and Welling (2016); Zhang et al. (2018). Besides, more complex variational families have been suggested: such as mixture models (Gershman et al., 2012; Guo et al., 2016; Miller et al., 2017; Locatello et al., 2018b,a), implicit models (Mescheder et al., 2017; Huszár, 2017; Tran et al., 2017; Yin and Zhou, 2018; Titsias and Ruiz, 2019), where the density of the variational distribution is intractable. Finally, variational inference based on normalizing flows has been developed in Rezende and Mohamed (2015); Kingma et al. (2016); Tomczak and Welling (2016); Louizos and Welling (2017); Berg et al. (2018). As a special case and motivated by Sklar's theorem (Sklar, 1959), variational inference based on families of copula densities and one-dimensional marginal distributions have been considered by Tran et al. (2015) where it is assumed that the copula is a vine copula (Bedford and Cooke, 2001) and by Han et al. (2016) where the copula is assumed to be a Gaussian copula together with non-parametric marginals using Bernstein polynomials. Recall that $c : [0,1]^d \to \mathbb{R}_+$ is a copula density if and only if its marginals are uniform on $[0,1]$, *i.e.* $\int_{[0,1]^{d-1}} c(u_1, \ldots, u_d) \mathrm{d}u_1 \cdots \mathrm{d}u_{i-1} \mathrm{d}u_{i+1} \cdots \mathrm{d}u_d = \mathbb{1}_{[0,1]}(u_i)$ for any $i \in \{1, \ldots, d\}$ and $u_i \in \mathbb{R}$. In the present work, we pursue these ideas but propose instead of using a family of copula densities, simply a family of densities $\{c_\theta : [0,1]^d \to \mathbb{R}_+\}_{\theta \in \Theta}$ on the hypercube $[0,1]^d$. This idea is motivated from the fact

that we are able to provide such a family which is both flexible and allow efficient computations.

The paper is organised as follow. In Section 4.2, we recall how one can sample more expressive distributions and compute their densities using a sequence of bijective and continuously differentiable transformations. In particular, we illustrate how to apply this idea in order to sample from a target density by first sampling a random variable U from its copula density c and then applying the marginal quantile function to each component of U. A new family of copula-like densities on the hypercube is constructed in Section 4.3 that allow for some flexibility in their dependence structure, while enjoying linear complexity in the dimension of the state space for generating samples and evaluating log-densities. A flexible variational distribution on $\mathbb{R}^d$ is introduced in Section 4.4 by sampling from such a copula-like density and then applying a sequence of transformations that include $\frac{1}{2}d\log d$ rotations over pairs of coordinates. We illustrate in Section 4.6 that for some target densities arising for instance as the posterior in a logistic regression model, the proposed density allows for a better approximation as measured by the KL-divergence compared to a Gaussian density. We conclude with applying the proposed methodology on Bayesian Neural Network models.

4.2 Variational Inference and Copulas

In order to obtain expressive variational distributions, the variational densities can be transformed through a sequence of invertible mappings, termed normalizing flows (Rezende et al., 2014). To be more specific, assume a series $\{\mathscr{T}_t : \mathbb{R}^d \to \mathbb{R}^d\}_{t=1}^T$ of C^1-diffeomorphisms[1] and a sample $X_0 \sim q_0$, where q_0 is a density function on $\mathbb{R}^d$. Then the random variable $X_T = \mathscr{T}_T \circ \mathscr{T}_{T-1} \circ \cdots \circ \mathscr{T}_1(X_0)$ has a density q_T that satisfies

$$\log q_T(x_T) = \log q_0(x) - \sum_{t=1}^{T} \log\det \left| \frac{\partial \mathscr{T}_t(x_t)}{\partial x_t} \right|, \tag{4.3}$$

with $x_t = \mathscr{T}_t \circ \mathscr{T}_{t-1} \circ \cdots \circ \mathscr{T}_1(x)$. To allow for scalable inferences with such densities, the transformations $\mathscr{T}_t$ must be chosen so that the determinant of their Jacobians can be computed efficiently. One possibility that satisfies this requirement is to choose volume-preserving flows that have a Jacobian-determinant of one. This can

[1] Recall that $f : \mathbb{R}^d \to \mathbb{R}^d$ is a C^1-diffeomorphisms if f is a bijection and f and f^{-1} are continuously differentiable.

be achieved by considering transformations $\mathscr{T}_t \colon x \mapsto H_t x$ where H_t is an orthogonal matrix as proposed in Tomczak and Welling (2016) using a Householder-projection matrix H_t.

An alternative construction of the same form can be used to construct a density using Sklar's theorem (Sklar, 1959; Moore and Spruill, 1975). It establishes that given a target density π on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, there exists a continuous function $C \colon [0,1]^d \to [0,1]$ and a probability space supporting a random variable $U = (U_1, \dots, U_d)$ valued in $[0,1]^d$, such that for any $x \in \mathbb{R}^d$, and $u \in [0,1]^d$,

$$\begin{aligned} &\mathbb{P}\left(U_1 \leqslant u_1, \cdots, U_d \leqslant u_d\right) = C(u_1, \cdots, u_d) \\ &\text{and} \quad \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_d} \pi(t) \mathrm{d}t = C(F_1(x_1), \dots, F_d(x_d)) \end{aligned} \tag{4.4}$$

where for any $i \in \{1, \dots, d\}$, F_i is the cumulative distribution function associated with π_i, so for any $x_i \in \mathbb{R}$, $F_i(x_i) = \int_{-\infty}^{x_i} \pi_i(t_i) \mathrm{d}t_i$ and π_i is the i^{th} marginal of π, so for any $x_i \in \mathbb{R}$, $\pi_i(x_i) = \int_{\mathbb{R}^{d-1}} \pi(x) \mathrm{d}x_1 \cdots \mathrm{d}x_{i-1} \mathrm{d}x_{i+1} \cdots \mathrm{d}x_d$. To illustrate how one can obtain such a continuous function C and random variable U, recall that π_i is assumed to be absolutely continuous with respect to the Lebesgue measure. Then for $(X_1, \dots, X_d) \sim \pi$, the random variable $U = \mathscr{G}^{-1}(X) = (F_1(X_1), \dots, F_d(X_d))$, where $\mathscr{G} \colon [0,1]^d \to \mathbb{R}^d$, with

$$\mathscr{G} \colon u \mapsto (F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)), \tag{4.5}$$

follows a law on the hypercube with uniform marginals. It can be readily shown that the cumulative distribution function C of U is continuous and satisfies (4.4). Note that taking the derivative of (4.4) yields

$$\pi(x) = c(F_1(x_1), \dots, F_d(x_d)) \prod_{i=1}^{d} \pi_i(x_i) \ ,$$

where $c(u_1, \dots, u_d) = \frac{\partial}{\partial u_1} \cdots \frac{\partial}{\partial u_d} C(u_1, \dots, u_d)$ is a copula density function by definition of C. One possibility to approximate a target density π is then to consider a parametric family of copula density functions $(c_\theta)_{\theta \in \Theta}$ for $\Theta \in \mathbb{R}^{p_c}$ and one parametric family of a d-dimensional vector of density functions $(f_1, \dots, f_d)_{\phi \in \Phi} \colon \mathbb{R}^d \to \mathbb{R}^d$ for $\Phi \subset \mathbb{R}^{p_f}$, and try to estimate $\theta \in \Theta$ and $\phi \in \Phi$ to get a good approximation of π via variational Bayesian methods. This idea was proposed by Han et al. (2016) and Tran

et al. (2015), where Gaussian and vine copulas were used, respectively. The main hurdle for using such family is their computational cost which can be prohibitive since the dimension of Θ is of order d^2. We remark that for latent Gaussian models with certain likelihood functions, a Gaussian variational approximation can scale linearly in the number of observations by using dual variables, see Opper and Archambeau (2009); Khan et al. (2013).

4.3 Copula-like Density

In this paper, we consider another approach which relies on a copula-like density function on $[0,1]^d$. Indeed, instead of an exact copula density function on $[0,1]^d$ with uniform marginals, we consider simply a density function on $[0,1]^d$ which allows to have a certain degree of freedom in the number of parameters we want to use.

We would like to remark that we have introduced a copula function C in probabilistic terms as a joint cumulative distribution function on the hypercube with uniform marginals. An equivalent definition can be given in analytical terms if $C : [0,1]^d \to [0,1]$ satisfies the following three conditions:

(i) C is grounded, *i.e.* $C(u_1, \ldots, u_d) = 0$ whenever $u_i = 0$ for at least one component $i \in \{1, \ldots, d\}$.

(ii) C is d-increasing, *i.e.* for all $u = (u_1, \ldots, u_d), v = (v_1, \ldots, v_d) \in [0,1]^d$ with $u_i < v_i$ for all $i \in \{1, \ldots, d\}$, it holds that

$$\sum_{(w_1, \ldots w_d) \in \times_{i=1}^d \{u_i, v_i\}} (-1)^{|\{i:\, w_i = u_i\}|} C(w_1, \ldots, w_d) \geqslant 0.$$

(iii) $C(1, \ldots, 1, u_i, 1, \ldots, 1) = u_i$ for all $i \in \{1, \ldots, d\}$ and $u_i \in [0,1]$.

It is clear that for any density c on the hypercube, the function

$$C(u_1, \ldots, u_d) = \int_0^{u_1} \cdots \int_0^{u_d} c(t_1, \ldots, t_d) \mathrm{d}t_1 \ldots \mathrm{d}t_d$$

is grounded and maps the hypercube onto $[0,1]$. It is also possible to show that for the random variable $(U_1, \ldots, U_d)$ with cumulative distribution C, one has the d-increasing property

$$0 \leqslant \mathbb{P}\left(\bigcap_{i=1}^d \{u_i < U_i < v_i\} \right) = \sum_{(w_1, \ldots w_d) \in \times_{i=1}^d \{u_i, v_i\}} (-1)^{|\{i:\, w_i = u_i\}|} C(w_1, \ldots, w_d),$$

see Mai and Scherer (2017) for details. However, property (iii) does not hold necessarily. The family of copula-like densities that we consider is given by

$$c_\theta(v_1,\dots,v_d) = \frac{\Gamma(\alpha^*)}{\mathrm{B}(a,b)}\left[\prod_{\ell=1}^{d}\left\{\frac{v_\ell^{\alpha_\ell-1}}{\Gamma(\alpha_\ell)}\right\}\right](v^*)^{-\alpha^*} \cdot\left(\max_{i\in\{1,\dots,d\}} v_i\right)^a\left[\left(1-\max_{i\in\{1,\dots,d\}} v_i\right)^{b-1}\right], \tag{4.6}$$

with the notation $v^* = \sum_{i=1}^d v_i$ and $\alpha^* = \sum_{i=1}^d \alpha_i$. Therefore $\theta = (a,b,(\alpha_i)_{i\in\{1,\dots,d\}}) \in (\mathbb{R}_+^* \times \mathbb{R}_+^* \times (\mathbb{R}_+^*)^d) = \Theta$. The following probabilistic construction is proven in Appendix 4.8.1 to allow for efficient sampling from the proposed copula-like density.

Proposition 11. *Let $\theta \in \Theta$ and suppose that*

1. $(W_1,\dots,W_d) \sim$ *Dirichlet*$(\alpha_1,\dots,\alpha_d)$;
2. $G \sim$ *Beta*(a,b);
3. $(V_1,\dots,V_d) = (GW_1/W^*,\dots,GW_d/W^*)$, *where* $W^* = \max_{i\in\{1,\dots,d\}} W_i$.

Then the distribution of $(V_1,\dots,V_d)$ has density with respect to the Lebesgue measure given by (4.6).

The proposed distribution builds up on Beta distributions, as they are the marginals of the Dirichlet distributed random variable $W \sim \mathrm{Dir}(\alpha)$, which is then multiplied with an independent random variable $G \sim \mathrm{Beta}(a,b)$. The resulting random variable $Y = WG$ follows a Beta-Liouville distribution, which allows to account for negative dependence, inherited from the Dirichlet distribution through a Beta stick-breaking construction, as well as positive dependence via a common Beta-factor. More precisely, one obtains

$$\mathrm{Cor}(Y_i,Y_j) = c_{ij}\left(\frac{\mathbb{E}[G^2]}{\alpha^\star+1} - \frac{\mathbb{E}[G]^2}{\alpha^\star}\right),$$

for some $c_{ij} > 0$ and $\alpha^\star = \sum_{k=1}^d \alpha_k$, cf. Fang (2017). Proposition 11 shows that one can transform the Beta-Liouville distribution living within the simplex to one that has support on the full hypercube, while also allowing for efficient sampling and log-density evaluations.

Now note that also $V^- = (1-V_1,\dots 1-V_d)$ is a sample on the hypercube if $V \sim c_\theta$, as is the convex combination $U = (U_1,\dots,U_d)$, where $U_i = \delta_i V_i + (1-\delta_i)(1-V_i)$ for

any $\delta \in [0,1]^d$. Put differently, we can write $U = \mathscr{H}(V)$, where

$$\mathscr{H} : v \mapsto (1-\delta)\,\mathrm{Id} + \{\mathrm{diag}(2\delta) - \mathrm{Id}\}v\ , \tag{4.7}$$

and Id is the identity operator. It is straightforward to see that $\mathscr{H}$ is a C^1-diffeomorphism for $\delta \in ([0,1]\backslash\{0.5\})^d$ from the hypercube into $I_1 \times \cdots \times I_d$, where $I_i = [\delta_i, 1-\delta_i]$ if $\delta_i \in [0, 0.5)$ and $I_i = [1-\delta_i, \delta_i]$ if $\delta_i \in (0.5, 1]$. Note that the Jacobian-determinant of $\mathscr{H}$ is efficiently computable and is simply equal to $|\prod_{i=1}^d (2\delta_i - 1)|$ for $\delta \in [0,1]^d$.

We suggest to take initially at random $\delta \in [0,1]^d$ for the transformation $\mathscr{H}$ such that

$$\mathbb{P}(\delta_i = \epsilon) = p \quad \text{and} \quad \mathbb{P}(\delta_i = 1-\epsilon) = 1-p \tag{4.8}$$

with $p, \epsilon \in (0,1)$. In our experiments, we set $\epsilon = 0.01$ and $p = 1/2$. We found that choosing a different (large enough) value of ϵ tends to yield no large difference, as this choice will get balanced by a different value of the standard deviation of the Gaussian marginal transformation. The motivation to consider $U = \mathscr{H}(V)$ with $V \sim c_\theta$ was first numerical stability since we need to compute quantile functions only on the interval $[\epsilon, 1-\epsilon]$ using this transformation. Second, this transformation can increase the flexibility of our proposed family. We found empirically that the components of $V \sim c_\theta$ tend to be non-negative in higher dimensions. However, using sometimes (more) the antithetic component of V by considering $U = \mathscr{H}(V)$, the transformed density can also describe negative dependencies in high dimensions. What comes to mind to obtain a flexible density is then to either optimize over the parameter δ parametrising the transformation $\mathscr{H}$ or considering δ as an auxiliary variable in the variational density, resorting to techniques developed for such hierarchical families, see for instance (Ranganath et al., 2016; Yin and Zhou, 2018; Titsias and Ruiz, 2019). However, this proved challenging in an initial attempt, since for $\delta_i = 0.5$, the transformation $\mathscr{H}$ becomes non-invertible, while restricting δ on say $\delta \in \{\epsilon, 1-\epsilon\}^d$, $\epsilon \approx 0$, seemed less easy to optimize. Consequently, we keep δ fixed after sampling it initially according to (4.8). A sensible choice was $p = 1/2$ since it leads to a balanced proportion of components of δ equal to ϵ and $1-\epsilon$. However, the sampled value of δ might not be optimal and we illustrate in the next section how the variational density can be made more flexible. We found empirically that not applying $\mathscr{H}$ or

setting for instance $\delta_i = 0.99$ for all $i \in \{1, \dots, d\}$ can lead to poor performance, see for example the Bayesian neural network in Section 4.6.4, where such an approach yields larger error rates than a mean-field model. To further illustrate the novel density on the hypercube, we show estimates of the empirical correlation matrix and the marginal distribution of the first component. We consider $d = 100$ with $a = 5$, $b = 1$ and $\alpha_i = c$ for any $i \in \{1, \dots, d\}$ for different choices of $c \in \{0.2, 1, 5\}$ in Figure 4.1. It can be observed that the marginal distribution is not uniform and that the correlations tend to be non-negative. We also show the empirical correlation matrix and the marginal distribution if one applies the transformation $\mathscr{H}$ to samples from c_θ with δ sampled according to (4.8) in Figure 4.2.

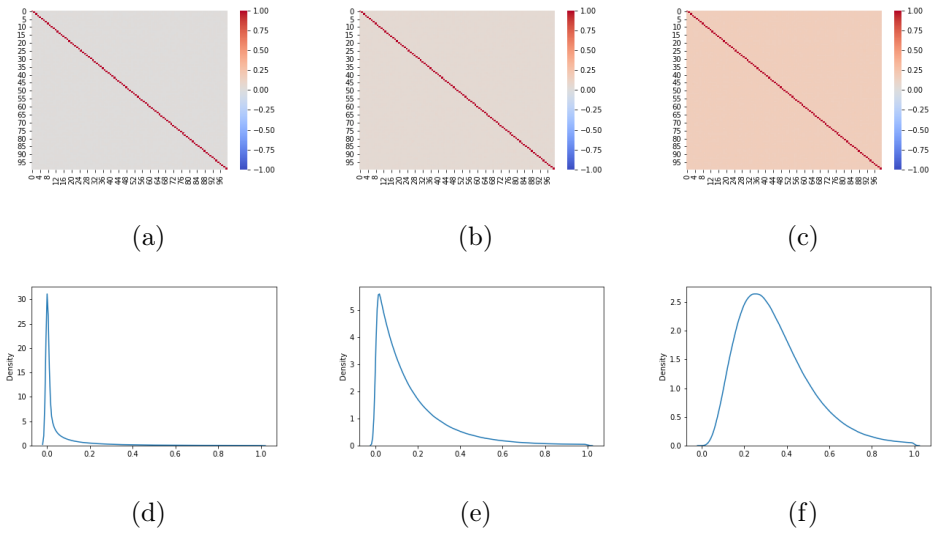


Figure 4.1: Empirical correlation matrix based on 50000 samples from c_θ with concentration parameter $c = 0.2$ in 4.1a, $c = 1$ in 4.1b and $c = 5$ in 4.1c. Kernel density estimate for the marginal distribution of the first component of c_θ with concentration parameter $c = 0.2$ in 4.1d, $c = 1$ in 4.1e and $c = 5$ in 4.1f.

4.4 Rotated Variational Density

We propose to apply rotations to the marginals in order to improve on the initial orientation that results from the sampled values of δ. Rotated copulas have been used before in low dimensions, see for instance Kosmidis and Karlis (2016), however, the set of orthogonal matrices has $d(d-1)/2$ free parameters. We reduce the number of free parameters by considering only rotation matrices $\mathcal{R}_d$ that are given as a product

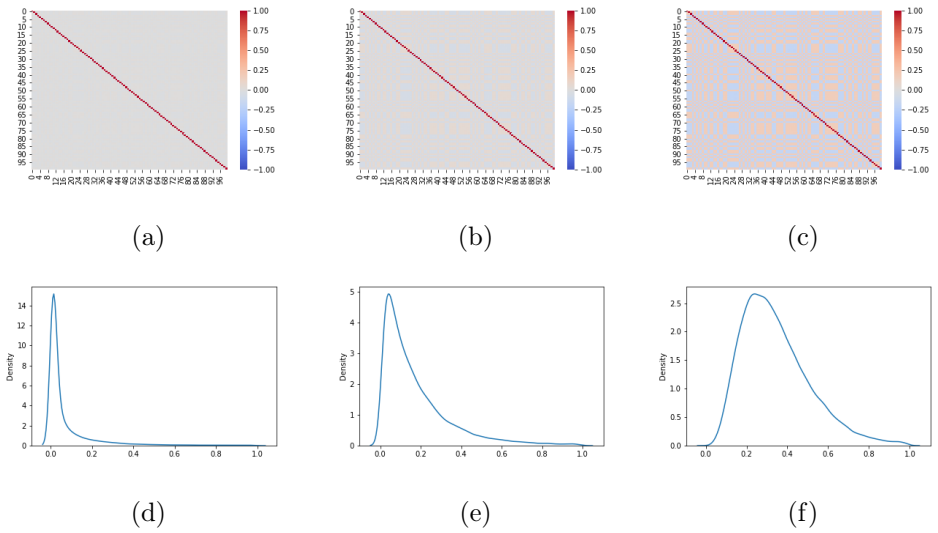


Figure 4.2: Empirical correlation matrix based on 50000 samples from c_θ after applying the transformation $\mathscr{H}$ with concentration parameter $c = 0.2$ in 4.2a, $c = 1$ in 4.2b and $c = 5$ in 4.2c. Kernel density estimate for the marginal distribution of the first component of c_θ with concentration parameter $c = 0.2$ in 4.2d, $c = 1$ in 4.2e and $c = 5$ in 4.2f.

of $d/2 \log d$ Givens rotations, following the FFT-style butterfly-architecture proposed in Genz (1998), see also Mathieu and LeCun (2014) and Munkhoeva et al. (2018) where such an architecture was used for approximating Hessians and kernel functions, respectively. Recall that a Givens rotation matrix (Golub and Van Loan, 2012) is a sparse matrix with one angle as its parameter that rotates two dimensions by this angle. If we assume for the moment that $d = 2^k$, $k \in \mathbb{N}^*$, then we consider k rotation matrices denoted $\mathcal{O}_1, \dots \mathcal{O}_k$ where for any $i \in \{1, \dots, k\}$, $\mathcal{O}_i$ contains $d/2$ independent rotations, *i.e.* is the product of $d/2$ independent Givens rotations. Givens rotations are arranged in a butterfly architecture that provides for a minimal number of rotations so that all coordinates can interact with one another in the rotation defined by $\mathcal{R}_d$. For illustration, consider the case $d = 4$, where the rotation matrix is fully described using $4 - 1$ parameters $\nu_1, \nu_2, \nu_3 \in \mathbb{R}$ by $\mathcal{R}_4 = \mathcal{O}_1\mathcal{O}_2$ with

$$\mathcal{O}_1\mathcal{O}_2 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & c_3 & -s_3 \\ 0 & 0 & s_3 & c_3 \end{bmatrix} \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ 0 & c_2 & 0 & -s_2 \\ s_2 & 0 & c_2 & 0 \\ 0 & s_2 & 0 & c_2 \end{bmatrix} = \begin{bmatrix} c_1c_2 & -s_1c_2 & -c_1s_2 & s_1s_2 \\ s_1c_2 & c_1c_2 & -s_1s_2 & -c_1s_s \\ c_3s_2 & -s_3s_2 & c_3c_2 & -s_3c_s \\ s_3s_2 & c_3s_2 & s_3c_2 & c_3c_2 \end{bmatrix},$$

where $c_i = \cos(\nu_i)$ and $s_i = \sin(\nu_i)$. We provide a precise recursive definition of $\mathcal{R}_d$ in Appendix 4.8.2 where we also describe the case where d is not a power of two. In general, we have a computational complexity of $\mathcal{O}(d \log d)$, due to the fact that $\mathcal{R}_d$ is a product of $\mathcal{O}(\log d)$ matrices each requiring $\mathcal{O}(d)$ operations. Moreover, note that $\mathcal{R}_d$ is parametrized by $d-1$ parameters $(\nu_i)_{i \in \{1 \dots d-1\}}$ and each $\mathcal{O}_i$ can be implemented as a sparse matrix, which implies a memory complexity of $\mathcal{O}(d)$. Furthermore, since $\mathcal{O}_i$ is orthonormal, we have $\mathcal{O}_i^{-1} = \mathcal{O}_i^\top$ and $|\det \mathcal{O}_i| = 1$.

To construct an expressive variational distribution, we consider as a base distribution q_0 the proposed copula-like density c_θ. We then apply the transformations $\mathscr{T}_1 = \mathscr{H}$ and $\mathscr{T}_2 = \mathscr{G}$. The operator $\mathscr{G}$ in (4.5) is defined via quantile functions of densities $f_1, \dots, f_d$, for which we choose Gaussian densities with parameter $\phi_f = (\mu_1, \dots, \mu_d, \sigma_1^2, \dots, \sigma_d^2) \in \mathbb{R}^d \times \mathbb{R}_+^d$. As a final transformation, we apply the volume-preserving operator

$$\mathscr{T}_3 \colon x \mapsto \mathcal{O}_1 \cdots \mathcal{O}_{\log d} x \tag{4.9}$$

that has parameter $\phi_{\mathcal{R}} = (\nu_1, \dots, \nu_{d-1}) \in \mathbb{R}^{d-1}$. Altogether, the parameter for the marginal-like densities that we optimize over is $\phi = (\phi_f, \phi_{\mathcal{R}})$ and simulation from the variational density boils down to the following algorithm.

Algorithm 2 Sampling from the rotated copula-like density.

1: Sample $(V_1, \dots, V_d) \sim c_\theta$ using Proposition 11.
2: Set $U = \mathscr{H}(V)$ where $\mathscr{H}$ is defined in (4.7).
3: Set $X' = \mathscr{G}(U)$, where $\mathscr{G}$ is defined in (4.5).
4: Set $X = \mathscr{T}_3(X')$, where $\mathscr{T}_3$ is defined in (4.9).

Note that we apply the rotations after we have transformed samples from the hypercube into $\mathbb{R}^d$, as the hypercube is not closed under Givens rotations. The variational density can then be evaluated using the normalizing flow formula (4.3). We optimize the variational lower bound $\mathcal{L}$ in (4.2) using reparametrization gradients, proposed by Kingma and Welling (2014); Rezende et al. (2014); Titsias and Lázaro-Gredilla (2014), but with an implicit reparametrization, cf. Figurnov et al. (2018), for Dirichlet and Beta distributions. Such reparametrized gradients for Dirichlet and Beta distributions are readily available for instance in tensorflow probability (Dillon et al., 2017). Using Monte Carlo samples of unbiased gradient estimates, one can

optimize the variational bound using some version of stochastic gradient descent. A more formal description is given in Appendix 4.8.3.

We would like to remark that such sparse rotations can be similarly applied to proper copulas. While there is no additional flexibility by rotating a full-rank Gaussian copula, applying such rotations to a Gaussian copula with a low-rank correlation yields a Gaussian distribution with a more flexible covariance structure if combined with Gaussian marginals. In our experiments, we therefore also compare variational families constructed by sampling $(V_1, \dots, V_d)$ from an independence copula in step 1 in Algorithm 2, *i.e.* V_i are independent and uniformly distributed on $[0, 1]$ for any $i \in \{1, \dots, d\}$, which results approximately in a Gaussian variational distribution if the effect of the transformation $\mathscr{H}$ is neglected. However, a more thorough analysis of such families is left for future work. Similarly, transformations different from the sparse rotations in step 4 in Algorithm 2 can be used in combination with a copula-like base density. Whilst we include a comparison with a simple Inverse Autoregressive Flow (Kingma et al., 2016) in our experiments, a more exhaustive study of non-linear transformations is beyond the scope of this work.

4.5 Related Work

Conceptually, our work is closely related to Tran et al. (2015); Han et al. (2016). It differs from Tran et al. (2015) in that it can be applied in high dimensions without having to search first for the most correlated variables using for instance a sequential tree selection algorithm (Dissmann et al., 2013). The approach in Han et al. (2016) considered a Gaussian dependence structure, but has only been considered in low-dimensional settings. On a more computational side, our approach is related to variational inference with normalizing flows (Rezende and Mohamed, 2015; Kingma et al., 2016; Tomczak and Welling, 2016; Louizos and Welling, 2017; Berg et al., 2018). In contrast to these works that introduce a parameter-free base distribution commonly in $\mathbb{R}^d$ as the latent state space, we also optimize over the parameters of the base distribution which is supported on the hypercube instead, although distributions supported for instance on the hypersphere as a state space have been considered in Davidson et al. (2018). Moreover, such approaches have been often used in the context of generative models using Variational Auto-Encoders (VAEs) (Kingma and Welling, 2014), yet it is in principle possible to apply the proposed variational copula-like

inference in an amortized fashion for VAEs.

A somewhat similar copula-like construction in the context of importance sampling has been proposed in Dellaportas and Tsionas (2018). However, sampling from this density requires a rejection step to ensure support on the hypercube, which would make optimization of the variational bound less straightforward. Lastly, Khaled and Kohn (2017) proposed a method to approximate copulas using mixture distributions, but these approximations have not been analysed neither in high dimensions nor in the context of variational inference.

4.6 Experiments

4.6.1 Bayesian Logistic Regression

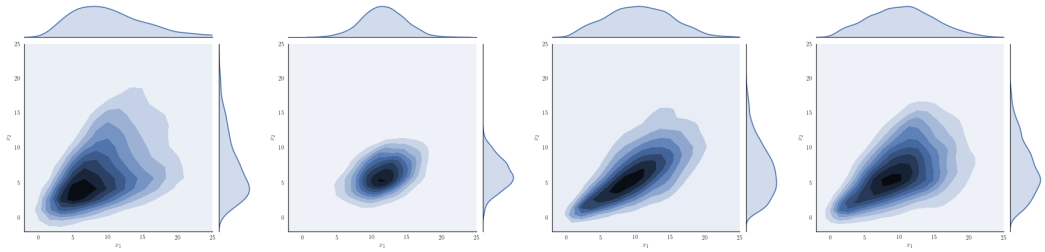
Consider the target distribution π on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ arising as the posterior of a d-dimensional logistic regression, assuming a Normal prior $\pi_0 = \mathcal{N}(0, \tau^{-1}I)$, $\tau = 0.01$, and likelihood function $L(y^i|x) = f(y^i x^\top \mathrm{a}^i)$, $f(z) = 1/(1+\mathrm{e}^{-z})$ with n observations $y^i \in \{-1, 1\}$ and fixed covariates $\mathrm{a}^i \in \mathbb{R}^d$ for $i \in \{1, \dots n\}$. We analyse a previously considered synthetic dataset where the posterior distribution is non-Gaussian, yet it can be well approximated with our copula-like construction. Concretely, we consider the synthetic dataset with $d = 2$ as in Murphy (2012), Section 8.4 and Khan et al. (2018) by generating 30 covariates $\mathrm{a} \in \mathbb{R}^2$ from a Gaussian $\mathcal{N}((1, 5)^\top, I)$ for instances in the first class, while we generate 30 covariates from $\mathcal{N}((-5, 1)^\top, 1.1^2 I)$ for instances in the second class. Samples from the target distribution using a Hamiltonian Monte Carlo (HMC) sampler (Duane et al., 1987; Neal, 2011) are shown in Figure 4.3a and one observes non-Gaussian marginals that are positively correlated with heavy right tails. Using a Gaussian variational approximation with either independent marginals or a full covariance matrix as shown in Figure 4.3b does not adequately approximate the target distribution. Our copula-like construction is able to approximate the target more closely, both without any rotations (Figure 4.3c) and with a rotation of the marginals (Figure 4.3d). This is also supported by the ELBO obtained for the different variational families given in Table 4.1.

4.6.2 Centred Horseshoe Priors

We illustrate our approach in a hierarchical Bayesian model that posits a priori a strong coupling of the latent parameters. As an example, we consider a Horseshoe

Table 4.1: Comparison of the ELBO between different variational families for the logistic regression experiment.

Variational family	ELBO
Mean-field Gaussian	-3.42
Full-covariance Gaussian	-2.97
Copula-like without rotations	-2.30
Copula-like with rotations	-2.19



(a) (b) (c) (d)

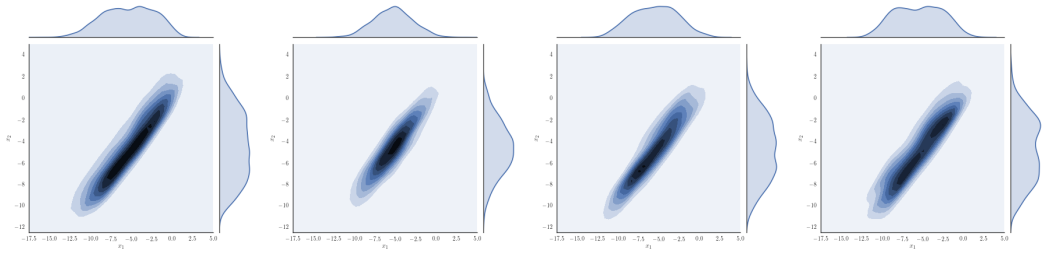
Figure 4.3: Target density for logistic regression using a HMC sampler in 4.3a with different variational approximations: Gaussian variational approximation with a full covariance matrix in 4.3b, copula-like variational approximation without any rotation in 4.3c and copula-like variational approximation with a rotation in 4.3d.

prior (Carvalho et al., 2010) that has been considered in the variational Gaussian copula framework in Han et al. (2016). To be more specific, consider the generative model $y|\lambda \sim \mathcal{N}(0, \lambda)$, with $\lambda \sim \mathcal{C}^+(0, 1)$, where $\mathcal{C}^+$ is a half-Cauchy distribution, *i.e.* $X \sim \mathcal{C}^+(0, b)$ has the density $p(x) \propto 1_{\mathbb{R}_+}(x)/(x^2+b^2)$. Note that we can represent a half-Cauchy distribution with Inverse Gamma and Gamma distributions using $X \sim \mathcal{C}^+(0, b) \iff X^2|Y \sim \mathcal{IG}(1/2, 1/Y); Y \sim \mathcal{IG}(1/2, 1/b^2)$, see Neville et al. (2014), with a rate parametrisation of the inverse gamma density $p(x) \propto 1_{\mathbb{R}_+}(x)x^{a-1}e^{-b/x}$ for $X \sim \mathcal{IG}(a, b)$. We revisit the toy model in Han et al. (2016) fixing $y = 0.01$. The model thus writes in a centred form as $\eta \sim \mathcal{G}(1/2, 1)$ and $\lambda|\eta \sim \mathcal{IG}(1/2, \eta)$. Following Han et al. (2016), we consider the posterior density on $\mathbb{R}^2$ of the log-transformed variables $(x_1, x_2) = (\log \eta_1, \log \lambda_1)$. In Figure 4.4, we show the approximate posterior distribution using a Gaussian family (4.4b) and a copula-like family (4.4c), together

with samples from a HMC sampler (4.4a). A copula-like density yields a higher ELBO, see Table 4.2. The experiments in Han et al. (2016) have shown that a Gaussian copula with a non-parametric mixture model fits the marginals more closely. To illustrate that it is possible to arrive at a more flexible variational family by using a mixture of copula-like densities, we have used a mixture of 3 copula-like densities in Figure 4.4d. Note that it is possible to accommodate multi-modal marginals using a Gaussian quantile transformation with a copula-like density. Eventually, the flexibility of the variational approximation can be increased using different complementary work. For instance, one could use the new density within a semi-implicit variational framework (Yin and Zhou, 2018) whose parameters are the output of a neural network conditional on some latent mixing variable.

Table 4.2: Comparison of the ELBO between different variational families for the centred horseshoe model.

Variational family	ELBO
Mean-field Gaussian	-1.24
Full-covariance Gaussian	-0.04
Copula-like	0.04
3-mixture copula-like	0.08



(a) (b) (c) (d)

Figure 4.4: Target density for the horseshoe model using a HMC sampler in 4.4a with different variational approximations: Gaussian variational approximation with a full covariance matrix in 4.4b, copula-like variational approximation including a rotation in 4.4c and a mixture of three copula-like densities with a one rotation and marginal-like density in 4.4d.

4.6.3 Bayesian Neural Networks with Normal Priors

We consider an L-hidden layer fully-connected neural network where each layer l, $1 \leqslant l \leqslant L+1$ has width d_l and is parametrised by a weight matrix $W^l \in \mathbb{R}^{d_{l-1} \times d_l}$ and bias vector $b^l \in \mathbb{R}^{d_l}$. Let $h^1 \in \mathbb{R}^{d_0}$ denote the input to the network and f be a point-wise non-linearity such as the ReLU function $f(a) = \max\{0, a\}$ and define the activations $a^l \in \mathbb{R}^{d_l}$ by $a^{l+1} = \sum_i h_i^l W_{i\cdot}^l + b^l$ for $l \geqslant 1$, and the post-activations as $h^l = f(a^l)$ for $l \geqslant 2$. We consider a regression likelihood function $L(\cdot|a^{L+2}, \sigma) = \mathcal{N}(a^{L+2}, \exp(0.5\sigma))$, and denote the concatenation of all parameters W, b and σ as x. We assume independent Normal priors for the entries of the weight matrix and bias vector with mean 0 and variance σ_0^2. Furthermore, we assume that $\log \sigma \sim \mathcal{N}(0, 16)$. Inference with the proposed variational family is applied on commonly considered UCI regression datasets, repeating the experimental set-up used in Gal and Ghahramani (2016). In particular, we use neural networks with ReLU activation functions and one hidden layer of size 50 for all datasets with the exception of the protein dataset that utilizes a hidden layer of size 100. We choose the hyper-parameter $\sigma_0^2 \in \{0.01, 0.1, 1., 10., 100.\}$ that performed best on a validation dataset in terms of its predictive log-likelihood. Optimization was performed using Adam (Kingma and Ba, 2014) with a learning rate of 0.002. We compare the predictive performance of a copula-like density c_θ and an independent copula as a base distribution in step 1 of Algorithm 2 and we apply different transformations $\mathscr{T}_3$ in step 4 of Algorithm 2: a) the proposed sparse rotation defined in (4.9); b) $\mathscr{T}_3 = \mathrm{Id}$; c) an affine autoregressive transformation $\mathscr{T}_3(x) = \{x - f_\mu(x)\}\exp(-f_\alpha(x))$, see Kingma et al. (2016), also known as an inverse autogressive flow (IAF). Here f_μ and f_α are autoregressive neural networks parametrized by μ and α satisfying $\frac{\partial f_\mu(x)_i}{\partial x_j} = \frac{\partial f_\alpha(x)_i}{\partial x_j} = 0$ for $i \leqslant j$ and which can be computed in a single forward pass by properly masking the weights in the neural networks (Germain et al., 2015). In our experiments, we use a one-hidden layer fully-connected network with width $d_1^{\mathrm{IAF}} = 50$ for f_μ and f_α. Note that for a d-dimensional target density, the size of the weight matrices are of order $d \cdot d_1^{\mathrm{IAF}}$, implying a higher complexity compared to the sparse rotation. We also compare the predictions against Bayes-by-Backprop (Blundell et al., 2015) using a mean-field model, with the results as reported in Mishkin et al. (2018) for a mean-field Bayes-by-Backprop and low-rank Gaussian approximation proposed

Table 4.3: Variational approximations with transformations and different base distributions. Test root mean-squared error for UCI regression datasets. Standard errors in parenthesis.

	Copula-like with rotation	Independent copula with rotation	Copula-like with IAF	Independent copula with IAF
Boston	3.43 (0.22)	3.51 (0.30)	3.21 (0.27)	3.61 (0.28)
Concrete	5.76 (0.14)	6.00 (0.13)	5.41 (0.10)	5.82 (0.11)
Energy	0.55 (0.01)	2.28 (0.11)	0.53 (0.02)	1.30 (0.10)
Kin8nm	**0.08 (0.00)**	**0.08 (0.00)**	**0.08 (0.00)**	**0.08 (0.00)**
Naval	**0.00 (0.00)**	**0.00 (0.00)**	**0.00 (0.00)**	**0.00 (0.00)**
Power	**4.02 (0.04)**	4.19 (0.04)	4.05 (0.04)	4.15 (0.04)
Wine	0.64 (0.01)	0.64 (0.01)	0.64 (0.01)	0.64 (0.01)
Yacht	1.35 (0.08)	1.38 (0.12)	**0.96 (0.06)**	1.25 (0.09)
Protein	**4.20 (0.01)**	4.51 (0.04)	4.31 (0.01)	4.51 (0.03)

therein called SLANG. Furthermore, we also report the results for Dropout inference (Gal and Ghahramani, 2016). The test root mean-squared errors are given in Table 4.3 and Table 4.4; the predictive test log-likelihood can be find in the Appendix 4.8.5 in Table 4.6 and Table 4.7. We can observe from Table 4.3 and Table 4.6 that using a copula-like base distribution instead of an independent copula improves the predictive performance, using either rotations or IAF as the final transformation. The same tables also indicate that for a given base distribution, the IAF tends to outperform the sparse rotations slightly. Table 4.4 and Table 4.7 suggest that copula-like densities without any transformation in the last step can be a competitive alternative to a benchmark mean-field or Gaussian approximation. Dropout tends to perform slightly better. However, note that Dropout with a Normal prior and a variational mixture distribution that includes a Dirac delta function as one component gives rise to a different objective, since the prior is not absolutely continuous with respect to the approximate posterior, see Hron et al. (2018).

4.6.4 Bayesian Neural Networks with Structured Priors

We illustrate our approach on a larger Bayesian neural network. To induce sparsity for the weights in the network, we consider a (regularised) Horseshoe prior (Piironen

Table 4.4: Copula-like variational approximation without rotations and benchmark results. Test root mean-squared error for UCI regression datasets. Standard errors in parenthesis.

	Copula-like without rotation	Bayes-by-Backprop (Mishkin et al., 2018)	SLANG (Mishkin et al., 2018)	Dropout (Mishkin et al., 2018)
Boston	3.22 (0.25)	3.43 (0.20)	3.21 (0.19)	**2.97 (0.19)**
Concrete	5.64 (0.14)	6.16 (0.13)	5.58 (0.12)	**5.23 (0.12)**
Energy	**0.52 (0.02)**	0.97 (0.09)	0.64 (0.04)	1.66 (0.04)
Kin8nm	**0.08 (0.00)**	**0.08 (0.00)**	**0.08 (0.00)**	0.10 (0.01)
Naval	**0.00 (0.00)**	**0.00 (0.00)**	**0.00 (0.00)**	0.01 (0.01)
Power	4.05 (0.04)	4.21 (0.03)	4.16 (0.04)	**4.02 (0.04)**
Wine	0.65 (0.01)	0.64 (0.01)	0.65 (0.01)	**0.62 (0.01)**
Yacht	1.23 (0.08)	1.13 (0.06)	1.08 (0.09)	1.11 (0.09)
Protein	4.31 (0.02)	NA	NA	4.27 (0.01)

et al., 2017) that has also been used increasingly as an alternative prior in Bayesian neural network to allow for sparse variational approximations, see Louizos et al. (2017); Ghosh and Doshi-Velez (2017) for mean-field models and Ghosh et al. (2018) for a structured Gaussian approximation. We consider again an L-hidden layer fully-connected neural network where we assume that the weight matrix $W^l \in \mathbb{R}^{d_{l-1} \times d_l}$ for any $l \in \{1, \ldots, L+1\}$ and any $i \in \{1, \ldots, d_{l-1}\}$ satisfies a priori

$$W^l_{i\cdot} | \lambda^l_i, \tau^l, c \sim \mathcal{N}(0, (\tau^l \tilde{\lambda}^l_i)^2 I) \propto \mathcal{N}(0, (\tau^l \lambda^l_i))^2 I) \mathcal{N}(0, c^2), \tag{4.10}$$

where $(\tilde{\lambda_i}^l)^2 = c^2 (\lambda^l_i)^2 / (c^2 + \tau^2 (\lambda^l_i)^2)$, $\lambda^l_i \sim \mathcal{C}^+(0,1)$, $\tau^l_i \sim \mathcal{C}^+(0, b_\tau)$ and $c^2 \sim \mathcal{IG}(\frac{\nu}{2}, \nu \frac{s^2}{2})$ for some hyper-parameters $b_\tau, \nu, s^2 > 0$. The vector $W^{(l)}_{i\cdot}$ represents all weights that interact with the i-th input neuron. The first Normal factor in (4.10) is a standard Horseshoe prior with a per layer global parameter τ^l that adapts to the overall sparsity in layer l and shrinks all weights in this layer to zero, due to the fact that $\mathcal{C}^+(0, b_\tau)$ allows for substantial mass near zero. The local shrinkage parameter λ^l_i allow for signals in the i-th input neuron because $\mathcal{C}^+(0,1)$ is heavy-tailed. However, this can leave large weights un-shrunk, and the second Normal factor in (4.10) induces a Student-$t_\nu(0, s^2)$ regularisation for weights far from zero, see Piironen et al. (2017) for details. We can rewrite the model in a non-centred form (Papaspiliopoulos et al., 2003), where the latent parameters are a priori independent, see also Louizos et al.

(2017); Ingraham and Marks (2017); Ghosh and Doshi-Velez (2017); Ghosh et al. (2018) for similar variational approximations. We write the model as $\eta_i^l \sim \mathcal{G}(1/2,1)$, $\hat{\lambda}_i^l \sim \mathcal{IG}(1/2,1)$, $\kappa^l \sim \mathcal{G}(1/2,1/b_\tau^2)$, $\hat{\tau}^l \sim \mathcal{IG}(1/2,1)$, $\beta_i^l \sim \mathcal{N}(0,I)$, $W_{i\cdot}^l = \tau^l \tilde{\lambda}_i^l \beta_i^l$, $\tau^l = \sqrt{\hat{\tau}^l \kappa^l}$, $\lambda_i^l = \sqrt{\hat{\lambda}_i^l \eta_i^l}$ and $(\tilde{\lambda}_i^l)^2 = c^2(\lambda_i^l)^2/(c^2 + (\tau^l)^2(\lambda_i^l)^2)$. The target density is the posterior of these variables, after applying a log-transformation if their prior is an (inverse) Gamma law.

We performed classification on MNIST using a 2-hidden layer fully-connected network where the hidden layers are of size 200 each. Further details about the algorithmic details are given in Appendix 4.8.4. Prediction errors for the variational families as considered in the preceding experiments are given in Table 4.5. We again find that a copula-like density outperforms the independent copula. Using a copula-like density without the rotation also performs competitively as long as one uses a balanced amount of its antithetic component via the transformation $\mathscr{H}$ with parameter δ; ignoring the transformation $\mathscr{H}$ or setting $\delta_i = 0.99$ for all $i \in \{1,\dots,d\}$ can limit the representative power of the variational family and can result in high predictive errors. The neural network function for the IAF considered here has two hidden layers of size 100×100. It can be seen that applying the rotations can be beneficial compared to the IAF for the copula-like density, whereas the two transformations perform similarly for the independent base distribution. We expect that more ad-hoc tricks can be used to adjust the rotations to some computational budget. For instance, one could include additional rotations for a group of latent variables such as those within one layer. Conversely, one could consider the series of sparse rotations $\mathcal{O}_1, \cdots, \mathcal{O}_k$, but with $2^k < d$, thereby allowing for rotations of the more adjacent latent variables only. Our experiment illustrates that the proposed approach can be used in high-dimensional structured Bayesian models without having to specify more model-specific dependency assumptions in the variatonal family. The prediction errors are in line with current work for fully connected networks using a Gaussian variational family with Normal priors, cf. Mishkin et al. (2018). Better predictive performance for a fully connected Bayesian network has been reported in Krueger et al. (2017) that use RealNVP (Dinh et al., 2016) as a normalising flows in a large network that is reparametrised using a weight normalization (Salimans and Kingma, 2016). It becomes scalable by opting to consider only variational inference over the Euclidean norm of $W_{i\cdot}^l$ and

Table 4.5: MNIST prediction errors.

Variational approximation with Horseshoe prior and size 200×200	Error Rate
Copula-like with rotations	1.70 %
Copula-like without rotations	1.78 %
Copula-like with IAF	2.04 %
Independent copula with IAF	2.88 %
Independent copula with rotations	2.90 %
Mean-field Gaussian	3.82 %
Copula-like without rotations and $\delta_i = 0.99$ for all $i \in \{1, \dots, d\}$	5.70 %

performing point estimation for the direction of the weight vector $W^l_{i.}/||W^l_{i.}||_2$. Such a parametrisation does not allow for a flexible dependence structure of the weights within one layer; and such a model architecture should be complementary to the proposed variational family in this work.

4.7 Conclusion

We have addressed the challenging problem of constructing a family of distributions that allows for some flexibility in its dependence structure, whilst also having a reasonable computational complexity of $\mathcal{O}(d \log d)$. Previously suggested variational families (Tran et al., 2015; Han et al., 2016) using copulas require either $\mathcal{O}(d^2)$ parameters to describe the full covariance matrix or all $d(d-1)/2$ pair copulas; or scale as $\mathcal{O}(dk)$ for some integer k by imposing some restrictions such as a low-rank Gaussian approximation of rank k or by truncating the number of levels in a vine copula using sequential tree selection so that the copula density becomes the product of kd pair copulas. It has been shown experimentally that it can constitute a useful replacement of a Gaussian approximation without requiring many algorithmic changes.

4.8 Appendix

4.8.1 Proof of Proposition 11

Proof. Let $f : \mathbb{R}^d \to \mathbb{R}_+$ be a positive and bounded function. We have by definition, using the expression of the density of the Dirichlet and Beta distributions, see Fang

(2017), and setting $u_d = 1 - \sum_{i=1}^{d-1} u_i$,

$$\mathbb{E}\left[f(V_1,\ldots,V_n)\right] = \frac{\Gamma(\alpha^\star)}{\mathrm{B}(a,b)} \int_{[0,1]^d} f\left\{gu_1/\max_{j\in\{1,\ldots,d\}} u_j, \ldots, gu_d/\max_{j\in\{1,\ldots,d\}} u_j\right\} \times g^{a-1}(1-g)^{b-1}\left\{\prod_{\ell=1}^{d} \frac{u_\ell^{\alpha_\ell-1}}{\Gamma(\alpha_\ell)}\right\} \mathrm{Leb}(g,u_1,\ldots,u_{d-1})$$

$$= \sum_{k=1}^{d} \frac{\Gamma(\alpha^\star)}{\mathrm{B}(a,b)} A_k \,, \tag{4.11}$$

where

$$A_k = \int_{[0,1]^d} \mathbb{1}\left\{u_k = \max_{j\in\{1,\ldots,d\}} u_j\right\} f\left\{gu_1/u_k, \ldots, gu_d/u_k\right\} \times g^{a-1}(1-g)^{b-1}\left\{\prod_{\ell=1}^{d} \frac{u_\ell^{\alpha_\ell-1}}{\Gamma(\alpha_\ell)}\right\} \mathrm{Leb}(g,u_1,\ldots,u_{d-1}) \,. \tag{4.12}$$

Then by symmetry, without loss of generality, we only need to consider A_1. Using the change of variable, $(g,u_1,u_2,\ldots,u_{d-1}) \mapsto (g,u_1,gu_2/u_1,\ldots,gu_{d-1}/u_1)$, which is a C^1-diffeomorphism from $\Delta_1 = \{(g,u_1,\ldots,u_{d-1}) \in [0,1]^d \,:\, u_1 = \max_{j\in\{1,\ldots,d\}} u_j\}$ to $\tilde{\Delta}_1 = \{(g,u_1,w_2,\ldots,w_{d-1}) \in [0,1]^d \,:\, \max_{j\in\{2,\ldots,d-1\}} w_j \leqslant g, g/u_1 - g - \sum_{j=2}^{d-1} w_j \leqslant g\}$, we get that

$$A_1 = \int_{\Delta_1} f\left\{g,\ldots,gu_d/u_1\right\} g^{a-1}(1-g)^{b-1}\left\{\prod_{\ell=1}^{d} \frac{u_\ell^{\alpha_\ell-1}}{\Gamma(\alpha_\ell)}\right\} \mathrm{Leb}(g,u_1,u_2,\ldots,u_{d-1})$$

$$= \int_{\tilde{\Delta}_1} f\left\{g,w_2,\ldots,w_{d-1},g/u_1-g-\sum_{i=2}^{d-1} w_i\right\} g^{a-1}(1-g)^{b-1} \times \left\{\prod_{\ell=2}^{d-2} \frac{(u_1 w_\ell/g)^{\alpha_\ell-1}}{\Gamma(\alpha_\ell)}\right\} \frac{u_1^{\alpha_1-1}}{\Gamma(\alpha_1)} \frac{(1-u_1-\sum_{i=2}^{d-1} u_1 w_i/g)^{\alpha_d-1}}{\Gamma(\alpha_d)} \frac{g^{d-2}}{u_1^{d-2}} \mathrm{Leb}(g,u_1,w_2,\ldots,w_{d-1})$$

$$= \int_{\tilde{\Delta}_1} f\left\{g,w_2,\ldots,w_{d-1},g/u_1-g-\sum_{i=2}^{d-1} w_i\right\} g^{a-1}(1-g)^{b-1} \times \left\{\prod_{\ell=2}^{d-2} \frac{w_\ell^{\alpha_\ell-1}}{\Gamma(\alpha_\ell)}\right\} \frac{u_1^{\alpha^\star-2}}{\Gamma(\alpha_1)} \frac{(g/u_1-g-\sum_{i=2}^{d-1} w_i)^{\alpha_d}}{\Gamma(\alpha_d)} g^{-\alpha^\star+\alpha_1+1} \mathrm{Leb}(g,u_1,w_2,\ldots,w_{d-1})$$

$$= \int_{\tilde{\Delta}_1} f\left\{g,w_2,\ldots,w_{d-1},g/u_1-g-\sum_{i=2}^{d-1} w_i\right\} g^{a-1}(1-g)^{b-1} \times \left\{\prod_{\ell=2}^{d-2} \frac{w_\ell^{\alpha_\ell-1}}{\Gamma(\alpha_\ell)}\right\} \frac{g^{\alpha_1-1}}{\Gamma(\alpha_1)} \frac{(g/u_1-g-\sum_{i=2}^{d-1} w_i)^{\alpha_d-1}}{\Gamma(\alpha_d)} (u_1/g)^{\alpha^\star-2} \mathrm{Leb}(g,u_1,w_2,\ldots,w_{d-1}) \,.$$

Now using the change of variable $(g,u_1,w_2,\ldots,w_{d-1}) \mapsto (g,g/u_1-\sum_{i=2}^{d-1} w_i,w_2,\ldots,w_{d-1}) = (g,w_d,\ldots,w_{d-1})$, which is a C^1-diffeomorphism from $\tilde{\Delta}_1$ to

$$\bar{\Delta}_1 = \{(g,w_d,w_2,\ldots,w_{d-1}) \,:\, \max_{j\in\{1,\ldots,d\}} w_j \leqslant g\} \,,$$

we obtain since $g/u_1 = g + \sum_{j=2}^{d} w_j$ that

$$A_1 = \int_{\bar{\Delta}_1} f(g, w_2, \ldots, w_{d-1}, w_d)) g^{a-1}(1-g)^{b-1} \times \left\{ \prod_{\ell=2}^{d} \frac{w_\ell^{\alpha_\ell - 1}}{\Gamma(\alpha_\ell)} \right\} \frac{g^{\alpha_1}}{\Gamma(\alpha_1)} \left\{ g + \sum_{j=1}^{d-1} w_j \right\}^{-\alpha^\star} \mathrm{Leb}(g, w_1, w_2, \ldots, w_{d-1}) \, .$$

Combining this result, (4.11) and (4.12) completes the proof. □

4.8.2 Butterfly rotation matrices

Suppose $d = 2^k$ for some $k \in \mathbb{N}$ and let $c_i = \cos \nu_i$ and $s_i = \sin \nu_i$. For $d = 1$, define $\mathcal{R}_1 = [1]$. Assume $\mathcal{R}_d$ has been defined. Then define

$$\mathcal{R}_{2d} = \begin{bmatrix} \mathcal{R}_d c_d & -\mathcal{R}_d s_d \\ \tilde{\mathcal{R}}_d s_d & \tilde{\mathcal{R}}_d c_d \end{bmatrix},$$

where $\tilde{\mathcal{R}}_d$ has the same form as $\mathcal{R}_d$ except that the c_i and s_i indices are all increased by d. So for instance

$$\mathcal{R}_2 = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix} \quad , \quad \tilde{\mathcal{R}}_2 = \begin{bmatrix} c_3 & -s_3 \\ s_3 & c_3 \end{bmatrix}.$$

Suppose now that d is not a power of 2 and let $k = \lceil \log d \rceil$. We construct $\mathcal{R}_d$ as a product of k factors $\mathcal{O}_1 \cdots \mathcal{O}_k$ as used in the construction of $\mathcal{R}_{2^k}$. For any $i \in \{1, \ldots k\}$, we then delete from $\mathcal{O}_i$ the last $2^k - d$ rows and columns. Then for every c_i in the remaining $d \times d$ matrix that is in the same column as a deleted s_i is replaced by 1. As an example, for $d = 5$, we have

$$\mathcal{R}_5 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 & 0 \\ s_1 & c_1 & 0 & 0 & 0 \\ 0 & 0 & c_3 & -s_3 & 0 \\ 0 & 0 & s_3 & c_3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & 0 & -s_2 & 0 & 0 \\ 0 & c_2 & 0 & -s_2 & 0 \\ s_2 & 0 & c_2 & 0 & 0 \\ 0 & s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_4 & 0 & 0 & 0 & -s_4 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ s_4 & 0 & 0 & 0 & c_4 \end{bmatrix}.$$

4.8.3 Optimization of the variational bound

Recall that for independent random variables $Z_i \sim \mathcal{G}(\alpha_i, 1)$, for $i \in \{1, \ldots d\}$, we have $\left(\frac{Z_1}{\sum_{j=1}^d Z_j}, \ldots \frac{Z_d}{\sum_{j=1}^d Z_j} \right) \sim \mathrm{Dirichlet}(\alpha_1, \ldots, \alpha_d)$, cf. Fang (2017). Similarly, for independent random variables $Z_{d+1} \sim \mathcal{G}(a, 1)$ and $Z_{d+2} \sim \mathcal{G}(b, 1)$, it holds that $\frac{Z_{d+1}}{Z_{d+1}+Z_{d+2}} \sim \mathrm{Beta}(a, b)$. Recall that the parameter of the rotated variational

family is $\xi = (\theta, \phi, \delta)$, where θ is the parameter of the copula-like base density, whereas $\phi = (\phi_f, \phi_{\mathcal{R}})$ denotes the parameters of the quantile transformation and the rotation, respectively. Furthermore, the parameter δ of the transformation $\mathscr{H}$ is kept fix. Using Proposition 11 and Algorithm 2 for some fixed δ, we can construct a function $(z, \phi) \mapsto f_{\phi,\delta}(z)$, $z = (z_1, \dots z_{d+2})$, that is almost everywhere continuously differentiable such that $f_{\phi,\delta}(Z_1, \dots Z_{d+2}) \sim q_\xi$, where q_ξ is the density of the proposed variational family with parameter $\xi = (\theta, \phi, \delta)$, that is the variational density q_ξ is the pushforward density of independent Gamma densities with parameter θ through the transport map $f_{\phi,\delta}$. Differentiability with respect to ϕ_f can be achieved by a continuous numerical approximation for the quantile function of a standard Gaussian and applying appropriate (re)normalisation. Furthermore, there exists an invertible standardization function $\mathcal{S}_\theta$ with $(z, \theta) \mapsto \mathcal{S}_\theta(z) = (\mathbb{P}(Z_1 \leqslant z_1), \dots, \mathbb{P}(Z_{d+2} \leqslant z_{d+2}))$ continuously differentiable such that $\mathcal{S}_\theta^{-1}(H)$ is equal to $(Z_1, \dots Z_{d+2})$ in distribution, where H is a $(d+2)$-dimensional vector of iid random variables with uniform marginals on $[0, 1]$. In particular, the distribution of H does not depend on ξ. The cumulative distribution function of Z_1 say at the point z_1 is the regularised incomplete Gamma function $\gamma(z_1, \alpha_1)$ that lacks an analytical expression though. However, one can apply automatic differentiation to a numerical method that approximates $\gamma(z_1, \alpha_1)$ yielding an approximation of $\frac{\partial \gamma(z_1,\alpha_1)}{\partial \alpha_1}$. Let us define

$$l(z, \phi, \delta) = \frac{\log L(y^{1:n} | f_{\phi,\delta}(z)) + \log \pi_0(f_{\phi,\delta}(z))}{\log q_\xi(f_{\phi,\delta}(z))}.$$

Then $\mathcal{L}(\xi) = \mathbb{E}[l(Z, \phi, \delta)] = \mathbb{E}\left[l(\mathcal{S}_\theta^{-1}(H), \phi, \delta)\right]$, where in the first expectation, the law of the random variable Z depends on θ. For a differentiable function $g \colon \mathbb{R}^n \to \mathbb{R}^m$, we denote by $\nabla_x g(x)$ the Jacobian of g, that is $\nabla_x g(x)_{ij} = \frac{\partial g_i(x)}{\partial x_j}$. Following the arguments in Figurnov et al. (2018), we obtain for the Jacobian of the variational bound

$$\begin{aligned}
\nabla_{\theta,\phi} \mathcal{L}(\xi) &= \mathbb{E}\left[\nabla_{\theta,\phi} l(\mathcal{S}_\theta^{-1}(H), \phi, \delta)\right] \\
&= \mathbb{E}\left[\nabla_z l(\mathcal{S}_\theta^{-1}(H), \phi, \delta) \nabla_{\theta,\phi} \mathcal{S}_\theta^{-1}(H) + \nabla_{\theta,\phi} l(\mathcal{S}_\theta^{-1}(H), \phi, \delta)\right] \\
&= \mathbb{E}\left[\nabla_z l(Z, \phi, \delta) \nabla_{\theta,\phi} Z + \nabla_{\theta,\phi} l(Z, \phi, \delta)\right], \qquad (4.13)
\end{aligned}$$

where $\nabla_\phi Z = 0$ and $\nabla_\theta Z = \nabla_\theta \mathcal{S}_\theta^{-1}(H)|_{H=\mathcal{S}_\theta(Z)}$ can be obtained by implicit differentiation of $S_\theta(Z) = H$ which results in $\nabla_\theta Z = -(\nabla_z \mathcal{S}_\theta(Z))^{-1} \nabla_\theta \mathcal{S}_\theta(Z)$. So for instance

$\frac{\partial Z_1}{\partial_{\alpha_1}} = -\frac{1}{p_{\alpha_1}(Z_1)}\frac{\partial \gamma(Z_1,\alpha_1)}{\partial_{\alpha_1}}$, with p_{α_1} being the density function of Z_1 and recalling that $\theta = (a, b, \alpha_1, \dots \alpha_d)$. We can thus optimize the variational bound using stochastic gradient descent with unbiased samples from (4.13). We remark that for instance in tensorflow probability (Dillon et al., 2017), such implicit gradients are used by default as long as one simulates from the copula-like density using Proposition 11, implements the density function c_θ from (4.6) and applies the bijective transformations according to Algorithm 2. In this case, optimization using the proposed density proceeds analogously as if one would use any reparametrisable variational family such as Gaussian distributions.

4.8.4 Additional details for Bayesian Neural Networks with Structured Priors

In the MNIST experiments, we train the network on 50000 training points out of 60000 and report the prediction error rates for the test set of 10000 images. We used a batch-size of 200 and used 4 Monte Carlo samples to compute the gradients during training and 100 Monte Carlo samples for the prediction on the test set. We used Adam with a learning rate in $\{0.0005, 0.0002\}$ for 20000 iterations. The hyper-parameter for the Horseshoe prior were $\nu = 4$, $s = 1$, so $c \sim \mathcal{IG}(2, 8)$, corresponding to a $t_4(0, 2^2)$ slab. Furthermore, for the global shrinkage factor, we have used $b_\tau \in \{0.1, 1\}$. The variational parameters of the copula-like density are restricted to be positive and we have defined them as the softmax: $x \mapsto \log(\exp(x) + 1)$ of unconstrained parameters, initialised so that $\text{softmax}^{-1}(\alpha_i) \sim \mathcal{N}(2, .01)$, $\text{softmax}^{-1}(a) = 15$ and $\text{softmax}^{-1}(b) = 2$. We have sampled δ according to (4.8) and initialised $\nu_i \sim \mathcal{U}(-0.2, 0.2)$ and the log-standard deviations of the marginal-like distribution as $\log \sigma_i = -3$. We aimed for an initial mean of 0 for β_i^l and of -3 for the log of the remaining variables. We therefore choose μ_i so that the quantile of an initial Monte Carlo estimate for the mean of V_i has the desired initial mean.

4.8.5 Additional results for Bayesian Neural Networks with Gaussian Priors

Table 4.6: Variational approximations with transformations and different base distributions. Test log-likelihood for UCI regression datasets. Standard errors in parenthesis.

	Copula-like with rotation	Independent copula with rotation	Copula-like with IAF	Independent copula with IAF
Boston	-2.85 (0.07)	-2.84 (0.09)	-2.78 (0.1)	-2.88 (0.09)
Concrete	-3.29 (0.03)	-3.30 (0.02)	-3.22 (0.02)	-3.26 (0.02)
Energy	-1.04 (0.02)	-2.34 (0.05)	**-0.93 (0.03)**	-1.78 (0.07)
Kin8nm	1.08 (0.01)	1.07 (0.01)	**1.10 (0.01)**	1.03 (0.01)
Naval	5.74 (0.05)	5.23 (0.05)	**5.97 (0.05)**	5.01 (0.05)
Power	-2.82 (0.01)	-2.85 (0.04)	-2.83 (0.04)	-2.85 (0.01)
Wine	-1.01 (0.01)	-1.02 (0.02)	-1.02 (0.02)	-1.02 (0.02)
Yacht	-2.01 (0.04)	-2.03 (0.06)	-1.69 (0.06)	-1.94 (0.07)
Protein	**-2.87 (0.00)**	-2.94 (0.00)	-2.90 (0.01)	-2.93 (0.01)

Table 4.7: Copula-like variational approximation without rotations and benchmark results. Test log-likelihood for UCI regression datasets. Standard errors in parenthesis.

	Copula-like without rotation	Bayes-by-Backprop (Mishkin et al., 2018)	SLANG (Mishkin et al., 2018)	Dropout (Mishkin et al., 2018)
Boston	-2.79 (0.08)	-2.66 (0.06)	-2.58 (0.05)	**-2.46 (0.06)**
Concrete	-3.25 (0.03)	-3.25 (0.02)	-3.13 (0.03)	**-3.04 (0.02)**
Energy	-1.00 (0.03)	-1.45 (0.02)	-1.12 (0.01)	-1.99 (0.02)
Kin8nm	1.09 (0.01)	1.07 (0.00)	1.06 (0.00)	0.95 (0.01)
Naval	5.45 (0.12)	4.61 (0.01)	4.76 (0.00)	3.80 (0.01)
Power	-2.83 (0.01)	-2.86 (0.01)	-2.84 (0.01)	**-2.80 (0.01)**
Wine	-1.02 (0.01)	-0.97 (0.01)	-0.97 (0.01)	**-0.93 (0.01)**
Yacht	-1.92 (0.06)	**-1.56 (0.03)**	-1.88 (0.01)	**-1.55 (0.03)**
Protein	-2.89 (0.01)	NA	NA	**-2.87 (0.01)**

Chapter 5

Gradient-based adaptive HMC

Hamiltonian Monte Carlo (HMC) is a popular Markov Chain Monte Carlo (MCMC) algorithm to sample from an unnormalized probability distribution. A leapfrog integrator is commonly used to implement HMC in practice, but its performance can be sensitive to the choice of mass matrix used therein. We develop a gradient-based algorithm that allows for the adaptation of the mass matrix by encouraging the leapfrog integrator to have high acceptance rates while also exploring all dimensions jointly. In contrast to previous work that adapt the hyperparameters of HMC using some form of expected squared jumping distance, the adaptation strategy suggested here aims to increase sampling efficiency by maximizing an approximation of the proposal entropy. We illustrate that using multiple gradients in the HMC proposal can be beneficial compared to a single gradient-step in Metropolis-adjusted Langevin proposals. Empirical evidence suggests that the adaptation method can outperform different versions of HMC schemes by adjusting the mass matrix to the geometry of the target distribution and by providing some control on the integration time.

5.1 Introduction

Consider the problem of sampling from a target density π on $\mathbb{R}^d$ of the form $\pi(q) \propto e^{-U(q)}$, with a *potential* energy $U\colon \mathbb{R}^d \to \mathbb{R}$ being twice continuously differentiable. HMC methods (Duane et al., 1987; Neal, 2011; Betancourt, 2017) sample from a *Boltzmann-Gibbs* distribution $\mu(q,p) \propto e^{-H(q,p)}$ on the phase-space $\mathbb{R}^{2d}$ based on the (separable) *Hamiltonian* function

$$H(q,p) = U(q) + K(p) \quad \text{with} \quad K(p) = \frac{1}{2} p^\top M^{-1} p.$$

The Hamiltonian represents the total energy that is split into a potential energy term U and a *kinetic* energy K which we assume is Gaussian for some symmetric positive definite *mass matrix* M. Suppose that $(q(t), p(t))_{t\in\mathbb{R}}$ evolve according to the differential equations

$$\frac{\mathrm{d}q(t)}{\mathrm{d}t} = \frac{\partial H(q(t),p(t))}{\partial p} = M^{-1}p(t) \quad \text{and} \quad \frac{\mathrm{d}p(t)}{\mathrm{d}t} = -\frac{\partial H(q(t),p(t))}{\partial q} = -\nabla U(q(t)). \tag{5.1}$$

Let $(\varphi_t)_{t\geqslant 0}$ denote the flow of the Hamiltonian system, that is for fixed t, φ_t maps each (q,p) to the solution of (5.1) that takes value (q,p) at time $t=0$. The exact HMC flow φ preserves volume and conserves the total energy *i.e.* $H\circ\varphi_t = H$. Consequently, the Boltzmann-Gibbs distribution μ is invariant under the Hamiltonian flow, that is $\mu(\varphi_t(E)) = \mu(E)$ for any Borel set $E\subset\mathbb{R}^{2d}$. Furthermore, the flow satisfies the generalized *reversibility* condition $\mathcal{F}\circ\varphi_t = \varphi_{-t}\circ\mathcal{F}$ with the flip operator $\mathcal{F}(q,p) = (q,-p)$. Put differently, the Hamiltonian dynamics go backward in time by negating the velocity. If an analytical expression for the exact flow were available, one could sample from μ using the invariant Markov chain that at state (q,p) first draws a new velocity $p'\sim\mathcal{N}(0,M)$ with the next state set to $\varphi_T(q,p')$ for some *integration time* $T>0$. Such a velocity refreshment is necessary as the HMC dynamics preserve the energy and so cannot be ergodic. However, the Hamiltonian flow cannot be computed exactly, except for very special potential functions. Numerical approximations to the exact solution of Hamiltonian's equations are thus routinely used, most commonly the *leapfrog* method, also known as (velocity) Verlet integrator (Hairer et al., 2003; Bou-Rabee and Sanz-Serna, 2018). For a step size $h>0$ and L steps, such an algorithm updates the previous state q_0 and a new velocity $p_0\sim\mathcal{N}(0,M)$ by setting, for $0\leqslant\ell\leqslant L-1$,

$$p_{\ell+\frac{1}{2}} = p_\ell - \frac{h}{2}\nabla U(q_\ell); \quad q_{\ell+1} = q_\ell + hM^{-1}p_{\ell+\frac{1}{2}}; \quad p_{\ell+1} = p_{\ell+\frac{1}{2}} - \frac{h}{2}\nabla U(q_{\ell+1}).$$

This scheme can be motivated by splitting the Hamiltonian wherein the kick mappings in the first and third step update only the momentum, while the drift mapping in the second step advances only the position q with constant speed. For $T=Lh$, the leapfrog integrator approximates $\varphi_T(q_0,p_0)$ by (q_L,p_L) while also preserving some geometric properties of φ, namely volume preservation and generalized reversibility. The leapfrog method is a second-order integrator, making an $\mathcal{O}(h^2)$ energy

error $H(q_L, p_L) - H(q_0, p_0)$. A μ-invariant Markov chain can be constructed using a Metropolis-Hastings acceptance step. More concretely, the proposed state (q_L, p_L) is accepted with the acceptance rate $a(q_0, p_0) = \min\{1, \exp[-(H(q_L, p_L) - H(q_0, p_0))]\}$, while the next state is set to $\mathcal{F}(q_0, p_0)$ in case of rejection, although the velocity flip is inconsequential for full refreshment strategies.

We want to explore here further the generalised speed measure introduced in Titsias and Dellaportas (2019) for adapting RWM or MALA that aim to achieve fast convergence by constructing proposals that (i) have a high average log-acceptance rate and (ii) have a high entropy. Whereas the entropy of the proposal in RWM or MALA algorithms can be evaluated efficiently, the multi-step nature of the HMC trajectories makes this computation less tractable. The recent work in Li et al. (2020) consider the same adaptation objective by learning a normalising flow that is inspired by a leapfrog proposal with a more tractable entropy by masking components in a leapfrog-style update via an affine coupling layer as used for RealNVPs (Dinh et al., 2016). Yu et al. (2019) sets the integration time by maximizing the proposal entropy for the exact HMC flow in Gaussian targets, while choosing the mass matrix to be the inverse of the sample covariance matrix.

5.2 Related work

The choice of the hyperparameters h, L and M can have a large impact on the efficiency of the sampler. For fixed L and M, a popular approach for adapting h is to target an acceptance rate of around 0.65 which is optimal for iid Gaussian targets in the limit $d \to \infty$ (Beskos et al., 2013) for a given integration time. HMC hyperparameters have been tuned using some form of *expected squared jumping distance* (ESJD) (Pasarica and Gelman, 2010), using for instance Bayesian optimization (Wang et al., 2013) or a gradient-based approach (Levy et al., 2018). A popular approach suggested in (Hoffman and Gelman, 2014) tunes L based on the ESJD by doubling L until the path makes a U-turn and retraces back towards the starting point, that is by stopping to increase L when the distance to the proposed state reaches a stationary point (Andrieu et al., 2020); see also Wu et al. (2018) for a variation and Park and Atchadé (2020) for a version using sequential proposals. The Riemann manifold HMC algorithm from Girolami and Calderhead (2011) has been suggested that uses a position dependent mass matrix $M(x)$ based on a non-separable Hamiltonian, but can

be computationally expensive, requiring $\mathcal{O}(d^3)$ operations in general. An alternative to choose M or more generally the kinetic energy K was proposed in Livingstone et al. (2019b) by analysing the behaviour of $x \mapsto \nabla K(\nabla U(x))$. Different pre-conditioning approaches have been compared for Gaussian targets in Langmore et al. (2019). A popular route has also been to first transform the target using tools from variational inference as in Hoffman et al. (2019) and then run a HMC sampler with unit mass matrix on the transformed density with a more favourable geometry.

A common setting to study the convergence of HMC assumes a log-concave target. In the case that U is m_1-strongly convex and m_2-smooth, Mangoubi and Smith (2017); Chen and Vempala (2019) analyse the ideal HMC algorithm with unit mass matrix where a higher condition number $\kappa = m_2/m_1$ implies slower mixing: The relaxation time, *i.e.* the inverse of the spectral gap, grows linear in κ, assuming the integration time is set to $T = \frac{1}{2\sqrt{m_2}}$. Chen et al. (2019b) establish non-asymptotic upper bounds on the mixing time using a leap-frog integrator where the step size h and the number L of steps depends explicitly on m_1 and m_2. Convergence guarantees are established using conductance profiles by obtaining (i) a high probability lower bound on the acceptance rate and (ii) an overlap bound, that is a lower bound on the KL-divergence between the HMC proposal densities at the starting positions q_0 and q_0', whenever q_0 is close to q_0'. While such bounds for controlling the mixing time might share some similarity with the generalised speed measure, they do not lend themselves easily to a gradient-based adaptation.

5.3 Entropy-based adaptation scheme

We derive a novel method to approximate the entropy of the proposed position after L leapfrog steps. Our approximation is based on the assumption that the Hessian of the target is locally constant around the mid-point of the HMC trajectory. This allows for a fast stochastic trace estimator of the marginal proposal entropy. We then develop a penalised loss function that can be minimized using stochastic gradient descent while sampling from the Markov chain in order to optimize a generalised speed measure.

5.3.1 Marginal proposal entropy

Suppose that $CC^\top = M^{-1}$, where C is defined by some parameters θ and can be a diagonal matrix, a full Cholesky factor, etc. Without loss of generality, the step size $h > 0$ can be fixed. We can reparameterise the momentum resampling step $p_0 \sim \mathcal{N}(0, M)$ by sampling $v \sim \mathcal{N}(0, \mathrm{I})$ and setting $p_0 = C^{-\top} v$. One can show by induction that the L-th step position q_L and momentum p_L of the leapfrog integrator can be represented as a function of v via

$$q_L = \mathcal{T}_L(v) = q_0 - \frac{Lh^2}{2} M^{-1} \nabla U(q_0) + LhCv - h^2 M^{-1} \Xi_L(v), \tag{5.2}$$

and

$$p_L = \mathcal{W}_L(v) = C^{-\top} v - \frac{h}{2} \left[\nabla U(q_0) + \nabla U \circ \mathcal{T}_L(v)\right] - h \sum_{i=1}^{n-1} \nabla U \circ \mathcal{T}_i(v)$$

where

$$\Xi_L(v) = \sum_{i=1}^{L-1} (L - i) \nabla U \circ \mathcal{T}_i(v), \tag{5.3}$$

see also Livingstone et al. (2019a); Durmus et al. (2017); Chen et al. (2019b) for the special case with an identity mass matrix. Observe that for $L = 1$ leap-frog steps, this reduces to a MALA proposal with preconditioning matrix M^{-1}.

Under regularity conditions, see for instance Durmus et al. (2017), the transformation $\mathcal{T}_L \colon \mathbb{R}^d \to \mathbb{R}^d$ is a C^1-diffeomorphism. With ν denoting the standard Gaussian density, the density r_L of the HMC proposal for the position q_L after L leapfrog steps is the pushforward density of ν via the map $\mathcal{T}_L$ so that[1]

$$\log r_L(\mathcal{T}_L(v)) = \log \nu(v) - \log |\det \mathsf{D}\mathcal{T}_L(v)|. \tag{5.4}$$

Observe that the density depends on the Jacobian of the transformation $\mathcal{T}_L \colon v \mapsto q_L$. We would like to avoid computing $\log |\det \mathsf{D}\mathcal{T}_L(v)|$ exactly. Define the residual transformation

$$\mathcal{S}_L \colon \mathbb{R}^d \to \mathbb{R}^d, \; v \mapsto \frac{1}{Lh} C^{-1} \mathcal{T}_L(v) - v. \tag{5.5}$$

Then $\mathsf{D}\mathcal{T}_L(v) = LhC(\mathrm{I} + \mathsf{D}\mathcal{S}_L(v))$ and consequently

$$\log |\det \mathsf{D}\mathcal{T}_L(v)| = d \log(Lh) + \log |\det C| + \log |\det(\mathrm{I} + \mathsf{D}\mathcal{S}_L(v))|. \tag{5.6}$$

[1] We denote the Jacobian matrix of a function $f \colon \mathbb{R}^d \to \mathbb{R}^d$ at the point x as $\mathsf{D}f(x)$.

Combining (5.4) and (5.6) yields the log-probability of the HMC proposal

$$\log r_L(\mathcal{T}_L(v)) = \log \nu(v) - d\log(Lh) - \log|\det C| - \log|\det(\mathrm{I}+\mathsf{D}\mathcal{S}_L(v))|. \quad (5.7)$$

Comparing the equations (5.2) and (5.5), one sees that $\mathcal{S}_L(v) = c - \frac{h}{L}C^\top \Xi_L(v)$ for some constant $c \in \mathbb{R}^d$ that depends on θ but is independent of v and consequently, $\mathsf{D}\mathcal{S}_L(v) = -\frac{h}{L}C^\top \mathsf{D}\Xi_L(v)$. We next show a recursive expression for $\mathsf{D}\mathcal{S}_L$ with a proof given in Appendix 5.6.2.

Lemma 12 (Jacobian representation)**.** *It holds that* $\mathsf{D}\mathcal{S}_1 = 0$ *and for any* $\ell \in \{2,\dots,L\}$, $v \in \mathbb{R}^d$,

$$\mathsf{D}\mathcal{S}_\ell(v) = -h^2 \sum_{i=1}^{\ell-1}(\ell-i)\frac{i}{\ell}C^\top \nabla^2 U\left(\mathcal{T}_i(v)\right) C\left(\mathrm{I}+\mathsf{D}\mathcal{S}_i(v)\right). \quad (5.8)$$

In particular, $\mathsf{D}\mathcal{S}_\ell(v)$ *is a symmetric matrix. Suppose further that* $L^2h^2 < \sup_{q\in\mathbb{R}^d}\frac{1}{4\left\|C^\top\nabla^2 U(q)C\right\|_p}$. *Then for any* $\ell \in \{1,\dots,L\}$ *and* $v \in \mathbb{R}^d$, *we have* $\|\mathsf{D}\mathcal{S}_\ell(v)\|_p < \frac{1}{8}$.

Consider for the moment a Gaussian target with potential function $U(q) = \frac{1}{2}(q-q_\star)^\top\Sigma^{-1}(q-q_\star)$ for $q_\star \in \mathbb{R}^d$ and positive definite $\Sigma \in \mathbb{R}^{d\times d}$. Then, due to (5.8), for any $q \in \mathbb{R}^d$, $v \in \mathbb{R}^d$,

$$\mathsf{D}\mathcal{S}_L(v) = -h^2\sum_{i=1}^{L-1}(L-i)\frac{i}{L}C^\top\Sigma^{-1}C(\mathrm{I}+\mathsf{D}\mathcal{S}_i(v)) = D_L + R_L(v),$$

where

$$D_L = -h^2 C^\top\Sigma^{-1}C\left(\sum_{i=1}^{L-1}(L-i)\frac{i}{L}\right) = -h^2\frac{L^2-1}{6}\,C^\top\Sigma^{-1}C$$

and a remainder term $R_L(v) = -h^2C^\top\Sigma^{-1}C\left(\sum_{i=1}^{L-1}(L-i)\frac{i}{L}\mathsf{D}\mathcal{S}_i(v)\right)$. From Lemma 12, we see that if $\left\|C^\top\Sigma^{-1}C\right\|_2 \leqslant \frac{h^2}{4L^2}$, then $\mathrm{I}+\mathsf{D}\mathcal{S}_L(v)$ and $-\mathsf{D}\mathcal{S}_L(v)$ are positive definite. Then R_L is also positive definite and $\log\det(\mathrm{I}+D_L) \leqslant \log|\det(\mathrm{I}+\mathsf{D}\mathcal{S}_L(v))|$ and we can maximize the lower bound instead. Notice that $R_2 = 0$ and one can include higher order terms $\mathcal{O}\left(\left[h^2C^\top\Sigma^{-1}C\right]^k\right)$, $k > 1$, in the approximation D_L, but we have not explored this systematically.

For an arbitrary potential energy U, we suggest to maximize

$$\mathcal{L}(\theta) = \log|\det(\mathrm{I}+D_L)| \quad \text{with} \quad D_L = -h^2\frac{L^2-1}{6}\,C^\top\nabla^2 U(q_{\lfloor L/2\rfloor})C \quad (5.9)$$

as an approximation of $\log|\det(\mathrm{I}+\mathsf{D}\mathcal{S}_L)|$. The intuition is that we assume that the target density can be approximated locally by a Gaussian one with precision matrix given by the Hessian of U at the mid-point $q_{\lfloor L/2\rfloor}$ of the trajectory.

Unlike in the Gaussian case, the matrix D_L depends on q_0 and v, although we make this dependence not explicit to simplify the notation. In general, however, neither $\mathsf{D}\mathcal{S}_L(v)$, $\mathrm{I}+\mathsf{D}\mathcal{S}_L(v)$, $R_L(v)$ nor D_L need to be positive definite. We want to optimize $\mathcal{L}(\theta)$ given in (5.9) even if we do not have access to the Hessian $\nabla^2 U$ explicitly, but only through Hessian-vector products $\nabla^2 U(q)w$ for some vector $w \in \mathbb{R}^d$. Vector-Jacobian products $\mathtt{vjp}(f,x,w) = w^\top \mathsf{D}f(x)$ for differentiable $f\colon \mathbb{R}^d \to \mathbb{R}^d$ can be computed efficiently via reverse-mode automatic differentiation, so that $\nabla^2 U(q)w = \mathtt{vjp}(\nabla U, q, w)^\top$ can be evaluated with complexity linear in d.

Suppose the multiplication with D_L is a contraction so that all eigenvalues of D_L have absolute values smaller than one. Then one can apply a Hutchinson stochastic trace estimator of $\log|\det(\mathrm{I}_d+D_{,L})|$ with a Taylor approximation, truncated and re-weighted using a Russian-roulette estimator (Lyne et al., 2015), see also Han et al. (2018); Behrmann et al. (2019); Chen et al. (2019a) for similar approaches in different settings. More concretely, let N be a positive random variable with support on $\mathbb{N}$ and let $p_k = \mathbb{P}(N \geqslant k)$. Then,

$$\mathcal{L}(\theta) = \log\det(\mathrm{I}+D_L) = \mathbb{E}_{N,\varepsilon}\left[\sum_{k=1}^{N} \frac{(-1)^{k+1}}{k p_k} \varepsilon^\top (D_L)^k \varepsilon\right], \tag{5.10}$$

where ε is drawn from a Rademacher distribution. While this yields an unbiased estimator for $\mathcal{L}(\theta)$ and its gradient as shown in Appendix 5.6.1.1 if D_L is contractive, it can be computationally expensive if N has a large mean or have a high variance if D_L has an eigenvalue that is close to 1 or -1, see (Lyne et al., 2015; Cornish et al., 2019). Since both the first order Gaussian approximation as well as the Russian Roulette estimator hinges on D_L having small absolute eigenvalues, we consider a constrained optimisation approach that penalises such large eigenvalues. For the random variable N that determines the truncation level in the Taylor series, we compute $b_N = (D_L)^N\varepsilon / \left\|(D_L)^N\varepsilon\right\|_2$ and $\mu_N = b_N^\top D_L b_N$. Note that this corresponds to applying N times the power iteration algorithm and with $|\lambda_1| > |\lambda_2| \geqslant \ldots \geqslant |\lambda_d|$ denoting the eigenvalues of the symmetric matrix D_L, almost surely $\mu_n \to \lambda_1$ for $n \to \infty$, see Golub and Van Loan (2012). For some $\delta \in (0,1)$, we choose some differentiable monotone increasing penalty function $h\colon \mathbb{R} \to \mathbb{R}$ such that $h(x) > 0$ for

$x > \delta$ and $h(x) = 0$ for $x \leqslant \delta$ and we add the term $\gamma h(|\mu_N|)$ for $\gamma > 0$ to the loss function that we introduce below, see Appendix 5.6.1.2 for an example of h.

5.3.2 Adaptation with a generalised speed measure

Extending the objective from Titsias and Dellaportas (2019) to adapt the HMC proposal, we aim to solve

$$\arg\min_{\theta} \int \int \pi(q_0)\nu(v)\Big[-\log a\left((q_0, v), (\mathcal{T}_L(v), \mathcal{W}_L(v))\right) + \beta \log r_L(\mathcal{T}_L(v))\Big] \mathrm{d}v\mathrm{d}q_0, \tag{5.11}$$

where $\mathcal{T}_L$, $\mathcal{W}_L$, r_L as well as the acceptance rate a depend on q_0 and the parameters θ we want to adapt. Also, the hyper-parameter $\beta > 0$ can be adapted online by increasing β if the acceptance rate is above a target acceptance rate $\alpha_\star$ and decreasing β otherwise. We choose $\alpha_\star = 0.67$, which is optimal for increasing d under independence assumptions (Beskos et al., 2013). One part of the objective constitutes minimizing the energy error $\Delta(q_0, v) = H(\mathcal{T}_L(v), \mathcal{W}_L(v)) - H(q_0, C^{-\top}v)$ that determines the log-acceptance rate via $\log a(q_0, C^{-\top}v) = \min\{0, -\Delta(q_0, v)\}$. Unbiased gradients of the energy error can be obtained without stopping any gradient calculations in the backward pass. However, we found that a multi-step extension of the biased fast MALA approximation from Titsias and Dellaportas (2019) tends to improve the adaptation by stopping gradients through ∇U as shown in Appendix 5.6.1.3.

Suppose that the current state of the Markov chain is q. We resample the momentum $v \sim \mathcal{N}(0, \mathrm{I})$ and aim to solve (5.11) by taking gradients of the penalised loss function

$$-\min\{0, -\Delta(q, v)\} - \beta\left(d \log h + \log|\det C| + \mathcal{L}(\theta) - \gamma h(|\mu_N|)\right),$$

as illustrated in Algorithm 3, which also shows how we update the hyperparameters β and γ. Algorithm 3 requires to choose learning rates ρ_θ, ρ_β and ρ_γ.

Instead of simulating a single Markov chain, the parameters can be updated using multiple parallel chains. We used 10 parallel chains throughout our experiments. The adaptive algorithm should also allow to make advantage of SIMD (single instruction, multiple data) operations, thereby benefiting from hardware accelerators such as GPUs. Pseudo-code for simulating from the adaptive chain can found in Algorithm 3.

Algorithm 3 Sample the next state q' and adapt β, γ and θ.

Sample velocity $v \sim \mathcal{N}(0, \mathrm{I})$ and set $p = C^{-\top} v$.

Apply integrator `LF` to obtain $(q_\ell, p_\ell, \nabla U(q_\ell))_{0 \leqslant \ell \leqslant L} = \mathtt{LF}(q, p)$.

Stop gradients $\nabla U(q_\ell) = \mathtt{stop\_grad}(\nabla U(q_\ell))$ for $0 \leqslant \ell \leqslant L$.

Compute $\Xi_L(v)$ using (5.3).

Compute $\Delta(q_0, v)$ using (5.13) and set $a = \min\{1, e^{-\Delta(q_0, v)}\}$.

Compute $\bar{\eta}_N, y = \textsc{Rademacher}()$.

Set $\mathcal{L}(\theta) = \mathtt{stop\_grad}(y)^\top D_L \varepsilon$.

Set $b_N = \mathtt{stop\_grad}\left(\frac{\bar{\eta}_N}{\|\bar{\eta}_N\|_2^2}\right)$ and $\mu_N = b_N^\top D_L b_N$.

$\mathcal{E}(\theta) = -\min\{0, -\Delta(q_0, v)\} - \beta\left(d \log h + \log|\det C| + \mathcal{L}(\theta) - \gamma h(|\mu_N|)\right)$.

Adapt $\theta \leftarrow \theta - \rho_\theta \nabla_\theta \mathcal{E}(\theta)$.

Adapt $\beta \leftarrow \Pi_\beta\left[\beta(1 + \rho_\beta(a - \alpha_\star)\right]$. $\#\Pi_\beta$ projects onto a compact set; default value $[10^{-2}, 10^2]$.

Adapt $\gamma \leftarrow \Pi_\gamma\left[\gamma + \rho_\gamma h(|\mu_N|)\right]$. $\#\Pi_\gamma$ projects onto a compact set; default value $[10^3, 10^5]$.

Sample $u \sim \mathcal{U}(0, 1)$ and set $q' = \mathbf{1}_{\{u \leqslant a\}}\, q_L + \mathbf{1}_{\{u > a\}}\, q$.

function $D_L(w)$:

  $\#D_L(w) = D_L w$ computes Hessian-vector products efficiently

  $z = \mathtt{vjp}(\nabla U, \mathtt{stop\_grad}(q_{\lfloor L/2 \rfloor}), Cw)^\top$

  return $-h^2 \frac{L^2 - 1}{6}\, C^\top z$

end function

function RADEMACHER:

  Sample Rademacher random variable ε and truncation level N.

  Initialise $y \leftarrow 0$ and $\bar{\eta}_0 = \varepsilon$.

  for $k = 1 \ldots N$ **do**

    #Apply a spectral normalisation for stability if D_L is not a contraction; $\delta' \in (0, 1)$.

    Set $\bar{\eta}_k = D_L \bar{\eta}_{k-1} \cdot \min\left\{1, \delta' \|\bar{\eta}_{k-1}\|_2 / \|D_L \bar{\eta}_{k-1}\|_2\right\}$ and $y \leftarrow y + \frac{(-1)^k}{p_k} \bar{\eta}_k$.

  end for

  return $\bar{\eta}_N, y$

end function

5.4 Numerical experiments

This section illustrates the mixing performance of the entropy-based sampler for a variety of target densities. First, we consider Gaussian targets either in high dimensions or with a high condition number. Our results confirm (i) that HMC scales better than MALA for high-dimensional Gaussian targets and (ii) that the adaptation scheme learns a mass matrix that is adjusted to the geometry of the target. Next, we apply the novel adaptation scheme to Bayesian logistic regression models and find that it often outperforms NUTS, except in a few data sets where some components might mix less efficiently. We also compare the entropy-based adaptation with Riemann-Manifold based samplers for a Log-Gaussian Cox point process models. We find that both schemes mix similarly, which indicates that the gradient-based adaptation scheme can learn a suitable mass matrix without having access to the expected Fisher information matrix. Then, we consider a high-dimensional stochastic volatility model where the entropy-based scheme performs favourably compared to alternatives and illustrate that efficient sparsity assumptions can be accommodated when learning the mass matrix. Finally, we show in a toy example how the suggested approach might be modified to sample from highly non-convex potentials. Our implementation builds up on tensorflow probability (Lao et al., 2020) with some target densities taken from Sountsov et al. (2020).

5.4.1 Gaussian targets

Anisotropic Gaussian distributions. We consider sampling from a multivariate Gaussian distribution $\mathcal{N}(0, \Sigma)$ with strictly convex potential $U(q) = \frac{1}{2} q^\top \Sigma^{-1} q$ for different covariance matrices Σ. For $c > 0$, assume a covariance matrix given by $\Sigma_{ij} = \delta_{ij} \exp\left(c(i-1)/(d-1) \log 10\right)$. We set (i) $c = 3$ and $d \in \{10^3, 10^4\}$ and (ii) $c = 6$ and $d = 100$, as considered in Sohl-Dickstein et al. (2014). The eigenvalues of the covariance matrix are thus distributed between 1 to 100 in setting (i), while they vary from 1 and 10^6 in setting (ii). The preconditioning factor C is assumed to be diagonal. We adapt the sampler for 4×10^4 steps in case (i) and for 10^5 steps in case (ii). We compute the minimum effective sample size (minESS) of all functions $q \mapsto q_i$ over $i \in \{1, \dots, d\}$ as shown in Figure 5.1 for $d = 10^3$ with leapfrog steps ranging from $L = 1$ to 10. We also compared it with a NUTS implementation in tensorflow probability (Lao et al., 2020) with a default maximum tree depth of 10 and

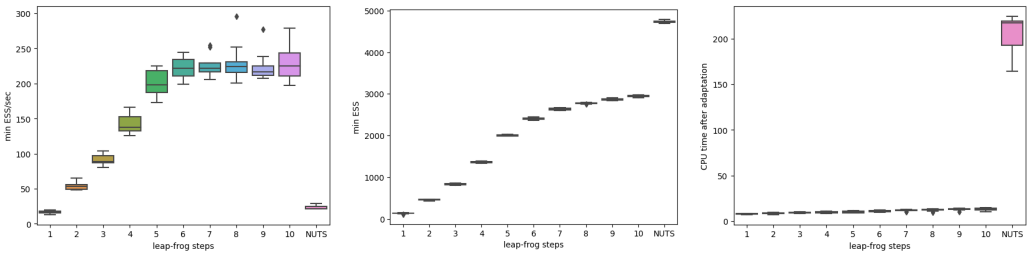
step sizes adapted using dual averaging. HMC performs well in terms of minESS/sec for $L > 1$ and the mass matrix adapts to the target covariance with the condition number of $C^\top \Sigma^{-1} C$ becoming relatively close to 1. The entropy objective also yields acceptance rates approaching 1 for increasing leap-frog steps and multiplication with D_L becomes a contraction. Results for $d = 10^4$ can be found in Figure 5.7 in Appendix 5.6.4.1 which indicate that as the dimension increases, using a more leap-frog steps becomes more advantageous. For the case (ii) of a very ill-conditioned target, results from Figure 5.8 in Appendix 5.6.4.2 also show that $L > 1$ is beneficial and that NUTS without some adaptation of the mass matrix mixes less efficiently. We want to emphasize that for fixed L, high acceptance rates for HMC need not be disadvantageous. This is illustrated in Figure 5.10 in Appendix 5.6.4.4 for a Gaussian target $\mathcal{N}(0, \mathrm{I})$ in dimension $d = 10$, where tuning just the step-size to achieve a target acceptance rate can lead to slow mixing for some L, because the proposal can make a U-turn.

Correlated Gaussian distribution. We sample from a 51-dimensional Gaussian target with covariance matrix given by the squared exponential kernel plus small white noise as in Titsias and Dellaportas (2019), with $k(x_i, x_j) = \exp\left(-\frac{1}{2}(x_i - x_j)^2/0.4^2\right) + .01\delta_{ij}$ on the regular grid $[0, 4]$. We consider a general Cholesky factor C. The adaptation is performed over 10^5 steps. Results over 10 runs are shown in Figure 5.9 in Appendix 5.6.4.3 which indicates that HMC with moderate L around 4 performs best.

5.4.2 Logistic regression

Consider a Bayesian logistic regression model with n data points $y_i \in \{0, 1\}$ and d-dimensional covariates $x_i \in \mathbb{R}^d$ for $i \in \{1, \ldots, n\}$. Assuming a Gaussian prior with covariance matrix Σ_0 implies a potential function $U(q) = \sum_{i=1}^n \left[-y_i x_i^\top q + \log\left(1 + e^{x_i^\top q}\right)\right] + \frac{1}{2} q^\top \Sigma_0^{-1} q$.

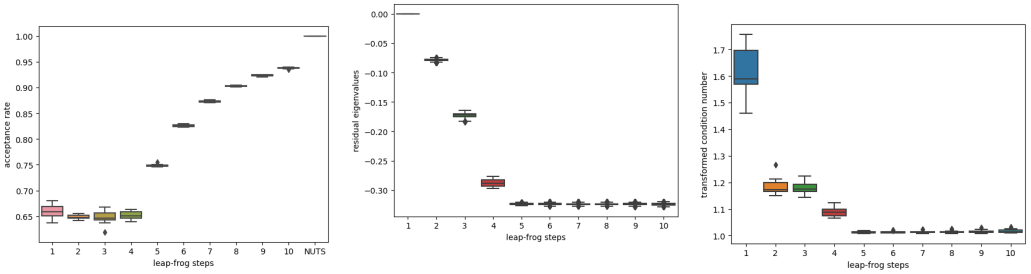
We considered six datasets (Australian Credit, Heart, Pima Indian, Ripley, German Credit and Caravan) that are commonly used for benchmarking inference methods, cf. Chopin et al. (2017). The state dimension ranges from $d = 3$ to $d = 87$. We choose $\Sigma_0 = \mathrm{I}$ and parametrize C via a Cholesky matrix. We adapt over 10^4 steps. HMC with a moderate number of leap-frog steps tends to perform better for four out of six data sets, with subpar performance for the Australian and Caravan data in terms of minESS/sec, albeit with higher median ESS/sec across dimensions



(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).

(b) Minimum effective sample size of $q \mapsto q_i$. NUTS adapted over 1/10-th of the steps.

(c) CPU time (after adaptation). NUTS adapted over 1/10-th of the steps.



(d) Average acceptance rate (after adaptation).

(e) Estimates of the eigenvalues of D_L using power iteration.

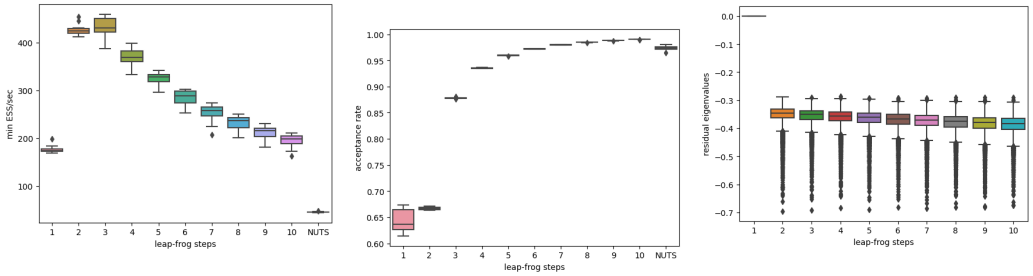
(f) Condition number of transformed Hessian $C^\top \Sigma^{-1} C$.

Figure 5.1: Anisotropic Gaussian target ($d = 1000$).

or for computing the potential energy function. The adaptive HMC algorithm tends to perform well if D_L is contractive during iterations of the Markov chain such as for the German Credit data set as shown in Figure 5.2. If this is not the case as for the Caravan data in Figure 5.3, the adapted HMC algorithm can perform worse than MALA or NUTS. More detailed diagnostics for all data sets can be found in Appendix 5.6.5.

5.4.3 Log-Gaussian Cox Point Process

We consider inference in a log-Gaussian Cox process model. This is an ideal setting for Riemann-Manifold MALA and HMC (Girolami and Calderhead, 2011), since a constant metric tensor is used therein that does not depend on the position, so the complexity is no longer cubic but only quadratic in the dimension d of the target. Consider an area on $[0, 1]^2$ that is discretized into grid locations (i, j), for $i, j = 1, \ldots, n$. The observations y_{ij} are Poisson distributed and conditionally independent given a latent intensity process $\lambda = \{\lambda\}_{ij}$ with means $\lambda_{ij} = m \exp(x_{ij})$ for $m = n^{-2}$ and a latent vector x

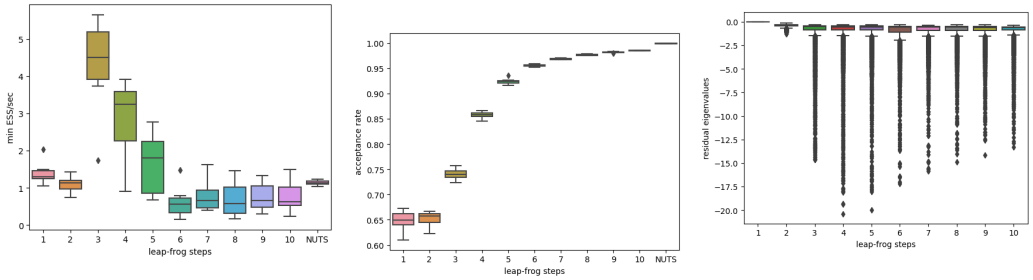


(a) Minimum effective sample size per second (after adaptation).

(b) Average acceptance rate (after adaptation).

(c) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.2: Bayesian logistic regression for German credit data set ($d = 25$).



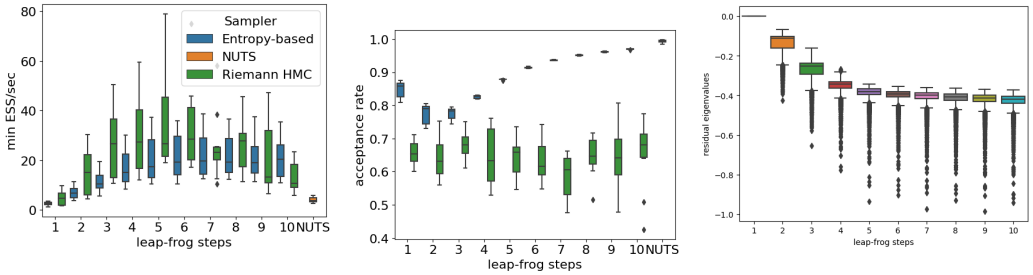
(a) Minimum effective sample size per second (after adaptation).

(b) Average acceptance rate (after adaptation).

(c) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.3: Bayesian logistic regression for caravan data set ($d = 87$).

drawn from a Gaussian process with constant mean function μ and covariance function $\Sigma_{(i,j),(i',j')} = \sigma_x^2 \exp\{-\sqrt{(i-i')^2 + (j-j')^2}/(n\beta)\}$. The target density is thus proportional to $p(y, x) \propto \prod_{i,j}^{n \times n} \exp\left[y_{ij} x_{ij} - m \exp(x_{ij})\right] \exp\left[-(x - \mu\mathbf{1})^\top \Sigma^{-1} (x - \mu\mathbf{1})/2\right]$. For the Riemann-Manifold based samplers, the preconditioning matrix is $M = \Lambda + \Sigma^{-1}$ where Λ is a diagonal matrix with diagonal elements $\{m \exp(\mu + \Sigma_{ii})\}_i$ and we adapt the step size using dual averaging. We generate simulated data for $d \in \{64, 256\}$. We adapt for 2000 steps using a Cholesky factor C. Figure 5.17 in Appendix 5.6.7 illustrates that the gradient-based adaptation can achieve a higher minESS/sec score for $d = 64$ with high acceptance rates for higher leap-frog steps. In dimension $d = 256$, the Riemann-Manifold samplers perform slightly better in terms of minESS/sec, see Figure 5.4 and Figure 5.18 for a comparison of the inverse mass matrices.



(a) Minimum effective sample size per second (after adaptation).

(b) Average acceptance rate (after adaptation).

(c) Estimates of the eigenvalues of D_L using power iteration.

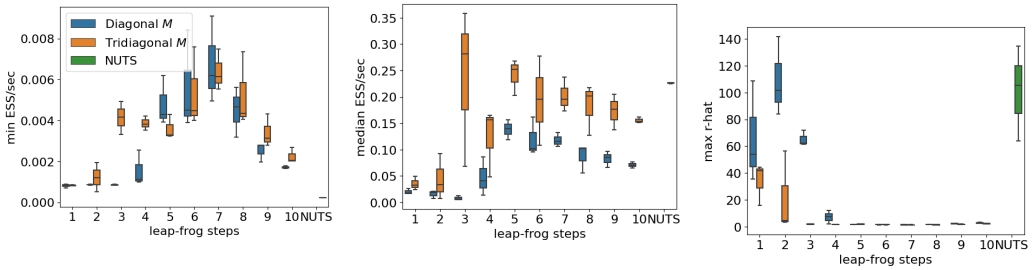
Figure 5.4: Cox process in dimension $d = 256$.

5.4.4 Stochastic volatility model

We consider a stochastic volatility model (Kim et al., 1998; Jacquier et al., 2002) that has been used with minor variations for adapting HMC (Girolami and Calderhead, 2011; Hoffman and Gelman, 2014; Wu et al., 2018). Assume that the latent log-volatilities follow an autoregressive AR(1) process so that $h_1 \sim \mathcal{N}(0, \sigma^2/(1-\phi^2))$ and for $t \in \{1, \dots, T-1\}$, $h_{t+1} = \phi h_t + \eta_{t+1}$ with $\eta_t \sim \mathcal{N}(0, \sigma^2)$. The observations follow the dynamics $y_t | h_t \sim \mathcal{N}(0, \exp(\mu + h_t))$. The prior distributions for the static parameters are: the persistence of the log-volatility process $(\phi + 1)/2 \sim \text{Beta}(20, 1.5)$; the mean log-volatility $\mu \sim \text{Cauchy}(0, 2)$; and the scale of the white-noise process $\sigma \sim \text{Half-Cauchy}(0, 1)$. We reparametrize ϕ and σ with a sigmoid- and softplus-transformation, respectively. Observe that the precision matrix of the AR(1) process is tridiagonal. Since a Cholesky factor of such a matrix is tridiagonal, we consider the parameterisation $C = B_\theta^{-1}$ for an upper-triangular and tridiagonal matrix B_θ. The required operations with such banded matrices have a complexity of $\mathcal{O}(d)$, see for instance Durrande et al. (2019). For comparison, we also consider a diagonal matrix C. We apply the model to ten years of daily returns of the S&P500 index, giving rise to a target dimension of $d = 2519$. In order to account for the different number of gradient evaluations, we use $3.5 \times 10^4 / L$ steps for the adaptation and for evaluating the sampler based on $L \in \{1, \dots, 10\}$ leapfrog steps. We run NUTS for 1000 steps which has a four times higher run-time compared to the other samplers. In addition to using effective sample size to assess convergence, we also report the potential scale reduction factor split-$\hat{R}$ (Gelman et al., 2013; Vehtari et al., 2021) where large values

are indicative of poor mixing. We report results over three replications in Figure 5.5 with more details in Figure 5.19, Appendix 5.6.8. First, HMC with moderately large L tends to improve the effective samples per computation time compared to the MALA case, while also having a smaller $\hat{R}$. Second, using a tridiagonal mass matrix improves mixing compared to a diagonal one, particularly for the latent log-volatilities as seen in the median ESS/sec or median $\hat{R}$ values. The largest absolute eigenvalue of D_L tends to be smaller for a tridiagonal mass matrix and the acceptance rates are approaching 100% more slowly for increasing L. Third, NUTS seems less efficient as does using a dual-adaptation scheme.

We imagine that similar efficient parametrizations of M or M^{-1} can be used for different generalisations of the above stochastic volatility model, such as including p sub-diagonals for log-volatilities having a higher-order AR(p) dynamics or multivariate extensions using a suitable block structure. Likewise, this approach might also be useful for inferences in different Gaussian Markov Random Field models with sparse precision matrices.



(a) Minimum effective sample size per second (after adaptation).

(b) Median effective sample size per second (after adaptation).

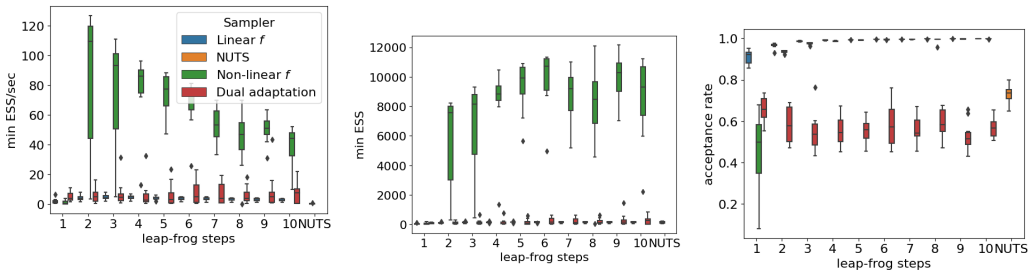
(c) Maximum $\hat{R}$ of $q \mapsto q_i$ (after adaptation).

Figure 5.5: Stochastic volatility model ($d = 2519$).

5.4.5 Learning non-linear transformations

To illustrate an extension to sample from highly non-convex targets by learning a non-linear transformation within the suggested framework as explained in greater detail in Appendix 5.6.3, we consider sampling from a two-dimensional Banana distribution that results from the transformation of $\mathcal{N}(0, \Lambda)$ where Λ is a diagonal matrix having entries 100 and 1 via the volume-preserving map $\phi_b(x) = (x_1, x_2 + b(x_1^2 - 100))$, for $b = 0.1$, cf. (Haario et al., 1999). We consider a RealNVP-type (Dinh et al.,

2016) transformation $f = f_3 \circ f_2 \circ f_1$ where $f_1(x_1, x_2) = (x_1, x_2 \cdot g(s(x_1)) + t(x_1))$, $f_2(x_1, x_2) = (x_1 \cdot g(s(x_2)) + t(x_1), x_2)$ and $f_3(x_1, x_2) = (c_1 x_1, c_2 x_2)$. The functions s and t are neural networks with two hidden layers of size 50. For numerical stability, we found it beneficial to use a modified affine scaling function g as a sigmoid function scaled on a restricted range such as $(0.5, 2)$, as also suggested in (Behrmann et al., 2021). As an alternative, we also consider learning a linear transformation $f(x) = Cx$ for a Cholesky matrix C as well as NUTS and a standard HMC sampler with step size adapted to achieve a target acceptance rate of 0.65. Figure 5.6 summarizes effective sample sizes where each method uses 4×10^5 samples before and after the adaptation. Whereas a linear transformation does not improve on standard HMC, applying a non-linear transformation can improve significantly the effective sample size even after taking into account the computational costs.



(a) Minimum effective sample size per second (after adaptation).

(b) Minimum effective sample size (after adaptation).

(c) Average acceptance rate (after adaptation).

Figure 5.6: Banana-shaped target in dimension $d = 2$.

5.5 Discussion and Outlook

Convergence. We have used Adam (Kingma and Ba, 2014) with a constant step size to adapt the mass matrix, but have stopped the adaptation after some fixed steps so that any convergence is preserved. Different conditions have been established so that infinite adaptive schemes that continue to modify C still converge to the correct invariant distribution, such as diminishing adaptation and containment (Roberts and Rosenthal, 2007). An analysis of the convergence properties of the adaptive HMC algorithm is left for future work.

Limitations. Our approach to learn a constant mass matrix can struggle for targets where the Hessian varies greatly across the state space, which can yield relatively

short integration times with very high acceptance rates. While this effect might be mitigated by considering non-linear transformations, it remains challenging to learn flexible transformations efficiently in high dimensions.

Variations of HMC. We have considered a standard HMC setting for a fixed number of leap-frog steps. One could consider a mixture of HMC kernels with different numbers of leap-frog steps and an interesting question would be how to learn the different mass matrices jointly in an efficient way.

Instead of a full velocity refreshment, partial refreshment strategies (Horowitz, 1991) can sometimes mix better. The suggested adaptation approach can yield very high acceptance rates particularly for increasing leap-frog steps and the learned mass matrix can be used with a partial refreshment. However, it would be interesting to analyse if the adaptation can be adjusted to such persistent velocity updates. It would also be of interest to analyse if similar ideas can be used to adapt different numerical integrators such as those suggested in Beskos et al. (2011) for target densities relative to a Gaussian measure.

Our focus was on learning a mass matrix so that samples from the Markov chain can be used for estimators that are consistent for increasing iterations. However, unbiased estimators might also be constructed using coupled HMC chains (Heng and Jacob, 2019) and one might ask if the adapted mass matrix leads to shorter meeting times in such a setting.

5.6 Appendix

5.6.1 Gradient terms for the adaptation scheme

5.6.1.1 Gradients for the entropy approximation

Following the arguments in Chen et al. (2019a), we can compute the gradient of the term in (5.10) using

$$\frac{\partial}{\partial\theta_i}\mathcal{L}(\theta) = \mathrm{T}\left(\sum_{k=0}^{\infty}(-1)^k \left[D_L\right]^k \frac{\partial}{\partial\theta_i}\left\{D_L\right\}\right) = \mathbb{E}_{N,\varepsilon}\left[\sum_{k=0}^{N}\frac{(-1)^k}{p_k}\varepsilon^\top \left[D_L\right]^k \frac{\partial}{\partial\theta_i}\left\{D_L\right\}\varepsilon\right],$$

which yields a stochastic gradient via a Russian-roulette estimator.

Additionally, to avoid gradients with infinite means even if D_L is not contractive, we consider a spectral normalisation, so that instead of computing recursively $\eta_0 = \varepsilon$

and $\eta_k = D_L \eta_{k-1}$ for $k \in \{1, \dots, N\}$, we set $\bar{\eta}_0 = \varepsilon$ and

$$\bar{\eta}_k = D_L \bar{\eta}_{k-1} \cdot \min\left\{1, \delta' \left\|\bar{\eta}_{k-1}\right\|_2 / \left\|D_L \bar{\eta}_{k-1}\right\|_2\right\} \tag{5.12}$$

for $k \in \{1, \dots, N\}$ and $\delta' \in (0, 1)$, such as $\delta' = 0.99$ in all our experiments. We obtain an estimator

$$\frac{\partial}{\partial \theta_i} \mathcal{L}(\theta) \approx \mathbb{E}_{N,\varepsilon}\left[\sum_{k=0}^{N} \frac{(-1)^k}{p_k} \bar{\eta}_k^\top \frac{\partial}{\partial \theta_i} \{D_L\} \varepsilon\right].$$

5.6.1.2 Gradients for the penalty function

We used the following penalty function

$$h(x) = (x - \delta)^2 \mathbb{1}_{\{x \in [\delta, \delta_2)\}} + ((\delta_2 - \delta)^2 + (\delta_2 - \delta)^2 (x - \delta_2)) \mathbb{1}_{\{x \geqslant \delta_2\}}$$

throughout our experiments with $\delta \in \{0.75, 0.95\}$, and $\delta_2 = 1 + \delta$. The motivation was to have a quadratic increase for the penalty term if the largest absolute eigenvalue approaches 1, and then smoothly switch to a linear function for values larger than δ_2. Gradients for this function can be computed routinely using automatic differentiation.

5.6.1.3 Gradients for the energy error

We can write the energy error as

$$\begin{aligned}
\Delta(q_0, v) &= U(\mathcal{T}_L(v)) - U(q_0) + K(\mathcal{W}_L(v)) - K(C^{-\top} v) \\
&= U\left(q_0 + LhCv - h^2 CC^\top \Xi_L(v) - L\frac{h^2}{2} CC^\top \nabla U(q_0)\right) - U(q_0) \\
&\quad + \frac{1}{2}\left\|v - \frac{h}{2} C\left[\nabla U(q_0) + \nabla U\left(q_L\right)\right] - hC \sum_{\ell=1}^{L-1} \nabla U(q_\ell)\right\|^2 - \frac{1}{2}\|v\|^2 .
\end{aligned} \tag{5.13}$$

Recall from (5.3) that $\Xi_L(v)$ is a weighted sum of potential energy gradients along the leap-frog trajectory. For computing gradients of the energy-error for the fast approximation, we therefore stop the gradient for all $\nabla U(q_\ell)$ for any $\ell \in \{1, \dots, L\}$.

5.6.2 Proof of Lemma 12

Proof. We generalise the arguments from Chen et al. (2019b), Lemma 7. Proceeding by induction over n, we have for the case $n = 1$, for any $v \in \mathbb{R}^d$, that $\mathsf{D}\mathcal{T}_1(v) = hC$ and $\mathcal{S}_1(v) = \frac{1}{h} C^{-1} q_0 - \frac{h}{2} C^\top \nabla U(q_0)$ with derivative of zero. For the case $n = 2$, using (5.2) and (5.3), one obtains

$$\mathsf{D}\mathcal{T}_2(v) - 2hC - h^3 CC^\top \nabla^2 U(\mathcal{T}_1(v)) C \tag{5.14}$$

and moreover

$$\mathsf{D}\mathcal{S}_2(v) = -\frac{h^2}{2} C^\top \nabla^2 U(\mathcal{T}_1(v)) C \tag{5.15}$$

which establishes (5.8). Clearly, $\|\mathsf{D}\mathcal{S}_2(v)\|_p < \frac{1}{8}$ if $2^2 h^2 < \frac{1}{4\left\|C^\top \nabla^2 U(\mathcal{T}_1(v))C\right\|_p}$. Further, for any $n < L$, again from (5.2) and (5.3),

$$\begin{aligned}
\mathsf{D}\mathcal{T}_{n+1}(v) &= (n+1)hC - h^2 CC^\top \mathsf{D}\Xi_{n+1}(v) \\
&= (n+1)hC - h^2 CC^\top \left[\sum_{i=1}^{n} (n+1-i)\nabla^2 U(\mathcal{T}_i(v))\mathsf{D}\mathcal{T}_i(v)\right] \\
&= (n+1)hC - h^2 CC^\top \left[\sum_{i=1}^{n} (n+1-i)\nabla^2 U(\mathcal{T}_i(v)) ihC\,(\mathrm{I} + \mathsf{D}\mathcal{S}_i(v))\right] \\
&= (n+1)hC + (n+1)hC \left[-h^2 \sum_{i=1}^{n} \frac{(n+1-i)}{n+1} iC^\top \nabla^2 U(\mathcal{T}_i(v)) C\,(\mathrm{I} + \mathsf{D}\mathcal{S}_i(v))\right],
\end{aligned}$$

which shows the representation (5.8) for the case $n+1$ by recalling that $\mathsf{D}\mathcal{T}_{n+1}(v) = (n+1)hC(\mathrm{I} + \mathsf{D}\mathcal{S}_{n+1}(v))$. Assume now that $\|\mathsf{D}\mathcal{S}_\ell(v)\|_p < 1/8$ holds for all $\ell \leqslant n$. Then for any $v \in \mathbb{R}^d$

$$\begin{aligned}
\|\mathsf{D}\mathcal{S}_{n+1}(v)\|_p &\leqslant \frac{h^2}{n+1} \sum_{i=1}^{n} i(n+1-i) \left\|C^\top \nabla^2 U(\mathcal{T}_i(v)) C\right\|_p \|\mathrm{I} + \mathsf{D}\mathcal{S}_i(v)\|_p \\
&\leqslant \frac{h^2}{n+1} \sum_{i=1}^{n} \frac{L^2}{4} \left\|C^\top \nabla^2 U(\mathcal{T}_i(v)) C\right\|_p \|\mathrm{I} + \mathsf{D}\mathcal{S}_i(v)\|_p \\
&\leqslant \frac{h^2}{n+1} \sum_{i=1}^{n} \frac{L^2}{4} \frac{1}{4L^2h^2} \left(1 + \frac{1}{8}\right) \leqslant \frac{1}{8}
\end{aligned}$$

where the second inequality follows from $(n+1-i)i \leqslant (\frac{n+1-i+i}{2})^2 \leqslant \frac{L^2}{4}$, whereas the third inequality follows from the induction hypothesis and the assumption $L^2h^2 < \sup_q \frac{1}{4\left\|C^\top \nabla^2 U(q)C\right\|_p}$. □

5.6.3 Extension to learn non-linear transformations

The suggested approach can perform poorly for non-convex potentials or even convex potentials such as arsing in a logistic regression model for some data sets. We illustrate here how to learn a reasonable proposal for a general potential function by considering some version of position-dependent preconditioning. Consider an invertible differentiable transformation $f\colon \mathbb{R}^d \to \mathbb{R}^d$. The idea now is to run HMC with unit mass matrix for the transformed variables $z = f^{-1}(q)$ where $q \sim \pi$. Write $\tilde{\pi}$ for the density of z and let $\tilde{U}$ be the corresponding potential energy function which is given by

$$\tilde{U}(z) = U(H(z)) - \log|\det \mathsf{D}f(z)|$$

with gradient

$$\nabla\tilde{U}(z) = \mathsf{D}f(z)^\top \nabla U(f(z)) - \nabla \log|\det \mathsf{D}f(z)|.$$

The transformation f as well as $\tilde{U}$ generally depend on some parameters θ that we again omit for a less convoluted notation. Our approach can be seen as an alternative for instance to Hoffman et al. (2019) where such a transformation is first learned by trying to approximate $\tilde{\pi}$ with a standard Gaussian density using variational inference, while the HMC hyperparameters are adapted in a second step using Bayesian optimisation.

We write $\tilde{\mathcal{T}}_L \colon v \mapsto z_L$ for the transformation that maps the initial velocity $v = p_0 \sim \mathcal{N}(0, \mathrm{I})$ to the L-th leapfrog step z_L, starting at z_0 based on the potential function $\tilde{U}$ with unit mass matrix $M = \mathrm{I}$. Analogously, we define the mapping $\tilde{\mathcal{W}}_L \colon v \mapsto p_L$ and similarly to (5.5), we define

$$\tilde{\mathcal{S}}_L(v) = \frac{1}{Lh}\tilde{\mathcal{T}}_L(v) - v.$$

We can then reparametrise the proposal at the point $q_0 = f(z_0)$ by $v \mapsto f(\tilde{\mathcal{T}}_L(v))$. Consequently, the log-density of the proposal is given by

$$\log r_L(f(\tilde{\mathcal{T}}_L(v))) = \log \nu(v) - \log|\det \mathsf{D}f(\tilde{\mathcal{T}}_L(v))| - \log|\det \mathsf{D}\tilde{\mathcal{T}}_L(v)|,$$

and we can write

$$\log|\det \mathsf{D}\tilde{\mathcal{T}}_L(v)| = d\log Lh + \log|\det(\mathrm{I} + \mathsf{D}\tilde{\mathcal{S}}_L(v))|.$$

We use the same approximation

$$\mathsf{D}\tilde{\mathcal{S}}_L(v) \approx -h^2 \frac{L^2-1}{6}\, \nabla^2\tilde{U}(z_{\lfloor L/2 \rfloor})$$

based on the transformed Hessian now.

Hessian-vector products can similarly be computed using vector-Jacobian products: With $g(z) = \texttt{grad}(\tilde{U}, z)$, we then compute $\nabla^2\tilde{U}(z)w = \texttt{vjp}(g, z, w)^\top$ for $z = f^{-1}(\texttt{stop\_grad}(f(z_{\lfloor L/2 \rfloor}))$. The motivation for stopping the gradients comes from considering the special case $f \colon z \mapsto Cz$ that corresponds to the position-independent preconditioning scheme above. For such a linear transformation,

$$\tilde{U}(z) = C^\top \nabla^2 U(Cz) C.$$

To recover the previous case, we stop gradients at $q_{\lfloor L/2 \rfloor} = f(z_{\lfloor L/2 \rfloor}) = Cz_{\lfloor L/2 \rfloor}$.

Gradients for the log-accept ratio can be computed based on the log-accept ratio of the transformed chain (Johnson and Geyer, 2012). The energy error of the transformed chain is

$$\begin{aligned}
\Delta_\theta(q_0, v) =& U_\theta(\tilde{\mathcal{T}}_L(v)) - U_\theta(f^{-1}(q_0)) + K(\tilde{\mathcal{W}}_L(v)) - K(v) \\
=& U\Big\{ f\Big[f^{-1}(q_0) + Lhv - h^2 \tilde{\Xi}_L(v) \\
& \qquad - L\frac{h^2}{2}\Big(\mathsf{D}f(f^{-1}(q_0))^\top \nabla U(q_0) - \nabla \log|\det \mathsf{D}f(f^{-1}(q_0))|\Big)\Big]\Big\} \\
& + \log|\det \mathsf{D}f(z_L)| - U(q) + \log|\det \mathsf{D}f(f^{-1}(q))| \\
& + \frac{1}{2}\Big\| v - \frac{h}{2}\big[\mathsf{D}f(z_0)^\top \nabla U(f(z_0)) - \nabla \log|\det \mathsf{D}f(z_0) + \mathsf{D}f(z_L)^\top \nabla U(f(z_L)) \\
& \qquad - \nabla \log|\det \mathsf{D}f(z_L)|\big] - h\sum_{\ell=1}^{L-1} \mathsf{D}f(z_\ell)^\top \nabla U(f(z_\ell)) - \nabla \log|\det \mathsf{D}f(z_\ell)|\Big\|^2 \\
& - \frac{1}{2}\|v\|^2,
\end{aligned}$$

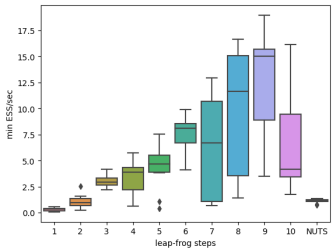
where

$$\tilde{\Xi}_L(v) = \sum_{i=1}^{L}(L-i)\Big[\mathsf{D}f(z_i)^\top \nabla U(f(z_i)) - \nabla \log|\det \mathsf{D}f(z_i)|\Big]$$

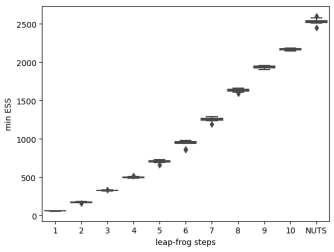
and $z_0, \ldots, z_L$ is the leap-frog trajectory starting at $z_0 = f^{-1}(q_0)$. We also stop all U gradients, *i.e.* $\nabla U(f(z_\ell)) \leftarrow \texttt{stop\_grad}(\nabla U(f(z_\ell))$. It can be seen this recovers the above setting if $f\colon z \mapsto Cz$.

5.6.4 Gaussian targets experiments

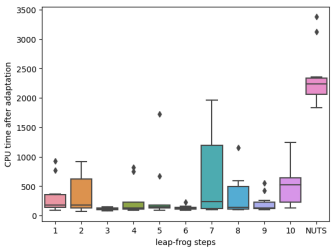
5.6.4.1 High-dimensional anisotropic Gaussian target



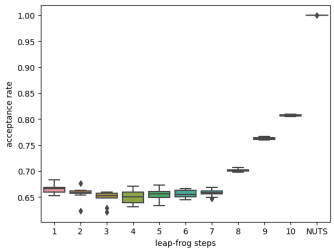
(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).



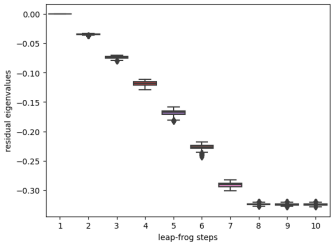
(b) Minimum effective sample size of $q \mapsto q_i$. NUTS adapted over 1/10-th of the steps.



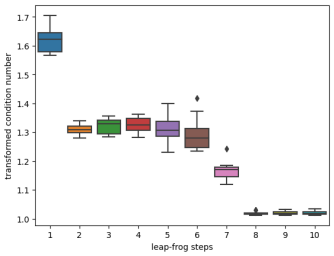
(c) CPU time (after adaptation). NUTS adapted over 1/10-th of the steps.



(d) Average acceptance rate (after adaptation)



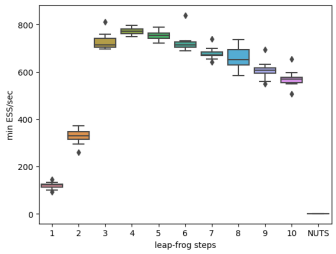
(e) Estimates of the eigenvalues of D_L using power iteration.



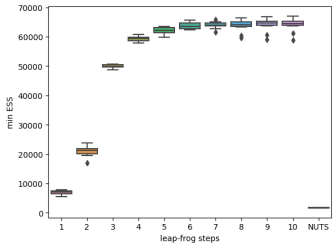
(f) Condition number of transformed Hessian $C^\top \Sigma^{-1} C$.

Figure 5.7: Independent Gaussian target ($d = 10000$).

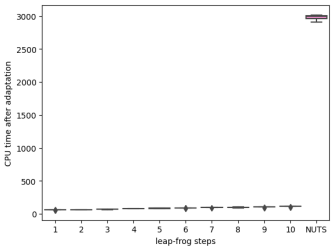
5.6.4.2 Ill-conditioned anisotropic Gaussian target



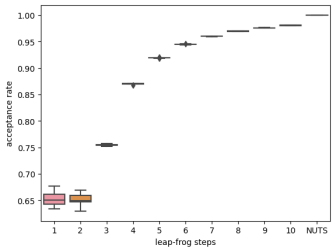
(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).



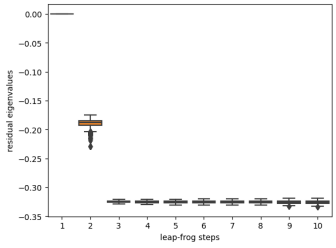
(b) Minimum effective sample size of $q \mapsto q_i$. NUTS adapted over 1/10-th of the steps.



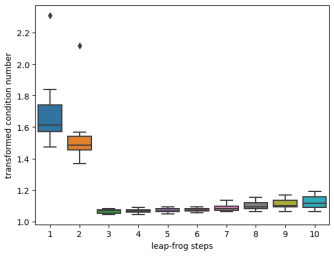
(c) CPU time (after adaptation). NUTS adapted over 1/10-th of the steps.



(d) Average acceptance rate (after adaptation)



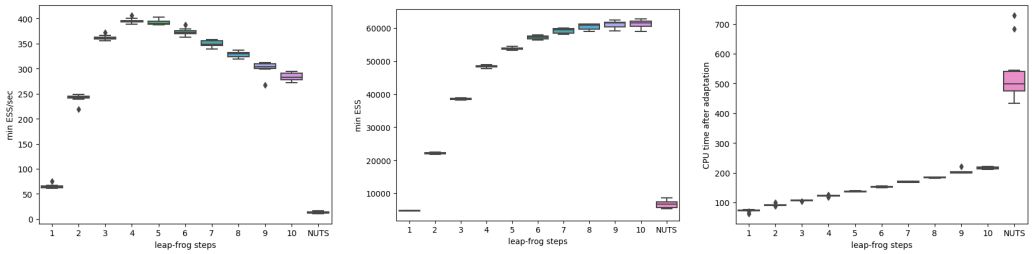
(e) Estimates of the eigenvalues of D_L using power iteration.



(f) Condition number of transformed Hessian $C^\top \Sigma^{-1} C$.

Figure 5.8: Ill-conditioned Gaussian target ($d = 100$).

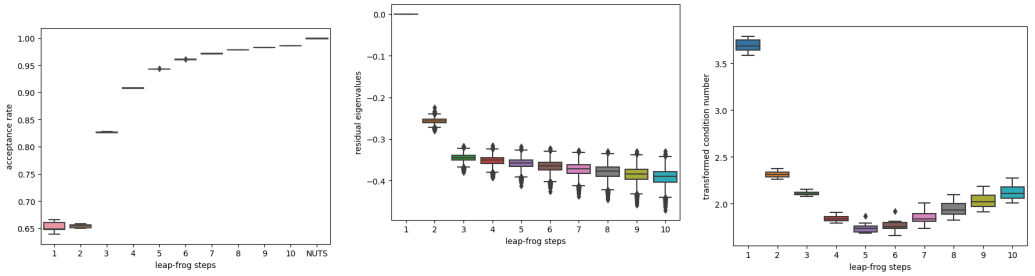
5.6.4.3 Correlated Gaussian target



(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).

(b) Minimum effective sample size of $q \mapsto q_i$. NUTS adapted over 1/10-th of the steps.

(c) CPU time (after adaptation). NUTS adapted over 1/10-th of the steps.



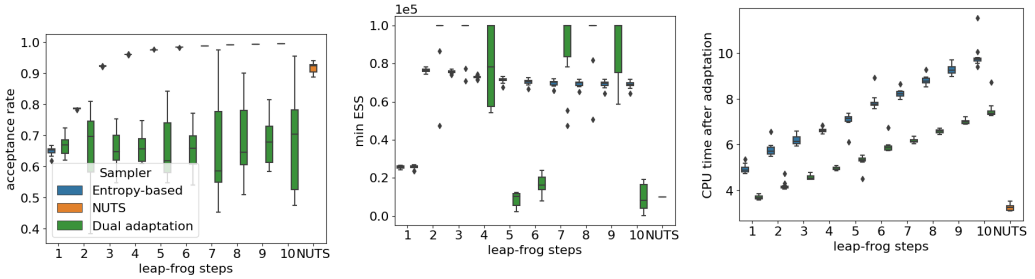
(d) Average acceptance rate (after adaptation)

(e) Estimates of the eigenvalues of D_L using power iteration.

(f) Condition number of transformed Hessian $C^\top \Sigma^{-1} C$.

Figure 5.9: Correlated Gaussian target ($d = 51$).

5.6.4.4 IID Gaussian target



(a) Average acceptance rate (after adaptation).

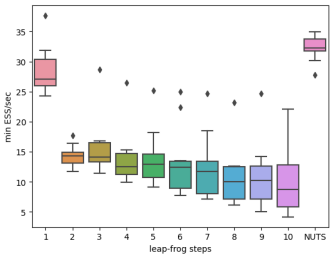
(b) Minimum effective sample size of $q \mapsto q_i$. NUTS uses 1/10-th of the samples.

(c) CPU time (after adaptation). NUTS uses 1/10-th of the samples.

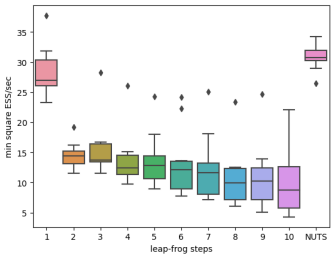
Figure 5.10: IID Gaussian target ($d = 10$).

5.6.5 Logistic regression experiments

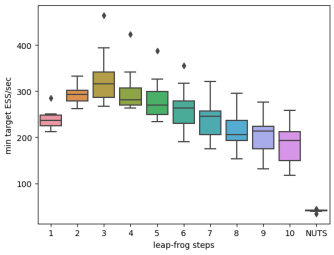
5.6.6 Australian credit data



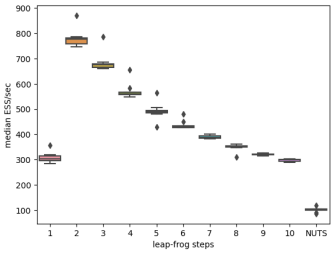
(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).



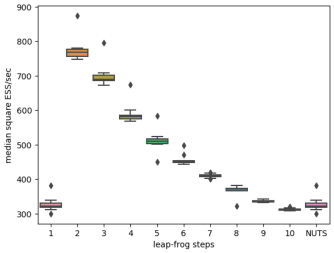
(b) Minimum effective sample size of of $q \mapsto q_i^2$ per second (after adaptation)



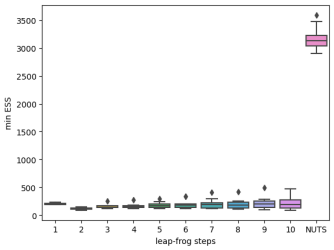
(c) Effective sample size of of $q \mapsto \log \pi(q)$ per second (after adaptation).



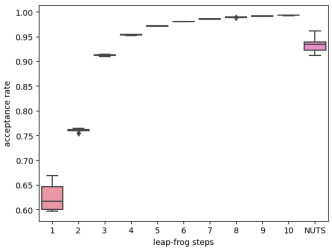
(d) Median effective sample size of $q \mapsto q_i$ per second (after adaptation).



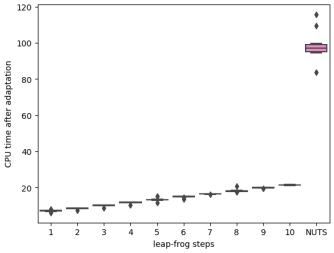
(e) Median effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).



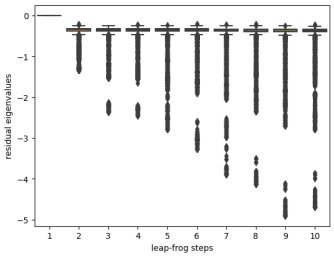
(f) Minimum effective sample size of $q \mapsto q_i$.



(g) Average acceptance rate (after adaptation)



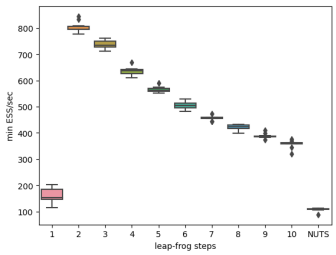
(h) CPU time (after adaptation)



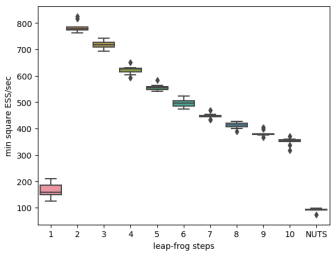
(i) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.11: Bayesian logistic regression for Australian Credit data set ($d = 15$).

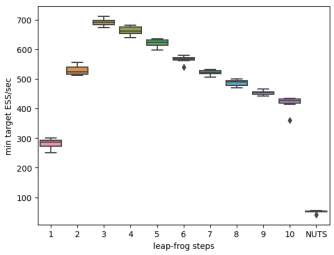
5.6.6.1 Heart data



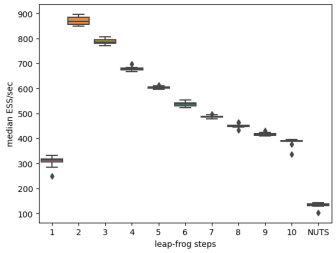
(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).



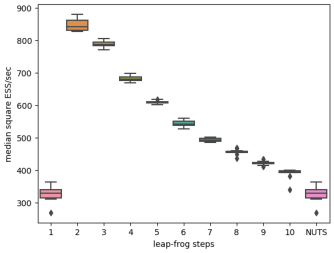
(b) Minimum effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).



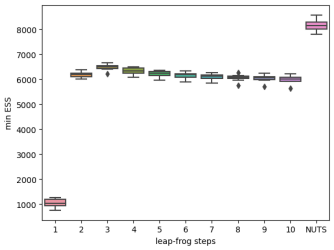
(c) Effective sample size of of $q \mapsto \log \pi(q)$ per second (after adaptation).



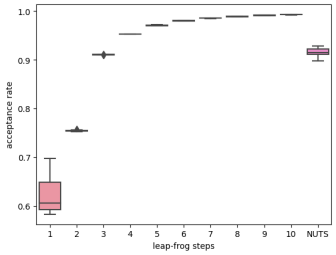
(d) Median effective sample size of $q \mapsto q_i$ per second (after adaptation).



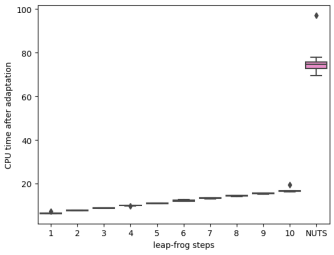
(e) Median effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).



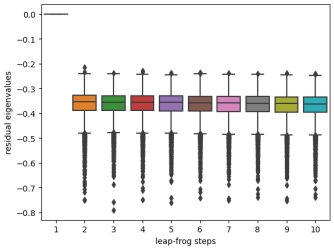
(f) Minimum effective sample size of $q \mapsto q_i$.



(g) Average acceptance rate (after adaptation).



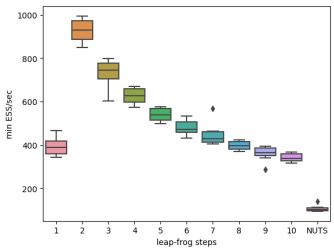
(h) CPU time (after adaptation).



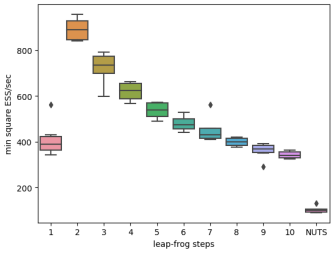
(i) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.12: Bayesian logistic regression for Caravan data set ($d = 14$).

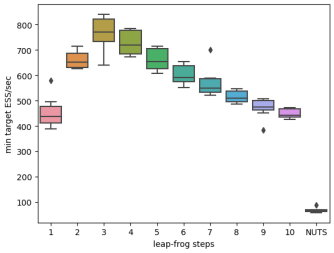
5.6.6.2 Pima data



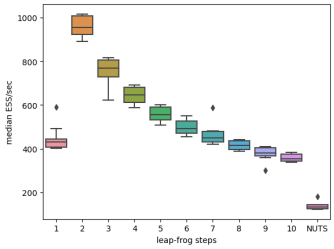
(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).



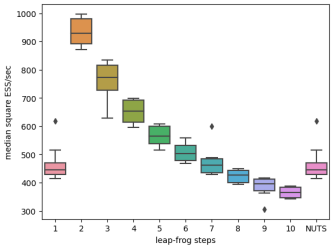
(b) Minimum effective sample size of of $q \mapsto q_i^2$ per second (after adaptation)



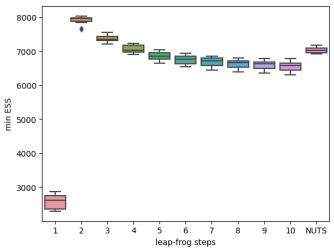
(c) Effective sample size of of $q \mapsto \log \pi(q)$ per second (after adaptation).



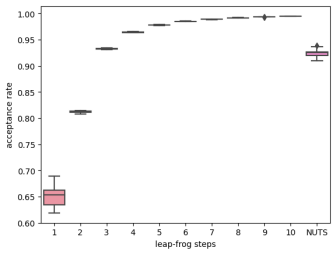
(d) Median effective sample size of $q \mapsto q_i$ per second (after adaptation).



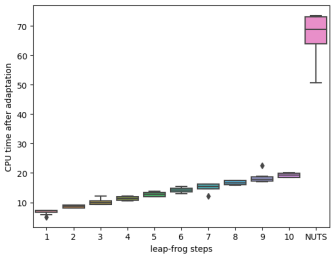
(e) Median effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).



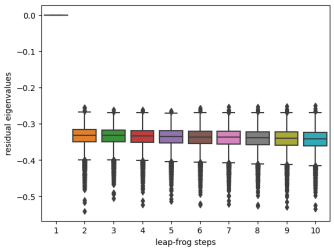
(f) Minimum effective sample size of $q \mapsto q_i$.



(g) Average acceptance rate (after adaptation)



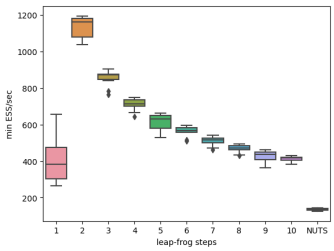
(h) CPU time (after adaptation)



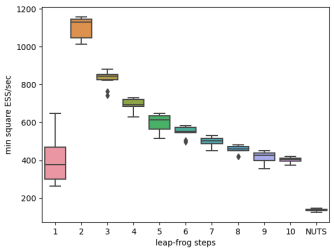
(i) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.13: Bayesian logistic regression for Pima data set ($d = 8$).

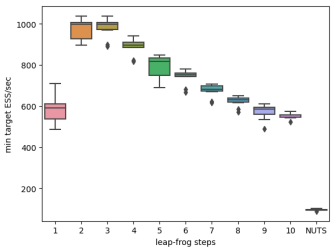
5.6.6.3 Ripley data



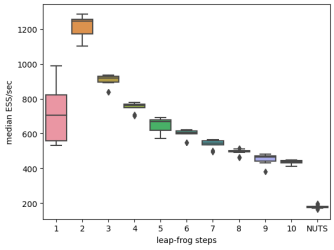
(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).



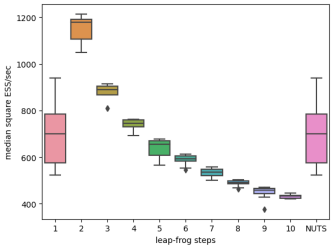
(b) Minimum effective sample size of of $q \mapsto q_i^2$ per second (after adaptation)



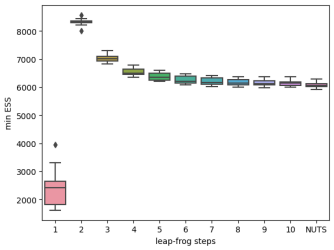
(c) Effective sample size of of $q \mapsto \log \pi(q)$ per second (after adaptation).



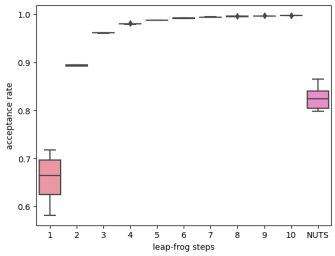
(d) Median effective sample size of $q \mapsto q_i$ per second (after adaptation).



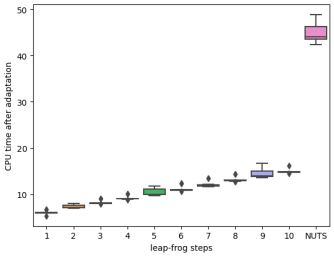
(e) Median effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).



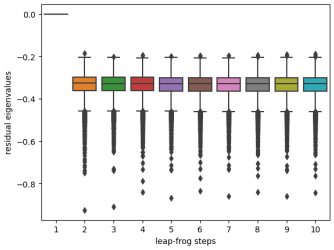
(f) Minimum effective sample size of $q \mapsto q_i$.



(g) Average acceptance rate (after adaptation)



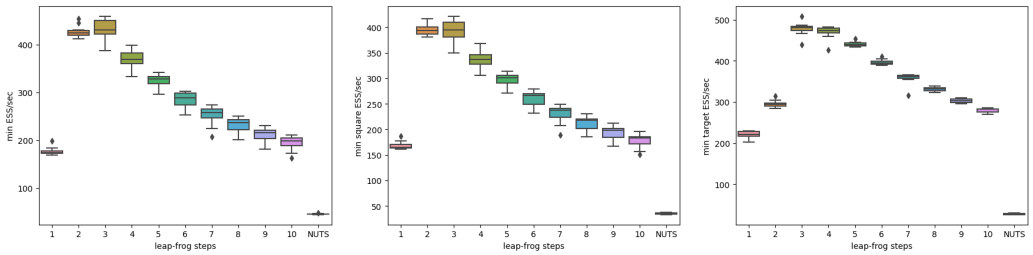
(h) CPU time (after adaptation)



(i) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.14: Bayesian logistic regression for Ripley data set ($d = 3$).

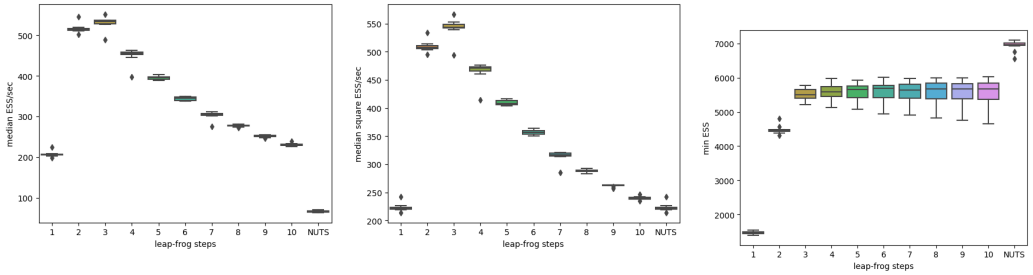
5.6.6.4 German credit data



(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).

(b) Minimum effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).

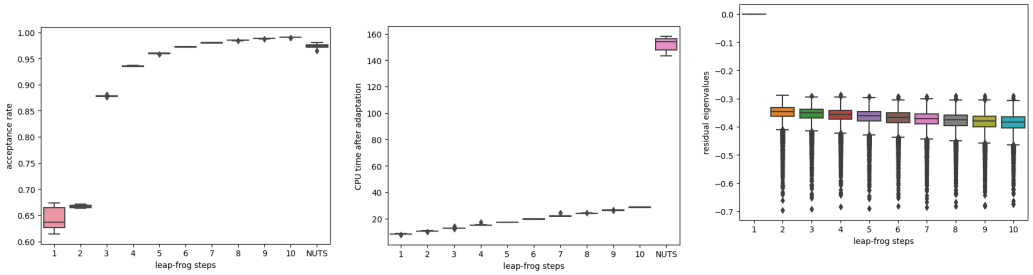
(c) Effective sample size of of $q \mapsto \log \pi(q)$ per second (after adaptation).



(d) Median effective sample size of $q \mapsto q_i$ per second (after adaptation).

(e) Median effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).

(f) Minimum effective sample size of $q \mapsto q_i$.



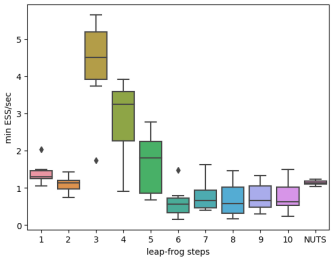
(g) Average acceptance rate (after adaptation).

(h) CPU time (after adaptation).

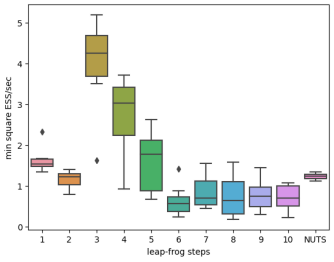
(i) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.15: Bayesian logistic regression for German credit data set ($d = 25$)..

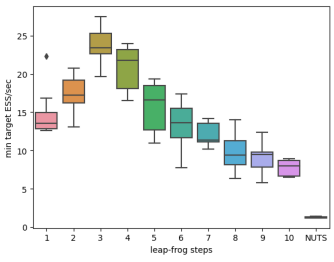
5.6.6.5 Caravan data



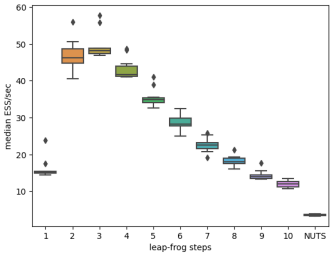
(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).



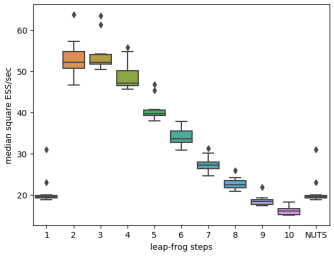
(b) Minimum effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).



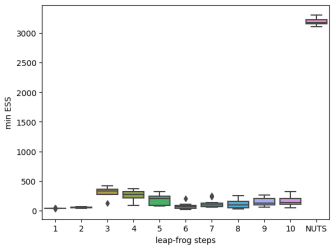
(c) Effective sample size of of $q \mapsto \log \pi(q)$ per second (after adaptation).



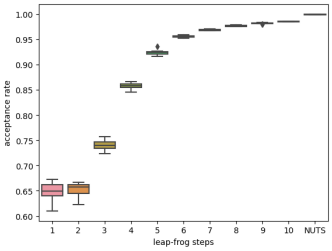
(d) Median effective sample size of $q \mapsto q_i$ per second (after adaptation).



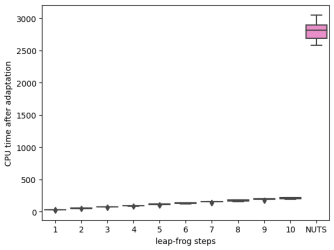
(e) Median effective sample size of of $q \mapsto q_i^2$ per second (after adaptation).



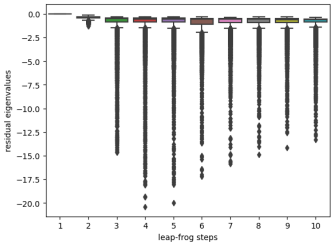
(f) Minimum effective sample size of $q \mapsto q_i$.



(g) Average acceptance rate (after adaptation).



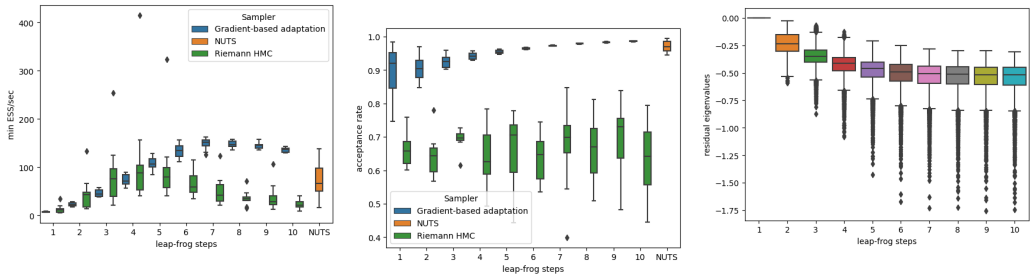
(h) CPU time (after adaptation).



(i) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.16: Bayesian logistic regression for Caravan data set ($d = 87$).

5.6.7 Log-Gaussian Cox Point Process

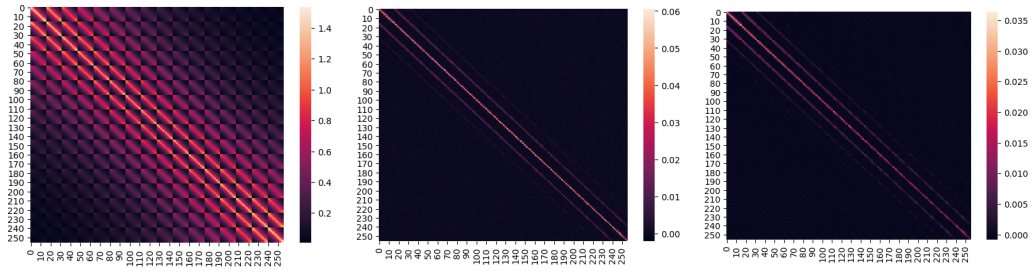


(a) Minimum effective sample size per second (after adaptation).

(b) Average acceptance rate (after adaptation).

(c) Estimates of the eigenvalues of D_L using power iteration.

Figure 5.17: Cox process in dimension $d = 64$.



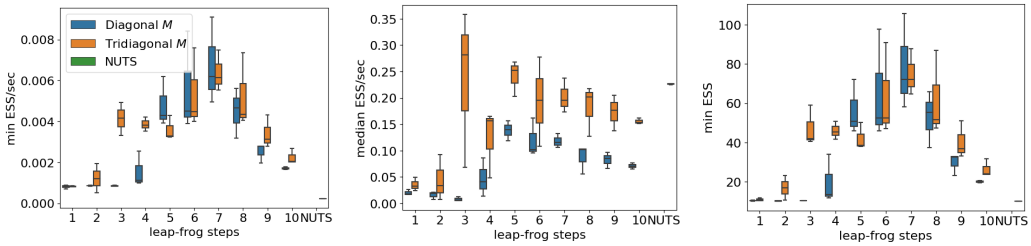
(a) Inverse mass matrix $(\Lambda + \Sigma^{-1})^{-1}$ of the Riemann manifold based samplers.

(b) Inverse mass matrix $CC^\top$ for the entropy-based scheme with $L = 1$.

(c) Inverse mass matrix $CC^\top$ for the entropy-based scheme with $L = 5$.

Figure 5.18: Inverse mass matrix for the Cox process in dimension $d = 256$ for the different schemes.

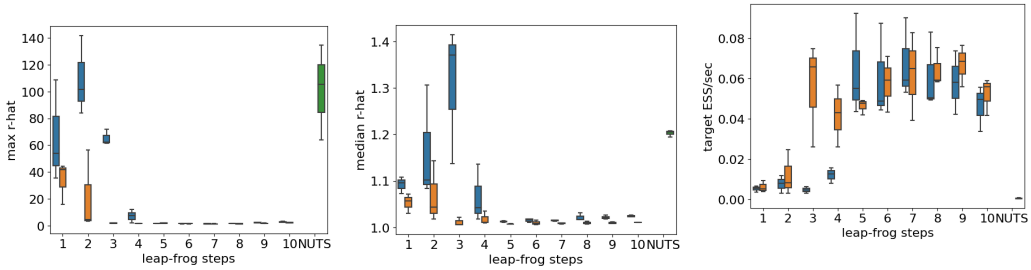
5.6.8 Stochastic volatility model



(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).

(b) Median effective sample size of $q \mapsto q_i$ per second (after adaptation).

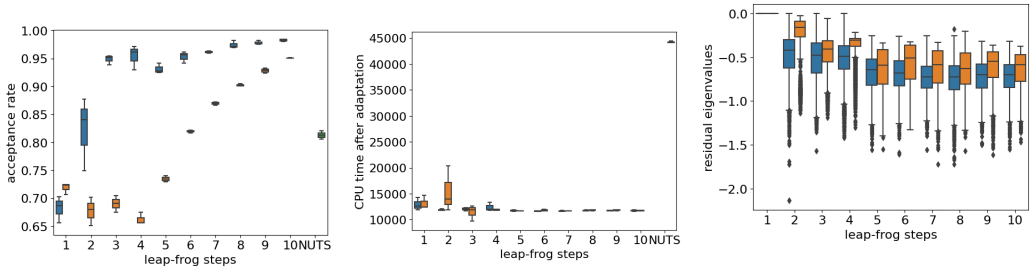
(c) Minimum effective sample size of $q \mapsto q_i$ (after adaptation).



(d) Maximum $\hat{R}$ of $q \mapsto q_i$ (after adaptation).

(e) Median $\hat{R}$ of $q \mapsto q_i$ (after adaptation).

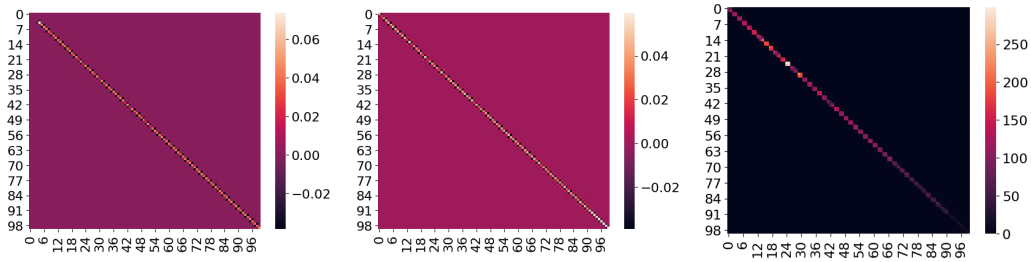
(f) Effective sample size of of $q \mapsto \log \pi(q)$ per second (after adaptation).



(g) Average acceptance rate (after adaptation).

(h) CPU time (after adaptation).

(i) Estimates of the eigenvalues of D_L using power iteration.

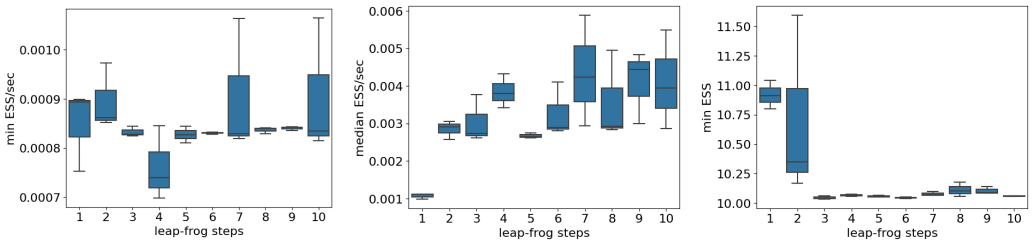


(j) First 100 dimensions of M^{-1} for $L = 5$ with a tridiagonal mass matrix.

(k) Last 100 dimensions of M^{-1} for $L = 5$ with a tridiagonal mass matrix.

(l) Last 100 dimensions of M for $L = 5$ with a tridiagonal mass matrix.

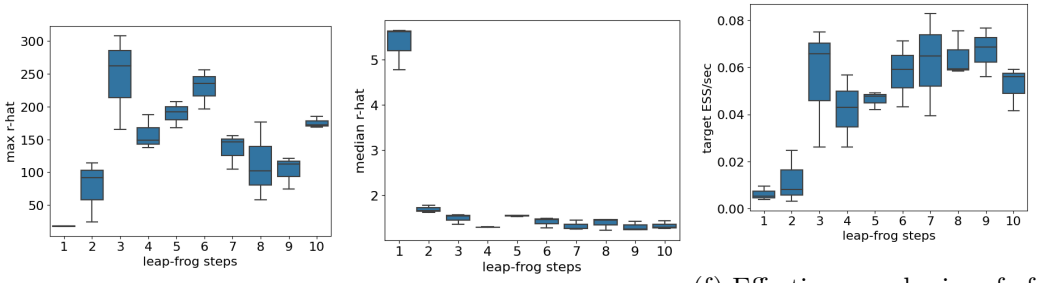
Figure 5.19: Entropy-based adaptation and NUTS for the Stochastic volatility model.



(a) Minimum effective sample size of $q \mapsto q_i$ per second (after adaptation).

(b) Median effective sample size of $q \mapsto q_i$ per second (after adaptation).

(c) Minimum effective sample size of $q \mapsto q_i$ (after adaptation).



(d) Maximum $\hat{R}$ of $q \mapsto q_i$ (after adaptation).

(e) Median $\hat{R}$ of $q \mapsto q_i$ (after adaptation).

(f) Effective sample size of of $q \mapsto \log \pi(q)$ per second (after adaptation).

Figure 5.20: Dual adaptation for the Stochastic volatility model.

Chapter 6

Outlook and future work

Different topics have been explored in this thesis. We summarize our contributions and remark on some limitations.

Chapter 2. We have introduced a new Bayesian framework for inferring DNA methylation patterns on a genome-wide scale. Our approach allows for more flexible methylation regimes that have not been explored previously. By making use of the dependence structure from a hidden Markov model, we have developed a novel method for detecting differentially methylated positions and regions. This new method is applicable for methylation data with low read depth or missing observations by borrowing strength from neighbouring CpG sites. A limitation of our model is that it does not include other covariates that might have an impact on the underlying methylation signal.

Chapter 3. We have developed a new approximate inference scheme for generic state space models. It has been shown that this can be seen as an alternative to particle MCMC methods on an extended state space. This method has been applied for inference in non-linear Hawkes processes with latent intensity dynamics, which can be seen as an alternative to approaches using Neural ODEs (Chen et al., 2020). Our approach has the limitation that it does not provide a tight upper bound and trying to target the smoothing distribution can be an alternative (Lawson et al., 2018). We have also ignored a high-variance term in our gradient estimator. Recently, Corenflos et al. (2021) have suggested an alternative using optimal transport ideas with a higher computational costs, whilst Scibior et al. (2021) proposed to modify the backward pass while retaining a standard particle filter in the forward pass.

Chapter 4. A novel variational density on $\mathbb{R}^d$ has been developed that allows for modeling dependencies with a reasonable complexity of $\mathcal{O}(d \log d)$ which is of similar order as previously suggested variational families under low-rank assumptions. We have applied the density for inference in a high-dimensional sparse Bayesian neural network. In contrast to Gaussian densities, a limitation of our density is that its theoretical properties are less-well understood. In most applications of Bayesian neural networks, the primary goal is often not to find a good approximation of the posterior distribution over the weights of the network, but rather to find a variational distribution that yields good predictive performance by Bayesian model averaging. In this case, more expressive distributions might not necessarily help, see for instance Swiatkowski et al. (2020) that show that a mean-field Gaussian model that is restricted to a low-rank structure for the reshaped posterior standard deviations can yield similar performance than an unrestricted mean-field model and Farquhar et al. (2020) that show that a mean-field assumption becomes less restrictive as the depth of the network increases.

Chapter 5. A new adaptation scheme for HMC has been proposed which can improve the mixing efficiency compared to some previously suggested adaptation procedures. We have considered adapting a standard HMC algorithm, but expect that some ideas can be extended to adapt different variants of HMC. However, the suggested adaptation strategy might be more limited to sampling from target densities where the Hessian of the log-density is relatively constant across the state space. We have also considered a moderate number of leapfrog steps and it remains questionable if the adaptation scheme is sensible if one wants to apply many more leapfrog steps in the order of $\mathcal{O}(10^3)$ as considered for instance in Izmailov et al. (2021) for inference in large Bayesian neural networks.

We conclude with two future projects as a follow-up of Chapter 5.

Adaptive MCMC for learning deep generative models. We are interested in learning deep generative latent variable models using Variational Autoencoders (VAEs) (Kingma and Welling, 2014; Rezende et al., 2014). Let $\mathsf{X} \subset \mathbb{R}^{d_x}$, $\mathsf{Z} \subset \mathbb{R}^{d_z}$ and fix some *prior* density $p(z)$ for $z \in \mathsf{Z}$, with all densities assumed with respect to the Lebesgue measure. Consider a conditional density $p_\theta(x|z)$, also called *decoder*, with $z \in \mathsf{Z}$, $x \in \mathsf{X}$ and $\theta \in \Theta \subset \mathbb{R}^{d_\theta}$. We can interpret this decoder as a generative network

that tries to explain a data point x using a latent variable z. This latent structure yields the following distribution of the data $p_\theta(x) = \int_{\mathsf{X}} p_\theta(x|z)p(z)\mathrm{d}z$. Denote the empirical distribution of some observed data set by μ. We want to minimize the negative log-likelihood with respect to μ, *i.e.* $\min_{\theta\in\Theta} \int_{\mathsf{X}} -\log p_\theta(x)\mu(\mathrm{d}x)$. Consider also the *posterior* density $p_\theta(z|x) \propto p(z)p_\theta(x|z)$ as well as the conditional distribution $q_\phi(z|x)$ with parameter $\phi \in \Phi \subset \mathbb{R}^{d_\phi}$, commonly termed *encoder*.

We fix $x \in \mathsf{X}$ and consider now a Markov kernel $M^k_{\phi_k,x}$ with parameter $\phi_k \in \Phi_k \subset \mathbb{R}^{d_{\phi_k}}$, that is reversible with respect to $p_\theta(z|x)$. We can construct a distribution $q_\phi(z|x)$ by first sampling from an initial tractable distribution $q^0_{\phi_0}(z|x)$ and then applying the K Markov kernels $M^k_{\phi_k,x}$ for $k \in \{1,\dots,K\}$. Put another way, we consider the following variational family

$$\mathcal{Q}_x = \{q^K_{\phi,x}(\cdot|x) = q^0_{\phi_0}(\cdot|x)M^1_{\phi_1,x}\dots M^K_{\phi_K,x}\ , \phi_k \in \Phi_k\}.$$

Our aim is to learn the parameters ϕ_k such as the pre-conditioning matrix when $M^k_{\phi_k,x}$ is a MALA or HMC kernel using the same objective as considered in Chapter 5, while also optimizing over ϕ_0 to minimize

$$\int_{\mathsf{X}} \mathrm{KL}\left(q^0_{\phi_0,}(\cdot|x)|p_\theta(\cdot|x)\right)\mu(\mathrm{d}x)$$

and over θ by maximizing

$$\int_{\mathsf{X}} \left(\mathbb{E}_{q^K_{\phi,x}(z|x)}\left[\log p_\theta(x|z)\right]\right)\mu(\mathrm{d}x).$$

Adaptive piecewise-peterministic Markov processes. We are interested in the problem of adapting piecewise-deterministic Markov processes (PDMPs) that can be used to sample from some target density π on $\mathcal{X} \subset \mathbb{R}^d$. For a review, see Fearnhead et al. (2018); Vanetti et al. (2017). We augment the state space $\mathcal{X}$ with a velocity variable $v \in \mathcal{V}$ drawn from some probability distribution ν. For the Zig-Zag (ZZ) process, cf. Bierkens et al. (2019), ν is the uniform distribution on $\mathcal{V} = \{-1,1\}^d$, while for the Bouncy-Particle Sampler (BPS), see Bouchard-Côté et al. (2018), ν is a rotation invariant distribution such as the uniform distribution on the d-dimensional unit-sphere or $\nu = \mathcal{N}(0,\sigma^2\mathrm{I})$. Let us recall a definition of a PDMP (X,V) on $\mathcal{X}\times\mathcal{V}$, cf. Davis (1984); Jacobsen (2006); Durmus et al. (2018). Consider a time-homogeneous differential flow $\varphi\colon \mathbb{R}_+ \times \mathcal{X}\times\mathcal{V} \to \mathbb{R}^d$, $(t,x,v) \mapsto \varphi_t(x,v)$ measurable and continuously differentiable that satisfies for all $(x,v) \in \mathcal{X}\times\mathcal{V}$,

$s, t > 0$, $\varphi_{t+s}(x, v) = \varphi_t(\varphi_s(x, v))$. The stochastic dynamics of (X, V) are based on a marked point process $(T_n, I_n, \tilde{X}_n, \tilde{V}_n)$ with marks $(I_n, \tilde{X}_n, \tilde{V}_n) \in \{1, .., \ell\} \times \mathcal{X} \times \mathcal{V}$ and jump times $T_1 < T_2 < ...$ being $\mathbb{R}_+$-valued random variables generated by ℓ intensities or jump rates $\lambda_i \colon \mathbb{R}_+ \times \mathcal{X} \times \mathcal{V} \to \mathbb{R}_+$, $(t, x) \mapsto \lambda_i(t, x, v)$. Furthermore, we introduce stochastic time-homogeneous Markov kernels $Q_i \colon \mathcal{X} \times \mathcal{V} \times \mathcal{B}(\mathcal{X} \times \mathcal{V}) \to [0, 1]$ with $x \mapsto Q_i(x, B)$ measurable and continuous for all $B \in \mathcal{B}(\mathcal{X} \times \mathcal{V})$, which describes the probability $Q_i(x, B)$ that the process jumps into set $B \in \mathcal{B}(\mathcal{X} \times \mathcal{V})$, given the state $(x, v) \in \mathcal{X} \times \mathcal{V}$ immediately before the jump of component i and $\varphi_s(\tilde{X}_n, \tilde{V}_n)$ is the state of (X, V) at time $T_n + s < T_{n+1}$.

Observe that the stochastic process (X, V) is a random variable from a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to the path space of right continuous functions with left limits that can be sampled from using thinning methods. The continuous-time process yields an embedded discrete-time Markov chain $(\tilde{X}_k, \tilde{V}_k, S_k, I_k)_{k \in \mathbb{N}}$ with $\tilde{X}_k = X_{T_k}$, $\tilde{V}_k = V_{T_k}$ and $T_k = \sum_{j=1}^{k} S_k$ having filtration $\tilde{\mathcal{F}}_n = \sigma((\tilde{X}_k, \tilde{V}_k, S_k, I_k)_{k \leqslant n})$ that satisfies

$$\begin{aligned} &\mathbb{P}\left((\tilde{X}_{k+1}, \tilde{V}_{k+1}) \in B, S_{k+1} \leqslant s, I_{k+1} = i | \tilde{\mathcal{F}}_k\right) \\ &= \int_0^s Q_i(\varphi_t(\tilde{X}_k, \tilde{V}_k), B) \lambda_i(\varphi_t(\tilde{X}_k, \tilde{V}_k)) \exp\left(-\sum_{j=1}^{\ell} \int_0^t \lambda_j(\varphi_u(\tilde{X}_k, \tilde{V}_k)) \mathrm{d}u\right) \mathrm{d}t. \end{aligned}$$

We aim to explore further adapting the transition dynamics of this embedded Markov chain, which includes all simulated events, including those where the jumps can be of size zero if events are simulated using thinning. For instance, one can consider learning a C^1-diffeomorphism $f_\theta \colon \mathcal{X} \to \mathcal{X}$ such that a standard BPS or ZZ sampler has stationary distribution $\pi_\theta \otimes \nu$, where $\pi_\theta(A) = \pi(f_\theta(A))$ for $A \in \mathcal{B}(\mathcal{X})$ and that $(f_\theta(\tilde{X}_k))_{k \in \mathbb{N}}$ with $\tilde{X}_k$ the position component of the embedded Markov chain has a large expected squared jumping distance or a large generalised speed measure.

Bibliography

Adams, R. P. and MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., Stuart, A., et al. (2017). Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431.

Akalin, A., Kormaksson, M., Li, S., Garrett-Bakelman, F. E., Figueroa, M. E., Melnick, A., and Mason, C. E. (2012). methylkit: a comprehensive r package for the analysis of genome-wide dna methylation profiles. *Genome biology*, 13(10):1–9.

Alenlöv, J. and Olsson, J. (2019). Particle-based adaptive-lag online marginal smoothing in general state-space models. *IEEE Transactions on Signal Processing*, 67(21):5571–5582.

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C., Lee, A., and Livingstone, S. (2020). A general perspective on the metropolis-hastings kernel. *arXiv preprint arXiv:2012.14881*.

Angermueller, C., Lee, H. J., Reik, W., and Stegle, O. (2017). Deepcpg: accurate prediction of single-cell dna methylation states using deep learning. *Genome biology*, 18(1):1–13.

Archer, E., Park, I. M., Buesing, L., Cunningham, J., and Paninski, L. (2015). Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*.

Attias, H. (2000). A variational baysian framework for graphical models. In *Advances in neural information processing systems*, pages 209–215.

Bacry, E., Bompaire, M., Gaïffas, S., and Poulsen, S. (2017). tick: a python library for statistical learning, with a particular emphasis on time-dependent modeling. *arXiv preprint arXiv:1707.03003*.

Bacry, E., Jaisson, T., and Muzy, J.-F. (2016). Estimation of slowly decreasing hawkes kernels: application to high-frequency order book dynamics. *Quantitative Finance*, pages 1–23.

Bacry, E., Mastromatteo, I., and Muzy, J.-F. (2015). Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005.

Barber, D. and Bishop, C. M. (1998). Ensemble learning for multi-layer networks. In *Advances in neural information processing systems*, pages 395–401.

Bardenet, R., Doucet, A., and Holmes, C. C. (2017). On markov chain monte carlo methods for tall data. *Journal of Machine Learning Research*, 18(47).

Bedford, T. and Cooke, R. M. (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial intelligence*, 32(1-4):245–268.

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. (2019). Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582.

Behrmann, J., Vicol, P., Wang, K.-C., Grosse, R., and Jacobsen, J.-H. (2021). Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR.

Benjamini, Y. and Heller, R. (2008). Screening for partial conjunction hypotheses. *Biometrics*, 64(4):1215–1222.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188.

Berg, R. v. d., Hasenclever, L., Tomczak, J. M., and Welling, M. (2018). Sylvester normalizing flows for variational inference. *arXiv preprint arXiv:1803.05649*.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. (2013). Optimal tuning of the hybrid monte carlo algorithm. *Bernoulli*, 19(5A):1501–1534.

Beskos, A., Pinski, F. J., Sanz-Serna, J. M., and Stuart, A. M. (2011). Hybrid monte carlo on hilbert spaces. *Stochastic Processes and their Applications*, 121(10):2201–2230.

Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*.

Bierkens, J., Fearnhead, P., Roberts, G., et al. (2019). The zig-zag process and super-efficient sampling for bayesian analysis of big data. *The Annals of Statistics*, 47(3):1288–1320.

Bird, A. (2002). Dna methylation patterns and epigenetic memory. *Genes & development*, 16(1):6–21.

Bissiri, P. G., Holmes, C. C., and Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 78(5):1103.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1613–1622.

Bock, C. (2012). Analysing and interpreting dna methylation data. *Nature Reviews Genetics*, 13(10):705–719.

Bou-Rabee, N. and Sanz-Serna, J. M. (2018). Geometric integrators and the hamiltonian monte carlo method. *Acta Numerica*, 27:113–206.

Bouchard-Côté, A., Vollmer, S. J., and Doucet, A. (2018). The bouncy particle sampler: A nonreversible rejection-free markov chain monte carlo method. *Journal of the American Statistical Association*, 113(522):855–867.

Boustati, A., Akyildiz, O. D., Damoulas, T., and Johansen, A. (2020). Generalised bayesian filtering via sequential monte carlo. *Advances in Neural Information Processing Systems*, 33.

Bowsher, C. G. et al. (2007). Modelling security market events in continuous time: Intensity based, multivariate point process models. *Journal of Econometrics*, 141(2):876–912.

Brémaud, P. and Massoulié, L. (1996). Stability of nonlinear hawkes processes. *The Annals of Probability*, pages 1563–1588.

Brémaud, P. and Massoulié, L. (2002). Power spectra of general shot noises and hawkes point processes with a random excitation. *Advances in Applied Probability*, 34(01):205–222.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

Burger, L., Gaidatzis, D., Schübeler, D., and Stadler, M. B. (2013). Identification of active regulatory regions from dna methylation data. *Nucleic acids research*, 41(16):e155–e155.

Cappé, O., Moulines, E., and Rydén, T. (2006). *Inference in hidden Markov models*. Springer Science & Business Media.

Caron, F., Doucet, A., and Gottardo, R. (2012). On-line changepoint detection and parameter estimation with application to genomic data. *Statistics and Computing*, 22(2):579–595.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480.

Chatterjee, S., Diaconis, P., et al. (2018). The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135.

Chen, R. T., Amos, B., and Nickel, M. (2020). Neural spatio-temporal point processes. *arXiv preprint arXiv:2011.04583*.

Chen, T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. (2019a). Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, pages 9913–9923.

Chen, Y., Dwivedi, R., Wainwright, M. J., and Yu, B. (2019b). Fast mixing of metropolized hamiltonian monte carlo: Benefits of multi-step gradients. *arXiv preprint arXiv:1905.12247*.

Chen, Z. and Vempala, S. S. (2019). Optimal convergence rate of hamiltonian monte carlo for strongly logconcave distributions. *arXiv preprint arXiv:1905.02313*.

Chib, S., Omori, Y., and Asai, M. (2009). Multivariate stochastic volatility. In *Handbook of Financial Time Series*, pages 365–400. Springer.

Chopin, N., Ridgway, J., et al. (2017). Leave pima indians alone: binary regression as a benchmark for bayesian computation. *Statistical Science*, 32(1):64–87.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988.

Corenflos, A., Thornton, J., Doucet, A., and Deligiannidis, G. (2021). Differentiable particle filtering via entropy-regularized optimal transport. *arXiv preprint arXiv:2102.07850*.

Cornebise, J., Moulines, É., and Olsson, J. (2008). Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, 18(4):461–480.

Cornish, R., Caterini, A. L., Deligiannidis, G., and Doucet, A. (2019). Relaxing bijectivity constraints with continuously indexed normalising flows. *arXiv preprint arXiv:1909.13833*.

Cremer, C., Morris, Q., and Duvenaud, D. (2017). Reinterpreting importance-weighted autoencoders. *arXiv preprint arXiv:1704.02916*.

Cui, S., Guha, S., Ferreira, M. A., Tegge, A. N., et al. (2015). hmmseq: A hidden markov model for detecting differentially expressed genes from rna-seq data. *The Annals of Applied Statistics*, 9(2):901–925.

Daley, D. J. and Vere-Jones, D. (2003). An introduction to the theory of point processes volume i: Elementary theory and methods.

Dassios, A. and Zhao, H. (2011). A dynamic contagion process. *Advances in applied probability*, 43(03):814–846.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*.

Davis, M. H. (1984). Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 353–388.

Del Moral, P. (1996). Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581.

Del Moral, P., Doucet, A., and Singh, S. S. (2015). Uniform stability of a particle approximation of the optimal filter derivative. *SIAM Journal on Control and Optimization*, 53(3):1278–1304.

Dellaportas, P. and Pourahmadi, M. (2012). Cholesky-garch models with applications to finance. *Statistics and Computing*, 22(4):849–855.

Dellaportas, P. and Tsionas, M. G. (2018). Importance sampling from posterior distributions using copula-like approximations. *Journal of Econometrics*.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. (2017). Tensorflow distributions. *arXiv preprint arXiv:1711.10604*.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Dissmann, J., Brechmann, E. C., Czado, C., and Kurowicka, D. (2013). Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69.

Douc, R., Moulines, E., and Stoffer, D. (2014). *Nonlinear time series: Theory, methods and applications with R examples*. Chapman and Hall/CRC.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.

Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. (2016). Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564. ACM.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

Duarte, A., Löcherbach, E., and Ost, G. (2016). Stability and perfect simulation of non-linear hawkes processes with erlang kernels. *arXiv preprint arXiv:1610.03300*.

Durmus, A., Guillin, A., and Monmarché, P. (2018). Piecewise deterministic markov processes and their invariant measure. *arXiv preprint arXiv:1807.05421*.

Durmus, A., Moulines, E., and Saksman, E. (2017). On the convergence of hamiltonian monte carlo. *arXiv preprint arXiv:1705.00166*.

Durrande, N., Adam, V., Bordeaux, L., Eleftheriadis, S., and Hensman, J. (2019). Banded matrix operators for gaussian markov models in the automatic differentiation era. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2780–2789. PMLR.

Ernst, J. and Kellis, M. (2015). Large-scale imputation of epigenomic datasets for systematic annotation of diverse human tissues. *Nature biotechnology*, 33(4):364–376.

Fang, K. W. (2017). *Symmetric Multivariate and Related Distributions*. Chapman and Hall/CRC.

Farquhar, S., Smith, L., and Gal, Y. (2020). Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. *arXiv preprint arXiv:2002.03704*.

Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. PhD thesis, Department of Statistics, University of Oxford, UK.

Fearnhead, P., Bierkens, J., Pollock, M., Roberts, G. O., et al. (2018). Piecewise deterministic markov processes for continuous-time monte carlo. *Statistical Science*, 33(3):386–412.

Fearnhead, P. and Clifford, P. (2003). On-line inference for hidden markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899.

Fearnhead, P. and Liu, Z. (2007a). On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605.

Fearnhead, P. and Liu, Z. (2007b). On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605.

Feng, H., Conneely, K. N., and Wu, H. (2014). A bayesian hierarchical model to detect differentially methylated loci from single nucleotide resolution sequencing data. *Nucleic acids research*, 42(8):e69–e69.

Figurnov, M., Mohamed, S., and Mnih, A. (2018). Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pages 441–452.

Finke, A., Doucet, A., and Johansen, A. M. (2016). On embedded hidden markov models and particle markov chain monte carlo methods. *ArXiv e-prints*.

Finke, A., Johansen, A. M., and Spanò, D. (2014). Static-parameter estimation in piecewise deterministic processes using particle gibbs samplers. *Annals of the Institute of Statistical Mathematics*, 66(3):577–609.

Florath, I., Butterbach, K., Müller, H., Bewerunge-Hudler, M., and Brenner, H. (2014). Cross-sectional and longitudinal changes in dna methylation with age: an epigenome-wide analysis revealing over 60 novel age-associated cpg sites. *Human molecular genetics*, 23(5):1186–1201.

Fraccaro, M., Sonderby, S. K., Paquet, U., and Winther, O. (2016). Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, pages 2199–2207.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rdInternational Conference on Machine Learning*.

Garagnani, P., Bacalini, M. G., Pirazzini, C., Gori, D., Giuliani, C., Mari, D., Di Blasio, A. M., Gentilini, D., Vitale, G., Collino, S., et al. (2012). Methylation of elovl 2 gene as a new epigenetic marker of age. *Aging cell*, 11(6):1132–1134.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Genz, A. (1998). Methods for generating random orthogonal matrices. *Monte Carlo and Quasi-Monte Carlo Methods*, pages 199–213.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. (2015). Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889.

Gershman, S. J., Hoffman, M. D., and Blei, D. M. (2012). Nonparametric variational inference. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 235–242. Omnipress.

Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339.

Geweke, J. and Amisano, G. (2010). Comparing and evaluating bayesian predictive distributions of asset returns. *International Journal of Forecasting*, 26(2):216–230.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459.

Ghosh, S. and Doshi-Velez, F. (2017). Model selection in bayesian neural networks via horseshoe priors. *arXiv preprint arXiv:1705.10388*.

Ghosh, S., Yao, J., and Doshi-Velez, F. (2018). Structured variational learning of bayesian neural networks with horseshoe priors. *arXiv preprint arXiv:1806.05975*.

Girolami, M. and Calderhead, B. (2011). Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168.

Goel, N., Karir, P., and Garg, V. K. (2017). Role of dna methylation in human age prediction. *Mechanisms of ageing and development*, 166:33–41.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F, Radar and Signal Processing*, 140(2):107–113.

Goyal, A., Sordoni, A., Côté, M.-A., Ke, N. R., and Bengio, Y. (2017). Z-forcing: Training stochastic recurrent networks. In *Advances in Neural Information Processing Systems*.

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. (2017). Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*.

Gu, S., Ghahramani, Z., and Turner, R. E. (2015). Neural adaptive sequential monte carlo. In *Advances in Neural Information Processing Systems*, pages 2629–2637.

Guarniero, P., Johansen, A. M., and Lee, A. (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association*, pages 1–12.

Guo, F., Wang, X., Fan, K., Broderick, T., and Dunson, D. B. (2016). Boosting variational inference. *arXiv preprint arXiv:1611.05559*.

Haario, H., Saksman, E., and Tamminen, J. (1999). Adaptive proposal distribution for random walk metropolis algorithm. *Computational Statistics*, 14(3):375–395.

Haggstrom, O., Rosenthal, J., et al. (2007). On variance conditions for markov chain clts. *Electronic Communications in Probability*, 12:454–464.

Hairer, E., Lubich, C., and Wanner, G. (2003). Geometric numerical integration illustrated by the störmer–verlet method. *Acta numerica*, 12:399–450.

Han, I., Avron, H., and Shin, J. (2018). Stochastic chebyshev gradient descent for spectral optimization. In *Advances in Neural Information Processing Systems*, pages 7386–7396.

Han, S., Liao, X., Dunson, D., and Carin, L. (2016). Variational gaussian copula inference. In *Artificial Intelligence and Statistics*, pages 829–838.

Hansen, K. D., Langmead, B., and Irizarry, R. A. (2012). Bsmooth: from whole genome bisulfite sequencing reads to differentially methylated regions. *Genome biology*, 13(10):R83.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Hawkes, A. G. (1971a). Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 438–443.

Hawkes, A. G. (1971b). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90.

Heng, J. and Jacob, P. E. (2019). Unbiased hamiltonian monte carlo with couplings. *Biometrika*, 106(2):287–302.

Hirt, M. and Dellaportas, P. (2019). Scalable bayesian learning for state space models using variational inference with smc samplers. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 76–86.

Hirt, M., Dellaportas, P., and Durmus, A. (2019). Copula-like variational inference. In *Advances in Neural Information Processing Systems*, pages 2959–2971.

Hirt, M., Titsias, M., and Dellaportas, P. (2021). Entropy-based adaptive hamiltonian monte carlo. Technical report. Submitted.

Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., and Vasudevan, S. (2019). Neutra-lizing bad geometry in hamiltonian monte carlo using neural transport. *arXiv preprint arXiv:1903.03704*.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

Honkela, A., Raiko, T., Kuusela, M., Tornio, M., and Karhunen, J. (2010). Approximate riemannian conjugate gradient learning for fixed-form variational bayes. *Journal of Machine Learning Research*, 11(Nov):3235–3268.

Horowitz, A. M. (1991). A generalized guided monte carlo algorithm. *Physics Letters B*, 268(2):247–252.

Hron, J., Matthews, D. G., Ghahramani, Z., et al. (2018). Variational bayesian dropout: Pitfalls and fixes. In *35th International Conference on Machine Learning, ICML 2018*, volume 5, pages 3199–3219.

Huggins, J. H., Roy, D. M., et al. (2019). Sequential monte carlo as approximate sampling: bounds, adaptive resampling via
$infty$-ess, and an application to particle gibbs. *Bernoulli*, 25(1):584–622.

Huszár, F. (2017). Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*.

Ingraham, J. and Marks, D. (2017). Variational inference for sparse and undirected models. In *International Conference on Machine Learning*, pages 1607–1616.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. (2021). What are bayesian neural network posteriors really like? *arXiv preprint arXiv:2104.14421*.

Jaakkola, T. and Jordan, M. (1997). A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, volume 82, page 4.

Jacobsen, M. (2006). *Point process theory and applications: marked point and piecewise deterministic processes*. Springer Science & Business Media.

Jacquier, E., Polson, N. G., and Rossi, P. E. (2002). Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, 20(1):69–87.

Jenkinson, G., Abante, J., Feinberg, A. P., and Goutsias, J. (2018). An information-theoretic approach to the modeling and analysis of whole-genome bisulfite sequencing data. *BMC bioinformatics*, 19(1):1–23.

Jenkinson, G., Pujadas, E., Goutsias, J., and Feinberg, A. P. (2017). Potential energy landscapes identify the information-theoretic nature of the epigenome. *Nature genetics*, 49(5):719–729.

Jewson, J., Smith, J. Q., and Holmes, C. (2018). Principles of bayesian inference using general divergence criteria. *Entropy*, 20(6):442.

Johndrow, J. E., Pillai, N. S., and Smith, A. (2020). No free lunch for approximate mcmc. *arXiv preprint arXiv:2010.12514*.

Johnson, L. T. and Geyer, C. J. (2012). Variable transformation to obtain geometric ergodicity in the random-walk metropolis algorithm. *The Annals of Statistics*, 40(6):3050–3076.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kahn, H. and Marshall, A. W. (1953). Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351.

Khaled, M. A. and Kohn, R. (2017). On approximating copulas by finite mixtures. *arXiv preprint arXiv:1705.10440*.

Khan, M. E., Aravkin, A., Friedlander, M., and Seeger, M. (2013). Fast dual variational inference for non-conjugate latent gaussian models. In *International Conference on Machine Learning*, pages 951–959.

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable bayesian deep learning by weight-perturbation in adam. In *International Conference on Machine Learning*, pages 2616–2625.

Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with arch models. *The review of economic studies*, 65(3):361–393.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

Klass, M., de Freitas, N., and Doucet, A. (2005). Towards practical n^2 monte carlo: The marginal particle filter. In *Proceedings of Uncertainty in Artificial Intelligence*.

Knoblauch, J., Jewson, J., and Damoulas, T. (2019). Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*.

Knoblauch, J., Jewson, J. E., and Damoulas, T. (2018). Doubly robust bayesian inference for non-stationary streaming data with *beta*-divergences. In *Advances in Neural Information Processing Systems*, pages 64–75.

Kosmidis, I. and Karlis, D. (2016). Model-based clustering using copulas with applications. *Statistics and computing*, 26(5):1079–1099.

Krishnan, R. G., Shalit, U., and Sontag, D. (2017). Structured inference networks for nonlinear state space models. In *AAAI*, pages 2101–2109.

Krueger, D., Huang, C.-W., Islam, R., Turner, R., Lacoste, A., and Courville, A. (2017). Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474.

Kulis, M. and Esteller, M. (2010). Dna methylation and cancer. In *Advances in genetics*, volume 70, pages 27–56. Elsevier.

Langmore, I., Dikovsky, M., Geraedts, S., Norgaard, P., and Von Behren, R. (2019). A condition number for hamiltonian monte carlo. *arXiv preprint arXiv:1905.09813*.

Lao, J., Suter, C., Langmore, I., Chimisov, C., Saxena, A., Sountsov, P., Moore, D., Saurous, R. A., Hoffman, M. D., and Dillon, J. V. (2020). tfp. mcmc: Modern markov chain monte carlo tools built for modern hardware. *arXiv preprint arXiv:2002.01184*.

Lawson, D., Tucker, G., Naesseth, C. A., Maddison, C. J., Adams, R. P., and Teh, Y. W. (2018). Twisted variational sequential monte carlo. In *Third workshop on Bayesian Deep Learning (NeurIPS 2018)*.

Le, T. A., Igl, M., Jin, T., Rainforth, T., and Wood, F. (2018). Auto-encoding sequential monte carlo. In *ICLR*.

Lee, A., Murray, L. M., and Johansen, A. M. (n.d.). Resampling in conditional smc algorithms. Manuscript in preparation.

Lee, Y., Lim, K. W., and Ong, C. S. (2016). Hawkes processes with stochastic excitations. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 79–88.

Levy, D., Hoffman, M. D., and Sohl-Dickstein, J. (2018). Generalizing hamiltonian monte carlo with neural networks. In *International Conference on Learning Representations*.

Li, Z., Chen, Y., and Sommer, F. T. (2020). A neural network mcmc sampler that maximizes proposal entropy. *arXiv preprint arXiv:2010.03587*.

Lian, W., Henao, R., Rao, V., Lucas, J., and Carin, L. (2015). A multitask point process predictive model. In *Proceedings of the 32st International Conference on Machine Learning (ICML-15), JMLR Workshop and Conference Proceedings*.

Libertini, E., Heath, S. C., Hamoudi, R. A., Gut, M., Ziller, M. J., Czyz, A., Ruotti, V., Stunnenberg, H. G., Frontini, M., Ouwehand, W. H., et al. (2016a). Information recovery from low coverage whole-genome bisulfite sequencing. *Nature Communications*, 7.

Libertini, E., Heath, S. C., Hamoudi, R. A., Gut, M., Ziller, M. J., Herrero, J., Czyz, A., Ruotti, V., Stunnenberg, H. G., Frontini, M., et al. (2016b). Saturation analysis for whole-genome bisulfite sequencing data. *Nature Biotechnology*, 34(7):691–693.

Lin, M. T., Zhang, J. L., Cheng, Q., and Chen, R. (2005). Independent particle filters. *Journal of the American Statistical Association*, 100(472):1412–1421.

Linderman, S. W. and Adams, R. P. (2014). Discovering latent network structure in point process data. In *ICML*, pages 1413–1421.

Linderman, S. W. and Adams, R. P. (2015). Scalable bayesian inference for excitatory point process networks. *arXiv preprint arXiv:1507.03228*.

Lindsten, F. and Schön, T. B. (2013). Backward simulation methods for monte carlo statistical inference. *Foundations and Trends® in Machine Learning*, 6(1):1–143.

Livingstone, S., Betancourt, M., Byrne, S., and Girolami, M. (2019a). On the geometric ergodicity of hamiltonian monte carlo. *Bernoulli*, 25(4A):3109–3138.

Livingstone, S., Faulkner, M. F., and Roberts, G. O. (2019b). Kinetic energy choice in hamiltonian/hybrid monte carlo. *Biometrika*, 106(2):303–319.

Locatello, F., Dresdner, G., Khanna, R., Valera, I., and Rätsch, G. (2018a). Boosting black box variational inference. In *Advances in Neural Information Processing Systems*, pages 3401–3411.

Locatello, F., Khanna, R., Ghosh, J., and Ratsch, G. (2018b). Boosting variational inference: an optimization perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 464–472.

Louizos, C., Ullrich, K., and Welling, M. (2017). Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems*, pages 3290–3300.

Louizos, C. and Welling, M. (2016). Structured and efficient variational deep learning with matrix gaussian posteriors. In *Proceedings of the 33rdInternational Conference on Machine Learning*.

Louizos, C. and Welling, M. (2017). Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*, pages 2218–2227.

Lyne, A.-M., Girolami, M., Atchadé, Y., Strathmann, H., Simpson, D., et al. (2015). On russian roulette estimates for bayesian inference with doubly-intractable likelihoods. *Statistical science*, 30(4):443–467.

Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O. (2016). Auxiliary deep generative models. In *International Conference on Machine Learning*, pages 1445–1453.

Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. (2017). Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6576–6586.

Mai, J.-F. and Scherer, M. (2017). *Simulating copulas: stochastic models, sampling algorithms, and applications*, volume 6. World Scientific.

Mangoubi, O. and Smith, A. (2017). Rapid mixing of hamiltonian monte carlo on strongly log-concave distributions. *arXiv preprint arXiv:1708.07114*.

Martens, J. (2014). New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*.

Martin, J. S., Jasra, A., and McCoy, E. (2013). Inference for a class of partially observed point process models. *Annals of the Institute of Statistical Mathematics*, 65(3):413–437.

Mathieu, M. and LeCun, Y. (2014). Fast approximation of rotations and hessians matrices. *arXiv preprint arXiv:1404.7195*.

Mei, H. and Eisner, J. M. (2017). The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, pages 6757–6767.

Merkel, A., Fernández-Callejo, M., Casals, E., Marco-Sola, S., Schuyler, R., Gut, I. G., and Heath, S. C. (2019). gembs: high throughput processing for dna methylation data from bisulfite sequencing. *Bioinformatics*, 35(5):737–742.

Mescheder, L., Nowozin, S., and Geiger, A. (2017). Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine learning (ICML)*.

Miller, A. C., Foti, N. J., and Adams, R. P. (2017). Variational boosting: Iteratively refining posterior approximations. In *International Conference on Machine Learning*, pages 2420–2429.

Mishkin, A., Kunstner, F., Nielsen, D., Schmidt, M., and Khan, M. E. (2018). Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. In *Advances in Neural Information Processing Systems*, pages 6246–6256.

Moore, D. S. and Spruill, M. C. (1975). Unified large-sample theory of general chi-squared statistics for tests of fit. *The Annals of Statistics*, pages 599–616.

Müller, P., Parmigiani, G., and Rice, K. (2007). Bayesian statistics 8, pp. 349-370. jm bernardo, mj bayarri, jo berger, ap dawid, d. heckerman, afm smith and m. west (eds.)

copyright oxford university press, 2007. In *Bayesian statistics 8: proceedings of the eighth Valencia International Meeting, June 2-6, 2006*, volume 8, page 349. Oxford University Press, USA.

Müller, P., Parmigiani, G., Robert, C., and Rousseau, J. (2004). Optimal sample size for multiple testing: the case of gene expression microarrays. *Journal of the American Statistical Association*, 99(468):990–1001.

Munkhoeva, M., Kapushev, Y., Burnaev, E., and Oseledets, I. (2018). Quadrature-based features for kernel approximation. In *Advances in Neural Information Processing Systems*, pages 9165–9174.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Naesseth, C. A., Linderman, S. W., Ranganath, R., and Blei, D. M. (2018). Variational sequential monte carlo. In *Proceedings of the 21st International Conference on Artificial Intelligence (AISTATS)*.

Naesseth, C. A., Lindsten, F., and Schon, T. B. (2019). Elements of sequential monte carlo. *FOUNDATIONS AND TRENDS IN MACHINE LEARNING*, 12(3):187–306.

Neal, R. M. (2003). Markov chain sampling for non-linear state space models using embedded hidden markov models. *ArXiv Mathematics e-prints*.

Neal, R. M. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Neal, R. M., Beal, M. J., and Roweis, S. T. (2004). Inferring state sequences for non-linear systems with embedded hidden markov models. *Advances in neural information processing systems*, 16:401–408.

Neville, S. E., Ormerod, J. T., Wand, M., et al. (2014). Mean field variational bayes for continuous sparse signal shrinkage: pitfalls and remedies. *Electronic Journal of Statistics*, 8(1):1113–1151.

Ogata, Y. (1981). On lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31.

Olsson, J. and Alenlöv, J. W. (2020). Particle-based online estimation of tangent filters with application to parameter estimation in nonlinear state-space models. *Annals of the Institute of Statistical Mathematics*, 72(2):545–576.

Ong, V. M.-H., Nott, D. J., and Smith, M. S. (2018). Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, 27(3):465–478.

Opper, M. and Archambeau, C. (2009). The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792.

Owen, A. B. (2013). Monte carlo theory, methods and examples. *Monte Carlo Theory, Methods and Examples. Art Owen.*

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2019). Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762.*

Papaspiliopoulos, O., Roberts, G. O., and Skold, M. (2003). Non-centred parameterisations for hierarchical models and data augmentation. *Bayesian Statistics*, 7.

Park, J. and Atchadé, Y. (2020). Markov chain monte carlo algorithms with sequential proposals. *Statistics and Computing*, 30(5):1325–1345.

Park, J.-L., Kim, J. H., Seo, E., Bae, D. H., Kim, S.-Y., Lee, H.-C., Woo, K.-M., and Kim, Y. S. (2016). Identification and evaluation of age-correlated dna methylation markers for forensic use. *Forensic Science International: Genetics*, 23:64–70.

Park, Y., Figueroa, M. E., Rozek, L. S., and Sartor, M. A. (2014). Methylsig: a whole genome dna methylation analysis pipeline. *Bioinformatics*, 30(17):2414–2422.

Pasarica, C. and Gelman, A. (2010). Adaptively scaling the metropolis algorithm using expected squared jumped distance. *Statistica Sinica*, pages 343–364.

Piironen, J., Vehtari, A., et al. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051.

Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80.

Quiroz, M., Nott, D. J., and Kohn, R. (2018). Gaussian variational approximation for high-dimensional state space models. *arXiv preprint arXiv:1801.07873*.

Rackham, O. J., Langley, S. R., Oates, T., Vradi, E., Harmston, N., Srivastava, P. K., Behmoaras, J., Dellaportas, P., Bottolo, L., and Petretto, E. (2017). A bayesian approach for analysis of whole-genome bisulfite sequencing data identifies disease-associated changes in dna methylation. *Genetics*, 205(4):1443–1458.

Rainforth, T., Kosiorek, A. R., Le, T. A., Maddison, C. J., Igl, M., Wood, F., and Teh, Y. W. (2018). Tighter variational bounds are not necessarily better. *arXiv preprint arXiv:1802.04537*.

Rakyan, V. K., Down, T. A., Balding, D. J., and Beck, S. (2011). Epigenome-wide association studies for common human diseases. *Nature Reviews Genetics*, 12(8):529–541.

Ranganath, R., Gerrish, S., and Blei, D. M. (2014). Black box variational inference. In *AISTATS*, pages 814–822.

Ranganath, R., Tran, D., and Blei, D. M. (2016). Hierarchical variational models. In *International Conference on Machine Learning*.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1530–1538.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1278–1286.

Ricci, J. (2014). *Applied Stochastic Control in High Frequency and Algorithmic Trading*. PhD thesis, University of Toronto.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Roberts, G. O. and Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of applied probability*, 44(2):458–475.

Roberts, G. O., Rosenthal, J. S., et al. (2004). General state space markov chains and mcmc algorithms. *Probability surveys*, 1:20–71.

Roberts, G. O. and Stramer, O. (2002). Langevin diffusions and metropolis-hastings algorithms. *Methodology and computing in applied probability*, 4(4):337–357.

Roberts, G. O., Tweedie, R. L., et al. (1996). Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

Robertson, K. D. (2005). Dna methylation and human disease. *Nature Reviews Genetics*, 6(8):597–610.

Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909.

Salimans, T., Kingma, D. P., Welling, M., et al. (2015). Markov chain monte carlo and variational inference: Bridging the gap. In *ICML*, volume 37, pages 1218–1226.

Scibior, A., Masrani, V., and Wood, F. (2021). Differentiable particle filtering without modifying the forward pass. *arXiv preprint arXiv:2106.10314*.

Shafi, A., Mitrea, C., Nguyen, T., and Draghici, S. (2018). A survey of the approaches for identifying differential methylation using bisulfite sequencing data. *Briefings in bioinformatics*, 19(5):737–753.

Shokoohi, F., Stephens, D. A., Bourque, G., Pastinen, T., Greenwood, C. M., and Labbe, A. (2019). A hidden markov model for identifying differentially methylated sites in bisulfite sequencing data. *Biometrics*, 75(1):210–221.

Sklar, M. (1959). Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231.

Sohl-Dickstein, J., Mudigonda, M., and DeWeese, M. (2014). Hamiltonian monte carlo without detailed balance. In *International Conference on Machine Learning*, pages 719–726.

Sountsov, P., Radul, A., and contributors (2020). Inference gym.

Stockwell, P. A., Chatterjee, A., Rodger, E. J., and Morison, I. M. (2014). Dmap: differential methylation analysis package for rrbs and wgbs data. *Bioinformatics*, 30(13):1814–1822.

Süli, E. and Mayers, D. F. (2003). *An introduction to numerical analysis*. Cambridge university press.

Sun, D., Xi, Y., Rodriguez, B., Park, H. J., Tong, P., Meong, M., Goodell, M. A., and Li, W. (2014). Moabs: model based analysis of bisulfite sequencing data. *Genome biology*, 15(2):1–12.

Sun, S. and Yu, X. (2016). Hmm-fisher: identifying differential methylation using a hidden markov model and fisher's exact test. *Statistical applications in genetics and molecular biology*, 15(1):55–67.

Sun, W., Reich, B. J., Cai, T. T., Guindani, M., and Schwartzman, A. (2015). False discovery control in large-scale spatial multiple testing. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 77(1):59.

Sun, W. and Tony Cai, T. (2009). Large-scale multiple testing under dependence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):393–424.

Swiatkowski, J., Roth, K., Veeling, B., Tran, L., Dillon, J., Snoek, J., Mandt, S., Salimans, T., Jenatton, R., and Nowozin, S. (2020). The k-tied normal distribution: A compact parameterization of gaussian mean field posteriors in bayesian neural networks. In *International Conference on Machine Learning*, pages 9289–9299. PMLR.

Tan, L. S. and Nott, D. J. (2018). Gaussian variational approximation with sparse precision matrices. *Statistics and Computing*, 28(2):259–275.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728.

Titsias, M. and Dellaportas, P. (2019). Gradient-based adaptive markov chain monte carlo. In *Advances in Neural Information Processing Systems*, pages 15704–15713.

Titsias, M. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1971–1979.

Titsias, M. K. and Ruiz, F. (2019). Unbiased implicit variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 167–176.

Tomczak, J. M. and Welling, M. (2016). Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*.

Tran, D., Blei, D., and Airoldi, E. M. (2015). Copula variational inference. In *Advances in Neural Information Processing Systems*, pages 3564–3572.

Tran, D., Ranganath, R., and Blei, D. M. (2017). Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. (2017). Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2624–2633.

Turner, R. E. and Sahani, M. (2011). Two problems with variational expectation maximisation for time-series models. *Bayesian Time series models*, pages 115–138.

Vanetti, P., Bouchard-Côté, A., Deligiannidis, G., and Doucet, A. (2017). Piecewise-deterministic markov chain monte carlo. *arXiv preprint arXiv:1707.05296*.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved r for assessing convergence of mcmc. *Bayesian analysis*, 1(1):1–28.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Wang, Y. and Blei, D. M. (2019). Frequentist consistency of variational bayes. *Journal of the American Statistical Association*, 114(527):1147–1161.

Wang, Y., Ye, X., Zhou, H., Zha, H., and Song, L. (2017). Linking micro event history to macro prediction in point process models. In *Artificial Intelligence and Statistics*, pages 1375–1384.

Wang, Z., Mohamed, S., and Freitas, N. (2013). Adaptive hamiltonian and riemann manifold monte carlo. In *International conference on machine learning*, pages 1462–1470.

Wei, Z., Sun, W., Wang, K., and Hakonarson, H. (2009). Multiple testing in genome-wide association studies via hidden markov models. *Bioinformatics*, 25(21):2802–2808.

Whiteley, N., Andrieu, C., and Doucet, A. (2010). Efficient bayesian inference for switching state-space models using discrete particle markov chain monte carlo methods. *arXiv preprint arXiv:1011.2437*.

Whiteley, N., Johansen, A. M., and Godsill, S. (2011). Monte carlo filtering of piecewise deterministic processes. *Journal of Computational and Graphical Statistics*, 20(1):119–139.

Winn, J. and Bishop, C. M. (2005). Variational message passing. *Journal of Machine Learning Research*, 6(Apr):661–694.

Wu, C., Stoehr, J., and Robert, C. P. (2018). Faster hamiltonian monte carlo by learning leapfrog scale. *arXiv preprint arXiv:1810.04449*.

Xiao, S., Farajtabar, M., Ye, X., Yan, J., Song, L., and Zha, H. (2017a). Wasserstein learning of deep generative point process models. In *Advances in Neural Information Processing Systems*, pages 3247–3257.

Xiao, S., Yan, J., Farajtabar, M., Song, L., Yang, X., and Zha, H. (2017b). Joint modeling of event sequence and time series with attentional twin recurrent neural networks. *arXiv preprint arXiv:1703.08524*.

Yildirim, S., Singh, S. S., and Doucet, A. (2013). An online expectation–maximization algorithm for changepoint models. *Journal of Computational and Graphical Statistics*, 22(4):906–926.

Yin, M. and Zhou, M. (2018). Semi-implicit variational inference. In *International Conference on Machine Learning*, pages 5646–5655.

Yu, T., Wang, H., and Li, J. (2019). Maximum conditional entropy hamiltonian monte carlo sampler. *arXiv preprint arXiv:1910.05275*.

Yu, X. and Sun, S. (2016). Hmm-dm: identifying differentially methylated regions using a hidden markov model. *Statistical applications in genetics and molecular biology*, 15(1):69–81.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. (2018). Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pages 5847–5856.