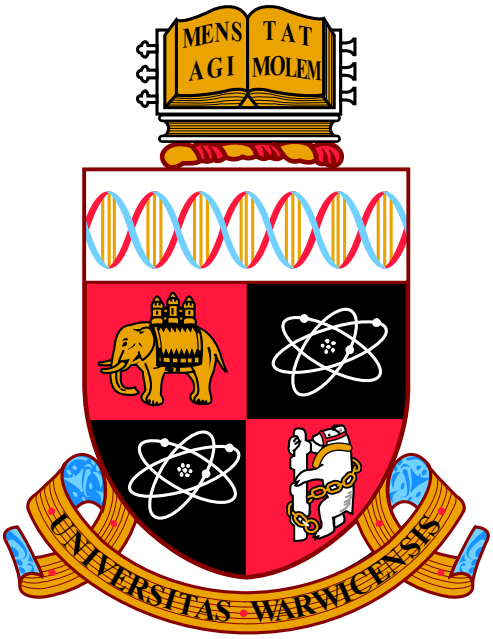


Sequential Monte Carlo variance estimators and global consensus

Lewis James Rendell

Thesis submitted for the degree of
Doctor of Philosophy

University of Warwick
Department of Statistics

June 2020

Contents

Figures

Tables

Acknowledgements

First and foremost I wish to thank my supervisors, Adam M. Johansen and Anthony Lee. The advice, discussions and ideas shared in our supervision meetings have truly enriched my PhD experience; the knowledge that I can count on their support has provided great consolation during the rougher stages of my doctoral journey. I cannot state strongly enough how grateful I am for their guidance, and for all the time they have invested in me.

I also wish to thank Krys Łatuszyński and Nikolas Kantas, for offering their time to read and examine this thesis. Their comments and suggestions have been most helpful in shaping its final form, and I am grateful for the keen interest they have shown in my work.

The work towards this thesis was generously supported by the Engineering and Physical Sciences Research Council (grant number EP/M508184/1).

Completing a PhD has been more difficult than I could ever have imagined, yet would have been immeasurably harder without the help and advice of my fellow doctoral students. Particular thanks go to Kenneth Lim, for his frequent welcome distractions; to David Selby, for being our office's favourite pedant; and to Alejandra Avalos Pacheco, for always suggesting a tea break at exactly the right time.

Finally I wish to thank Mum, Dad, Emily and Jack, who don't quite understand what it is that I've been up to during these past five years, but whose love and support has been invaluable.

Declaration

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree, nor has it been submitted for examination to any institution other than the University of Warwick. The work presented (including data generated and data analysis) was carried out by the author, except where otherwise indicated. Part III contains material also included in the following pre-print, which has been submitted for publication:

Rendell, L. J., Johansen, A. M., Lee, A., and Whiteley, N. (2018). Global consensus Monte Carlo. *arXiv preprint arXiv:1807.09288*.

Lewis James Rendell

Abstract

This thesis makes contributions in two main areas relating to sequential Monte Carlo (SMC) samplers, a class of sequential simulation algorithms used to approximate sequences of probability distributions defined on a common space. Firstly, we consider settings in which one has a single distribution of interest, from which obtaining samples using simple Markov chain Monte Carlo techniques may not be straightforward. We consider the problem of tuning an SMC sampler in this context, selecting an appropriate sequence of distributions to ensure efficient exploration of the space and to control the variance of the resulting estimators. We formalise this as a minimisation problem relating to an asymptotic variance, deriving expressions for a number of relevant quantities and solving this problem for some simple models. We also investigate procedures for selecting such a sequence in practice, utilising recently-proposed methods for cheaply estimating the variances of SMC-based estimators. Secondly, we consider the problem of approximating Bayesian posterior distributions, when these depend on large data sets distributed across multiple computers. Inspired by global variable consensus optimisation, we introduce a novel framework for simulation in distributed settings, proposing a Markov chain Monte Carlo algorithm on an extended state space. Based on the construction of an instrumental hierarchical model, a tuning parameter controls the fidelity to the original model. We also propose the use of these Markov kernels within an SMC sampler. We propose a method for using SMC variance estimators within a bias correction procedure, and propose a stopping rule for the SMC sampler, allowing the automatic selection of the tuning parameter. In contrast to similar distributed Monte Carlo algorithms, this approach requires few distributional assumptions. The performance of the algorithms is illustrated with a number of simulated examples.

Introduction

Context

Since the introduction of the bootstrap particle filter by Gordon et al. (1993), sequential Monte Carlo (SMC) methods have been widely used for the purpose of approximating sequences of recursively-defined probability distributions. Based on sequential importance sampling, SMC algorithms have found application in wide variety of areas, including motion tracking, financial time series analysis and signal processing (see Doucet et al., 2001, for a comprehensive review). Within this framework, Del Moral et al. (2006) proposed a methodology allowing such methods to be used for approximating arbitrary sequences of probability measures defined on some common space. The resulting algorithms are known as sequential Monte Carlo samplers and have been applied to a number of inference settings, which we will explore throughout this work.

Sequential Monte Carlo algorithms possess a number of convenient properties. They do not require the imposition of strict distributional assumptions (cf. the Kalman filter of Kalman, 1960, which is exact only for linear Gaussian models), and many of the resulting estimators have beneficial convergence properties (e.g. central limit theorems). However, they can be computationally expensive, and there are a number of open problems relating to the tuning of such methods. We consider some such problems within this thesis.

Recently-proposed variance estimation procedures for SMC algorithms provide a key motivation for this work. Specifically, a number of authors (Chan and Lai, 2013; Lee and Whiteley, 2018; Olsson and Douc, 2019) have proposed procedures for numerically assessing the Monte Carlo error of estimates formed by such algorithms, as a simple by-product of the execution of the SMC algorithm. Given the simplicity and low computational cost of these variance estimators, there is great potential for these techniques to be used within practical procedures for the tuning of SMC methods.

This thesis considers such tuning problems in two main areas. Firstly, we consider settings in which there is a single distribution of interest from which it may be difficult to draw samples using simple Markov chain Monte Carlo techniques; for example, a distribution with well-separated modes. A common approach in such cases is to construct an SMC sampler to approximate a sequence of distributions ending with this distribution of interest, in order to facilitate exploration of the space. While such constructions have been applied in a number of areas (we later provide a review), the question remains of how best

to choose this sequence in order to control the variance of the resulting estimators. We make a number of contributions to this open problem, including:

- We propose a formalisation of this problem as a transdimensional minimisation problem, of a quantity involving an asymptotic variance;
- We derive expressions for such asymptotic variances in this specific setting;
- Considering the case in which it it is possible to draw independent and identically distributed values from each distribution in the sequence, we derive properties of the solutions to this problem for some commonly-used models, proposing heuristic procedures for use in more realistic settings;
- We present results from investigations into the application of SMC variance estimation procedures to this problem.

Secondly, we consider the problem of approximating a probability distribution dependent on data distributed across multiple machines, a setting that is increasingly common in modern Bayesian inference. The role of communication latency presents serious challenges for standard computational techniques, which require repeated evaluations of the likelihood function. Our contributions in this area include the following:

- We introduce a novel framework for simulation in these distributed settings, based on the construction of an instrumental hierarchical model;
- We propose a distributed Metropolis-within-Gibbs algorithm within this framework, providing guidance on its practical implementation;
- We propose an SMC implementation of the framework and several heuristic procedures for the tuning of the resulting algorithm, including a novel application of SMC variance estimation procedures;
- We provide theoretical and empirical analyses of our proposed algorithms, demonstrating their behaviour in practical settings and comparing their performance with alternative approaches.

Outline

This thesis is divided into three parts, of which Parts II and III contain novel results, methodology and analysis. We provide below a summary of the main themes and results in each chapter. A detailed list of novel contributions is included within the conclusion of the thesis, and can be found on page 180.

Part I serves as a review of sequential Monte Carlo methods and their applications, introducing concepts and notation that will be used throughout the thesis.

- **Chapter 1** reviews the basics of importance sampling and its application to sequential simulation, which we introduce in the context of discrete time Feynman–Kac models. This provides a framework through which we describe the canonical SMC algorithm and some common variants. We also review properties of estimators resulting from these algorithms, and recently-proposed approaches to estimating the variances of these random quantities.

- **Chapter 2** introduces sequential Monte Carlo samplers. We describe their general form and construction within the SMC framework, and review their applications in several common settings.

Part II focuses on the problem of selecting an appropriate sequence of distributions for use in an SMC sampler, when there is a single distribution of primary interest.

- **Chapter 3** describes this problem and reviews some previous approaches to selecting such a sequence. As earlier indicated, we propose a formulation of this problem in terms of the minimisation of a quantity involving an asymptotic variance. To assist later analysis of this optimisation problem we then derive expressions for the asymptotic variances of estimators resulting from SMC algorithms.

- **Chapter 4** provides theoretical analyses of this problem in some perfectly-mixing settings, in which the resulting expressions have tractable forms. We derive some properties of the optimal sequences of distributions for these models, providing several novel results from which we propose heuristics for practical applications.

- **Chapter 5** considers approaches to solving this problem in more realistic settings, in which the Markov kernels used mix more poorly. We demonstrate the behaviour of some relevant quantities and investigate procedures that may be used to select a sequence of distributions automatically, employing the previously-mentioned SMC variance estimators.

Part III considers the problem of simulating from a Bayesian posterior distribution, when this depends on a large data set distributed across multiple computers.

- **Chapter 6** reviews Markov chain Monte Carlo methods and the computational issues with their implementation in such settings, describing a number of alternative approaches that may be advantageous in such cases. We also describe the issue of communication latency when the wall-clock time available for simulation is limited.

- **Chapter 7** introduces a novel framework for simulation in distributed settings. Motivated by ideas from the distributed optimisation literature we describe the construction of an instrumental hierarchical model, from which we propose a Markov chain Monte Carlo algorithm on an extended state space. We describe a number of settings in which our approach may be beneficial, discussing considerations in its implementation and providing a theoretical analysis of its properties when applied to a simple model.

- **Chapter 8** proposes the use of an SMC sampler employing the Markov kernels formed using our framework. Within this context we propose a method for using many of the resulting estimators within a bias correction procedure, which includes an application of SMC variance estimators. We also describe a procedure for determining automatically when to terminate the algorithm, with the aim of achieving a bias–variance trade-off.

- **Chapter 9** presents a number of simulated examples, demonstrating the role of various tuning parameters in our proposed algorithms and the performance of our heuristic procedures for the SMC sampler. We also compare our algorithms with a straightforward Markov chain Monte Carlo approach and some simple embarrassingly parallel approaches, demonstrating regimes in which our framework may result in estimators of lower mean squared error for a fixed time budget.

We conclude by summarising the main contributions of the thesis, and proposing a number of directions for future research.

Notation

We introduce some of the notation that will be used throughout this thesis. In the following definitions $(\mathsf{X}, \mathcal{X})$, $(\mathsf{Y}, \mathcal{Y})$ and $(\mathsf{Z}, \mathcal{Z})$ represent generic Polish spaces.

Sets and vectors

The set of real numbers is denoted by $\mathbb{R}$; the set of positive real numbers is denoted by $\mathbb{R}_+$.

For integers a and b, $\{a, \dots, b\}$ denotes the set of integers n such that $a \leq n \leq b$. For any sequence of integer-indexed objects x_n, we denote by $x_{a:b}$ the vector of values $(x_a, \dots, x_b)$; that is, the vector of those x_n for which $a \leq n \leq b$.

For a collection of measurable spaces $(\mathsf{X}_i, \mathcal{X}_i)$, $i \in \{1, \dots, n\}$, we denote the product space by $\prod_{i=1}^n \mathsf{X}_i$ and the corresponding product σ-algebra by $\bigotimes_{i=1}^n \mathcal{X}_i$. If $(\mathsf{X}_i, \mathcal{X}_i) = (\mathsf{X}, \mathcal{X})$ for all $i \in \{1, \dots, n\}$, we write X^n and $\mathcal{X}^{\otimes n}$ respectively.

Functions

For any $A \in \mathcal{X}$, $\mathbb{1}_A : \mathsf{X} \to \mathbb{R}$ denotes the indicator function on A; that is, for $x \in \mathsf{X}$,

$$\mathbb{1}_A(x) := \begin{cases} 1 & x \in A, \\ 0 & \text{otherwise.} \end{cases}$$

Without a subscript we define $\mathbb{1} := \mathbb{1}_{\mathsf{X}}$, so that $\mathbb{1} : \mathsf{X} \to \mathbb{R}$ denotes the constant function equal to 1; that is, $\mathbb{1}(x) = 1$ for all $x \in \mathsf{X}$. The identity function, denoted $\mathrm{Id} : \mathsf{X} \to \mathsf{X}$, is defined by $\mathrm{Id}(x) := x$ for all $x \in \mathsf{X}$.

For two $\mathcal{X}$-measurable bounded functions $f, g : \mathsf{X} \to \mathbb{R}$, we denote the pointwise product by $f \cdot g : \mathsf{X} \to \mathbb{R}$, defined such that $(f \cdot g)(x) := f(x)g(x)$ for all $x \in \mathsf{X}$. For $c \in \mathbb{R}$, we define $(f + c)(x) = f(x) + c$ and $(cf)(x) = c \cdot f(x)$ for all $x \in \mathsf{X}$.

Measures and kernels

For any σ-finite measure μ with domain $\mathcal{X}$, σ-finite integral kernel K with domain $\mathsf{X} \times \mathcal{Y}$, $\mathcal{X}$-measurable function $\varphi : \mathsf{X} \to \mathbb{R}$ and $\mathcal{Y}$-measurable function $\psi : \mathsf{Y} \to \mathbb{R}$, we define

$$\begin{aligned} \mu(\varphi) &:= \int_{\mathsf{X}} \varphi(x)\mu(\mathrm{d}x), \\ K(\psi)(x) &:= \int_{\mathsf{Y}} K(x, \mathrm{d}y)\psi(y) \quad \text{for } x \in \mathsf{X}, \\ \mu K(A) &:= \int_{\mathsf{X}} \mu(\mathrm{d}x)K(x, A) \quad \text{for } A \in \mathcal{Y}, \\ \varphi K(x, A) &:= \varphi(x)K(x, A) \qquad \text{for } x \in \mathsf{X},\ A \in \mathcal{Y}. \end{aligned}$$

We allow the obvious extensions of these definitions to cases in which the functions φ and ψ have codomain $\mathbb{R}^d$, where d is a positive integer. For a σ-finite integral kernel L with domain $\mathsf{Y} \times \mathcal{Z}$, the composition of K with L is denoted by

$$KL(x, B) := \int_{\mathsf{Y}} K(x, \mathrm{d}y)L(y, B) \quad \text{for } x \in \mathsf{X},\ B \in \mathcal{Z}.$$

For any signed measure π and σ-finite measure μ, we write $\pi \ll \mu$ if π is absolutely continuous with respect to μ; that is, for all $A \in \mathcal{X}$, if $\mu(A) = 0$ then $\pi(A) = 0$. We denote by $\mathrm{d}\pi/\mathrm{d}\mu$ the corresponding Radon–Nikodym derivative of π with respect to μ. This is an $\mathcal{X}$-measurable function taking values in the extended real numbers, with the property that for any $A \in \mathcal{X}$,

$$\int_A \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\mu(\mathrm{d}x) = \pi(A).$$

By the Radon–Nikodym theorem this exists when $\pi \ll \mu$, and is unique up to a μ-null set (Shiryaev, 1996, page 196).

The Dirac measure at $x \in \mathsf{X}$ is denoted $\delta_x : \mathcal{X} \longrightarrow [0,1]$, and defined by $\delta_x(A) := \mathbb{1}_A(x)$ for all $A \in \mathcal{X}$. The identity kernel, denoted $\mathrm{Id} : \mathsf{X} \times \mathcal{X} \longrightarrow [0,1]$, is such that $\mathrm{Id}(x, \cdot) := \delta_x(\cdot)$ for all $x \in \mathsf{X}$.

Distributions and densities

For $\mu \in \mathbb{R}^d$ and some positive definite matrix $\Sigma \in \mathbb{R}^{d \times d}$, we denote by $\mathcal{N}(\mu, \Sigma)$ the normal distribution on $\mathbb{R}^d$ with mean μ and covariance matrix Σ. This distribution admits a density with respect to the Lebesgue measure that we shall denote by

$$\mathcal{N}(x; \mu, \Sigma) := \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^\mathsf{T}\Sigma^{-1}(x-\mu)\right), \qquad x \in \mathbb{R}^d.$$

For a positive integer n and a vector of non-negative values $p_{1:n}$ with $\sum_{i=1}^n p_i > 0$, $\mathrm{Categorical}(p_{1:n})$ denotes the categorical distribution over $\{1, \dots, n\}$ with probabilities proportional to $p_{1:n}$. That is to say, the probability mass of this distribution at $i \in \{1, \dots, n\}$ is $p_i / \sum_{j=1}^n p_j$.

Part I.

A review of sequential Monte Carlo

1. Sequential Monte Carlo methods

1.1. Importance sampling

Before introducing the sequential simulation algorithms that form the focus of this thesis, we begin by reviewing the fundamental concepts from which they are derived. The term 'Monte Carlo methods' describes a class of algorithms that employ random sampling for the purpose of estimating fixed quantities. In most non-trivial settings, the quantities of interest may be expressed as integrals with respect to some probability distribution; to this end we may see the role of a Monte Carlo method as forming an empirical measure approximating this distribution.

Suppose π is a probability measure defined on a measurable space $(\mathsf{X}, \mathcal{X})$, for which one looks to estimate the integral $\pi(\varphi) = \int_{\mathsf{X}} \varphi(x)\pi(\mathrm{d}x)$ for some $\mathcal{X}$-measurable function φ; that is, the expected value of $\varphi(X)$ when X is distributed according to π. If one can simulate observations of random variables $(X^i)_{i=1}^N$ that are independent and identically distributed (IID) according to π, then an approximation of π may be constructed as

$$\pi^N := \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}. \tag{1.1}$$

This empirical measure places equal probability mass at each simulated value; the canonical Monte Carlo estimator of $\pi(\varphi)$ is then obtained as the integral

$$\pi^N(\varphi) = \frac{1}{N} \sum_{i=1}^{N} \varphi(X^i).$$

Such estimators exhibit a number of useful properties: for example, $\pi^N(\varphi)$ is unbiased as an estimator of $\pi(\varphi)$, and obeys a strong law of large numbers. However, the main appeal of Monte Carlo estimators of integrals is their rate of convergence. If $\varphi(X)$ has finite variance when $X \sim \pi$ then a central limit theorem (CLT) holds, from which it follows that $\pi^N(\varphi)$ converges to $\pi(\varphi)$ at rate $\mathcal{O}(N^{-1/2})$, independently of the dimension of X. This stands in contrast to estimates formed using numerical quadrature approaches; as a function of the number of evaluations of the integrand, such estimates generally converge at a rate that decreases with the dimension of the space (see e.g. Kuo and Sloan, 2005).

A limitation of this approach is that in many settings of interest, producing IID samples from the distribution of interest π is computationally infeasible. In such cases *importance*

sampling provides a means of forming an empirical measure approximating π, by drawing samples from another measure μ defined on the same space. In the most general setting we require π to be absolutely continuous with respect to μ; in this case the Radon–Nikodym derivative of π with respect to μ is defined (uniquely, up to a μ-null set) and we may write

$$\pi(A) = \int_A \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\mu(\mathrm{d}x). \tag{1.2}$$

The distribution μ, termed the *importance distribution*, is chosen in order that observations of IID random variables $(X^i)_{i=1}^N$ may readily be simulated. An importance sampling approximation of π may then be formed using a Monte Carlo approximation of (1.2), as

$$\pi^N := \frac{1}{N}\sum_{i=1}^N w(X^i)\delta_{X^i}, \tag{1.3}$$

where $w := \mathrm{d}\pi/\mathrm{d}\mu$. The evaluations $w(X^i)$ of this Radon–Nikodym derivative are known as *importance weights*. If π and μ admit densities with respect to a common dominating measure, then $\mathrm{d}\pi/\mathrm{d}\mu$ is simply the ratio of these densities.

Having simulated IID random variables $(X^i)_{i=1}^N$ according to μ, the computation of estimators $\pi^N(\varphi)$ requires the evaluation of these importance weights. However, it may only be feasible to evaluate the Radon–Nikodym derivative up to some multiplicative normalising constant, of which computation is intractable. This is the case when one can only evaluate some unnormalised density $\bar{\pi}$ of π, such that $\pi(A) = \int_A \bar{\pi}(x)\,\mathrm{d}x/Z$ for all $A \in \mathcal{X}$, and the integral $Z := \int_{\mathsf{X}} \bar{\pi}(x)\,\mathrm{d}x$ is unknown.

By abuse of notation, henceforth let $\bar{\pi}$ also denote the measure given by $\bar{\pi}(A) = \int_A \bar{\pi}(x)\,\mathrm{d}x$ for $A \in \mathcal{X}$, so that $\pi = \bar{\pi}/Z$. An approximation of $\bar{\pi}$ may be formed analogously to (1.3) as

$$\bar{\pi}^N := \frac{1}{N}\sum_{i=1}^N \bar{w}(X^i)\delta_{X^i}, \tag{1.4}$$

where $\bar{w} := \mathrm{d}\bar{\pi}/\mathrm{d}\mu$, so that the importance weights may be evaluated using the ratio of the (unnormalised) densities of $\bar{\pi}$ and μ. Since $Z = \pi(\mathsf{X})$, this normalising constant may be estimated as

$$Z^N := \bar{\pi}^N(\mathsf{X}) = \frac{1}{N}\sum_{i=1}^N \bar{w}(X^i). \tag{1.5}$$

Taking the ratio of (1.4) and (1.5) gives a *self-normalised importance sampling* approximation of π, as the weighted empirical measure

$$\pi^N := \frac{\sum_{i=1}^N W^i \delta_{X^i}}{\sum_{i'=1}^N W^{i'}}, \tag{1.6}$$

where $W^i := \bar{w}(X^i)$. While estimators $\pi^N(\varphi)$ formed using (1.6) are biased, they are consistent in the number of samples N and obey a CLT (see Geweke, 1989, for a summary of

these and other properties).

Importance sampling techniques are ubiquitous in statistical inference, forming the basic elements of many common Monte Carlo methods. Indeed, as shown by Finke (2015), a wide range of such simulation algorithms may be viewed as applications of importance sampling, to approximate appropriately-defined distributions on an extended space.

1.2. Sequential importance sampling

The Monte Carlo methods presented so far may be used to approximate a single distribution of interest π. However, Monte Carlo techniques are also commonly employed to approximate each in a sequence of recursively-defined probability measures, which may not admit closed-form expressions. Such sequential simulation algorithms are collectively known as *sequential Monte Carlo* (SMC) methods, and are the primary focus of this work.

In this chapter we will introduce the canonical SMC algorithm and discuss the properties of the resulting particle approximations. To facilitate this exposition we first describe a general framework for recursively defining a sequence of probability measures, introducing notation that will be used throughout this thesis.

1.2.1. Discrete time Feynman–Kac models

For some non-negative integer n, consider a collection of measurable spaces $(\mathsf{X}_p, \mathcal{X}_p)$, $p \in \{0, \dots, n\}$. Let $M_0 : \mathcal{X}_0 \longrightarrow [0, 1]$ be a probability measure on $(\mathsf{X}_0, \mathcal{X}_0)$, and let $(M_p)_{p=1}^n$ be a sequence of Markov kernels, such that each $M_p : \mathsf{X}_{p-1} \times \mathcal{X}_p \longrightarrow [0, 1]$ is a Markov kernel from $(\mathsf{X}_{p-1}, \mathcal{X}_{p-1})$ to $(\mathsf{X}_p, \mathcal{X}_p)$. Additionally, define a sequence of functions $(G_p)_{p=0}^{n-1}$, such that each $G_p : \mathsf{X}_p \longrightarrow [0, \infty)$ is a non-negative bounded $\mathcal{X}_p$-measurable function; these shall henceforth be referred to as *potential functions*.

Define a sequence of measures $(\gamma_p)_{p=0}^n$ such that $\gamma_0 := M_0$ is a measure on $(\mathsf{X}_0, \mathcal{X}_0)$, and for $p \in \{1, \dots, n\}$, γ_p is a measure on $(\mathsf{X}_p, \mathcal{X}_p)$ defined recursively by

$$\gamma_p(A) := \int_{\mathsf{X}_{p-1}} \gamma_{p-1}(\mathrm{d}x) G_{p-1}(x) M_p(x, A), \quad A \in \mathcal{X}_p. \tag{1.7}$$

We make the following assumption about the supports of the potential functions.

Assumption 1.1. For each $p \in \{0, \dots, n-1\}$, $\gamma_p(G_p) > 0$.

A sufficient condition for this is to hold that each potential function G_p is strictly positive. Under Assumption 1.1, $\gamma_p(\mathsf{X}_p) > 0$ for all $p \in \{0, \dots, n\}$, and so one may normalise each γ_p to obtain a sequence of probability measures $(\eta_p)_{p=0}^n$. That is, for each $p \in \{0, \dots, n\}$, define

$$\eta_p := \frac{\gamma_p}{\gamma_p(\mathsf{X}_p)}. \tag{1.8}$$

The sequences $(\gamma_p)_{p=0}^n$ and $(\eta_p)_{p=0}^n$ are respectively referred to as the unnormalised and normalised *prediction Feynman–Kac models* associated with $(G_p)_{p=0}^{n-1}$ and $(M_p)_{p=0}^n$ (Del Moral, 2004).

The normalising constant $\gamma_p(\mathsf{X}_p)$ in (1.8) may be expressed in terms of the normalised models and potential functions as

$$\gamma_p(\mathsf{X}_p) = \prod_{q=0}^{p-1} \eta_q(G_q),$$

which follows inductively from the observations that $\gamma_0(\mathsf{X}_0) = 1$, and

$$\gamma_p(\mathsf{X}_p) = \eta_{p-1}(G_{p-1})\gamma_{p-1}(\mathsf{X}_{p-1}).$$

It further follows that the unnormalised models may be expressed as

$$\gamma_p = \left(\prod_{q=0}^{p-1} \eta_q(G_q)\right) \eta_p. \tag{1.9}$$

One may define a further sequence of measures $(\hat{\gamma}_p)_{p=0}^{n-1}$, where each $\hat{\gamma}_p$ is a measure on $(\mathsf{X}_p, \mathcal{X}_p)$ defined by

$$\hat{\gamma}_p(A) \coloneqq \int_A G_p(x)\gamma_p(\mathrm{d}x), \quad A \in \mathcal{X}_p. \tag{1.10}$$

Assumption 1.1 ensures that $\hat{\gamma}_p(\mathsf{X}_p) > 0$ for all $p \in \{0, \dots, n-1\}$, and so analogously to the prediction models we may define for each $p \in \{0, \dots, n-1\}$

$$\hat{\eta}_p \coloneqq \frac{\hat{\gamma}_p}{\hat{\gamma}_p(\mathsf{X}_p)}. \tag{1.11}$$

The sequences $(\hat{\gamma}_p)_{p=0}^{n-1}$ and $(\hat{\eta}_p)_{p=0}^{n-1}$ are, respectively, the unnormalised and normalised *updated Feynman–Kac models* associated with $(G_p)_{p=0}^{n-1}$ and $(M_p)_{p=0}^n$. These definitions may be extended to the case $p = n$ in the case that an additional potential function is defined. That is to say, if we have a non-negative bounded $\mathcal{X}_n$-measurable function $G_n : \mathsf{X}_n \longrightarrow [0, \infty)$, then one may analogously define $\hat{\gamma}_n$ and, if Assumption 1.1 holds for $p = n$, $\hat{\eta}_n$.

Feynman–Kac models find application in such varied fields as particle physics, engineering science and industrial chemistry (see Del Moral, 2004 and references therein). Alongside the sequential simulation problems we shall go on to consider, within statistics they are important in the context of hidden Markov models. The application of such models to filtering problems is widespread in such areas as signal processing and time series analysis; a review is provided by Murphy (2012, Sections 17.3.1, 18.2, 23.5).

1.2.2. Particle approximations

Owing to the recursive definition (1.7) of γ_p as earlier mentioned, Feynman–Kac models do not typically admit closed-form expressions. A notable exception is the case in which $(G_p)_{p=0}^{n-1}$ and $(M_p)_{p=0}^{n}$ describe a linear Gaussian model, so that all the resulting measures are Gaussian and may be computed exactly using the Kalman filter (Kalman, 1960). While Feynman–Kac models may be approximated via appropriate functional approximations (the extended Kalman filter, for example, produces Gaussian approximations using a linearisation technique), Monte Carlo methods provide a more widely-applicable approach.

We may employ importance sampling by considering the problem of approximating an appropriate measure on the product space $\tilde{\mathsf{X}} := \prod_{p=0}^{n} \mathsf{X}_p$. For some set $\tilde{A}$ belonging to the product σ-algebra $\tilde{\mathcal{X}} := \bigotimes_{p=0}^{n} \mathcal{X}_p$, consider the probability measure defined by $\tilde{\eta}_n(\tilde{A}) := \tilde{\gamma}_n(\tilde{A})/\tilde{\gamma}_n(\tilde{\mathsf{X}})$, where

$$\tilde{\gamma}_n(\tilde{A}) := \int_{\tilde{A}} M_0(\mathrm{d}x_0) \prod_{p=1}^{n} G_{p-1}(x_{p-1}) M_p(x_{p-1}, \mathrm{d}x_p). \tag{1.12}$$

This probability measure is seen to admit η_n as a marginal distribution.

To draw samples from an importance distribution, one can sample a value ζ_0 from the probability measure M_0, and draw each successive ζ_p by applying each Markov kernel M_p in turn. This is equivalent to sampling the whole 'path' $(\zeta_0, \ldots, \zeta_n)$ from the probability measure given by

$$\mu_n(\tilde{A}) := \int_{\tilde{A}} M_0(\mathrm{d}x_0) \prod_{p=1}^{n} M_p(x_{p-1}, \mathrm{d}x_p).$$

For computing the necessary importance weights, it may be seen that the Radon–Nikodym derivative of the unnormalised measure $\tilde{\gamma}_n$ with respect to μ_n is the pointwise product of the potential functions $(G_p)_{p=0}^{n-1}$. We may thereby form a self-normalised importance sampling approximation of $\tilde{\eta}_n$ according to (1.6); an approximation of the Feynman–Kac model η_n is obtained as the marginal on X_n of the resulting weighted empirical measure.

This suggests the use of a sequential procedure for generating estimates of each η_p. In the case $p = 0$, an approximation of η_0 is produced by sampling a number of *particles* $(\zeta_0^i)_{i=1}^N$ from M_0, and forming a Monte Carlo approximation of the form (1.1). This may be seen as a weighted empirical measure of the form (1.6), in which each particle has an initial weight $W_0^i = 1$. One continues sequentially: for each $p \in \{1, \ldots, n\}$, the next 'generation' of particles is sampled as $\zeta_p^i \sim M_p(\zeta_{p-1}^i, \cdot)$. The weight W_p^i of each particle is computed as the product of the previous weight W_{p-1}^i and the *incremental weight* given by $G_{p-1}(\zeta_{p-1}^i)$. A weighted empirical measure approximating η_p may then be constructed according to (1.6).

This procedure is known as *sequential importance sampling* (SIS), and forms the simplest SMC method. We present this as Algorithm 1.1; a detailed summary of the algorithm and its properties is provided by Doucet and Johansen (2011).

Algorithm 1.1 Sequential importance sampling

1. At time $p = 0$:
   - For $i \in \{1, \dots, N\}$ set $W_0^i \longleftarrow 1$ and independently sample $\zeta_0^i \sim M_0(\cdot)$.
2. At time $p = 1, \dots, n$:
   - For $i \in \{1, \dots, N\}$ set $W_p^i \longleftarrow W_{p-1}^i G_{p-1}(\zeta_{p-1}^i)$ and independently sample $\zeta_p^i \sim M_p(\zeta_{p-1}^i, \cdot)$.

We define

$$\eta_n^N := \frac{\sum_{i=1}^N W_n^i \delta_{\zeta_n^i}}{\sum_{i'=1}^N W_n^{i'}}$$

as the weighted empirical measure (1.6) approximating η_n. Since this approximation is formed via self-normalised importance sampling we may also estimate the unnormalised model γ_n according to (1.4), as

$$\gamma_n^N := \frac{1}{N} \sum_{i=1}^N W_n^i \delta_{\zeta_n^i} = \left(\frac{1}{N} \sum_{i=1}^N W_n^i \right) \eta_n^N.$$

An estimator of the normalising constant $\gamma_n(\mathsf{X}_n)$ is thereby obtained as $\gamma_n^N(\mathsf{X}_n) = \sum_{i=1}^N W_n^i / N$, as in (1.5).

For the updated measures, note that for any $\mathcal{X}_n$-measurable bounded function $\varphi : \mathsf{X}_n \longrightarrow \mathbb{R}$, one has the following relationships (Del Moral, 2004, page 60):

$$\hat{\gamma}_n(\varphi) = \gamma_n(\varphi \cdot G_n), \qquad \hat{\eta}_n(\varphi) = \frac{\eta_n(\varphi \cdot G_n)}{\eta_n(G_n)}. \tag{1.13}$$

By replacing γ_n and η_n by their particle approximations, one obtains the appropriate particle approximations of the given functionals. For $A \in \mathcal{X}_n$, estimators for $\hat{\gamma}_n(A)$ and $\hat{\eta}_n(A)$ may be obtained by taking $\varphi = \mathbb{1}_A$, the indicator function on A.

1.3. Resampling

A significant problem with SIS as presented in Algorithm 1.1 is that the algorithm generally exhibits degeneracy in practice. In a great number of settings the variance of estimators such as $\eta_p^N(\varphi)$ increases with p, due to increasing variability in the weights W_p^i (see for example Doucet et al., 2000, Proposition 1 for a hidden Markov model setting). If a small number of these weights take relatively large values, then the empirical measure η_p^N places most of its mass on the corresponding particles. The number of particles making a non-negligible contribution to such estimators is therefore much smaller than the true sample size N. This notion may be formalised; we discuss this in Section 1.3.2.

A solution to this problem uses a resampling technique, rejuvenating the set of particles by stochastically replicating those with the highest weights and removing those with the lowest weights. Formally, following the computation of each particle's updated weight (by the multiplication of its previous weight by its incremental weight), a new set of N particles is sampled with replacement from the current set. For this resampling step, a requirement is imposed that the expected number of replicates of each particle in the new set is proportional to its updated weight. Having generated this resampled set of particles, each such particle is assigned an equal weight of 1.

The resulting procedure is known as *sequential importance sampling with resampling*. The use of resampling in every iteration results in the canonical SMC algorithm, which we present as Algorithm 1.2. In this case the collection of particles obtained following each iteration is equally-weighted, and so the resampling step only requires consideration of the incremental weights $(G_{p-1}(\zeta_{p-1}^i))_{i=1}^N$.

We present this resampling step in terms of the sampling of random indices A_{p-1}^i. The particle $\zeta_{p-1}^{A_{p-1}^i}$ may be interpreted as the *ancestor* of the particle ζ_p^i, so that $(\zeta_{p-1}^{A_{p-1}^i})_{i=1}^N$ is the set of particles obtained by resampling from $(\zeta_{p-1}^i)_{i=1}^N$.

Algorithm 1.2 Sequential importance sampling with resampling

1. At time $p = 0$:
   - For $i \in \{1, \dots, N\}$ independently sample $\zeta_0^i \sim M_0(\cdot)$.
2. At time $p = 1, \dots, n$:
   - For $i \in \{1, \dots, N\}$ independently sample
   $$A_{p-1}^i \sim \text{Categorical}\big(G_{p-1}(\zeta_{p-1}^1), \dots, G_{p-1}(\zeta_{p-1}^N)\big)\,.$$
   - For $i \in \{1, \dots, N\}$ independently sample $\zeta_p^i \sim M_p(\zeta_{p-1}^{A_{p-1}^i}, \cdot)$.

Here, each ancestor index A_{p-1}^i is sampled independently from the appropriate categorical distribution on $\{1, \dots, N\}$; that is, A_{p-1}^i takes the value $j \in \{1, \dots, N\}$ with probability proportional to $G_{p-1}(\zeta_{p-1}^j)$. This resampling scheme is known as *multinomial resampling* and was first proposed within the 'bootstrap particle filter' of Gordon et al. (1993), a form of Algorithm 1.2 for approximating filtering distributions of hidden Markov models. Other schemes are possible (see Douc et al., 2005; Gerber et al., 2019, for a summary of some schemes and their properties), though within this thesis we shall solely consider the use of multinomial resampling. The use of this simple scheme facilitates analysis of the algorithm, validating the results and estimators we shall introduce in Sections 1.4 and 1.5.

For the setting of Algorithm 1.2 in which resampling takes place in every time step, the

particle approximations of η_n and γ_n are given by

$$\eta_n^N := \frac{1}{N}\sum_{i=1}^{N}\delta_{\zeta_n^i}, \qquad \gamma_n^N := \left(\prod_{p=0}^{n-1}\eta_p^N(G_p)\right)\eta_n^N. \tag{1.14}$$

The latter is seen to take a form directly comparable with (1.9); re-expressing this as

$$\gamma_n^N = \left(\prod_{p=0}^{n-1}\frac{1}{N}\sum_{i=1}^{N}G_p(\zeta_p^i)\right)\eta_n^N, \tag{1.15}$$

we see that this may be computed by storing the evaluations of the potential functions, as used in each resampling step.

1.3.1. Occasional resampling

As noted, the purpose of resampling the particles is to prevent the estimators resulting from SIS from becoming dominated by a small subset of the particles, due to high variance of the importance weights. However, resampling the particles carries a computational cost, and indeed introduces some additional variance. This may be observed in the central limit theorem of Chopin (2004), for example, in which each resampling step contributes a term to the asymptotic variance expression; the same phenomenon may be observed in the expressions we consider in Section 1.4.1.

One may therefore choose to conduct resampling in only a subset of the n iterations of Algorithm 1.2. The result is that one maintains a collection of weighted particles as in SIS, with the particles only being equally-weighted following those iterations in which resampling is used. The subset of iterations in which to resample could be chosen in advance (i.e. deterministically), though adaptive approaches are common in practice, as will be discussed in Section 1.3.2.

An SMC algorithm using such 'occasional resampling' is presented as Algorithm 1.3. By resampling in every iteration one recovers Algorithm 1.2; choosing never to resample results in the form of SIS presented in Algorithm 1.1.

Using notation from Del Moral et al. (2006), let r_n be the number of times resampling occurs between times 0 and n. For $j \in \{1, \ldots, r_n\}$ (which may be empty), let k_j denote the time index at which the jth resampling step occurs; additionally define $k_0 := 0$ and $k_{r_n+1} := n+1$. Then the particle approximations of η_n and γ_n are given by

$$\eta_n^N := \frac{\sum_{i=1}^{N} W_n^i \delta_{\zeta_n^i}}{\sum_{i'=1}^{N} W_n^{i'}}, \qquad \gamma_n^N := \left(\prod_{j=1}^{r_n}\frac{1}{N}\sum_{i=1}^{N}\tilde{W}_{k_j}^i\right)\left(\frac{1}{N}\sum_{i=1}^{N}W_n^i\right)\eta_n^N. \tag{1.16}$$

Algorithm 1.3 Sequential importance sampling with occasional resampling

1. At time $p = 0$:
   - For $i \in \{1, \dots, N\}$ set $W_0^i \leftarrow 1$ and independently sample $\zeta_0^i \sim M_0(\cdot)$.
2. At time $p = 1, \dots, n$,
   - For $i \in \{1, \dots, N\}$ set $\tilde{W}_p^i \leftarrow W_{p-1}^i G_{p-1}(\zeta_{p-1}^i)$.
   - If resampling in the pth iteration:
     - For $i \in \{1, \dots, N\}$ independently sample $A_{p-1}^i \sim \text{Categorical}\big(\tilde{W}_p^1, \dots, \tilde{W}_p^N\big)$ and set $W_p^i \leftarrow 1$.
   - Else:
     - For $i \in \{1, \dots, N\}$ set $A_{p-1}^i \leftarrow i$ and set $W_p^i \leftarrow \tilde{W}_p^i$.
   - For $i \in \{1, \dots, N\}$ independently sample $\zeta_p^i \sim M_p(\zeta_{p-1}^{A_{p-1}^i}, \cdot)$.

These may be expressed explicitly in terms of the potential functions as

$$\eta_n^N = \frac{\sum_{i=1}^N \left[\prod_{p=k_{r_n}}^{n-1} G_p(\zeta_p^i)\right] \delta_{\zeta_n^i}}{\sum_{i'=1}^N \left[\prod_{p=k_{r_n}}^{n-1} G_p(\zeta_p^{i'})\right]}, \tag{1.17}$$

$$\gamma_n^N = \left(\prod_{j=0}^{r_n-1} \frac{1}{N} \sum_{i=1}^N \left[\prod_{p=k_j}^{k_{j+1}-1} G_p(\zeta_p^i)\right]\right) \left(\frac{1}{N} \sum_{i=1}^N \left[\prod_{p=k_{r_n}}^{n-1} G_p(\zeta_p^i)\right]\right) \eta_n^N. \tag{1.18}$$

1.3.1.1. Excursion Feynman–Kac models

The particle approximations (1.14) formed when resampling in every iteration are a special case of the more general forms (1.17)–(1.18). However, we also notice that (1.17)–(1.18) may be viewed as instances of the 'always resampling' particle approximations (1.14). This is clearest in the case $k_{r_n} = n$; that is, when resampling occurs in the nth step. In this case, η_n^N is an unweighted empirical measure. Regarding γ_n^N, the last factor in (1.18) evaluates to 1; comparing with (1.15), we see that the n individual potential functions G_p are replaced by r_n products of such functions between resampling steps.

Indeed, suppose the times at which resampling takes place are chosen deterministically. Then we may view Algorithm 1.3 as an instance of Algorithm 1.2, by considering the tensor products of Markov kernels applied between resampling points, and the corresponding products of potential functions. We may thereby assert that results for particle approximations generated by the simpler Algorithm 1.2 will also hold for those generated using other deterministic choices of resampling times.

We formalise this as follows. Retaining the stated definitions of r_n and $k_{0:r_n+1}$, for each $j \in \{0, \dots, r_n\}$ define the *excursion space* $\mathsf{X}_j' := \prod_{p=k_j}^{k_{j+1}-1} \mathsf{X}_p$, with corresponding product σ-algebra $\mathcal{X}_j' := \bigotimes_{p=k_j}^{k_{j+1}-1} \mathcal{X}_p$. In the following descriptions, an arbitrary element of X_j' shall

be denoted by $x'_j := (x_{k_j}, \dots, x_{k_{j+1}-1})$.

Define a probability measure M'_0 on $(\mathsf{X}'_0, \mathcal{X}'_0)$ by

$$M'_0(\mathrm{d}x'_0) := M_0(\mathrm{d}x_0) \prod_{p=1}^{k_1-1} M_p(x_{p-1}, \mathrm{d}x_p).$$

For $j \in \{1, \dots, r_n\}$, let M'_j be a Markov kernel from $(\mathsf{X}'_{j-1}, \mathcal{X}'_{j-1})$ to $(\mathsf{X}'_j, \mathcal{X}'_j)$ defined by

$$M'_j(x'_{j-1}, \mathrm{d}x'_j) := \prod_{p=k_j}^{k_{j+1}-1} M_p(x_{p-1}, \mathrm{d}x_p). \tag{1.19}$$

Define also a sequence of non-negative functions $(G'_j)_{j=0}^{r_n}$, where $G'_j : \mathsf{X}'_j \longrightarrow [0, \infty)$ is given by

$$G'_j(x'_j) := \prod_{p=k_j}^{k_{j+1}-1} G_p(x_p), \quad j \in \{0, \dots, r_n - 1\}; \qquad G'_{r_n}(x'_{r_n}) := \prod_{p=k_{r_n}}^{n-1} G_p(x_p). \tag{1.20}$$

For these measures and potential functions, let $(\gamma'_j)_{j=0}^{r_n}$ and $(\eta'_j)_{j=0}^{r_n}$ be the associated unnormalised and normalised prediction Feynman–Kac models, as defined in (1.7)–(1.8); similarly, let $(\hat{\gamma}'_j)_{j=0}^{r_n}$ and $(\hat{\eta}'_j)_{j=0}^{r_n}$ be the corresponding updated measures, defined by (1.10)–(1.11). Such measures, defined on the sequence of excursion spaces, may be termed *excursion Feynman–Kac models*. Retain $(\gamma_p)_{p=0}^n$ and $(\eta_p)_{p=0}^n$ as notation for the unnormalised and normalised Feynman–Kac models associated with $(G_p)_{p=0}^{n-1}$ and $(M_p)_{p=0}^n$.

To view the connection between these sets of measures, first consider the case in which $k_{r_n} = n$ (corresponding to resampling occurring in the nth step). In this case, the final excursion space is $\mathsf{X}'_{r_n} = \mathsf{X}_n$, so that γ'_{r_n} and η'_{r_n} are measures on $(\mathsf{X}_n, \mathcal{X}_n)$. Indeed, we have

$$\gamma_n = \gamma'_{r_n}, \qquad \eta_n = \eta'_{r_n},$$

which may be shown simply by expressing each measure in terms of integrals, using their definitions (1.7) and (1.8).

In the more general case the final excursion space is $\mathsf{X}'_{r_n} = \prod_{p=k_{r_n}}^n \mathsf{X}_p$; this requires consideration of the updated excursion models defined on this space, in order to account for the particles' final weights when resampling does not take place in the final iteration. For any $A \in \mathcal{X}_n$, let $A' \in \mathcal{X}'_{r_n}$ be defined by $A' := \left[\prod_{p=k_{r_n}}^{n-1} \mathsf{X}_p\right] \times A$. Then we have that

$$\gamma_n(A) = \hat{\gamma}'_{r_n}(A'), \qquad \eta_n(A) = \hat{\eta}'_{r_n}(A'),$$

so that η_n may effectively be seen as the marginal distribution of $\hat{\eta}'_{r_n}$ on X_n.

As previously stated, the consequence is that for deterministic choices of resampling times, particle approximations generated by Algorithm 1.3 may be seen as having been generated by a form of Algorithm 1.2 in which different Markov kernels and potential functions are used. We shall exploit this connection in several of the results that follow.

1.3.2. Adaptive resampling

While the choice of iterations in which to conduct resampling may be pre-determined, it is common in practice for these resampling times to be determined adaptively. That is to say, the decision of whether to conduct resampling during each iteration of Algorithm 1.3 is made only immediately before that resampling step would take place, based on the evolution of the algorithm up to that point. Given that the purpose of resampling is to avoid high variance in the importance weights, it follows that one can choose to resample only when the variance of the current set of importance weights is sufficiently high.

The most widely-used adaptive resampling approach, proposed by Liu and Chen (1995), employs a quantity known as the *effective sample size* (ESS). The name of this quantity relates to the problem of estimating integrals $\pi(\varphi)$, for some probability measure π and function of interest φ. In general, one cannot obtain IID samples distributed according to π, and so one forms an estimator using N samples generated by some other method (e.g. from an importance distribution). The ESS represents the number of IID samples from π that would be required to create a simple Monte Carlo estimator (1.1) of the same variance.

In the setting of importance sampling, consider the approximation (1.6) of some distribution π, formed by drawing N samples from an importance distribution μ. Kong et al. (1994) define the corresponding effective sample size to be

$$\mathrm{ESS} := \frac{N}{1 + \mathrm{var}_\mu(\mathrm{d}\pi/\mathrm{d}\mu)}, \tag{1.21}$$

where $\mathrm{var}_\mu(\mathrm{d}\pi/\mathrm{d}\mu)$ denotes the variance of the given Radon–Nikodym derivative when its argument is distributed according to μ. This expression is motivated by considering two estimators of integrals $\pi(\varphi)$: that formed using this importance sampling procedure, and that using the approximation (1.1), formed using N IID samples distributed according to π. The denominator of (1.21) represents a simple approximation of the ratio of the variances of these estimators, that is independent of φ.

For the self-normalised importance sampling approximation (1.6), the effective sample size (1.21) may be estimated empirically by

$$\mathrm{ESS}^N := \frac{N}{1 + \frac{1}{N}\sum_{i=1}^{N}\left(\frac{N W^i}{\sum_{i'=1}^{N} W^{i'}} - 1\right)^2} = \left[\sum_{i=1}^{N}\left(\frac{W^i}{\sum_{i'=1}^{N} W^{i'}}\right)^2\right]^{-1}. \tag{1.22}$$

As is intuitive, if all the weights W^i were equal (e.g. because $\mu = \pi$), this empirical ESS would equal N. If instead the normalised weights of all but one particle were to tend to 0, the ESS would tend to 1.

In Algorithm 1.3, the updated weight $\tilde{W}_p^i$ of each particle is computed immediately before each (possible) resampling step. The empirical ESS may then be computed according to

(1.22) as

$$\mathrm{ESS}_p^N := \left[\sum_{i=1}^N \left(\frac{\tilde{W}_p^i}{\sum_{i'=1}^N \tilde{W}_p^{i'}}\right)^2\right]^{-1} = \frac{\left(\sum_{i'=1}^N W_{p-1}^{i'} G_{p-1}(\zeta_{p-1}^{i'})\right)^2}{\sum_{i=1}^N \left(W_{p-1}^i G_{p-1}(\zeta_{p-1}^i)\right)^2} \tag{1.23}$$

The proposal of Liu and Chen (1995) is to carry out resampling only when this effective sample size falls below a pre-determined threshold. That is, for some $\tau \in [1, N]$, resampling is conducted in the pth iteration only if $\mathrm{ESS}_p^N < \tau$. A number of convergence results have been derived for this adaptive setting by Del Moral et al. (2012b), by considering the asymptotic behaviour of the resulting sequence of resampling times; we discuss these results and their consequences in Sections 1.4 and 1.5.

Finally, while the ESS-based form of adaptive resampling is by far the most common in the literature, other schemes are certainly possible, by making the decision of whether (and how) to resample based on some generic functional of all previous particle values (Whiteley et al., 2016).

1.4. Properties of particle approximations

The following summarises some key properties of the particle approximations γ_n^N and η_n^N resulting from such SMC algorithms. We present the results below in the context of Algorithm 1.2, in which resampling is always used, and thereafter describe their applicability to the more general setting of occasional resampling.

For all results in this section, we take $\varphi : \mathsf{X}_n \to \mathbb{R}$ to be a bounded $\mathcal{X}_n$-measurable function. We first present an unbiasedness result for particle approximations of unnormalised Feynman–Kac models:

Proposition 1.2. $\mathbb{E}\left[\gamma_n^N(\varphi)\right] = \gamma_n(\varphi)$ *(Del Moral, 2004, Proposition 7.4.1).*

Considering $\varphi = \mathbb{1}$, this implies that the normalising constant $\gamma_n(\mathbb{1}) = \gamma_n(\mathsf{X}_n)$ may be estimated unbiasedly. No such result holds for the normalised models; that is, $\eta_n^N(\varphi)$ is not unbiased as an estimator of $\eta_n(\varphi)$. However, the corresponding estimates are consistent in the number of particles:

Proposition 1.3. *For any $r \geq 1$,*

$$\sup_{N\geq 1} \sqrt{N}\, \mathbb{E}\left[\left|\gamma_n^N(\varphi) - \gamma_n(\varphi)\right|^r\right]^{1/r} < \infty \quad \text{and} \quad \sup_{N\geq 1} \sqrt{N}\, \mathbb{E}\left[\left|\eta_n^N(\varphi) - \eta_n(\varphi)\right|^r\right]^{1/r} < \infty$$

(Del Moral, 2004, Theorems 7.4.2 and 7.4.4 respectively). Therefore, $\gamma_n^N(\varphi)$ and $\eta_n^N(\varphi)$ converge almost surely to $\gamma_n(\varphi)$ and $\eta_n(\varphi)$ respectively as the number of particles N tends to infinity.

A central limit theorem (CLT) also holds for $\gamma_n^N(\varphi)$ and $\eta_n^N(\varphi)$:

Proposition 1.4. *As $N \longrightarrow \infty$, $\sqrt{N}\left(\gamma_n^N(\varphi) - \gamma_n(\varphi)\right)$ and $\sqrt{N}\left(\eta_n^N(\varphi) - \eta_n(\varphi)\right)$ both converge weakly to normal distributions with mean zero and finite variance (Del Moral, 2004, Propositions 9.4.1 and 9.4.2 respectively).*

For all of these propositions, analogous results may be shown to hold for particle approximations $\hat{\gamma}_n^N$ and $\hat{\eta}_n^N$ of the updated Feynman–Kac models by consideration of the identities (1.13) and, in the case of $\hat{\eta}_n^N$, application of results such as Minkowski's inequality. As a consequence, when resampling occurs at deterministically-chosen times the same results may be seen to hold for particle approximations formed by Algorithm 1.3. This follows from the discussion in Section 1.3.1.1, since these may be viewed as particle approximations of suitably-defined (updated) excursion models.

This does not apply when adaptive resampling is employed, in which case the sequence of resampling times is random. In particular the arguments used to prove Proposition 1.2 do not hold, so that when adaptive resampling is used, $\gamma_n^N(\varphi)$ is *not* unbiased as an estimator of $\gamma_n(\varphi)$. However one may prove analogues of Propositions 1.3 and 1.4, which consider asymptotic behaviour in N rather than fixed N. This follows from the results of Del Moral et al. (2012b), who show that under certain assumptions, the sequence of random resampling times converges almost surely to some deterministic sequence as the number of particles tends to infinity. By considering this limiting behaviour one can show that a number of asymptotic results assuming deterministically-chosen resampling also apply in adaptive settings, including the above consistency and CLT results.

1.4.1. Asymptotic variances

In the CLT for $\gamma_n^N(\varphi)$ presented in Proposition 1.4, the variance of the limiting normal distribution is known as the *asymptotic variance* of $\gamma_n^N(\varphi)$ as $N \longrightarrow \infty$, which may be defined as

$$\lim_{N\to\infty} N \operatorname{var}\left(\gamma_n^N(\varphi)\right).$$

In practice, it is often numerically simpler to deal with a 'normalised' form of this asymptotic variance: we define the *relative asymptotic variance* of $\gamma_n^N(\varphi)$ as $N \longrightarrow \infty$ as

$$\lim_{N\to\infty} N \operatorname{var}\left(\frac{\gamma_n^N(\varphi)}{\gamma_n(\mathbb{1})}\right), \tag{1.24}$$

noting again that $\gamma_n(\mathbb{1}) = \gamma_n(\mathsf{X}_n)$ is the normalising constant of γ_n.

For the normalised models, the asymptotic variance of $\eta_n^N(\varphi)$ as $N \longrightarrow \infty$ is similarly defined as

$$\lim_{N\to\infty} N \operatorname{var}\left(\eta_n^N(\varphi)\right). \tag{1.25}$$

By Lee and Whiteley (2018), this may be expressed in the form (1.24) as the relative asymptotic variance as $N \longrightarrow \infty$ of $\gamma_n^N(\varphi - \eta_n(\varphi))$.

The relative asymptotic variance (1.24) admits a decomposition that is convenient for the purposes of analysis. This has been studied by Del Moral (2004, Section 9.4) and Chopin (2004); in the latter case, a collection of recursively-defined terms is derived by applying CLT arguments to the particle set at each iteration of the algorithm, conditional on past iterations. We here summarise the derivation presented by Lee and Whiteley (2018), describing this in the setting that Algorithm 1.2 is used (i.e. resampling takes place in every iteration), so that γ_n^N is of the form (1.15). The generalisation to occasional resampling follows by consideration of the corresponding excursion models, and we discuss this subsequently.

For fixed N, Cérou et al. (2011) provide an expression for the 'non-asymptotic' variance $\mathrm{var}\big(\gamma_n^N(\varphi)/\gamma_n(\mathbb{1})\big)$ as a sum of 2^{n+1} terms. These may be thought of as empirical measures on the joint path space of two particles, conditioned to exhibit a form of coalescence at each in a subset of the $n+1$ time steps, in a manner similar to the doubly conditional particle filter of Andrieu et al. (2018). After multiplying by N and taking the limit as $N \longrightarrow \infty$, all but $n+2$ of these terms vanish. Regrouping these remaining values into $n+1$ terms allows the relative asymptotic variance of $\gamma_n^N(\varphi)$ to be expressed as

$$\lim_{N\to\infty} N \,\mathrm{var}\left(\frac{\gamma_n^N(\varphi)}{\gamma_n(\mathbb{1})}\right) = \sum_{p=0}^{n} v_{p,n}(\varphi). \tag{1.26}$$

Each term $v_{p,n}(\varphi)$ is formed in part from a measure on the joint path space of two Feynman–Kac models conditioned to exhibit interdependence at the pth time step.

Expressing each $v_{p,n}(\varphi)$ in terms of the potential functions $(G_p)_{p=0}^{n-1}$ and Markov kernels $(M_p)_{p=1}^{n}$ is facilitated by first defining a collection of integral kernels $(Q_p)_{p=1}^{n}$, where for $p \in \{1,\dots,n\}$, $Q_p : \mathsf{X}_{p-1} \times \mathcal{X}_p \longrightarrow [0,\infty)$ is defined such that

$$Q_p(x_{p-1}, A) \coloneqq G_{p-1}(x_{p-1}) M_p(x_{p-1}, A), \qquad x_p \in \mathsf{X}_{p-1},\ A \in \mathcal{X}_p. \tag{1.27}$$

Compositions of these kernels give a second sequence of kernels $(Q_{p,n})_{p=0}^{n}$. For $p \in \{0,\dots,n\}$, define $Q_{p,n} : \mathsf{X}_p \times \mathcal{X}_n \longrightarrow [0,\infty)$ by

$$Q_{n,n} \coloneqq \mathrm{Id}, \qquad Q_{p,n} \coloneqq Q_{p+1} \cdots Q_n,\ p \in \{0,\dots,n-1\}. \tag{1.28}$$

Similar constructions are employed by Del Moral (2004, Section 7.2) and Chopin (2004, Equation 9), allowing these authors' recursive formulae to be expressed in closed form. Then it may be shown (Lee and Whiteley, 2018, Remark 2) that

$$v_{p,n}(\varphi) = \frac{\eta_p(Q_{p,n}(\varphi)^2)}{\eta_p(Q_{p,n}(\mathbb{1}))^2} - \eta_n(\varphi)^2. \tag{1.29}$$

We shall return to this result in Chapter 3.

1.4.1.1. Updated Feynman–Kac models

For the updated Feynman–Kac models $\hat{\gamma}_n$ and $\hat{\eta}_n$ the (relative) asymptotic variances of estimators $\hat{\gamma}_n^N(\varphi)$ and $\hat{\eta}_n^N(\varphi)$ may be defined similarly, and admit analogous decompositions. For completeness, and in preparation for the subsequent discussion on occasional resampling, we here restate the above results as they relate to updated Feynman–Kac models. These results form a summary of those presented in Section 5 of Lee and Whiteley (2018).

The asymptotic variance of $\hat{\gamma}_n^N(\varphi)$ as $N \longrightarrow \infty$ is defined as $\lim_{N\to\infty} N \operatorname{var}\left(\hat{\gamma}_n^N(\varphi)\right)$, and its *relative* asymptotic variance as $N \longrightarrow \infty$ is

$$\lim_{N\to\infty} N \operatorname{var}\left(\frac{\hat{\gamma}_n^N(\varphi)}{\hat{\gamma}_n(\mathbb{1})}\right). \tag{1.30}$$

The asymptotic variance of $\hat{\eta}_n^N(\varphi)$ as $N \longrightarrow \infty$ is similarly defined as $\lim_{N\to\infty} N \operatorname{var}\left(\hat{\eta}_n^N(\varphi)\right)$, and may be expressed in the form (1.30) as the relative asymptotic variance as $N \longrightarrow \infty$ of $\hat{\gamma}_n^N(\varphi - \hat{\eta}_n(\varphi))$.

The relative asymptotic variance (1.30) may be expressed as sum of $n+1$ terms:

$$\lim_{N\to\infty} N \operatorname{var}\left(\frac{\hat{\gamma}_n^N(\varphi)}{\hat{\gamma}_n(\mathbb{1})}\right) = \sum_{p=0}^{n} \hat{v}_{p,n}(\varphi). \tag{1.31}$$

By consideration of the identities (1.13), we may obtain a relationship between these terms $\hat{v}_{p,n}(\varphi)$, and the terms $v_{p,n}(\varphi)$ in the decomposition (1.26) corresponding to the 'non-updated' model γ_n. Specifically, we have for each $p \in \{0, \ldots, n\}$ that

$$\hat{v}_{p,n}(\varphi) = \frac{v_{p,n}(\varphi \cdot G_n)}{\eta_n(G_n)^2}. \tag{1.32}$$

An expression for $\hat{v}_{p,n}(\varphi)$ analogous to (1.29) may be obtained by substituting that expression for $v_{p,n}(\varphi)$ into (1.32), giving

$$\hat{v}_{p,n}(\varphi) = \frac{\eta_p(Q_{p,n}(\varphi \cdot G_n)^2)}{\eta_p(Q_{p,n}(G_n))^2} - \hat{\eta}_n(\varphi)^2. \tag{1.33}$$

1.4.1.2. Occasional resampling

Consider now an SMC algorithm employing occasional resampling as in Algorithm 1.3, and the resulting estimators $\gamma_n^N(\varphi)$, defined according to (1.18). Recall the definition of r_n as the number of instances of resampling between times 0 and n. For $\gamma_n^N(\varphi)$ defined according to (1.18), the relative asymptotic variance as $N \longrightarrow \infty$ admits a decomposition analogous to (1.26), as a sum of $r_n + 1$ terms, each of which may be expressed in a manner comparable to (1.29). The derivation of such a result follows directly from consideration of the appropriate excursion Feynman–Kac models as introduced in Section 1.3.1.1. Since the relevant expressions do not appear to have been explicitly stated elsewhere in the

literature, we present these in full below.

In the following exposition of results we consider a fixed sequence of resampling times $k_{1:r_n}$. However, these results also apply to estimators formed using adaptive resampling, due to the previously-discussed results of Del Moral et al. (2012b). In this case the sequence $k_{1:r_n}$, which determines the collection of excursion models, corresponds to the almost sure limit of the sequence of resampling times as $N \longrightarrow \infty$.

Recall that when occasional resampling is used, the resulting particle approximations correspond to approximations of updated Feynman–Kac models on such a sequence of excursion spaces. Appropriate analogues of the previous results for updated models therefore hold in this setting. Specifically, the relative asymptotic variance of $\gamma_n^N(\varphi)$, defined as in (1.18), may be decomposed in the manner of (1.31); extending the notation introduced in Section 1.3.1.1, we may write

$$\lim_{N\longrightarrow\infty} N \operatorname{var}\left(\frac{\hat{\gamma}_n^N(\varphi)}{\hat{\gamma}_n(\mathbb{1})}\right) = \sum_{j=0}^{r_n} \hat{v}'_{j,r_n}(\varphi). \tag{1.34}$$

We proceed to derive an expression of the form (1.33) for each of the terms in this decomposition. Using the potential functions $(G'_j)_{j=0}^{r_n}$ and Markov kernels $(M'_j)_{j=1}^{r_n}$, we may define a collection of integral kernels $(Q'_j)_{j=1}^{r_n}$ analogously to (1.27). That is, for $j \in \{1, \dots, r_n\}$, define $Q'_j : \mathsf{X}'_{j-1} \times \mathcal{X}'_j \longrightarrow [0, \infty)$ such that

$$Q'_j(x'_{j-1}, A) \coloneqq G'_{j-1}(x'_{j-1}) M'_j(x'_{j-1}, A), \qquad x'_j \in \mathsf{X}'_{j-1},\ A \in \mathcal{X}'_j. \tag{1.35}$$

Compositions of these kernels give a second sequence of kernels $(Q'_{j,r_n})_{j=0}^{r_n}$, as in (1.28): for $j \in \{0, \dots, r_n\}$, define $Q'_{j,r_n} : \mathsf{X}'_j \times \mathcal{X}'_{r_n} \longrightarrow [0, \infty)$ by

$$Q'_{r_n,r_n} \coloneqq \mathrm{Id}, \qquad Q'_{j,r_n} \coloneqq Q'_{j+1} \cdots Q'_{r_n},\ j \in \{0, \dots, r_n - 1\}. \tag{1.36}$$

As in Section 1.3.1.1, let $(\eta'_j)_{j=0}^{r_n}$ denote the normalised prediction Feynman–Kac models associated with $(G'_j)_{j=0}^{r_n}$ and $(M'_j)_{j=0}^{r_n}$, and $(\hat{\eta}'_j)_{j=0}^{r_n}$ the corresponding updated models. Furthermore, for any bounded $\mathcal{X}_n$-measurable function $\varphi : \mathsf{X}_n \longrightarrow \mathbb{R}$, let $\bar{\varphi} : \mathsf{X}'_{r_n} \longrightarrow \mathbb{R}$ denote the function given by

$$\bar{\varphi}(x'_{r_n}) = \bar{\varphi}(x_{k_{r_n}}, \dots, x_n) \coloneqq \varphi(x_n). \tag{1.37}$$

Then an analogue of (1.33) is given by

$$\hat{v}'_{j,r_n}(\varphi) \coloneqq \frac{\eta'_j(Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi})^2)}{\eta'_j(Q'_{j,r_n}(G'_{r_n}))^2} - \hat{\eta}'_{r_n}(\bar{\varphi})^2 \tag{1.38}$$

for $j \in \{0, \dots, r_n\}$. The subtrahend at the end of this expression may be simplified by observing that $\hat{\eta}'_{r_n}(\bar{\varphi}) = \eta_n(\varphi)$.

In the special case that $k_{r_n} = n$, corresponding to resampling occurring in the nth step,

the final potential function is $G'_{r_n} = \mathbb{1}$. The final updated measure $\hat{\eta}'_{r_n}$ is therefore equal to its 'non-updated' form η'_{r_n}; additionally, $\bar{\varphi} = \varphi$. We may therefore simplify (1.38), obtaining

$$v'_{j,r_n}(\varphi) := \frac{\eta'_j(Q'_{j,r_n}(\varphi)^2)}{\eta'_j(Q'_{j,r_n}(\mathbb{1})^2} - \eta'_{r_n}(\varphi)^2 \tag{1.39}$$

for $j \in \{0, \dots, r_n\}$, with the relative asymptotic variance of (1.18) being equal to $\sum_{j=0}^{r_n} v'_{j,r_n}(\varphi)$. This expression is directly comparable to (1.29). A further simplification to the final subtracted term in (1.39) is possible, since $\eta'_{r_n} = \eta_n$.

1.5. Variance estimation

In order to quantify the Monte Carlo error of such estimators as $\gamma_n^N(\varphi)$ and $\eta_n^N(\varphi)$, it is useful to estimate their variance. A simple approach would require running the SMC algorithm many times to obtain IID replicates of the estimator of interest, and computing their sample variance. Since this may be computationally costly, several alternative approaches to variance estimation have been proposed in the literature. These commonly focus on estimating the variance of $\gamma_n^N(\mathsf{X}_n)$, the estimator of the normalising constant of γ_n. For example, Bhadra and Ionides (2016) propose a method based on fitting an AR(1) 'meta-model' to the estimation errors of each $\gamma_p^N(\mathsf{X}_p)$, and Kostov and Whiteley (2017) utilise a 'pairs algorithm' to form an unbiased estimator of the second moment of $\gamma_n^N(\mathsf{X}_n)$.

In several settings it is also convenient to estimate the *asymptotic* variances of SMC estimators as $N \longrightarrow \infty$. A notable work in this field is that of Chan and Lai (2013), who propose an estimator of the asymptotic variance of $\hat{\eta}_n^N(\varphi)$, corresponding to the updated normalised Feynman–Kac model. This estimator is consistent in N and may be computed using the same run of the SMC algorithm used to generate $\hat{\eta}_n^N(\varphi)$ itself, by considering the sequence of ancestors of each particle as generated by the resampling process. This assumes the use of the multinomial resampling scheme described in Section 1.3.

Building on this work, Lee and Whiteley (2018) propose a collection of variance estimators that may similarly be computed using a single realisation of Algorithm 1.2 (i.e. using resampling in every iteration), by considering the 'genealogy' of the particles. These estimators are presented in the context of estimating the relative asymptotic variance (1.24) of $\gamma_n^N(\varphi)$, though the authors give several results validating their use in estimating more general asymptotic variances. We here introduce the authors' notation for these estimators, in order to facilitate reference to these throughout this thesis.

For $i \in \{1, \dots, N\}$, one may retrace the ancestry of the ith particle ζ_n^i to determine its ancestor among the initial set of particles sampled at time 0. Let E_n^i denote the index of this zeroth-generation ancestor; if these are recorded during the running of the algorithm,

then for any bounded $\mathcal{X}_n$-measurable function $\varphi : \mathsf{X}_n \longrightarrow \mathbb{R}$ one may compute

$$V_n^N(\varphi) := \frac{1}{N^2}\left[\left(\sum_{i=1}^{N}\varphi(\zeta_n^i)\right)^2 - \left(\frac{N}{N-1}\right)^{n+1}\sum_{i,j\,:\,E_n^i\neq E_n^j}\varphi(\zeta_n^i)\varphi(\zeta_n^j)\right], \tag{1.40}$$

given as Equation 4 in Lee and Whiteley (2018).

This quantity may be used to construct unbiased or consistent estimators of several (asymptotic) variances. By Theorem 1 in that work,

- $\gamma_n^N(\mathsf{X}_n)^2 V_n^N(\varphi)$ is an unbiased estimator of $\mathrm{var}(\gamma_n^N(\varphi))$ for any N;
- $N V_n^N(\varphi)$ converges in probability as $N \longrightarrow \infty$ to (1.24), the relative asymptotic variance of $\gamma_n^N(\varphi)$ as $N \longrightarrow \infty$;
- $N V_n^N(\varphi - \eta_n^N(\varphi))$ converges in probability as $N \longrightarrow \infty$ to (1.25), the asymptotic variance of $\eta_n^N(\varphi)$ as $N \longrightarrow \infty$.

Since $V_n^N(\varphi)$ may be computed as a by-product of the SMC algorithm, the (asymptotic) variances of estimators such as $\gamma_n^N(\varphi)$ and $\eta_n^N(\varphi)$ may be estimated and returned alongside the estimators themselves. The ease of computing such variance estimators presents many opportunities for their use in tuning the SMC algorithm, and we will consider some such ideas in this thesis.

In settings where n is large, this estimator may exhibit numerical instability. As n increases, all N particles will eventually share a common zeroth-generation ancestor with probability 1, since the number of unique zeroth-generation ancestors among the particle set decreases (non-strictly) with each resampling step. If this occurs, then we see from (1.40) that $V_n^N(\varphi)$ collapses to zero. To avoid this Olsson and Douc (2019) propose a modified estimator, utilising not the zeroth-generation ancestors, but rather the ancestors belonging to some more recent generation. The improved numerical stability comes at the cost of an asymptotic bias, for which the authors provide bounds.

Lee and Whiteley (2018, Section 4.2) additionally propose a collection of quantities that may be used to estimate the terms $v_{p,n}(\varphi)$ in the decomposition (1.26) of the relative asymptotic variance of $\gamma_n^N(\varphi)$. These estimators are denoted $v_{p,n}^N(\varphi)$ for $p \in \{0, \ldots, n\}$, and may similarly be computed as a by-product of a single realisation of the SMC algorithm. We omit the definition here; full pseudocode for computing these quantities is provided by the authors. Theorem 3 of that work presents results relating to unbiasedness and consistency, comparable to those for $V_n^N(\varphi)$; in particular, each $v_{p,n}^N(\varphi)$ converges in probability to $v_{p,n}(\varphi)$ as $N \longrightarrow \infty$.

Generalisations of these variance estimators to settings involving the updated Feynman–Kac models $\hat{\gamma}_n$ and $\hat{\eta}_n$ are also proposed (Lee and Whiteley, 2018, Section 5), based on the identities (1.13). These include $\hat{V}_n^N(\varphi)$, defined in terms of (1.40) as $V_n^N(\varphi)/\eta_n^N(G_n)^2$; the properties of this quantity are analogous to those of $V_n^N(\varphi)$, and are detailed in Theorem 4

of the authors' work. Similarly, a collection of estimators $\hat{v}_{p,n}^N(\varphi)$ is derived, which are convergent in probability to each term $\hat{v}_{p,n}(\varphi)$ in the asymptotic variance decomposition (1.31).

1.5.1. Occasional resampling

Variance estimators such as these may also be used in settings where occasional resampling is employed; again, this follows from considering the resulting particle approximations in terms of updated Feynman–Kac models defined on a sequence of excursion spaces. To this end, the (relative) asymptotic variance of such estimators as $\gamma_n^N(\varphi)$ and $\eta_n^N(\varphi)$ may be estimated as a by-product of Algorithm 1.3.

The terms (1.38) or (1.39) in the decomposition of the relative asymptotic variance of $\gamma_n^N(\varphi)$ may be estimated similarly, using an appropriate form of the estimators $\hat{v}_{p,n}^N(\varphi)$ or $v_{p,n}^N(\varphi)$. However, when adaptive resampling is used the sequence of resampling times resulting from a single realisation of the algorithm may not correspond to the almost sure limit of this sequence as $N \longrightarrow \infty$, which is generally unknown. Consequently there is not necessarily a one-to-one correspondence between the realisations of estimates of the form $v_{p,n}^N(\varphi)$ and the terms in the true decomposition of the relative asymptotic variance of $\gamma_n^N(\varphi)$. Caution must therefore be exercised when using the term-by-term estimators $v_{p,n}^N(\varphi)$ in adaptive settings, ensuring that N is sufficiently large that the sequence of resampling times is equal to its almost sure limit with high probability.

1.6. Summary

This chapter serves as a review of the structure and properties of sequential Monte Carlo algorithms, a class of simulation-based methods that may be used to approximate sequences of recursively-defined probability measures. The concepts and initial results introduced here will be fundamental to the ideas we later discuss, and so we shall frequently refer back to sections of this chapter. In particular, the notation we have defined in this chapter will be used throughout this thesis.

The ideas reviewed here will primarily be considered in the context of sequential Monte Carlo samplers, a subclass of SMC algorithms that we introduce in the next chapter.

2. Sequential Monte Carlo samplers

2.1. Methodology

The SMC algorithms presented so far are clearly suited to producing particle approximations of Feynman–Kac models $(\gamma_p)_{p=0}^n$ and $(\eta_p)_{p=0}^n$. However, it is not immediately clear how they may be exploited to produce empirical approximations of probability measures that do not readily conform to the Feynman–Kac framework. We consider in this chapter the problem of approximating a sequence of probability measures $(\pi_p)_{p=0}^n$, defined on a common measurable space $(\mathsf{X}, \mathcal{X})$. It shall be assumed that these measures admit densities with respect to some common dominating measure $\mathrm{d}x$. That is, for $A \in \mathcal{X}$,

$$\pi_p(A) = \frac{1}{Z_p} \int_A \bar{\pi}_p(x)\, \mathrm{d}x,$$

where $\bar{\pi}_p$ is an unnormalised density that can be computed at each $x \in \mathsf{X}$, and the normalising constant $Z_p := \int_{\mathsf{X}} \bar{\pi}_p(x)\, \mathrm{d}x$ may be unknown.

A methodology allowing such sequential sampling using SMC methods was proposed by Del Moral et al. (2006), in the form of *sequential Monte Carlo samplers*. The generic form that we shall describe was first proposed by Gilks and Berzuini (2001) and Chopin (2002) in different settings, building upon ideas in Crooks (1998) and Neal (2001). The basis of these algorithms lies in the view of SIS described in Section 1.2.2, as a form of importance sampling for approximating the joint distribution of the entire 'path' $(\zeta_0^i, \ldots, \zeta_n^i)$ of each particle. These joint distributions are defined in terms of the potential functions and Markov kernels, as in (1.12); the idea is therefore to specify these appropriately in order that the distributions of interest π_p are admitted as marginal distributions.

Specifically, one constructs a sequence of recursively-defined target distributions on joint spaces of increasing dimension. To achieve this, one defines a sequence of 'backward-in-time' Markov kernels $(L_p)_{p=0}^{n-1}$. For each $p \in \{0, \ldots, n\}$ we define a probability measure $\tilde{\eta}_p$ on the joint space X^{p+1} by taking the tensor product of π_p with a subsequence of these backward kernels; for some set $\tilde{A}$ belonging to the product σ-algebra $\mathcal{X}^{\otimes p+1}$, we have

$$\tilde{\eta}_p(\tilde{A}) := \int_{\tilde{A}} \pi_p(\mathrm{d}x_p) \prod_{q=0}^{p-1} L_q(x_{q+1}, \mathrm{d}x_q). \tag{2.1}$$

This is seen to admit π_p as a marginal distribution. The choice of backward kernels is

formally arbitrary, though greatly affects the efficiency of the algorithm. In practice one chooses backward kernels 'close to' an optimal choice given in Proposition 1 of Del Moral et al. (2006), which minimises the variance of the importance weights.

In the case that $M_0 = \pi_0$, and M_p is a Markov kernel invariant with respect to π_p for each $p \in \{1, \dots, n\}$, a popular choice is to take the backward kernels $(L_p)_{p=0}^{n-1}$ to be the time reversals of the Markov kernels $(M_p)_{p=1}^{n}$, as shall later be formally introduced in Definition 3.2. The joint distribution (2.1) then corresponds to the form (1.12), introduced in the motivation of sequential importance sampling, with potential functions given by

$$G_p(x) = \frac{\bar{\pi}_{p+1}(x)}{\bar{\pi}_p(x)} = \frac{Z_{p+1}}{Z_p}\frac{\mathrm{d}\pi_{p+1}}{\mathrm{d}\pi_p}(x), \qquad x \in \mathsf{X}, \quad p \in \{0, \dots, n-1\}. \tag{2.2}$$

Such a definition is possible as long as the sequence of supports of the distributions $(\pi_p)_{p=0}^{n}$ is non-increasing, so that $\pi_p \ll \pi_{p-1}$ for all $p \in \{1, \dots, n\}$. This allows the construction of an SIS algorithm of the structure presented in Algorithm 1.1, with incremental weights computed using ratios of successive unnormalised density functions.

Assuming that $\pi_{p-1} \approx \pi_p$, the resulting incremental weights should have reasonably low variance. A number of other ways to specify the required backward kernels are proposed Del Moral et al. (2006), which may yield incremental weights of a lower variance in specialised settings. However, these typically have a form that depends on the Markov kernels M_p; such expressions may be intractable, for example if each M_p is a Markov chain Monte Carlo (MCMC) kernel, which does not admit a density with respect to the Lebesgue measure. Furthermore, such incremental weights cannot be expressed as potential functions evaluated on the current particle set, necessitating a form of SIS that is subtly different to that presented in Chapter 1 in the context of Feynman–Kac models.

For this reason, throughout this thesis we shall solely consider SMC samplers employing potential functions of the form (2.2). The resulting Feynman–Kac models are such that, for $p \in \{0, \dots, n\}$,

$$\eta_p = \pi_p, \qquad \gamma_p(\mathsf{X}) = \frac{Z_p}{Z_0}.$$

Particle approximations of these quantities may be formed using any of the SMC algorithms previously described, for these choices of $(G_p)_{p=0}^{n-1}$ and $(M_p)_{p=0}^{n}$. To make this explicit, we present here as Algorithm 2.1 an SMC sampler using occasional resampling (i.e. in the form of Algorithm 1.3).

We see that approximations of each π_p may be formed as the empirical measures η_p^N, computed as in (1.17). The ratio of normalising constants Z_p/Z_0 may be estimated by $\gamma_p^N(\mathsf{X})$, according to (1.18). This is particularly advantageous since in several applications, normalising constants Z_p (or ratios thereof) are the primary objects of inference. We shall describe some such settings in the review that follows.

Algorithm 2.1 Sequential Monte Carlo sampler

1. At time $p = 0$:
   - For $i \in \{1, \ldots, N\}$ set $W_0^i \longleftarrow 1$ and independently sample $\zeta_0^i \sim \pi_0(\cdot)$.
2. At time $p = 1, \ldots, n$,
   - For $i \in \{1, \ldots, N\}$ set $\tilde{W}_p^i \longleftarrow W_{p-1}^i \dfrac{\bar{\pi}_p(\zeta_{p-1}^i)}{\bar{\pi}_{p-1}(\zeta_{p-1}^i)}$.
   - If resampling in the pth iteration:
     - For $i \in \{1, \ldots, N\}$ independently sample $A_{p-1}^i \sim \text{Categorical}\big(\tilde{W}_p^1, \ldots, \tilde{W}_p^N\big)$ and set $W_p^i \longleftarrow 1$.
   - Else:
     - For $i \in \{1, \ldots, N\}$ set $A_{p-1}^i \longleftarrow i$ and set $W_p^i \longleftarrow \tilde{W}_p^i$.
   - For $i \in \{1, \ldots, N\}$ independently sample $\zeta_p^i \sim M_p(\zeta_{p-1}^{A_{p-1}^i}, \cdot)$, where M_p is a Markov kernel admitting π_p as an invariant distribution.

2.2. Tempering

While the methodology of SMC samplers is based on sampling from each in a sequence of distributions, in many settings there is only one distribution $\pi_\star$ of direct interest. If $\pi_\star$ has multiple well-separated modes, or has a non-trivial covariance structure, then it may be difficult to produce adequate approximations using a single $\pi_\star$-invariant Markov chain. For example, a random walk Metropolis sampler (Metropolis et al., 1953) may struggle to explore the space X efficiently in such scenarios and may therefore exhibit high autocorrelation. In such cases it is common to construct a sequence of distributions $(\pi_p)_{p=0}^n$ in which $\pi_n := \pi_\star$ is the final such distribution. One chooses an initial distribution π_0 that is relatively tractable compared to $\pi_\star$, and from which one may readily produce samples. The intermediate distributions form a gradual transition between π_0 and $\pi_\star$.

The benefit of using SMC samplers to produce samples from each of these distributions in turn is that the problem of constructing a suitably well-mixing Markov kernel is reduced. By first sampling a collection of particles from the benign distribution π_0, and then applying each Markov kernel M_p (resampling as necessary), these particles are 'moved through' the sequence of distributions π_p. Their nth-generation incarnations may be used to form a particle approximation η_n^N of $\pi_n = \pi_\star$.

Constructing well-mixing Markov kernels M_p (e.g. MCMC kernels) targeting those distributions later in the sequence may be difficult, though may be easier for earlier distributions if π_0 is sufficiently simple. The application of these well-mixing kernels helps to distribute the particles throughout the space in all areas of high density. By constructing the sequence of distributions $(\pi_p)_{p=0}^n$ so that consecutive distributions π_{p-1} and π_p are suf-

ficiently 'similar', it will be straightforward to move the particles from the areas of high density of π_{p-1} to those of π_p, by application of M_p.

In particular, the features of $\pi_\star$ that make it difficult to form a well-mixing $\pi_\star$-invariant Markov kernel may be gradually introduced to each of the intermediate distributions. For example, in the case that $\pi_\star$ is highly multimodal one can choose π_0 be a broad unimodal distribution, with its mass gradually separating out into the modes of $\pi_\star$ as one progresses through the sequence. This behaviour is illustrated in Figure 2.1, for a sequence of univariate distributions interpolating between $\pi_0 = \mathcal{N}(0, 10^2)$ and $\pi_\star = 0.3\mathcal{N}(-10, 0.4^2) + 0.7\mathcal{N}(10, 0.8^2)$. Although $\pi_\star$ has modes that are separated by a region of negligible probability mass, earlier distributions in the sequence place non-negligible mass on a large connected set, making it straightforward to construct well-mixing MCMC kernels targeting these distributions.

Analogously to each π_p, assume that for $A \in \mathcal{X}$

$$\pi_\star(A) = \frac{1}{Z_\star} \int_A \bar{\pi}_\star(x)\,\mathrm{d}x,$$

where $\bar{\pi}_\star$ is an unnormalised density with respect to a dominating measure $\mathrm{d}x$ that can be computed at each $x \in \mathsf{X}$, and the normalising constant $Z_\star := \int_{\mathsf{X}} \bar{\pi}_\star(x)\,\mathrm{d}x$ may be unknown. For a given sequence of distributions $(\pi_p)_{p=0}^n$ we therefore have $\pi_\star = \pi_n$, $\bar{\pi}_\star = \bar{\pi}_n$, and $Z_\star = Z_n$.

A commonly-used approach in the literature for producing such a sequence of distributions is known as *tempering*. This is so named by analogy with the technique in metallurgy, in which alloys are toughened by being heated to high temperatures and slowly cooled. This allows the atoms within to redistribute themselves, in order that they may ultimately settle in a more stable arrangement. In a comparable way, sampling from π_0 distributes particles across the state space, with these gradually settling in the areas of high mass of $\pi_\star$.

Within the tempering framework, one considers functions on X of the form

$$\bar{\pi}_0(\cdot)^{1-\beta}\bar{\pi}_\star(\cdot)^{\beta}, \tag{2.3}$$

where $\beta \in [0, 1]$. This defines a collection of unnormalised densities that smoothly interpolate between $\bar{\pi}_0$ and $\bar{\pi}_\star$ as β increases from 0 to 1. An appropriate sequence of distributions may therefore be obtained by evaluating (2.3) at a discrete set of values of β. Formally, one defines a *temperature schedule* $(\beta_p)_{p=0}^n$ such that

$$0 =: \beta_0 < \beta_1 < \cdots < \beta_n := 1.$$

Each value β_p is known as an *inverse temperature*, by analogy with metallurgical tempering. The resulting sequence of tempered distributions $(\pi_p)_{p=0}^n$ is defined in terms of the

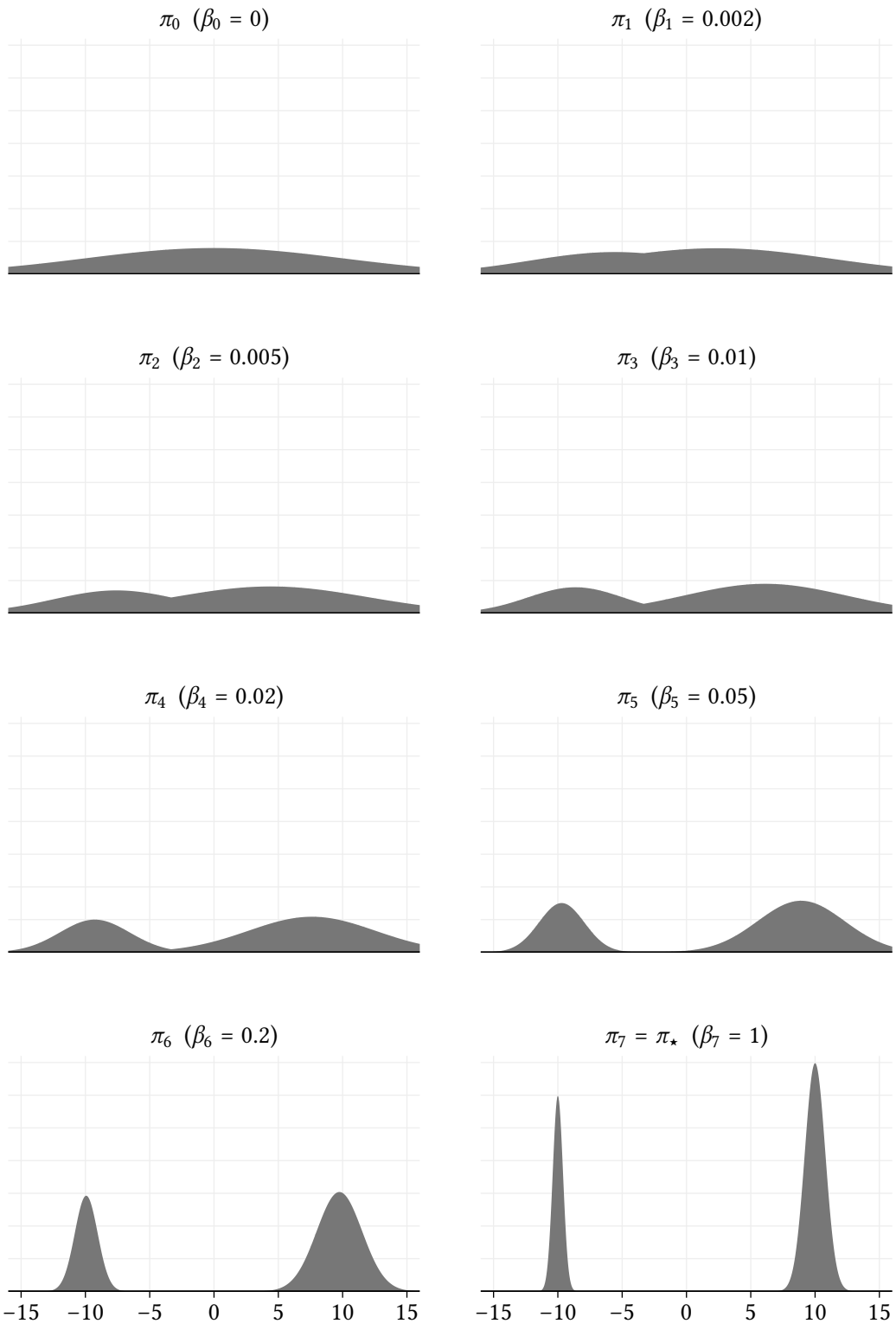


Figure 2.1.: Density functions of each in a tempered sequence of distributions, with $\pi_0 = \mathcal{N}(0, 10^2)$, $\pi_\star = 0.3\mathcal{N}(-10, 0.4^2) + 0.7\mathcal{N}(10, 0.8^2)$, and intermediate distributions specified by the stated inverse temperatures $\beta_{0:7}$.

corresponding unnormalised density functions, given by

$$\bar{\pi}_p(x) = \bar{\pi}_0(x)^{1-\beta_p} \bar{\pi}_\star(x)^{\beta_p}, \qquad x \in \mathsf{X}. \tag{2.4}$$

The density functions depicted in Figure 2.1 are an example of such a tempered sequence, specified by the temperature schedule $\beta_{0:7} = (0,\ 0.002,\ 0.005,\ 0.01,\ 0.02,\ 0.05,\ 0.2,\ 1)$.

Using these distributions in the previously-described framework of the SMC sampler requires $M_0 = \pi_0$, and M_p to be a π_p-invariant Markov kernel for $p \in \{1, \ldots, n\}$, typically an MCMC kernel. The potential functions (2.2) may conveniently be expressed as

$$G_p(x) = \left(\frac{\bar{\pi}_\star(x)}{\bar{\pi}_0(x)} \right)^{\beta_{p+1} - \beta_p}, \qquad x \in \mathsf{X}, \quad p \in \{0, \ldots, n-1\}. \tag{2.5}$$

As previously mentioned, the resulting Feynman–Kac measures are such that $\eta_n = \pi_n = \pi_\star$. Additionally, $\gamma_n(\mathsf{X})$ is equal to the ratio of normalising constants $Z_n/Z_0 = Z_\star/Z_0$.

The use of SMC samplers with tempered sequences of distributions bears many similarities to annealed importance sampling (Neal, 2001), an earlier algorithm which may be viewed as a special case of this framework. We discuss this connection in Section 3.2.1.

2.2.1. Bayesian posteriors

A common setting to which SMC-based tempering is applied is that of sampling from a Bayesian posterior distribution on some parameter space X. For example, in Bayesian mixture modelling the use of exchangeable priors results in non-identifiable mixture components, yielding highly multimodal posteriors from which sampling via simple MCMC methods is difficult. Jasra et al. (2005) provide a full description of this 'label switching' problem, and Del Moral et al. (2006, Section 4) present an SMC sampler for use in this setting. In another application, Fan et al. (2008) describe how tempering may be used in an SMC approach to the Bayesian analysis of generalised linear mixed models.

Writing $p(x)$ for the prior density, and $p(y \mid x)$ for the likelihood of the observations y given x, a common choice is to take $\bar{\pi}_0(x) = p(x)$ and $\bar{\pi}_\star(x) = p(x)p(y \mid x)$. This gives the posterior distribution of x given y as the target $\pi_\star$, with $\gamma_n(\mathsf{X}) = Z_\star/Z_0 = Z_\star$ being the marginal likelihood of the observations y. The resulting procedure is known as *likelihood tempering*; the intermediate distributions have unnormalised densities that may be expressed as

$$\bar{\pi}_p(x) = p(x)p(y \mid x)^{\beta_p}.$$

Such expressions have been termed *power posteriors* by Friel and Pettitt (2008) and have been widely studied, particularly with regard to path sampling (as shall be further discussed in Section 3.2.3).

A similar approach may be used for the purposes of assessing model fit. Suppose $p(x \mid$

M) is a prior density on X given some model M, and $p(y \mid x, M)$ is the corresponding likelihood function. Taking $\bar{\pi}_0(x) = p(x \mid M)$ and $\bar{\pi}_\star(x) = p(x \mid M)p(y \mid x, M)$ results in the normalising constant ratio $\gamma_n(\mathsf{X}) = Z_\star / Z_0 = Z_\star$ being the marginal likelihood of y under the model M; that is, the evidence for model M. This forms the basis of the 'SMC2' algorithm of Zhou et al. (2016). The same authors provide an additional algorithm, albeit not employing tempering, for the purposes of directly estimating evidence ratios between models (Bayes factors), exploiting the ability of SMC samplers to unbiasedly estimate ratios of normalising constants.

In the case of sampling from a Bayesian posterior with likelihood $p(y \mid x) = p(y_{1:m} \mid x)$, an alternative to likelihood tempering has been suggested by Chopin (2002): defining a sequence

$$0 =: m_0 < m_1 < \cdots < m_n := m,$$

one takes $\bar{\pi}_p(x) = p(x)p(y_{1:m_p} \mid x)$. That is, one uses the prior as the initial distribution, with successive distributions incorporating the likelihood contribution of a batch of observations. Although these densities are not of the form (2.4), this approach is known as *data tempering* by analogy. If the batches of observations are conditionally independent given the parameter x, then incremental weights of the form (2.2) correspond to the likelihood contributions of each such batch.

2.2.2. Tempering outside the SMC framework

Tempering is a general technique for which application is not restricted to SMC samplers. Several methods have been proposed for using tempered distributions in MCMC settings, in order to benefit from the better mixing of Markov kernels targeting distributions corresponding to lower inverse temperatures. We here provide a brief summary of some such methods, though note that these are outside the scope of this thesis.

Simulated tempering (Marinari and Parisi, 1992; Geyer and Thompson, 1995) is an MCMC technique proposed as a solution to the poor mixing observed when using a single $\pi_\star$-invariant MCMC kernel. By defining an artificial joint distribution over the state space X and the space of temperatures, an MCMC kernel is constructed with the true target as its invariant distribution (after marginalising over the temperature space). The resulting sampler benefits from more efficient mixing at lower inverse temperatures. Motivated by this approach but avoiding the need to extend the state space, Neal (1996) proposed the use of 'tempered transitions', in which a new state is proposed by applying to the current state a sequence of Markov kernels (and their time reversals) leaving invariant each distribution in a tempered sequence. Considering the corresponding inverse temperatures, these Markov kernels may be be seen to apply a repeated 'heating and cooling' effect.

Parallel tempering (Geyer, 1991) has similar aims, running several Markov chains in parallel, targeting each distribution $\pi_0, \ldots, \pi_n$ in a tempered sequence. As well as new

values for each chain being proposed and accepted or rejected in the usual manner, the algorithm also occasionally proposes the swapping of values between two chains targeting consecutive distributions. This may be done using an appropriately-defined Metropolis–Hastings acceptance probability, without altering the invariant distribution of either chain. A full review of this technique is given by Earl and Deem (2005).

We finally note that tempered sequences of distributions play a key role in path sampling and thermodynamic integration, which we discuss in Section 3.2.3.

2.3. Other applications

Defining a sequence of intermediate distributions by means of a temperature schedule is a common approach in the literature, and is often favoured for its generality. In principle however, any appropriate sequence of distributions may be chosen to interpolate between π_0 and $\pi_\star$. We here review some proposed applications of SMC samplers in which alternative parametric approaches are employed for the purpose of specifying this sequence.

2.3.1. Bayesian inference

Alongside the widespread use of tempering as described in Section 2.2.1, a number of specially-tailored SMC samplers have been proposed for problems in Bayesian inference. For example, an SMC sampler for a Bayesian binary probit regression model is considered by Del Moral et al. (2007, Section 4.2), in which the variance of the error term decreases between successive distributions until reaching its true value. Del Moral et al. (2006, Section 5) apply the methodology to the problem of estimating the intensity function of an inhomogeneous Poisson process sequentially in time, since the generic SMC framework can easily accommodate transdimensional parameter spaces.

In settings where the likelihood is intractable or expensive to compute, the framework of *approximate Bayesian computation* (ABC) provides a means of approximating a posterior distribution via the simulation of pseudo-data (see Marin et al., 2012, for a review). In its simplest form this takes the role of a rejection sampler, with an empirical measure formed from those parameter values for which the resulting pseudo-observations are sufficiently 'similar to' the true observations. A first SMC sampler for use in ABC contexts was proposed by Sisson et al. (2007), in which the tolerance level used in this rejection step decreases in each iteration, so that successive distributions form closer approximations of the true posterior. Later proposals include that of Del Moral et al. (2012a), in which this decreasing sequence of tolerance levels may be chosen adaptively.

2.3.2. Rare events

For a probability distribution π, a rare event $E \in \mathcal{X}$ is such that $\pi(E) \ll 1$; for example, the tail $E = [c, \infty)$ of a distribution on the real line. Estimation of $\pi(E)$ can be difficult using simple Monte Carlo techniques, even in settings where it is straightforward to sample from π. For example, a simple estimator using (1.1) may require very large numbers of samples to be drawn for the resulting estimate to be non-zero.

Del Moral et al. (2006) describe the basis of an approach using SMC samplers, requiring the specification of a decreasing sequence in $\mathcal{X}$ given by

$$\mathsf{X} =: E_0 \supseteq E_1 \supseteq \cdots \supseteq E_n := E.$$

Given $E_{0:n}$, one may construct a sequence of distributions π_p by considering the restriction of π to each of these sets in turn (normalised appropriately, so that each is a probability measure). Suppose π admits a (possibly unnormalised) density $\bar{\pi}$ with respect to some dominating measure; then an unnormalised density for each distribution in this sequence is given by $\bar{\pi}_p = \bar{\pi} \cdot \mathbb{1}_{E_p}$. Therefore $\pi(E)$ is equal to the ratio of normalising constants Z_n/Z_0, for which the SMC sampler framework provides an unbiased estimator.

We consider this setting more fully in Section 4.2, where we derive conditions under which the sequence of sets $E_{0:n}$ may be considered optimal; the same setting is investigated by Cérou et al. (2012) in the parametric setting in which $E_{0:n}$ are chosen as superlevel sets of a given function. SMC samplers for use in this context have also been proposed by Johansen et al. (2006), who describe two such procedures for estimating rare event probabilities in relation to the trajectories of Markov chains.

These SMC approaches to rare event estimation bear close similarities to multilevel splitting algorithms, including the RESTART algorithm (Villén-Altamirano and Villén-Altamirano, 1991). Such procedures also exhibit a clear genealogical structure, requiring the simulation of many Markov chains that are then cloned or discarded depending on their trajectories. A recent review of adaptive multilevel splitting, including a discussion of its links to sequential Monte Carlo, is provided by Cérou et al. (2019).

2.3.3. Interpolation to independence

A general method for constructing sequences of intermediate distributions in multidimensional settings was proposed by Paulin et al. (2019). Specifically, suppose that the distribution of interest $\pi_\star$ is defined on $\mathsf{X} = \mathsf{E}^n$, where E is some space of finite volume. Suppose also that we may construct a sequence of distributions $(\varpi_p)_{p=1}^n$ that are respectively defined on product spaces $(\mathsf{E}^p)_{p=1}^n$, with $\varpi_n = \pi_\star$.

To construct a sequence of interpolating distributions $(\pi_p)_{p=0}^n$ defined on X, one takes π_0 to be the uniform distribution on X, and $\pi_n = \varpi_n = \pi_\star$. For $p \in \{1, \ldots, n-1\}$, one defines π_p such that for any $X \sim \pi_p$, the first p components of X are distributed according to ϖ_p,

and the remaining $n - p$ components are each distributed uniformly on E, independently of all other components. This construction, previously introduced in the context of Stein's method (Chen and Röllin, 2010, Section 3.4), is known as *interpolation to independence*.

In settings where the target distribution $\pi_\star$ is multimodal, Paulin et al. (2019) describe how the resulting interpolating sequence allows Markov kernels M_p with good mixing properties to be constructed. The authors describe this for an example based on a Potts model, proving an upper bound on the asymptotic variances of the resulting estimators $\eta_n^N(\varphi)$.

2.3.4. Simulated annealing

Simulated annealing is a general-purpose optimisation algorithm, first proposed by Kirkpatrick et al. (1983) for the purpose of finding the argument that maximises a continuous bounded function $f : \mathsf{X} \longrightarrow \mathbb{R}$. The algorithm, using concepts from statistical mechanics, requires an increasing sequence of non-negative inverse temperatures $(\beta_p)_{p\geq 0}$ such that $\beta_p \longrightarrow \infty$ as $p \longrightarrow \infty$.

For each $p \geq 0$, one defines a Markov kernel M_p for which the invariant distribution admits a density at $x \in \mathsf{X}$ proportional to $\exp(\beta_p f(x))$. One may thereby construct an inhomogeneous Markov chain $(X_p)_{p\geq 0}$ such that $X_p \mid X_{p-1} \sim M_p(X_{p-1}, \cdot)$. Under certain conditions, X_p converges in probability to the argument that maximises f as $p \longrightarrow \infty$. Locatelli (2000) provides a review of several such convergence results.

As suggested by Del Moral et al. (2006), global optimisation of a density function π on X may be carried out in an SMC framework by taking each π_p to have a density at $x \in \mathsf{X}$ proportional to $\pi(x)^{\beta_p}$. Zhou and Chen (2013) investigate such a setup for the optimisation of compactly-supported functions f, obtaining various convergence results in a setting in which π_p admits a density at x proportional to $\exp(\beta_p f(x))$, with $\beta_0 = 0$.

2.4. Summary

The SMC sampler framework reviewed here may in theory be used to generate approximations of arbitrary sequences of probability measures defined on a common space. However, we shall primarily consider the setting introduced in Section 2.2, in which only a single distribution $\pi_\star$ is of direct interest. The problem of tuning SMC samplers, in particular that of choosing the sequence of distributions $(\pi_p)_{p=0}^n$, shall be the focus of the next part of this thesis.

Part II.

Schedule selection for sequential Monte Carlo samplers

3. The schedule selection problem

3.1. Overview

Consider the previously-described SMC sampler framework as used to form a particle approximation of some target distribution $\pi_\star$. A sequence of distributions $(\pi_p)_{p=0}^n$ is chosen to interpolate between some initial distribution π_0 and the final distribution $\pi_n := \pi_\star$; by analogy with temperature schedules as introduced in Section 2.2, we shall refer to such a sequence as a *distribution schedule*. An open question, which shall form a focus of this part of the thesis, is: how is such a distribution schedule best chosen?

In the discussion that follows, we shall assume some fixed predetermined choice of the initial distribution π_0. Although in theory this may be chosen freely, for several inference problems a natural choice of π_0 presents itself: for example, the prior distribution in Bayesian problems (as discussed in Section 2.2.1), or the distribution of interest when estimating rare event probabilities (see Section 2.3.2). More generally, in order to construct an SMC sampler using incremental weights of the form (2.2), we require that the sequence of supports of the distributions $(\pi_p)_{p=0}^n$ is non-increasing. In order that this may hold, and in order to assist the efficient movement of particles to the areas of high mass of $\pi_\star$ via application of the Markov kernels, one typically chooses π_0 to assign non-negligible mass to a large connected subset of the space. To this end, the practitioner's choice of π_0 may be seen to represent their prior beliefs about where the mass of $\pi_\star$ lies, and is therefore problem-specific.

Given π_0 and $\pi_\star$, the intermediate distributions may be chosen in any convenient way; the use of tempering is one such approach, with this sequence specified via a temperature schedule. However, the choice of sequence can have a significant impact on the efficiency of the algorithm, as we shall proceed to explore. In essence, the problem of choosing a distribution schedule is that of answering the following two questions:

- How many intermediate distributions should there be (i.e. what should n be chosen as)?
- What should these intermediate distributions $\pi_1, \dots, \pi_{n-1}$ be chosen as?

Given that estimators $\eta_n^N(\varphi)$ and $\gamma_n^N(\varphi)$ resulting from SMC samplers are consistent in N by Proposition 1.3 (and in the latter case unbiased, by Proposition 1.2), a particular consideration is how the choice of this sequence affects the variance of such estimators. Within

the SMC sampler presented in Algorithm 2.1, the incremental weights (2.2) correspond to ratios of unnormalised density functions belonging to consecutive distributions in the schedule. It follows that consecutive distributions should be sufficiently 'similar' in order to control the variance of such incremental weights. This naturally raises the question of how to define 'similarity' in this context.

Furthermore, by increasing n so that there are more distributions separating π_0 and $\pi_\star$, it becomes possible to make consecutive distributions more similar. This may in turn result in a lower variance of the incremental weights and resulting estimators. However, by inspection of Algorithm 2.1 we see that the time complexity of the SMC sampler is $\mathcal{O}(nN)$, since each additional distribution adds an extra iteration. For a fixed computational budget, an increase in n must therefore be carefully balanced with the number of particles N.

Another issue relates to the Markov kernels: recall that for each $p \in \{1, \ldots, n\}$, M_p is invariant with respect to π_p, typically an MCMC kernel. As described in Chapter 2 in most settings of practical interest it is more difficult to construct well-mixing Markov kernels targeting $\pi_\star$ than π_0. Consequently, for any given schedule $(\pi_p)_{p=0}^n$, Markov kernels M_p corresponding to distributions later in the sequence will typically exhibit poorer mixing. The advantage of an SMC sampler is that one benefits from the better mixing of kernels targeting distributions more similar to π_0; therefore, it may be useful to have a large number of such distributions that are 'closer to' π_0 than $\pi_\star$. For example, if the distributions $(\pi_p)_{p=0}^n$ are determined by a temperature schedule, it may be beneficial for many of the inverse temperatures to be close to 0, rather than uniformly spacing these values between 0 and 1. Figure 2.1 exemplifies this; by choosing many small inverse temperatures, the multimodality of $\pi_\star$ is introduced gradually.

In practical settings the distribution schedule is typically specified parametrically, for example via a temperature schedule or, for the rare event estimation procedure described in Section 2.3.2, via a decreasing sequence of sets. While we shall frequently consider such specifications over the course of the next chapters, the problem defined by the two previously-stated questions is fundamentally nonparametric in nature.

A third related question is:

- In which iterations of the SMC sampler should resampling take place?

For the SMC sampler presented in Algorithm 2.1, in which the iterations are indexed $\{1, \ldots, n\}$, we shall refer to the subset of iterations in which resampling takes place as the *resampling schedule*.

As discussed in Section 1.3.1, the choice of this resampling schedule determines the form of estimators $\eta_n^N(\varphi)$ and $\gamma_n^N(\varphi)$. It follows that if one wishes to minimise the variance of any such estimator, for any choice of distribution schedule $(\pi_p)_{p=0}^n$ there is an optimal choice of resampling schedule, and therefore these two schedules should be chosen concurrently. Although adaptive resampling is commonly used in practice, for example using the effective sample size as described in Section 1.3.2, there is no guarantee that the resulting

resampling schedule is optimal, nor it is clear how the ESS threshold τ should be chosen optimally.

Given the interplay between these two problems, we shall refer to the problem of jointly selecting a distribution schedule and resampling schedule as the *schedule selection problem* for SMC samplers. Within this chapter, we shall formally formulate this as a transdimensional optimisation problem, by introducing a quantity that may be used to compare choices of distribution and resampling schedules. We shall then proceed to derive a decomposition of this quantity, in order to facilitate its theoretical and empirical analysis over the following chapters.

3.2. Approaches to temperature schedule selection

We begin with a review of some proposed approaches to parametric schedule selection in practice. In the setting of tempering, the problem of selecting a distribution schedule reduces to that of selecting a temperature schedule of increasing values between 0 and 1. Given the simplicity and wide applicability of this approach, many existing proposals for selecting $(\pi_p)_{p=0}^n$ consider this specific problem. We here provide a summary of relevant results from the literature.

3.2.1. Annealed importance sampling

Annealed importance sampling (Neal, 2001) is a procedure for sampling from a tempered sequence of distributions that predates the more general framework of SMC samplers. It corresponds exactly, however, to an SMC sampler applied to a tempered sequence of distributions $(\pi_p)_{p=0}^n$ using no resampling (i.e. using SIS as in Algorithm 1.1), with incremental weights given by (2.5). Since no resampling is employed, minimising the variance of the resulting estimators amounts to minimising the variance of the importance weights.

This work follows a previous article by the same author on 'tempered transitions' (Neal, 1996), which employ tempered distributions in an MCMC setting (see Section 2.2.2 for a brief description). Based on arguments from this previous work, the author describes an approach to selecting a temperature schedule in a setting in which $\bar{\pi}_0(x)^{1-\beta}\bar{\pi}_\star(x)^\beta$ is approximately Gaussian over a certain range of $\beta \in [0, 1]$, and $\bar{\pi}_0$ is approximately constant in those regions in which these unnormalised densities are high. It is argued that, in order to minimise the variance of the importance weights, the inverse temperatures in this range should be geometrically spaced; that is, β_{p+1}/β_p should be constant. Such an argument assumes the use of perfectly-mixing Markov kernels, as we shall consider in Chapter 4. The accompanying empirical studies use experimentally-chosen temperature schedules based on this heuristic.

3.2.2. Adaptive temperature selection

A temperature schedule may be generated in an automated manner by using an adaptive procedure: rather than specifying the schedule in advance, each successive inverse temperature is chosen online using the sampled particles. Specifically, one sets $\beta_0 = 0$, and then for each $p \geq 1$ the value of β_p is determined at pth step of the SMC sampler. Beskos et al. (2016) present a number of convergence results for such adaptive tempering procedures.

Jasra et al. (2011) propose such a procedure in which the choice of each inverse temperature is made using the ESS, a quantity usually employed for the purpose of assessing the variance of the importance weights as described in Section 1.3.2. Recalling its empirical definition (1.23) at the pth time step, it may be noted that this depends on the weights $(W_{p-1}^i)_{i=1}^N$ from the previous time step and the incremental weights $(G_{p-1}(\zeta_{p-1}^i))_{i=1}^N$. From the form of (2.5) it may be seen that these incremental weights depend only on β_{p-1} and β_p, of which the former will have already been determined.

The suggestion is to fix the decay of the ESS in each iteration, by selecting the next value β_p in order that $\mathrm{ESS}_p^N = \mathrm{ESS}_p^\star$ for an appropriately-chosen $\mathrm{ESS}_p^\star \in [1, N]$. The solution to this equation must be found numerically, for example using the bisection method; if the solution is greater than 1, then one sets $\beta_p = 1$ (and so the algorithm terminates following this time step). The aim of this procedure, when used with adaptive resampling based on the ESS, is to control the frequency at which resampling occurs (that is, how often the ESS falls below the resampling threshold τ). A similar technique is employed by Schäfer and Chopin (2013).

An alternative approach by Zhou et al. (2016) aims to allow direct control over the dissimilarity between consecutive distributions, where this is measured by the following quantity.

Definition 3.1. For two probability measures π and μ defined on $(\mathsf{X}, \mathcal{X})$ such that $\pi \ll \mu$, the *chi-squared distance* of π from μ (also known as the *chi-squared divergence* or *Pearson divergence*) is defined as

$$D_{\chi^2}(\pi \,\|\, \mu) := \int_{\mathsf{X}} \left(\frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x) - 1 \right)^2 \mu(\mathrm{d}x).$$

This measure of dissimilarity does not define a metric, though may be seen as a measure of the difference between the two distributions, being an instance of an f-divergence as introduced by Csiszár (1963). Its definition may be formulated in a number of equivalent ways; for example, we may also write

$$D_{\chi^2}(\pi \,\|\, \mu) = \int_{\mathsf{X}} \left(\frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x) \right)^2 \mu(\mathrm{d}x) - 1 \tag{3.1}$$

and

$$D_{\chi^2}(\pi \,\|\, \mu) = \int_{\mathsf{X}} \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x) - 1. \tag{3.2}$$

Another useful formulation is

$$D_{\chi^2}(\pi \,\|\, \mu) = \mathrm{var}_{\mu}\left(\frac{\mathrm{d}\pi}{\mathrm{d}\mu}\right); \tag{3.3}$$

that is to say, the variance of the Radon–Nikodym derivative $\mathrm{d}\pi/\mathrm{d}\mu$ when its argument is distributed according to μ. As noted by Chen (2005), this is equal to the variance of the importance weights when μ is used as an importance distribution, in order to compute estimates of expectations with respect to π. As such, the chi-squared distance provides a useful measure of the efficiency of an importance sampling algorithm.

The variance (3.3) has previously been encountered in the exact form (1.21) of the effective sample size of an importance sample, which may therefore be viewed as a function of this chi-squared distance. Within the context of SIS, Zhou et al. (2016, and supplementary material) propose estimating this variance in the exact ESS by directly using the weighted particle approximation of $\eta_{p-1} = \pi_{p-1}$. This results in a quantity that the authors define as the *conditional effective sample size* (CESS). In the pth time step this is computed as

$$\begin{aligned} \mathrm{CESS}_p^N &:= \frac{N\left(\sum_{i'=1}^N W_{p-1}^{i'} G_{p-1}(\zeta_{p-1}^{i'})\right)^2}{\sum_{i=1}^N W_{p-1}^{i}\left(G_{p-1}(\zeta_{p-1}^{i})\right)^2} \\ &= \left[\sum_{i=1}^N N W_{p-1}^{i}\left(\frac{G_{p-1}(\zeta_{p-1}^{i})}{\sum_{i'=1}^N N W_{p-1}^{i'} G_{p-1}(\zeta_{p-1}^{i'})}\right)^2\right]^{-1}. \end{aligned} \tag{3.4}$$

In general, this differs from the empirical effective sample size ESS_p^N defined in (1.23): for $p > 1$, the equality $\mathrm{CESS}_p^N = \mathrm{ESS}_p^N$ holds only if resampling has been carried out in the previous $(p-1)$th time step.

Similarly to ESS-based adaptive temperature selection, the idea is to choose each β_p such that $\mathrm{CESS}_p^N = \mathrm{CESS}^\star$, for some value $\mathrm{CESS}^\star \in [1, N]$. The benefit of using such a criterion is that the discrepancy between successive distributions may be controlled directly, such that the chi-squared distance between each pair of distributions is roughly constant. Larger values of $\mathrm{CESS}^\star$ result in lower chi-squared distances (and therefore, in practice, a longer sequence of tempered distributions). In contrast it is noted by Zhou et al. (2016) that, unless resampling is used in every iteration, the ESS-based adaptive procedure does not result in an approximately constant discrepancy between successive distributions. It is therefore proposed to utilise the ESS only for the purposes of adaptive resampling, alongside use of the CESS for adaptive temperature selection.

We finally comment that in order to form a weighted empirical measure η_n^N approximating $\pi_\star$, for example as in (1.17), there are two possible applications of adaptive SMC

procedures (including those described here, as well as the adaptive resampling methods described in Section 1.3.2). The methods could be used 'as is', with particle approximations formed directly from the output of the resulting adaptive SMC sampler. Alternatively, such methods could be used in a 'pilot run' purely for the purpose of generating a distribution and/or resampling schedule, which can then be used in a second (non-adaptive) SMC sampler. This has some advantages; for example, given the fixed choice of schedules obtained in the pilot run, the normalising constant estimator $\gamma_n^N(\mathsf{X})$ obtained from the final run is unbiased, following Proposition 1.2.

3.2.3. Path sampling

Consider a family of probability measures $\{\varrho_\beta : \beta \in [0,1]\}$ that smoothly interpolate between $\varrho_0 := \pi_0$ and $\varrho_1 := \pi_\star$ as β increases from 0 to 1. For each $\beta \in [0,1]$ let $\bar{\varrho}_\beta$ be the corresponding unnormalised density (with respect to a common dominating measure $\mathrm{d}x$), so that

$$\varrho_\beta(A) = \frac{1}{\mathcal{Z}_\beta} \int_A \bar{\varrho}_\beta(x)\,\mathrm{d}x,$$

where $\mathcal{Z}_\beta := \int_{\mathsf{X}} \bar{\varrho}_\beta(x)\,\mathrm{d}x$ is unknown. Within the context of tempering, each density $\bar{\varrho}_\beta$ takes the form (2.3); that is,

$$\bar{\varrho}_\beta(x) = \bar{\pi}_0(x)^{1-\beta}\bar{\pi}_\star(x)^\beta, \qquad x \in \mathsf{X}, \quad \beta \in [0,1], \tag{3.5}$$

so that $\bar{\varrho}_0 = \bar{\pi}_0$, $\bar{\varrho}_1 = \bar{\pi}_\star$, $\mathcal{Z}_0 = Z_0$, $\mathcal{Z}_1 = Z_\star$.

As previously described, there are many settings in which one wishes to perform inference on (ratios of) normalising constants, such as model comparison. A general method for obtaining suitable estimators relies on the *path sampling* identity (Gelman and Meng, 1998); this states that, if each $\bar{\varrho}_\beta(x)$ is differentiable with respect to β, then the logarithm of the ratio of normalising constants satisfies

$$\begin{aligned}\log\left(\frac{Z_\star}{Z_0}\right) = \log\left(\frac{\mathcal{Z}_1}{\mathcal{Z}_0}\right) &= \int_0^1 \left[\frac{1}{\mathcal{Z}_\beta}\int_{\mathsf{X}} \frac{\mathrm{d}\log\bar{\varrho}_\beta(x)}{\mathrm{d}\beta}\bar{\varrho}_\beta(x)\,\mathrm{d}x\right]\mathrm{d}\beta \\ &= \int_0^1 \varrho_\beta\left(\frac{\mathrm{d}\log\bar{\varrho}_\beta}{\mathrm{d}\beta}\right)\mathrm{d}\beta. \end{aligned} \tag{3.6}$$

For each β in a 'path' of discrete values between 0 and 1, one may form a Monte Carlo estimator of the integrand of (3.6) by generating samples distributed according to ϱ_β. Using numerical integration (for example, the 'trapezium' quadrature rule), a biased estimator of $\log(Z_\star/Z_0)$ may therefore be obtained. Such an estimation procedure is known as *thermodynamic integration*, owing to its applications in theoretical physics.

Although the SMC sampler framework naturally provides an unbiased estimator of $Z_\star/Z_0$, estimators based on the path sampling identity can also be computed. For un-

normalised densities of the form (3.5), the identity (3.6) may be expressed as

$$\log\left(\frac{Z_\star}{Z_0}\right) = \int_0^1 \varrho_\beta\left(\log\frac{\bar{\pi}_\star(\cdot)}{\bar{\pi}_0(\cdot)}\right)\mathrm{d}\beta. \tag{3.7}$$

Running an SMC sampler for the sequence of tempered distributions defined by (2.4) gives particle approximations of each distribution $\pi_p = \varrho_{\beta_p}$, so that the integrand of (3.7) may be estimated at values of β provided by the temperature schedule. Applications of the path sampling estimator in an SMC setting may be seen in Johansen et al. (2006) and Zhou et al. (2016), among others. In general, however, this approach may be used with any scheme that permits sampling from ϱ_β at given values of $\beta \in [0, 1]$.

Many approaches to temperature schedule selection for path sampling are aimed at minimising the bias of the resulting estimator, rather than the variance: such approaches frequently assume that arbitrarily many IID samples from each tempering distribution may be obtained, and solely consider the bias due to the discretisation error of the quadrature rule. While these results are therefore not necessarily directly applicable to SMC samplers (for which an unbiased estimator exists, and for which the samples are not IID), a brief review is provided here for completeness.

In an application of power posteriors (introduced in Section 2.2.1), Lartillot and Philippe (2006) only considered the use of uniformly-spaced β; that is, $\beta_p = p/n$ for $p \in \{0, \dots, n\}$. Friel and Pettitt (2008, Section 4.1), using power posteriors for the purpose of computing a Bayes factor, empirically investigated a number of temperature schedules generated by $\beta_p = (p/n)^c$ for $c > 0$. For a specific linear regression problem, the best results (in terms of a low bias and standard error) were obtained using $c = 3$ or 5, and n between 20 and 100.

Schedules of this form were considered further by Calderhead and Girolami (2009, Section 3.2.1), with the aim of minimising a sum of symmetrised Kullback–Leibler divergences, corresponding to biases in marginal likelihood estimates. The best empirical results were found when $\beta_p = (p/n)^c$ with $c = 5$; other schedules considered included those of the form $\beta_p = 1 - (p/n)^c$, which place more inverse temperatures close to 1 (and performed much less favourably). With similar aims, Friel et al. (2014) propose an adaptive temperature selection procedure requiring computation of the integrand of (3.6), empirically showing improvements over the schedule given by $\beta_p = (p/n)^5$ at little extra computational cost.

3.3. A criterion for optimality

We now return to the more general nonparametric problem of selecting a distribution schedule, rather than solely considering the constrained problem associated with tempering. In the remainder of this chapter we present a mathematical formulation of the schedule selection problem introduced in Section 3.1, which shall form the focus of subsequent chapters.

To construct a quantity that may be used to compare different sequences of distributions $(\pi_p)_{p=0}^n$, we consider the problem of estimating the normalising constant $\gamma_n(\mathsf{X}) = \gamma_n(\mathbb{1})$. Using Algorithm 2.1, this may be estimated by

$$\gamma_n^N(\mathbb{1}) = \left(\prod_{j=0}^{r_n-1} \frac{1}{N}\sum_{i=1}^{N}\left[\prod_{p=k_j}^{k_{j+1}-1} \frac{\bar{\pi}_{p+1}(\zeta_p^i)}{\bar{\pi}_p(\zeta_p^i)}\right]\right)\left(\frac{1}{N}\sum_{i=1}^{N}\left[\prod_{p=k_{r_n}}^{n-1} \frac{\bar{\pi}_{p+1}(\zeta_p^i)}{\bar{\pi}_p(\zeta_p^i)}\right]\right). \tag{3.8}$$

This follows directly from (1.18), expressing each potential function G_p according to (2.2). By Proposition 1.2, this estimator is unbiased for any number of particles N, at least when the distribution and resampling schedules are chosen deterministically (i.e. not using adaptive procedures).

Recall that in the setting of an SMC sampler targeting $\pi_\star$ with an initial distribution π_0, $\gamma_n(\mathbb{1})$ is equal to $Z_\star/Z_0$, the ratio of the normalising constants of these distributions' unnormalised densities. As discussed in Chapter 2, normalising constants and their ratios are the primary objects of inference in many settings, such as Bayes factors in model comparison problems, or the probabilities of rare events. In these cases, a clear objective in schedule selection is to minimise the variance of $\gamma_n^N(\mathbb{1})$.

We argue that such an objective may be useful more generally, even when other estimators $\gamma_n^N(\varphi)$ and $\eta_n^N(\varphi)$ are of more direct interest. Noting that (3.8) depends on all the incremental weights computed throughout the procedure, a low variance of this estimator may be suggestive of low variances of more general estimators. Similarly, if $\gamma_n(\mathbb{1}) = \gamma_n(\mathsf{X})$ is estimated with low variance, then this may indicate that the space X has been well explored, with particles in all areas of high mass of $\pi_\star$.

Let us introduce some notation. Denote by $\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n)$ the relative asymptotic variance (1.24) of the normalising constant estimator $\gamma_n^N(\mathbb{1})$ when the distribution schedule is $\pi_{0:n}$, and the subset of time indices in which resampling takes place is $R_n \subseteq \{1, \dots, n\}$. Formally, defining $\gamma_n^N(\mathbb{1})$ as in (3.8), we denote

$$\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n) := \lim_{N\to\infty} N \operatorname{var}\left(\frac{\gamma_n^N(\mathbb{1})}{\gamma_n(\mathbb{1})}\right), \tag{3.9}$$

where $R_n := \{k_j : j \in \{1, \dots, r_n\}\}$. Although such a quantity also depends on the Markov kernels M_p, we omit this from the notation, assuming some fixed method of constructing these kernels to leave each π_p invariant. When it is clear from the context, we shall sometimes suppress the dependence of this quantity on the distribution and resampling schedules, writing $\sigma^2_{\mathbb{1}} := \sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n)$.

From the definition of the relative asymptotic variance as a limit in the number of particles N, a useful approximation when N is sufficiently large is

$$\operatorname{var}\left(\gamma_n^N(\mathbb{1})\right) \propto \operatorname{var}\left(\frac{\gamma_n^N(\mathbb{1})}{\gamma_n(\mathbb{1})}\right) \approx \frac{\sigma^2_{\mathbb{1}}}{N} = \frac{n\sigma^2_{\mathbb{1}}}{nN}.$$

As earlier mentioned, the time complexity of Algorithm 2.1 is $\mathcal{O}(nN)$. Therefore if the total computational time available is fixed, and N is sufficiently large, the variance of the normalising constant estimator is minimised approximately when $n\sigma_{\mathbb{1}}^2$ is minimised.

The relevance of this quantity is further motivated by the notion of algorithmic efficiency. A metric commonly used to compare simulation algorithms is the variance of some estimator of interest, multiplied by the cost of its computation. Hammersley and Handscomb (1964, page 22) define the *efficiency* of a Monte Carlo method to be the reciprocal of this cost–variance product; this definition was formalised and extended by Glynn and Whitt (1992), who consider the asymptotic efficiency as the computational budget tends to infinity.

For a fixed choice of distribution schedule and resampling schedule, the time cost of the SMC sampler is determined by the number of particles N. For large N, we see that this computational cost is approximately proportional to nN, and the variance of $\gamma_n^N(\mathbb{1})$ is approximately proportional to $\sigma_{\mathbb{1}}^2/N$. The reciprocal of the cost–variance product $n\sigma_{\mathbb{1}}^2$ may therefore be viewed as an approximate 'asymptotic efficiency value', as defined by Glynn and Whitt (1992, Section 2), which may be used to compare the asymptotic efficiency of estimators of the normalising constant.

We therefore propose using exactly this quantity to assess the performance of distribution schedules and resampling schedules; that is, n times the relative asymptotic variance as $N \longrightarrow \infty$ of $\gamma_n^N(\mathbb{1})$. Correspondingly we propose that finding the optimal solution to the schedule selection problem, as described through the questions posed in Section 3.1, is equivalent to determining

$$\underset{(n,\, \pi_{1:n-1},\, R_n)}{\arg\min} \; n\sigma_{\mathbb{1}}^2(\pi_{0:n}, R_n).$$

This criterion has a number of advantages. The use of the *asymptotic* variance is beneficial because, by definition, it is independent of the number of particles used. This allows the performance of a sequence of distributions to be measured in a way that is not influenced by the choice of N, which may then simply be chosen to be as large as is feasible. Furthermore, as described in Section 1.4.1 relative asymptotic variances of the form (1.24) admit a decomposition that facilitates their analysis; we shall exploit this in the following section.

On a closely related note, as discussed in Section 3.1 the specification of a distribution schedule requires a choice of the length n. By increasing n it is possible to make consecutive distributions more similar (in a manner that we shall formalise), which may result in lower variances of the incremental weights, and of estimators such as $\gamma_n^N(\mathbb{1})$. The chosen quantity accounts for this, while appropriately penalising longer (and more computationally costly) sequences.

Finally, as discussed in Section 1.5 a number of techniques have been proposed for estimating relative asymptotic variances of $\gamma_n^N(\varphi)$. These include the previously-introduced

variance estimators of Lee and Whiteley (2018), which may be computed as a by-product of running the SMC sampler. For any given schedule, the quantity $n\sigma_{\mathbb{1}}^2$ may therefore be estimated at low computational cost. This raises the possibility of using such estimators in practical procedures for schedule selection; we will explore this idea further in Section 5.2.

3.4. The relative asymptotic variance decomposition

We now proceed to derive some properties of the relative asymptotic variance as $N \longrightarrow \infty$ of estimators $\gamma_n^N(\varphi)$, in the setting of SMC samplers. Within this thesis the results we obtain will be of direct interest only when $\varphi = \mathbb{1}$, in which case this asymptotic variance corresponds to the quantity $\sigma_{\mathbb{1}}^2$ introduced above. Nevertheless, for purposes of exposition we shall derive all results in this section in the general case, before explicitly stating the results when $\varphi = \mathbb{1}$ in Section 3.4.2.

We note that similar calculations to those considered in this section have previously been encountered by other authors in smoothing contexts; these include the results of Del Moral et al. (2010) and Douc et al. (2011). The expressions we derive are specific to the SMC sampler setting we consider, and will be used extensively in later chapters.

We begin by considering the setting in which resampling is used in every iteration of the SMC sampler, i.e. using a resampling schedule $R_n = \{1, \dots, n\}$, equivalent to a form of Algorithm 1.2. The generalisation to settings in which occasional resampling is used follows by consideration of excursion models as in Section 1.3.1.1, and is discussed subsequently.

Recall that in this case, the SMC particle approximation γ_n^N is of the form (1.15). For a bounded $\mathcal{X}$-measurable function $\varphi : \mathsf{X} \longrightarrow \mathbb{R}$, the relative asymptotic variance of $\gamma_n^N(\varphi)$ may be decomposed according to (1.26) as $\sum_{p=0}^n v_{p,n}(\varphi)$. Each term $v_{p,n}(\varphi)$ admits an expression (1.29); since $\eta_p = \pi_p$ for an SMC sampler, we may initially re-express this as

$$v_{p,n}(\varphi) = \frac{\pi_p(Q_{p,n}(\varphi)^2)}{\pi_p(Q_{p,n}(\mathbb{1}))^2} - \pi_n(\varphi)^2. \tag{3.10}$$

Within the remainder of this section we simplify this further, by deriving a more explicit expression for $Q_{p,n}(\varphi)(x_p) \coloneqq \int_{\mathsf{X}} Q_{p,n}(x_p, \mathrm{d}x_n)\varphi(x_n)$.

As previously mentioned, within this thesis we shall only consider SMC samplers of the form presented in Algorithm 2.1. That is, we consider the case where $M_0 = \pi_0$, each Markov kernel M_p admits π_p as an invariant distribution, and the potential functions G_p take the form (2.2). In this common setting we may express $Q_{p,n}(\varphi)$, and therefore $v_{p,n}(\varphi)$, in terms of the Markov kernels $(M_p)_{p=1}^n$.

Many of the expressions that we derive exhibit similarities to the smoothing results of Del Moral et al. (2010) and Douc et al. (2011), which are based on appropriately-defined 'backward kernels'. We also begin by introducing such a definition. Let $K : \mathsf{X} \times \mathcal{X} \longrightarrow [0, 1]$ be a Markov kernel on $(\mathsf{X}, \mathcal{X})$ that leaves some probability measure π invariant; that is,

$\pi K = \pi$. For any $A \in \mathcal{X}$, consider also the integral kernel $\mathbb{1}_A K$, recalling that this is defined such that for any $x \in \mathsf{X}$ and $B \in \mathcal{X}$,

$$\mathbb{1}_A K(x, B) := \mathbb{1}_A(x) K(x, B).$$

The measure $\pi(\mathbb{1}_A K)$ on $(\mathsf{X}, \mathcal{X})$ is then given by, for any $B \in \mathcal{X}$,

$$\pi(\mathbb{1}_A K)(B) = \int_X \pi(\mathrm{d}x) \mathbb{1}_A(x) K(x, B) = \int_A \pi(x) K(x, B).$$

We see straightforwardly that for any $A \in \mathcal{X}$, $\pi(\mathbb{1}_A K)$ is absolutely continuous with respect to $\pi K = \pi$. It follows that the Radon–Nikodym derivative $\mathrm{d}\pi(\mathbb{1}_A K)/\mathrm{d}\pi$ is well defined, allowing us to define the following Markov kernel.

Definition 3.2. Let $K : \mathsf{X} \times \mathcal{X} \longrightarrow [0, 1]$ be a Markov kernel on $(\mathsf{X}, \mathcal{X})$ that leaves some probability measure π invariant. The *time reversal* of K, denoted $K^* : \mathsf{X} \times \mathcal{X} \longrightarrow [0, 1]$, is defined for $x \in \mathsf{X}$, $A \in \mathcal{X}$ by

$$K^*(x, A) := \frac{\mathrm{d}\pi(\mathbb{1}_A K)}{\mathrm{d}\pi}(x).$$

The time reversal is so named since it satisfies the property

$$\int_A \pi(\mathrm{d}x) \int_B K(x, \mathrm{d}x') = \int_B \pi(\mathrm{d}x') \int_A K^*(x', \mathrm{d}x) \quad \text{for all } A, B \in \mathcal{X}. \tag{3.11}$$

Taking $B = \mathsf{X}$ in this expression gives $\pi(A) = \int_{\mathsf{X}} \pi(\mathrm{d}x') K^*(x', A)$, from which we see that K^* leaves π invariant.

We now return to the problem of deriving an expression for $Q_{p,n}(\varphi)$, as appears in the terms (3.10) of the relative asymptotic variance decomposition. In preparation, we first present a lemma relating to the kernels $(Q_p)_{p=1}^n$ defined in (1.27), recalling from (1.28) that $Q_{p,n} = Q_{p+1} \cdots Q_n$ for $p < n$, and $Q_{n,n} = \mathrm{Id}$. Considering the time reversals $(M_p^*)_{p=1}^n$ of the Markov kernels $(M_p)_{p=1}^n$ used in the SMC sampler, we obtain the following result.

Lemma 3.3. *Let $M_0 = \pi_0$, M_p be a Markov kernel admitting π_p as an invariant distribution (for $p \geq 1$), and each G_p take the form* (2.2)*. For $p \in \{1, \ldots, n\}$ let Q_p be defined as in* (1.27)*. Then for all $A, B \in \mathcal{X}$,*

$$\int_A \pi_{p-1}(\mathrm{d}x_{p-1}) \int_B Q_p(x_{p-1}, \mathrm{d}x_p) = \frac{Z_p}{Z_{p-1}} \int_B \pi_p(\mathrm{d}x_p) \int_A M_p^*(x_p, \mathrm{d}x_{p-1}).$$

Proof. Let $p \in \{1, \dots, n\}$ and $A, B \in \mathcal{X}$. Using the definition (1.27) of Q_p, we have

$$\begin{aligned}&\int_A \pi_{p-1}(\mathrm{d}x_{p-1}) \int_B Q_p(x_{p-1}, \mathrm{d}x_p) \\ &= \int_A \pi_{p-1}(\mathrm{d}x_{p-1}) G_{p-1}(x_{p-1}) \int_B M_p(x_{p-1}, \mathrm{d}x_p).\end{aligned}$$

Writing $G_{p-1}(x_{p-1})$ in the form (2.2), this is equal to

$$\begin{aligned}&\frac{Z_p}{Z_{p-1}} \int_A \pi_{p-1}(\mathrm{d}x_{p-1}) \frac{\mathrm{d}\pi_p}{\mathrm{d}\pi_{p-1}}(x_{p-1}) \int_B M_p(x_{p-1}, \mathrm{d}x_p) \\ &= \frac{Z_p}{Z_{p-1}} \int_A \pi_p(\mathrm{d}x_{p-1}) \int_B M_p(x_{p-1}, \mathrm{d}x_p) \\ &= \frac{Z_p}{Z_{p-1}} \int_B \pi_p(\mathrm{d}x_p) \int_A M_p^*(x_p, \mathrm{d}x_{p-1}),\end{aligned}$$

where the last step follows from property (3.11) of the time reversal kernel M_p^*. ∎

We may now use this result to derive an expression for the function $Q_{p,n}(\varphi) : \mathsf{X} \to \mathbb{R}$, as appears in (3.10). To facilitate this, we define a sequence of kernels $(M_{n,p})_{p=0}^n$ by taking compositions of the time reversal kernels $(M_p^*)_{p=1}^n$. Specifically, for $p \in \{0, \dots, n\}$ define $M_{n,p} : \mathsf{X} \times \mathcal{X} \to [0, 1]$ by

$$M_{n,n} := \mathrm{Id}; \qquad M_{n,p} := M_n^* \cdots M_{p+1}^*, \ p \in \{0, \dots, n-1\}. \tag{3.12}$$

This allows us to express $Q_{p,n}(\varphi)$ in terms of a Radon–Nikodym derivative.

Proposition 3.4. *Let $M_0 = \pi_0$, M_p be a Markov kernel admitting π_p as an invariant distribution (for $p \geq 1$), and each G_p take the form* (2.2)*. For $p \in \{0, \dots, n\}$ let $Q_{p,n}$ be defined as in* (1.28)*, and let $\varphi : \mathsf{X} \to \mathbb{R}$ be a bounded $\mathcal{X}$-measurable function. Then the function $Q_{p,n}(\varphi) : \mathsf{X} \to \mathbb{R}$, defined such that $Q_{p,n}(\varphi)(x_p) := \int_{\mathsf{X}} Q_{p,n}(x_p, \mathrm{d}x_n)\varphi(x_n)$ for $x_p \in \mathsf{X}$, is π_p-almost everywhere of the form*

$$Q_{p,n}(\varphi)(x_p) = \frac{Z_n}{Z_p} \frac{\mathrm{d}\pi_n(\varphi M_{n,p})}{\mathrm{d}\pi_p}(x_p),$$

where the signed measure $\pi_n(\varphi M_{n,p})$ is given by, for $A \in \mathcal{X}$,

$$\pi_n(\varphi M_{n,p})(A) = \int_{\mathsf{X}} \pi_n(\mathrm{d}x_n)\varphi(x_n) M_{n,p}(x_n, A).$$

Proof. We begin with the case $p = n$. Since $Q_{n,n} := \mathrm{Id}$ one has that for all $x_n \in \mathsf{X}$,

$$Q_{n,n}(\varphi)(x_n) = \mathrm{Id}(\varphi)(x_n) = \varphi(x_n). \tag{3.13}$$

Now for some $A \in \mathcal{X}$, consider the integral

$$\int_A \pi_n(\mathrm{d}x_n)\left[\frac{Z_n}{Z_n}\frac{\mathrm{d}\pi_n(\varphi M_{n,n})}{\mathrm{d}\pi_n}(x_n)\right] = \pi_n(\varphi M_{n,n})(A) = \int_{\mathsf{X}} \pi_n(\mathrm{d}x_n)\varphi(x_n)M_{n,n}(x_n, A). \tag{3.14}$$

Since $M_{n,n} := \mathrm{Id}$, we have $M_{n,n}(x_n, A) = \mathrm{Id}(x_n, A) = \delta_{x_n}(A) = \mathbb{1}_A(x_n)$. Therefore we may re-express (3.14) as

$$\int_{\mathsf{X}} \pi_n(\mathrm{d}x_n)\varphi(x_n)\mathbb{1}_A(x_n) = \int_A \pi_n(\mathrm{d}x_n)\varphi(x_n).$$

It follows that for π_n-almost all $x_n \in \mathsf{X}$ the bracketed expression in (3.14) is equal to $\varphi(x_n)$. Following (3.13), this is equal to $Q_{n,n}(\varphi)(x_n)$. The statement therefore holds for $p = n$.

Proceeding inductively, suppose that the statement is true for $p = q \geq 1$. For some $A \in \mathcal{X}$, consider the integral

$$\begin{aligned}&\int_A \pi_{q-1}(\mathrm{d}x_{q-1})Q_{q-1,n}(\varphi)(x_{q-1}) \\ &= \int_A \pi_{q-1}(\mathrm{d}x_{q-1})\int_{\mathsf{X}} Q_{q-1,n}(x_{q-1}, \mathrm{d}x_n)\varphi(x_n).\end{aligned} \tag{3.15}$$

By (1.28) we have $Q_{q-1,n} = Q_q Q_{q,n}$, and so we may re-express this as

$$\begin{aligned}&\int_A \pi_{q-1}(\mathrm{d}x_{q-1})\int_{\mathsf{X}} Q_q(x_{q-1}, \mathrm{d}x_q)\int_{\mathsf{X}} Q_{q,n}(x_q, \mathrm{d}x_n)\varphi(x_n) \\ &= \int_A \pi_{q-1}(\mathrm{d}x_{q-1})\int_{\mathsf{X}} Q_q(x_{q-1}, \mathrm{d}x_q)Q_{q,n}(\varphi)(x_q).\end{aligned}$$

Applying Lemma 3.3, this is equal to

$$\frac{Z_q}{Z_{q-1}}\int_{\mathsf{X}} \pi_q(\mathrm{d}x_q)Q_{q,n}(\varphi)(x_q)\int_A M_q^*(x_q, \mathrm{d}x_{q-1}).$$

Applying the inductive assumption, we see that this is equal to

$$\begin{aligned}&\frac{Z_q}{Z_{q-1}}\frac{Z_n}{Z_q}\int_{\mathsf{X}} \pi_q(\mathrm{d}x_q)\frac{\mathrm{d}\pi_n(\varphi M_{n,q})}{\mathrm{d}\pi_q}(x_q)\int_A M_q^*(x_q, \mathrm{d}x_{q-1}) \\ &= \frac{Z_n}{Z_{q-1}}\int_{\mathsf{X}} \pi_n(\varphi M_{n,q})(\mathrm{d}x_q)M_q^*(x_q, A) \\ &= \frac{Z_n}{Z_{q-1}}\int_{\mathsf{X}} \pi_n(\mathrm{d}x_n)\varphi(x_n)\int_{\mathsf{X}} M_{n,q}(x_n, \mathrm{d}x_q)M_q^*(x_q, A);\end{aligned}$$

by (3.12) we have that $M_{n,q}M_q^* = M_{n,q-1}$, and so (3.15) may be finally simplified as

$$\frac{Z_n}{Z_{q-1}}\int_{\mathsf{X}} \pi_n(\mathrm{d}x_n)\varphi(x_n)M_{n,q-1}(x_n, A) = \frac{Z_n}{Z_{q-1}}\pi_n(\varphi M_{n,q-1})(A). \tag{3.16}$$

Now suppose $\pi_{q-1}(A) = 0$. Then the integral (3.15) equals zero, and so (3.16) is also zero. It follows that $\pi_n(\varphi M_{n,q-1}) \ll \pi_{q-1}$. Comparing (3.15) and (3.16) we see that $Q_{q-1,n}(\varphi)$ is

equal to Z_n/Z_{q-1} times the corresponding Radon–Nikodym derivative, as required.

The result therefore holds for all $p \in \{0, \dots, n\}$. ∎

Using this result, we may re-express (3.10) to obtain a more explicit expression for each $v_{p,n}(\varphi)$ term in this SMC sampler setting.

Proposition 3.5. *Let $M_0 = \pi_0$, M_p be a Markov kernel admitting π_p as an invariant distribution (for $p \geq 1$), and each G_p take the form (2.2). Let $\varphi : \mathsf{X} \to \mathbb{R}$ be a bounded $\mathcal{X}$-measurable function. Then for $p \in \{0, \dots, n\}$,*

$$v_{p,n}(\varphi) = \int_{\mathsf{X}} \left(\frac{\mathrm{d}\pi_n(\varphi M_{n,p})}{\mathrm{d}\pi_p}(x_p) \right)^2 \pi_p(\mathrm{d}x_p) \; - \; \pi_n(\varphi)^2,$$

where the signed measure $\pi_n(\varphi M_{n,p})$ is as given in Proposition 3.4.

Proof. Consider the expression (3.10) for $v_{p,n}(\varphi)$ in this SMC sampler setting. Using the form of $Q_{p,n}(\varphi)$ from Proposition 3.4, the numerator of the fraction in this expression may be rewritten as

$$\begin{aligned} \pi_p(Q_{p,n}(\varphi)^2) &= \int_{\mathsf{X}} Q_{p,n}(\varphi)(x_p)^2 \pi_p(\mathrm{d}x_p) \\ &= \left(\frac{Z_n}{Z_p} \right)^2 \int_{\mathsf{X}} \left(\frac{\mathrm{d}\pi_n(\varphi M_{n,p})}{\mathrm{d}\pi_p}(x_p) \right)^2 \pi_p(\mathrm{d}x_p). \end{aligned} \tag{3.17}$$

Now consider the denominator of the fraction in (3.10), which is the square of $\pi_p(Q_{p,n}(\mathbb{1}))$. Again using Proposition 3.4, and noting that $\mathbb{1} M_{n,p} = M_{n,p}$, we have that

$$\begin{aligned} \pi_p(Q_{p,n}(\mathbb{1})) &= \int_{\mathsf{X}} Q_{p,n}(\mathbb{1})(x_p) \pi_p(\mathrm{d}x_p) \\ &= \frac{Z_n}{Z_p} \int_{\mathsf{X}} \frac{\mathrm{d}\pi_n M_{n,p}}{\mathrm{d}\pi_p}(x_p) \pi_p(\mathrm{d}x_p) \\ &= \frac{Z_n}{Z_p} \int_{\mathsf{X}} \pi_n M_{n,p}(\mathrm{d}x_p). \end{aligned}$$

Using the definition (3.12) of $M_{n,p}$, we therefore have

$$\pi_p(Q_{p,n}(\mathbb{1})) \; = \; \frac{Z_n}{Z_p} \int_{\mathsf{X}} \pi_n(\mathrm{d}x_n) \int_{\mathsf{X}} M_n^*(x_n, \mathrm{d}x_{n-1}) \cdots \int_{\mathsf{X}} M_{p+1}^*(x_{p+1}, \mathrm{d}x_p) \; = \; \frac{Z_n}{Z_p}. \tag{3.18}$$

The stated result follows directly from substituting (3.17) and (3.18) into (3.10). ∎

This expression for $v_{p,n}(\varphi)$ admits a particularly convenient form when $\varphi = \mathbb{1}$, in which case the terms $v_{p,n}(\mathbb{1})$ are those in the decomposition of the relative asymptotic variance $\sigma^2_{\mathbb{1}}$, introduced in Section 3.3. We will detail this result in Section 3.4.2 (where it is presented as Proposition 3.10), and discuss its relevance in the context of schedule selection.

3.4.1. Occasional resampling

We now discuss estimators of the form $\gamma_n^N(\varphi)$ resulting from the use of other resampling schedules $R_n \subseteq \{1, \dots, n\}$ within the SMC sampler, i.e. by using a form of Algorithm 1.3. To present the results for this setting, we continue using the notation for excursion Feynman–Kac models employed in Section 1.3.1.1; for example, we denote the resampling schedule by $R_n := \{k_j : j \in \{1, \dots, r_n\}\}$.

Since in an SMC sampler all the distributions π_p are defined on a common space X, the excursion spaces are of the form $\mathsf{X}'_j = \mathsf{X}^{k_{j+1}-k_j}$. One may also show that for $j \in \{0, \dots, r_n\}$, the normalised excursion Feynman–Kac models are of the form

$$\eta'_j(\mathrm{d}x'_j) = \pi_{k_j}(\mathrm{d}x_{k_j}) \prod_{p=k_j+1}^{k_{j+1}-1} M_p(x_{p-1}, \mathrm{d}x_p), \tag{3.19}$$

with the corresponding normalising constants being $\gamma'_j(\mathsf{X}'_j) = Z_{k_j}/Z_0$.

When occasional resampling is used, the particle approximation γ_n^N takes the general form (1.18). Recall from Section 1.4.1.2 that the relative asymptotic variance of the associated estimator $\gamma_n^N(\varphi)$ may be decomposed according to (1.34), as $\sum_{j=0}^{r_n} \hat{v}'_{j,r_n}(\varphi)$. As previously stated in (1.38), each of these $r_n + 1$ terms may be expressed as

$$\hat{v}'_{j,r_n}(\varphi) = \frac{\eta'_j(Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi})^2)}{\eta'_j(Q'_{j,r_n}(G'_{r_n}))^2} - \hat{\eta}'_{r_n}(\bar{\varphi})^2,$$

where $\bar{\varphi} : \mathsf{X}'_{r_n} \longrightarrow \mathbb{R}$ is defined as in (1.37), being such that $\bar{\varphi}(x_{k_{r_n}}, \dots, x_n) := \varphi(x_n)$.

We shall proceed to derive results relating to this decomposition, analogous to those derived in the setting in which resampling always occurs. Again, we consider the use of an SMC sampler with potential functions G_p of the form (2.2); in this case each term $\hat{v}'_{j,r_n}(\varphi)$ may be expressed in terms of a chi-squared distance, similar to the 'always resampling' case. We follow a similar approach to achieve this result, first providing a lemma relating to the kernels Q'_j defined in (1.35).

Within Lemma 3.3, we considered the time reversals $(M_p^*)_{p=1}^n$ of the Markov kernels $(M_p)_{p=1}^n$ used in the SMC sampler, defining these according to Definition 3.2. Considering the Markov kernels introduced in the construction of the excursion Feynman–Kac models, recall from (1.19) that for $j \in \{1, \dots, r_n\}$, M'_j is a Markov kernel from $(\mathsf{X}'_{j-1}, \mathcal{X}'_{j-1})$ to $(\mathsf{X}'_j, \mathcal{X}'_j)$. Since these two spaces may differ in dimension, the time reversal of M'_j is not well-defined according to Definition 3.2. However, considering the definition (1.19) of M'_j we may define an appropriate 'reversal' $(M'_j)^* : \mathsf{X}'_j \times \mathcal{X}'_{j-1} \longrightarrow [0, 1]$ by

$$(M'_j)^*(x'_j, \mathrm{d}x'_{j-1}) := \prod_{p=k_j}^{k_{j-1}+1} M_p^*(x_p, \mathrm{d}x_{p-1}). \tag{3.20}$$

This allows us to derive the following analogue of Lemma 3.3.

Lemma 3.6. *Let $M_0 = \pi_0$, M_p be a Markov kernel admitting π_p as an invariant distribution (for $p \geq 1$), and each G_p take the form* (2.2)*. For $j \in \{1, \dots, r_n\}$ let Q'_j be defined as in* (1.35)*. Then for all $A \in \mathcal{X}'_{j-1}$ and $B \in \mathcal{X}'_j$,*

$$\int_A \eta'_{j-1}(\mathrm{d}x'_{j-1}) \int_B Q'_j(x'_{j-1}, \mathrm{d}x'_j) \;=\; \frac{Z_{k_j}}{Z_{k_{j-1}}} \int_B \eta'_j(\mathrm{d}x'_j) \int_A (M'_j)^*(x'_j, \mathrm{d}x'_{j-1}).$$

Proof. Let $j \in \{1, \dots, r_n\}$, $A \in \mathcal{X}'_{j-1}$ and $B \in \mathcal{X}'_j$. Using the definition (1.35) of Q'_j, we have

$$\begin{aligned} &\int_A \eta'_{j-1}(\mathrm{d}x'_{j-1}) \int_B Q'_j(x'_{j-1}, \mathrm{d}x'_j) \\ &= \int_A \eta'_{j-1}(\mathrm{d}x'_{j-1}) G'_{j-1}(x'_{j-1}) \int_B M'_j(x'_{j-1}, \mathrm{d}x'_j). \end{aligned} \tag{3.21}$$

For notational simplicity, let us now consider the integrand of this expression; that is, $\eta'_{j-1}(\mathrm{d}x'_{j-1}) G'_{j-1}(x'_{j-1}) M'_j(x'_{j-1}, \mathrm{d}x'_j)$. Using the form (3.19) of η'_{j-1}, the form (1.20) of G'_{j-1} and the form (1.19) of M'_j, this is

$$\begin{aligned} &\left[\pi_{k_{j-1}}(\mathrm{d}x_{k_{j-1}}) \prod_{p=k_{j-1}+1}^{k_j-1} M_p(x_{p-1}, \mathrm{d}x_p)\right] \left[\prod_{p=k_{j-1}}^{k_j-1} G_p(x_p)\right] \left[\prod_{p=k_j}^{k_{j+1}-1} M_p(x_{p-1}, \mathrm{d}x_p)\right] \\ &= \pi_{k_{j-1}}(\mathrm{d}x_{k_{j-1}}) \left[\prod_{p=k_{j-1}+1}^{k_j} G_{p-1}(x_{p-1}) M_p(x_{p-1}, \mathrm{d}x_p)\right] \left[\prod_{p=k_j+1}^{k_{j+1}-1} M_p(x_{p-1}, \mathrm{d}x_p)\right] \\ &= \pi_{k_{j-1}}(\mathrm{d}x_{k_{j-1}}) \left[\prod_{p=k_{j-1}+1}^{k_j} Q_p(x_{p-1}, \mathrm{d}x_p)\right] \left[\prod_{p=k_j+1}^{k_{j+1}-1} M_p(x_{p-1}, \mathrm{d}x_p)\right], \end{aligned}$$

where, after regrouping the terms, we have used the definition (1.27) of Q_p.

By repeatedly applying Lemma 3.3 for $p \in \{k_{j-1} + 1, \dots, k_j\}$, we find that the integral (3.21) of this expression over $A \in \mathcal{X}'_{j-1}$, $B \in \mathcal{X}'_j$ is equal to that over

$$\begin{aligned} &\left[\prod_{p=k_{j-1}+1}^{k_j} \frac{Z_p}{Z_{p-1}} M_p^*(x_p, \mathrm{d}x_{p-1})\right] \pi_{k_j}(\mathrm{d}x_{k_j}) \left[\prod_{p=k_j+1}^{k_{j+1}-1} M_p(x_{p-1}, \mathrm{d}x_p)\right] \\ &= \frac{Z_{k_j}}{Z_{k_{j-1}}} \left[\prod_{p=k_{j-1}+1}^{k_j} M_p^*(x_p, \mathrm{d}x_{p-1})\right] \left[\pi_{k_j}(\mathrm{d}x_{k_j}) \prod_{p=k_j+1}^{k_{j+1}-1} M_p(x_{p-1}, \mathrm{d}x_p)\right], \end{aligned}$$

Of the two bracketed terms, we identify the former as $(M'_j)^*(x'_j, \mathrm{d}x'_{j-1})$, as defined in (3.20), and the latter as η'_j, written in the form (3.19). Therefore (3.21), which is the integral of this expression over $A \in \mathcal{X}'_{j-1}$, $B \in \mathcal{X}'_j$, is equal to

$$\frac{Z_{k_j}}{Z_{k_{j-1}}} \int_B \eta'_j(\mathrm{d}x'_j) \int_A (M'_j)^*(x'_j, \mathrm{d}x'_{j-1}),$$

as required. ∎

Using this result we now provide an analogue of Proposition 3.4: for Q'_{j,r_n} as defined in (1.36), we give an expression for the function $Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi}) : \mathsf{X}'_j \to \mathbb{R}$, as appears in (1.38). This has the form

$$Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi})(x'_j) := \int_{\mathsf{X}'_{r_n}} Q'_{j,r_n}(x'_j, \mathrm{d}x'_{r_n}) G'_{r_n}(x'_{r_n}) \bar{\varphi}(x'_{r_n}),$$

where we recall from (1.37) that $\bar{\varphi} : \mathsf{X}'_{r_n} \to \mathbb{R}$ is such that $\bar{\varphi}(x'_{r_n}) = \bar{\varphi}(x_{k_{r_n}}, \ldots, x_n) := \varphi(x_n)$.

Analogously to (3.12), we first define a sequence of kernels $(M'_{r_n,j})_{j=0}^{r_n}$ in terms of the 'reversal' kernels introduced in (3.20). For $j \in \{0, \ldots, r_n\}$ we define $M'_{r_n,j} : \mathsf{X}'_{r_n} \times \mathcal{X}'_j \to [0,1]$ by

$$M'_{r_n,r_n} := \mathrm{Id}; \qquad M'_{r_n,j} := (M'_{r_n})^* \cdots (M'_{j+1})^*, \ j \in \{0, \ldots, r_n - 1\}. \tag{3.22}$$

Proposition 3.7. *Let $M_0 = \pi_0$, M_p be a Markov kernel admitting π_p as an invariant distribution (for $p \geq 1$), and each G_p take the form (2.2). For $j \in \{0, \ldots, r_n\}$ let Q'_{j,r_n} be defined as in (1.36); also let $\varphi : \mathsf{X} \to \mathbb{R}$ be a bounded $\mathcal{X}$-measurable function, with $\bar{\varphi} : \mathsf{X}'_{r_n} \to \mathbb{R}$ defined according to (1.37). Then the function $Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi}) : \mathsf{X}'_j \to \mathbb{R}$, defined such that $Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi})(x'_j) := \int_{\mathsf{X}'_{r_n}} Q'_{j,r_n}(x'_j, \mathrm{d}x'_{r_n}) G'_{r_n}(x'_{r_n}) \bar{\varphi}(x'_{r_n})$ for $x'_j \in \mathsf{X}'_j$, is η'_j-almost everywhere of the form*

$$Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi})(x'_j) = \frac{Z_n}{Z_{k_j}} \frac{\mathrm{d}\hat{\eta}'_{r_n}(\bar{\varphi} M'_{r_n,j})}{\mathrm{d}\eta'_j}(x'_j),$$

where the signed measure $\hat{\eta}'_{r_n}(\bar{\varphi} M'_{r_n,j})$ is given by, for $A \in \mathcal{X}'_j$,

$$\hat{\eta}'_{r_n}(\bar{\varphi} M'_{r_n,j})(A) = \int_{\mathsf{X}'_n} \hat{\eta}'_{r_n}(\mathrm{d}x'_n) \bar{\varphi}(x'_n) M'_{r_n,j}(x'_n, A).$$

Proof. We begin with the case $j = r_n$. Since $Q'_{r_n,r_n} := \mathrm{Id}$ one has that for all $x'_{r_n} \in \mathsf{X}'_{r_n}$,

$$Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi})(x'_{r_n}) = \mathrm{Id}(G'_{r_n} \cdot \bar{\varphi})(x'_{r_n}) = G'_{r_n}(x'_{r_n}) \bar{\varphi}(x'_{r_n}). \tag{3.23}$$

Now for some $A \in \mathcal{X}'_{r_n}$, consider the integral

$$\begin{aligned} \int_A \eta'_{r_n}(\mathrm{d}x'_{r_n}) \left[\frac{Z_n}{Z_{k_{r_n}}} \frac{\mathrm{d}\hat{\eta}'_{r_n}(\bar{\varphi} M'_{r_n,r_n})}{\mathrm{d}\eta'_{r_n}} \right] &= \frac{Z_n}{Z_{k_{r_n}}} \hat{\eta}'_{r_n}(\bar{\varphi} M'_{r_n,r_n})(A) \\ &= \int_{\mathsf{X}'_{r_n}} \frac{Z_n}{Z_{k_{r_n}}} \hat{\eta}'_{r_n}(\mathrm{d}x'_{r_n}) \bar{\varphi}(x'_{r_n}) M'_{r_n,r_n}(x'_{r_n}, A). \end{aligned} \tag{3.24}$$

Since $M_{n,n} := \mathrm{Id}$, we have $M'_{r_n,r_n}(x'_{r_n}, A) = \mathrm{Id}(x'_{r_n}, A) = \delta_{x'_{r_n}}(A) = \mathbb{1}_A(x'_{r_n})$. Therefore we may re-express (3.24) as

$$\int_{\mathsf{X}'_{r_n}} \frac{Z_n}{Z_{k_{r_n}}} \hat{\eta}'_{r_n}(\mathrm{d}x'_{r_n}) \bar{\varphi}(x'_{r_n}) \mathbb{1}_A(x'_{r_n}) = \int_A \frac{Z_n}{Z_{k_{r_n}}} \hat{\eta}'_{r_n}(\mathrm{d}x'_{r_n}) \bar{\varphi}(x'_{r_n}). \tag{3.25}$$

The normalising constant ratio $Z_n / Z_{k_{r_n}}$ may be expressed in terms of the Feynman–Kac

models as

$$\frac{Z_n}{Z_{k_{r_n}}} = \frac{Z_n/Z_0}{Z_{k_{r_n}}/Z_0} = \frac{\hat{\gamma}'_{r_n}(\mathsf{X}'_{r_n})}{\gamma'_{r_n}(\mathsf{X}'_{r_n})}.$$

Therefore,

$$\frac{Z_n}{Z_{k_{r_n}}}\hat{\eta}'_{r_n}(\mathrm{d}x'_{r_n}) = \frac{\hat{\gamma}'_{r_n}(\mathsf{X}'_{r_n})\hat{\eta}'_{r_n}(\mathrm{d}x'_{r_n})}{\gamma'_{r_n}(\mathsf{X}'_{r_n})} = \frac{\hat{\gamma}'_{r_n}(\mathrm{d}x'_{r_n})}{\gamma'_{r_n}(\mathsf{X}'_{r_n})} = \frac{\gamma'_{r_n}(\mathrm{d}x'_{r_n})G'_{r_n}(x'_{r_n})}{\gamma'_{r_n}(\mathsf{X}'_{r_n})} = \eta'_{r_n}(\mathrm{d}x'_{r_n})G'_{r_n}(x'_{r_n}),$$

where we have used the definitions (1.8) and (1.11) of the normalised prediction and updated Feynman–Kac models respectively. We may therefore re-express (3.25) as

$$\int_A \eta'_{r_n}(\mathrm{d}x'_{r_n})G'_{r_n}(x'_{r_n})\bar{\varphi}(x'_{r_n}).$$

Since this is equal to (3.24), it follows that for η'_{r_n}-almost all $x'_{r_n} \in \mathsf{X}'_{r_n}$ the bracketed expression in (3.24) is equal to $G'_{r_n}(x'_{r_n})\bar{\varphi}(x'_{r_n})$. Following (3.23), this is equal to $Q'_{j,r_n}(G'_{r_n} \cdot \bar{\varphi})(x'_{r_n})$. The statement therefore holds for $j = r_n$.

To complete the proof one follows essentially the same steps as in the inductive step of the proof of Proposition 3.4; we summarise the main steps here, omitting the intermediate expressions. Suppose that the statement is true for $j = \ell \geq 1$, and for some $A \in \mathcal{X}'_{\ell-1}$ consider the integral

$$\int_A \eta'_{\ell-1}(\mathrm{d}x'_{\ell-1})Q'_{\ell-1,r_n}(G'_{r_n} \cdot \bar{\varphi})(x'_{\ell-1}). \tag{3.26}$$

By (1.36) we have $Q'_{\ell-1,r_n} = Q'_\ell Q'_{\ell,r_n}$, and so we may re-express this as

$$\int_A \eta'_{\ell-1}(\mathrm{d}x'_{\ell-1}) \int_{\mathsf{X}'_\ell} Q'_\ell(x'_{\ell-1}, \mathrm{d}x'_\ell)Q'_{\ell,r_n}(G'_{r_n} \cdot \bar{\varphi})(x'_\ell).$$

Applying Lemma 3.6, followed by the inductive assumption, gives

$$\begin{aligned}
&\frac{Z_{k_\ell}}{Z_{k_{\ell-1}}} \int_{\mathsf{X}'_\ell} \eta'_\ell(\mathrm{d}x'_\ell)Q'_{\ell,r_n}(G'_{r_n} \cdot \bar{\varphi})(x'_\ell) \int_A (M'_\ell)^*(x'_\ell, \mathrm{d}x'_{\ell-1}) \\
&= \frac{Z_{k_\ell}}{Z_{k_{\ell-1}}}\frac{Z_n}{Z_{k_\ell}} \int_{\mathsf{X}'_\ell} \eta'_\ell(\mathrm{d}x'_\ell) \frac{\mathrm{d}\hat{\eta}'_{r_n}(\bar{\varphi}M'_{r_n,\ell})}{\mathrm{d}\eta'_\ell}(x'_\ell) \int_A (M'_\ell)^*(x'_\ell, \mathrm{d}x'_{\ell-1}).
\end{aligned}$$

Using the identity $M'_{r_n,\ell}(M'_\ell)^* = M'_{r_n,\ell-1}$ as follows from (3.22), one may now follow essentially the same final steps as those in the proof of Proposition 3.4 to simplify this expression, and therefore (3.26), as

$$\frac{Z_n}{Z_{k_{\ell-1}}}\hat{\eta}'_{r_n}(\bar{\varphi}M'_{r_n,\ell-1})(A). \tag{3.27}$$

Now suppose $\eta'_{\ell-1}(A) = 0$. Then the integral (3.26) equals zero, and so (3.27) is also zero. It follows that $\hat{\eta}'_{r_n}(\bar{\varphi}M'_{r_n,\ell-1}) \ll \eta'_{\ell-1}$. Comparing (3.26) and (3.27) we see that $Q'_{\ell-1,r_n}(G'_{r_n} \cdot \bar{\varphi})$ is equal to $Z_n/Z_{k_{\ell-1}}$ times the corresponding Radon–Nikodym derivative, as required.

The result therefore holds for all $j \in \{0, \ldots, r_n\}$. ∎

Finally, the following analogue of Proposition 3.5 may then be obtained.. This gives an explicit form for the terms $\hat{v}'_{j,r_n}(\varphi)$ in the relative asymptotic variance decomposition of $\gamma_n^N(\varphi)$, for an SMC sampler with resampling schedule $R_n := \{k_j : j \in \{1, \dots, r_n\}\}$.

Proposition 3.8. *Let $M_0 = \pi_0$, M_p be a Markov kernel admitting π_p as an invariant distribution (for $p \geq 1$), and each G_p take the form (2.2). Let $\varphi : \mathsf{X} \longrightarrow \mathbb{R}$ be a bounded $\mathcal{X}$-measurable function, with $\bar{\varphi} : \mathsf{X}'_{r_n} \longrightarrow \mathbb{R}$ defined according to (1.37). Then for $j \in \{0, \dots, r_n\}$,*

$$\hat{v}'_{j,r_n}(\varphi) = \int_{\mathsf{X}'_j} \left(\frac{\mathrm{d}\hat{\eta}'_{r_n}(\bar{\varphi} M'_{r_n,j})}{\mathrm{d}\eta'_j}(x'_j) \right)^2 \eta'_j(\mathrm{d}x'_j) - \pi_n(\varphi)^2. \tag{3.28}$$

The proof is essentially identical to that of Proposition 3.5, requiring the application of Proposition 3.7 to expression (1.38).

3.4.2. The normalising constant estimator

To complete this section, we now consider some properties of the estimator $\gamma_n^N(\mathbb{1})$ of the normalising constant of γ_n, and the decomposition of its relative asymptotic variance as $N \longrightarrow \infty$. Following Section 3.3 this quantity will be of particular interest to us in comparing distribution and resampling schedules for SMC samplers; we shall therefore make extensive use of the results in this section throughout the following chapters.

As usual, we begin by considering the use of an SMC algorithm employing resampling in every iteration, i.e. Algorithm 1.2. In this case the estimator of the normalising constant takes the form

$$\gamma_n^N(\mathbb{1}) = \prod_{p=0}^{n-1} \frac{1}{N} \sum_{i=1}^{N} G_p(\zeta_p^i) \tag{3.29}$$

which follows directly from (1.15). We observe that this estimator is independent of the final set of particles $(\zeta_n^i)_{i=1}^N$.

The relative asymptotic variance of this estimator may be decomposed according to (1.26) as $\sum_{p=0}^{n} v_{p,n}(\mathbb{1})$. An expression for each term in this decomposition may be obtained by taking $\varphi = \mathbb{1}$ in (1.29); noting that $\eta_n(\mathbb{1}) = 1$ since η_n is a probability measure, we obtain

$$v_{p,n}(\mathbb{1}) = \frac{\eta_p(Q_{p,n}(\mathbb{1})^2)}{\eta_p(Q_{p,n}(\mathbb{1}))^2} - 1. \tag{3.30}$$

Remark 3.9. Consider (3.30) in the case $p = n$. Since $Q_{n,n} := \mathrm{Id}$ by (1.28), and $\mathrm{Id}(x, \cdot) = \delta_x(\cdot)$ is a probability measure for all $x \in \mathsf{X}_n$, we have $Q_{n,n}(\mathbb{1}) = \mathbb{1}$. Therefore, $\eta_n(Q_{n,n}(\mathbb{1})) = \eta_n(Q_{n,n}(\mathbb{1})^2) = 1$. It follows that $v_{n,n}(\mathbb{1}) = 0$ for any Feynman–Kac model; as such, when $\varphi = \mathbb{1}$ the final term in the decomposition (1.26) vanishes.

We now specifically consider this estimator as formed using an SMC sampler; in this case the expression (3.29) for $\gamma_n^N(\mathbb{1})$ may be computed by expressing the the potential functions

G_p according to (2.2). The relative asymptotic variance of this estimator corresponds to $\sigma^2_{\mathbb{1}}$ as defined in (3.9), for the resampling schedule $R_n = \{1, \dots, n\}$. We therefore have

$$\sigma^2_{\mathbb{1}}(\pi_{0:n}, \{1, \dots, n\}) = \sum_{p=0}^{n} v_{p,n}(\mathbb{1}). \tag{3.31}$$

An expression for each of these terms may obtained by taking $\varphi = \mathbb{1}$ in Proposition 3.5. Indeed, in this case we may show that each term $v_{p,n}(\mathbb{1})$ may be expressed as a chi-squared distance between appropriately-defined distributions. We detail this result below.

Proposition 3.10. *Let $M_0 = \pi_0$, M_p be a Markov kernel admitting π_p as an invariant distribution (for $p \geq 1$), and each G_p take the form* (2.2)*. Then for $p \in \{0, \dots, n\}$,*

$$v_{p,n}(\mathbb{1}) = D_{\chi^2}(\pi_n M_{n,p} \,\|\, \pi_p).$$

Proof. By Proposition 3.5, we have

$$\begin{aligned} v_{p,n}(\mathbb{1}) &= \int_{\mathsf{X}} \left(\frac{\mathrm{d}\pi_n(\mathbb{1} M_{n,p})}{\mathrm{d}\pi_p}(x_p) \right)^2 \pi_p(\mathrm{d}x_p) - \pi_n(\mathbb{1})^2 \\ &= \int_{\mathsf{X}} \left(\frac{\mathrm{d}\pi_n M_{n,p}}{\mathrm{d}\pi_p}(x_p) \right)^2 \pi_p(\mathrm{d}x_p) - 1. \end{aligned}$$

Since $\pi_n M_{n,p}$ is a probability measure this corresponds to Definition 3.1 of the chi-squared distance of $\pi_n M_{n,p}$ from π_p, written in the form (3.1). ∎

As has previously been discussed, the variance of the incremental weights (2.2) is dependent on the 'similarity' between successive distributions in the sequence $(\pi_p)_{p=0}^n$. The above result allows us to formalise this notion of similarity, and its effect on the variances of estimators such as $\gamma_n^N(\mathbb{1})$; it also further motivates the use of $\sigma^2_{\mathbb{1}}$ as a quantity for assessing and comparing the performances of distribution schedules. However, this quantity is not *directly* dependent on the similarity between the distributions $(\pi_p)_{p=0}^n$, since it also depends on the time reversals of the Markov kernels $(M_p)_{p=1}^n$. In particular, the mixing properties of these time reversal kernels strongly influence the terms $v_{p,n}(\mathbb{1})$.

Consider the measure $\pi_n M_{n,p}$ that appears in the result of Proposition 3.10. One may find that when the time reversal kernels exhibit good mixing, $\pi_n M_{n,p}$ resembles π_{p+1} (for $p < n$), and so the corresponding term $v_{p,n}(\mathbb{1})$ approximates the chi-squared distance of π_{p+1} from π_p. This may indeed be seen as the 'similarity' between consecutive distributions; this is further explored in Chapter 4, which considers the extreme setting in which each Markov kernel M_p (and its time reversal) exhibits *perfect* mixing. When in contrast the time reversal kernels mix poorly, $\pi_n M_{n,p}$ is generally comparable to π_n, and so the $v_{p,n}(\mathbb{1})$ terms are larger.

In Section 5.1 we shall explore a collection of simply-defined Markov kernels in which the mixing quality is determined by a sequence of parameters. The description therein of

the resulting forms of $\pi_n M_{n,p}$ exemplifies the properties discussed above, and may assist in building an intuition for the effects of the Markov kernels' mixing properties on the terms $v_{p,n}(\mathbb{1})$.

3.4.2.1. Occasional resampling

We now turn our attention to the use of occasional resampling, and the resulting SMC estimator of the normalising constant of γ_n. In this case, $\gamma_n^N(\mathbb{1})$ takes the general form

$$\gamma_n^N(\mathbb{1}) = \left(\prod_{j=0}^{r_n-1} \frac{1}{N} \sum_{i=1}^{N} \left[\prod_{p=k_j}^{k_{j+1}-1} G_p(\zeta_p^i) \right] \right) \left(\frac{1}{N} \sum_{i=1}^{N} \left[\prod_{p=k_{r_n}}^{n-1} G_p(\zeta_p^i) \right] \right). \tag{3.32}$$

Each use of resampling may be seen to contribute an additional factor to this estimator, which is a product of $r_n + 1$ factors. The apparent contradiction with the expression (3.29) resulting from always resampling, which is a product of n factors (rather than $n + 1$), may be explained by the following remark.

Remark 3.11. Given a sequence of resampling times $k_{1:r_{n-1}}$ occurring in the first $n - 1$ iterations of Algorithm 1.3, whether resampling is conducted in the nth iteration of the algorithm has no effect on the normalising constant estimator $\gamma_n^N(\mathbb{1})$. This follows from inspection of (3.32) in the case that resampling does not occur in the nth iteration (so that $r_n = r_{n-1}$), and in the case that such resampling does occur (so that $r_n = r_{n-1} + 1$, with $k_{r_n} = n$). In the latter case, the additional factor introduced to (3.32) is the final (rightmost) factor, which evaluates to 1.

Consider the relative asymptotic variance of (3.32), as $N \longrightarrow \infty$. As discussed in Section 1.4.1.2, this admits a decomposition of the form $\sum_{j=0}^{r_n} \hat{v}'_{j,r_n}(\mathbb{1})$. Taking $\varphi = \mathbb{1}$ in (1.38), each term in this decomposition may be expressed as

$$\hat{v}'_{j,r_n}(\mathbb{1}) = \frac{\eta'_j(Q'_{j,r_n}(G'_{r_n})^2)}{\eta'_j(Q'_{j,r_n}(G'_{r_n}))^2} - 1. \tag{3.33}$$

Considering this expression for $j = r_n$, note that $\hat{v}'_{r_n,r_n}(\mathbb{1})$ is typically non-zero, in contrast to Remark 3.9. In general $\hat{v}'_{r_n,r_n}(\mathbb{1}) = 0$ only in the special case that $k_{r_n} = n$, so that $G'_{r_n} = \mathbb{1}$.

Remark 3.12. In the decomposition of the relative asymptotic variance of (3.32), an additional summand (3.33) is introduced by each instance of resampling. By consideration of (3.33) it may be shown that resampling in the nth iteration leaves the first r_{n-1} summands unchanged. As discussed in Remark 3.11, resampling in the nth iteration has no effect on the estimator $\gamma_n^N(\mathbb{1})$; therefore this must have no effect on its relative asymptotic variance, and so the additional summand this introduces must be zero. This may be seen to explain the above observation that $\hat{v}'_{r_n,r_n}(\mathbb{1}) = 0$ when $k_{r_n} = n$, of which the result in Remark 3.9 is a special case.

Again, we now focus on (3.32) as formed by an SMC sampler, in which case this corresponds to (3.8). The relative asymptotic variance of this estimator is $\sigma^2_{\mathbb{1}}$ as defined in (3.9), and so we have

$$\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n) = \sum_{j=0}^{r_n} \hat{v}'_{j,r_n}(\mathbb{1}) \tag{3.34}$$

where $R_n := \{k_j : j \in \{1, \dots, r_n\}\}$. Analogously to Proposition 3.10, each term in this expression may be expressed as a chi-squared distance:

Proposition 3.13. *Let $M_0 = \pi_0$, M_p be a Markov kernel admitting π_p as an invariant distribution (for $p \geq 1$), and each G_p take the form* (2.2)*. Then for $j \in \{0, \dots, r_n\}$,*

$$\hat{v}'_{j,r_n}(\mathbb{1}) = D_{\chi^2}(\hat{\eta}'_{r_n} M'_{r_n,j} \,\|\, \eta'_j). \tag{3.35}$$

The proof of this result is essentially the same as that of Proposition 3.10, being obtained by taking $\varphi = \mathbb{1}$ in Proposition 3.8 and comparing the resulting expression with Definition 3.1 of the chi-squared distance.

Remark 3.14. In the special case that $k_{r_n} = n$, corresponding to resampling occurring in the nth step, we have from Section 1.4.1.2 that $\hat{v}'_{j,r_n}(\mathbb{1}) = v'_{j,r_n}(\mathbb{1})$, as defined in (1.39). As also discussed therein, we also have $\hat{\eta}'_{r_n} = \eta'_{r_n}$; this is in turn equal to η_n, which for an SMC sampler corresponds to the final distribution π_n. Therefore for $j \in \{0, \dots, r_n\}$, (3.35) simplifies in this case to

$$v'_{j,r_n}(\mathbb{1}) = D_{\chi^2}(\pi_n M'_{r_n,j} \,\|\, \eta'_j).$$

Recall from Remark 3.12 that resampling in the nth iteration has no effect on $\sigma^2_{\mathbb{1}}$; it is therefore always possible to decompose $\sigma^2_{\mathbb{1}}$ as a sum of these simpler expressions, by assuming that resampling in the nth iteration does indeed take place.

3.5. Summary

In this chapter we have introduced the schedule selection problem for SMC samplers, proposing a formulation of this problem in terms of the relative asymptotic variance $\sigma^2_{\mathbb{1}}$ of the normalising constant estimator $\gamma^N_n(\mathbb{1})$. In the following chapters we shall continue to investigate the quantity $n\sigma^2_{\mathbb{1}}$ introduced in Section 3.3, making frequent use of the expressions obtained in Section 3.4.2.

The general results in Section 3.4 pertain to the relative asymptotic variance of $\gamma^N_n(\varphi)$, for arbitrary bounded $\mathcal{X}$-measurable test functions φ. As discussed in Section 3.3, the normalising constant estimator $\gamma^N_n(\mathbb{1})$ is a natural quantity of interest, and for this reason we shall focus on the setting $\varphi = \mathbb{1}$ in subsequent chapters. However, our proposed approach to solving the schedule selection problem could in theory be generalised, optimising the asymptotic variance of some other estimator; while we do not pursue this here, the more general results may be useful for this purpose.

4. Optimal schedules for perfectly-mixing Markov kernels

4.1. The perfectly-mixing setting

A special case of the SMC sampler is that in which all the Markov kernels $(M_p)_{p=1}^n$ are perfectly mixing, so that in the pth iteration of the algorithm one may generate IID samples exactly distributed according to π_p. Formally, for $p \in \{1, \dots, n\}$ this may be expressed as

$$M_p(x, \cdot) = \pi_p(\cdot) \qquad \text{for all } x \in \mathsf{X}.$$

Within this chapter it is this specific case that shall be studied.

Although the ability to draw IID samples would generally preclude the need to use an SMC sampler, such a setting might be useful for estimating the normalising constants of the corresponding densities, when these are computationally intractable. Furthermore, this special case is convenient for analysis, since it allows closed-form expressions for the relative asymptotic variance $\sigma^2_{\mathbb{1}}$ to be derived; other authors (e.g. Cérou et al., 2012) analyse this case for the same reason. Following Section 3.3 we may therefore derive properties of the optimal schedules of distributions $(\pi_p)_{p=0}^n$, by determining those that minimise $n\sigma^2_{\mathbb{1}}$. Results in this setting may provide a useful baseline for the behaviour of the relative asymptotic variance $\sigma^2_{\mathbb{1}}$, and of the optimal schedule, in more general settings.

The simple form of these Markov kernels allows further simplification of the expression for $v_{p,n}(\mathbb{1})$ given in Proposition 3.10, giving the following result. This gives a useful characterisation of the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$: specifically, when resampling is used in every iteration, this is equal to the sum of the chi-squared distances between consecutive distributions in the sequence $(\pi_p)_{p=0}^n$.

Proposition 4.1. *For $p \in \{1, \dots, n\}$ let $M_p(x, \cdot) = \pi_p(\cdot)$ for all $x \in \mathsf{X}$. Then for an SMC sampler using resampling in every iteration, the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$ as $N \longrightarrow \infty$ is given by*

$$\sigma^2_{\mathbb{1}}(\pi_{0:n}, \{1, \dots, n\}) = \sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p).$$

Proof. By (3.31), $\sigma^2_{\mathbb{1}}(\pi_{0:n}, \{1, \dots, n\}) = \sum_{p=0}^n v_{p,n}(\mathbb{1})$. By Remark 3.9, the summand for which $p = n$ is equal to 0. Consider therefore $v_{p,n}(\mathbb{1})$ for $p < n$, which by Proposition 3.10 is equal to $D_{\chi^2}(\pi_n M_{n,p} \,\|\, \pi_p)$.

Since $M_p(x, \cdot) = \pi_p(\cdot)$ for all $x \in \mathsf{X}$ it follows directly from Definition 3.2 that $M_p^*(x, \cdot) = \pi_p(\cdot)$ for all $x \in \mathsf{X}$, for all $p \in \{1, \dots, n\}$. Therefore for $p \in \{0, \dots, n-1\}$, for $A \in \mathcal{X}$,

$$\begin{aligned}\pi_n M_{n,p}(A) &= \int_{\mathsf{X}} \pi_n(\mathrm{d}x_n) \int_{\mathsf{X}} M_n^*(x_n, \mathrm{d}x_{n-1}) \cdots \int_{\mathsf{X}} M_{p+2}^*(x_{p+2}, \mathrm{d}x_{p+1}) M_{p+1}^*(x_{p+1}, A) \\ &= \int_{\mathsf{X}} \pi_n(\mathrm{d}x_n) \int_{\mathsf{X}} \pi_n(\mathrm{d}x_{n-1}) \cdots \int_{\mathsf{X}} \pi_{p+2}(\mathrm{d}x_{p+1}) \pi_{p+1}(A) \\ &= \pi_{p+1}(A),\end{aligned}$$

and so $\pi_n M_{n,p} = \pi_{p+1}$. The stated result follows. ∎

This result helps to formalise the notion that the variances of estimators are reduced by ensuring that successive distributions are sufficiently 'similar': in the perfectly-mixing setting, the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$ will be large if consecutive distributions are dissimilar in the sense of the chi-squared distance.

As discussed, this result only holds if resampling is used in every iteration of the SMC sampler. In the more general setting of occasional resampling, a similar expression for the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$ may be obtained in terms of the chi-squared distances $D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$. We present this result below, which may be seen to follow from Proposition 3.13.

Proposition 4.2. *For $p \in \{1, \dots, n\}$ let $M_p(x, \cdot) = \pi_p(\cdot)$ for all $x \in \mathsf{X}$. Then for an SMC sampler using resampling schedule $R_n := \{k_j : j \in \{1, \dots, r_n\}\}$, the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$ as $N \to \infty$ is given by*

$$\sigma_{\mathbb{1}}^2(\pi_{0:n}, R_n) = \sum_{j=0}^{r_n - 1} \left[\left(\prod_{p=k_j}^{k_{j+1}-1} \left[D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p) + 1 \right] \right) - 1 \right].$$

Proof. As discussed in Remark 3.11, whether resampling takes place in the nth iteration of the SMC sampler has no effect on the form of $\gamma_n^N(\mathbb{1})$, and therefore no effect on its asymptotic variance. Without loss of generality we may therefore assume that resampling does indeed take place in this final step, so that $k_{r_n} = n$. In this case the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$ as $N \to \infty$ may be expressed as $\sum_{j=0}^{r_n} v'_{j,r_n}(\mathbb{1})$, where $v'_{j,r_n}(\mathbb{1}) = D_{\chi^2}(\pi_n M'_{r_n,j} \,\|\, \eta'_j)$ following Remark 3.14.

First consider this expression for $j = r_n$. Since $M'_{r_n,r_n} := \mathrm{Id}$ we have $\pi_n M'_{r_n,r_n} = \pi_n$; also, $\eta'_{r_n} = \eta_n$, which for an SMC sampler is equal to the final distribution π_n. It therefore follows that $v'_{r_n,r_n}(\mathbb{1}) = D_{\chi^2}(\pi_n \,\|\, \pi_n) = 0$, in line with Remark 3.12.

Consider therefore $v'_{j,r_n}(\mathbb{1})$ for $j < r_n$. The excursion Feynman–Kac model η'_j is of the form (3.19); in this perfectly-mixing setting, this simplifies to

$$\eta'_j(\mathrm{d}x'_j) = \prod_{p=k_j}^{k_{j+1}-1} \pi_p(\mathrm{d}x_p). \tag{4.1}$$

We look to obtain a similar simplified expression for the reversal kernels $(M'_j)^*$ defined in (3.20). As described in the proof of Proposition 4.1, since $M_p(x, \cdot) = \pi_p(\cdot)$ for all $x \in \mathsf{X}$ it follows directly from Definition 3.2 that $M^*_p(x, \cdot) = \pi_p(\cdot)$ for all $x \in \mathsf{X}$ for $p \in \{1, \dots, n\}$. Therefore for $j \in \{1, \dots, r_n\}$, one has for all $x'_j \in \mathsf{X}'_j$ that

$$(M'_j)^*(x'_j, \mathrm{d}x'_{j-1}) := \prod_{p=k_j}^{k_{j-1}+1} M^*_p(x_p, \mathrm{d}x_{p-1}) = \prod_{p=k_j}^{k_{j-1}+1} \pi_p(\mathrm{d}x_{p-1}) = \prod_{p=k_j-1}^{k_{j-1}} \pi_{p+1}(\mathrm{d}x_p).$$

We now proceed similarly to Proposition 4.1. Using the above expression for $(M'_j)^*$, and noting that $\mathsf{X}'_{r_n} = \mathsf{X}$ since $k_{r_n} = n$, we have for $j \in \{0, \dots, r_n - 1\}$ that

$$\begin{aligned}
\pi_n M'_{r_n:j}(\mathrm{d}x'_j) &= \int_{\mathsf{X}'_{r_n}} \pi_n(\mathrm{d}x'_{r_n}) \left[\prod_{\ell=r_n}^{j+2} \int_{\mathsf{X}'_\ell} (M'_\ell)^*(x'_\ell, \mathrm{d}x'_{\ell-1}) \right] (M'_{j+1})^*(x'_{j+1}, \mathrm{d}x'_j) \\
&= \int_{\mathsf{X}} \pi_n(\mathrm{d}x_n) \left[\prod_{\ell=r_n}^{j+2} \prod_{p=k_\ell-1}^{k_{\ell-1}} \int_{\mathsf{X}} \pi_{p+1}(\mathrm{d}x_p) \right] \prod_{p=k_{j+1}-1}^{k_j} \pi_{p+1}(\mathrm{d}x_p) \\
&= \prod_{p=k_j}^{k_{j+1}-1} \pi_{p+1}(\mathrm{d}x_p). \qquad (4.2)
\end{aligned}$$

Using the form (3.1) of the chi-squared distance, we may express $v'_{j,r_n}(\mathbb{1})$ as an integral:

$$v'_{j,r_n}(\mathbb{1}) = D_{\chi^2}(\pi_n M'_{r_n:j} \,\|\, \eta'_j) = \int_{\mathsf{X}'_j} \left(\frac{\mathrm{d}\pi_n M'_{r_n:j}}{\mathrm{d}\eta'_j}(x'_j) \right)^2 \eta'_j(\mathrm{d}x'_j) - 1.$$

Substituting (4.1) and (4.2), this may be written as

$$v'_{j,r_n}(\mathbb{1}) = \int_{\mathsf{X}^{k_{j+1}-k_j}} \left(\prod_{p=k_j}^{k_{j+1}-1} \frac{\mathrm{d}\pi_{p+1}}{\mathrm{d}\pi_p}(x_p) \right)^2 \prod_{p=k_j}^{k_{j+1}-1} \pi_p(\mathrm{d}x_p) - 1,$$

where the decomposition of the Radon–Nikodym derivative $\mathrm{d}\pi_n M'_{r_n:j} / \mathrm{d}\eta'_j$ into a product of such derivatives follows from its definition. Therefore,

$$\begin{aligned}
v'_{j,r_n}(\mathbb{1}) &= \prod_{p=k_j}^{k_{j+1}-1} \int_{\mathsf{X}} \left(\frac{\mathrm{d}\pi_{p+1}}{\mathrm{d}\pi_p}(x_p) \right)^2 \pi_p(\mathrm{d}x_p) - 1 \\
&= \prod_{p=k_j}^{k_{j+1}-1} \left[D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p) + 1 \right] - 1,
\end{aligned}$$

again using the form (3.1) of the chi-squared distance. The stated result follows. ∎

The main utility of this result is in showing that, when perfectly-mixing Markov kernels are used, the relative asymptotic variance $\sigma^2_{\mathbb{1}}$ is minimised when resampling is used in every iteration of the SMC sampler. We detail this result below, which has the effect of solving the problem of selecting a resampling schedule in perfectly-mixing settings.

Proposition 4.3. *For $p \in \{1, \dots, n\}$ let $M_p(x, \cdot) = \pi_p(\cdot)$ for all $x \in \mathsf{X}$. Then $\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n)$ is minimised when $\{1, \dots, n-1\} \subseteq R_n$; that is, the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$ as $N \longrightarrow \infty$ is minimised when resampling is used in each of the first $n-1$ iterations of the SMC sampler.*

Proof. We begin by showing that $\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n) \geq \sigma^2_{\mathbb{1}}(\pi_{0:n}, \{1, \dots, n\})$ for any resampling schedule $R_n \subseteq \{1, \dots, n\}$. Denoting $d_p \coloneqq D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ for $p \in \{0, \dots, n-1\}$, by Proposition 4.2 we have

$$\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n) = \sum_{j=0}^{r_n - 1} \left[\left(\prod_{p=k_j}^{k_{j+1}-1} \left[d_p + 1 \right] \right) - 1 \right],$$

where $R_n \coloneqq \{k_j : j \in \{1, \dots, r_n\}\}$.

By Definition 3.1 of the chi-squared distance, $d_p \geq 0$ for all $p \in \{0, \dots, n-1\}$. Since $\prod_{i=1}^m [1 + x_i] \geq 1 + \sum_{i=1}^m x_i$ for any non-negative real values $x_{1:m}$, we have

$$\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n) = \sum_{j=0}^{r_n - 1} \left[\left(\prod_{p=k_j}^{k_{j+1}-1} \left[d_p + 1 \right] \right) - 1 \right] \geq \sum_{j=0}^{r_n - 1} \left[\sum_{p=k_j}^{k_{j+1}-1} d_p \right] = \sum_{p=0}^{n-1} d_p = \sigma^2_{\mathbb{1}}(\pi_{0:n}, \{1, \dots, n\}),$$

where the final equality follows from Proposition 4.1. Therefore, the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$ as $N \longrightarrow \infty$ is minimised when resampling is used in every iteration of the SMC sampler.

Again from Remark 3.11, for any sequence of resampling times $k_{1:r_{n-1}}$ occurring in the first $n-1$ iterations, the use of resampling in the nth iteration has no effect on the form of $\gamma_n^N(\mathbb{1})$ and therefore no effect on its asymptotic variance. It follows that the value of $\sigma^2_{\mathbb{1}}$ resulting from resampling in all but the final iteration is equal to that resulting from resampling in every iteration, which by the above computations is the minimal value over all resampling schedules. ∎

This result has an intuitive interpretation: if one can draw IID samples from each intermediate distribution then it is always preferable for these to be unweighted, since the accumulation of importance weights will only increase the variance of resulting estimators. When it is not possible to draw IID samples in this way, this argument does not hold; as discussed in Section 1.3.1, in general some form of occasional resampling is preferred in order to minimise such an asymptotic variance.

It follows that in perfectly-mixing settings, that for any sequence of distributions, the optimal resampling schedule is that in which resampling is used in every iteration. For the remainder of this chapter, in which our objective is to minimise n times the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$, we therefore solely consider this setting. We proceed to determine properties of the optimal distribution schedules for some simple choices of the target distribution $\pi_\star$ and initial distribution π_0. In each case we consider two problems, which collectively correspond to the optimisation problem described in Section 3.3:

- For fixed n, π_0 and $\pi_n := \pi_\star$, which sequence $\pi_{1:n-1}$ minimises $\sigma^2_{\mathbb{1}}$?
- Considering for each n the value of $\sigma^2_{\mathbb{1}}$ attained by this optimal distribution schedule, which value of n minimises $n\sigma^2_{\mathbb{1}}$?

Following Proposition 4.1, the first of these two problems corresponds to determining the sequence $(\pi_p)_{p=0}^n$ that, for fixed π_0 and π_n, minimises the sum of chi-squared distances between consecutive distributions. We shall make extensive use of this formulation throughout this chapter.

The assumption of perfect mixing may limit the practical applications of the theoretical results we derive, since in many realistic settings the Markov kernels may mix poorly. To this end, the ideas and investigations we shall later present in Chapter 5 may provide a more practical contribution to the problem of tuning SMC samplers. Nonetheless, the results in this chapter may be useful for such tuning in well-mixing settings, and we shall propose a number of heuristic procedures based on the theoretical results we derive.

4.1.1. Preliminary results

Before proceeding, we introduce some simple results based on Jensen's inequality that will be useful in proving later results. Specifically, given a function defined on the positive real numbers, we consider the problem of minimising the sum of n evaluations of this function, under the constraint that the product of the n arguments is fixed. By the following lemma, under a convexity condition this is achieved when all n arguments are equal.

Lemma 4.4. *For some function $f : \mathbb{R}_+ \longrightarrow \mathbb{R}$ consider the problem of minimising $\sum_{p=1}^n f(x_p)$, under the constraint that $\prod_{p=1}^n x_p = \kappa > 0$. If $f(\exp(x))$ is convex as a function of $x \in \mathbb{R}$, then the unique minimal value is $nf(\kappa^{1/n})$, attained when $x_p = \kappa^{1/n}$ for all $p \in \{1, \dots, n\}$.*

Proof. By Jensen's inequality, applied to the convex function $x \mapsto f(\exp(x))$ and evaluated at the values $\log(x_1), \dots, \log(x_n)$,

$$\frac{1}{n}\sum_{p=1}^{n} f\big(\exp\big(\log\big(x_p\big)\big)\big) \;\geq\; f\left(\exp\left(\frac{1}{n}\sum_{p=1}^{n}\log\big(x_p\big)\right)\right).$$

Equality holds if and only if all values of $\log(x_p)$ are equal, which occurs if and only if all x_i are equal, by the injectivity of the logarithm. Rearranging,

$$\sum_{p=1}^{n} f(x_p) \;\geq\; n \cdot f\left(\left[\prod_{p=1}^{n} x_p\right]^{1/n}\right).$$

Since $\prod_{p=1}^n x_p = \kappa$, it follows that $\sum_{p=1}^n f(x_p)$ is bounded below by $nf(\kappa^{1/n})$, and this value is attained if and only if $x_1 = \cdots = x_n = \kappa^{1/n}$. ∎

This may be seen as a generalisation of the inequality of arithmetic and geometric means, which corresponds to the case in which f is the identity function.

We also present a multivariate form of this result. In this case, we consider minimising the sum of n evaluations of a function defined on $\mathbb{R}_+^d$, under the constraint that the componentwise product of the n arguments is fixed.

Lemma 4.5. *For some function $f : \mathbb{R}_+^d \longrightarrow \mathbb{R}$ consider the problem of minimising*

$$\sum_{p=1}^{n} f(x_{p,1}, \dots, x_{p,d}),$$

under the constraints that $\prod_{p=1}^{n} x_{p,i} = \kappa_i > 0$ for $i \in \{1, \dots, d\}$. If $f(\exp(x_1), \dots, \exp(x_d))$ is convex as a function of $(x_1, \dots, x_d) \in \mathbb{R}_+^d$, then the unique minimal value is

$$nf(\kappa_1^{1/n}, \dots, \kappa_d^{1/n}),$$

attained when $x_{p,i} = \kappa_i^{1/n}$ for all $p \in \{1, \dots, n\}$, $i \in \{1, \dots, d\}$.

The proof is analogous to that of the univariate result.

4.2. Restrictions on nested sets

A particular setting of interest is that in which the sequence of distributions $(\pi_p)_{p=0}^n$ comprises restrictions of some probability measure π on each of a sequence of nested sets (normalised appropriately, so that each π_p is a probability measure). We shall assume that π admits a (possibly unnormalised) density $\bar{\pi}$ with respect to some dominating measure $\mathrm{d}x$. That is, for $A \in \mathcal{X}$

$$\pi(A) = \frac{1}{Z} \int_A \bar{\pi}(x)\, \mathrm{d}x,$$

where the normalising constant $Z := \int_{\mathsf{X}} \bar{\pi}(x)\, \mathrm{d}x$ may be unknown.

For some $E_0, E_\star \in \mathcal{X}$ such that $E_\star$ is a proper subset of E_0, we assume that the initial distribution π_0 and final distribution $\pi_\star$ admit unnormalised densities that may respectively be expressed as

$$\bar{\pi}_0 = \bar{\pi} \cdot \mathbb{1}_{E_0}, \qquad \bar{\pi}_\star = \bar{\pi} \cdot \mathbb{1}_{E_\star}.$$

Expressing the probability measures π_0 and $\pi_\star$ in terms of π, we have that for $A \in \mathcal{X}$,

$$\pi_0(A) = \frac{\pi(A \cap E_0)}{\pi(E_0)}, \qquad \pi_\star(A) = \frac{\pi(A \cap E_\star)}{\pi(E_\star)}.$$

Considering the normalising constants of $\bar{\pi}_0$ and $\bar{\pi}_\star$, one has that

$$Z_0 = \int_{\mathsf{X}} \bar{\pi}_0(x)\, \mathrm{d}x = Z \cdot \pi(E_0), \qquad Z_\star = \int_{\mathsf{X}} \bar{\pi}_\star(x)\, \mathrm{d}x = Z \cdot \pi(E_\star).$$

In the analysis that follows, it will be convenient to define $\kappa := Z_0/Z_\star \geq 1$; indeed we shall assume that $\kappa > 1$, so that $\pi(E_\star)$ is strictly less than $\pi(E_0)$.

To determine a sequence of distributions $(\pi_p)_{p=0}^n$, we define a decreasing sequence in $\mathcal{X}$ given by

$$E_0 \supseteq E_1 \supseteq \cdots \supseteq E_n := E_\star.$$

For $p \in \{0, \ldots, n\}$, let π_p be the normalised restriction of π on E_p; that is, for $A \in \mathcal{X}$

$$\pi_p(A) = \frac{\pi(A \cap E_p)}{\pi(E_p)}. \tag{4.3}$$

This admits an unnormalised probability density function given by $\bar{\pi}_p = \bar{\pi} \cdot \mathbb{1}_{E_p}$, with normalising constant $Z_p = \int_{\mathsf{X}} \bar{\pi}_p(x)\,\mathrm{d}x = Z \cdot \pi(E_p)$. An illustration of such a sequence of unnormalised density functions is presented in Figure 4.1.

Such sequences arise in a number of applications of the SMC sampler framework. As described in Chapter 2, for an SMC sampler one has $\gamma_n(\mathbb{1}) = Z_\star/Z_0$. Here this ratio is $Z_\star/Z_0 = \pi(E_\star)/\pi(E_0)$; that is, the conditional probability under π of $E_\star$ given E_0. If one chooses $E_0 = \mathsf{X}$ so that $\pi_0 = \pi$, then this corresponds to the probability of the event $E_\star$ under π. Using an SMC sampler one may therefore obtain an unbiased estimator $\gamma_n^N(\mathbb{1})$ of this probability of interest, κ^{-1}.

The benefit of this SMC approach is greatest when κ^{-1} is very small, in which case direct estimation of this quantity using simple Monte Carlo techniques may require very large numbers of samples in order to obtain a non-zero estimate. In contrast, use of an SMC sampler effectively reduces the problem to that of estimating the conditional probability of E_p given E_{p-1} for $p \in \{1, \ldots, n\}$, corresponding to the n iterations of the algorithm. The potential functions (2.2), used to compute each particle's incremental weight, have the convenient form

$$G_p = \mathbb{1}_{E_{p+1}}.$$

For these reasons, SMC samplers have been widely employed for rare event estimation as described in Section 2.3.2, by taking $E_\star$ as the rare event for which we wish to estimate the probability, and $E_0 = \mathsf{X}$. Comparable constructions have also been used in rare event estimation algorithms based on multilevel splitting, and have been well studied in this context (see e.g. Lagnoux, 2006).

Similarly, this methodology may be useful for estimating the volumes of convex sets in high-dimensional space, being complementary (although somewhat different in construction) to MCMC methods addressing this problem. These methods are of practical interest because, while the exact evaluation of such volumes is a #P-hard problem, Monte Carlo methods allow approximations of arbitrarily small error to be computed in polynomial time. Jerrum and Sinclair (1996, Section 12.5.2) provide a review of such methods; a recent MCMC approach to this problem has been proposed by Chevallier et al. (2018).

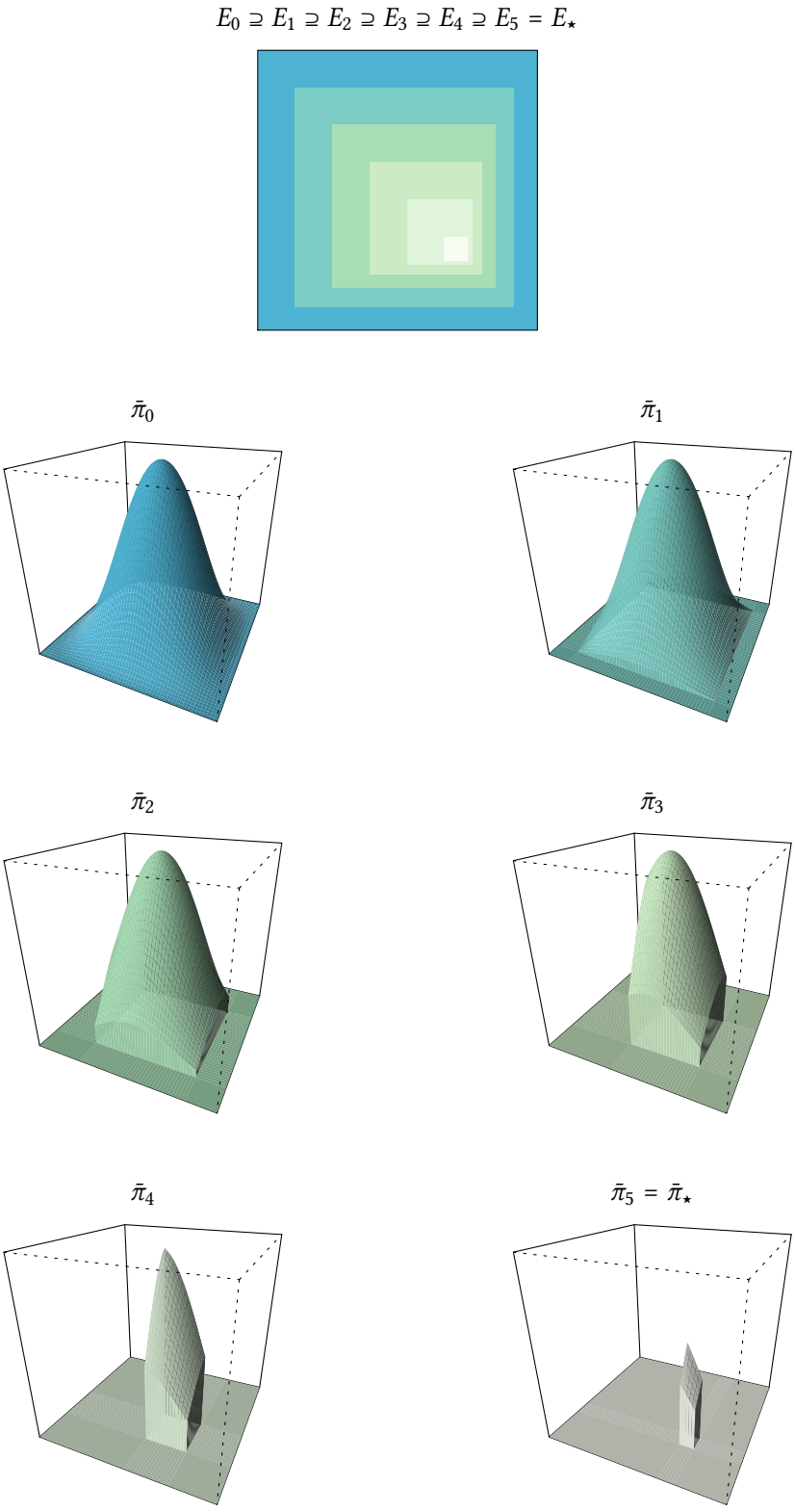


Figure 4.1.: Restrictions of a bivariate density function on a sequence of nested sets.

In other applications the quantity to be approximated is $\pi_\star$ itself; for example in SMC implementations of approximate Bayesian computation (ABC), as introduced in Section 2.3.1. In Del Moral et al. (2012a), the decreasing sequence of tolerance levels corresponds to a decreasing sequence of subsets of the joint space of parameters and pseudo-observations. The resulting distributions $(\pi_p)_{p=0}^n$ may be seen as restrictions of the overall joint distribution on this space to each of these subsets, with the marginals of these distributions forming a sequence of improving approximations of the true posterior.

4.2.1. Optimal distribution schedule for fixed n

As emphasised, in this setting the selection of a distribution schedule $(\pi_p)_{p=0}^n$ is equivalent to selecting the sequence of sets $(E_p)_{p=0}^n$, given E_0 and $E_n := E_\star$. In the results that follow we derive conditions for this sequence to be optimal, by considering the two problems described at the end of Section 4.1. Firstly, for fixed n we find conditions under which a sequence $(E_p)_{p=0}^n$ minimises the relative asymptotic variance $\sigma_{\mathbb{1}}^2$; we shall then find the value of n for which the corresponding value of $n\sigma_{\mathbb{1}}^2$ is minimised.

In the case of perfectly-mixing Markov kernels $(M_p)_{p=1}^n$ that we consider here, the former problem of choosing an optimal sequence $(E_p)_{p=0}^n$ for fixed n has already been considered by Cérou et al. (2012, Section 2.3). Proposition 4.6 provides a full proof of this known result, here presented explicitly in terms of the chi-squared distances between consecutive distributions.

Proposition 4.6. *For a fixed value of n, consider a distribution schedule $(\pi_p)_{p=0}^n$ defined by (4.3), for some probability measure π and decreasing sequence of sets $(E_p)_{p=0}^n$. For fixed E_0 and E_n, the sum $\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ of chi-squared distances between consecutive distributions is minimised when the probabilities of $(E_p)_{p=0}^n$ under π form a geometric progression, in which case all of these chi-squared distances are equal. Specifically, its minimal value is $n(\kappa^{1/n} - 1)$, attained when $\pi(E_p) = \kappa^{-1/n}\pi(E_{p-1})$ for all $p \in \{1, \ldots, n\}$.*

Proof. For $p \in \{0, \ldots, n-1\}$, the chi-squared distance of π_{p+1} from π_p is given by

$$\begin{aligned} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p) &= \int_{\mathsf{X}} \left(\frac{\mathrm{d}\pi_{p+1}}{\mathrm{d}\pi_p}(x) \right)^2 \pi_p(\mathrm{d}x) - 1 \\ &= \int_{\mathsf{X}} \left(\frac{\bar{\pi}(x)\mathbb{1}_{E_{p+1}}(x)/Z_{p+1}}{\bar{\pi}(x)\mathbb{1}_{E_p}(x)/Z_p} \right)^2 \left(\bar{\pi}(x)\mathbb{1}_{E_p}(x)/Z_p \right) \mathrm{d}x - 1 \\ &= \frac{Z_p}{Z_{p+1}^2} \int_{E_{p+1}} \bar{\pi}(x)\,\mathrm{d}x - 1 \\ &= \frac{Z_p}{Z_{p+1}} - 1. \end{aligned}$$

We therefore look to minimise

$$\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p) = \sum_{p=0}^{n-1} \left(\frac{Z_p}{Z_{p+1}} - 1 \right) = \sum_{p=1}^{n} \left(\xi_p - 1 \right),$$

where for $p \in \{1, \dots, n\}$,

$$\xi_p := \frac{Z_{p-1}}{Z_p} = \frac{Z \cdot \pi(E_{p-1})}{Z \cdot \pi(E_p)} = \frac{\pi(E_{p-1})}{\pi(E_p)}.$$

Since $E_{p-1} \supseteq E_p$, we have $\xi_p \geq 1$ for each p. An additional constraint is obtained by noting that

$$\prod_{p=1}^{n} \xi_p = \frac{Z_0}{Z_\star} = \kappa.$$

In summary, one looks to choose n numbers $(\xi_p)_{p=1}^n$ no less than 1, with fixed product κ, in order to minimise $\sum_{p=1}^n f(\xi_p)$ where $f(x) := x - 1$.

The function $x \mapsto f(\exp(x)) = \exp(x) - 1$ is convex, and so the optimal choice may be found using Lemma 4.4. By this result the unique minimal value of $\sum_{p=1}^n (\xi_p - 1)$ is $n(\kappa^{1/n} - 1)$, obtained when $\xi_1 = \xi_2 = \cdots = \xi_n = \kappa^{1/n}$. While Lemma 4.4 only requires each ξ_p to be positive, this solution satisfies the additional constraint that $\xi_p \geq 1$ for each p. The sum of chi-squared distances therefore takes its minimal value of $n(\kappa^{1/n} - 1)$ when all of the chi-squared distances are equal, occurring when $\pi(E_{p-1})/\pi(E_p) = \kappa^{1/n}$ for all $p \in \{1, \dots, n\}$. ∎

Following Proposition 4.1 this result implies that for fixed n, the relative asymptotic variance $\sigma_{\mathbb{1}}^2$ is minimised in this perfectly-mixing setting by choosing $(\pi_p)_{p=0}^n$ to be equally spaced in the sense of the chi-squared distance. This may be achieved by choosing any sequence of sets $(E_p)_{p=0}^n$ for which the probabilities under π form a geometric progression.

Remark 4.7. From Proposition 4.6, the minimal value of $\sigma_{\mathbb{1}}^2$ for a schedule of fixed length n is $n(\kappa^{1/n} - 1)$. We see that this quantity converges to $\log(\kappa)$ as $n \longrightarrow \infty$. It follows that when constructing distribution schedules in the way considered here (i.e. using restrictions of π on each of a sequence of nested sets), it is not possible to make the relative asymptotic variance $\sigma_{\mathbb{1}}^2$ arbitrarily small.

4.2.2. Optimal schedule length n

We now turn to the problem of choosing n in order to minimise $n\sigma_{\mathbb{1}}^2$. By Proposition 4.6, the minimal value of $\sigma_{\mathbb{1}}^2$ over all distribution schedules with fixed n is given by $n(\kappa^{1/n} - 1)$; it follows that one should choose n as the value $n^\star$ that minimises $n \times n(\kappa^{1/n} - 1) = n^2(\kappa^{1/n} - 1)$. Finding this integer value is facilitated by considering this as a function of $n \in \mathbb{R}_+$, leading to the following result.

Proposition 4.8. *The function $f : \mathbb{R}_+ \longrightarrow \mathbb{R}$ given by $f(x) := x^2(\kappa^{1/x} - 1)$, where $\kappa > 1$ is a constant, is convex and has exactly one local minimum, which is therefore also the global minimum. This minimum is attained when*

$$x = \frac{\log(\kappa)}{W_0(-2e^{-2}) + 2} \approx 0.6275 \log(\kappa),$$

where W_0 denotes the principal branch of the Lambert W function.

Proof. Let us first show that f is a convex function. By elementary calculus,

$$\begin{aligned} f(x) &= x^2 \left(\exp\left(\frac{\log(\kappa)}{x} \right) - 1 \right), \\ f'(x) &= (2x - \log(\kappa)) \exp\left(\frac{\log(\kappa)}{x} \right) - 2x, \\ f''(x) &= \left(2 - \frac{2\log(\kappa)}{x} + \frac{\log(\kappa)^2}{x^2} \right) \exp\left(\frac{\log(\kappa)}{x} \right) - 2. \end{aligned} \tag{4.4}$$

To prove the convexity of f, we show that this second derivative is positive for all possible values of x and κ. To this end, we consider the function $g : \mathbb{R}_+ \longrightarrow \mathbb{R}$ defined by

$$g(y) := \left(2 - 2y + y^2\right) \exp(y) - 2,$$

so that $f''(x) = g(\log(\kappa)/x)$. For fixed $\kappa > 1$ the mapping from $\mathbb{R}_+ \longrightarrow \mathbb{R}_+$ given by $x \mapsto \log(\kappa)/x$ is bijective, and therefore it is sufficient to show that $g(y) > 0$ for all $y \in \mathbb{R}_+$.

The first derivative of g is $g'(y) = y^2 \exp(y)$, which is positive for all $y \in \mathbb{R}_+$, and so g is increasing. Since $\lim_{y \to 0} g(y) = 0$, it follows that $g(y) > 0$ for all $y \in \mathbb{R}_+$. Therefore $f''(x) > 0$ for all $x \in \mathbb{R}_+$, so that f is convex. Since it is additionally the case that $\lim_{x \to 0} f(x) = \lim_{x \to \infty} f(x) = \infty$, it follows that f has exactly one local minimum in $\mathbb{R}_+$, which is therefore also the global minimum.

The value of x at which this minimum is attained is therefore the unique solution of $f'(x) = 0$. Consider a solution of the form $x = \alpha \log(\kappa)$ for some $\alpha > 0$ (since we require $x \in \mathbb{R}_+$, and $\log(\kappa) > 0$). Then from (4.4),

$$(2\alpha \log(\kappa) - \log(\kappa)) \exp\left(\frac{\log(\kappa)}{\alpha \log(\kappa)} \right) - 2\alpha \log(\kappa) = 0,$$

from which one obtains

$$\left(2 - \frac{1}{\alpha}\right) \exp\left(\frac{1}{\alpha}\right) - 2 = 0. \tag{4.5}$$

Denoting $\alpha' := 1/\alpha - 2$, this becomes

$$-\alpha' \exp(\alpha' + 2) - 2 = 0$$

which is rearranged to give

$$\alpha' \exp(\alpha') = -2\exp(-2). \tag{4.6}$$

Although this is ostensibly solved when $\alpha' = -2$, this does not correspond to a real value of $\alpha = 1/(\alpha' + 2)$ satisfying (4.5); indeed, since $\alpha \in \mathbb{R}_+$ one requires $\alpha' \in (-2, 0)$. Since f has a unique minimum, there is exactly one such value of α' satisfying (4.6).

The Lambert W function is the multivalued inverse of the function mapping $z \in \mathbb{C}$ to $z\exp(z)$. For $w \in (-\mathrm{e}^{-1}, 0)$ there are two real values of z satisfying $z\exp(z) = w$; the greater of these two values is $W_0(w)$, where W_0 denotes the principal branch of the Lambert W function, and satisfies $W_0(w) \in (-1, 0)$ (Corless et al., 1996).

Considering (4.6), since $-2\mathrm{e}^{-2} \in (-\mathrm{e}^{-1}, 0)$ it follows that $\alpha' = W_0(-2\mathrm{e}^{-2}) \approx -0.406$, so that

$$\alpha = \frac{1}{\alpha' + 2} = \frac{1}{W_0(-2\mathrm{e}^{-2}) + 2} \approx 0.6275.$$

The unique minimum of f is therefore obtained at $x = \alpha\log(\kappa)$, for this value of α. ∎

Since this function f is convex, it follows that the integer value $n^\star$ that minimises $n\sigma_{\mathbb{1}}^2$ is one of the two nearest integers to this value (either its floor or its ceiling).

This result may be useful for tuning adaptive approaches to selecting the sets $(E_p)_{p=0}^n$. Cérou et al. (2012) propose taking the particle population $(\zeta_p^i)_{i=1}^N$ at time p, and choosing E_{p+1} so that the proportion of these particles lying within this set is equal to some predetermined $\rho \in (0, 1)$. Since the particles $(\zeta_p^i)_{i=1}^N$ form an approximation of π_p, it follows that E_{p+1} is chosen so that $\pi(E_{p+1}) \approx \rho \cdot \pi(E_p)$. The choice of ρ therefore determines n.

Remark 4.9. By Proposition 4.6, to obtain a sequence $(\pi_p)_{p=0}^n$ of length n one should choose $\rho = \kappa^{-1/n}$. Using the optimal value $n^\star$ that follows from Proposition 4.8, it follows that in order to minimise $n\sigma_{\mathbb{1}}^2$ one should choose

$$\rho = \kappa^{-1/n^\star} \approx \kappa^{-1/(\alpha\log(\kappa))} = \mathrm{e}^{-1/\alpha}.$$

Using (4.5), this may be evaluated as

$$\mathrm{e}^{-1/\alpha} = \frac{1}{2}\left(2 - \frac{1}{\alpha}\right) = -\frac{1}{2}W_0(-2\mathrm{e}^{-2}) \approx 0.203.$$

This optimal choice of ρ does not depend on κ, and therefore requires no knowledge of the relative probabilities of E_0 and $E_\star$ under π.

While this result depends on the use of perfectly-mixing Markov kernels, it may provide a useful heuristic for tuning adaptive algorithms when well-mixing kernels may be constructed; for example, for choosing ρ in the algorithm of Cérou et al. (2012). In settings where the Markov kernels used mix poorly, it may be difficult to move the particles from the areas of positive mass of each π_{p-1} to those of the next distribution π_p by application

of the Markov kernel M_p. In general it may therefore be desirable for consecutive distributions to be more similar than would result from this choice of ρ, and so Remark 4.9 may be seen to provide a heuristic lower bound for choosing this tuning parameter.

4.2.3. Uniform distributions on nested balls

The effect of the dimension of the space X on these results is best exemplified in a special case. Suppose that $\mathsf{X} = \mathbb{R}^d$, with E_0 and $E_\star$ being open d-dimensional balls of radius r_0 and $r_\star$ respectively, both centred at some point $x_0 \in \mathbb{R}^d$. That is, for some $q \geq 1$ define

$$E_0 := \left\{ x \in \mathbb{R}^d \,:\, \|x - x_0\|_{L_q} < r_0 \right\},$$
$$E_\star := \left\{ x \in \mathbb{R}^d \,:\, \|x - x_0\|_{L_q} < r_\star \right\},$$

where $\|\cdot\|_{L_q}$ denotes the L_q norm. Assume that $r_0 = kr_\star$ with $k > 1$, so that $E_\star \subset E_0$ as required. Denote by V_0 and $V_\star$ the respective volumes of E_0 and $E_\star$.

A decreasing sequence of nested sets can in this case be formed by defining a decreasing sequence of radii $(r_p)_{p=0}^n$, where $r_0 > r_1 > \cdots > r_n := r_\star$. For $p \in \{0, \ldots, n\}$ one may then take

$$E_p := \left\{ x \in \mathbb{R}^d \,:\, \|x - x_0\|_{L_q} < r_p \right\},$$

defining V_p as the volume of E_p.

If one chooses the distribution π to be the uniform distribution on E_0, then for each $p \in \{0, \ldots, n\}$, π_p is the uniform distribution on E_p, admitting the unnormalised probability density function $\bar{\pi}_p = \mathbb{1}_{E_p}$. As such $Z_p = \int_{\mathsf{X}} \bar{\pi}_p(x)\,\mathrm{d}x = V_p/V_0$, so that the normalising constants are proportional to the volumes of the sets. Since $r_0 = kr_\star$, one has

$$\kappa = \frac{Z_0}{Z_\star} = \frac{V_0}{V_\star} = \frac{r_0^d}{r_\star^d} = k^d,$$

since the d-dimensional volume of an L_q ball is proportional to the dth power of its radius.

By Proposition 4.6, if n is fixed then the optimal value of $\sigma_{\mathbb{1}}^2$ in the perfectly-mixing case is $n(k^{d/n} - 1)$, obtained when $V_p = k^{-d/n} V_{p-1}$ for all $p \in \{1, \ldots, n\}$. This in turn corresponds to choosing the sequence of radii to follow a geometric progression, with $r_p = k^{-1/n} r_{p-1}$ for all $p \in \{1, \ldots, n\}$. Proposition 4.8 implies that the optimal value of n in the sense of minimising $n\sigma_{\mathbb{1}}^2$ is $n^\star \approx \alpha d \log(k)$ (where $\alpha \approx 0.6275$), which is a linear function of the dimension d. The corresponding minimal value of $n\sigma_{\mathbb{1}}^2$ is then

$$\begin{aligned}(n^\star)^2(k^{d/n^\star} - 1) &= (\alpha d \log(k))^2 \left(k^{d/(\alpha d \log(k))} - 1\right) \\ &= \alpha^2(\mathrm{e}^{1/\alpha} - 1) d^2 \log(k)^2,\end{aligned}$$

which is quadratic in d.

4.3. Normal distributions with equal means

The convenient properties of Gaussian densities have long been exploited in sequential estimation problems, notably in the Kalman filter (Kalman, 1960), used to compute exact filtering and smoothing distributions associated with linear Gaussian state space models. For the same reasons, SMC samplers targeting sequences of normal distributions $(\pi_p)_{p=0}^n$ provide useful examples for theoretical analysis. In particular, in the perfectly-mixing case all of the Markov kernels M_p admit Gaussian densities, facilitating the derivation of closed-form expressions for quantities of interest.

Furthermore, the ubiquity of normal distributions leads to many possible practical applications of such analytical results. For example, as discussed in Sections 2.2.1 and 2.3.1, a common application of SMC samplers chooses the target distribution $\pi_\star$ to be some Bayesian posterior. The Bernstein–von Mises theorem (see e.g. van der Vaart, 2000, Section 10.2) gives conditions under which such a posterior converges weakly to a normal distribution as the number of observations tends to infinity.

Let us therefore consider an SMC sampler targeting some normal distribution $\pi_\star = \mathcal{N}(\mu_\star, \Sigma_\star)$, initialised with a normal distribution $\pi_0 = \mathcal{N}(\mu_0, \Sigma_0)$. The problem of selecting a sequence or path of distributions interpolating two normal distributions has been investigated in several applications outside the SMC framework, for example in relation to optimal transport in stochastic systems (see e.g. Chen et al., 2015). For the setting of this section, in which we aim to choose a distribution schedule $(\pi_p)_{p=0}^n$ for use in an SMC sampler, we shall restrict our attention to schedules comprised of normal distributions. That is, for $p \in \{0, \dots, n\}$ we consider $\pi_p = \mathcal{N}(\mu_p, \Sigma_p)$, with mean $\mu_p \in \mathbb{R}^d$ and positive definite covariance matrix $\Sigma_p \in \mathbb{R}^{d \times d}$.

We shall aim to derive conditions under which such a distribution schedule may be deemed to be optimal in perfectly-mixing settings, in the sense described in Section 4.1. Following Proposition 4.1, for fixed n this corresponds to determining the sequence of normal distributions for which the sum of chi-squared distances $\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \parallel \pi_p)$ is minimised. Consider two normal distributions with respective means $\mu_\mathrm{A}, \mu_\mathrm{B} \in \mathbb{R}^d$, and respective positive definite covariance matrices $\Sigma_\mathrm{A}, \Sigma_\mathrm{B} \in \mathbb{R}^{d \times d}$; provided that $2\Sigma_\mathrm{A}^{-1} - \Sigma_\mathrm{B}^{-1}$ is positive definite, the chi-squared distance of $\mathcal{N}(\mu_\mathrm{A}, \Sigma_\mathrm{A})$ from $\mathcal{N}(\mu_\mathrm{B}, \Sigma_\mathrm{B})$ is given by (Bock, 2012, page 158)

$$
\begin{aligned}
&D_{\chi^2}\left(\mathcal{N}(\mu_\mathrm{A}, \Sigma_\mathrm{A}) \| \mathcal{N}(\mu_\mathrm{B}, \Sigma_\mathrm{B})\right) = \\
&\quad \frac{\det(\Sigma_\mathrm{B}\Sigma_\mathrm{A}^{-1})}{\sqrt{\det(2\Sigma_\mathrm{B}\Sigma_\mathrm{A}^{-1} - I)}} \exp\left(\frac{1}{2}\left[\left(2\Sigma_\mathrm{A}^{-1}\mu_\mathrm{A} - \Sigma_\mathrm{B}^{-1}\mu_\mathrm{B}\right)^\mathsf{T} \left(2\Sigma_\mathrm{A}^{-1} - \Sigma_\mathrm{B}^{-1}\right)^{-1} \left(2\Sigma_\mathrm{A}^{-1}\mu_\mathrm{A} - \Sigma_\mathrm{B}^{-1}\mu_\mathrm{B}\right)\right.\right. \\
&\qquad \left.\left. + \mu_\mathrm{B}^\mathsf{T}\Sigma_\mathrm{B}^{-1}\mu_\mathrm{B} - 2\mu_\mathrm{A}^\mathsf{T}\Sigma_\mathrm{A}^{-1}\mu_\mathrm{A}\right]\right) - 1.
\end{aligned}
$$

By elementary manipulations, this may be simplified to

$$D_{\chi^2}\left(\mathcal{N}(\mu_A, \Sigma_A) \| \mathcal{N}(\mu_B, \Sigma_B)\right) = \frac{\det(\Sigma_B \Sigma_A^{-1})}{\sqrt{\det(2\Sigma_B \Sigma_A^{-1} - I)}} \exp\left[(\mu_A - \mu_B)^\top (2\Sigma_B - \Sigma_A)^{-1} (\mu_A - \mu_B)\right] - 1. \tag{4.7}$$

If $2\Sigma_A^{-1} - \Sigma_B^{-1}$ is not positive definite then the integral in Definition 3.1 of the chi-squared distance does not converge, and so $D_{\chi^2}\left(\mathcal{N}(\mu_A, \Sigma_A) \| \mathcal{N}(\mu_B, \Sigma_B)\right)$ is undefined.

We shall make the following assumption of the target distribution $\pi_\star = \mathcal{N}(\mu_\star, \Sigma_\star)$ and initial distribution $\pi_0 = \mathcal{N}(\mu_0, \Sigma_0)$:

Assumption 4.10. The chi-squared distance of $\pi_\star$ from π_0 is well defined; that is, $2\Sigma_\star^{-1} - \Sigma_0^{-1}$ is positive definite.

As has previously been discussed, π_0 is generally chosen to place non-negligible mass on a large subset of the space, with the intention that this includes the areas of high mass of $\pi_\star$. This assumption will therefore hold in most practical settings involving normal or approximately normal distributions, since π_0 will typically be much more diffuse than $\pi_\star$.

In general the optimal sequences of parameters $(\mu_p)_{p=0}^n$ and $(\Sigma_p)_{p=0}^n$, in the sense of minimising the sum of chi-squared distances between consecutive distributions, do not admit closed-form expressions. However, one setting that is conducive to analysis is that in which the mean of the target distribution is equal to that of the initial distribution; that is, $\mu_0 = \mu_\star = \mu \in \mathbb{R}^d$. While this is unlikely in practice, this may provide a useful approximation of the more general setting. For example, if $\pi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ is very diffuse compared to $\pi_\star = \mathcal{N}(\mu_\star, \Sigma_\star)$, then $\mu_\star$ may lie in an area of high mass of π_0, so that it is 'close to' μ_0 relative to the marginal standard deviations of π_0.

For example, the expression (4.7) for the chi-squared distance of $\mathcal{N}(\mu_A, \Sigma_A)$ from $\mathcal{N}(\mu_B, \Sigma_B)$ is dependent on the means solely through their difference, via the quadratic form

$$(\mu_A - \mu_B)^\top (2\Sigma_B - \Sigma_A)^{-1} (\mu_A - \mu_B). \tag{4.8}$$

If $\mu_A = \mu_B$ then this evaluates to zero; however, if $2\Sigma_B - \Sigma_A$ is sufficiently diffuse, then this quadratic form may have a value close to zero even if $\mu_A \neq \mu_B$. As such, if Σ_A is sufficiently concentrated compared to Σ_B, this chi-squared distance may be well approximated by assuming that the means are equal.

We consider this setting for the remainder of this section; that is, we consider $\pi_\star = \mathcal{N}(\mu, \Sigma_\star)$ and $\pi_0 = \mathcal{N}(\mu, \Sigma_0)$ for some common mean $\mu \in \mathbb{R}^d$, and we look to derive properties of the optimal sequence of interpolating normal distributions $(\pi_p)_{p=0}^n$.

4.3.1. Optimal distribution schedule for fixed n

As previously, we begin by considering this schedule selection problem for a fixed schedule length n. The following initial result relates to the sequence of means $(\mu_p)_{p=0}^n$ associated with each of the distributions in the optimal schedule. In this setting in which the target distribution $\pi_\star$ and initial distribution π_0 have the same mean, the relative asymptotic variance $\sigma_{\mathbb{1}}^2$ is always minimised when each of the intermediate distributions also has this mean value.

Lemma 4.11. *Consider the sequence of distributions $(\pi_p)_{p=0}^n$ such that for $p \in \{0, \dots, n\}$, $\pi_p = \mathcal{N}(\mu_p, \Sigma_p)$ for some $\mu_p \in \mathbb{R}^d$ and positive definite $\Sigma_p \in \mathbb{R}^{d\times d}$. Suppose $\mu_0 = \mu_n = \mu$. Then for any fixed sequence of covariance matrices $(\Sigma_p)_{p=0}^n$, the sequence of means $(\mu_p)_{p=0}^n$ that minimises $\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ is given by $\mu_p = \mu$ for all $p \in \{0, \dots, n\}$.*

Proof. For a square matrix $M \in \mathbb{R}^{d\times d}$, we write $M \succ 0$ when M is positive definite. Consider the expression (4.7) for the chi-squared distance of $\mathcal{N}(\mu_A, \Sigma_A)$ from $\mathcal{N}(\mu_B, \Sigma_B)$, which holds if $2\Sigma_A^{-1} - \Sigma_B^{-1} \succ 0$, or equivalently, $(\Sigma_A/2)^{-1} - \Sigma_B^{-1} \succ 0$.

For any $M, N \in \mathbb{R}^{d\times d}$ with $M, N \succ 0$, we have that M, N are invertible, and $M^{-1}, N^{-1} \succ 0$ (Horn and Johnson, 1985, page 430). Additionally, if $M - N \succ 0$, then $N^{-1} - M^{-1} \succ 0$ (Horn and Johnson, 1985, Corollary 7.7.4a); therefore,

$$(\Sigma_A/2)^{-1} - \Sigma_B^{-1} \succ 0 \implies \Sigma_B - \Sigma_A/2 \succ 0.$$

Furthermore, for any $\alpha > 0$, if $M \succ 0$ then $\alpha M \succ 0$ (Horn and Johnson, 1985, Observation 7.1.3); applying this result for $\alpha = 2$, the above implies $2\Sigma_B - \Sigma_A \succ 0$, and therefore $(2\Sigma_B - \Sigma_A)^{-1} \succ 0$.

It follows that if the chi-squared distance $D_{\chi^2}\big(\mathcal{N}(\mu_A, \Sigma_A) \,\|\, \mathcal{N}(\mu_B, \Sigma_B)\big)$ is defined, then the quadratic form (4.8) is always positive, except when $\mu_A = \mu_B$, in which case it is zero. The chi-squared distance (4.7) is an increasing function of this quadratic form. Therefore, for fixed Σ_A and Σ_B the chi-squared distance may be seen as a function of the difference $\mu_A - \mu_B$ that is minimised when this is zero.

In the case of the sequence $(\pi_p)_{p=0}^n$ we see that choosing $\mu_p = \mu$ for all $p \in \{0, \dots, n\}$ satisfies the constraint that $\mu_0 = \mu_n = \mu$. Since the difference $\mu_{p+1} - \mu_p$ between each pair of means is zero, it follows that for any fixed sequence of covariance matrices $(\Sigma_p)_{p=0}^n$, this sequence minimises the sum $\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$. ∎

It follows that in this setting, the optimal distribution schedule of normal distributions $(\pi_p)_{p=0}^n$ is such that $\pi_p = \mathcal{N}(\mu, \Sigma_p)$ for all $p \in \{0, \dots, n\}$, so that the problem of selecting a distribution schedule may be reduced to that of selecting a sequence of covariance matrices. In the case of a fixed schedule length n, for given Σ_0 and $\Sigma_n := \Sigma_\star$, we look to choose

$(\Sigma_p)_{p=1}^{n-1}$ to minimise

$$\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p) = \sum_{p=0}^{n-1} \left[\frac{\det\left(\Sigma_p \Sigma_{p+1}^{-1}\right)}{\sqrt{\det\left(2\Sigma_p \Sigma_{p+1}^{-1} - I\right)}} - 1 \right], \tag{4.9}$$

where we require $2\Sigma_{p+1}^{-1} - \Sigma_p^{-1}$ to be positive definite for all $p \in \{0, \dots, n-1\}$, so that all these chi-squared distances are defined.

In general, this constrained matrix optimisation problem is not easily solved. However, closed-form expressions for an optimal sequence of covariance matrices may be derived in the special case that Σ_0 and $\Sigma_\star$ are simultaneously diagonalisable. That is, we assume there exists some orthogonal matrix $S \in \mathbb{R}^{d\times d}$, and diagonal matrices $M_0, M_\star \in \mathbb{R}^{d\times d}$, such that

$$\Sigma_0 = S^{-1} M_0 S, \qquad \Sigma_\star = S^{-1} M_\star S.$$

Noting that all covariance matrices are (individually) diagonalisable, this occurs if and only if Σ_0 and $\Sigma_\star$ commute (Horn and Johnson, 1985, Theorem 1.3.12). This includes the case where π_0 is chosen to be a spherical Gaussian (so that $\Sigma_0 = kI$ for some $k > 0$). Although this assumption is restrictive, analysis of this simple setting allows the development of heuristics that may be useful more generally. Furthermore, consideration of this case allows us to conjecture a result for the more general case, which we discuss subsequently.

We here consider intermediate covariance matrices $(\Sigma_p)_{p=1}^{n-1}$ of the form $\Sigma_p = S^{-1} M_p S$ for $p \in \{1, \dots, n-1\}$, so that these are all diagonalisable in the same basis as Σ_0 and $\Sigma_\star$. The problem of finding the optimal distribution schedule, in the sense of minimising $\sigma_{\mathbb{1}}^2$, reduces to the problem to finding the optimal sequence of diagonal matrices $(M_p)_{p=0}^n$. The following result provides an expression for this sequence; we consider its practical relevance in the discussion that follows.

Proposition 4.12. *For a fixed value of n, consider a distribution schedule $(\pi_p)_{p=0}^n$ defined by $\pi_p = \mathcal{N}(\mu, S^{-1} M_p S)$ for $p \in \{0, \dots, n\}$, for some $\mu \in \mathbb{R}^d$, orthogonal matrix $S \in \mathbb{R}^{d\times d}$ and sequence of diagonal matrices $(M_p)_{p=0}^n$ in $\mathbb{R}^{d\times d}$. For fixed M_0 and M_n, the sum $\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ of chi-squared distances between consecutive distributions is minimised when the sequence $(M_p)_{p=0}^n$ is chosen to be a geometric progression, in which case all of these chi-squared distances are equal. Specifically, writing $M_0 M_n^{-1} = \operatorname{diag}(\kappa_1, \dots, \kappa_d)$ the minimal value of $\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ is*

$$n \left[\prod_{i=1}^{d} \frac{\kappa_i^{1/n}}{\sqrt{2\kappa_i^{1/n} - 1}} - 1 \right],$$

obtained when $M_p = \operatorname{diag}(\kappa_1^{-1/n}, \dots, \kappa_d^{-1/n}) M_{p-1}$ for all $p \in \{1, \dots, n\}$.

Proof. For $p \in \{0, \dots, n-1\}$, we have by elementary manipulations that

$$\Sigma_p \Sigma_{p+1}^{-1} = S^{-1} M_p M_{p+1}^{-1} S.$$

From (4.9), the sum of chi-squared distances may therefore be expressed as

$$\begin{aligned}\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p) &= \sum_{p=0}^{n-1} \left[\frac{\det\left(S^{-1} M_p M_{p+1}^{-1} S\right)}{\sqrt{\det\left(S^{-1}\left[2M_p M_{p+1}^{-1} - I\right] S\right)}} - 1 \right] \\ &= \sum_{p=0}^{n-1} \left[\frac{\det\left(M_p M_{p+1}^{-1}\right)}{\sqrt{\det\left(2M_p M_{p+1}^{-1} - I\right)}} - 1 \right].\end{aligned}$$

For $p \in \{0, \dots, n\}$, let the ith diagonal element of M_p be denoted by $m_{p,i}$, so that $M_p = \operatorname{diag}(m_{p,1}, \dots, m_{p,d})$. Then, noting that the determinant of a diagonal matrix is equal to the product of its diagonal elements, we have

$$\begin{aligned}\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p) &= \sum_{p=0}^{n-1} \left[\frac{\prod_{i=1}^{d} m_{p,i}/m_{p+1,i}}{\sqrt{\prod_{i=1}^{d} \left[2m_{p,i}/m_{p+1,i} - 1\right]}} - 1 \right] \\ &= \sum_{p=0}^{n-1} \left[\prod_{i=1}^{d} \frac{m_{p,i}/m_{p+1,i}}{\sqrt{2m_{p,i}/m_{p+1,i} - 1}} \right] - n.\end{aligned}$$

Denoting $\xi_{p,i} := m_{p-1,i}/m_{p,i}$, we therefore look to minimise

$$\sum_{p=1}^{n} \left[\prod_{i=1}^{d} \frac{\xi_{p,i}}{\sqrt{2\xi_{p,i} - 1}} \right] - n.$$

We now derive the constraints for this optimisation problem. Firstly, we require each chi-squared distance in the sum to be well defined. For $p \in \{0, \dots, n-1\}$, $D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ is well defined when $2\Sigma_{p+1}^{-1} - \Sigma_p^{-1} = S^{-1}\left[2M_{p+1}^{-1} - M_p^{-1}\right] S$ is positive definite. Since S is an orthogonal matrix, this occurs if and only if $2M_{p+1}^{-1} - M_p^{-1}$ is positive definite (by, e.g., Horn and Johnson, 1985, Observation 7.1.7). This diagonal matrix is positive definite if and only if all its diagonal elements are positive; so for all $i \in \{1, \dots, d\}$ we require

$$\frac{2}{m_{p+1,i}} - \frac{1}{m_{p,i}} > 0 \implies \frac{m_{p,i}}{m_{p+1,i}} > \frac{1}{2}.$$

Therefore, for $p \in \{1, \dots, n\}$ and $i \in \{1, \dots, d\}$ we require $\xi_{p,i} > 1/2$.

An additional constraint is obtained by considering the product over p of $\xi_{p,i}$, for fixed $i \in \{1, \dots, d\}$. Recall that M_0 and M_n are both fixed; write $M_0 M_n^{-1} = \operatorname{diag}(\kappa_1, \dots, \kappa_d)$. Then

for $i \in \{1, \dots, d\}$ we have

$$\prod_{p=1}^{n} \xi_{p,i} = \prod_{p=1}^{n} \frac{m_{p-1,i}}{m_{p,i}} = \frac{m_{0,i}}{m_{n,i}} = \kappa_i.$$

To summarise, our constrained optimisation problem is as follows. Define $g : (1/2, \infty) \to \mathbb{R}_+$ by $g(x) := x/\sqrt{2x-1}$, and define $\tilde{g} : (1/2, \infty)^d \to \mathbb{R}_+$ by

$$\tilde{g}(x_1, \dots, x_d) := \prod_{i=1}^{d} g(x_i). \tag{4.10}$$

We look to choose n vectors in $(1/2, \infty)^d$, which we denote $(\xi_{p,1}, \dots, \xi_{p,d})$ for $p \in \{1, \dots, n\}$, with fixed componentwise product $(\kappa_1, \dots, \kappa_d)$, in order to minimise $\sum_{p=1}^{n} \tilde{g}(\xi_{p,1}, \dots, \xi_{p,d}) - n$.

This problem suggests the use of Lemma 4.5, for which it is necessary that the function $(x_1, \dots, x_d) \mapsto \tilde{g}(\exp(x_1), \dots, \exp(x_d))$ is convex. From the definition (4.10) of $\tilde{g}$ in terms of the function g, we see that since g is non-negative, it is sufficient to show that the function $x \mapsto g(\exp(x))$ is convex. By elementary computations, the second derivative of $g(\exp(x))$ with respect to x is

$$\frac{\exp(x)\left[(\exp(x)-1)^2 + \exp(x)\right]}{(2\exp(x)-1)^{5/2}},$$

which is readily seen to be positive when $\exp(x) > 1/2$. The convexity condition therefore holds.

By Lemma 4.5, it follows that the unique minimal value of $\sum_{p=1}^{n} \tilde{g}(\xi_{p,1}, \dots, \xi_{p,d}) - n$ is

$$n\tilde{g}(\kappa_1^{1/n}, \dots, \kappa_d^{1/n}) - n = n\left[\prod_{i=1}^{d} \frac{\kappa_i^{1/n}}{\sqrt{2\kappa_i^{1/n} - 1}} - 1\right],$$

obtained when $\xi_{p,i} = \kappa_i^{1/n}$ for all $p \in \{1, \dots, n\}$, $i \in \{1, \dots, d\}$. To confirm that this satisfies the constraint that each $\xi_{p,i}$ is greater than $1/2$, recall from Assumption 4.10 that $D_{\chi^2}(\pi_n \| \pi_0)$ is well defined. By a similar argument to that used to show $\xi_{p,i} > 1/2$, we find that each diagonal element κ_i of $M_0 M_n^{-1}$ is greater than $1/2$, and therefore $\kappa_i^{1/n} > 1/2$ for each $i \in \{1, \dots, d\}$.

The sum of chi-squared distances therefore takes this minimal value when $M_{p-1}M_p^{-1} = \operatorname{diag}(\kappa_1^{1/n}, \dots, \kappa_d^{1/n})$ for all $p \in \{1, \dots, n\}$, which results in all the chi-squared distances being equal. ∎

From this result we find that, among all sequences of intermediate covariance matrices $(\Sigma_p)_{p=0}^{n}$ that are diagonalisable with respect to the same common basis as Σ_0 and $\Sigma_\star$, the optimal sequence is itself a geometric progression of matrices. Denote $K := \Sigma_0 \Sigma_\star^{-1}$; then the optimal sequence described in Proposition 4.12 is such that $\Sigma_{p-1} = K^{1/n}\Sigma_p$ for all $p \in \{1, \dots, n\}$. Here, $K^{1/n}$ is the principal nth root of K, which may be obtained from the

eigendecomposition of K by replacing all its eigenvalues with their nth roots; this is well defined since $K = \Sigma_0 \Sigma_\star^{-1}$ is a product of positive definite matrices, and therefore all its eigenvalues are positive (Horn and Johnson, 1985, Corollary 7.6.2).

Remark 4.13. Consider the minimal value of $\sigma_{\mathbb{1}}^2$ for a fixed sequence of length n, as given in Proposition 4.12. For any values of $\kappa_{1:d}$, this value converges to 0 as $n \longrightarrow \infty$ (as may be shown using an appropriate Taylor expansion, for example; we later use this technique in the proof of Proposition 4.16). It follows that when using distribution schedules of this form, the relative asymptotic variance $\sigma_{\mathbb{1}}^2$ may be made arbitrarily small by introducing additional intermediate distributions. This stands in contrast to the 'nested sets' setting of Section 4.2, for which this is not the case (see Remark 4.7).

A possible practical application of Proposition 4.12 is described in the following remark.

Remark 4.14. In general, the optimal distribution schedule described in Proposition 4.12 does not correspond to a temperature schedule. An exception is the case in which $\Sigma_0 = \kappa \Sigma_\star$ for some $\kappa > 1/2$, so that the covariance matrices of the intermediate distributions are of the form $\Sigma_p = \kappa^{1-p/n} \Sigma_\star$. Assuming $\kappa \neq 1$, the optimal distribution schedule of length n corresponds to that generated by a temperature schedule $(\beta_p)_{p=0}^n$ given by

$$\beta_p = \frac{\kappa^{p/n} - 1}{\kappa - 1}, \qquad p \in \{0, \dots, n\}. \tag{4.11}$$

This could be used as the basis for a heuristic approach for selecting a temperature schedule in general settings involving approximately normal distributions, given an approximation of the relative scales of the initial and final distributions. For example, define $\tilde{\kappa} := \det(\Sigma_0 \Sigma_\star^{-1})^{1/d}$, which corresponds to the ratio of standardised generalised variances of Σ_0 and $\Sigma_\star$ (as defined by SenGupta, 1987). A temperature schedule could be generated using (4.11), with κ replaced by some approximation of this value.

As previously discussed, for a sequence of distributions $(\pi_p)_{p=0}^n$ with $\pi_p = \mathcal{N}(\mu, \Sigma_p)$, the general problem of finding the optimal sequence of covariance matrices to minimise the sum of chi-squared distances is not easily solved analytically. However, we may solve this optimisation problem numerically. Consider the setting of Proposition 4.12, where the initial and final covariance matrices are simultaneously diagonalisable. Numerical investigations of such settings suggest that the sequence of intermediate covariance matrices described in that result is not only the optimal sequence of the form $\Sigma_p = S^{-1} M_p S$, $p \in \{0, \dots, n\}$, but the optimal sequence over all feasible sequences of covariance matrices.

Indeed, numerical investigations suggest that in the most general setting where Σ_0 and $\Sigma_\star$ are not simultaneously diagonalisable, the optimal sequence corresponds to the previously-described geometric progression of matrices. We therefore conjecture the following general result for the optimal distribution schedule between normal distributions with equal means.

Conjecture 4.15. *For a fixed value of n, consider a distribution schedule $(\pi_p)_{p=0}^n$ defined by $\pi_p = \mathcal{N}(\mu, \Sigma_p)$ for $p \in \{0, \dots, n\}$, for some $\mu \in \mathbb{R}^d$ and sequence of positive definite covariance matrices $(\Sigma_p)_{p=0}^n$ in $\mathbb{R}^{d\times d}$. Suppose Σ_0 and Σ_n are fixed; denote $K \coloneqq \Sigma_0\Sigma_n^{-1}$. The sum $\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ of chi-squared distances between consecutive distributions is minimised when $\Sigma_{p-1} = K^{1/n}\Sigma_p$ for all $p \in \{1, \dots, n\}$, where $K^{1/n}$ is the principal nth root of K. In this case case all of these chi-squared distances are equal, and the minimal value of $\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ is*

$$n\left[\frac{\det\left(K^{1/n}\right)}{\sqrt{\det\left(2K^{1/n} - I\right)}} - 1\right].$$

Proving this result would require an analytical solution to the following constrained matrix optimisation problem. Denoting $\Xi_p \coloneqq \Sigma_{p-1}\Sigma_p^{-1}$ for $p \in \{1, \dots, n\}$, from (4.9) we look to minimise

$$\sum_{p=1}^{n}\left[\frac{\det\left(\Xi_p\right)}{\sqrt{\det\left(2\Xi_p - I\right)}} - 1\right]$$

under the constraints that $\prod_{p=1}^n \Xi_p = \Sigma_0\Sigma_n^{-1} =: K$, and $2\Sigma_p^{-1} - \Sigma_{p-1}^{-1}$ is positive definite for all $p \in \{1, \dots, n\}$. Notably, the latter constraint does not correspond to each Ξ_p being positive definite; indeed, in general K is not symmetric. A generalisation the proof of Proposition 4.12 may be possible, but would not be straightforward: for example, it would require a suitable generalisation of Lemma 4.5, and would appear to depend on the convexity of a suitably-defined function involving matrix determinants.

4.3.2. Optimal schedule length n

We now look to find the value n that minimises $n\sigma_{\mathbb{1}}^2$, when $\sigma_{\mathbb{1}}^2$ takes its minimal possible value for fixed n. For clarity of exposition, we shall here analyse the setting described in Remark 4.14, in which the initial and target covariance matrices are equal up to some multiplicative constant; that is, $\Sigma_0 = \kappa\Sigma_\star$ for some $\kappa > 1/2$. In this case, the values $\kappa_{1:d}$ described in Proposition 4.12 are all equal to κ, so that the minimal value of $\sigma_{\mathbb{1}}^2$ for fixed n is

$$n\left[\left(\frac{\kappa^{1/n}}{\sqrt{2\kappa^{1/n} - 1}}\right)^d - 1\right]. \tag{4.12}$$

As well as simplifying the notation, this setting is also convenient for considering the role of the dimension d in this optimisation problem. We emphasise however that a similar analysis may be performed in the more general case described by Proposition 4.12.

We therefore look to find the value $n^\star$ that minimises n times the minimal value of $\sigma_{\mathbb{1}}^2$ given by (4.12); that is, the value $n^\star$ that minimises

$$n^2\left[\left(\frac{\kappa^{1/n}}{\sqrt{2\kappa^{1/n} - 1}}\right)^d - 1\right].$$

As in Section 4.2.2, we approach this problem by considering this as a function of real-valued $n \in \mathbb{R}_+$. It is not possible to find the stationary points of this function analytically; however, the existence of a global minimum can be proved by considering its asymptotic behaviour, as in the following result.

Proposition 4.16. *The function $f : \mathbb{R}_+ \longrightarrow \mathbb{R}$ given by*

$$f(x) := x^2 \left[\left(\frac{\kappa^{1/x}}{\sqrt{2\kappa^{1/x} - 1}} \right)^d - 1 \right],$$

where $\kappa > 1/2$ and $d \geq 1$ are constants, has a global minimum value of the form $h(d)\log^2(\kappa)$, where $0 < h(d) < d/2$.

Proof. It is convenient first to define $\tilde{f}, g : \mathbb{R}_+ \longrightarrow \mathbb{R}$ such that

$$\tilde{f}(x) := x^2 \left[\left(\frac{\exp(1/x)}{\sqrt{2\exp(1/x) - 1}} \right)^d - 1 \right], \qquad g(x) := \left(\frac{\exp(x)}{\sqrt{2\exp(x) - 1}} \right)^d. \tag{4.13}$$

We see that $f(x) = \log^2(\kappa)\tilde{f}(x/\log(\kappa))$ and $\tilde{f}(x) = x^2[g(1/x) - 1]$.

Consider a Taylor series expansion of g about some point $a \in \mathbb{R}_+$. Writing the remainder term in Lagrange form, one has that for $y \in \mathbb{R}_+$,

$$g(y) = g(a) + g'(a)(y-a) + \frac{g''(a)}{2}(y-a)^2 + \frac{g'''(a + t(y-a))}{6}(y-a)^3 \tag{4.14}$$

for some $t \in (0, 1)$. Elementary calculations give

$$\begin{aligned}
g'(y) &= d\frac{\exp(y) - 1}{2\exp(y) - 1}g(y), \\
g''(y) &= d\left[\frac{\exp(y) - 1}{2\exp(y) - 1}g'(y) + \frac{\exp(y)}{(2\exp(y) - 1)^2}g(y)\right], \\
g'''(y) &= d\left[\frac{\exp(y) - 1}{2\exp(y) - 1}g''(y) + 2\frac{\exp(y)}{(2\exp(y) - 1)^2}g'(y) - \frac{\exp(y)(2\exp(y) + 1)}{(2\exp(y) - 1)^3}g(y)\right].
\end{aligned}$$

Considering the limits of $g(y)$ and these first three derivatives as $y \longrightarrow 0$, we find

$$\begin{aligned}
\lim_{y\to 0} g(y) &= 1, \\
\lim_{y\to 0} g'(y) &= 0, \\
\lim_{y\to 0} g''(y) &= d, \\
\lim_{y\to 0} g'''(y) &= -3d.
\end{aligned}$$

Taking the limit of (4.14) as $a \longrightarrow 0$, one therefore has that for some $t \in (0,1)$,

$$g(y) = 1 + \frac{dy^2}{2} + \frac{g'''(ty)}{6}y^3.$$

Therefore, for $x = 1/y \in \mathbb{R}_+$, we have

$$\tilde{f}(x) = x^2\left[g\left(\frac{1}{x}\right) - 1\right] = \frac{d}{2} + \frac{g'''(t/x)}{6x}. \tag{4.15}$$

Note that since t is bounded, $\lim_{x\to\infty} g'''(t/x) = \lim_{y\to 0} g'''(y) = -3d$. It follows that

$$\lim_{x\to\infty} \frac{g'''(t/x)}{6x} = 0;$$

in addition, since g''' is continuous and $\lim_{x\to\infty} g'''(t/x)$ is negative, this limit is approached from below.

It follows from (4.15) that

$$\lim_{x\to\infty} \tilde{f}(x) = \frac{d}{2}$$

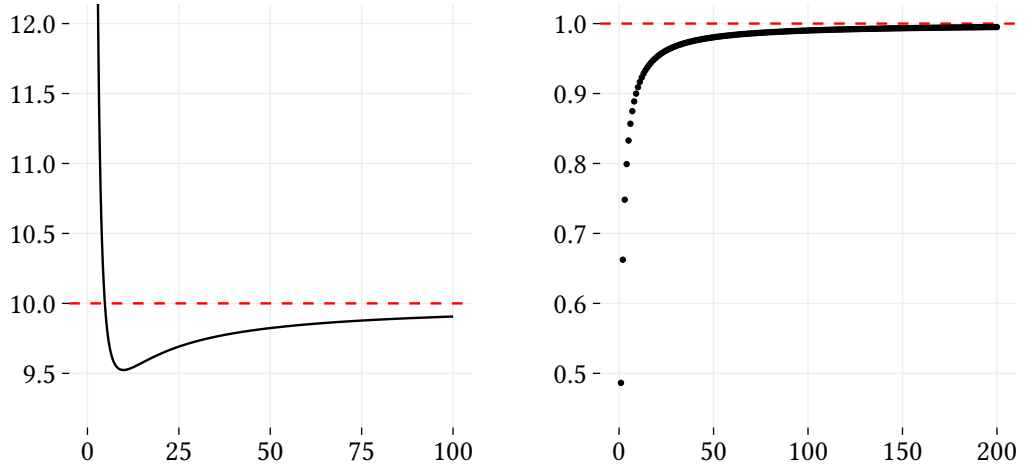
and that this limit is approached from below, so that for sufficiently large x, $\tilde{f}(x) < d/2$. Noting that $\lim_{x\to 0} \tilde{f}(x) = \infty$, it follows that $\tilde{f}$ must have a global minimum; denoting this value by $h(d)$, one has $0 < h(d) < d/2$. Finally, since $f(x) = \log^2(\kappa)\tilde{f}(x/\log(\kappa))$, f must have a global minimum value of $h(d)\log^2(\kappa)$. ∎

As well as proving that an optimal $n^\star$ does exist in this scenario, this result makes clear the dependence of the minimal value of $n\sigma_{\mathbb{1}}^2$ on each of d and κ. In particular, it may be seen that the minimal value cannot grow faster than linearly with d. Another consequence of this result is that the value of x that minimises $f(x)$ must be of the form $\tilde{h}(d)\log(\kappa)$, where $\tilde{h}$ is some function of d.

This result leads to a possible strategy for choosing n. As described, $n\sigma_{\mathbb{1}}^2$ converges to $d\log^2(\kappa)/2$ as n tends to infinity, with this convergence occurring from below. On the other hand, $f(x)$ diverges to infinity as x tends to zero. This therefore suggests that in order to control $n\sigma_{\mathbb{1}}^2$, a safe (if possibly suboptimal) approach is to choose a large value of n, since for n sufficiently large this expression is bounded above by $d\log^2(\kappa)/2$.

Numerical findings allow various possible results about the nature of $n^\star$ to be conjectured. Figure 4.2a shows a typical shape of the graph of $\tilde{f}$ as defined in (4.13), a linear transformation of f that is independent of κ. This is observed to exhibit only one local (and global) minimum, which may be found by numerical optimisation. For each in grid of values of d, Figure 4.2b displays the ratio of this minimal value and $d/2$.

These results suggest that the relationship between the minimal value of $n\sigma_{\mathbb{1}}^2$ and d is asymptotically linear. Recalling that we denote by $h(d)$ the global minimum of $\tilde{f}$, it may additionally be conjectured that as $d \longrightarrow \infty$, $h(d)/(d/2) \longrightarrow 1$. This would imply that as



(a) $\tilde{f}(x)$ against x, and the constant function of value $d/2$ (dashed), when $d = 20$.

(b) $\min_{x \in \mathbb{R}_+} \tilde{f}(x)/(d/2)$ against d, and the constant function of value 1 (dashed).

Figure 4.2.: Plots describing $\tilde{f}$ as defined in (4.11), a linear transformation of the function f considered in Proposition 4.16, which corresponds to the value of $n\sigma_{\mathbb{1}}^2$ in a setting involving normal distributions. Note the truncated vertical axes.

the dimension tends to infinity, the difference vanishes between the optimal value of $n\sigma_{\mathbb{1}}^2$, and its asymptotic value as $n \longrightarrow \infty$. From a practical perspective, this means that the 'safe' approach of simply taking n very large should give a result that is close to optimal in high-dimensional settings.

These numerical optimisations also allow us to make a similar conjecture about the minimising value $n^\star$ of n: it appears that as $d \longrightarrow \infty$, $n^\star/(d \log(\kappa)/2) \longrightarrow 1$. As well as implying an asymptotically linear relationship, this gives an approximate value for $n^\star$ when d is large. While κ would usually not be known in advance, such a result could form the basis of a heuristic for choosing n for a high-dimensional space.

4.4. Properties of the chi-squared distance

In the results we have presented in this chapter, the distribution schedule $(\pi_p)_{p=0}^n$ minimising the sum of chi-squared distances $\sum_{p=0}^n D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p)$ is such that all of the chi-squared distances in this sum are equal. Unfortunately, this does not hold in general. To demonstrate this, we may consider the finite state space $\mathsf{X} = \{0, 1\}$, on which any probability measure is of the form $(1 - \rho)\delta_0 + \rho\delta_1$ for some $\rho \in [0, 1]$. This is the Bernoulli distribution with parameter ρ, which we shall denote by $\mathrm{Ber}(\rho)$.

Consider a schedule of distributions $(\pi_p)_{p=0}^n$ on X with $n = 2$, so that for $p \in \{0, 1, 2\}$ we may write $\pi_p = \mathrm{Ber}(\rho_p)$, for some $\rho_p \in [0, 1]$. Given π_0 and $\pi_2 = \pi_\star$, so that ρ_0 and ρ_2 are fixed, the selection of a distribution schedule is equivalent to choosing ρ_1. We shall consider the chi-squared distances between consecutive distributions as functions of ρ_1,

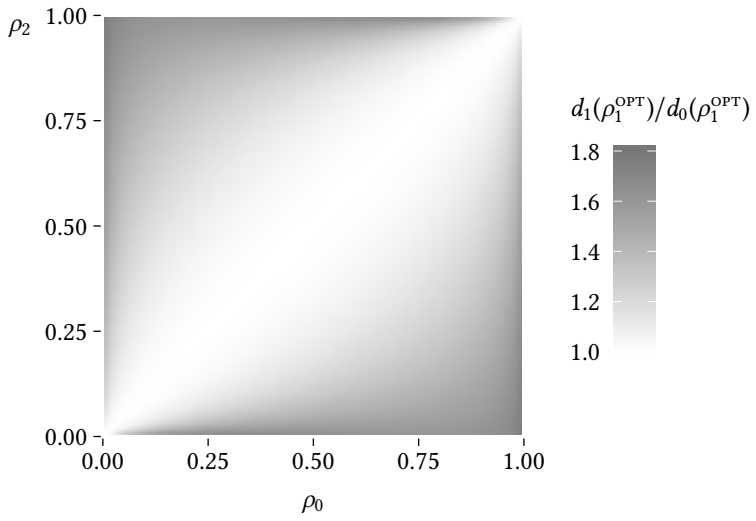


Figure 4.3.: Ratio of the expressions $d_1(\rho_1)$ and $d_0(\rho_1)$ defined in (4.17) and (4.16) respectively, describing the chi-squared distances between pairs of Bernoulli distributions, evaluated at the value ρ_1^{OPT} that minimises their sum. Displayed for a grid of pairs of values (ρ_0, ρ_2), with the convention that this ratio is 1 when both chi-squared distances are zero.

denoting these by d_0 and d_1 respectively. These take the forms

$$d_0(\rho_1) = D_{\chi^2}\big(\text{Ber}(\rho_1) \| \text{Ber}(\rho_0)\big) = \frac{(1-\rho_1)^2}{1-\rho_0} + \frac{\rho_1^2}{\rho_0} - 1, \tag{4.16}$$

$$d_1(\rho_1) = D_{\chi^2}\big(\text{Ber}(\rho_2) \| \text{Ber}(\rho_1)\big) = \frac{(1-\rho_2)^2}{1-\rho_1} + \frac{\rho_2^2}{\rho_1} - 1. \tag{4.17}$$

We look to find the value ρ_1^{OPT} of ρ_1 that minimises the sum of these two chi-squared distances.

Despite the simplicity of this setting, this optimisation problem does not admit a closed-form solution. However, for a given choice of ρ_0 and ρ_2 this may be solved numerically. We find that in general, the value ρ_1^{OPT} that minimises the sum $d_0(\rho_1) + d_1(\rho_1)$ does not result in the two chi-squared distances $d_0(\rho_1)$ and $d_1(\rho_1)$ being equal. Figure 4.3 illustrates this, displaying the ratio $d_1(\rho_1^{\text{OPT}})/d_0(\rho_1^{\text{OPT}})$ of the resulting chi-squared distances over a grid of values of ρ_0 and ρ_2, with the convention that this ratio is 1 if both chi-squared distances are zero. We here see that minimising the sum of chi-squared distances generally requires the first two distributions (π_0 and π_1) to be somewhat more similar than the final two (π_1 and π_2).

It follows that even in this perfectly-mixing setting, constructing a schedule of equally-spaced distributions (in the sense of the chi-squared distance) may not be optimal. As described in Section 3.2.2, an adaptive approach to parametric schedule selection proposed by Zhou et al. (2016) aims to construct a distribution schedule in which the chi-squared

distances between consecutive pairs of distributions are approximately constant. Considering the optimality criterion introduced in Section 3.3, there is therefore scope to develop methods that improve on this adaptive procedure. Indeed, in more general settings the terms in the decomposition of $\sigma^2_{\mathbb{1}}$ do not directly correspond to chi-squared distances between consecutive distributions, but also depend on the Markov kernels and resampling schedule, as shown in Proposition 3.13. We shall consider this in the following chapter.

As previously discussed in Section 3.2.2, the chi-squared distance is not a true 'distance', in the sense of being a metric. From Definition 3.1 we see that $D_{\chi^2}(\pi \,\|\, \mu)$ is non-negative, and equal to zero if and only if $\pi = \mu$; however, it is not symmetric in its arguments.

Rather, the chi-squared distance belongs to a class of functions known as f-divergences, introduced by Csiszár (1963) for quantifying the dissimilarity between two probability distributions defined on the same space (see Csiszár and Shields, 2004, Section 4 for a review of the definition and properties). A number of dissimilarity measures widely used in statistics and information theory are examples of such f-divergences, including the Kullback–Leibler divergence and total variation distance. The chi-squared distance has been well studied in this context; for example, Sason and Verdú (2016) derive a number of inequalities between the chi-squared distance and other f-divergences.

A consequence of the chi-squared distance not defining a metric is that it does not follow a triangle inequality. That is, for distributions π, ν and μ on X, the sum of $D_{\chi^2}(\pi \,\|\, \nu)$ and $D_{\chi^2}(\nu \,\|\, \mu)$ is not bounded below by $D_{\chi^2}(\pi \,\|\, \mu)$. Indeed as demonstrated by the results in this chapter, adding an additional intermediate distribution to a schedule may reduce the sum of chi-squared distances, and so the sum $D_{\chi^2}(\pi \,\|\, \nu) + D_{\chi^2}(\nu \,\|\, \mu)$ may in fact be rather less than $D_{\chi^2}(\pi \,\|\, \mu)$. By the following result, it is always possible to choose an intermediate distribution ν to achieve such a reduction.

Proposition 4.17. *Let π and μ be distributions on $(\mathsf{X}, \mathcal{X})$ with $\pi \ll \mu$. Define $\nu := (\pi + \mu)/2$. Then*

$$D_{\chi^2}(\pi \,\|\, \nu) + D_{\chi^2}(\nu \,\|\, \mu) \leq \frac{3}{4} D_{\chi^2}(\pi \,\|\, \mu).$$

Proof. First consider the term $D_{\chi^2}(\nu \,\|\, \mu)$. Since $\pi \ll \mu$ it is readily seen that $\nu \ll \mu$, so this chi-squared distance is well defined and may be expressed according to (3.2) as

$$\begin{aligned} D_{\chi^2}(\nu \,\|\, \mu) &= \int_{\mathsf{X}} \frac{\mathrm{d}\nu}{\mathrm{d}\mu}(x) \nu(\mathrm{d}x) - 1 \\ &= \frac{1}{2} \int_{\mathsf{X}} \frac{\mathrm{d}\nu}{\mathrm{d}\mu}(x) \mu(\mathrm{d}x) + \frac{1}{2} \int_{\mathsf{X}} \frac{\mathrm{d}\nu}{\mathrm{d}\mu}(x) \pi(\mathrm{d}x) - 1 \\ &= \frac{1}{2} + \frac{1}{2} \int_{\mathsf{X}} \frac{\mathrm{d}\nu}{\mathrm{d}\mu}(x) \pi(\mathrm{d}x) - 1. \end{aligned}$$

From the definition of the Radon–Nikodym derivative it may be shown that $\mathrm{d}\nu/\mathrm{d}\mu =$

$(\mathrm{d}\pi/\mathrm{d}\mu + \mathrm{d}\mu/\mathrm{d}\mu)/2$, and so

$$\begin{aligned} D_{\chi^2}(\nu \,\|\, \mu) &= \frac{1}{2} + \frac{1}{4}\int_{\mathsf{X}} \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x) + \frac{1}{4}\int_{\mathsf{X}} \frac{\mathrm{d}\mu}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x) - 1 \\ &= \frac{1}{2} + \frac{1}{4}\int_{\mathsf{X}} \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x) + \frac{1}{4} - 1 \\ &= \frac{1}{4}\left(\int_{\mathsf{X}} \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x) - 1\right) \\ &= \frac{1}{4} D_{\chi^2}(\pi \,\|\, \mu). \end{aligned} \tag{4.18}$$

Now consider the term $D_{\chi^2}(\pi \,\|\, \nu)$. We readily find that $\pi \ll \nu$ and $\mu \ll \nu$, and so from the definition of the Radon–Nikodym derivative it may be shown that

$$\frac{\mathrm{d}\pi}{\mathrm{d}\nu} \;=\; 2\frac{\mathrm{d}\pi}{\mathrm{d}(\pi+\mu)} \;=\; 2\left(\frac{\mathrm{d}(\pi+\mu)}{\mathrm{d}(\pi+\mu)} - \frac{\mathrm{d}\mu}{\mathrm{d}(\pi+\mu)}\right) \;=\; 2\left(\mathbb{1} - \frac{1}{2}\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\right).$$

Therefore,

$$\begin{aligned} D_{\chi^2}(\pi \,\|\, \nu) \;=\; \int_{\mathsf{X}} \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x) - 1 \;&=\; 2\int_{\mathsf{X}} \left(1 - \frac{1}{2}\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(x)\right)\pi(\mathrm{d}x) - 1 \\ &=\; 1 - \int_{\mathsf{X}} \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(x)\pi(\mathrm{d}x). \end{aligned} \tag{4.19}$$

To rewrite (4.19) we first note that since the Radon–Nikodym derivatives $\mathrm{d}\mu/\mathrm{d}\nu$ and $\mathrm{d}\nu/\mathrm{d}\mu$ are well defined, we have that for any $A \in \mathsf{X}$,

$$\int_A \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(x)\frac{\mathrm{d}\nu}{\mathrm{d}\mu}(x)\mu(\mathrm{d}x) = \int_A \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(x)\nu(\mathrm{d}x) = \mu(A).$$

It follows that the product of $\mathrm{d}\mu/\mathrm{d}\nu$ and $\mathrm{d}\nu/\mathrm{d}\mu$ must equal 1 μ-almost everywhere, and so $\mathrm{d}\mu/\mathrm{d}\nu = (\mathrm{d}\nu/\mathrm{d}\mu)^{-1}$ μ-almost everywhere. By symmetry, this is also true ν-almost everywhere; and since π is a linear combination of μ and ν, it is true π-almost everywhere.

Rewriting the integrand of (4.19) accordingly, and then applying Jensen's inequality with respect to the convex function $x \mapsto x^{-1}$, we have

$$\begin{aligned} D_{\chi^2}(\pi \,\|\, \nu) &= 1 - \int_{\mathsf{X}} \left(\frac{\mathrm{d}\nu}{\mathrm{d}\mu}(x)\right)^{-1} \pi(\mathrm{d}x) \\ &\le 1 - \left(\int_{\mathsf{X}} \frac{\mathrm{d}\nu}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x)\right)^{-1} \\ &= 1 - \left(\frac{1}{2}\int_{\mathsf{X}} \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x) + \frac{1}{2}\int_{\mathsf{X}} \frac{\mathrm{d}\mu}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x)\right)^{-1} \\ &= 1 - 2\left(\int_{\mathsf{X}} \frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\pi(\mathrm{d}x) + 1\right)^{-1}. \end{aligned}$$

By the form (3.2) of the chi-squared distance we therefore have

$$D_{\chi^2}(\pi \,\|\, \nu) \;\leq\; 1 - \frac{2}{D_{\chi^2}(\pi \,\|\, \mu) + 2} \;=\; \frac{D_{\chi^2}(\pi \,\|\, \mu)}{D_{\chi^2}(\pi \,\|\, \mu) + 2}. \tag{4.20}$$

Finally, combining (4.18) and (4.20) we have

$$D_{\chi^2}(\pi \,\|\, \nu) + D_{\chi^2}(\nu \,\|\, \mu) \;\leq\; \left(\frac{1}{D_{\chi^2}(\pi \,\|\, \mu) + 2} + \frac{1}{4} \right) D_{\chi^2}(\pi \,\|\, \mu) \;\leq\; \frac{3}{4} D_{\chi^2}(\pi \,\|\, \mu),$$

as required. ∎

Owing to the 'reduction factor' of 3/4 in this result it follows that for any initial distribution π_0 and final distribution $\pi_\star$, it is always possible to construct a schedule for which the relative asymptotic variance $\sigma^2_{\mathbb{1}}$ is arbitrarily small, by adding intermediate mixture distributions in this manner. We make no claims of optimality; indeed such intermediate distributions may be computationally impractical for use in an SMC sampler. Other approaches to constructing a distribution schedule may not allow $\sigma^2_{\mathbb{1}}$ to be made arbitrarily small (see e.g. Remark 4.7).

A consequence of this result is that in perfectly-mixing settings, the minimal value of $\sigma^2_{\mathbb{1}}$ over all distribution schedules of length n is strictly decreasing as a function of n:

Proposition 4.18. *For any $n \geq 2$,*

$$\min_{\pi_1, \ldots, \pi_{n-1}} \sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p) \;>\; \min_{\pi'_1, \ldots, \pi'_n} \sum_{p=0}^{n} D_{\chi^2}(\pi'_{p+1} \,\|\, \pi'_p),$$

where all distributions are defined on some common space X, with $\pi_0 = \pi'_0$ and $\pi_n = \pi'_{n+1} = \pi_\star \neq \pi_0$.

Proof. Consider the optimal distribution schedule of length n, in the sense of minimising the sum of chi-squared distances between consecutive distributions. We may extend this to a schedule of length $n+1$ by inserting a new distribution between two existing consecutive distributions. In the sum of chi-squared distances this has the effect of replacing the corresponding summand with two new terms, with all other summands unaffected.

Suppose we insert the new intermediate distribution between two existing consecutive distributions that are distinct; since $\pi_0 \neq \pi_\star$ at least one such pair exists. By Proposition 4.17, it is always possible to choose the new distribution so that the total of the two new chi-squared distances is strictly less than the summand it replaces.

The sum of chi-squared distances for the original schedule of length n is therefore strictly greater than that of the new schedule of length $n+1$; this in turn must be greater than or equal to that of the *optimal* distribution of length $n+1$. ∎

This formalises the notion that increasing n allows consecutive distributions to be more

'similar', resulting in lower variances of the SMC incremental weights and of such estimators as $\gamma_n^N(\mathbb{1})$.

4.4.1. Chained chi-squared distances

Returning to the problem of minimising the sum of chi-squared distances described in Proposition 4.1, a similar problem has previously been considered for another f-divergence, albeit in a rather different context. Motivated by applications to Wright–Fisher processes in genetics, Pavlichin and Weissman (2016) consider sums of Kullback–Leibler divergences: denoting by $D_{\mathrm{KL}}(\pi \,\|\, \mu)$ the Kullback–Leibler divergence of π from μ, the authors define the 'n-fold chained Kullback–Leibler divergence' of $\pi_\star$ from π_0 as

$$D_{\mathrm{KL}}^{(n)}(\pi_\star \,\|\, \pi_0) := \min_{\pi_1,\ldots,\pi_{n-1}} \sum_{p=0}^{n-1} D_{\mathrm{KL}}(\pi_{p+1} \,\|\, \pi_p), \tag{4.21}$$

where $\pi_n = \pi_\star$. That is, over all sequences of distributions $(\pi_p)_{p=0}^n$ interpolating between π_0 and $\pi_\star$, one takes the minimal value of the sum of Kullback–Leibler divergences between consecutive distributions. There are clear similarities between this and the problem we have considered in this chapter; generalising the nomenclature, for fixed n one might describe the minimal value of $\sigma_{\mathbb{1}}^2$ over all distribution schedules as the 'n-fold chained chi-squared distance' of $\pi_\star$ from π_0.

Pavlichin and Weissman (2016) derive a number of properties of this dissimilarity measure, and of the optimal path of interpolating distributions, in the case that all these distributions are defined on a finite space X. A number of these results use only those properties of the Kullback–Leibler divergence that are common to all f-divergences, and therefore also apply to chi-squared distances. For example, one may show that when considered as a function of π_n and π_0, the minimal value of $\sigma_{\mathbb{1}}^2$ is jointly convex. In settings where the set of probability measures on X is closed (with respect to an appropriate topology), this implies that the optimal distribution schedule of length n exists and is unique (cf. Pavlichin and Weissman, 2016, Theorem 1.1).

For a finite state space X, the authors also derive a form for the optimal path of interpolating distributions for general n, which in turn allows the study of the asymptotic behaviour of (4.21) in n. To achieve this one first considers the case $n = 2$; using the method of Lagrange multipliers one can derive an expression for the distribution π_1 that minimises

$$D_{\mathrm{KL}}(\pi_1 \,\|\, \pi_0) + D_{\mathrm{KL}}(\pi_2 \,\|\, \pi_1).$$

Unfortunately, such an approach cannot be directly applied to chi-squared distances. Again considering a finite state space X, one could apply Lagrange multipliers to minimise

$$D_{\chi^2}(\pi_1 \,\|\, \pi_0) + D_{\chi^2}(\pi_2 \,\|\, \pi_1) = \sum_{x \in \mathsf{X}} \left[\frac{\pi_1(\{x\})^2}{\pi_0(\{x\})} + \frac{\pi_2(\{x\})^2}{\pi_1(\{x\})} \right] - 2$$

under the constraint that the sum of $\pi_1(\{x\})$ over $x \in \mathsf{X}$ is 1. One finds that the resulting solutions satisfy the cubic equation

$$\frac{2}{\pi_0(\{x\})}\pi_1(\{x\})^3 + \lambda\pi_1(\{x\})^2 - \pi_2(\{x\})^2 = 0.$$

While a closed-form solution does exist for fixed λ (and may be obtained using computer algebra software), the resulting expression is unwieldy; indeed, there is no closed form for λ, which must be chosen so that the resulting solutions satisfy the required constraint.

There may nonetheless be scope to use similar methods to derive properties of the optimal schedule of distributions in perfectly-mixing settings, and of their asymptotic behaviour in n. Such results would be useful in determining the optimal schedule length in the sense of minimising $n\sigma^2_{\mathbb{1}}$, and could lead to the development of heuristics for choosing the spacing of intermediate distributions in practical settings.

4.5. Summary

The theoretical results and heuristics presented in this chapter may be practically useful; for example, in settings where the Markov kernels M_p mix well, analysis of this perfectly-mixing setting may provide a useful approximation of the behaviour of the SMC sampler. As discussed however, in many settings of practical interest, the Markov kernels may mix poorly. In the following chapter, we shall consider the schedule selection problem in this more general setting, describing work towards more widely-applicable procedures to selecting distribution and resampling schedules.

5. Approaches to schedule selection for general Markov kernels

5.1. Numerical optimisations for normal distributions

We now return to the more general schedule selection problem, considering settings in which the Markov kernels $(M_p)_{p=1}^n$ may not be perfectly-mixing. This chapter explores the properties and behaviour of relevant quantities, as well as investigating some practical approaches to schedule selection in such general settings.

We continue to view the issue of schedule selection in terms of the optimisation problem described in Section 3.3. Recall from (3.34) and Proposition 3.13 that for a distribution schedule $\pi_{0:n}$ and resampling schedule $R_n \coloneqq \{k_j : j \in \{1, \dots, r_n\}\}$, the relative asymptotic variance of the normalising constant estimator is given by

$$\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n) = \sum_{j=0}^{r_n} D_{\chi^2}(\hat{\eta}'_{r_n} M'_{r_n,j} \,\|\, \eta'_j), \tag{5.1}$$

where the sequence of excursion Feynman–Kac models is defined with respect to the resampling schedule. As previously stated, we assume that there is some fixed method of constructing the Markov kernels M_p to leave each π_p invariant.

In order to explore how the mixing properties of the kernels affect this quantity, we first consider a simple method of constructing imperfectly-mixing Markov kernels. Within this setting we shall investigate a model that allows closed-form expressions for each term in the decomposition (5.1) to be derived, which we shall in turn use to conduct numerical optimisations.

To facilitate comparison with the results in Chapter 4, and to assist in the interpretation of the terms in the asymptotic variance decomposition, in this section we shall consider the setting in which resampling takes place in every iteration. From (3.31) and Proposition 3.10, in this case the decomposition (5.1) may be expressed as

$$\sigma^2_{\mathbb{1}}(\pi_{0:n}, \{1, \dots, n\}) = \sum_{p=0}^{n} D_{\chi^2}(\pi_n M_{n,p} \,\|\, \pi_p), \tag{5.2}$$

and the schedule selection problem reduces to that of choosing a distribution schedule. We emphasise however that the numerical optimisation results in this section could be extended, by also optimising over the space of resampling schedules.

Specifically, we shall consider Markov kernels that may be expressed as, for $p \in \{1, \dots, n\}$,

$$M_p(x, \cdot) = \epsilon_p \pi_p(\cdot) + (1 - \epsilon_p)\delta_x(\cdot) \tag{5.3}$$

for all $x \in \mathsf{X}$, where $\epsilon_p \in [0, 1]$. That is, application of M_p at $x \in \mathsf{X}$ results in a sample from π_p with probability ϵ_p, and returns the same value x with probability $1 - \epsilon_p$; it is straightforward to show that this kernel leaves π_p invariant. We here see some similarities with Metropolis–Hastings MCMC kernels, in which a proposed value is accepted with some probability, else the current value is returned (a brief summary is later given in Section 6.1). To this end, the value ϵ_p may be seen as a proxy for the mixing quality of M_p.

Now consider the probability measures $\pi_n M_{n,p}$ in (5.2). Recalling from (3.12) that $M_{n,n} = \mathrm{Id}$ we find that $\pi_n M_{n,n} = \pi_n$, so that the nth term in the asymptotic variance decomposition (5.2) vanishes, in line with Remark 3.9. For $p < n$, $M_{n,p}$ is defined in (3.12) in terms of the time reversal kernels M_p^*, themselves given by Definition 3.2. For M_p as given in (5.3), these may be shown to have the simple form $M_p^* = M_p$.

The consequence is that for this choice of Markov kernels M_p the probability measures $\pi_n M_{n,p}$ in (5.2) admit convenient expressions. To build an intuition for the forms of these measures, and for the effects of the 'mixing quality parameters' ϵ_p on the relative asymptotic variance $\sigma^2_{\mathbb{1}}$, it is instructive to consider first the two extreme cases:

- **If $\epsilon_p = 1$ for all p**, one has for each p that $M_p(x, \cdot) = \pi_p(x)$ for all $x \in \mathsf{X}$, and so each M_p exhibits perfect mixing, corresponding to the setting of Chapter 4. As has previously been shown in Proposition 4.1, in this case $\pi_n M_{n,p} = \pi_{p+1}$ for $p < n$, and the relative asymptotic variance (5.2) is given by

$$\sum_{p=0}^{n-1} D_{\chi^2}(\pi_{p+1} \,\|\, \pi_p).$$

- **If $\epsilon_p = 0$ for all p**, one has for each p that $M_p(x, \cdot) = \delta_x(\cdot)$ for all $x \in \mathsf{X}$, and so each M_p exhibits no mixing. In this case $\pi_n M_{n,p} = \pi_n$ for all p, and so the relative asymptotic variance (5.2) is given by

$$\sum_{p=0}^{n-1} D_{\chi^2}(\pi_n \,\|\, \pi_p).$$

- **In the general case**, we may derive a recursive expression for $\pi_n M_{n,p}$ by noting that $M_{n,p} = M_{n,p+1} M_{p+1}^*$ for $p < n$; we thereby obtain

$$\pi_n M_{n,p} = \epsilon_{p+1} \pi_{p+1} + (1 - \epsilon_{p+1}) \pi_n M_{n,p+1}$$

  for $p \in \{0, \dots, n-1\}$. It follows that for $p < n$ each $\pi_n M_{n,p}$ can be expressed explicitly as a mixture of the distributions $(\pi_q)_{q=p+1}^n$. This represents a generalisation of the

two extreme cases above. In particular, if the ϵ_p values are closer to 1 (representing good mixing) then this mixture distribution places higher weight on the 'earlier' distributions near π_{p+1}; if the ϵ_p values are closer to 0 (representing poor mixing) then higher weight is placed on the 'later' distributions near π_n.

When these Markov kernels are used, a setting in which the terms in (5.2) admit closed-form expressions is that in which $(\pi_p)_{p=0}^n$ are all chosen as normal distributions. In this case, each such term corresponds to the chi-squared distance of a mixture of normal distributions from a single such distribution. The resulting expressions are unwieldy but tractable, being a weighted sum of terms resembling the chi-squared distance (4.7) between two normal distributions.

In the results that follow we consider a target distribution $\pi_\star = \mathcal{N}(\mu_\star, \Sigma_\star)$, an initial distribution $\pi_0 = \mathcal{N}(\mu_0, \Sigma_0)$, and a distribution schedule $\pi_{0:n}$ specified by a temperature schedule $\beta_{0:n}$ (as described in Section 2.2). To reflect the common scenario in which Markov kernels corresponding to higher inverse temperatures mix more poorly, we assume that the values ϵ_p appearing in the Markov kernels (5.3) are determined according to some decreasing function of the inverse temperatures β_p. The corresponding problem of optimising (5.2), considered as a function of the temperature schedule, does not in general admit a closed-form solution; we here present results obtained from optimising this quantity numerically.

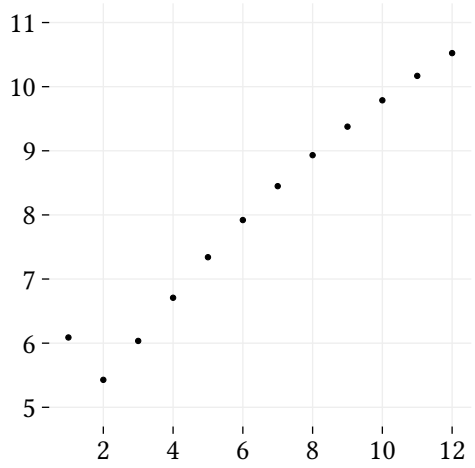
Figure 5.1 presents such results in the setting that $\pi_0 = \mathcal{N}(0, 100)$, $\pi_\star = \mathcal{N}(0, 1)$, and $\epsilon_p = 1 - 0.9\beta_p$ for $p \in \{1, \dots, n\}$. For various values of n the optimal temperature schedule $\beta_{0:n}$, in the sense of minimising $n\sigma_{\mathbb{1}}^2$, was found using a numerical procedure; for $n \in \{1, \dots, 12\}$ the corresponding minimal values of this quantity are displayed in Figure 5.1a. Here, the minimal value of $n\sigma_{\mathbb{1}}^2$ was obtained when $n = 2$, using temperature schedule $\beta_{0:n} = (0,\ 0.0764,\ 1)$.

For comparison, we present in Figure 5.1b the optimal temperatures schedules of length $n = 6$ and $n = 12$. We see that the former is not obtained by 'thinning' the latter, i.e. by taking every other value: for example, β_5 in the optimal sequence with length 6 is not equal to β_{10} in the optimal sequence of length 12. It follows that in this imperfectly-mixing setting, the optimal sequence $\beta_{0:n}$ of length n can *not* be expressed as

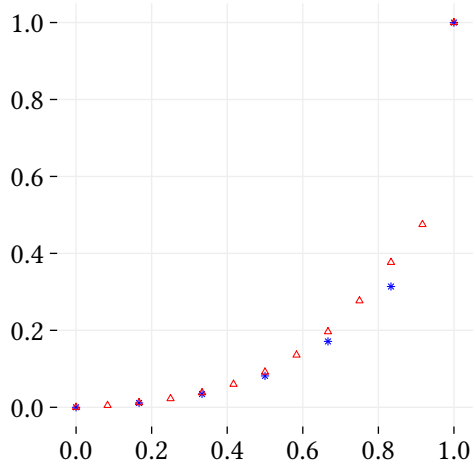
$$\beta_p = f(p/n), \quad p \in \{0, \dots, n\}, \tag{5.4}$$

where $f : [0, 1] \longrightarrow [0, 1]$ is some function that does not depend on n. This stands in contrast to the perfectly-mixing setting, for the same choices of π_0 and $\pi_\star$: by Remark 4.14 the optimal schedule in that setting is given by (4.11), which is indeed of the form (5.4).

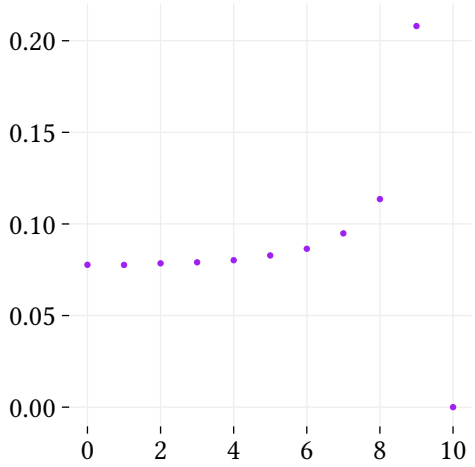
We note that several authors have described methods for temperature schedule selection via the (possibly implicit) specification of such a 'temperature map' f as in (5.4). For example, Heng et al. (2015) consider partial differential equations relating to an associated



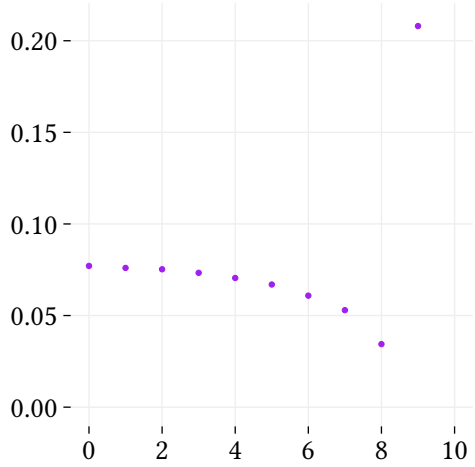
(a) min $n\sigma_{\mathbb{1}}^2$ against n, where the minimum in each case is taken over all temperature schedules $\beta_{0:n}$ of length n.



(b) The optimal temperature schedules $\beta_{0:n}$ for $n = 6$ (blue stars) and $n = 12$ (red triangles), plotted as β_p against p/n for $p \in \{0, \dots, n\}$.



(c) For the optimal temperature schedule of length $n = 10$, the terms $v_{p,n}(\mathbb{1})$ in the decomposition of $\sigma_{\mathbb{1}}^2$, plotted against p for $p \in \{0, \dots, n\}$.



(d) For the optimal temperature schedule of length $n = 10$, the chi-squared distances $D_{\chi^2}(\pi_{p+1} \| \pi_p)$ between consecutive distributions, plotted against p for $p \in \{0, \dots, n-1\}$.

Figure 5.1.: Results from the numerical optimisation of $n\sigma_{\mathbb{1}}^2$ by selection of a temperature schedule $\beta_{0:n}$, for various values of n. Here, $\pi_0 = \mathcal{N}(0, 100)$, $\pi_\star = \mathcal{N}(0, 1)$, and the Markov kernels are of the form (5.3), with $\epsilon_{1:n}$ determined by $\epsilon_p = 1 - 0.9\beta_p$, $p \in \{1, \dots, n\}$.

flow transport problem; Zhou et al. (2016) also use this formulation within the description of their adaptive schedule selection method. While there is merit in selecting a temperature schedule via the choice of such a function, we see that there is no single function in general that will generate the optimal schedule of every length n.

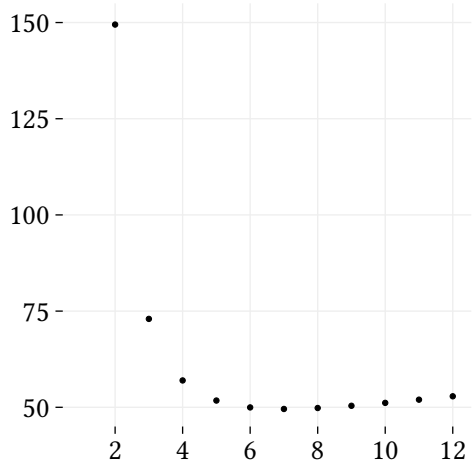
A further comparison with the perfectly-mixing setting may be made by considering the values of the terms $v_{p,n}(\mathbb{1})$ in the decomposition of $\sigma^2_{\mathbb{1}}$. Recall from Proposition 4.12 that for these choices of π_0 and $\pi_\star$, when perfectly-mixing Markov kernels are used one should choose the intermediate distributions to be equally-spaced in terms of the chi-squared distance, so that for $p < n$ all the terms $v_{p,n}(\mathbb{1})$ are equal. We see from Figure 5.1c, which presents these terms for the optimal distribution schedule of length $n = 10$, that this is not the case in this imperfectly-mixing setting. Instead, it is optimal for terms corresponding to larger values of p to be larger.

Since in this case the terms $v_{p,n}(\mathbb{1})$ are not equal to the chi-squared distances between consecutive distributions, we present these separately in Figure 5.1d. These follow a rather different pattern, gradually decreasing up until the last pair of distributions, which are separated by a much greater chi-squared distance. The intuition is that as the mixing of the Markov kernels worsens, it becomes harder to move the particles to the areas of high mass of the following distribution. To counteract this, consecutive distributions are chosen to be more similar. Eventually the mixing becomes so poor that adding additional intermediate distributions offers no benefit over moving directly to $\pi_\star$.

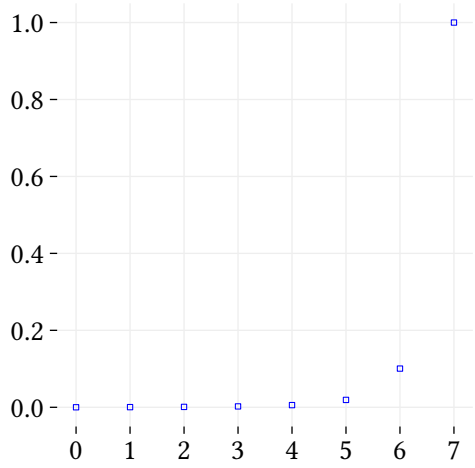
The patterns exhibited here are nontrivial, and indeed may be different for different models. Figure 5.2 presents similar results for a setting in which the initial distribution is $\pi_0 = \mathcal{N}(100, 1000)$, but is otherwise unchanged from the setting of Figure 5.1. Here, the optimal pattern of chi-squared distances between successive distributions (presented in Figure 5.2d), while similarly requiring a somewhat larger separation between the final two distributions, is qualitatively rather different to that in the former setting.

For further comparison we also present results for the same choices of $\pi_0 = \mathcal{N}(100, 1000)$ and $\pi_\star = \mathcal{N}(0, 1)$, but with the mixing parameters $\epsilon_{1:n}$ determined from the temperature schedule by $\epsilon_p = \exp(-30\beta_p)$ for $p \in \{1, \dots, n\}$. Here the mixing properties worsen more rapidly with the inverse temperature, which may better reflect the mixing in many realistic settings (e.g. the bimodal example of Figure 2.1). We see in Figure 5.3b that the optimal schedule $\beta_{0:n}$ over all values of n, here obtained when $n = 5$, places all its intermediate inverse temperatures very close to 0 (the penultimate value is $\beta_4 = 0.0159$). We also see in Figure 5.3d that the optimal pattern of chi-squared distances between successive distributions (here for $n = 10$) exhibits a very wide range of values; for example, here the value of $D_{\chi^2}(\pi_{10} \,\|\, \pi_9)$ is over 116 times larger than $D_{\chi^2}(\pi_9 \,\|\, \pi_8)$.

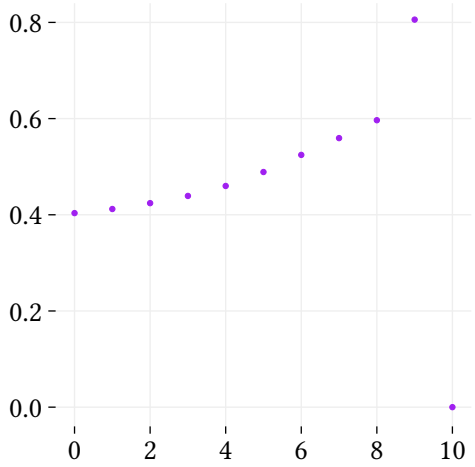
If one were to choose a distribution schedule solely by consideration of the relative and/or absolute sizes of the $v_{p,n}(\mathbb{1})$ terms and/or chi-squared distances, it is not clear what the optimal strategy would be. The adaptive procedure of Zhou et al. (2016) aims



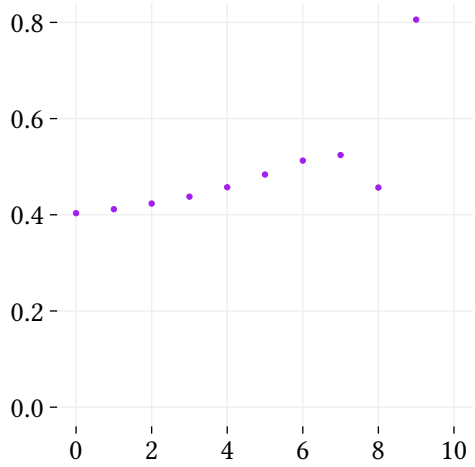
(a) $\min n\sigma^2_{\mathbb{1}}$ against n, where the minimum in each case is taken over all temperature schedules $\beta_{0:n}$ of length n.



(b) The optimal temperature schedule $\beta_{0:n}$ for $n = 7$, plotted as β_p against p for $p \in \{0, \dots, n\}$.

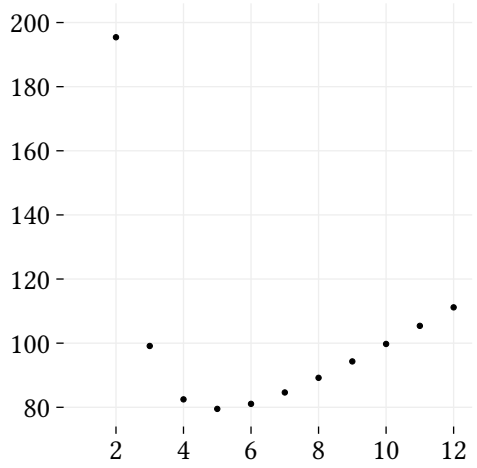


(c) For the optimal temperature schedule of length $n = 10$, the terms $v_{p,n}(\mathbb{1})$ in the decomposition of $\sigma^2_{\mathbb{1}}$, plotted against p for $p \in \{0, \dots, n\}$.

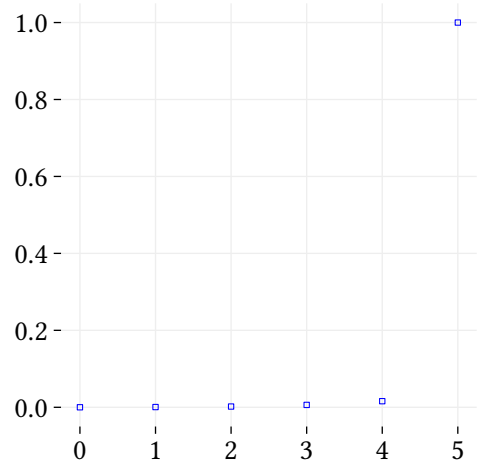


(d) For the optimal temperature schedule of length $n = 10$, the chi-squared distances $D_{\chi^2}(\pi_{p+1} \| \pi_p)$ between consecutive distributions, plotted against p for $p \in \{0, \dots, n-1\}$.

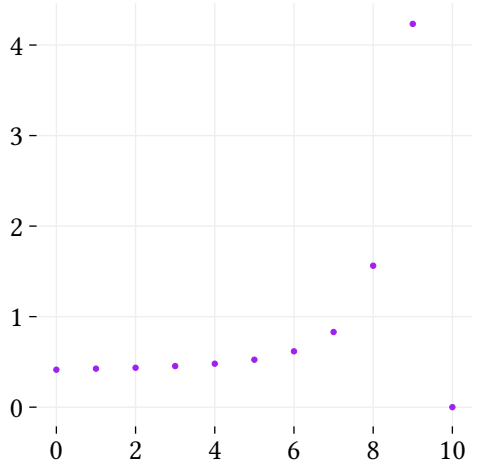
Figure 5.2.: Results from the numerical optimisation of $n\sigma^2_{\mathbb{1}}$ by selection of a temperature schedule $\beta_{0:n}$, for various values of n. Here, $\pi_0 = \mathcal{N}(100, 1000)$, $\pi_\star = \mathcal{N}(0, 1)$, and the Markov kernels are of the form (5.3), with $\epsilon_{1:n}$ determined by $\epsilon_p = 1 - 0.9\beta_p$, $p \in \{1, \dots, n\}$.



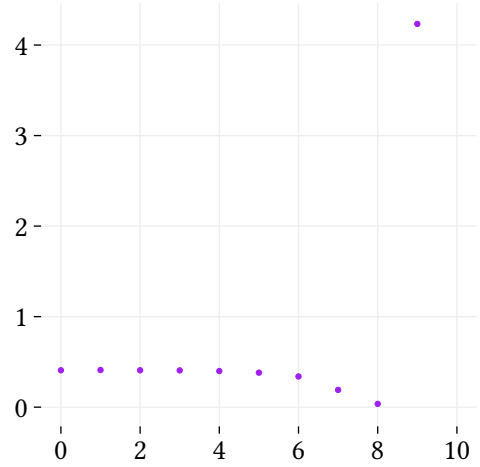
(a) $\min n\sigma^2_{\mathbb{1}}$ against n, where the minimum in each case is taken over all temperature schedules $\beta_{0:n}$ of length n.



(b) The optimal temperature schedule $\beta_{0:n}$ for $n = 5$, plotted as β_p against p for $p \in \{0, \dots, n\}$.



(c) For the optimal temperature schedule of length $n = 10$, the terms $v_{p,n}(\mathbb{1})$ in the decomposition of $\sigma^2_{\mathbb{1}}$, plotted against p for $p \in \{0, \dots, n\}$.



(d) For the optimal temperature schedule of length $n = 10$, the chi-squared distances $D_{\chi^2}(\pi_{p+1} \| \pi_p)$ between consecutive distributions, plotted against p for $p \in \{0, \dots, n-1\}$.

Figure 5.3.: Results from the numerical optimisation of $n\sigma^2_{\mathbb{1}}$ by selection of a temperature schedule $\beta_{0:n}$, for various values of n. Here, $\pi_0 = \mathcal{N}(100, 1000)$, $\pi_\star = \mathcal{N}(0, 1)$, and the Markov kernels are of the form (5.3), with $\epsilon_{1:n}$ determined by $\epsilon_p = \exp(-30\beta_p)$, $p \in \{1, \dots, n\}$.

to construct a distribution schedule in which the chi-squared distances between consecutive distributions are all equal, but in this imperfectly-mixing setting this may be highly suboptimal (and even with perfect mixing this is not the optimal choice for all models, as discussed in Section 4.4). Although the simplicity of adaptive procedures is to their benefit, this motivates the investigation of alternative approaches to schedule selection, some of which we shall proceed to investigate.

5.2. Procedures using variance estimators

Since the quantity that we look to minimise depends on the relative asymptotic variance $\sigma^2_{\mathbb{1}}$ of the normalising constant estimator $\gamma_n^N(\mathbb{1})$, we may consider approaches to schedule selection that utilise the variance estimation techniques introduced in Section 1.5. As previously detailed, these allow consistent estimators of quantities including $\sigma^2_{\mathbb{1}}$ to be generated as a by-product of running an SMC sampler.

Within this section we explore several possible schedule selection procedures implementing these variance estimators. We do not explore any adaptive or online procedures here: this is partly because of the difficulties in constructing suitable approaches as previously explained, but also because estimating $\sigma^2_{\mathbb{1}}$ for any given schedule necessarily requires a complete realisation of the corresponding SMC sampler. Instead, we consider approaches that compare schedules by running multiple SMC samplers, using different schedules in each case.

Such an approach may be computationally more expensive than an adaptive algorithm. However, one can consider a setting in which such a procedure used in a 'pilot run' using relatively few particles, to determine a distribution schedule and resampling schedule to be used in some final run with many more particles. This is particularly applicable given that $n\sigma^2_{\mathbb{1}}$, the quantity with which we compare schedules, is independent of the number of particles N. Such an idea can be compared with the use of adaptive SMC procedures in pilot runs, as described at the end of Section 3.2.2.

5.2.1. Building a schedule by addition of intermediate distributions

An initial idea is to attempt to construct a distribution schedule 'from scratch'. That is to say, beginning with a schedule comprising only the initial distribution π_0 and final distribution $\pi_\star$, one could build a schedule by inserting intermediate distributions one by one, until an appropriate sequence is obtained. Since any valid distribution schedule could be constructed this way, such an approach could potentially be made fully general.

A basic framework for such an algorithm might include a scheme for proposing the insertion of a new intermediate distribution, with this insertion accepted if it results in a lower (estimated) value of $n\sigma^2_{\mathbb{1}}$, and rejected otherwise. The final temperature schedule

Algorithm 5.1 Temperature schedule building algorithm (basic structure)

1. Initialise $n \leftarrow 1$, temperature schedule $\boldsymbol{\beta} := \beta_{0:n} \leftarrow (0, 1)$, resampling schedule $R \leftarrow \{1\}$, and indexing variable $p \leftarrow 0$.
2. Run an SMC sampler for these temperature/resampling schedules, and store an estimate of $n\sigma^2_{\mathbb{1}}$.
3. While $p < n$:
   a) Propose a new temperature schedule $\boldsymbol{\beta}'$ by adding a new inverse temperature β^{NEW} after β_p, such that $\beta_p < \beta^{\text{NEW}} < \beta_{p+1}$.
   b) Run two SMC samplers for this temperature schedule, one with resampling taking place in the newly-added iteration, and one without such resampling (all other iterations using resampling as before). In each case, compute an estimate of $n\sigma^2_{\mathbb{1}}$.
   c) Compare these estimates of $n\sigma^2_{\mathbb{1}}$ with that corresponding to the existing temperature schedule $\boldsymbol{\beta}$ and resampling schedule R; retain the schedules for which this estimate is lowest. That is:
      - If the proposal of adding β^{NEW} (with or without associated resampling) results in a lower estimated $n\sigma^2_{\mathbb{1}}$, set $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}'$, $n \leftarrow n + 1$, and update R appropriately.
      - Else, either return to Step 3a to propose another new temperature between β_p and β_{p+1}, repeating for some fixed maximum number of attempts; or set $p \leftarrow p + 1$.
4. Return the temperature schedule $\boldsymbol{\beta}$ and resampling schedule R.

could then be returned once some termination criterion is satisfied: for example, after the insertion of additional distributions has been rejected multiple times.

Several such schemes were investigated in the context of temperature schedule selection; the basic structure of the algorithms explored is presented in Algorithm 5.1. The idea of this approach, which essentially follows a recursive pattern, comes from the observation that any temperature schedule $\beta_{0:n}$ may be viewed as partitioning the interval $[0, 1]$ into n subintervals. Beginning with an initial schedule containing only 0 and 1, one proposes the addition of a new inverse temperature between these values. If this is accepted to be inserted into the sequence, then this divides the interval $[0, 1]$ into two subintervals; the process of proposing a new temperature can then be repeated for each of these subintervals. If instead the addition of a new inverse temperature in the interval is rejected (or perhaps if multiple proposals are rejected), then no further proposals are made for insertions of new inverse temperatures in this interval. The algorithm terminates once it has been decided to insert no new inverse temperatures in any of the existing subintervals.

One advantage of this structure is that when proposing new inverse temperatures, the

values of the existing inverse temperatures are taken into account. For example, if at any point the schedule primarily contains values near 0, it may be advantageous to consider inserting new values within this region, rather than attempting to distribute them more uniformly across the interval $[0, 1]$. This is supported by the results of the previous section, in which many of the optimal schedules placed almost all of their values very close to 0.

Rather than concurrently specifying a resampling schedule, this approach could be used to specify only a distribution schedule, for example by using resampling in every iteration when estimating $n\sigma_{\mathbb{1}}^2$ for each schedule. We instead consider a simple (if possibly suboptimal) approach to specifying a resampling schedule, in which a fixed resampling status (i.e. the use or non-use of resampling) is associated with the iteration corresponding to each inverse temperature. When proposing to insert a new inverse temperature, one estimates $n\sigma_{\mathbb{1}}^2$ both with and without resampling taking place in the newly added iteration, the resampling statuses of iterations corresponding to other inverse temperatures being unchanged), considering whichever choice results in the lower estimate.

Many instances of this framework were investigated, with a number of different approaches explored for proposing a new inverse temperature within a given subinterval. However, even on simple examples there were frequent problems with the algorithm failing to terminate. Because the insertion of a new inverse temperature into a subinterval has the effect of splitting it into two, this causes the total number of subintervals to be investigated to increase by one, and so it is quite possible for the procedure never to terminate.

The primary cause of these issues is the variance associated with estimators of $n\sigma_{\mathbb{1}}^2$. In our investigations, we estimated the relative asymptotic variance $\sigma_{\mathbb{1}}^2$ associated with a given schedule by running the associated SMC sampler and evaluating $NV_n^N(\mathbb{1})$, where V_n^N is as defined in (1.40). As mentioned in Section 1.5, this estimator of $\sigma_{\mathbb{1}}^2$ is consistent in N; however for any fixed N, this variance estimator itself has a non-zero variance. In practice this can be large relative to the value of $\sigma_{\mathbb{1}}^2$ it estimates, making it difficult to compare schedules based on these values and confidently determine which has the lower *true* value of $\sigma_{\mathbb{1}}^2$.

To provide an example, consider the simple univariate setting in which $\pi_0 = \mathcal{N}(0, 10^2)$ and $\pi_\star = 0.3\mathcal{N}(-20, 0.4^2) + 0.7\mathcal{N}(20, 0.8^2)$. Suppose one considers inserting the additional inverse temperature 0.01 into the temperature schedule (0, 0.04, 0.16, 0.36, 0.64, 1). Figure 5.4 shows a boxplot of the estimated values of $n\sigma_{\mathbb{1}}^2$ resulting from running an SMC sampler 100 times with each schedule (using $N = 2^{10}$ particles, and always resampling). We see that the variance of these distributions is relatively large compared to their mean; a single observation of each value may give a very misleading impression of the relative sizes of $n\sigma_{\mathbb{1}}^2$. The practical effect is that in cases such as this, procedures based on Algorithm 5.1 often returned very different temperature schedules when run multiple times. One could reduce the variance of these estimators simply by increasing the number of particles N

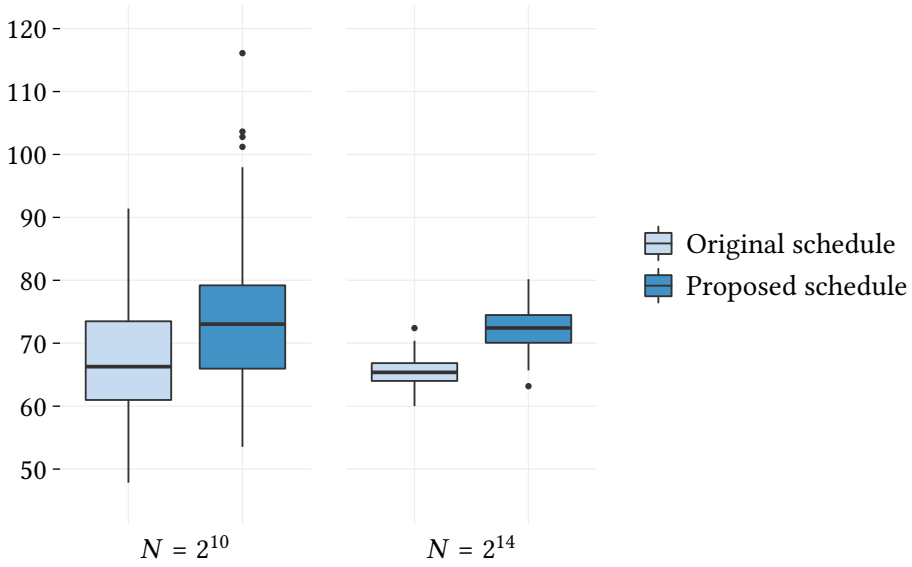


Figure 5.4.: Box plots of values of $nNV_n^N(\mathbb{1})$, which forms an estimator of $n\sigma_{\mathbb{1}}^2$, resulting from 100 realisations of an SMC sampler. Here, $\pi_0 = \mathcal{N}(0, 10^2)$, $\pi_\star = 0.3\mathcal{N}(-20, 0.4^2) + 0.7\mathcal{N}(20, 0.8^2)$, and two temperature schedules are considered: an 'original schedule' given by (0, 0.04, 0.16, 0.36, 0.64, 1), and a 'proposed schedule' comprising the same values with an additional inverse temperature of 0.01. Results are shown for SMC samplers using resampling in every iteration, with $N = 2^{10}$ particles and separately with $N = 2^{14}$ particles.

in the SMC samplers with which they are evaluated, as seen in the results using $N = 2^{14}$ particles in Figure 5.4. However, this would have the effect of significantly increasing the already high computational cost of such an approach.

With this in mind, computationally cheaper approximations of the procedure described in Algorithm 5.1 were also investigated. For example, we considered whether instead of estimating the relative asymptotic variance of $\gamma_n^N(\mathbb{1})$ for each schedule, one could consider that of $\gamma_p^N(\mathbb{1})$ for various values of $p \leq n$, essentially running an SMC sampler for 'as long as necessary' to make some comparison. These procedures suffered the same problems when run at comparable computational cost. Attempts were also made to make Algorithm 5.1 more conservative when assessing the proposed insertion of a new inverse temperature, for example by introducing an additional threshold on the resulting improvement in the estimated value of $n\sigma_{\mathbb{1}}^2$, though these did not prevent the issues with non-termination.

In a small number of cases, the procedures that were investigated terminated with no new intermediate inverse temperatures being added, since all those proposed were rejected. A possible cause for this is the difficulty in estimating asymptotic variances such as $\sigma_{\mathbb{1}}^2$ when these are very high. Consider the variance estimator $V_n^N(\mathbb{1})$, which when multiplied by N gives a consistent estimator of $\sigma_{\mathbb{1}}^2$. From the definition (1.40) of V_n^N, we find that

$$1 - \left(\frac{N}{N-1}\right)^n \leq V_n^N(\mathbb{1}) \leq 1;$$

these lower and upper bounds are attained when the particles' zeroth-generation ancestors are, respectively, all distinct and all equal.

For fixed N, the consistent estimator $nNV_n^N(\mathbb{1})$ of $n\sigma_{\mathbb{1}}^2$ is therefore bounded above by nN. The result is that when comparing two poorly-performing schedules that both result in a high degree of particle degeneracy when used in an SMC sampler, it is common for the resulting value of $nNV_n^N(\mathbb{1})$ to be smaller for the shorter sequence, purely by virtue of n being smaller. This may the case even when the *true* value of $n\sigma_{\mathbb{1}}^2$ corresponding to the longer schedule is, though still high, rather lower than that of the shorter schedule.

Observing such behaviour in practice may impart some useful information; for example, it might suggest that the areas of high mass of π_0 do not coincide well with those of $\pi_\star$, and therefore that a different choice of initial distribution should be considered. However, it poses obvious problems for a procedure that is initialised using a schedule containing no intermediate distributions, which would generally be expected to perform poorly.

5.2.2. Refining a schedule by removal of intermediate distributions

As an alternative to the previous approach of building a schedule from scratch, we now consider refining an existing schedule by the removal of intermediate distributions. Consider for example the adaptive schedule selection procedure of Zhou et al. (2016), previously introduced in Section 3.2.2. Within this procedure, the approximate chi-squared

distance between successive distributions is controlled by choosing the conditional effective sample size (CESS), defined in (3.4), to be equal to a tuning parameter CESS*. This may result in a long sequence of closely-spaced distributions; we consider how such a sequence might be improved by removing some of these, and reducing the value of $n\sigma^2_{\mathbb{1}}$. This may be useful in light of the results of Section 5.1, in which it was seen that the optimal schedules are typically not equally spaced in the sense of the chi-squared distance.

A motivation for this approach is the behaviour of the terms in the decomposition of $\sigma^2_{\mathbb{1}}$. Consider first these terms $v_{p,n}(\mathbb{1})$ in the case that resampling takes place in every iteration, so that each such term may be expressed as $D_{\chi^2}(\pi_n M_{n,p} \,\|\, \pi_p)$, following Proposition 3.10. We see that this depends only on those distributions π_q for which $q \in \{p, \dots, n\}$, either directly or via Markov kernels leaving these distributions invariant. The effect is that altering any distribution π_q, where $q \in \{0, \dots, n\}$, only affects those decomposition terms $v_{p,n}(\mathbb{1})$ for which $p \leq q$.

Similarly, consider removing π_q from the schedule, making no other changes. The decomposition of $\sigma^2_{\mathbb{1}}$ for this new schedule will have one term fewer than that for the original schedule, but the final $n - q$ terms will be unchanged. Similar statements can be made for the terms $v_{p,n}(\varphi)$ in the decompositions of more general relative asymptotic variances, as well as for the decomposition terms resulting from the use of occasional resampling.

This suggests the following procedure. Given an initial schedule $(\pi_p)_{p=0}^n$, propose the removal of π_{n-1}; that is, the last intermediate distribution. If the proposed new schedule results in a lower value of $n\sigma^2_{\mathbb{1}}$ than the original schedule, then accept this removal; otherwise, retain this inverse temperature. Now move one step 'back' through the schedule, proposing the removal of the previous intermediate distribution (i.e. the original π_{n-2}) and accepting/rejecting this by comparing the resulting values of $n\sigma^2_{\mathbb{1}}$. Continue in this manner, working 'backwards' through the distribution schedule until the removal of each intermediate distribution has been considered, returning the resulting 'thinned' schedule.

The reasoning for this approach is twofold. Firstly, as seen in the empirical results of Section 5.1, it is often preferable for distributions near the end of the schedule to be more spaced more widely, and so it may be beneficial to first consider the removal of distributions within this region. Secondly, once it has been determined to retain a given distribution π_q in the schedule, the removal of any earlier distribution will not have any effect on the final $n - q$ terms in the decomposition of $\sigma^2_{\mathbb{1}}$. The idea is that choosing to retain π_q is an indication that the contribution of these later terms is 'sufficiently low'; we may therefore 'bank' these low values and proceed to consider the removal of earlier distributions, which will not affect these later terms.

As previously, such a procedure could be executed in practice by using estimated values of each $\sigma^2_{\mathbb{1}}$, for example by computing $NV_n^N(\mathbb{1})$ or $\sum_{p=0}^n v_{p,n}^N(\mathbb{1})$, in the latter case using the term-by-term estimators proposed by Lee and Whiteley (2018, Section 4.2). Compared to the procedure described in Section 5.2.1, a benefit of this approach is that it must terminate

after $n - 1$ iterations (i.e. once each of the $n - 1$ intermediate distributions has been considered for removal). In practice however, we found that the same issues of high variance prevented the construction of a robust procedure, and the returned distribution schedules varied considerably between simulations.

For example, consider comparing a distribution schedule with some proposed schedule which is identical except for the removal of some π_q. In the decomposition of $\sigma^2_{\mathbb{1}}$ relating to each schedule, the final $n - q$ terms are common; however, the estimated values of these terms (e.g. the observed values of $v^N_{p,n}(\mathbb{1})$) may differ. This is a simple consequence of the need to run an SMC sampler in full to estimate $\sigma^2_{\mathbb{1}}$ for each schedule. To reduce these problems, some approaches were investigated for reducing the random variation between these realisations. For example, one could run an SMC sampler for the first $q - 1$ iterations (over which the two schedules are identical), and then continue from this point for the remainder of each of the two schedules separately, so that the realisations of the first $q - 1$ iterations are shared. Although this had the other benefit of reducing the computational cost of the procedure, in turn allowing a greater number of particles to be used for the same overall computational budget, this did not result in improved robustness.

5.2.3. Simulated annealing using reversible jump MCMC

Given the aim of minimising $n\sigma^2_{\mathbb{1}}$ over all distribution and resampling schedules, an alternative approach would be to use a form of simulated annealing, previously introduced in Section 2.3.4. As earlier described, this is a probabilistic optimisation technique for determining the argument that maximises a continuous bounded function $f : \mathsf{E} \longrightarrow \mathbb{R}$. Given an initial value $x \in \mathsf{E}$, one applies a sequence of Markov kernels K_i such that for $i > 0$, K_i leaves invariant the distribution with density at x proportional to $\exp(\alpha_i f(x))$. As previously mentioned the values α_i may be viewed as inverse temperatures; one requires that α_i diverges to ∞ as $i \longrightarrow \infty$.

Here we aim to construct Markov kernels K_i, $i > 0$, such that K_i leaves invariant the distribution with density at $(\pi_{0:n}, R_n)$ proportional to

$$\exp\left[-\alpha_i \cdot n\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n)\right]. \tag{5.5}$$

The space E over which we optimise is the space of all distribution schedules $\pi_{0:n}$ (for fixed π_0 and $\pi_n := \pi_\star$), for all possible lengths n, together with all corresponding resampling schedules $R_n \subseteq \{1, \ldots, n\}$. In practice, one might restrict consideration to intermediate distributions belonging to some parametric family (e.g. those determined by a temperature schedule).

In simulated annealing it is common for these Markov kernels to be constructed as MCMC kernels. However, implementation of such kernels typically requires repeated evaluations of the unnormalised target density. Here, this would require the ability to exactly

compute (5.5), and therefore $n\sigma^2_{\mathbb{1}}(\pi_{0:n}, R_n)$, for each distribution schedule and resampling schedule. Instead, within the construction of the MCMC kernel we consider replacing evaluations of $\sigma^2_{\mathbb{1}}$ with realisations of a consistent estimator, using one of the approaches introduced in Section 1.5. For example, one could compute $N V_n^N(\mathbb{1})$, where V_n^N is as defined in (1.40), for the corresponding choice of distribution and resampling schedules.

The resulting Markov kernels are not guaranteed to have the correct invariant distribution. Indeed, since the resulting estimate of (5.5) is not unbiased, one cannot use this to construct a pseudo-marginal MCMC kernel (we later provide a review of such methods, proposed by Andrieu and Roberts, 2009, in Section 6.2.1). However, for a heuristic procedure that is not necessarily intended to produce the optimal schedule (rather, an acceptable one), this may be sufficient. Furthermore, simulated annealing schemes using biased estimates have been shown to enjoy similar convergence properties in certain contexts (e.g. Rubenthaler et al., 2009, who consider a different setting in which the estimation error reduces with time at an appropriate rate).

The problem of selecting an optimal distribution schedule and resampling schedule is transdimensional in nature. Considering $(\pi_{0:n}, R_n)$ as a parameter vector, the dimension of the optimal parameter is unknown, since we consider schedules of all possible lengths n. A convenient framework for describing MCMC methods as applied to such transdimensional problems is that of *reversible jump Markov chain Monte Carlo* (RJMCMC), introduced by Green (1995); a practical guide to its implementation is provided by Green and Hastie (2009). Denoting by Θ_n the set of all possible $(\pi_{0:n}, R_n)$ for a given value of n, the space of all possible distribution and resampling schedules is $\bigcup_{n=1}^{\infty} \Theta_n$. The RJMCMC framework may be used to construct a Markov chain on a space given by $\bigcup_{n=1}^{\infty} (\{n\} \times \Theta_n)$.

We investigated settings in which the distribution schedule is determined by a temperature schedule, as in Section 2.2. To propose new distribution/resampling schedules we considered a number of possible moves, randomly choosing one of the following in each iteration:

- Adding a new inverse temperature between a consecutive pair of existing inverse temperatures β_p and β_{p+1} (selected uniformly at random from all existing pairs), drawing this from the uniform distribution on (β_p, β_{p+1});

- Removing an existing inverse temperature (selected uniformly at random from all those strictly between 0 and 1);

- Changing whether resampling takes place in an intermediate iteration (selected uniformly at random);

- Splitting an existing inverse temperature β_p (selected uniformly at random from all those strictly between 0 and 1) into two new inverse temperatures; i.e. removing β_p

while adding two new inverse temperatures drawn independently from the uniform distribution on $(\beta_{p-1}, \beta_{p+1})$;

- Merging two existing consecutive inverse temperatures β_p and β_{p+1} (selected uniformly at random from all existing pairs); i.e. removing β_p and β_{p+1} while adding one new inverse temperature drawn from the uniform distribution on (β_p, β_{p+1});

- Jittering all existing inverse temperatures by adding IID normal random variates to each, on a logit scale.

We ran a number of investigative simulations to assess the behaviour of simulated annealing in this context. To initialise the Markov chain in each case we ran an SMC algorithm once, adaptively determining a resampling schedule using the ESS-based procedure of Liu and Chen (1995), and a distribution schedule using the CESS-based procedure of Zhou et al. (2016). In the case of the latter, we considered various values of the parameter CESS*, used to determine the approximate chi-squared distance between successive distributions as described in Section 3.2.2. One aim of these experiments was to determine the extent to which the schedules chosen by these adaptive procedures can be improved upon, with respect to minimising $n\sigma_{\mathbb{1}}^2$.

We present in Table 5.1 results for two simple univariate settings, in which the initial distribution π_0 is a broad normal distribution, and the distribution of interest $\pi_\star$ has two well-separated modes. These settings may be seen as comparable to the illustrative example previously presented in Figure 2.1. For these experiments we ran the simulated annealing algorithm for 2000 iterations with annealing schedule $\alpha_i = 1.005^{i-1}$. In order to compute $V_n^N(\mathbb{1})$ for each proposed schedule (as used in the estimation of $n\sigma_{\mathbb{1}}^2$), we ran an SMC sampler with $N = 2^{11}$ particles. The values of $n\sigma_{\mathbb{1}}^2$ presented in Table 5.1 are estimates that were generated subsequently, using the same approach but with 2^{14} particles.

These results paint a mixed picture. In the first example, presented in Table 5.1a, the simulated annealing procedure succeeds in improving on the schedules regarding by the adaptive procedures, and is robust to the choice of initial schedule. In particular, we see that for large choices of the CESS* parameter used to specify a distribution schedule in the procedure of Zhou et al. (2016), the resulting schedules can be much longer than is optimal. The schedules found to be optimal by the simulated annealing procedure were far shorter (with n between 2 and 4), and resulted in rather lower values of $n\sigma_{\mathbb{1}}^2$.

The results for the second example, shown in Table 5.1b, display rather less robustness. While the simulated annealing procedure still tends to prefer shorter schedules, the values of $n\sigma_{\mathbb{1}}^2$ associated with these exhibit very large variances. Once again, this appears to be due to the high variance of the estimator $V_n^N(\mathbb{1})$: when comparing two schedules with very small n the variances of the corresponding estimates of $n\sigma_{\mathbb{1}}^2$ may be very large, so that one cannot determine with reasonable confidence which schedule has the lower *true* value of

	Mean ± standard deviation			
	Initial schedule		Final schedule	
CESS* used for initialisation	n	$n\sigma^2_{\mathbb{1}}$	n	$n\sigma^2_{\mathbb{1}}$
$0.99N$	51.9 ± 0.3	42.71 ± 0.90	2.1 ± 0.3	17.95 ± 0.55
$0.90N$	14.0 ± 0.0	32.94 ± 1.44	2.2 ± 0.4	18.41 ± 1.11
$0.70N$	7.0 ± 0.0	24.00 ± 0.94	2.6 ± 0.8	17.54 ± 0.85
$0.50N$	4.1 ± 0.3	21.14 ± 2.37	2.6 ± 0.7	17.17 ± 0.78

(a) Results for $\pi_0 = \mathcal{N}(0, 10^2)$, $\pi_\star = 0.3\mathcal{N}(-10, 0.1^2) + 0.7\mathcal{N}(10, 0.2^2)$.

	Mean ± standard deviation			
	Initial schedule		Final schedule	
CESS* used for initialisation	n	$n\sigma^2_{\mathbb{1}}$	n	$n\sigma^2_{\mathbb{1}}$
$0.99N$	71.4 ± 0.7	1549.3 ± 47.7	5.4 ± 3.5	1583.8 ± 2701.9
$0.90N$	24.3 ± 0.5	719.3 ± 44.6	4.3 ± 3.1	2831.1 ± 3044.3
$0.70N$	12.0 ± 0.0	540.3 ± 34.6	3.5 ± 2.7	1246.3 ± 1522.1
$0.50N$	8.0 ± 0.0	497.8 ± 31.1	3.3 ± 2.2	942.9 ± 1527.4

(b) Results for $\pi_0 = \mathcal{N}(0, 10^2)$, $\pi_\star = 0.9\mathcal{N}(-30, 0.1^2) + 0.1\mathcal{N}(30, 0.2^2)$.

Table 5.1.: Results of two simulation studies of the simulated annealing procedure. The values presented correspond to the initial distribution and resampling schedules, as obtained using the CESS-based procedure of Zhou et al. (2016) with various values of the tuning parameter CESS*, and the final schedules following the simulated annealing procedure. Presented are the sample mean ± standard deviation of n, and of $n\sigma^2_{\mathbb{1}}$ (estimated as described in the main text), computed over 10 replicates in each case.

$n\sigma^2_{\mathbb{1}}$. The resulting RJMCMC algorithm may therefore behave erratically[1]. Furthermore the boundedness of $V_n^N(\mathbb{1})$ for fixed N, also as previously discussed in Section 5.2.1, results in a tendency for the procedure to prefer shorter sequences in cases where the associated asymptotic variances are high.

While it may be possible to improve on the specific algorithm that has been investigated here, for example by considering other move types, these variance issues inhibit the use of RJMCMC in this way to construct a robust schedule selection procedure. Furthermore the computational cost of this approach, requiring a very large number of realisations of SMC algorithms (one in each iteration, in order to compute $V_n^N(\mathbb{1})$ for the proposed schedule), means that such a construction is unlikely to be of use in practical settings.

[1] As previously mentioned, the Markov kernels used in this procedure cannot be viewed as pseudo-marginal kernels in the sense of Andrieu and Roberts (2009), since they do not employ unbiased estimators of the target densities. However, to understand the role of the estimator variance in this setting it may be instructive to consider the literature on tuning pseudo-marginal kernels, in particular the effect of estimator variance on their mixing properties. We later provide a brief review of this, albeit in a different context, in Section 7.2.2.

While this could be alleviated by reducing the number of particles used within these SMC realisations, this would have the effect of further increasing the variance of the estimators $V_n^N(\mathbb{1})$.

5.3. Related problems

Although we have here focused on the problem of specifying a distribution schedule and resampling schedule, there are several related problems in the tuning of SMC samplers for which variance estimators may be useful. While we do not investigate these directly in this thesis, we detail some of these issues here, to place them into the wider context and describe possible procedures that could be investigated.

When defining the quantity $\sigma_{\mathbb{1}}^2$ in Section 3.3, we noted that since this depends on the sequence of Markov kernels $(M_p)_{p=1}^n$, we must assume some fixed approach to constructing these to leave the corresponding distributions $(\pi_p)_{p=1}^n$ invariant. While this characterisation is useful, this obscures the more general problem of tuning Markov kernels in order to ensure convenient mixing properties. For example:

- If one chooses each M_p to be an MCMC kernel targeting π_p, how should the proposal distributions be chosen?
- If each M_p is to comprise multiple applications of an MCMC kernel targeting π_p, how many iterations should be used in each case?

The former concern is pertinent to the schedule selection problem since in the construction of an MCMC kernel, the optimal choice of proposal kernel depends directly on the target distribution. If successive distributions in the schedule are reasonably similar, so that $\pi_{p-1} \approx \pi_p$ for $p \in \{1, \ldots, n\}$, it is often useful to use the particle approximation η_{p-1}^N of π_{p-1} as the basis for choosing the proposal kernel for M_p. For example, consider choosing the covariance matrix for a Gaussian random walk Metropolis kernel; assuming the covariance matrix of π_{p-1} approximates that of π_p, one could use the matrix associated with η_{p-1}^N, scaled appropriately (e.g. using the results of Roberts and Rosenthal, 2001).

Since such a sequence of Markov kernels is determined directly by the distribution schedule, in this manner the ideas investigated in this chapter to be extended directly to the simultaneous selection of a sequence of proposal kernels. In other settings however, such a construction may be highly suboptimal, requiring the consideration of other approaches. For example, consider an SMC sampler targeting a distribution with well-separated modes; as the intermediate distributions begin also to possess this property, proposal kernels constructed in the described manner will generally become less efficient, with poor mixing properties. To this end one may wish to compare multiple approaches to constructing the sequence of Markov kernels $(M_p)_{p=1}^n$, for which one could consider a procedure based on variance estimation.

The most general approach would involve the selection a schedule of Markov kernels to minimise $n\sigma_{\mathbb{1}}^2$, rather than simply choosing a distribution schedule, although the resulting optimisation problem would be incredibly complex. Instead, one might consider using estimated values of $n\sigma_{\mathbb{1}}^2$ to compare sequences of Markov kernels for a fixed choice of distribution schedule, selecting the sequence that results in the lowest value.

The second issue described above, that of determining how many MCMC kernel iterations each Markov kernel should comprise, has a more direct connection to the issue of schedule selection. Suppose we have an intermediate distribution π, and that it is difficult to construct a well-mixing MCMC kernel leaving π invariant. To improve mixing one might consider using multiple applications of such a kernel, and to this end there are two possible choices for the construction of an SMC sampler:

- A distribution schedule including π only once, with the corresponding Markov kernel comprising k iterations of a π-invariant MCMC kernel;
- A distribution schedule that is identical, except that this single instance of π is replaced by k consecutive repeated instances of π, with the corresponding Markov kernels each comprising one single iteration of a π-invariant MCMC kernel.

When is each of these choices to be preferred?

If one chooses (deterministically) for resampling not to take place during any of the iterations corresponding to these replicated distributions, then these two choices are equivalent in practice. To this end, the first of these two options may be seen as a special case of the second. It follows that from a theoretical perspective, choosing each Markov kernel to comprise a single MCMC kernel (repeating distributions as necessary) may be preferable, since it permits greater flexibility in when the choice of resampling times. Of course, a distribution schedule containing multiple copies of the same distribution may itself not be optimal, when compared to other schedules of the same length; it may be better to instead have distributions that are closely spaced but distinct.

Nonetheless, there may be settings in which it may be preferable for each Markov kernel to comprise multiple applications of an MCMC kernel. Consider the use of adaptive resampling based on the ESS, as described in Section 1.3.2. The (repeated) application of an MCMC kernel to each particle may be conducted in parallel across all particles; in contrast, the computation of the ESS requires the values of all particles to be collected. In settings where this 'collection' or communication process is inefficient (for example, if the algorithm is executed in a distributed manner and there is some latency involved in communicating their values between machines), it may be advantageous to restrict the number of possible resampling steps.

While this may be seen theoretically in terms of choosing an appropriate distribution and resampling schedule as discussed, for the purposes of implementation it may be most convenient to view this in terms of the numbers of MCMC kernel applications in each

iteration. To this end, different choices of these numbers could be compared by considering the corresponding value of $\sigma^2_{\mathbb{1}}$, multiplied by some proxy for the total computational cost (analogous to the use of $n\sigma^2_{\mathbb{1}}$ for distribution schedule selection). Again, this tuning could be achieved in practice using estimators such as $NV_n^N(\mathbb{1})$, either to compare different choices for a fixed distribution schedule, or to determine these tuning parameters concurrently with the selection of the distribution schedule.

5.4. Summary

Within this chapter we have considered the properties of optimal distribution schedules in settings where the Markov kernels do not mix perfectly, and investigations towards procedures for schedule selection in such settings, based on the variance estimators of Lee and Whiteley (2018). Unfortunately, these investigations have not resulted in a procedure that is robust, while having an acceptable computational cost. While the procedures described could be implemented using SMC samplers employing many more particles, thus alleviating many of the issues relating to the variances of the SMC variance estimators, this would preclude their use for schedule selection in practical settings.

Nonetheless, it is hoped that the findings of these initial investigations may inform and motivate the future development of techniques for the tuning of SMC algorithms using these variance estimators, for example in those areas discussed in Section 5.3. To this end the discussion of Section 3.3, and the proposal to use $n\sigma^2_{\mathbb{1}}$ to compare distribution/resampling schedules, may find application for the more general purpose of comparing tuning choices. We provide further discussion of possible research directions in the concluding remarks at the end of the thesis.

Part III.

A Monte Carlo framework for distributed settings

6. Markov chain Monte Carlo and big data

6.1. Markov chain Monte Carlo

The remainder of this thesis considers problems in approximating Bayesian posteriors, when these depend on large data sets that are distributed across several computers. In this chapter we summarise the difficulties in constructing efficient simulation procedures for these settings, and review several approaches that have been proposed in the literature. To motivate these discussions, we begin with a short review of a class of simulation methods that includes many algorithms commonly applied to Bayesian inference problems.

As first considered in Chapter 1, suppose we wish to approximate some probability measure π, defined on a measurable space $(\mathsf{E}, \mathcal{E})$. If we are able to draw IID samples from this distribution, then a simple Monte Carlo approximation of π can be formed as (1.1). However, in most settings of practical interest it is computationally infeasible to draw independent samples distributed according to π.

An alternative approach is to construct a Markov kernel K on $(\mathsf{E}, \mathcal{E})$ that leaves π invariant; that is, such that $\pi K = \pi$. Under certain conditions, the simulation of a homogeneous Markov chain by recursive application of K results in a sequence of random variables that are dependent, but each approximately distributed according to π. These samples may then be used to construct a Monte Carlo approximation of π. This idea forms the basis of a range of algorithms known as *Markov chain Monte Carlo* (MCMC) methods.

Specifically, consider the discrete time Markov chain defined by

$$Z^0 \sim \pi, \qquad Z^i \sim K(Z^{i-1}, \cdot), \ i > 0.$$

Since K leaves π invariant, it follows that each Z^i is marginally distributed according to π. This suggests that π may be well approximated by the empirical measure

$$\pi^N := \frac{1}{N} \sum_{i=1}^{N} \delta_{Z^i}, \tag{6.1}$$

for some N, analogously to (1.1) in which IID samples are used. For an $\mathcal{X}$-measurable function φ, an estimate of the integral $\pi(\varphi)$ is thereby obtained as

$$\pi^N(\varphi) = \frac{1}{N} \sum_{i=1}^{N} \varphi(Z^i).$$

In practice however, the kernel K may be such that consecutive states are highly correlated, so that the resulting chain mixes poorly. The values $(Z^i)_{i=1}^N$ may behave very differently to IID samples from π (for example, they may not belong to all areas of non-negligible mass), so that (6.1) forms a poor approximation of π. Furthermore, it is generally not possible to sample the initial state from π, and so Z^0 is drawn from some other distribution μ. It is therefore not guaranteed that any Z^i is exactly marginally distributed according to π; depending on the kernel K, the resulting chain may not even have π as its limiting distribution.

Nonetheless, under certain conditions on K we may show that estimators of the form (6.1) have similar asymptotic properties to those based on IID samples. For example, if the resulting Markov chain is Harris recurrent, estimators of the form $\pi^N(\varphi)$ obey a strong law of large numbers. A number of CLT results for such estimators also exist, holding under varying conditions on the Markov chain. A summary of various such results is provided by Roberts and Rosenthal (2004); a fuller theoretical review is provided by Meyn and Tweedie (2009).

A number of methods have been proposed for constructing the Markov kernel K to leave π, the *target distribution*, invariant. For purposes of exposition, we here present a simple method that forms the basis of many more recently-proposed algorithms.

The *Metropolis–Hastings algorithm* was first proposed in the chemical physics literature by Metropolis et al. (1953), and was later generalised by Hastings (1970). This procedure, detailed in Algorithm 6.1, requires that π admits an (unnormalised) density $\bar{\pi}$ with respect to some dominating measure that may be evaluated at each $z \in \mathsf{E}$. One also requires a kernel $Q : \mathsf{E} \times \mathcal{E} \longrightarrow [0, 1]$ known as the *proposal kernel*, chosen such that one may readily sample from $Q(z, \cdot)$ for all $z \in \mathsf{E}$; we assume that this admits a density $q(z, \cdot)$ with respect to the same dominating measure.

Algorithm 6.1 Metropolis–Hastings algorithm

1. Set initial state Z^0.
2. For $i = 1, \ldots, N$,
   - Set $Z \longleftarrow Z^{i-1}$ and sample $Z' \sim Q(Z, \cdot)$.
   - Set
   $$r(Z, Z') \longleftarrow \frac{\bar{\pi}(Z')}{\bar{\pi}(Z)} \frac{q(Z', Z)}{q(Z, Z')}. \tag{6.2}$$
   - Sample U according to the uniform distribution on $(0, 1)$.
   - If $U < r(Z, Z')$, set $Z^i \longleftarrow Z'$. Else, set $Z^i \longleftarrow Z$.

Given the current value Z of the chain, a new value Z' is proposed by application of Q. With some probability, the proposed value Z' is accepted as the next value of the chain;

else it is rejected, and the chain's current value Z is carried forward. It is straightforward to show that the Markov kernel defined by this procedure is reversible with respect to π, and therefore leaves π invariant (see e.g. Roberts and Rosenthal, 2004, Proposition 2).

Since Algorithm 6.1 only requires evaluations of an unnormalised density $\bar{\pi}$, the use of Metropolis–Hastings and related MCMC methods is widespread in Bayesian inference, for the purpose of approximating posterior distributions. Beyond the simplest models (e.g. those employing conjugate prior distributions), these typically have complex forms that make the exact evaluation of integrals $\pi(\varphi)$ computationally intractable, and so these are often approximated using Monte Carlo methods.

Specifically, suppose π is a Bayesian posterior distribution for the statistical parameter Z, which takes values on $z \in \mathsf{E} \subseteq \mathbb{R}^d$. By abuse of notation, henceforth let π also denote the density of this distribution with respect to some version of the Lebesgue measure. Denote the corresponding prior density by μ, and the likelihood function by $L(\cdot \,|\, \mathbf{y})$, where $\mathbf{y}$ represents the observed data. Then we have

$$\pi(z) \;=\; \frac{\mu(z)L(z \,|\, \mathbf{y})}{\int_{\mathsf{E}} \mu(z')L(z' \,|\, \mathbf{y}) \,\mathrm{d}z'} \;\propto\; \mu(z)L(z \,|\, \mathbf{y}) \;=\; \bar{\pi}(z), \tag{6.3}$$

and so construction of a Metropolis–Hastings kernel leaving π invariant does not require evaluation of the marginal likelihood $\int_{\mathsf{E}} \mu(z)L(z \,|\, \mathbf{y}) \,\mathrm{d}z$, which is typically computationally intensive.

6.2. MCMC methods for big data

We now continue to consider this Bayesian setting, in the case that the data set $\mathbf{y}$ is very large. Consider the case in which the data comprise n observations $y_{1:n}$ that are conditionally independent given Z. Then from (6.3) we have

$$\pi(z) \;\propto\; \mu(z) \prod_{i=1}^{n} L_i(z \,|\, y_i), \tag{6.4}$$

where $L_i(\cdot \,|\, y_i)$ is the likelihood contribution of the observation y_i.

Within Algorithm 6.1, the quantity $r(Z, Z')$ defined in (6.2) is known as the *acceptance ratio*. We see that each iteration of Algorithm 6.1 requires evaluation of $r(Z, Z')$, and therefore of $\bar{\pi}$ at the proposed value Z'. Each iteration therefore requires evaluation of the likelihood contribution of each of the n observations, to compute just one bit of information: whether to accept or reject Z' as the new value of the chain.

Suppose one aims to estimate integrals $\pi(\varphi)$, for appropriate $\mathcal{E}$-measurable functions φ. If unlimited time were available, one could simply run a Metropolis–Hastings sampler targeting π for as long as necessary; given one of the aforementioned CLT results the resulting estimators are consistent, and so the error of the realised estimates may be made

arbitrarily small.

In practice however, the available computation time is limited. When n is very large the number of iterations of Algorithm 6.1 that can be completed may be relatively few, since the required acceptance ratios are expensive to evaluate. Various authors have therefore suggested alternative MCMC approaches to forming an empirical approximation π^N of π in this setting, which may be advantageous compared to this 'direct' Metropolis–Hastings approach.

To motivate many of these methods, we may consider the *mean squared error* (MSE) of the resulting estimator $\pi^N(\varphi)$ of an integral $\pi(\varphi)$; that is,

$$\mathbb{E}\Big[\big(\pi^N(\varphi) - \pi(\varphi)\big)^2\Big] = \mathbb{E}\big[\big(\pi^N(\varphi) - \pi(\varphi)\big)\big]^2 + \operatorname{var}\big[\pi^N(\varphi)\big]. \tag{6.5}$$

An alternative method may generate estimators $\pi^N(\varphi)$ that are asymptotically biased in N, for example because the Markov chain formed does not target π, but some approximation thereof. Alternatively, the estimators may have a higher *asymptotic* variance than those obtained from a straightforward Metropolis–Hastings approach; that is, a higher value of $\lim_{N\to\infty} N \operatorname{var}[\pi^N(\varphi)]$. However, for a fixed time budget it may be possible to obtain estimators of a lower mean squared error than would be possible using the simpler approach. For example it may be possible to draw a greater number of samples N, so that even if the resulting estimators exhibit a slightly increased bias, their (non-asymptotic) variance is greatly reduced.

We proceed to summarise a number of such approaches that have been proposed in the literature; a fuller review of such methods is provided by Bardenet et al. (2017). A number of approaches are particularly applicable in settings where the data $y_{1:n}$ are stored on multiple computers, so that any MCMC method requires some degree of communication between these machines. We shall detail these separately in Section 6.3.

6.2.1. Pseudo-marginal MCMC

A natural idea is to construct an MCMC algorithm in which one computes only a fraction of the n likelihood contributions $L_i(z \mid y_i)$ in each iteration. If a fixed subsample of these likelihood contributions are used, then the resulting target posterior may be very different from π, as defined in (6.4). However, by using a random subsample in each iteration it is possible to construct a Markov chain with π as its invariant distribution.

To this end, it is useful to consider MCMC algorithms in which rather than computing the unnormalised density $\bar{\pi}(z)$ exactly, one computes some computationally cheap estimate of this quantity. Indeed we may show that within such methods as Algorithm 6.1, directly replacing these unnormalised density evaluations by unbiased estimates does not change the invariant distribution of the resulting Markov chain. The resulting framework is known as *pseudo-marginal MCMC*. The generic form of this class of algorithms was

described by Andrieu and Roberts (2009), following earlier work on 'noisy' Monte Carlo methods by various authors (Kennedy and Kuti, 1985; Lin et al., 2000; Beaumont, 2003).

We formalise this framework as follows. Consider the latent variable (or collection thereof) used in the formation of the unbiased estimator of the unnormalised target density $\bar{\pi}(z)$; we shall denote this by Ξ, where this takes values $\xi \in \mathsf{X}$. Suppose that for any $z \in \mathsf{E}$ we may sample Ξ according to some probability measure ρ_z, and that for some function $\tilde{\pi} : \mathsf{E} \times \mathsf{X} \longrightarrow \mathbb{R}$, the random variable $\tilde{\pi}(z, \Xi)$ is unbiased as an estimator of $\bar{\pi}(z)$. That is,

$$\int_{\mathsf{X}} \tilde{\pi}(z, \xi)\rho_z(\mathrm{d}\xi) = \bar{\pi}(z) \propto \pi(z).$$

Using this notation, we present as Algorithm 6.2 the resulting pseudo-marginal form of the Metropolis–Hastings algorithm.

Algorithm 6.2 Pseudo-marginal Metropolis–Hastings algorithm

1. Set initial state (Z^0, Ξ^0).
2. For $i = 1, \ldots, N$,
   - Set $Z \leftarrow Z^{i-1}$ and $\Xi \leftarrow \Xi^{i-1}$.
   - Sample $Z' \sim Q(Z, \cdot)$ and $\Xi' \sim \rho_{Z'}(\cdot)$.
   - Set
   $$r((Z, \Xi); (Z', \Xi')) \leftarrow \frac{\tilde{\pi}(Z', \Xi')}{\bar{\pi}(Z, \Xi)} \frac{q(Z', Z)}{q(Z, Z')}. \tag{6.6}$$
   - Sample U according to the uniform distribution on $(0, 1)$.
   - If $U < r((Z, \Xi); (Z', \Xi'))$, set $Z^i \leftarrow Z'$ and $\Xi^i \leftarrow \Xi'$. Else, set $Z^i \leftarrow Z$ and $\Xi^i \leftarrow \Xi$.

The acceptance ratio (6.6) used in this algorithm takes essentially the same form as the corresponding quantity (6.2) in Algorithm 6.1, but with each evaluation of $\bar{\pi}$ replaced by its unbiased estimator. Formally, we may view Algorithm 6.2 as a Metropolis–Hastings sampler on the extended state space $\mathsf{E} \times \mathsf{X}$. Considering the resulting invariant distribution, one may show that its marginal distribution on E is exactly equal to π, the distribution of interest. The values $(Z^i)_{i=1}^N$ may therefore be used to form an empirical measure (6.1) approximating π.

Returning to the big data problem, several authors have proposed pseudo-marginal approaches to sampling from π, estimating $\bar{\pi}(z)$ by computing only a subsample of the likelihood contributions. While it is straightforward to generate an unbiased estimator of the *log-likelihood* in this context, additional techniques are required to build an unbiased estimator of the likelihood, and therefore of the unnormalised target density. Bardenet et al. (2017, Section 4.2) describe one such approach, though note that the resulting Markov chain may mix poorly in practice; Quiroz et al. (2019) propose a similar technique within

this framework.

The 'firefly Monte Carlo' approach of Maclaurin and Adams (2014), which requires a positive lower bound on the likelihood contribution of each datum, similarly takes the form of an MCMC algorithm on an extended state space. As shown by Bardenet et al. (2017, Section 4.3) this may also take a pseudo-marginal form.

We shall further discuss pseudo-marginal kernels in Section 7.2.2, where we discuss the tuning of such algorithms within the context of our proposed simulation framework.

6.2.2. Other approaches

Outside of the pseudo-marginal framework, a number of other MCMC approaches have been proposed to generate approximations of the full posterior π. Many of these approaches are 'asymptotically inexact', resulting in a Markov chain with an invariant distribution that is not exactly π, but is intended to be a close approximation.

Korattikara et al. (2014) propose that in each iteration of Algorithm 6.1, a small random subsample of the data is used to approximate the acceptance ratio (6.2). Based on this approximation, a hypothesis test is conducted to determine whether there is sufficient evidence that the proposed Z' would be accepted (or rejected) based on the *true* acceptance ratio. If so, acceptance or rejection takes place accordingly, else the acceptance ratio is re-approximated with a larger subsample and the test is reconducted. Motivated by this method Bardenet et al. (2014) propose a similar adaptive subsampling approach, in which confidence intervals for the logarithm of the true acceptance ratio are computed according to a concentration inequality.

Among other approaches, Huggins et al. (2016) consider using a *fixed* weighted subset of the data, which may be used to construct an approximate posterior with an unnormalised density that may be evaluated cheaply. A number of procedures applying stochastic optimisation techniques have also been proposed for this problem (Welling and Teh, 2011; Hoffman et al., 2013)

6.3. MCMC methods for distributed data

Remaining in the Bayesian setting, we now assume that the likelihood function can be expressed as a product of b terms, each of which depends on data stored on a single machine. That is, we assume the posterior density for the statistical parameter Z satisfies

$$\pi(z) \propto \mu(z) \prod_{j=1}^{b} f_j(z). \tag{6.7}$$

We assume that f_j is computable on computing node j and involves consideration of $\mathbf{y}_j$, the jth subset or 'block' of the full data set, which comprises b such blocks. This setting is

common in cases where the data set is very large, since for example it may not fit into the memory of a single machine.

Suppose we wish to evaluate (6.7) for some $z \in \mathsf{E}$. This requires the value of z to be communicated to each of the b nodes, on which the corresponding partial likelihood $f_j(z)$ is computed; these values must then be communicated back to the first machine, in order for the product to be taken. The consequence is that each evaluation of the target density on some central node, and therefore each iteration of Algorithm 6.1 performed on that node, requires the communication of values to and from each of the b worker nodes.

Although the partial likelihood terms $f_j(z)$ may be computed in parallel, this 'message passing' poses a problem in settings where the time budget is limited, due to the issue of *latency*: the delay or lag associated with communication between machines. When this latency is relatively high compared to the time taken to evaluate each partial likelihood term, one may spend a relatively large proportion of time communicating values between machines, rather than on likelihood computation. This 'wasteful' use of the available wall-clock time may mean that relatively few iterations of an MCMC sampler targeting π may be completed.

As previously discussed, a number of alternative approaches have therefore been proposed that may be advantageous in this context, allowing integrals $\pi(\varphi)$ to be estimated with a lower mean squared error. We detail some of these here.

6.3.1. Embarrassingly parallel algorithms

In order to reduce the amount of inter-node communication required, a number of authors have proposed methods that allow a separate MCMC chain to be generated on each of the b computing nodes. Rather than targeting the density of interest (6.7), each of these chains targets a density that depends only on the block of data $\mathbf{y}_j$ stored on that node. Since these densities may be computed locally, each of these MCMC chains may be executed without the need for communication with other nodes. Only after the generation of all b local chains are these values communicated to a central node, where they are used to generate an approximation of the full target density (6.7).

Algorithms of this form require communication between the nodes only at the very beginning and end of the procedure, and therefore fall into the MapReduce framework (Dean and Ghemawat, 2008). Owing to the simplicity with which each of the b individual chains can be run separately and concurrently, these algorithms are commonly known as *embarrassingly parallel* algorithms.

The target densities of each of the b local chains are commonly referred to as *subposterior densities*, being of the form of a posterior density, but each containing only the partial likelihood term $f_j(z)$ relating to the corresponding node. Accordingly, it is necessary to define an appropriate pseudo-prior density for each chain. A common approach is to assign each subposterior density an equal share of the prior information imparted by the true

prior density $\mu(z)$, by using a 'fractionated' prior density proportional to $\mu(z)^{1/b}$. The jth subposterior density is then proportional to

$$\mu(z)^{1/b} f_j(z); \tag{6.8}$$

we observe that by taking the product of this expression over $j \in \{1, \dots, b\}$, one recovers the full target density (6.7). We shall later discuss some properties and potential problems with this construction in Section 7.2.3, in the context of our proposed simulation framework.

Having generated a collection of samples from each of these subposterior densities, a final post-processing step is used to aggregate these samples, forming an approximation of the true target density π. A simple approach proposed by Scott et al. (2016), which has motivated several more recent techniques, takes the form of weighted averaging. After generating b MCMC chains of equal length N, which we shall denote $(Z_j^i)_{i=1}^N$ for $j \in \{1, \dots, b\}$, one averages these elementwise to compute a 'consensus chain' $(Z^i)_{i=1}^N$. Specifically, suppose the state space E is some subset of $\mathbb{R}^d$; for some choice of weight matrices $W_{1:b} \in \mathbb{R}^{d \times d}$, one takes

$$Z^i := \left(\sum_{j=1}^b W_j \right)^{-1} \sum_{j=1}^b W_j Z_j^i$$

for $i \in \{1, \dots, N\}$. It is intended that these values approximate a collection of N samples drawn from the true posterior density π, so that a weighted empirical measure approximating π may be formed according to (6.1). This embarrassingly parallel algorithm is known as *consensus Monte Carlo*.

Motivated by Bayesian asymptotics, the authors suggest taking each weight matrix W_j to be an estimate of the precision matrix of the corresponding subposterior distribution, computed using the samples from the jth chain. Indeed, if each subposterior density is Gaussian, this approach results in a consensus chain comprising samples asymptotically distributed according to π. We present this form of the procedure as Algorithm 6.3.

Algorithm 6.3 Consensus Monte Carlo

1. For $j = 1, \dots, b$, generate a chain of N samples $(Z_j^i)_{i=1}^N$ approximately distributed according to the jth subposterior density, i.e. the density proportional to (6.8).
2. Combine these b chains:
   - For $j = 1, \dots, b$, compute an estimate S_j of the covariance matrix of $(Z_j^i)_{i=1}^N$.
   - For $i = 1, \dots, N$, set
   $$Z^i \leftarrow \left(\sum_{j=1}^b S_j^{-1} \right)^{-1} \sum_{j=1}^b S_j^{-1} Z_j^i.$$

In cases where the subposterior distributions exhibit near-Gaussianity this performs well, with the final 'consensus chain' providing good approximations of posterior expec-

tations. However, there are no theoretical guarantees associated with this approach in settings in which the subposterior densities are poorly approximated by Gaussians. In such cases consensus Monte Carlo sometimes performs poorly in forming an approximation of the posterior π (as in examples of Wang et al., 2015; Srivastava et al., 2015; Dai et al., 2019), and so the resulting estimates of integrals $\pi(\varphi)$ exhibit high bias.

Various authors have therefore proposed more generally-applicable techniques for utilising the values from each of the b chains in order to approximate posterior expectations. For example, Neiswanger et al. (2014) propose a strategy based on kernel density estimation; based on this approach, Scott (2017) suggests a strategy based on finite mixture models, though notes that both methods may be impractical in high-dimensional settings. The same author proposes a model-agnostic method employing sequential importance sampling, which is also observed to perform poorly in high dimensions.

Other proposed approaches include combining the chains using random partition trees (Wang et al., 2015), choosing a function with which to aggregate the chains via variational optimisation (Rabinovich et al., 2015), and taking a suitably-defined average of empirical measures approximating each subposterior distribution (Minsker et al., 2014; Srivastava et al., 2015). A recent proposal of Dai et al. (2019) introduces a rejection sampler employing Brownian bridges; the authors do not directly address 'big data' settings, reserving this for future work.

As well as proposing an aggregation method for this framework based on rejection sampling, Wang and Dunson (2013) propose an alternative embarrassingly parallel algorithm that takes a slightly different approach. Rather than drawing samples directly from each of the subposterior distributions, one begins with an initial collection of N samples representing a rough empirical approximation of π. To each of these samples a refinement step is applied, which corresponds to a single iteration of an appropriately-defined Gibbs kernel on an extended state space. We explain this approach in greater detail in Section 7.2.3, comparing it to the simulation framework we shall later propose.

6.3.2. Other approaches

Outside of the embarrassingly parallel framework, an alternative approach to the problem of sampling from π has been proposed by Xu et al. (2014). Within their framework, separate MCMC chains are run in parallel on each node, targeting densities in which those partial likelihood terms $f_j(z)$ that cannot be computed on that node are approximated by density functions from an exponential family (up to normalising constants).

Rather than communication only occurring at the end of the algorithm, the sample moments of each chain's values are shared among the nodes regularly. This allows the parameters of the approximating densities to be iteratively updated via expectation propagation, in order that each chain's target density forms a close approximation of the true target π. Again, this method is most effective when the partial likelihood terms are well approxi-

mated by the surrogate terms (often chosen to be Gaussian, following Bayesian asymptotic arguments).

Finally, a recent proposal of Jordan et al. (2019), aims to further reduce the cost of communication resulting from embarrassingly parallel approaches. Rather than a separate MCMC chain being run on each node and communicated to a central node for aggregation, one only communicates the gradient of each subposterior density (at some initial point). These are used to construct an approximation of the posterior density; a single MCMC chain targeting this distribution may then be executed on the central node. This approach may therefore be most advantageous when there are limitations on communication bandwidth, in addition to high latency.

6.4. Summary

We have here reviewed the basis of Markov chain Monte Carlo, and algorithms within this framework that may be advantageous in Bayesian settings in which the data set is large and computational time is limited. The setting of Section 6.3, in which the data are distributed across multiple machines, shall form the focus of the following chapters. We shall proceed to propose a Metropolis-within-Gibbs algorithm for approximating (6.7) in this context, comparing with a number of the approaches and ideas that have been discussed here.

7. Global consensus Monte Carlo

7.1. The instrumental hierarchical model

Within this chapter we shall introduce a novel framework for inference in distributed settings; that is, in the setting described in Section 6.3, in which the distribution of interest π admits a density of the form (6.7). We describe the construction of an MCMC algorithm on an extended state space targeting a distribution that, when appropriately marginalised, provides an approximation of the density of interest π. For simplicity, throughout this chapter we shall occasionally abuse notation by using the same symbol for a probability measure and for its density with respect to a suitable form of the Lebesgue measure.

The framework we propose may be viewed in terms of an instrumental hierarchical model. Alongside the variable of interest Z, we introduce a collection of b instrumental variables each also defined on E, denoted by $X_{1:b}$. On the extended state space $\mathsf{E} \times \mathsf{E}^b$, we define the probability density function $\tilde{\pi}_\lambda$ by

$$\tilde{\pi}_\lambda(z, x_{1:b}) \propto \mu(z) \prod_{j=1}^{b} K_j^{(\lambda)}(z, x_j) f_j(x_j), \tag{7.1}$$

where for each $j \in \{1, \ldots, b\}$, $\{K_j^{(\lambda)} : \lambda \in \mathbb{R}_+\}$ is a family of Markov transition densities on E. Defining

$$f_j^{(\lambda)}(z) := \int_{\mathsf{E}} K_j^{(\lambda)}(z, x) f_j(x) \, \mathrm{d}x, \tag{7.2}$$

the density of the Z-marginal of $\tilde{\pi}_\lambda$ may be written as

$$\pi_\lambda(z) := \int_{\mathsf{E}^b} \tilde{\pi}_\lambda(z, x_{1:b}) \, \mathrm{d}x_{1:b} \propto \mu(z) \prod_{j=1}^{b} f_j^{(\lambda)}(z). \tag{7.3}$$

Here, we may view each $f_j^{(\lambda)}$ as a smoothed form of f_j, with π_λ being the corresponding smoothed form of the density of interest (6.7).

The role of λ is to control the fidelity of $f_j^{(\lambda)}$ to f_j, and so we assume the following in the sequel.

Assumption 7.1. For all $\lambda > 0$, $f_j^{(\lambda)}$ is bounded for each $j \in \{1, \ldots, b\}$; and $f_j^{(\lambda)} \longrightarrow f_j$ pointwise as $\lambda \longrightarrow 0$ for each $j \in \{1, \ldots, b\}$.

For example, Assumption 7.1 implies that π_λ converges in total variation to π by Scheffé's lemma (Scheffé, 1947), and therefore $\pi_\lambda(\varphi) \longrightarrow \pi(\varphi)$ for bounded $\varphi : \mathsf{E} \longrightarrow \mathbb{R}$. A sufficient

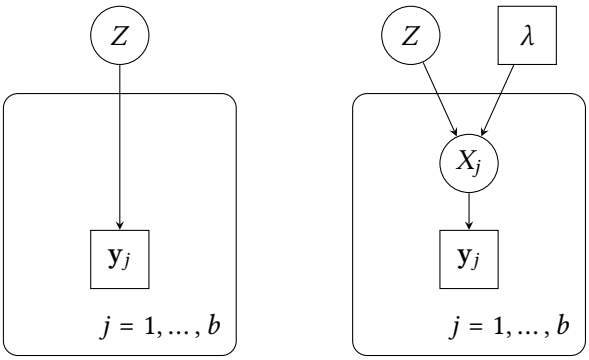


Figure 7.1.: Directed acyclic graphs, representing the original statistical model (left) and the instrumental model we construct (right).

condition for Assumption 7.1 to hold is that for each $j \in \{1, \ldots, b\}$ and for μ-almost all $z \in \mathsf{E}$, the probability measure associated with $K_j^{(\lambda)}(z, \cdot)$ converges weakly to the Dirac measure concentrated at z. In particular, Assumption 7.1 is satisfied for essentially any kernel that may be used for kernel density estimation; here, λ takes a similar role to that of the smoothing bandwidth. Another perspective is that Assumption 7.1 is satisfied by taking $K_j^{(\lambda)}(z, \cdot)$ to be a 'mollifier' function, as originally introduced by Friedrichs (1944).

On a first reading one may wish to assume that the $K_j^{(\lambda)}$ are chosen to be independent of j; for example, with $\mathsf{E} = \mathbb{R}$ one could take $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, \lambda)$. We describe some considerations in choosing these transition kernels in Section 7.3.2, and describe settings in which choosing these to differ with j may be beneficial.

The instrumental hierarchical model is presented diagrammatically in Figure 7.1. The variables $X_{1:b}$ may be seen as 'proxies' for Z associated with each of the data subsets, which are conditionally independent given Z and λ. Loosely speaking, λ represents the extent to which we allow the local variables $X_{1:b}$ to differ from the global variable Z. In terms of computation, it is the separation of Z from the subsets of the data $\mathbf{y}_{1:b}$, given $X_{1:b}$ introduced by the instrumental model, that can be exploited by distributed algorithms.

This construction was initially presented in Rendell et al. (2018). Essentially the same framework has been independently and contemporaneously proposed in a serial context by Vono et al. (2019a). Rather than distributing the computation, the authors focus on the setting where $b = 1$ to obtain a relaxation of the original simulation problem, constructing a Gibbs sampler via a 'variable splitting' approach (the case in which $b > 1$ is described in an appendix). An implementation of this approach for problems in binary logistic regression is proposed in Vono et al. (2018), with a number of non-asymptotic and convergence results presented more recently in Vono et al. (2019b). Our contemporaneous work focuses on applications of this framework in distributed settings. In Chapter 8, we also provide a sequential Monte Carlo implementation of the framework for use in this context.

This approach to constructing an artificial joint target density is easily extended to ac-

commodate random effects models, in which the original statistical model itself contains local variables associated with each data subset. These variables may be retained in the resulting instrumental model, alongside the local proxies $X_{1:b}$ for Z. We detail this fully in Section 7.5.

7.1.1. Motivating concepts

The framework we describe is motivated by concepts in distributed optimisation, a connection that is also explored in the contemporaneous work of Vono et al. (2019a). The *global consensus* optimisation problem is that of minimising a sum of functions on a common domain, under the constraint that their arguments are all equal to some global common value (see Boyd et al., 2011, Section 7 for a review). That is, for a collection of functions $g_{1:b}$, $h : \mathsf{E} \longrightarrow \mathbb{R}$ one looks to minimise the expression

$$\sum_{j=1}^{b} g_j(x_j) + h(z)$$

under the constraint that $z = x_j$ for $j \in \{1, \dots, b\}$. This problem is of practical interest when each of the functions g_j may only be evaluated on its own processor, with h computable on some central node.

Consider the problem of maximising $\pi(z)$ as defined in (6.7), for example to compute a MAP (maximum *a posteriori*) estimator. If for each $j \in \{1, \dots, b\}$ one uses the Gaussian kernel density $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, \lambda)$, then taking the negative logarithm of (7.1) gives

$$-\log \tilde{\pi}_\lambda(z, x_{1:b}) = C - \log \mu(z) - \sum_{j=1}^{b} \log f_j(x_j) + \frac{1}{2\lambda} \sum_{j=1}^{b} (z - x_j)^2 \tag{7.4}$$

where C is a normalising constant. Maximising π is equivalent to minimising this function under the constraint that $z = x_j$ for $j \in \{1, \dots, b\}$. We may view this as a global consensus optimisation problem, with an L_2 penalty term introduced by the Gaussian kernels $K_j^{(\lambda)}$. This facilitates the application of the alternating direction method of multipliers (Bertsekas and Tsitsiklis, 1989), a numerical method for solving such problems in distributed settings. Specifically, (7.4) corresponds to using $1/\lambda$ as the penalty parameter in this procedure.

Beyond this link with distributed optimisation, there are some similarities between our framework and approximate Bayesian computation (ABC), as previously introduced in Section 2.3.1 (see Marin et al., 2012, for a review of such methods). In both cases one introduces a kernel that can be viewed as acting to smooth the likelihood. In the case of (7.1) the role of λ is to control the scale of smoothing that occurs in the parameter space; in contrast, the tolerance parameter used in ABC controls the extent of a comparable form of smoothing in the observation (or summary statistic) space.

When using the Gaussian kernel density $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, \lambda)$, the smoothing (7.2) of

the partial likelihood terms corresponds to the application of a Weierstrass transform (see Zayed, 1996, Chapter 18 for a definition and summary of its properties). This connection, and the generalisation to other kernel density functions, is also exploited by an embarrassingly parallel algorithm proposed by Wang and Dunson (2013). We discuss their approach and compare with our proposed algorithm in Section 7.2.3.

7.2. Distributed Metropolis-within-Gibbs

The instrumental model described forms the basis of our proposed global consensus framework; *global consensus Monte Carlo* is correspondingly the application of Monte Carlo methods to form an approximation π_λ^N of the smoothed distribution π_λ, which has density (7.3). If λ is chosen to be sufficiently small, then π_λ provides an approximation of the distribution of interest π. An approximation of $\pi(\varphi)$, for some $\mathcal{E}$-measurable function $\varphi : \mathsf{E} \longrightarrow \mathbb{R}$, is therefore given by $\pi_\lambda^N(\varphi)$.

We here describe the construction of a Metropolis-within-Gibbs Markov kernel that leaves invariant $\tilde{\pi}_\lambda$, the joint distribution with density (7.1). Given a chain with values denoted $(Z^i, X_{1:b}^i)$ for $i \in \{1, \dots, N\}$, an approximation of the Z-marginal π_λ may be formed according to (6.1) as

$$\pi_\lambda^N := \frac{1}{N} \sum_{i=1}^{N} \delta_{Z^i},$$

An approximation of the integral $\pi(\varphi)$ is therefore given by

$$\pi_\lambda^N(\varphi) = \frac{1}{N} \sum_{i=1}^{N} \varphi(Z^i). \tag{7.5}$$

The Metropolis-within-Gibbs kernel we consider utilises the full conditional densities

$$\tilde{\pi}_\lambda(x_j \,|\, z) \;\propto\; K_j^{(\lambda)}(z, x_j) f_j(x_j) \tag{7.6}$$

for $j \in \{1, \dots, b\}$, and

$$\tilde{\pi}_\lambda(z \,|\, x_{1:b}) \;\propto\; \mu(z) \prod_{j=1}^{b} K_j^{(\lambda)}(z, x_j), \tag{7.7}$$

where (7.6) follows from the mutual conditional independence of $X_{1:b}$ given Z. Here we observe that $K_j^{(\lambda)}(z, x_j)$ simultaneously provides a pseudo-prior for X_j and a pseudo-likelihood for Z.

We define $M_1^{(\lambda)}$ to be a $\tilde{\pi}_\lambda$-invariant Markov kernel that fixes z. Specifically, we consider a kernel of the form

$$M_1^{(\lambda)}((z, x_{1:b}); \mathrm{d}(z', x'_{1:b})) = \delta_z(\mathrm{d}z') \prod_{j=1}^{b} P_{j,z}^{(\lambda)}(x_j, \mathrm{d}x'_j), \tag{7.8}$$

where for each j, $P_{j,z}^{(\lambda)}(x_j, \cdot)$ is a Markov kernel leaving (7.6) invariant. We similarly define

$M_2^{(\lambda)}$ to be a $\tilde{\pi}_\lambda$-invariant Markov kernel that fixes $x_{1:b}$,

$$M_2^{(\lambda)}((z, x_{1:b}); \mathrm{d}(z', x'_{1:b})) = \left[\prod_{j=1}^{b} \delta_{x_j}(\mathrm{d}x'_j)\right] P^{(\lambda)}_{x_{1:b}}(z, \mathrm{d}z'), \tag{7.9}$$

where $P^{(\lambda)}_{x_{1:b}}(z, \cdot)$ is a Markov kernel leaving (7.7) invariant.

Using these Markov kernels we construct an MCMC kernel that leaves $\tilde{\pi}_\lambda$ invariant; we present the resulting sampling procedure as Algorithm 7.1.

Algorithm 7.1 Global consensus Monte Carlo: MCMC algorithm

1. Fix $\lambda > 0$, and set initial state $(Z^0, X^0_{1:b})$.
2. For $i = 1, \ldots, N$,
   - For $j \in \{1, \ldots, b\}$, independently sample $X^i_j \sim P^{(\lambda)}_{j,Z^{i-1}}(X^{i-1}_j, \cdot)$.
   - Sample $Z^i \sim P^{(\lambda)}_{X^i_{1:b}}(Z^{i-1}, \cdot)$.

The interest from a distributed perspective is that the full conditional density (7.6) of each X_j, for given values x_j and z, depends only on the jth block of data (through the partial likelihood f_j) and may be computed on the jth machine. Within Algorithm 7.1, the sampling of each X^i_j from $P^{(\lambda)}_{j,Z^{i-1}}(X^{i-1}_j, \cdot)$ may therefore occur on the jth machine; these $X^i_{1:b}$ may then be communicated to a central machine that draws Z^i.

In the special case in which one may sample exactly from the conditional distributions (7.6)–(7.7), Algorithm 7.1 takes the form of a Gibbs sampler. This setting is particularly amenable to analysis, and we provide such a study in Section 7.4. The same Gibbs sampler construction has recently been proposed independently by Vono et al. (2019a); rather than considering distributed computation, their main objective was to improve algorithmic performance by constructing full conditional distributions that are more tractable than the full posterior density.

Typically however it is not possible to sample exactly from the full conditional distributions (7.6) of each X_j; in this case one may choose each $P^{(\lambda)}_{j,z}$ to be an MCMC kernel, so that Algorithm 7.1 takes a Metropolis-within-Gibbs form. Our approach has particular benefits in this setting, discussed in the following section.

7.2.1. Repeated MCMC kernel iterations

When sampling exactly from the full conditional densities (7.6) is not possible, one may choose the Markov kernels $P^{(\lambda)}_{j,z}$ to comprise multiple iterations of an MCMC kernel leaving (7.6) invariant. As described above, the full conditional density of X_j may be computed on the jth machine without requiring communication between machines; it follows that multiple MCMC accept/reject steps may be conducted on each of the b nodes, without

inter-node communication. This stands in contrast to MCMC approaches directly targeting π, in which such communication is required for each evaluation of (6.7), and therefore for every accept/reject step.

Similar to such 'direct' MCMC approaches, each iteration of Algorithm 7.1 requires communication to and from each of the b machines on which the data are stored: the current value of Z must be communicated from some central machine to each of these b machines, and the updated values of $X_{1:b}$ must be communicated back to the central machine in order to update Z. However as described above, each such iteration may contain multiple evaluations of each partial likelihood f_j. In cases where the communication latency is high, the resulting sampler will spend a greater proportion of time on likelihood computations compared to a 'direct' approach, with less time lost due to latency. The result is that more time is spent exploring the state space, which may in turn result in faster mixing (e.g. with respect to wall-clock time).

To analyse this more concretely, we consider an abstracted distributed setting:

- Let ℓ represent the approximate wall-clock time required to compute each $f_j(z)$ for a given $z \in \mathsf{E}$, which we here assume is independent of j for simplicity.
- Let the communication latency be C; for the purposes of this analysis we shall consider the additional time taken due to bandwidth restrictions to be negligible.
- Assume also that the time taken to compute the prior μ, and the global consensus transition densities $K_j^{(\lambda)}$, is negligible.

In an MCMC approach directly targeting the full posterior, each accept/reject step requires communication to and from each node in order to compute the posterior density $\pi(z)$ at the proposed $z \in \mathsf{E}$. Assuming that the likelihood contributions of each block may be computed synchronously, the time taken by each iteration of such an algorithm is therefore approximately $\ell + 2C$.

Within our proposed global consensus approach computations of the full conditional densities of each X_j, and of Z, may each occur on a single node. Suppose that the Markov kernels $P_{j,z}^{(\lambda)}$ comprise k iterations of an MCMC kernel leaving $\pi_\lambda(x_j \mid z)$ invariant. Then, under the same assumptions of synchronous computation, a single iteration of Algorithm 7.1 (generating one new value of the Z-chain) requires a time of approximately $k\ell + 2C$.

The consequence is that, while our proposed approach would generally generate fewer samples per unit of wall-clock time, the proportion of time spent on likelihood computation (rather than communication) may be made far greater: $k\ell/(k\ell + 2C)$ for global consensus Monte Carlo, versus $\ell/(\ell + 2C)$ for the 'direct' MCMC approach. This may be especially important when the latency C is large compared to the likelihood computation time ℓ; by choosing the number of MCMC kernel applications k to be sufficiently large, the resulting sampler will spend a greater proportion of time exploring the state space, and

may therefore exhibit faster mixing (with respect to wall-clock time). This approach may be particularly useful for high-dimensional settings, in which constructing a well-mixing MCMC kernel can be difficult.

The 'local' application of MCMC kernels in our framework may also allow a wider range of such kernels to be computationally feasible than in a direct approach. For example, in the case where the state space E is multi-dimensional, one may wish to use 'componentwise' proposals, in which new values are proposed for each individual component (or collection thereof), with all others held fixed. Updating each component in turn may be infeasible in a direct MCMC approach, due to the communication latency involved in computing the acceptance probability for each proposed value.

Similarly, our proposed framework may also be beneficial when using adaptive MCMC algorithms in distributed settings (see Andrieu and Thoms, 2008, for a review). In an MCMC approach directly targeting the full posterior π, adaptation of the proposal distribution may be slow (in the sense of wall-clock time) due the communication required in each accept/reject step. Within the global consensus framework, for which the acceptance probabilities required by the MCMC kernels $P_{j,z}^{(\lambda)}$ may be computed locally, several accept/reject steps may take place for each new value of the Z-chain. Relative to the number of Z-samples generated, the adaptation of the proposals used by these local MCMC kernels may be faster than the adaptation of the proposal used by a kernel directly targeting π. This may contribute to better mixing of the resulting Z-chain.

Finally, when each $P_{j,z}^{(\lambda)}$ comprises enough MCMC iterations to exhibit good mixing, the resulting Metropolis-within-Gibbs algorithm may behave similarly to the corresponding Gibbs sampler. Our analysis of the Gibbs setting in Section 7.4 may therefore be informative about this more general setting.

7.2.2. Pseudo-marginal MCMC kernels

Compared to an MCMC algorithm directly targeting the full posterior, a particular setting in which our proposed approach may exhibit benefits is that in which pseudo-marginal MCMC kernels are used, as previously introduced in Section 6.2.1. The efficiency of pseudo-marginal algorithms can depend heavily on the noise of the unbiased estimates of the target density, used in computing the acceptance ratio (see e.g. Andrieu and Vihola, 2015).

Returning to the notation introduced in Section 6.2.1, suppose that the collection of latent variables used to form these estimates is $\Xi = \Xi_{1:N'}$. For example, in Beaumont (2003) these correspond to N' IID samples drawn from some importance distribution; in the 'particle marginal Metropolis–Hastings' sampler of Andrieu et al. (2010), these are the N' particles in an SMC algorithm used to estimate the marginal likelihood. A larger value of N' results in unbiased estimates of the target density that are lower in variance, but more expensive to evaluate.

In practice, one chooses N' to balance the computational cost of computing these es-

timators with the mixing properties of the resulting algorithm. The problem of choosing N' has been considered by several authors (Pitt et al., 2012; Doucet et al., 2015; Sherlock et al., 2015), who consider tuning the variances of estimates of the target log-density. Again using the notation of Section 6.2.1, these variances may be expressed as

$$\mathrm{var}_{\rho_z}[\log \tilde{\pi}(z, \cdot)]$$

for values of $z \in \mathsf{E}$. Under various assumptions, the authors find that the optimal variance is generally between 1 and 4.

In our distributed setting, suppose that each partial likelihood f_j cannot easily be evaluated exactly, but that for any $z \in \mathsf{E}$ one may compute an unbiased estimate of each likelihood contribution $f_j(z)$, independently for each j. For a pseudo-marginal MCMC sampler directly targeting the full posterior (6.7), each iteration requires unbiased estimation of all b likelihood contributions. The variance of each estimate of the target log-density is therefore equal to the sum of the variances of these log-likelihood estimates.

In contrast, within our algorithm the full conditional densities (7.6) each depend on only one f_j. Suppose the Markov kernels $P_{j,z}^{(\lambda)}$ targeting these densities are chosen as pseudo-marginal kernels. In order for the estimates of the target log-densities to have comparable variances to those in the 'direct' MCMC algorithm (e.g. close to the aforementioned proposed optimal values), one may use estimators of $f_j(z)$ of greater variance than would be used in the direct algorithm, as there is no other source of variance.

The consequence is that the construction of well-mixing pseudo-marginal kernels may here be achieved by generating cheaper estimates of each likelihood contribution than would be required in an MCMC approach directly targeting the full posterior. That is to say, pseudo-marginal MCMC kernels used within our approach may use much lower values of N' than kernels of the same construction used to target π directly.

Correspondingly, the use of Algorithm 7.1 to target π_λ may allow *more* samples to be drawn per unit wall-clock time than an algorithm directly targeting π, and using pseudo-marginal kernels of comparable mixing quality. Embarrassingly parallel algorithms also use MCMC samplers targeting densities that only depend on a single partial likelihood f_j, and benefit from this property similarly.

We further describe this phenomenon in Section 9.4, where we consider a numerical example employing pseudo-marginal kernels.

7.2.3. Comparisons with embarrassingly parallel approaches

Our proposed algorithm has similar, but not identical, objectives to the embarrassingly parallel algorithms introduced in Section 6.3.1. Instead of aiming to minimise entirely communication between nodes, our algorithm avoids the aggregation step common to such approaches, and the difficulties that may be associated with its construction. For

example, our approach avoids the need to make distributional assumptions of π, such as the Gaussian assumption that is implicit in the averaging step of consensus Monte Carlo (Scott et al., 2016). Our framework also does not depend on techniques that may fail in high-dimensional settings (e.g. kernel density estimation), as used in some of the other proposed aggregation techniques described in Section 6.3.1.

A potential issue common to embarrassingly parallel approaches is the treatment of the prior density μ. Each subposterior density (6.8) is typically assigned an equal share of the prior information in the form of a fractionated prior density $\mu(z)^{1/b}$, but it is not clear when this approach is satisfactory. For example, suppose μ belongs to an exponential family; any property that is not invariant to multiplying the canonical parameters by a constant will not be preserved in the fractionated prior. For several common distributions (including gamma and Wishart), this is true of the first moment. As such if $\mu(z)^{1/b}$ is proportional to a valid probability density function of z, then the corresponding distribution may be qualitatively very different to the full prior. Although Scott et al. (2016) note that fractionated priors perform poorly on some examples (for which tailored solutions are provided), no other general way of assigning prior information to each block naturally presents itself. In contrast our approach avoids this problem entirely, with μ providing prior information for Z at the 'global' level.

The embarrassingly parallel approaches proposed by Wang and Dunson (2013) bear some relation to our proposed framework, being based on the application of Weierstrass transforms to each subposterior density (6.8). This results in a collection of smoothed densities, analogous to the manner in which (7.2) represents a smoothed form of the partial likelihood f_j. As well as proposing a method for aggregating subposterior chains based on rejection sampling, the authors propose a technique for 'refining' an initial posterior approximation. Specifically, given an initial collection of N values approximating samples from the full posterior density, a refinement step is applied to each value, which may be expressed as the application of a Gibbs kernel on an extended state space.

Comparing with the global consensus framework, this is analogous to first re-expressing the prior density $\mu(z)$ in (7.1) as a product of b fractionated prior densities $\mu(z)^{1/b}$, and absorbing these into the b partial likelihood terms $f_j(z)$. One then applies Algorithm 7.1 for one iteration with N different initial values. Wang and Dunson (2013) indicate that it may be beneficial to repeat this process, each time possibly using a different 'refinement parameter' (corresponding to λ in our framework); this bears some similarities to our proposed SMC algorithm, which we introduce in Chapter 8.

7.3. Implementation considerations

7.3.1. Choosing the regularisation parameter

Within Algorithm 7.1, the regularisation parameter λ takes the role of a tuning parameter. We can view its effect on the mean squared error of approximations (7.5) of $\pi(\varphi)$ using the bias–variance decomposition

$$\mathbb{E}\left[\left(\pi_\lambda^N(\varphi) - \pi(\varphi)\right)^2\right] = \left[\pi_\lambda(\varphi) - \pi(\varphi)\right]^2 + \operatorname{var}\left[\pi_\lambda^N(\varphi)\right]; \tag{7.10}$$

when $\mathbb{E}[\pi_\lambda^N(\varphi)] = \pi_\lambda(\varphi)$ this holds exactly, and corresponds to (6.5). In many practical cases (7.10) will provide a very accurate approximation for large N, as the squared bias of $\pi_\lambda^N(\varphi)$ is typically asymptotically negligible in comparison to its variance.

The decomposition (7.10) separates the contributions to the error from the bias introduced by the instrumental model and the variance associated with the MCMC approximation. If λ is too large, the squared bias term in (7.10) can dominate while if λ is too small, the Markov chain may exhibit poor mixing due to strong conditional dependencies between $X_{1:b}$ and Z, and so the variance term in (7.10) can dominate.

It follows that λ should ideally be chosen in order to balance these two considerations; the effect of λ is investigated theoretically in the analysis in Section 7.4, and empirically in the examples of Chapter 9. An alternative that we explore in Chapter 8 is to use Markov kernels formed via Algorithm 7.1 within an SMC sampler. In this manner a decreasing sequence of λ values may be considered, which may result in lower-variance estimates for small λ values; we also describe a possible bias correction technique.

7.3.2. Choosing the Markov transition densities

As discussed in Section 7.1, in order to obtain the desired convergence properties of π_λ we require that Assumption 7.1 is satisfied, which follows from an appropriate choice of the Markov transition densities $K_j^{(\lambda)}$. For a state space $\mathsf{E} = \mathbb{R}$, a simple choice would be to take

$$K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, c_j\lambda) \tag{7.11}$$

for $j \in \{1, \ldots, b\}$, where $c_1, \ldots, c_b$ are positive values; we discuss subsequently how these might be chosen. Similarly, if $\mathsf{E} = \mathbb{R}^d$, one might choose $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, c_j\lambda I)$, and if E is some subset of $\mathbb{R}^d$, one might choose each $K_j^{(\lambda)}$ to correspond to a Gaussian density on some transformed space. We describe some such choices in the examples of Sections 9.2 and 9.4, where $\mathsf{E} = \mathbb{R}_+$ and $\mathsf{E} = [0, 1]^d$ respectively.

Depending on μ and $f_{1:b}$, appropriate choices of $K_j^{(\lambda)}$ may enable direct sampling from some of the conditional distributions (7.6)–(7.7) of Z and $X_{1:b}$. For example, $K_j^{(\lambda)}(\cdot, x_j)$ is a pseudo-likelihood for $Z \sim \mu$, and so if μ is conjugate to $K_j^{(\lambda)}(\cdot, x_j)$ for each $j \in \{1, \ldots, b\}$, then

the conditional distribution of Z given $X_{1:b}$ will be from the same family as μ. Similarly, one might choose for each $K_j^{(\lambda)}(z, \cdot)$ a conjugate prior for the partial likelihood terms f_j, so that the conditional distribution of each X_j given $\mathbf{y}_j$ and Z is from the same family as $K_j^{(\lambda)}(z, \cdot)$.

It may also be appropriate to choose the Markov transition densities to have relative scales comparable to those of the corresponding partial likelihood terms. To motivate this consider a univariate setting in which the partial likelihood terms are Gaussian, so that we may write $f_j(z) = \mathcal{N}(\mu_j; z, \sigma_j^2)$ for each $j \in \{1, \dots, b\}$. Suppose one uses the Gaussian transition densities (7.11), where $c_1, \dots, c_b$ are positive values controlling the relative strengths of association between Z and the local variables $X_{1:b}$.

As seen in (7.3), in the approximating density π_λ the partial likelihood terms f_j are replaced by smoothed terms (7.2), in this case given by

$$f_j^{(\lambda)}(z) \propto \mathcal{N}(\mu_j;\, z,\, \sigma_j^2 + c_j\lambda). \tag{7.12}$$

The resulting smoothed posterior density is presented as (7.14) in Section 7.4, where this setting is further explored. In this case, the role of λ may be seen as 'diluting' or down-weighting the contribution of each partial likelihood to the posterior distribution π_λ. A natural choice is to take $c_j \propto \sigma_j^2$, so that the dilution of each f_j is in proportion to the strength of its contribution to π. In this case (7.12) becomes

$$f_j^{(\lambda)}(z) \propto \mathcal{N}(\mu_j;\, z,\, (1 + c\lambda)\sigma_j^2)$$

for some constant c. The relative strengths of contribution of the $f_{1:b}$ are thereby preserved in the posterior density π_λ.

A particular case of interest in that in which the blocks of data $\mathbf{y}_j$ differ in size. If each datum y_ℓ has a likelihood contribution of the form $\mathcal{N}(y_\ell; z, \sigma^2)$, then the jth partial likelihood may be expressed as $f_j(z) \propto \mathcal{N}(\bar{y}_j; z, \sigma^2/n_j)$, where n_j is the number of data in the jth block and $\bar{y}_j$ is their mean. Taking $c_j \propto 1/n_j$, the smoothed partial likelihood (7.12) becomes

$$f_j^{(\lambda)}(z) \propto \mathcal{N}(\bar{y}_j;\, z,\, (\sigma^2 + c\lambda)/n_j) \tag{7.13}$$

for some c, so that the information from each observation is diluted in a consistent way. We present a numerical demonstration of the use of these scaled Markov transition densities in Section 9.1.1, for a Gaussian model corresponding to the setting described here. Motivated by Bayesian asymptotic arguments, we suggest that this scaling of the regularisation parameter in inverse proportion to the relative block sizes may be beneficial in more general settings.

The effect of such choices on the Metropolis-within-Gibbs algorithm is most readily seen by considering the improper uniform prior $\mu(z) \propto 1$ (a Gaussian prior is considered

in Section 7.4). Taking $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, c_j\lambda)$, the conditional density of Z given $X_{1:b}$ is

$$\tilde{\pi}_\lambda(z \mid x_{1:b}) = \mathcal{N}\left(z;\ \frac{\sum_{j=1}^b x_j/c_j}{\sum_{j=1}^b 1/c_j},\ \frac{\lambda}{\sum_{j=1}^b 1/c_j}\right).$$

Therefore, when updating Z given the local variables' current values $x_{1:b}$, the choice of $c_{1:b}$ dictates the relative influence of each such value. For example, we might expect the local variables corresponding to larger blocks to be more informative about the distribution of Z, which further justifies choosing $c_{1:b}$ to be inversely proportional to the block sizes.

In a multidimensional setting, one could control the covariance structure of each X_j given Z by using transition densities of the form $\mathcal{N}(x; z, \lambda\Psi_j)$, where $\Psi_{1:b}$ are positive definite matrices. By a similar Gaussian analysis, one could preserve the relative strengths of contribution of the partial likelihood terms by choosing for each Ψ_j an approximation of the covariance matrix of f_j.

7.4. Theoretical analysis for a simple model

To study the theoretical properties of our algorithm, we here consider a simple model where the goal is to infer the mean of a normal distribution. While our approach does not require the distribution π to be approximately Gaussian, the behaviour of our algorithm in this simple setting is particularly amenable to analysis. The results here may also be indicative of performance for regular models with abundant data due to the Bernstein–von Mises theorem (see e.g. van der Vaart, 2000, Section 10.2).

Let $\mu(z) = \mathcal{N}(z; \mu_0, \sigma_0^2)$, and for each $j \in \{1, \dots, b\}$ let $f_j(z) = \mathcal{N}(\mu_j; z, \sigma_j^2)$ and $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, c_j\lambda)$, following Section 7.3.2. We obtain

$$\pi_\lambda(z) = \mathcal{N}\left(z;\ \delta_\lambda^2\left[\frac{\mu_0}{\sigma_0^2} + \sum_{j=1}^b \frac{\mu_j}{\sigma_j^2 + c_j\lambda}\right],\ \delta_\lambda^2\right), \quad \delta_\lambda^2 = \left(\frac{1}{\sigma_0^2} + \sum_{j=1}^b \frac{1}{\sigma_j^2 + c_j\lambda}\right)^{-1}, \tag{7.14}$$

and $\pi(z)$ can be recovered by taking $\lambda = 0$ in (7.14). The corresponding full conditional densities for (7.14) are

$$\tilde{\pi}_\lambda(x_j \mid z) = \mathcal{N}\left(x_j;\ \frac{\sigma_j^2 z + c_j\lambda\mu_j}{\sigma_j^2 + c_j\lambda},\ \frac{c_j\lambda\sigma_j^2}{\sigma_j^2 + c_j\lambda}\right)$$

for $j \in \{1, \dots, b\}$, and

$$\tilde{\pi}_\lambda(z \mid x_{1:b}) = \mathcal{N}\left(z;\ \tilde{\delta}_\lambda^2\left[\frac{\mu_0}{\sigma_0^2} + \sum_{j=1}^b \frac{x_j}{c_j\lambda}\right],\ \tilde{\delta}_\lambda^2\right), \quad \tilde{\delta}_\lambda^2 = \left(\frac{1}{\sigma_0^2} + \sum_{j=1}^b \frac{1}{c_j\lambda}\right)^{-1}.$$

We consider the form of Algorithm 7.1 in which $M_1^{(\lambda)}$ and $M_2^{(\lambda)}$ as defined in (7.8)–(7.9) are Gibbs kernels. That is, we consider the case in which we may draw samples exactly from

the full conditional distributions (7.6)–(7.7). We choose this setting to facilitate analysis of the resulting Markov chain, but these results may be informative about more general Metropolis-within-Gibbs settings in which well-mixing Markov kernels are used (e.g. those comprising multiple MCMC kernel iterations, as described in Section 7.2.1).

Since $X_{1:b}$ are conditionally independent given Z and can therefore be updated simultaneously given Z, Algorithm 7.1 may be viewed analytically as a Gibbs sampler on two variables: Z and $X_{1:b}$. For a two-variable Gibbs Markov chain, each of the two 'marginal' chains (the sequences of states for each of the two variables) is also a Markov chain. In this setting we may therefore consider the Z-chain with transition kernel given by

$$M_{12}^{(\lambda)}(z, A) = \int_A \tilde{\pi}_\lambda(z' \mid x_{1:b}) \, \mathrm{d}z' \int_{\mathsf{E}^b} \left[\prod_{j=1}^{b} \tilde{\pi}_\lambda(x_j \mid z) \right] \mathrm{d}x_{1:b} \tag{7.15}$$

for $A \in \mathcal{E}$. Observing that $\tilde{\pi}_\lambda(z' \mid x_{1:b})$ depends on $x_{1:b}$ only through the sum $\sum_{j=1}^{b} x_j / c_j$, one can thereby show that the Z-chain defined by (7.15) is an AR(1) process. Specifically,

$$Z^i = C + \alpha Z^{i-1} + \epsilon_i, \qquad i > 0,$$

where

$$\alpha := \tilde{\delta}_\lambda^2 \sum_{j=1}^{b} \frac{\sigma_j^2}{c_j \lambda (\sigma_j^2 + c_j \lambda)}, \qquad C := \tilde{\delta}_\lambda^2 \left(\frac{\mu_0}{\sigma_0^2} + \sum_{j=1}^{b} \frac{\mu_j}{\sigma_j^2 + c_j \lambda} \right),$$

and the ϵ_i are IID zero-mean normal random variables, with variance

$$\tilde{\delta}_\lambda^2 \left(1 + \tilde{\delta}_\lambda^2 \sum_{j=1}^{b} \frac{\sigma_j^2}{c_j \lambda (\sigma_j^2 + c_j \lambda)} \right).$$

It follows that the autocorrelation of lag k is given by α^k for $k \geq 0$, and that $\alpha \longrightarrow 1$ as $\lambda \longrightarrow 0$.

7.4.1. Inferring the mean of a normal distribution

We now consider making the number of data n explicit in this setting. In particular, for some $z^* \in \mathbb{R}$ consider realisations $y_{1:n}$ of IID $\mathcal{N}(z^*, \sigma^2)$ random variables, grouped into b blocks. For simplicity, assume that b divides n, that each block contains n/b observations, and that the observations are allocated to the blocks sequentially, so that the jth block comprises those y_ℓ for which $\ell \in B_j := \{(j-1)n/b + 1, \ldots, jn/b\}$. Then

$$f_j(z) = \prod_{\ell \in B_j} \mathcal{N}(y_\ell; z, \sigma^2) \propto \mathcal{N}\left(\frac{b}{n} \sum_{\ell \in B_j} y_\ell; \; z, \; \frac{b}{n} \sigma^2 \right). \tag{7.16}$$

Since the blocks are of equal size in this case, so that each partial likelihood is of the same scale, we consider using $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, \lambda)$ for each j. From (7.14), we obtain

$$\pi_\lambda(z) = \mathcal{N}\left(z;\, \delta_\lambda^2 \left[\frac{\mu_0}{\sigma_0^2} + \frac{n\bar{y}}{\sigma^2 + n\lambda/b}\right],\, \delta_\lambda^2\right), \quad \delta_\lambda^2 = \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2 + n\lambda/b}\right)^{-1}. \tag{7.17}$$

Letting Id here denote the identity function on $\mathbb{R}$, we consider an estimator $\pi_\lambda^N(\mathrm{Id})$ of the posterior first moment $\pi(\mathrm{Id})$, of the form (7.5). We analyse its mean squared error using the bias–variance decomposition (7.10). The bias is

$$\pi_\lambda(\mathrm{Id}) - \pi(\mathrm{Id}) = \frac{n^2\,(\lambda/b)\,\sigma_0^2\,(\mu_0 - \bar{y})}{\left(\sigma^2 + n\sigma_0^2\right)\left(\sigma^2 + n\sigma_0^2 + n\lambda/b\right)}. \tag{7.18}$$

To assess the variance of $\pi_\lambda^N(\mathrm{Id})$, we consider the associated *asymptotic* variance,

$$\lim_{N\to\infty} N \operatorname{var}(\pi_\lambda^N(\varphi)) = \operatorname{var}(\varphi(Z_0))\left[1 + 2\sum_{k=1}^{\infty} \operatorname{corr}(\varphi(Z_0), \varphi(Z_k))\right], \qquad Z_0 \sim \pi_\lambda, \tag{7.19}$$

for φ square-integrable with respect to π_λ. As discussed earlier the Z-chain is an AR(1) process, and the autocorrelations are entirely determined by the autoregressive parameter

$$\alpha = \frac{n\sigma^2\sigma_0^2}{(\sigma^2 + n\lambda/b)\left(n\sigma_0^2 + n\lambda/b\right)},$$

from which one can find that the asymptotic variance for $\varphi = \mathrm{Id}$ is

$$\frac{\sigma_0^2\left(\sigma^2 + n\lambda/b\right)\left[(n\lambda/b)^2 + \left(\sigma^2 + n\sigma_0^2\right)(n\lambda/b) + 2n\sigma^2\sigma_0^2\right]}{(n\lambda/b)\left(\sigma^2 + n\sigma_0^2 + n\lambda/b\right)^2}. \tag{7.20}$$

Following the definition (7.19) of this asymptotic variance, dividing this expression by N gives an approximation of the variance term in (7.10) for large N.

As a caveat to this and the following analysis, estimation of the mean in Gaussian settings may not accurately reflect what happens in more complex settings. For example, if one uses an improper uniform prior then $\pi_\lambda(\mathrm{Id})$ is equal to $\pi(\mathrm{Id})$ for all λ, as seen in (7.18) with $\sigma_0^2 \longrightarrow \infty$; this will not be true in general.

One may also note that in this Gaussian setting, the variance of π_λ will always exceed the variance of the true target π, since the variance expression in (7.17) is an increasing function of λ. The effect is that estimation of the posterior variance in Gaussian settings is likely to result in positive bias, and confidence intervals for $\pi(\mathrm{Id})$ may be conservative. This, of course, simply reflects the fact that marginally the instrumental model can be viewed as replacing the original likelihood with a smoothed version as shown in (7.3).

7.4.1.1. Asymptotic optimisation of λ for large N

For fixed n, we consider the problem of choosing λ as a function of the chain length N, so as to minimise the mean squared error of the posterior mean estimator. This involves considering the contributions of the bias and variance to the mean squared error (7.10), in light of (7.18) and (7.20). Intuitively, with larger values of N, smaller values of λ can be used to reduce the bias while keeping the variance small. Defining $B(\lambda)$ to be the bias as given in (7.18), we see that as $\lambda \longrightarrow 0$,

$$\frac{B(\lambda)}{\lambda} \longrightarrow \frac{n^2\sigma_0^2\left(\mu_0 - \bar{y}\right)}{b\left(\sigma^2 + n\sigma_0^2\right)^2} =: B_\star.$$

Similarly, denoting by $V(\lambda)$ the asymptotic variance (7.20), we see that

$$\lambda V(\lambda) \longrightarrow \frac{2b\sigma^4\sigma_0^4}{\left(\sigma^2 + n\sigma_0^2\right)^2} =: V_\star.$$

For small λ, the MSE of the estimate is given approximately by

$$\mathbb{E}\left[\left(\pi_\lambda^N(\mathrm{Id}) - \pi(\mathrm{Id})\right)^2\right] \approx (\lambda B_\star)^2 + \frac{1}{N}\frac{V_\star}{\lambda}, \tag{7.21}$$

which may be shown to be minimised when

$$\lambda^3 = \frac{V_\star}{2B_\star^2 N} = \frac{b^3\sigma^4\left(\sigma^2 + n\sigma_0^2\right)^2}{n^4 N\left(\mu_0 - \bar{y}\right)^2}. \tag{7.22}$$

We see that, for a fixed number of data n, we should scale λ with the number of samples N as $\mathcal{O}(N^{-1/3})$. Substituting the corresponding value of λ into (7.21), we find that the corresponding minimal MSE behaves as $\mathcal{O}(N^{-2/3})$. Specifically, this minimal MSE is given by

$$\frac{3n^{4/3}\sigma^{8/3}\sigma_0^4\left(\mu_0 - \bar{y}\right)^{2/3}}{N^{2/3}\left(\sigma^2 + n\sigma_0^2\right)^{8/3}},$$

in which the contribution of the variance is twice that of the squared bias.

Note that in this example, all dependence on λ and b in the smoothed likelihood (7.17) is through their ratio λ/b. The result is that splitting the data into more blocks has the same effect as reducing λ, and so these results may be adapted to consider optimisation of the ratio λ/b. This relationship may not be representative of these variables' behaviour in other models; but in cases where Bernstein–von Mises arguments hold, such results may be useful in settings where the number of blocks b may be chosen by the practitioner.

$\gamma < 0$	$\delta^2_{(n)} \longrightarrow \sigma_0^2$	$\xi_{(n)} \xrightarrow{\text{a.s.}} \dfrac{\mu_0}{\sigma_0^2}$	$\mu_{(n)} \xrightarrow{\text{a.s.}} \mu_0$
$\gamma = 0$	$\delta^2_{(n)} \longrightarrow \left[\dfrac{1}{\sigma_0^2}+\dfrac{1}{c}\right]^{-1}$	$\xi_{(n)} \xrightarrow{\text{a.s.}} \dfrac{\mu_0}{\sigma_0^2}+\dfrac{z^*}{c}$	$\mu_{(n)} \xrightarrow{\text{a.s.}} \left[\dfrac{1}{\sigma_0^2}+\dfrac{1}{c}\right]^{-1}\left[\dfrac{\mu_0}{\sigma_0^2}+\dfrac{z^*}{c}\right]$
$\gamma \in (0,1)$	$n^\gamma \delta^2_{(n)} \longrightarrow c$	$n^{-\gamma}\xi_{(n)} \xrightarrow{\text{a.s.}} \dfrac{z^*}{c}$	$\mu_{(n)} \xrightarrow{\text{a.s.}} z^*$
$\gamma = 1$	$n\delta^2_{(n)} \longrightarrow \sigma^2+c$	$n^{-1}\xi_{(n)} \xrightarrow{\text{a.s.}} \dfrac{z^*}{\sigma^2+c}$	$\mu_{(n)} \xrightarrow{\text{a.s.}} z^*$
$\gamma > 1$	$n\delta^2_{(n)} \longrightarrow \sigma^2$	$n^{-1}\xi_{(n)} \xrightarrow{\text{a.s.}} \dfrac{z^*}{\sigma^2}$	$\mu_{(n)} \xrightarrow{\text{a.s.}} z^*$

Table 7.1.: Convergence results as $n \longrightarrow \infty$ relating to the terms in the approximate posterior distribution (7.23), when $\lambda_n/b_n = cn^{-\gamma}$ for some constant $c > 0$.

7.4.1.2. Posterior consistency and coverage of credible intervals as $n \longrightarrow \infty$

We now consider the behaviour of the algorithm as the number of data n tends to infinity. Recalling that we assume the true parameter value to be $z^* \in \mathbb{R}$, we may consider the consistency of the posterior distribution (7.17) by treating the data $Y_{1:n}$ as random. We denote their mean by $\bar{Y}_n$, which is normally distributed with mean z^* and variance σ^2/n. We shall also consider allowing λ and b to vary with n; making this explicit in the notation, (7.17) becomes

$$\pi_{\lambda_n}(z) = \mathcal{N}\left(z; \mu_{(n)}, \delta^2_{(n)}\right) = \mathcal{N}\left(z; \delta^2_{(n)}\xi_{(n)}, \delta^2_{(n)}\right), \tag{7.23}$$

where $\mu_{(n)} = \delta^2_{(n)}\xi_{(n)}$, and

$$\xi_{(n)} = \frac{\mu_0}{\sigma_0^2} + \frac{n\bar{Y}_n}{\sigma^2 + n\lambda_n/b_n}, \qquad \delta^2_{(n)} = \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2 + n\lambda_n/b_n}\right)^{-1}.$$

We consider $\lambda_n/b_n = cn^{-\gamma}$, for some constant $c > 0$. Convergence results for different values of γ are displayed in Table 7.1, obtained using the fact that $\bar{Y}_n \xrightarrow{\text{a.s.}} z^*$. We see that the posterior is consistent (see e.g. Ghosh and Ramamoorthi, 2003, Chapter 1) if $\gamma > 0$. Moreover, if $\gamma > 1$ then $1-\alpha$ credible intervals will have asymptotically a coverage probability of exactly $1-\alpha$ due to the convergence $n\delta^2_{(n)} \longrightarrow \sigma^2$.

If $\gamma \in (0,1)$ then the rate of approximate posterior contraction is too conservative, while if $\gamma = 1$ the corresponding credible intervals will be too wide by a constant factor depending on c. From a practical perspective, one can consider the case in which n/b corresponds to the maximum number of data that can be processed on an individual computing node. In such a setting, letting $b_n \propto n$ is reasonable and we require in addition that λ_n is decreasing to obtain credible intervals with asymptotically exact coverage.

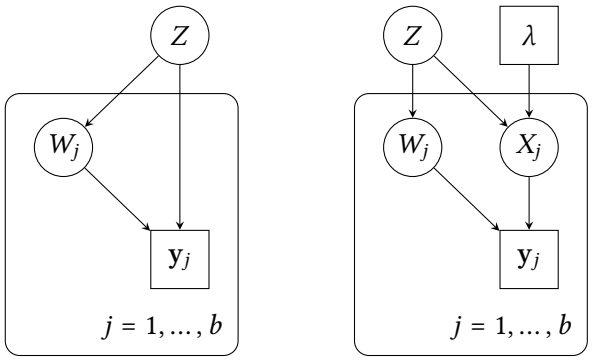


Figure 7.2.: Directed acyclic graphs for the random effects model, representing the original statistical model (left) and the instrumental model (right).

7.5. Random effects models

The approach described in Section 7.1 for constructing a joint target density on an extended state space can easily accommodate statistical models that not only contain a global variable Z, but also contain local variables $W_{1:b}$ associated with each data subset that are conditionally independent given the data. Models of this form include random effects models (see e.g. Laird and Ware, 1982) in which we assume that each block of data describes a different subpopulation, and that the variance of the data in each block has a subpopulation-specific component.

Specifically, suppose the true posterior density of $(Z, W_{1:b})$ satisfies

$$\pi(z, w_{1:b}) \propto \mu(z) \prod_{j=1}^{b} \nu_j(z, w_j) f_j(z, w_j) \tag{7.24}$$

where Z takes values $z \in \mathsf{E} \subseteq \mathbb{R}^d$, W_j takes values $w_j \in \mathsf{W}_j$, and μ and $\nu_{1:b}$ are prior densities. Again, we introduce a collection of b instrumental variables each defined on E, denoted by $X_{1:b}$, which may be viewed as local proxies for Z. This allows the construction of an artificial joint density $\tilde{\pi}_\lambda$ on $\mathsf{E} \times \mathsf{E}^b \times \prod_{j=1}^{b} \mathsf{W}_j$ given by

$$\tilde{\pi}_\lambda(z, x_{1:b}, w_{1:b}) \propto \mu(z) \prod_{j=1}^{b} \nu_j(z, w_j) K_j^{(\lambda)}(z, x_j) f_j(x_j, w_j). \tag{7.25}$$

The resulting instrumental hierarchical model is presented in Figure 7.2.

As previously, defining

$$f_j^{(\lambda)}(z, w) := \int_{\mathsf{E}} K_j^{(\lambda)}(z, x) f_j(x, w) \, \mathrm{d}x,$$

we may marginalise out $X_{1:b}$ from $\tilde{\pi}_\lambda$ to obtain

$$\pi_\lambda(z, w_{1:b}) := \int_{\mathsf{E}^b} \tilde{\pi}_\lambda(z, x_{1:b}, w_{1:b}) \,\mathrm{d}x_{1:b} \;\propto\; \mu(z) \prod_{j=1}^{b} \nu_j(z, w_j) f_j^{(\lambda)}(z, w_j). \tag{7.26}$$

Under Assumption 7.1 we obtain analogous convergence results to those described in Section 7.1, so that π_λ forms an approximation of π for sufficiently small λ.

The construction of a Metropolis-within-Gibbs sampler targeting the joint density (7.25) is conceptually similar to the approach described in Section 7.2, though we detail this here for completeness. We consider the full conditional densities of (X_j, W_j) for $j \in \{1, \ldots, b\}$; these take the form

$$\tilde{\pi}_\lambda(x_j, w_j \,|\, z) \;\propto\; \nu_j(z, w_j) K_j^{(\lambda)}(z, x_j) f_j(x_j, w_j), \tag{7.27}$$

where we note the mutual conditional independence of $(X_j, W_j)_{j=1}^b$ given Z. We also require the full conditional density of Z, which here is

$$\tilde{\pi}_\lambda(z \,|\, x_{1:b}, w_{1:b}) \;\propto\; \mu(z) \prod_{j=1}^{b} \nu_j(z, w_j) K_j^{(\lambda)}(z, x_j). \tag{7.28}$$

Again, we define $M_1^{(\lambda)}$ to be a $\tilde{\pi}_\lambda$-invariant Markov kernel that fixes z, taking this to be of the form

$$M_1^{(\lambda)}\big((z, x_{1:b}, w_{1:b}); \mathrm{d}(z', x'_{1:b}, w'_{1:b})\big) = \delta_z(\mathrm{d}z') \prod_{j=1}^{b} P_{j,z}^{(\lambda)}\big((x_j, w_j), \mathrm{d}(x'_j, w'_j)\big),$$

where for each j, $P_{j,z}^{(\lambda)}\big((x_j, w_j), \cdot\big)$ is a Markov kernel leaving (7.27) invariant. We similarly define $M_2^{(\lambda)}$ to be a $\tilde{\pi}_\lambda$-invariant Markov kernel that fixes $x_{1:b}$ and $w_{1:b}$,

$$M_2^{(\lambda)}\big((z, x_{1:b}, w_{1:b}); \mathrm{d}(z', x'_{1:b}, w'_{1:b})\big) = \left[\prod_{j=1}^{b} \delta_{x_j}(\mathrm{d}x'_j)\delta_{w_j}(\mathrm{d}w'_j)\right] P_{x_{1:b}, w_{1:b}}^{(\lambda)}(z, \mathrm{d}z'),$$

where $P_{x_{1:b}, w_{1:b}}^{(\lambda)}(z, \cdot)$ is a Markov kernel leaving (7.28) invariant.

The resulting Metropolis-within-Gibbs algorithm is presented as Algorithm 7.2. This may be implemented in essentially the same distributed manner as Algorithm 7.1, since sampling from each $P_{j,z}^{(\lambda)}\big((X_j, W_j), \cdot\big)$ may occur on the jth computing node, with implementation of $M_2^{(\lambda)}$ taking place on a central node.

An approximation of π_λ, the marginal distribution of $(Z, W_{1:b})$ with density (7.26), is obtained as

$$\pi_\lambda^N := \frac{1}{N} \sum_{i=1}^{N} \delta_{(Z^i, W_{1:b}^i)}.$$

If λ is sufficiently small that π_λ forms an approximation of π, then for some function of interest $\varphi : \mathsf{E} \times \prod_{j=1}^b \mathsf{W}_j \longrightarrow \mathbb{R}$, an approximation of $\pi(\varphi)$ is given by $\pi_\lambda^N(\varphi)$.

Algorithm 7.2 Global consensus Monte Carlo: MCMC algorithm with random effects

1. Fix $\lambda > 0$, and set initial state $(Z^0, X^0_{1:b}, W^0_{1:b})$.
2. For $i = 1, \dots, N$,
   - For $j \in \{1, \dots, b\}$, independently sample $(X^i_j, W^i_j) \sim P^{(\lambda)}_{j,Z^{i-1}}\big((X^{i-1}_j, W^{i-1}_j), \cdot\big)$.
   - Sample $Z^i \sim P^{(\lambda)}_{X^i_{1:b}, W^i_{1:b}}(Z^{i-1}, \cdot)$.

We provide a numerical example of the use of this algorithm, as applied to a stochastic volatility model, in Section 9.4.

7.6. Summary

We have presented a new framework for sampling in distributed settings. Given that our proposed approach makes no additional assumptions on the form of the likelihood beyond its factorised form as in (6.7), we expect that our algorithm will be most effective in those big data settings for which approximate Gaussianity of the likelihood contributions may not hold. These may include high-dimensional settings, for which some subsets of the data may be relatively uninformative about the parameter. In such cases the likelihood contributions may be highly non-Gaussian, so that the consensus Monte Carlo of Scott et al. (2016), presented as Algorithm 6.3, may result in estimates of high bias. Simultaneously, the high dimensionality may preclude the use of alternative combination techniques in embarrassingly parallel algorithms (e.g. the use of kernel density estimates, as discussed in Section 6.3.1).

The examples of Chapter 9 provide numerical demonstrations of Algorithm 7.1 applied to a number of models, illustrating the implementation considerations described in Section 7.3. We also note that our proposed Metropolis-within-Gibbs algorithm constitutes only one possible approach to inference within the instrumental hierarchical model that we propose; in the following chapter, we shall describe how these kernels might be used within an SMC sampler. We shall discuss other possible extensions in the conclusion of the thesis.

8. A sequential Monte Carlo approach to global consensus

8.1. Constructing an SMC sampler

As discussed in Section 7.3.1, as λ approaches zero estimators of the form (7.5) resulting from Algorithm 7.1 exhibit lower bias but higher variance, due to poorer mixing of the resulting Markov chain. In order to obtain lower-variance estimators for λ values close to 0, we present in this chapter an application of sequential Monte Carlo (SMC) methodology within our framework, allowing the generation of estimates for each in a sequence of values of λ.

Specifically we consider the construction of an SMC sampler, using the methodology introduced in Chapter 2. For a decreasing sequence of λ values that we shall denote $\lambda_0, \dots, \lambda_n$, we consider the sequence of distributions $\tilde{\pi}_{\lambda_p}$ on $\mathsf{E} \times \mathsf{E}^b$, as defined in (7.1). A collection of Markov kernels M_p leaving each such distribution invariant may be constructed according to Algorithm 7.1. The potential functions G_p, computed according to (2.2), take the simple form

$$G_p(z, x_{1:b}) = \frac{\tilde{\pi}_{\lambda_{p+1}}(z, x_{1:b})}{\tilde{\pi}_{\lambda_p}(z, x_{1:b})} = \prod_{j=1}^{b} \frac{K_j^{(\lambda_{p+1})}(z, x_j)}{K_j^{(\lambda_p)}(z, x_j)}.$$

Using Algorithm 2.1 in this setting results in an SMC implementation of the global consensus framework, which we present as Algorithm 8.1.

After each iteration of the algorithm, a particle approximation of $\tilde{\pi}_{\lambda_p}$ may be formed according to (1.16) as

$$\tilde{\pi}_{\lambda_p}^N := \frac{\sum_{i=1}^N W_p^i \delta_{\zeta_p^i}}{\sum_{i=1}^N W_p^i}.$$

Following Proposition 1.3, under weak conditions $\tilde{\pi}_{\lambda_p}^N(\varphi)$ converges almost surely to $\tilde{\pi}_{\lambda_p}(\varphi)$ as $N \to \infty$. One can also define the particle approximations of π_{λ_p}, the Z-marginals of these distributions as defined in (7.3), as

$$\pi_{\lambda_p}^N := \frac{\sum_{i=1}^N W_p^i \delta_{Z_p^i}}{\sum_{i=1}^N W_p^i}, \tag{8.1}$$

where Z_p^i is the first component of the particle ζ_p^i.

Algorithm 8.1 Global consensus Monte Carlo: SMC algorithm

1. At time $p = 0$:
   - For $i \in \{1, \dots, N\}$ set $W_0^i \longleftarrow 1$ and independently sample $\zeta_0^i = (Z_0^i, X_{0,1:b}^i) \sim \tilde{\pi}_{\lambda_0}$.
2. At time $p = 1, \dots, n$,
   - For $i \in \{1, \dots, N\}$ set $\tilde{W}_p^i \longleftarrow W_{p-1}^i \prod_{j=1}^{b} \frac{K_j^{(\lambda_p)}(Z_{p-1}^i, X_{p-1,j}^i)}{K_j^{(\lambda_{p-1})}(Z_{p-1}^i, X_{p-1,j}^i)}$.
   - If resampling in the pth iteration:
     - For $i \in \{1, \dots, N\}$ independently sample $A_{p-1}^i \sim \text{Categorical}\big(\tilde{W}_p^1, \dots, \tilde{W}_p^N\big)$ and set $W_p^i \longleftarrow 1$.
   - Else:
     - For $i \in \{1, \dots, N\}$ set $A_{p-1}^i \longleftarrow i$ and set $W_p^i \longleftarrow \tilde{W}_p^i$.
   - For $i \in \{1, \dots, N\}$ independently sample $\zeta_p^i \sim M_p(\zeta_{p-1}^{A_{p-1}^i}, \cdot)$, where M_p is a $\tilde{\pi}_{\lambda_p}$-invariant MCMC kernel constructed in the manner of Algorithm 7.1.

In the distributed setting described in Section 6.3, a careful implementation of the MCMC kernels used may allow the inter-node communication to be interleaved with the likelihood computations associated with the particles. Specifically, recall the form of Algorithm 7.1, in which all the local variables $X_{1:b}$ are updated on their respective computing nodes, with these values sent to a central node in order to update the global variable Z. In Algorithm 8.1, the local components of all N particles may be updated on the worker nodes, with their updated values all communicated to the central node simultaneously (i.e. rather than separately for each particle). The effect is that the costs associated with communication latency are reduced.

Although Algorithm 8.1 is specified for simplicity in terms of a fixed sequence $\lambda_0, \dots, \lambda_n$, a primary motivation for the SMC approach is that the sequence used can be determined adaptively while running the algorithm, using one of the techniques described in Section 3.2.2. Within the examples in Chapter 9 we employ the procedure proposed by Zhou et al. (2016), based on the conditional effective sample size (CESS). Given an initial value λ_0 and a choice of the CESS parameter used in this adaptive procedure, the sequence of decreasing λ values is determined in an automated way. In contrast to tempering, in the context of which this procedure was introduced in Section 3.2.2, there is no natural final value of λ at which the SMC algorithm should be terminated. We detail a possible approach to determining when to stop the algorithm, based on minimising the mean squared error (7.10), in Section 8.3.

With regard to initialisation, if it is not possible to sample from $\tilde{\pi}_{\lambda_0}$ one could instead use samples obtained by importance sampling, or one could initialise an SMC sampler with

some tractable distribution and use tempering or similar techniques to reach $\tilde{\pi}_{\lambda_0}$. At the expense of the introduction of an additional approximation, an alternative would be to run a $\tilde{\pi}_{\lambda_0}$-invariant Markov chain, and obtain an initial collection of particles by thinning the output (an approach that may be validated using results of Finke et al., 2020). Specifically, one could use Algorithm 7.1 to generate such samples for some large λ_0, benefiting from its good mixing and low autocorrelation when λ is sufficiently large. The effect of Algorithm 8.1 may then be seen as refining or improving the resulting estimators, by bringing the parameter λ closer to zero.

The benefits of SMC samplers described in Chapter 2 may make this approach preferable to the Markov chain scheme of Algorithm 7.1; for example, Algorithm 8.1 may be more robust to multimodality of π. Another point in favour of this approach is that many of the particle approximations (8.1) can be used to form a final estimate of $\pi(\varphi)$, which we explore in the following section.

8.2. Bias correction using local linear regression

Since Algorithm 8.1 generates a particle approximation of π_λ for many values of λ, a natural idea is to regress the values of $\pi_\lambda^N(\varphi)$ on λ, extrapolating to $\lambda = 0$ to obtain an estimate of $\pi(\varphi)$. A similar idea has been used for bias correction in the context of ABC, albeit not in an SMC setting, regressing on the discrepancy between the observed data and simulated pseudo-observations (Beaumont et al., 2002; Blum and François, 2010).

Under very mild assumptions on the transition densities $K_j^{(\lambda)}$, $\pi_\lambda(\varphi)$ is continuous as a function of λ. Considering a first-order Taylor expansion of this function, a simple approach is to model the dependence of $\pi_\lambda(\varphi)$ on λ as linear, for λ sufficiently close to 0. The Gaussian setting described in Section 7.4.1 illustrates this approach; in that case, define by $\psi(\lambda)$ the first moment of π_λ, which has density (7.17). A Taylor expansion about $\lambda = 0$ gives

$$\psi(\lambda) = \psi(0) - \frac{n(\mu_0 - \bar{y})}{n + \sigma^2/\sigma_0^2} \sum_{k=1}^{\infty} \left(-\frac{\lambda}{b(\sigma^2/n + \sigma_0^2)} \right)^k, \tag{8.2}$$

in which the linear term in the sum dominates for sufficiently small λ. A similar argument may be applied to the second and higher moments of π_λ.

Having determined a subset of the values of λ used for which a linear approximation is appropriate, e.g. using the approach later described in Section 8.2.2, one can use linear least squares to carry out the regression. To account for the SMC estimates $\pi_{\lambda_p}^N(\varphi)$ having different variances, we propose the use of weighted least squares, with the 'observations' $\pi_{\lambda_p}^N(\varphi)$ assigned weights approximately inversely proportional to their variances. To compute these weights in practice, we propose using the SMC variance estimators first discussed in Section 1.5; we detail an approach to this in Section 8.2.1. A bias-corrected estimate of $\pi(\varphi)$ is then obtained by extrapolating the resulting fit to $\lambda = 0$, which corresponds to taking

the estimated intercept term.

To make this explicit, first consider the case in which $\varphi : \mathsf{E} \longrightarrow \mathbb{R}$, so that the estimates $\pi_\lambda^N(\varphi)$ are univariate. For each value λ_p denote the corresponding SMC estimate by $\eta_p \coloneqq \pi_{\lambda_p}^N(\varphi)$, and let v_p denote some proxy for the variance of this estimate. Then for some set of indices $S \coloneqq \{p^*, \dots, n\}$ chosen such that the relationship between η_p and λ_p is approximately linear for $p \in S$, we fit a linear model via weighted least squares, with weights proportional to $1/v_p$.

For the resulting fitted model, the slope parameter is computed as

$$\frac{\sum_{p \in S}(\lambda_p - \tilde{\lambda}_S)(\eta_p - \tilde{\eta}_S)/v_p}{\sum_{p \in S}(\lambda_p - \tilde{\lambda}_S)^2/v_p},$$

where $\tilde{\lambda}_S$ and $\tilde{\eta}_S$ denote weighted means given by

$$\tilde{\lambda}_S \coloneqq \frac{\sum_{p \in S} \lambda_p/v_p}{\sum_{p \in S} 1/v_p}, \qquad \tilde{\eta}_S \coloneqq \frac{\sum_{p \in S} \eta_p/v_p}{\sum_{p \in S} 1/v_p}.$$

A bias-corrected estimate for $\pi(\varphi)$ is obtained as the intercept of the fitted model, which is computed as

$$\pi_S^{\mathrm{BC}}(\varphi) \coloneqq \tilde{\eta}_S - \tilde{\lambda}_S \frac{\sum_{p \in S}(\lambda_p - \tilde{\lambda}_S)(\eta_p - \tilde{\eta}_S)/v_p}{\sum_{p \in S}(\lambda_p - \tilde{\lambda}_S)^2/v_p}. \tag{8.3}$$

The formal justification of this estimate assumes that the observations are uncorrelated, which does not hold here. We demonstrate in Section 9.1.2 how this simple approach can nevertheless be effective. In principle, however, one could use generalised least squares combined with some approximation of the full covariance matrix of the SMC estimates.

In the more general case where $\varphi : \mathsf{E} \longrightarrow \mathbb{R}^d$ for $d > 1$, we propose simply evaluating (8.3) for each component of this quantity separately, which corresponds to fitting an independent weighted least squares regression to each component. This facilitates the use of the variance estimators described in the following section, though in principle one could use multivariate weighted least squares or other approaches.

8.2.1. Variance estimation for weighted least squares

We propose the weighted form of least squares here since, as the values of λ used in the SMC procedure approach zero, the estimators generated may increase in variance: partly due to poorer mixing of the MCMC kernels as previously described, but also due to the gradual degeneracy of the particle set. In order to estimate the variances of the SMC estimators one may use any of the approaches described in Section 1.5, which use the genealogy of the particles to compute such an estimator using only the observed realisation of the particle filter. Using any such procedure, one may estimate the variance of $\pi_\lambda^N(\varphi)$ for each λ value considered by Algorithm 8.1, with these values used for each v_p in (8.3).

Within our examples, we use the estimator V_p^N proposed by Lee and Whiteley (2018) and previously introduced in Section 1.5; for fixed N this coincides with an earlier proposal of Chan and Lai (2013) up to a multiplicative constant. Specifically, after the pth step of the SMC sampler we compute $V_p^N(\varphi - \eta_p^N(\varphi))$ according to (1.40); when multiplied by N this provides a consistent estimator of the *asymptotic* variance of each estimate $\pi_{\lambda_p}^N(\varphi)$, as defined in (1.25). While this is not equivalent to computing the true variance of each estimate, for fixed large N the relative sizes of these estimates should provide a useful indicator of the relative variances of each estimate $\pi_\lambda^N(\varphi)$.

In Section 9.1.2 we show empirically that inversely weighting the SMC estimates according to these estimated variances can result in more stable bias-corrected estimates as the particle set degenerates. We also explain in Section 8.3 how these estimated variances can be used within a rule to determine when to terminate the algorithm.

The asymptotic variance estimator described is consistent in N. However, if in practice resampling at the pth time step causes the particle set to degenerate to having a single common ancestor, then the estimator evaluates to zero, and so it is impossible to use this value as the inverse weight v_p in (8.3). Such an outcome may be interpreted as a warning that too few particles have been used for the resulting SMC estimates to be reliable, and that a greater number should be used when re-running the procedure. An alternative would be to use the fixed-lag estimators of Olsson and Douc (2019), which as previously discussed in Section 1.5 benefit from improved numerical stability in such scenarios.

8.2.2. Determining a subset of estimates to use for linear regression

If the local linear regression approach for bias correction is used, then the practitioner must determine a value of λ below which the dependence of $\pi_\lambda(\varphi)$ on λ is approximately linear. For this purpose, we propose a heuristic based on the coefficient of determination, commonly denoted R^2; here, this may be thought of as the proportion of the variance of the observed values of $\pi_\lambda^N(\varphi)$ that is explained by an assumed linear dependence on λ.

To define this explicitly, consider the weighted least squares fit for which (8.3) is the resulting bias-corrected estimate. Extending the notation used therein, let $\hat{\eta}_p^S$ denote the predicted value of η_p under the model, which is computed as

$$\hat{\eta}_p^S := \tilde{\eta}_S - (\tilde{\lambda}_S - \lambda_p)\frac{\sum_{q\in S}(\lambda_q - \tilde{\lambda}_S)(\eta_q - \tilde{\eta}_S)/v_q}{\sum_{q\in S}(\lambda_q - \tilde{\lambda}_S)^2/v_q}.$$

Then the coefficient of determination R_S^2 for this weighted least squares model fit may be computed as the ratio of the weighted sum of squared errors and the weighted total sum of squares. That is,

$$R_S^2 := \frac{\sum_{p\in S}(\hat{\eta}_p^S - \tilde{\eta}_S)^2/v_p}{\sum_{p\in S}(\eta_p - \tilde{\eta}_S)^2/v_p} = 1 - \frac{\sum_{p\in S}(\eta_p - \hat{\eta}_p^S)^2/v_p}{\sum_{p\in S}(\eta_p - \tilde{\eta}_S)^2/v_p}. \tag{8.4}$$

Algorithm 8.2 Linear regression inclusion procedure for SMC bias correction

For some test function $\varphi : \mathsf{E} \longrightarrow \mathbb{R}$:

1. Complete Algorithm 8.1, generating and storing estimates $\eta_p := \pi^N_{\lambda_p}(\varphi)$ using the particle approximations (8.1), and estimates v_p of their variances, for $p \in \{0, \dots, n\}$.
2. Initialise the set of indices of estimates to be used in regression as $S \longleftarrow \{0, \dots, n\}$.
3. Regress η_p against λ_p using weighted least squares, with weights $1/v_p$, for $p \in S$. Compute the coefficient of determination R^2_S according to (8.4).
4. If $|S| \leq 3$, proceed to Step 6. Else, set $S' \longleftarrow S \setminus \{\min(S)\}$, and regress η_p against λ_p using weighted least squares, with weights $1/v_p$, for $p \in S'$. Compute $R^2_{S'}$ according to (8.4).
5. If $R^2_{S'} > R^2_S$, set $S \longleftarrow S'$, and return to Step 4. Otherwise, proceed to Step 6.
6. Return the bias-corrected estimate $\pi^{\mathrm{BC}}_S(\varphi)$, computed according to (8.3).

The heuristic procedure for determining such a subset of the estimates is presented in Algorithm 8.2. After completion of Algorithm 8.1, one conducts weighted least squares in the manner described in Section 8.2, including all values of λ_p and the corresponding SMC estimates $\pi^N_{\lambda_p}(\varphi)$, and computing the R^2 value for the resulting fit. One then re-conducts the regression, without the observation in the subset corresponding to the largest λ value. If this results in a greater R^2 value, this observation should henceforth be excluded from the least squares regression. One continues to apply this procedure, each time repeating the regression without the observation corresponding to the highest remaining value of λ, until doing so no longer results in a model with a greater R^2 value than the current fit. The regression fit at this point may then be used to compute the bias-corrected estimate for $\pi(\varphi)$, so that in (8.3), S corresponds to the set of indices of the remaining λ values.

The motivation for this approach is that if this largest λ value is not sufficiently close to zero for $\pi_\lambda(\varphi)$ to be approximately linear in λ, then retaining the corresponding SMC estimate in the regression may result in a large proportion of the variance in the data being unexplained by a linear dependence. By excluding the corresponding SMC estimate, one would expect the linear fit applied to the remaining estimates to better describe their variance, and therefore to have a greater R^2 value.

This heuristic approach has a natural online implementation, allowing a bias-corrected estimate to be computed after each step of the algorithm. We use this online form within our proposed stopping rule in Section 8.3, in which it forms Step 3 of Algorithm 8.3. Specifically, we maintain a set of the SMC estimates to be used in the regression (and the corresponding values of λ), initialising this to be empty. After the pth step of the SMC sampler, the newly-generated SMC estimate $\pi^N_{\lambda_p}(\varphi)$ is added to this set (with the corresponding λ_p). One conducts weighted least squares on this set of estimates and then, as long as

the set contains more than 3 estimates, one proceeds in the manner described above, reconducting the regression without the observation in the set corresponding to the highest value of λ. If this results in a fit with a higher R^2 value, then the omitted SMC estimate is henceforth excluded from the set used for regression, and this step is repeated. If not, then one terminates this procedure and proceeds to the next step of the SMC sampler.

8.3. Stopping rule

As λ approaches zero we expect the bias resulting from estimating $\pi(\varphi)$ by $\pi_\lambda^N(\varphi)$ to decrease, while the variance of the resulting estimators may increase due to poorer mixing of the associated Markov kernels. Based on the bias–variance decomposition (7.10) of the mean squared error, we here propose a procedure for determining when to terminate the SMC sampler, in order to achieve such a bias–variance trade-off. Since the mean squared error is only well-defined when $\varphi : \mathsf{E} \longrightarrow \mathbb{R}$, so that the estimates $\pi_\lambda^N(\varphi)$ are univariate, we first describe the stopping rule in this setting. We subsequently describe a possible extension to multivariate functions $\varphi : \mathsf{E} \longrightarrow \mathbb{R}$, based on a simple generalisation of the mean squared error.

At each stage, having computed an updated bias-corrected estimate via the online procedure described in Section 8.2.2, one may subtract this value from each of the SMC estimates generated so far in order to produce an estimate of the bias in each case. As discussed in Section 8.2, we also have an estimate of the variance of each SMC estimate, as used in the weighted linear regression procedure. As such, at each stage we are able to estimate the mean squared error (MSE) of each SMC estimate so far generated, by squaring each estimate of the bias and adding the appropriate estimate of the variance.

The formation of these mean squared error estimates is based on, but does not exactly correspond to, the bias–variance decomposition (7.10). For example, the particle-based SMC estimates $\pi_\lambda^N(\varphi)$ are not unbiased as estimators of $\pi(\varphi)$, although by Proposition 1.3 they are consistent in the number of particles N. Furthermore, the bias-corrected estimate itself is not unbiased, since it is formed based on approximate local linearity rather than a true linear dependency of $\pi_\lambda^N(\varphi)$ on λ. Nonetheless, the use of this heuristic approach in our proposed stopping rule has been found to work well in practice, resulting in estimates of low mean squared error; we discuss one such example in Section 9.1.2.

Note that, since the bias-corrected estimate of $\pi(\varphi)$ is updated after each step (to take into account the most recent estimate), the estimated mean squared errors of all previous estimates may also all be updated after each step. After each SMC estimate is generated we may therefore determine which SMC estimate, of all those generated so far, has the lowest estimated mean squared error. We propose that, for some κ, the SMC sampler should be terminated after the same previous estimate is found to have the lowest mean squared error of all those generated so far, for κ consecutive iterations. Following the termination

of the algorithm via the stopping rule this SMC estimate, which has been consistently found to have the lowest estimated MSE, may be returned as the final estimate for $\pi(\varphi)$.

Algorithm 8.3 Global consensus Monte Carlo: SMC algorithm with stopping rule

For some test function $\varphi : \mathsf{E} \longrightarrow \mathbb{R}$ and stopping rule parameter κ:

1. Initialise the time index as $p \longleftarrow 0$, and the set of indices of estimates to be used in regression as $S \longleftarrow \emptyset$.
2. Complete the pth iteration of Algorithm 8.1, generating and storing an estimate $\eta_p := \pi^N_{\lambda_p}(\varphi)$ using the particle approximation (8.1), and an estimate v_p of its variance.
3. Set $S \longleftarrow S \cup \{p\}$. If $|S| > 1$:
   a) Regress η_q against λ_q using weighted least squares, with weights $1/v_q$, for $q \in S$. Compute the coefficient of determination R^2_S according to (8.4).
   b) If $|S| \leq 3$, proceed to Step 4. Else, set $S' \longleftarrow S \setminus \{\min(S)\}$, and regress η_q against λ_q using weighted least squares, with weights $1/v_q$, for $q \in S'$. Compute $R^2_{S'}$ according to (8.4).
   c) If $R^2_{S'} > R^2_S$, set $S \longleftarrow S'$, and return to Step 3b. Otherwise, proceed to Step 4.
4. Set $m_p \longleftarrow \pi^{\mathrm{BC}}_S(\varphi)$, a bias-corrected estimate computed according to (8.3).
5. Set
$$i_p \longleftarrow \underset{q \in \{0,\dots,p\}}{\arg\min} \left[(\eta_q - m_p)^2 + v_q\right],$$
which corresponds to taking the index of the SMC estimate with the lowest estimated mean squared error (MSE).
6. If $p > \kappa$, and $(i_{p-\kappa+1}, \dots, i_p)$ are all equal, terminate the algorithm, returning the estimate η_{i_p} of lowest estimated MSE (and/or m_p, the final bias-corrected estimate). Else, set $p \longleftarrow p + 1$ and return to Step 2.

This approach is described in Algorithm 8.3. In our simulation studies, we found that taking $\kappa = 15$ worked well in balancing robustness with the computational complexity of the resulting algorithm. We present the results of such experiments for a simple model in Section 9.1.2.1.

As an alternative, one may choose to return the final bias-corrected estimate, for which this approach also provides a justifiable stopping rule: consistently finding that the same previous estimate has the lowest MSE suggests stability in our estimates of the MSEs of each previous $\pi^N_\lambda(\varphi)$, and therefore in the bias-corrected estimate. Furthermore, since we expect the biases of the estimates $\pi^N_\lambda(\varphi)$ to decrease as λ approaches zero, consistently finding that a previous SMC estimate has the lowest MSE suggests that more recent estimates are of higher variances. Again, this is also indicative of a stabilisation of the bias-corrected estimate, since new observations are included in the regression-based bias correction pro-

cedure with weights inversely proportional to these variances.

Finally, we discuss how this might be generalised in settings where $\varphi : \mathsf{E} \to \mathbb{R}^d$ for $d > 1$, so that the estimates $\pi_\lambda^N(\varphi)$ are multivariate. As previously discussed in Section 8.2, a bias-corrected estimate for each component of π_λ may be computed separately, by conducting independent regressions. Within Algorithm 8.3, this would require maintaining a separate set of regression indices S for each component.

In a multivariate setting, the mean squared L_2 error corresponds to the sum of the mean squared errors in each of the d components. Within Step 5 of Algorithm 8.3, one could therefore estimate the MSE of each component of each previously-computed SMC estimate, and take the sum. One would similarly determine the index of the previously-computed estimate for which this is lowest, stopping once this has remained unchanged for κ iterations.

8.4. Summary

The SMC sampler presented here may be computationally intensive, and may therefore be most useful in lower-dimensional settings. However, as discussed it has the benefit of allowing communication costs to be combined across particles, and avoids the need to specify a single value of the regularisation parameter λ. We present numerical demonstrations of this SMC approach, and of the associated heuristic procedures we propose, in the following chapter.

9. Examples and applications

9.1. Simple Gaussian models

Within this final chapter we present results from simulation studies of our proposed algorithms, in order to illustrate some of the properties discussed in the previous chapters, and to compare their performance to other simulation approaches when applied to realistic examples. To illustrate the role of λ in our framework and to supplement the theoretical analysis of Section 7.4, we begin with some numerical results based on univariate Gaussian models.

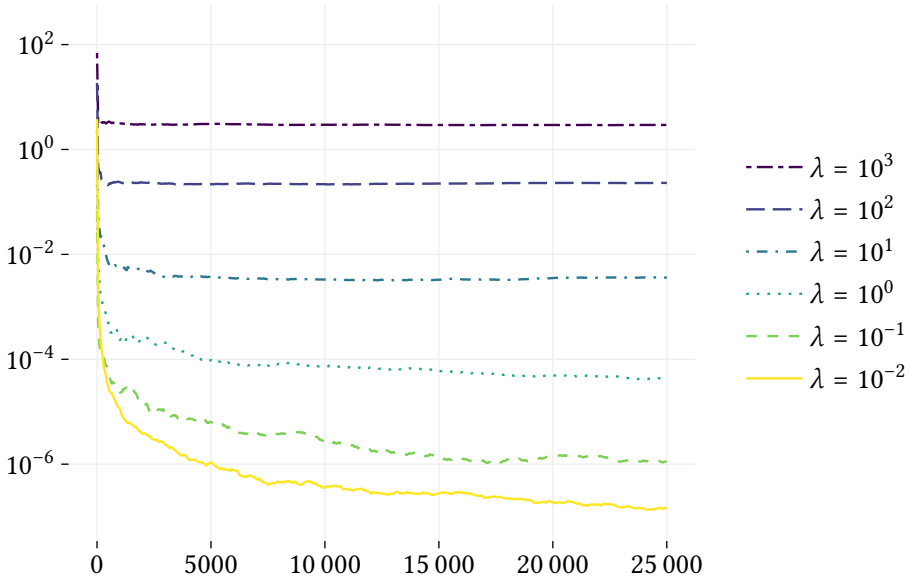
9.1.1. Gibbs sampler

For our first Gaussian example we drew $n = 20\,000$ IID samples from a normal distribution with mean 12.4 and variance 10, splitting these into $b = 4$ equal blocks of size 5000. The partial likelihood terms f_j therefore take the form (7.16). For the prior density we use $\mu(z) = \mathcal{N}(z; 10, 100)$, and for the Markov transition kernels we use $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, \lambda)$. We ran Algorithm 7.1 as a Gibbs sampler, i.e. such that the kernels (7.8)–(7.9) corresponded to the drawing of IID samples exactly from the full conditional distributions (7.6)–(7.7). For each of various λ values we obtained a chain of length $N = 25\,000$, repeating this for a total of 25 replicates.

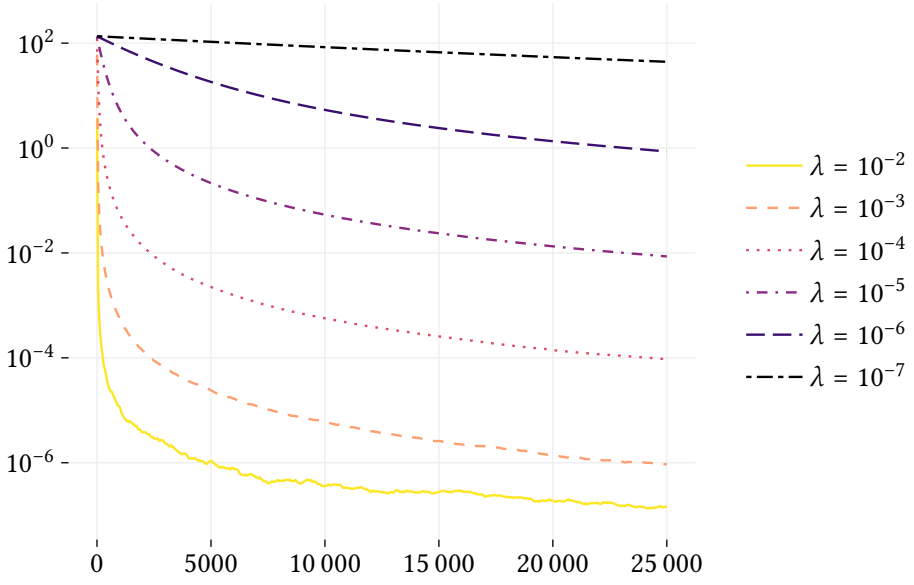
We consider the problem of estimating the posterior mean $\pi(\mathrm{Id})$, as described in Section 7.4.1. Figure 9.1 shows, for each value of λ used, the behaviour of the mean squared error of the estimator $\pi_\lambda^N(\mathrm{Id})$ as a function of the number of samples N. We see in Figure 9.1a that for larger values of λ, this becomes approximately constant once the number of samples N is sufficiently large, since this becomes dominated by the squared bias. For the smallest values of λ, presented in Figure 9.1b, the chains mix poorly due to high autocorrelation, and so the MSE decreases slowly. Similar behaviour may also be observed in the later examples of this chapter.

The role of λ in achieving a bias–variance trade-off is evident in these results. We see that when $N = 25\,000$ samples are used, the choice of $\lambda = 10^{-2}$ (presented in both Figure 9.1a and Figure 9.1b) results in the estimator $\pi_\lambda^N(\mathrm{Id})$ of lowest MSE, among all those values of λ considered. This is close to the approximately optimal value obtained from (7.22), which for this model evaluates to approximately 0.0104.

As discussed in Section 7.3.2, in settings where the blocks of data differ in size, it may



(a) Values of λ between 10^{-2} and 10^3.



(b) Values of λ between 10^{-7} and 10^{-2}.

Figure 9.1.: Mean squared errors of estimates $\pi_\lambda^N(\mathrm{Id})$, plotted on a logarithmic scale against the number of samples N, as obtained by the Gibbs sampler form of the global consensus MCMC algorithm applied to the Gaussian example model. Results are shown for various choices of the regularisation parameter λ; in each case, MSE values are computed over 25 replicates of the algorithm.

be beneficial to choose the Markov transition densities $K_j^{(\lambda)}$ to have relative scales that are inversely proportional to these block sizes. To demonstrate this, we repartitioned the data into $b = 4$ blocks of size $n_1 = 1000$, $n_2 = 3000$, $n_3 = 6000$ and $n_4 = 10\,000$. For the Markov transition densities we take $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, c_j\lambda)$ as in (7.11), considering two possible choices for the values c_j:

- $c_j = c/n_j$ for $j \in \{1, \dots, b\}$. In this case, the smoothed partial likelihood terms (7.2) take the form $f_j^{(\lambda)}(z) \propto \mathcal{N}(\bar{y}_j;\, z,\, (\sigma^2 + c\lambda)/n_j)$ following (7.13).
- $c_j = 1$ for $j \in \{1, \dots, b\}$. In this case, $f_j^{(\lambda)}(z) \propto \mathcal{N}(\bar{y}_j;\, z,\, (\sigma^2 + \lambda n_j)/n_j)$.

Recall from Section 7.3.2 that for the first of these choices, the effect is to 'dilute' the information from each observation in a consistent way, as seen in the resulting form of $f_j^{(\lambda)}$. In order to make a fair comparison between these two choices, we choose the constant c so that the average 'dilution' of each observation is the same in both cases. Here, this occurs when $c = \sum_{j=1}^{b} n_j^2/n = 7300$.

For each of these two choices of the Markov transition densities $K_j^{(\lambda)}$ we again ran the Gibbs sampler form of Algorithm 7.1 to obtain a chain of length $N = 25\,000$, repeating this for a total of 25 replicates in each case. We show in Figure 9.2 the MSE of $\pi_\lambda^N(\mathrm{Id})$ as a function of N, in the case $\lambda = 10^{-3}$. We see that estimates of lower MSE are obtained by choosing the scales c_j to be inversely proportional to the block sizes, as recommended.

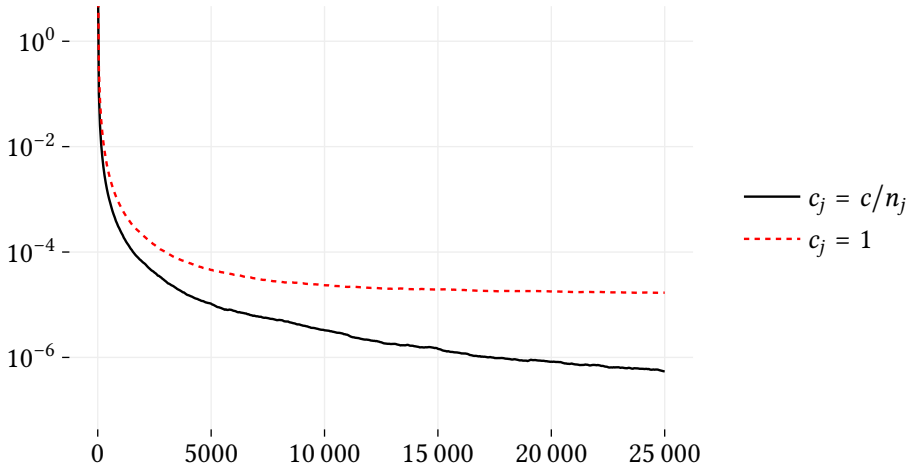


Figure 9.2.: Mean squared errors of estimates $\pi_\lambda^N(\mathrm{Id})$, plotted on a logarithmic scale against the number of samples N, as obtained by the Gibbs sampler form of the global consensus MCMC algorithm applied to the Gaussian example model with differing block sizes. Here, we use Markov transition densities $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, c_j\lambda)$, $j \in \{1, \dots, b\}$, with $\lambda = 10^{-3}$ and with two different choices of $c_{1:b}$ as explained in the main text. In each case, MSE values are computed over 25 replicates of the algorithm.

9.1.2. SMC sampler and bias correction procedure

To demonstrate the SMC sampler proposed in Section 8.1, and the bias correction technique described in Section 8.2, we again study a univariate Gaussian model of the form described in Section 7.4, with the aim of estimating the posterior first moment $\pi(\mathrm{Id})$.

We consider here a case with $b = 32$, taking $f_j(z) = \mathcal{N}(\mu_j; z, 1)$ for $j \in \{1, \ldots, b\}$, with the values μ_j drawn independently from a normal distribution with mean 4 and variance 1. For the Markov transition kernels we use $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, \lambda)$. For the purposes of illustrating the local linear regression approach to bias correction we consider the (quite concentrated) prior density $\mu(z) = \mathcal{N}(z; 4, 1)$. In this case, we see that the dependence of $\pi_\lambda(\mathrm{Id})$ on λ is highly non-linear on the range $\lambda \in (0, 1000)$, as shown in Figure 9.3a.

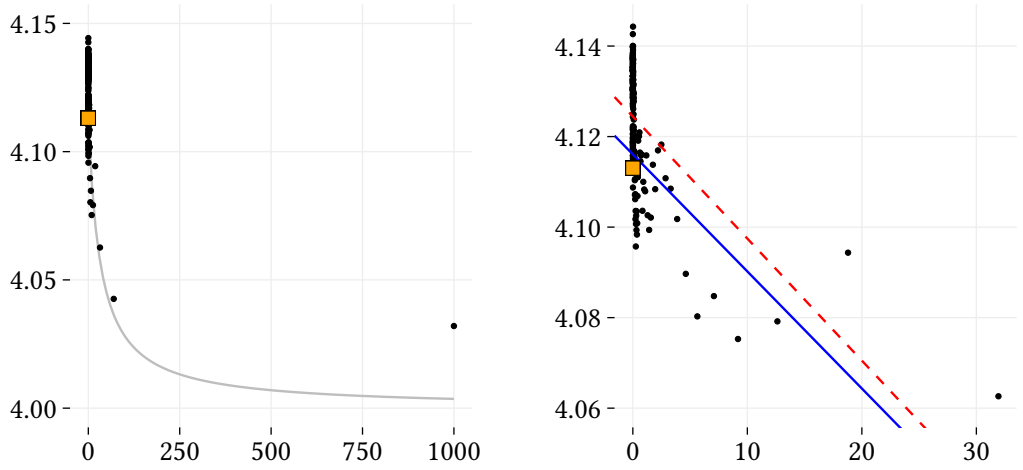
We constructed an SMC sampler using $N = 2500$ particles; we used sequences of λ values beginning with $\lambda_0 = 1000$, with subsequent values determined adaptively according to the procedure proposed by Zhou et al. (2016), for which we used parameter $\mathrm{CESS}^\star = 0.95N$. For the purposes of illustrating the bias correction technique we here consider sequences of λ values of fixed length $n = 200$; we will describe the use of the proposed stopping rule subsequently.

To construct Markov kernels invariant with respect to each distribution $\tilde{\pi}_\lambda$, we used Gibbs kernels constructed in the manner of Algorithm 7.1. That is to say that in each time step of the SMC sampler (i.e. for each value of λ) and for each particle, each of $X_{1:b}$ was updated by drawing exactly from its conditional distribution, after which Z was updated similarly.

Figure 9.3a shows the SMC estimate $\pi_\lambda^N(\mathrm{Id})$ obtained for each λ, in a single run of this algorithm. To determine a subset of these estimates to be used for local linear regression, we used the approach described in Section 8.2.2; the resulting subset is displayed in Figure 9.3b. In this case, we see that for the smallest values of λ considered, the estimates exhibit increased variance, due to the poorer mixing of the Markov kernels, and the degeneracy of the particle set.

As described in Section 8.2.1, when conducting local least squared regression we weight each estimate in inverse proportion to its estimated (asymptotic) variance. For the estimates plotted in Figure 9.3b, these relative weights are presented in Figure 9.4, with λ on a log scale for clarity. The resulting weighted least squares fit is overplotted in Figure 9.3b, together with the corresponding *unweighted* (ordinary) least squares fit. We see that for these results, the weighted least squares fit better reflects the local linear dependence on λ, being less influenced by the high-variance estimates near 0, which correspondingly carry less weight in the regression.

As discussed in Section 8.1, we may view the SMC sampler as a method to improve or 'refine' the estimator that would be formed using the initial set of particles, i.e. $\pi_{\lambda_0}^N(\mathrm{Id})$, where $\lambda_0 = 1000$. A straightforward choice of such a refined estimator would therefore be the SMC estimate $\pi_{\lambda_n}^N(\mathrm{Id})$ corresponding to λ_n, the final (smallest) λ value considered. We ran



(a) All estimates $\pi_\lambda^N(\mathrm{Id})$, and the true value of $\pi_\lambda(\mathrm{Id})$ as a function of λ (solid grey line).

(b) Subset of estimates used for local linear regression, and linear fits using least squares: weighted (solid blue line) and unweighted (dashed red line).

Figure 9.3.: Estimates $\pi_\lambda^N(\mathrm{Id})$ plotted against λ, as obtained at each step of a single run of the SMC sampler for the Gaussian example model. The orange square indicates the true value of $\pi(\mathrm{Id}) \approx 4.113$.

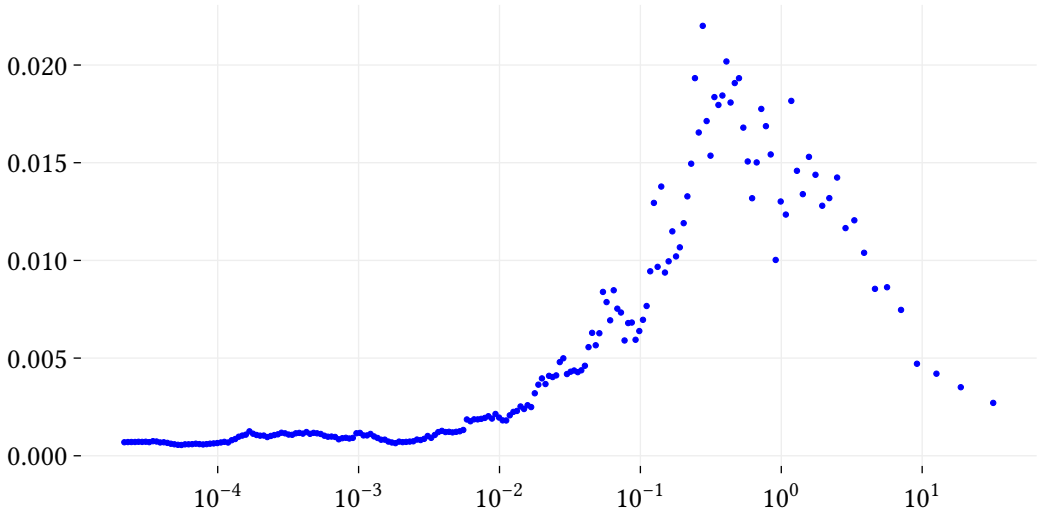


Figure 9.4.: For the estimates $\pi_\lambda^N(\mathrm{Id})$ plotted in Figure 9.3b, the estimates' relative weights as used in the weighted least squares bias correction technique, plotted against λ on a logarithmic scale.

Estimate	Mean squared error
Initial SMC estimate	1.32×10^{-2}
Final SMC estimate	1.13×10^{-3}
Bias-corrected estimate, using WLS	3.60×10^{-5}
Bias-corrected estimate, using OLS	2.57×10^{-4}

Table 9.1.: For the Gaussian example model, the mean squared error of four estimators of $\pi(\mathrm{Id})$: the SMC estimate $\pi^N_{\lambda_0}(\mathrm{Id})$ corresponding to the initial (largest) λ value; the estimate $\pi^N_{\lambda_n}(\mathrm{Id})$ corresponding to the final (smallest) λ value; the bias-corrected estimate (8.3) computed using weighted least squares (WLS); and the analogous estimate resulting from using unweighted ordinary least squares (OLS). All MSE values are computed over 25 replicates of an SMC sampler using a sequence of λ values of fixed length $n = 200$.

the SMC sampler 25 times; the value of λ_n varied between runs due to the adaptive specification of the sequence of distributions, but each time was approximately 2.2×10^{-5}. For each run of the SMC sampler we also computed a bias-corrected estimate (8.3) of $\pi(\mathrm{Id})$ using weighted least squares as described above; that is, the intercept of the local least squares linear fit. Additionally, for purposes of comparison we also computed a bias-corrected estimate using ordinary (unweighted) least squares.

The mean squared error of each such estimate is presented in Table 9.1. The weighted least squares approach was observed to result in a rather lower MSE than the simpler approach of considering solely the final λ value. Using unweighted least squares to compute a bias-corrected estimate resulted in an MSE between these two values.

9.1.2.1. Using the proposed stopping rule

We subsequently considered the effects of using the stopping rule proposed in Section 8.3, retroactively applying the procedure of Algorithm 8.3 to each of the 25 simulations. As previously described, our proposed stopping rule requires the specification of a parameter κ. After every iteration of the SMC sampler, a bias-corrected estimate of $\pi(\varphi)$ is computed, which is used to estimate the MSE of each previously-computed SMC estimate $\pi^N_{\lambda_p}(\mathrm{Id})$. If the same previously-computed estimate is found to have the lowest estimated MSE for κ successive iterations, one stops and returns this estimate.

In Figure 9.5 we show, for values of κ ranging from 3 to 20, the average index p corresponding to the optimal SMC estimate $\pi^N_{\lambda_p}(\mathrm{Id})$, as determined using the stopping rule. We see that this stabilises once κ is greater than about 10. We also present in Figure 9.5 the average number of SMC iterations required before termination of the algorithm, for each value of κ considered; note that this is necessarily an increasing function of κ. This supports our previous recommendation to take $\kappa = 15$, balancing the robustness of the resulting estimator with the computational cost of the SMC sampler.

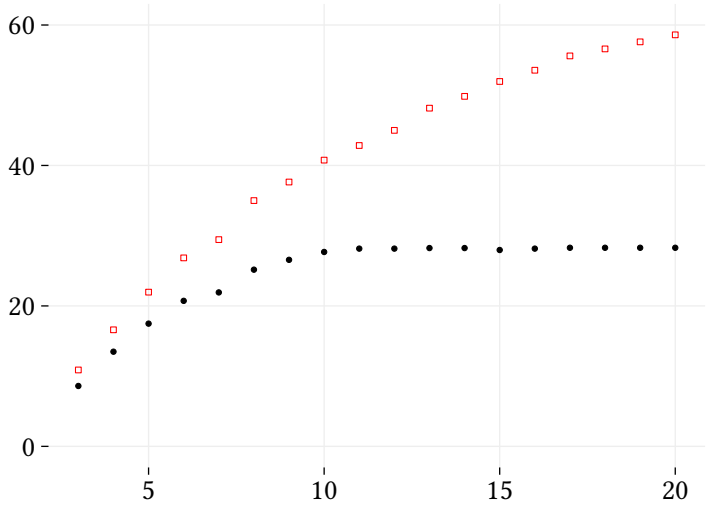


Figure 9.5.: For various values of the stopping rule parameter κ (on the horizontal axis), the average index p corresponding to the SMC estimate $\pi^N_{\lambda_p}(\mathrm{Id})$ of lowest estimated MSE, as determined using the stopping rule (filled black circles); and the average number of SMC iterations required before termination (red squares). All averages are computed over 25 replicates of an SMC sampler applied to the Gaussian example model.

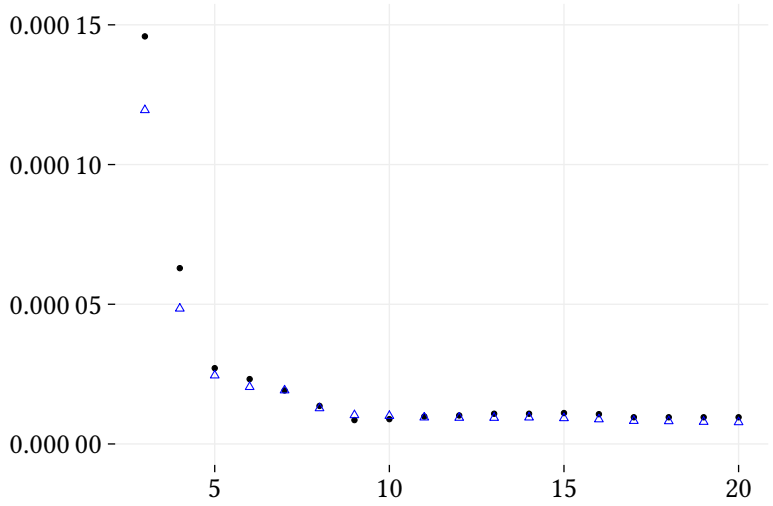


Figure 9.6.: For various values of the stopping rule parameter κ (on the horizontal axis), the mean squared error of two estimators of $\pi(\mathrm{Id})$: the SMC estimate $\pi^N_{\lambda_p}(\mathrm{Id})$ of lowest *estimated* MSE, as determined using the stopping rule (filled black circles); and the bias-corrected estimate at the time of termination by the stopping rule (blue triangles). All MSE values are computed over 25 replicates of an SMC sampler applied to the Gaussian example model.

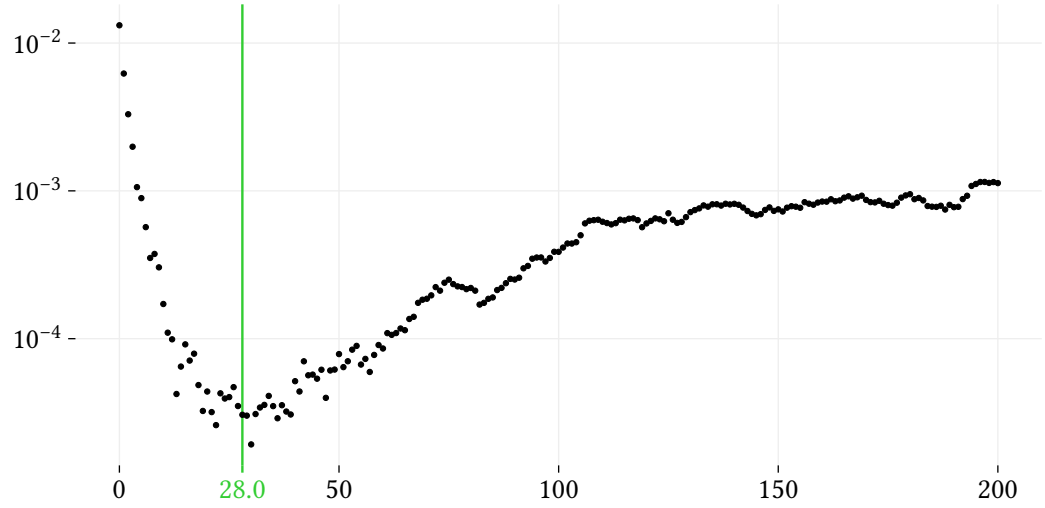


Figure 9.7.: Mean squared errors of estimates $\pi^N_{\lambda_p}(\mathrm{Id})$ of $\pi(\mathrm{Id})$ plotted against p, as obtained at each step of an SMC sampler for the Gaussian example model. The green vertical line represents the average index p corresponding to the SMC estimate of lowest estimated MSE, as determined using the stopping rule with $\kappa = 15$. All MSE values are computed over 25 replicates, with the sequences of values λ_p determined adaptively according to the description in the main text.

In Figure 9.6 we show for each value of κ the *true* MSE corresponding to the estimator chosen by the stopping rule. Again, we see that our previously-suggested choice of $\kappa = 15$ provides a good balance, here between a low mean squared error and the total computational cost.

A natural question is whether this estimator, chosen for having the lowest estimated MSE, does indeed have the lowest true MSE of all estimators computed by the SMC sampler. For our simulations in which we used sequences of λ values of length $n = 200$, Figure 9.7 shows for each $p \in \{0, \dots, n\}$ the true MSE of each estimator $\pi^N_{\lambda_p}(\mathrm{Id})$, as computed over the 25 replicates. Note that since the sequences of values λ_p were chosen adaptively, these were not identical for each simulation, though they were very similar in practice. We see that the minimal true MSE belongs to the estimator $\pi^N_{\lambda_p}(\mathrm{Id})$ for which $p = 30$; for each run the corresponding λ value was approximately 0.5. The estimator returned by the stopping rule (with $\kappa = 15$) corresponds on average to $p = 28.0$, which is seen in Figure 9.7 to be within the region of lowest mean squared error.

As discussed in Section 8.3, an alternative to returning the estimator of lowest estimated MSE is to return the final bias-corrected estimate, i.e. that computed using all iterations at the time of termination. In Figure 9.6 we show for each value of the stopping rule parameter κ the MSE corresponding to this final bias-corrected estimator, which is seen to behave largely similarly to the stopping-rule-based estimator previously discussed.

Finally, we show in Table 9.2 the mean squared error of the two estimates described here,

Estimate	Mean squared error
SMC estimate of lowest estimated MSE	1.11×10^{-5}
Bias-corrected estimate	9.23×10^{-6}

Table 9.2.: For the Gaussian example model, the mean squared error of two estimators of $\pi(\mathrm{Id})$: the estimate of lowest *estimated* MSE, as determined using the procedure described in Algorithm 8.3; and the final bias-corrected estimate (8.3) computed using weighted least squares. Both MSE values are computed over 25 replicates of an SMC sampler, terminated according to the proposed stopping rule with parameter $\kappa = 15$.

when using the stopping rule with $\kappa = 15$. These MSE values are comparable, and lower than the values in Table 9.1, for which $n = 200$ iterations were used. We see therefore that using the stopping rule to choose the SMC estimate of lowest estimated MSE here results in a superior estimator to that obtained by simply taking the final SMC estimate after a fixed number of iterations. We also find that the bias-corrected estimate here performs slightly better than the corresponding estimate obtained after using 200 iterations: in that case, the SMC estimates corresponding to the very smallest values of λ are of high variance and can distort the regression, despite being appropriately weighted.

9.2. Log-normal model

To compare the posterior approximations formed by the global consensus algorithm described in Section 7.2 with those formed by some of the embarrassingly parallel approaches discussed in Section 6.3.1, we conduct a simulation study based on a simple model. Let $\mathcal{LN}(x; \mu, \sigma^2)$ denote the density of a log-normal distribution with parameters (μ, σ^2); that is,

$$\mathcal{LN}(x; \mu, \sigma^2) = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log(x) - \mu)^2}{2\sigma^2}\right).$$

One may consider a model with prior density $\mu(z) = \mathcal{LN}(z; \mu_0, \sigma_0^2)$ and likelihood contributions $f_j(z) = \mathcal{LN}(\log(\mu_j); \log(z), \sigma_j^2)$ for $j \in \{1, \ldots, b\}$. This may be seen as a reparametrisation of the Gaussian model analysed in Section 7.4, in which each likelihood contribution is that of a data subset with a Gaussian likelihood. This convenient setting allows for the target distribution π to be expressed analytically. For the implementation of the global consensus algorithm, we choose Markov transition kernels given by $K_j^{(\lambda)}(z, x) = \mathcal{LN}(x; \log(z), \lambda)$ for each $j \in \{1, \ldots, b\}$, which satisfy Assumption 7.1; this allows for exact sampling from all the full conditional distributions.

As a toy example to illustrate the effects of non-Gaussian partial likelihoods we consider a case in which $f_j(z) = \mathcal{LN}(\log(\mu_j); \log(z), 1)$ for each j, and $\mu(z) = \mathcal{LN}(z; 0, 25)$. Here we took $b = 32$, and selected the location parameters μ_j as IID samples from a standard normal distribution. We ran global consensus Monte Carlo (GCMC) using the Gibbs sampler form

Algorithm		Mean ± MCSE of estimate of $\pi(\varphi)$		
		$\varphi(z) = z$ $[\pi(\varphi) = 1.141]$	$\varphi(z) = z^5$ $[\pi(\varphi) = 2.644]$	$\varphi(z) = \log(z)$ $[\pi(\varphi) = 0.1164]$
GCMC	$\lambda = 10^{1}$	1.329 ± 0.003	121.154 ± 10.487	0.1151 ± 0.0019
	$\lambda = 10^{0}$	1.159 ± 0.002	3.901 ± 0.037	**0.1165 ± 0.0014**
	$\lambda = 10^{-1}$	**1.144 ± 0.003**	**2.763 ± 0.044**	0.1173 ± 0.0030
	$\lambda = 10^{-2}$	1.140 ± 0.011	2.648 ± 0.143	0.1150 ± 0.0090
	$\lambda = 10^{-3}$	1.142 ± 0.022	2.661 ± 0.295	0.1191 ± 0.0199
	$\lambda = 10^{-4}$	1.120 ± 0.077	3.505 ± 1.136	0.1630 ± 0.0651
	$\lambda = 10^{-5}$	1.400 ± 0.110	6.195 ± 2.217	0.3283 ± 0.0810
CMC		1.073 ± 0.010	16.092 ± 5.675	0.0135 ± 0.0095
NDPE		1.148 ± 0.029	2.800 ± 0.385	0.1231 ± 0.0246
WRS		1.111 ± 0.007	2.444 ± 0.086	0.0862 ± 0.0063

Table 9.3.: True values and estimates of $\pi(\varphi)$, for various test functions φ, for the first log-normal toy model. Estimates obtained using global consensus Monte Carlo (GCMC) with various values of λ, and three embarrassingly parallel methods (CMC, NDPE, WRS; see main text for descriptions). For each method the mean estimate ± Monte Carlo standard error (MCSE) is presented, as computed over 25 replicates; the estimator corresponding to the lowest mean squared error is printed in bold.

of Algorithm 7.1, for values of λ between 10^{-5} and 10. For comparison with embarrassingly parallel methods we also drew samples from each subposterior distribution as defined in (6.8), combining the samples using various approaches. These are:

- the consensus Monte Carlo (CMC) averaging of Scott et al. (2016);
- the nonparametric density product estimation (NDPE) approach of Neiswanger et al. (2014);
- the Weierstrass rejection sampling (WRS) combination technique of Wang and Dunson (2013), using their R implementation (github.com/wwrechard/weierstrass).

In each case we ran the algorithm 25 times, drawing $N = 10^5$ samples.

To demonstrate the role of λ in the bias–variance decomposition (7.10), Table 9.3 presents the means and standard deviations of estimates of $\pi(\varphi)$, for various test functions φ. In estimating the first moment of π, GCMC generates a low-bias estimator when λ is chosen to be sufficiently small; however, as expected, the variance of such estimators increases when very small values of λ are chosen. While the other methods produce estimators of reasonably low variance, these exhibit somewhat higher bias. For CMC the bias is especially pronounced when estimating higher moments of the posterior distribution, as exemplified by the estimates of the fifth moment. Note however that high biases are also

Algorithm		Mean ± MCSE of estimate of $\pi(\varphi)$	
		$\varphi(z) = z$ $[\pi(\varphi) = 1.011\,39]$	$\varphi(z) = \log(z)$ $[\pi(\varphi) = 0.011\,32]$
GCMC	$\lambda = 10^{1}$	1.180 22 ± 0.002 097	0.011 21 ± 0.001 537
	$\lambda = 10^{0}$	1.027 40 ± 0.000 609	0.011 42 ± 0.000 623
	$\lambda = 10^{-1}$	1.013 05 ± 0.000 156	0.011 40 ± 0.000 154
	$\lambda = 10^{-2}$	1.011 55 ± 0.000 067	0.011 33 ± 0.000 066
	$\lambda = 10^{-3}$	1.011 40 ± 0.000 020	0.011 32 ± 0.000 020
	$\lambda = 10^{-4}$	**1.011 39 ± 0.000 013**	**0.011 33 ± 0.000 013**
	$\lambda = 10^{-5}$	1.011 39 ± 0.000 023	0.011 32 ± 0.000 023
CMC	data not permuted	0.998 28 ± 0.000 081	−0.001 72 ± 0.000 081
	data permuted	1.011 41 ± 0.000 007	0.011 35 ± 0.000 007
NDPE	data not permuted	1.015 56 ± 0.000 077	0.015 18 ± 0.000 076
	data permuted	1.011 55 ± 0.000 077	0.011 23 ± 0.000 077
WRS	data not permuted	0.998 67 ± 0.000 420	−0.001 33 ± 0.000 420
	data permuted	1.011 35 ± 0.000 039	0.011 29 ± 0.000 039

Table 9.4.: True values and estimates of $\pi(\varphi)$, for various test functions φ, for the second log-normal model. For the three embarrassingly parallel approaches (CMC, NDPE, WRS) we present results obtained both without and with first permuting and repartitioning the data into new blocks. For each method the mean estimate ± Monte Carlo standard error (MCSE) is presented, as computed over 25 replicates; the estimator corresponding to the lowest mean squared error is printed in bold.

introduced when using GCMC with large values of λ (as seen here with $\lambda = 10$), for which π_λ is a poor approximation of π.

Also of note are estimates of $\int \log(z)\pi(z)\,\mathrm{d}z$, corresponding to the mean of the Gaussian model of which this a reparametrisation. While global consensus Monte Carlo performs well across a range of λ values, the other methods perform less favourably; consensus Monte Carlo produces an estimate that is incorrect by an order of magnitude. While this could be solved by a simple reparametrisation of the problem in this case, in more general settings no such straightforward solution may exist.

As an additional example, we generated a data set comprising $b = 32$ blocks, each containing 10^4 data. Within the jth block, the data were generated as IID observations of a log-normal random variable with parameters $(\mu_j, 1)$; the parameters μ_j were drawn independently from a normal distribution with mean 0 and variance 10^{-2}. We took $f_j(z) = \mathcal{LN}(\bar{y}_j; \log(z), 10^{-4})$, with each $\bar{y}_j$ being the geometric mean of the observations in the jth block; we used the same prior $\mu(z) = \mathcal{LN}(z; 0, 25)$ as previously. While this represents a misspecified model, it is useful in exemplifying the behaviour of global consensus Monte Carlo in cases where there are differences between the blocks of data.

Table 9.4 shows the estimates of $\int z\pi(z)\,\mathrm{d}z$ and $\int \log(z)\pi(z)\,\mathrm{d}z$, from 25 runs in each algorithmic setting. Global consensus Monte Carlo produces low-bias estimates for a range of λ values. In contrast, the embarrassingly parallel methods result in somewhat larger biases; this is particularly the case for the expected value of the logarithm in the cases of CMC and WRS, which behave similarly on this example. The NDPE method, which is based on kernel density estimation, works reasonably well for this univariate model.

When the data are first randomly permuted and repartitioned into 32 new blocks, the performances of the embarrassingly parallel methods are improved, though we still find that for appropriately-chosen λ, GCMC estimators attain a lower mean squared error. Furthermore, for large distributed data sets permutation of the data in this manner may not be feasible, for example if security restrictions prevent the transfer of data between machines. Note that permuting the data has no effect on the performance of global consensus Monte Carlo for this model, since the Z-chain resulting from a Gibbs sampler depends only on the geometric mean of the entire data set (this may be shown using similar arguments to those in Section 7.4, in which the Z-chain in a Gaussian setting behaves as an AR(1) process).

9.3. Bayesian logistic regression

Binary logistic regression models are commonly used in settings related to marketing. In web design for example, A/B testing may be used to determine which content choices lead to maximised user interaction, such as the user clicking on a product for sale.

We assume that we have a data set of size n formed of responses $\eta_\ell \in \{-1, 1\}$, and vectors $\xi_\ell \in \{0, 1\}^d$ of binary covariates, where $\ell \in \{1, \dots, n\}$. The likelihood contribution of each block of data then takes the form

$$f_j(z) = \prod_{\ell \in B_j} S(\eta_\ell z^\mathsf{T} \xi_\ell)$$

for $z \in \mathbb{R}^d$, where B_j is the set of indices ℓ included in the jth block of data, and $S : \mathbb{R} \longrightarrow [0, 1]$ denotes the logistic function, $S(x) := (1 + \exp(x))^{-1}$.

For the prior μ, we use a product of independent zero-mean Gaussians, with standard deviation 20 for the parameter corresponding to the intercept term, and 5 for all other parameters. For the Markov transition densities in GCMC, we use multivariate spherical Gaussian densities: $K_j^{(\lambda)}(z, x) = \mathcal{N}(x; z, \lambda I)$ for each $j \in \{1, \dots, b\}$.

We investigated several such simulated data sets and the efficacy of various approaches in approximating the true posterior π. To illustrate the bias–variance trade-off described in Section 7.3.1, in the presentation of these results we focus on the estimation of the posterior first moment $\pi(\mathrm{Id})$. While our global consensus approach was consistently successful in forming estimators with low mean squared error in each component, in low-dimensional settings the application of consensus Monte Carlo often resulted in marginal improve-

ments. However, in many higher-dimensional settings, the estimators resulting from CMC and other embarrassingly parallel approaches exhibited relatively large biases.

We present here an example in which the d predictors correspond to p binary input variables, their pairwise products, and an intercept term, so that $d = 1 + p + \binom{p}{2}$. In settings where the interaction effects corresponding to these pairwise products are of interest, the dimensionality d of the space can be very large compared to p.

We used a simulated data set with $p = 20$ input variables, resulting in a parameter space of dimension $d = 211$. The data comprise $n = 80\,000$ observations, split into $b = 8$ equally-sized blocks. Each observation of the 20 binary variables was generated from a Bernoulli distribution with parameter 0.1, and for each vector of covariates, the response was generated from the correct model, for a fixed underlying parameter vector z^*.

9.3.1. Metropolis-within-Gibbs

We applied GCMC for values of λ between 10^{-2} and 1. We used a Metropolis-within-Gibbs formulation of Algorithm 7.1, sampling directly from the Gaussian conditional distribution of Z given $X_{1:b}$. To sample approximately from the conditional distributions of each X_j given Z we used Markov kernels $P_{j,z}^{(\lambda)}$ comprising $k = 20$ iterations of a random walk Metropolis kernel.

As mentioned in Section 7.2.1, in settings of high communication latency our approach allows a greater proportion of wall-clock time to be spent on likelihood contributions, compared to an MCMC chain directly targeting the full posterior π. To compare across settings, we therefore consider an abstracted distributed setting of the form described in Section 7.2.1, here assuming that the latency is 10 times the time taken to compute each partial likelihood f_j. To use the notation of Section 7.2.1, we assume that $C = 10\ell$.

We also compare with the same embarrassingly parallel approaches as in Section 9.2 (CMC, NDPE, WRS), which are comparatively unaffected by communication latency. For these methods, we again used random walk Metropolis to draw samples from each subposterior distribution. To ease computation, we thinned these chains before applying the combination step, taking every kth value; in practice, the estimators obtained using these thinned chains behaved very similarly to those obtained using all subposterior samples.

To provide a 'ground truth' against which to compare the results we ran a random walk Metropolis chain of length 500 000 targeting π. For all our random walk Metropolis samplers we used Gaussian proposal kernels. To determine the covariance matrices of these, we formed a Laplace approximation of the target density following the approach of Chopin and Ridgway (2017), scaling the resulting covariance matrix optimally according to results of Roberts and Rosenthal (2001).

For each algorithmic setting, we ran the corresponding sampler 25 times. To compare the resulting estimators of the posterior mean we computed the mean squared error of each of the d components of the posterior mean, summing these to obtain a 'mean sum of

Algorithm		Mean sum of squared errors
GCMC	$\lambda = 10^{0}$	0.1835
	$\lambda = 10^{-0.5}$	0.1379
	$\lambda = 10^{-1}$	0.0770
	$\lambda = 10^{-1.5}$	**0.0478**
	$\lambda = 10^{-2}$	0.0662
CMC		0.3710
NDPE		0.8476
WRS		0.6402
Direct MCMC		0.0884

Table 9.5.: Mean sum of squared errors over all d components of estimates of the posterior mean for the logistic regression model, formed using various algorithmic approaches as described in the main text, during an approximate wall-clock time equal to 200 000 times that required to compute a single partial likelihood f_j. All values computed over 25 replicates, with the lowest value printed in bold.

squared errors'.

Table 9.5 compares the values obtained by each algorithm after an approximate wall-clock time equal to 200 000 times the time taken to compute a single partial likelihood f_j. Accounting for latency in the abstracted distributed setting described above, the GCMC approach is able to generate 5000 approximate posterior samples during this time, spending 50% of time on likelihood computations. In contrast, a direct MCMC approach generates 9523 samples, but would only spend 4.8% of the time on likelihood computations, with the remainder lost due to latency.

The result is that the estimators generated by GCMC for appropriately-chosen λ exhibit lower mean sums of squared errors: we conduct many more accept/reject steps in each round of inter-node communication than if we were to target π directly, and so it becomes possible to achieve faster mixing of the Z-chain (and a better estimator) compared to such a direct approach. This may be seen when comparing the effective sample size (ESS) of each chain, where we estimate this via the 'batch means' approach of Vats et al. (2019): we find that the average ESS of the direct MCMC chains is only 1111, while depending on the choice of λ, the shorter GCMC chains have average ESS values between 1327 and 4577.

Despite being unaffected by latency and therefore allowing many more samples to be drawn, the embarrassingly parallel approaches (CMC, NDPE, WRS) perform poorly compared to GCMC. This is particularly true of the nonparametric density product estimation (NDPE) method of Neiswanger et al. (2014): while asymptotically exact even in non-Gaussian settings, the resulting estimator is based on kernel density estimators and is not effective in this high-dimensional setting.

Figure 9.8 shows the mean sums of squared errors as a function of the approximate wall-

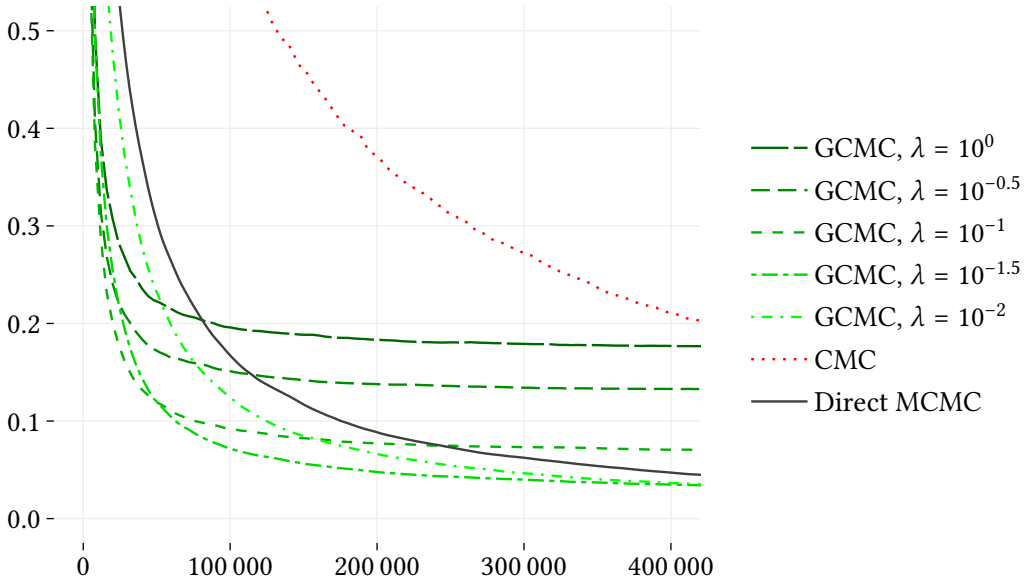


Figure 9.8.: Mean sum of squared errors over all d components of estimates of the posterior mean for the logistic regression model, formed using various algorithmic approaches as described in the main text. Values plotted against the approximate wall-clock time, relative to the time taken to compute a single partial likelihood term. All values computed over 25 replicates.

clock time (for simplicity we include only the best-performing of the three embarrassingly parallel methods, omitting the results for NDPE and WRS). We see that for large enough λ, the GCMC estimators $\pi_\lambda^N(\mathrm{Id})$ exhibit rather lower values than the corresponding CMC and 'direct' MCMC estimators. As the number of samples used grows, the squared bias of these estimators begins to dominate, and so smaller λ values result in lower mean squared errors. As λ becomes smaller the autocorrelation of the resulting Z-chain increases; indeed we found that for λ too small, the GCMC estimator $\pi_\lambda^N(\mathrm{Id})$ will always have a greater mean squared error than the 'direct' MCMC estimator, no matter how much time is used. Of course, since an MCMC estimator formed by directly targeting π is consistent in N, given sufficient time such an estimator will always outperform estimators formed using GCMC, which are biased for any λ. However, in many practical big data settings it may be infeasible to draw large numbers of samples using the available time budget.

Rather than summing the squared numerical errors across all d components, one might consider the error in each component of the posterior mean, relative to its true marginal standard deviation. Given the sparsity of the data in this example this may be more meaningful, since the marginal posterior variances of the parameters corresponding to interaction terms are rather larger than those of the other parameters. Figure 9.9 shows, as a function of the number of the approximate wall-clock time and for each algorithmic setting, the mean absolute value of this 'standardised' error, averaged across all d com-

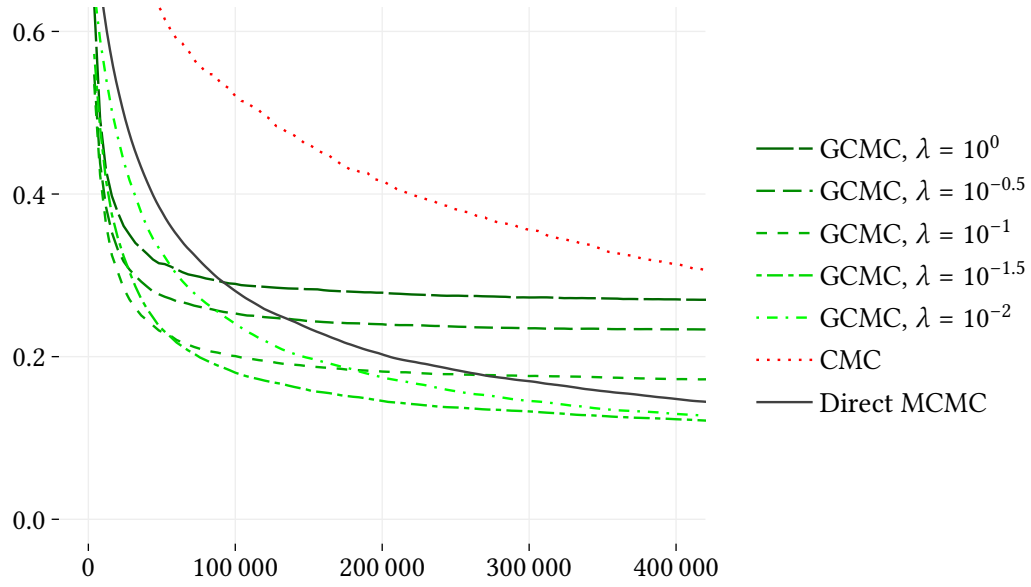


Figure 9.9.: Absolute value of the error in each component of the posterior mean, divided by the corresponding true standard deviation, averaged over all d components, for the logistic regression model and for various algorithmic approaches as described in the main text. Values plotted against the approximate wall-clock time, relative to the time taken to compute a single partial likelihood term. All values computed over 25 replicates.

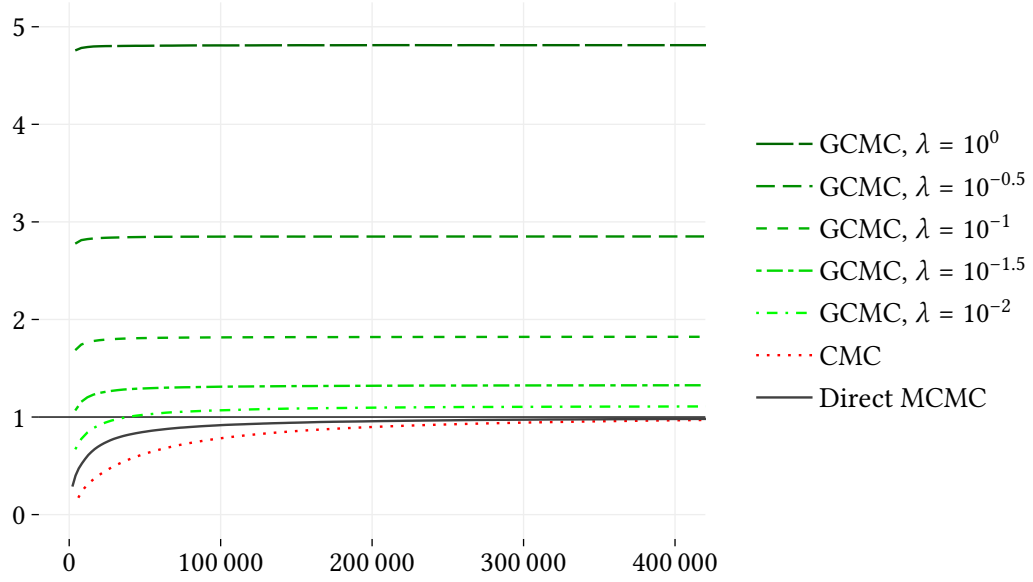


Figure 9.10.: Ratio between the estimated marginal posterior standard deviation and corresponding true standard deviation, averaged over all d component, for the logistic regression model and for various algorithmic approaches as described in the main text. Values plotted against the approximate wall-clock time, relative to the time taken to compute a single partial likelihood term. All values computed over 25 replicates.

ponents; that is to say, the mean absolute value of the componentwise error, relative to the corresponding marginal standard deviation. We see essentially the same pattern of behaviour: for large enough λ the GCMC estimators $\pi_\lambda^N(\mathrm{Id})$ outperform those formed using the other approaches, over this range of wall-clock times.

Recall that the approximate posterior π_λ used in GCMC may be viewed as a smoothed form of the true posterior π, as seen in (7.3). The result is that π_λ typically has a greater variance than π. Figure 9.10 shows, again as a function of the approximate wall-clock time and for each algorithmic setting, the ratio between the estimated marginal standard deviation of each component of the posterior distribution and the corresponding true standard deviation of that component, averaged over all d components. We see that for λ sufficiently large, the approximate posteriors π_λ^N are rather more diffuse than the true posterior, and the CMC approximation. Consequently, a smaller value of λ may be preferable when using GCMC to estimate higher moments of π.

9.3.2. SMC sampler

We also applied the SMC procedure of Chapter 8 to this logistic regression model. While we found that the SMC approach was most effective in lower-dimensional settings in which it is less computationally expensive, the SMC procedure can be more widely use-

Estimate	Mean sum of squared errors
Initial SMC estimate	0.0692
SMC estimate after 100 iterations	0.0418
Bias-corrected estimate after 100 iterations	0.0682
SMC estimate of lowest estimated MSE	0.0367

Table 9.6.: For the logistic regression example, the mean sums of squared errors of estimators formed using the SMC procedure. These are the SMC estimate $\pi^N_{\lambda_0}(\mathrm{Id})$ corresponding to the initial (largest) λ value; the SMC estimate $\pi^N_{\lambda_{100}}(\mathrm{Id})$ obtained after 100 iterations; the bias-corrected estimate (8.3) at this point; and the estimator chosen by the stopping rule proposed in Section 8.3 (with parameter $\kappa = 15$). All values are computed over 25 replicates of an SMC sampler.

ful as a means of 'refining' the estimator formed using a single λ value, as discussed in Section 8.1.

We used $N = 1250$ particles, initialising the particle set by thinning the chain generated by the Metropolis-within-Gibbs procedure with $\lambda = 10^{-1}$. To generate a sequence of subsequent λ values we used the adaptive procedure of Zhou et al. (2016), using tuning parameter $\mathrm{CESS}^\star = 0.98N$. For the Markov kernels M_p we used Metropolis-within-Gibbs kernels constructed according to Algorithm 7.1 as previously, with each update of X_j given Z comprising $k = 50$ iterations of a random walk Metropolis kernel.

The mean sums of squared errors of various estimators associated with this approach are presented in Table 9.6. The estimator $\pi^N_{\lambda_0}(\mathrm{Id})$ formed using the initial particle set was found to have a mean sum of squared errors of 0.0692. After a fixed number of iterations ($n = 100$) the resulting SMC estimate exhibited a mean sum of squared errors of 0.0418; this represents a decrease of 40%, and has the benefit of avoiding the need to carefully specify a single value for λ.

Used alone, the bias correction procedure of Section 8.2 was found to perform best in lower-dimensional settings (as in Section 9.1.2; here, it resulted in a mean sum of squared errors of 0.0682 after 100 iterations. However, improved results were obtained using the stopping rule we propose in Section 8.3 (with stopping parameter $\kappa = 15$), which is based on our proposed bias correction procedure. The estimator selected by this stopping rule, which automatically determines when to terminate the algorithm, obtained a mean sum of squared errors of 0.0367, a decrease of 47% from the estimator generated using the initial particle set.

9.4. Stochastic volatility model

Finally, we provide an example demonstrating the framework for random effects models described in Section 7.5, and the use of pseudo-marginal MCMC kernels as discussed in Section 7.2.2. The example we consider is based on stochastic volatility models, widely

used in mathematical finance in the context of asset pricing.

Specifically, we consider a model based on the 'basic multivariate stochastic volatility model' of Chib et al. (2009, Section 2). We assume that the data are observations of a sequence $Y_{0:T}$ of random variables, distributed according to the following hidden Markov model:

$$\begin{aligned} X_0 &\sim \mathcal{N}(m, U^*); \\ X_t \,|\, X_{t-1} &\sim \mathcal{N}(m + \operatorname{diag}(\phi)(X_{t-1} - m), U), \quad && t \in \{1, \dots, T\}; \\ Y_t \,|\, X_t &\sim \mathcal{N}(0, \exp(\operatorname{diag}(X_t))), && t \in \{0, \dots, T\}. \end{aligned}$$

Here $m \in \mathbb{R}^d$ is the mean of the latent process, $\phi \in [0,1]^d$ is the mean reversion parameter, and $U \in \mathbb{R}^{d \times d}$ is a positive definite covariance matrix. The (i,j)th element of the matrix U^* is equal to the (i,j)th element of U divided by $1 - \phi_i \phi_j$.

For our illustrative example we consider b blocks of data, with the jth block comprising n_j time series of length T_j. We model each block of data as having its own mean parameter m_j and covariance matrix U_j, with the mean reversion parameter ϕ being common to all the time series.

Using the notation of Section 7.5, we have $Z = \phi$, and $W_j = (m_j, U_j)$ for $j \in \{1, \dots, b\}$. To construct a posterior distribution of the form (7.24), we choose the prior μ over $Z = \phi$ to be uniform on $[0,1]^d$. The prior distributions ν_j of the local variables are assigned as follows. Each m_j is given an improper uniform prior; the diagonal elements of each U_j are given independent inverse gamma priors (mean 0.2, variance 1); and the off-diagonal elements of the corresponding correlation matrices are given independent triangular priors, i.e. each element has a prior density at $x \in [-1,1]$ given by $1 - |x|$.

The partial likelihood terms in (7.24) may be written as

$$f_j(z, w_j) = \prod_{i=1}^{n_j} f_{j,i}(z, w_j), \tag{9.1}$$

where $f_{j,i}$ is the likelihood contribution of the ith times series in the jth block; denote this time series by $y_{1:T_j}^{(j,i)}$. For given values of $z = \phi$ and $w_j = (m_j, U_j)$, the likelihood contribution $f_{j,i}(z, w_j)$ may be estimated using a sequential Monte Carlo algorithm. Specifically one may use any of the algorithms detailed in Chapter 1, taking

$$\begin{aligned} M_0(\cdot) &= \mathcal{N}(m_j, U_j^*); \\ M_j(x, \cdot) &= \mathcal{N}(m_j + \operatorname{diag}(\phi)(x - m_j), U_j), \quad && x \in \mathbb{R}^d, \ t \in \{1, \dots, T\}; \\ G_j(\cdot) &= \mathcal{N}(y_t^{(j,i)}; 0, \exp(\operatorname{diag}(\cdot))), && x \in \mathbb{R}^d, \ t \in \{0, \dots, T\}. \end{aligned}$$

An estimator of the marginal likelihood $f_{j,i}(z, w_j)$ is then obtained as the normalising constant estimator $\gamma_n^N(\mathbb{1})$; by Proposition 1.2, this is unbiased. By estimating each such likeli-

hood contribution independently and taking the appropriate product as in (9.1), we obtain an unbiased estimator of each partial likelihood $f_j(z, w_j)$, and therefore unbiased estimators of the posterior density (7.24).

It follows that we may draw samples approximately distributed according to the full posterior using a pseudo-marginal MCMC algorithm. This essentially takes the form of a particle marginal MCMC algorithm as proposed by Andrieu et al. (2010) except that multiple independent SMC algorithms are used, with each likelihood contribution estimated independently. Similarly any density dependent on only one of the partial likelihood terms (9.1), such as the subposterior distributions used in consensus Monte Carlo (CMC), may be approximated using such an algorithm.

In the examples that follow, we compare our global consensus Monte Carlo (GCMC) approach with the CMC approach of Scott et al. (2016), and an approach in which one directly targets the full posterior density. For GCMC, which in this case takes the form of Algorithm 7.2, we require a choice of transition kernels $K_j^{(\lambda)}$ on $\mathsf{E} = [0,1]^d$. We choose each kernel to correspond to a product of independent zero-mean Gaussian kernels, each with variance λ, on a probit scale. Denoting by $\boldsymbol{\Phi}^{-1} : [0,1]^d \to \mathbb{R}^d$ the function that applies the quantile function of the standard normal distribution to each component of its argument, we may write

$$K_j^{(\lambda)}(z, x) = \frac{\mathcal{N}(\boldsymbol{\Phi}^{-1}(x); \boldsymbol{\Phi}^{-1}(z), \lambda I)}{\mathcal{N}(\boldsymbol{\Phi}^{-1}(x); 0, I)},$$

where the term in the denominator corresponds to the Jacobian determinant associated with the probit transformation. It is readily found that this choice is sufficient for Assumption 7.1 to hold.

For this example in which the prior distribution on Z is uniform on $[0,1]^d$, a benefit of this choice is that it allows direct sampling from the full conditional distribution of Z, which is of the form (7.28). Specifically, we find that

$$\tilde{\pi}_\lambda(z \,|\, x_{1:b}, w_{1:b}) = \frac{\mathcal{N}\left(\boldsymbol{\Phi}^{-1}(z); \frac{\sum_{j=1}^b \boldsymbol{\Phi}^{-1}(x_j)}{\lambda + b}, \frac{\lambda I}{\lambda + b}\right)}{\mathcal{N}(\boldsymbol{\Phi}^{-1}(z); 0, I)}.$$

One can sample according to this density by first drawing an auxiliary variable

$$\tilde{Z} \,|\, X_{1:b}, W_{1:b}, \lambda \sim \mathcal{N}\left(\frac{\sum_{j=1}^b \boldsymbol{\Phi}^{-1}(X_j)}{\lambda + b}, \frac{\lambda I}{\lambda + b}\right)$$

and then taking $Z = \boldsymbol{\Phi}(\tilde{Z})$, where $\boldsymbol{\Phi} : \mathbb{R}^d \to [0,1]^d$ applies the cumulative distribution function of the standard normal distribution to each component of its argument.

We used pseudo-marginal Metropolis–Hastings kernels of the form earlier described to sample from distributions involving the likelihood contributions (9.1). That is, we used

these kernels to sample from:

- the full conditional distributions of each (X_j, W_j) in GCMC;
- each subposterior distribution in CMC;
- and the full posterior in the 'direct' MCMC approach.

For proposal kernels we used random walk kernels, proposing new values for all components simultaneously using additive normal innovations (on a probit scale for Z or X_j, and on an untransformed scale each W_j). We determined the scale simply by approximating the scale of each likelihood contribution using a short MCMC chain, using these and the corresponding (pseudo-)prior to obtain a rough approximation of the target covariance matrix, which we scaled according to the optimal scaling result of Sherlock et al. (2015, Corollary 1 and following remarks).

As discussed in Section 7.2.2, in order to achieve good mixing in the 'direct' MCMC approach one generally requires each likelihood contribution to be estimated with a lower variance than would be necessary in the other two settings. This is because the target density in that case depends on a larger number of likelihood contributions (i.e. those from all b blocks of data, rather than only one).

We found that in order for all three algorithmic approaches to possess comparable mixing properties, it was necessary for the SMC algorithms used within the pseudo-marginal kernels to use b times more particles when using the 'direct' MCMC approach than when using the GCMC and CMC approaches. We assessed this in practice by looking at the variance of estimates of the target log-density, and comparing with the optimal values proposed by the tuning methods described in Section 7.2.2. In order for the three algorithmic approaches to be comparable (in the sense of having comparable computational cost), the results we shall present for the 'direct' MCMC approach use b times fewer MCMC samples than the other two approaches.

We first considered a simple model in which the time series are bivariate ($d = 2$). We have observations of 8 time series each of length $T = 50$, divided into $b = 2$ blocks of equal size. For the SMC samplers used with in the pseudo-marginal kernels, we chose the number of particles in order that the variance of the resulting estimates of the log-density was around 3.2, following Sherlock et al. (2015). Within the GCMC and CMC we required 250 particles for each such SMC run, while the direct MCMC approach required 500. To account for this additional computational cost we ran chains of length 10 000 for the GCMC and CMC settings, comparing with chains of length 5000 for the direct approach. Within GCMC we considered a range of λ values between 10^{-4} and 1.

Table 9.7 shows the mean sum of squared errors of the posterior mean, as computed over 10 replicates for each algorithmic setting. That is, in each case we computed the squared error in the posterior mean of each parameter, summed this over all parameters,

Algorithm		Mean sum of squared errors	
		All parameters	$m_{1:b}$ only
GCMC	$\lambda = 10^{0}$	0.1988	0.1230
	$\lambda = 10^{-0.5}$	0.1258	0.0821
	$\lambda = 10^{-1}$	0.0562	0.0225
	$\lambda = 10^{-1.5}$	0.0288	0.0100
	$\lambda = 10^{-2}$	0.0377	0.0174
	$\lambda = 10^{-2.5}$	0.0233	0.0089
	$\lambda = 10^{-3}$	0.0368	0.0194
	$\lambda = 10^{-3.5}$	0.0704	0.0372
	$\lambda = 10^{-4}$	0.0761	0.0280
CMC		0.0597	**0.0088**
Direct MCMC		**0.0229**	0.0148

Table 9.7.: For the first stochastic volatility model, the mean sum of squared errors of the posterior mean, where the sum is taken over all parameters; and where the sum is taken only over the parameters $m_{1:b}$ describing the mean of the latent process. All values computed over 10 replicates of each algorithmic approach, as described in the main text; the lowest value in each column is printed in bold.

and then took the mean of this value over all 10 replicates. We see that for comparable computational cost the direct MCMC approach performs best here, although the GCMC approach performs comparably for appropriately-chosen λ.

In many settings the parameters $m_{1:b}$ are of particular interest; they represent the mean of the latent process, and may therefore be viewed as quantifying the volatility in the observations. We therefore also present in Table 9.7 the mean sum of squared errors for this subset of parameters only. While the CMC approach performed best in this regard, again we see that GCMC performs well for a range of λ values. Indeed, the GCMC results have the beneficial property of providing low-error estimates of the parameters of direct interest, while still obtaining reasonable estimates of the other model parameters (which may still be of secondary interest).

As a second example, we considered a similar model in which 12 bivariate time series of length $T = 50$ were split into $b = 3$ equal blocks. The GCMC and CMC approaches used SMC samplers with 200 particles, with chains of length 12 000 generated; the direct MCMC approach used 600 particles and chains of length 4000. The results in this case, presented in Table 9.8, show that the GCMC approach generally performed poorly in estimating the posterior mean, with the resulting mean sums of squared errors being rather larger than those obtained via the other two approaches. However when considering only the 'mean volatility' parameters $m_{1:b}$, for an appropriate choice of λ GCMC was able to attain a rather *lower* mean sum of squared errors than either of the other two approaches (most of the remaining error resulted from poor estimation of the noise covariance matrices U_j).

Algorithm		Mean sum of squared errors: All parameters	Mean sum of squared errors: $m_{1:b}$ only
GCMC	$\lambda = 10^{0}$	4.8583	4.6994
	$\lambda = 10^{-0.5}$	0.7659	0.5526
	$\lambda = 10^{-1}$	0.4251	**0.0107**
	$\lambda = 10^{-1.5}$	1.4520	0.0243
	$\lambda = 10^{-2}$	2.8413	0.0347
	$\lambda = 10^{-2.5}$	3.9541	0.0404
	$\lambda = 10^{-3}$	4.3179	0.0423
	$\lambda = 10^{-3.5}$	4.6233	0.0449
	$\lambda = 10^{-4}$	4.6803	0.0533
CMC		**0.0783**	0.0295
Direct MCMC		0.1044	0.0729

Table 9.8.: For the second stochastic volatility model, the mean sum of squared errors of the posterior mean, where the sum is taken over all parameters; and where the sum is taken only over the parameters $m_{1:b}$ describing the mean of the latent process. All values computed over 10 replicates of each algorithmic approach, as described in the main text; the lowest value in each column is printed in bold.

The reasons for this behaviour are not clear. This may be purely due to the specifics of this example (e.g. a property of the proposal kernel), or some inherent advantage of the GCMC approach in the estimation of these parameters, perhaps since of all the parameters in the model only $m_{1:b}$ are defined on the whole real line. Nonetheless the surprising patterns in these results could provide useful direction for further investigation, as we shall soon discuss in the concluding remarks of this thesis.

9.5. Summary

We have here presented some illustrative examples of our proposed simulation framework, investigating the role of various tuning parameters and providing a comparison to some simple embarrassingly parallel approaches, as well as a more straightforward direct approach. These examples demonstrate the key settings in which we might expect our framework to outperform other algorithms with similar aims: settings in which the likelihood contributions f_j may not be approximately Gaussian, for example in models using high-dimensional 'wide data'; and settings using pseudo-marginal kernels, as discussed in Section 7.2.2.

It is hoped that these results might provide motivation for other applications of the framework models involving high-dimensional distributed data, or for the development of other techniques and methods within the global consensus framework. We detail some such ideas in the conclusion that follows.

Conclusion

Summary

This thesis has considered several problems in the tuning of sequential Monte Carlo methods, the role of variance estimation techniques in such problems, and issues in a number of connected areas including distributed simulation procedures. Following a review of SMC methodology and applications, Part II of this thesis has explored the schedule selection problem for SMC samplers, making a number of key contributions. In particular, we have proposed in Section 3.3 a formulation of this as a minimisation problem, using an objective function that is dependent on the relative asymptotic variance of the normalising constant estimator $\gamma_n^N(\mathbb{1})$. By consideration of the decomposition of this asymptotic variance, we have investigated a number of properties of our proposed objective function $n\sigma_{\mathbb{1}}^2$.

In Chapter 4 we have studied this optimisation problem analytically for settings in which one can construct perfectly-mixing Markov kernels. Our theoretical results, and the heuristics we propose based on these, may be of practical relevance in many settings. For example, our results for restrictions on nested sets find direct application in the rare events estimation procedure of Cérou et al. (2012), and our findings for normal distributions will be useful in many settings involving large data sets, due to the Bernstein–von Mises theorem and other CLT results. The refinement and generalisation of these may provide a useful starting point for future developments, for which we shall later propose possible directions.

The work of Chapter 5 has considered the problem of schedule selection in settings where more general Markov kernels are used. The numerical optimisations of Section 5.1 provide new insights into the behaviour of the $v_{p,n}(\mathbb{1})$ terms in realistic settings. In light of this, we have investigated the use of variance estimation techniques within schedule selection procedures. While these investigations did not lead to the development of a robust schedule selection algorithm, it is clear that the principle of using such variance estimators to tune SMC samplers holds promise, and it is hoped that these discussions may inform future research.

Part III has focused on the problem of inference in settings involving large distributed data sets, and the issues in constructing efficient simulation algorithms for approximating Bayesian posterior distributions. Within Chapter 7 we have proposed a novel framework that may be applied in such settings, describing a distributed Metropolis-within-Gibbs

algorithm that may offer benefits over embarrassingly parallel algorithms, and over an MCMC approach that directly targets the full posterior. We have considered a number of issues in its practical implementation and have analysed the theoretical properties of our proposed algorithm in a Gaussian setting, providing insight into its asymptotic behaviour.

Given the role of the tuning parameter λ in achieving a bias–variance trade-off, we have proposed in Chapter 8 an SMC sampler formulation of the framework. Within this context we have described a simple approach to utilising many of the estimators formed during the execution of the algorithm, regressing on λ in order to obtain bias-corrected estimators of integrals. Our proposed use of SMC variance estimators within weighted least squares represents a novel application of these; empirically, we have found that the resulting linear regression procedure achieves improvements over an unweighted approach. We will consider various possible improvements to these ideas, and to our proposed stopping rule, in the discussion that follows these concluding remarks.

The simulation results of Chapter 9 demonstrate a number of settings in which our proposed approach may offer benefits over a straightforward MCMC approach, and over some simple embarrassingly parallel methods. Given the ever-increasing sizes of data sets in modern statistical settings, we believe that this work forms a valuable contribution to the literature on this highly topical problem.

Contributions

We provide here a detailed list of the novel contributions of this thesis.

Part I comprised a review of the literature and methodology of sequential Monte Carlo methods, and of the applications of sequential Monte Carlo samplers. Rather than presenting new results, the primary role of these first two chapters was to introduce the ideas and notation that form the basis of later work. We note however that many of the expressions in Sections 1.4.1.1 and 1.4.1.2, explicitly describing asymptotic variance decompositions associated with updated and excursion Feynman–Kac models, have not previously been published in these forms.

Part II introduced the open problem of schedule selection for SMC samplers, making a number of key contributions.

- We have proposed in Section 3.3 a formalisation of this problem as a transdimensional optimisation problem, based on the relative asymptotic variance of the normalising constant estimator.

- We have in Section 3.4 derived expressions for the asymptotic variance decomposition terms $v_{p,n}(\varphi)$ in SMC sampler settings. To our knowledge these results have not previously been presented in these forms; this includes the results of Propositions 3.10 and 3.13, in which the terms $v_{p,n}(\mathbb{1})$ are expressed as chi-squared distances.

- For the SMC approach to rare event estimation of Cérou et al. (2012), we have in Section 4.2 extended the authors' optimality result by determining the optimal sequence length n (Proposition 4.8). For the authors' proposed algorithm we have also derived the optimal tuning parameter value in a perfectly-mixing setting (Remark 4.9).

- For sequences of normal distributions, we have in Section 4.3 derived several results describing the optimal such sequence in a specialised setting (e.g. Proposition 4.12), proposing several practical heuristics based on these (e.g. Remark 4.14).

- We have demonstrated in Section 4.4 some properties of the problem of minimising a sum of chi-squared distances, including the proof of a 'reversed triangle inequality' (Proposition 4.17).

- In Section 5.1 we have provided numerical optimisation results for the optimal distribution schedules in a simple imperfectly-mixing setting, providing new insight into the optimal behaviour of the asymptotic variance decomposition terms in various such settings.

- We have proposed the novel application of SMC variance estimators to the problem of schedule selection. In Section 5.2 we have described various possible approaches to this, empirically analysing their behaviour and documenting their limitations.

Part III devised a novel framework for Bayesian inference on large data sets, proposing and investigating an MCMC algorithm and SMC sampler for use in distributed settings.

- We have introduced this framework in Section 7.1 by defining an instrumental hierarchical model, proposing in Section 7.2 the construction of a distributed Metropolis-within-Gibbs algorithm (Algorithm 7.1).

- We have described in Section 7.3 various considerations in the implementation of our algorithm, and have analysed in Section 7.4 the behaviour of the algorithm in a simple tractable setting, providing results describing its asymptotic properties.

- We have proposed in Section 7.5 an extension of our framework to random effects models, and a form of our distributed Metropolis-within-Gibbs algorithm for use in such settings (Algorithm 7.2).

- We have constructed an SMC formulation of this framework (Algorithm 8.1). Within this context we have proposed in Section 8.2 a bias correction technique for use in SMC contexts, based on local linear regression. This includes a novel application of SMC variance estimators proposed in Section 8.2.1, for inverse-variance weighting within weighted least squares,.

- We have proposed heuristic procedures to be used in combination with this bias correction technique: a procedure for determining which SMC estimators to include in the local linear regression (Algorithm 8.2) and a stopping rule for the SMC sampler (Algorithm 8.3).

- We have provided an empirical study of our proposed algorithms and procedures in Chapter 9, including comparisons with a straightforward MCMC approach and with some embarrassingly parallel approaches.

Directions for further research

Procedures for schedule selection

As suggested previously, our results for optimal distribution schedules in perfectly-mixing settings may find practical application more generally, particularly when well-mixing kernels are used. There is however scope to refine or extend these results, which may allow the derivation of additional heuristic procedures for schedule selection. Considering the results for normal distributions in Section 4.3 it would be particularly useful to develop the results of Proposition 4.16 further, deriving a clearer heuristic for choosing the length n of the distribution schedule.

Another direction would be to investigate further the properties of the chi-squared distance. In Proposition 4.17 we have shown that a sum of chi-squared distances can always be reduced by inserting an intermediate mixture distribution between two existing distributions. Useful extensions might include determining the *optimal* mixture distribution in this case; obtaining a similar result when the intermediate distribution is formed by tempering; and deriving tighter bounds on the resulting reduction in the sum of chi-squared distances. Such results might in turn lead to practical heuristic procedures for schedule selection. Initial investigations into these topics have led to interesting findings, though to avoid digression from the main aims of the thesis these have not been included here.

The results of Chapter 5 demonstrated that while using variance estimation techniques for schedule selection is a powerful idea in principle, the construction of a robust procedure is difficult in practice. Given our documented issues with the variance of these estimators, a useful avenue of investigation may be the development of variance reduction techniques for these estimators, perhaps motivated by similar ideas used in MCMC (see Glasserman, 2004, Chapter 4 for a review). This may facilitate the development of procedures for schedule selection, and for other tuning issues such as those described in Section 5.3.

Extensions of the schedule selection problem

As discussed in the context of tempering and path sampling in Section 3.2.3, a distribution schedule $(\pi_p)_{p=0}^n$ may be viewed as the discretisation of a continuum of probability measures that smoothly interpolate between π_0 and $\pi_\star$. Indeed while SMC samplers run in discrete time, many of the associated concepts admit natural extensions to continuous time. In particular, the discrete time Feynman–Kac models introduced in Section 1.2.1 have well-studied continuous analogues (see e.g. Del Moral, 2004, Section 1.3.1). The investigation of such constructions could therefore provide insight into the behaviour of distribution schedules in discrete settings. For example, such ideas are used by Beskos et al. (2014) in an investigation of the asymptotic behaviour of the ESS in SMC samplers, as the dimension of the space X tends to infinity. A similar approach could be used to investigate the asymptotic properties of optimal schedules.

Within our discussion of the schedule selection problem we have focused on the construction of a distribution schedule that begins with the initial distribution π_0 and ends with the distribution of interest $\pi_\star$. In other settings however we require a sequence that approaches but does not attain $\pi_\star$, for example because this distribution corresponds to a point mass and therefore does not admit a density, preventing the evaluation of the SMC incremental weights (2.2). A common feature of such settings is the need to balance the fidelity of these distributions to $\pi_\star$ with the difficulty of constructing well-mixing Markov kernels leaving these distributions invariant. Examples include SMC samplers for approximate Bayesian computation (e.g. Sisson et al., 2007; Del Moral et al., 2012a) in which the tolerance parameter approaches but does not reach zero, and our proposed SMC sampler for global consensus Monte Carlo in Chapter 8.

In such cases the problem of selecting a distribution schedule is essentially extended, since one must also choose the final distribution in the sequence. The stopping rule that we propose for our global consensus algorithm in Section 8.3 provides a heuristic approach for a specific setting, but is not fully general (a point that we shall return to shortly). Instead, one might consider how to extend our formulation of the schedule selection problem to account for this additional dimension. This provides ample opportunity for future research, with a number of the ideas and results from this thesis likely to prove useful in this setting also (for example, our expressions for the asymptotic variance decomposition terms). Variance estimators could also play a useful role in the development of a stopping rule for practical application in general settings.

The proposed global consensus algorithms

Regarding our proposed global consensus framework in Part III, a promising idea for future work is the further investigation of the properties of our proposed algorithms. Our theoretical investigations in Section 7.4 focus on a Gaussian setting, since this is partic-

ularly amenable to analysis; we found that theoretical analysis of our algorithms when applied to other models is generally more difficult. Although many of the properties of this simple setting would be expected to apply more generally (as we have demonstrated in our empirical studies), further investigations may be insightful, and may lead to new heuristics for the choice of the tuning parameter λ. As earlier mentioned, essentially the same framework to that presented here (and in Rendell et al., 2018) was independently and contemporaneously proposed by Vono et al. (2019a) for use in a serial context. Subsequent to much of the work in this thesis, various non-asymptotic and convergence results have been published in Vono et al. (2019b), several of which may also find application in our distributed setting.

Our proposed bias correction technique for the global consensus SMC algorithm, and the use of SMC variance estimators within weighted linear regression for this purpose, is a simple idea that may also be useful in other contexts. However, theoretical analysis of this scheme is complex; further investigations into its properties may lead to useful refinements of our proposed approach, particularly in the high-dimensional settings in which it was less successful. Additionally, the use of non-linear procedures (such as that proposed in an ABC context by Blum and François, 2010) may provide a more robust alternative with more theoretical guarantees. It may therefore be insightful to investigate some such alternative approaches and the role that SMC variance estimators might play within these.

The stopping rule that we have proposed for the SMC sampler in Section 8.3 is intended to assist the practitioner in the choice of the tuning parameter λ, and is designed around the bias–variance trade-off that motivates the framework as a whole. While our approach is seen empirically to have useful properties, there is room for further work and improvement. For example, our proposed stopping rule assumes that for the true posterior distribution π, there is one integral $\pi(\varphi)$ that we wish to estimate with minimal mean squared error. In practice however there may be several such integrals of interest, and so an improved approach would be independent of the choice of test function φ, instead being suitable for the estimation of integrals of many such functions. This idea may be compared with our use of $n\sigma_{\mathbb{1}}^2$ in schedule selection, since a low-variance estimator of the normalising constant may be indicative of low variances of other estimators. Although we investigated various stopping rules utilising the normalising constant estimator (or estimated variance thereof), none performed especially well in obtaining low-MSE estimators of integrals with respect to the true posterior distribution. As previously mentioned however, there is scope to develop general stopping rules for this and similar SMC algorithms within the broader context of schedule selection.

Other global consensus applications

Finally, we stress that the MCMC and SMC algorithms that we have presented constitute only two possible approaches to inference using the instrumental hierarchical model that

we propose. Considering the decreasing sequence of λ values in our SMC algorithm, over which fidelity to the original model increases while mixing quality decreases, a natural idea is to investigate other algorithmic structures utilising such sequences of distributions. In particular several of the tempering-based algorithms reviewed in Section 2.2.2 could be applied within our framework, in which the role of λ is comparable to that of an inverse temperature. For example, by assigning a prior distribution to λ in the instrumental model we introduce in Section 7.1, one could construct a joint distribution over all parameters (including λ), allowing the application of a form of simulated tempering. Another idea would be an implementation of parallel tempering, though this would require a careful construction that minimises inter-node communication.

Multilevel Monte Carlo (see Giles, 2015, for a review) might also provide a useful direction for investigation. Under appropriate convergence conditions (i.e. using an appropriate reformulation of Assumption 7.1) such an approach might allow the construction of estimators with much lower MSE, compared to that achieved using our proposed MCMC algorithm. In summary, our global consensus framework offers an exciting basis for the exploration of new sampling algorithms.

Abbreviations

ABC	approximate Bayesian computation
CESS	conditional effective sample size
CLT	central limit theorem
CMC	consensus Monte Carlo
ESS	effective sample size
GCMC	global consensus Monte Carlo
IID	independent and identically distributed
MCMC	Markov chain Monte Carlo
MCSE	Monte Carlo standard error
MSE	mean squared error
NDPE	nonparametric density product estimation
OLS	ordinary least squares
RJMCMC	reversible jump Markov chain Monte Carlo
SIS	sequential importance sampling
SMC	sequential Monte Carlo
WLS	weighted least squares
WRS	Weierstrass rejection sampling

References

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 Vihola, M. (2018). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2):842–872.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, 37(2):697–725.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373.

Andrieu, C. and Vihola, M. (2015). Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms. *Annals of Applied Probability*, 25(2):1030–1077.

Bardenet, R., Doucet, A., and Holmes, C. (2014). Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. In *Proceedings of the 31st International Conference on Machine Learning*, pages 405–413.

Bardenet, R., Doucet, A., and Holmes, C. (2017). On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(1):1515–1557.

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035.

Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and Distributed Computation: Numerical Methods*, volume 23. Prentice Hall.

Beskos, A., Crisan, D., and Jasra, A. (2014). On the stability of sequential Monte Carlo methods in high dimensions. *Annals of Applied Probability*, 24(4):1396–1445.

Beskos, A., Jasra, A., Kantas, N., and Thiery, A. (2016). On the convergence of adaptive sequential Monte Carlo methods. *Annals of Applied Probability*, 26(2):1111–1146.

Bhadra, A. and Ionides, E. L. (2016). Adaptive particle allocation in iterated sequential Monte Carlo via approximating meta-models. *Statistics and Computing*, 26(1):393–407.

Blum, M. G. and François, O. (2010). Non-linear regression models for approximate Bayesian computation. *Statistics and Computing*, 20(1):63–73.

Bock, H.-H. (2012). Dissimilarity measures for probability distributions. In Bock, H.-H. and Diday, E., editors, *Analysis of Symbolic Data*, pages 153–164. Springer Science & Business Media.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Calderhead, B. and Girolami, M. (2009). Estimating Bayes factors via thermodynamic integration and population MCMC. *Computational Statistics & Data Analysis*, 53(12):4028–4045.

Cérou, F., Del Moral, P., Furon, T., and Guyader, A. (2012). Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, 22(3):795–808.

Cérou, F., Del Moral, P., and Guyader, A. (2011). A nonasymptotic theorem for unnormalized Feynman–Kac particle models. *Annales de l'Institut Henri Poincaré B, Probability and Statistics*, 47:629–649.

Cérou, F., Guyader, A., and Rousset, M. (2019). Adaptive multilevel splitting: Historical perspective and recent results. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(4):043108.

Chan, H. P. and Lai, T. L. (2013). A general theory of particle filters in hidden Markov models and some applications. *Annals of Statistics*, 41(6):2877–2904.

Chen, L. H. and Röllin, A. (2010). Stein couplings for normal approximation. *arXiv preprint arXiv:1003.6039*.

Chen, Y. (2005). Another look at rejection sampling through importance sampling. *Statistics & Probability Letters*, 72(4):277–283.

Chen, Y., Georgiou, T. T., and Pavon, M. (2015). Optimal steering of a linear stochastic system to a final probability distribution, Part I. *IEEE Transactions on Automatic Control*, 61(5):1158–1169.

Chevallier, A., Pion, S., and Cazals, F. (2018). Hamiltonian Monte Carlo with boundary reflections, and application to polytope volume calculations. Research Report RR-9222, INRIA Sophia Antipolis.

Chib, S., Omori, Y., and Asai, M. (2009). Multivariate stochastic volatility. In Andersen, T. G., Davis, R. A., Kreiß, J.-P., and Mikosch, T. V., editors, *Handbook of Financial Time Series*, pages 365–400. Springer.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539–552.

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Annals of Statistics*, 32(6):2385–2411.

Chopin, N. and Ridgway, J. (2017). Leave Pima Indians alone: binary regression as a benchmark for Bayesian computation. *Statistical Science*, 32(1):64–87.

Corless, R. M., Gonnet, G. H., Hare, D. E., Jeffrey, D. J., and Knuth, D. E. (1996). On the Lambert *W* function. *Advances in Computational Mathematics*, 5(1):329–359.

Crooks, G. E. (1998). Nonequilibrium measurements of free energy differences for microscopically reversible Markovian systems. *Journal of Statistical Physics*, 90(5–6):1481–1487.

Csiszár, I. (1963). Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizität von Markoffschen Ketten. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 8:85–108.

Csiszár, I. and Shields, P. C. (2004). Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528.

Dai, H., Pollock, M., and Roberts, G. (2019). Monte Carlo fusion. *Journal of Applied Probability*, 56(1):174–191.

Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Del Moral, P. (2004). *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer Verlag.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436.

Del Moral, P., Doucet, A., and Jasra, A. (2007). Sequential Monte Carlo for Bayesian computation. In Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., and West, M., editors, *Bayesian Statistics 8*, pages 115–148. Oxford University Press.

Del Moral, P., Doucet, A., and Jasra, A. (2012a). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020.

Del Moral, P., Doucet, A., and Jasra, A. (2012b). On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278.

Del Moral, P., Doucet, A., and Singh, S. S. (2010). A backward particle interpretation of Feynman–Kac formulae. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(5):947–975.

Douc, R., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69. IEEE.

Douc, R., Garivier, A., Moulines, E., and Olsson, J. (2011). Sequential Monte Carlo smoothing for general state space hidden Markov models. *Annals of Applied Probability*, 21(6):2109–2145.

Doucet, A., De Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer.

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. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In Crisan, D. and Rozovskii, B., editors, *Handbook of Nonlinear Filtering*. Cambridge University Press.

Doucet, A., Pitt, M. K., Deligiannidis, G., and Kohn, R. (2015). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313.

Earl, D. J. and Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916.

Fan, Y., Leslie, D. S., and Wand, M. P. (2008). Generalised linear mixed model analysis via sequential Monte Carlo sampling. *Electronic Journal of Statistics*, 2:916–938.

Finke, A. (2015). *On extended state-space constructions for Monte Carlo methods*. PhD thesis, University of Warwick.

Finke, A., Doucet, A., and Johansen, A. M. (2020). Limit theorems for sequential MCMC methods. *Advances in Applied Probability*, 52(2). In press.

Friedrichs, K. O. (1944). The identity of weak and strong extensions of differential operators. *Transactions of the American Mathematical Society*, 55(1):132–151.

Friel, N., Hurn, M., and Wyse, J. (2014). Improving power posterior estimation of statistical evidence. *Statistics and Computing*, 24(5):709–723.

Friel, N. and Pettitt, A. N. (2008). Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13(2):163–185.

Gerber, M., Chopin, N., and Whiteley, N. (2019). Negative association, ordering and convergence of resampling methods. *Annals of Statistics*, 47:2236–2260.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, 57(6):1317–1339.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In Keramigas, E., editor, *Computer Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 156–163. Interface Foundation of North America.

Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90(431):909–920.

Ghosh, J. and Ramamoorthi, R. (2003). *Bayesian Nonparametrics*. Springer Science & Business Media.

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Applications of Mathematics: Stochastic Modelling and Applied Probability. Springer.

Glynn, P. W. and Whitt, W. (1992). The asymptotic efficiency of simulation estimators. *Operations Research*, 40(3):505–520.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, pages 107–113. IET.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Green, P. J. and Hastie, D. I. (2009). Reversible jump MCMC. Technical report, University of Bristol.

Hammersley, J. M. and Handscomb, D. C. (1964). *Monte Carlo Methods*. Monographs on Applied Probability and Statistics. Methuen.

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

Heng, J., Doucet, A., and Pokern, Y. (2015). Gibbs flow for approximate transport with applications to Bayesian computation. *arXiv preprint arXiv:1509.08787*.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press.

Huggins, J., Campbell, T., and Broderick, T. (2016). Coresets for scalable Bayesian logistic regression. In *Advances in Neural Information Processing Systems*, pages 4080–4088.

Jasra, A., Holmes, C., and Stephens, D. (2005). Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 20(1):50–67.

Jasra, A., Stephens, D. A., Doucet, A., and Tsagaris, T. (2011). Inference for Lévy-driven stochastic volatility models via adaptive sequential Monte Carlo. *Scandinavian Journal of Statistics*, 38(1):1–22.

Jerrum, M. and Sinclair, A. (1996). The Markov chain Monte Carlo method: an approach to approximate counting and integration. In Hochbaum, D., editor, *Approximation Algorithms for NP-hard Problems*, pages 482–519. PWS Publishing.

Johansen, A. M., Del Moral, P., and Doucet, A. (2006). Sequential Monte Carlo samplers for rare events. In *Proceedings of the 6th International Workshop on Rare Event Estimation*, pages 256–267.

Jordan, M. I., Lee, J. D., and Yang, Y. (2019). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 114(526):668–681.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kennedy, A. D. and Kuti, J. (1985). Noise without noise: a new Monte Carlo method. *Physical Review Letters*, 54(23):2473.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288.

Korattikara, A., Chen, Y., and Welling, M. (2014). Austerity in MCMC land: Cutting the Metropolis–Hastings budget. In *Proceedings of the 31st International Conference on Machine Learning*.

Kostov, S. and Whiteley, N. (2017). An algorithm for approximating the second moment of the normalizing constant estimate from a particle filter. *Methodology and Computing in Applied Probability*, 19(799–818):3.

Kuo, F. Y. and Sloan, I. H. (2005). Lifting the curse of dimensionality. *Notices of the American Mathematical Society*, 52(11):1320–1328.

Lagnoux, A. (2006). Rare event simulation. *Probability in the Engineering and Informational Sciences*, 20(1):45–66.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, 38(4):963–974.

Lartillot, N. and Philippe, H. (2006). Computing Bayes factors using thermodynamic integration. *Systematic Biology*, 55(2):195–207.

Lee, A. and Whiteley, N. (2018). Variance estimation in the particle filter. *Biometrika*, 105(3):609–625.

Lin, L., Liu, K., and Sloan, J. (2000). A noisy Monte Carlo algorithm. *Physical Review D*, 61(7):074505.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576.

Locatelli, M. (2000). Simulated annealing algorithms for continuous global optimization: convergence conditions. *Journal of Optimization Theory and Applications*, 104(1):121–133.

Maclaurin, D. and Adams, R. P. (2014). Firefly Monte Carlo: Exact MCMC with subsets of data. In *Proceedings of the 30th International Conference on Uncertainty in Artificial Intelligence*, pages 543–552.

Marin, J. M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

Marinari, E. and Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters*, 19(6):451–458.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Meyn, S. P. and Tweedie, R. L. (2009). *Markov chains and stochastic stability*. Cambridge University Press, 2nd edition.

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. (2014). Scalable and robust Bayesian inference via the median posterior. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1656–1664.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Neal, R. M. (1996). Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6(4):353–366.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

Neiswanger, W., Wang, C., and Xing, E. (2014). Asymptotically exact, embarrassingly parallel MCMC. In *Proceedings of the 30th International Conference on Uncertainty in Artificial Intelligence*, pages 623–632.

Olsson, J. and Douc, R. (2019). Numerically stable online estimation of variance in particle filters. *Bernoulli*, 25(2):1504–1535.

Paulin, D., Jasra, A., and Thiery, A. (2019). Error bounds for sequential Monte Carlo samplers for multimodal distributions. *Bernoulli*, 25(1):310–340.

Pavlichin, D. S. and Weissman, T. (2016). Chained Kullback–Leibler divergences. In *IEEE International Symposium on Information Theory*, pages 580–584.

Pitt, M. K., dos Santos Silva, R., Giordani, P., and Kohn, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151.

Quiroz, M., Kohn, R., Villani, M., and Tran, M.-N. (2019). Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, 114(526):831–843.

Rabinovich, M., Angelino, E., and Jordan, M. I. (2015). Variational consensus Monte Carlo. In *Advances in Neural Information Processing Systems*, pages 1207–1215.

Rendell, L. J., Johansen, A. M., Lee, A., and Whiteley, N. (2018). Global consensus Monte Carlo. *arXiv preprint arXiv:1807.09288*.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis–Hastings algorithms. *Statistical Science*, 16(4):351–367.

Roberts, G. O. and Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71.

Rubenthaler, S., Rydén, T., and Wiktorsson, M. (2009). Fast simulated annealing in $\mathbb{R}^d$ with an application to maximum likelihood estimation in state-space models. *Stochastic Processes and their Applications*, 119(6):1912–1931.

Sason, I. and Verdú, S. (2016). f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006.

Schäfer, C. and Chopin, N. (2013). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing*, 23(2):163–184.

Scheffé, H. (1947). A useful convergence theorem for probability distributions. *Annals of Mathematical Statistics*, 18(3):434–438.

Scott, S. L. (2017). Comparing consensus Monte Carlo strategies for distributed Bayesian computation. *Brazilian Journal of Probability and Statistics*, 31(4):668–685.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88.

SenGupta, A. (1987). Tests for standardized generalized variances of multivariate normal populations of possibly different dimensions. *Journal of Multivariate Analysis*, 23(2):209–219.

Sherlock, C., Thiery, A. H., Roberts, G. O., and Rosenthal, J. S. (2015). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Annals of Statistics*, 43(1):238–275.

Shiryaev, A. N. (1996). *Probability*, volume 95 of *Graduate Texts in Mathematics*. Springer Verlag.

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765.

Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pages 912–920.

van der Vaart, A. W. (2000). *Asymptotic Statistics*, volume 3 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press.

Vats, D., Flegal, J. M., and Jones, G. L. (2019). Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*, 106(2):321–337.

Villén-Altamirano, M. and Villén-Altamirano, J. (1991). RESTART: A method for accelerating rare event simulations. *Queueing, Performance and Control in ATM*, 3:71–76.

Vono, M., Dobigeon, N., and Chainais, P. (2018). Sparse Bayesian binary logistic regression using the split-and-augmented Gibbs sampler. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*.

Vono, M., Dobigeon, N., and Chainais, P. (2019a). Split-and-augmented Gibbs sampler — application to large-scale inference problems. *IEEE Transactions on Signal Processing*, 67(6):1648–1661.

Vono, M., Paulin, D., and Doucet, A. (2019b). Efficient MCMC sampling with dimension-free convergence rate using ADMM-type splitting. *arXiv preprint arXiv:1905.11937*.

Wang, X. and Dunson, D. B. (2013). Parallelizing MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605*.

Wang, X., Guo, F., Heller, K. A., and Dunson, D. B. (2015). Parallelizing MCMC with random partition trees. In *Advances in Neural Information Processing Systems*, pages 451–459.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688.

Whiteley, N., Lee, A., and Heine, K. (2016). On the role of interaction in sequential Monte Carlo algorithms. *Bernoulli*, 22(1):494–529.

Xu, M., Lakshminarayanan, B., Teh, Y. W., Zhu, J., and Zhang, B. (2014). Distributed Bayesian posterior sampling via moment sharing. In *Advances in Neural Information Processing Systems*, pages 3356–3364.

Zayed, A. I. (1996). *Handbook of Function and Generalized Function Transformations*. CRC Press.

Zhou, E. and Chen, X. (2013). Sequential Monte Carlo simulated annealing. *Journal of Global Optimization*, 55(1):101–124.

Zhou, Y., Johansen, A. M., and Aston, J. A. (2016). Towards automatic model comparison: an adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics*, 25(3):701–726.