Axel Finke

On Extended State-Space Constructions for Monte Carlo Methods

Thesis submitted for the degree of Doctor of Philosophy

Department of Statistics
University of Warwick
July 2015

Typeset in LaTeX using Linux Libertine & KOMA-Script.

to my family

Summary

This thesis develops computationally efficient methodology in two areas. Firstly, we consider a particularly challenging class of discretely observed continuous-time point-process models. For these, we analyse and improve an existing filtering algorithm based on *sequential Monte Carlo* (SMC) methods. To estimate the static parameters in such models, we devise novel particle Gibbs samplers. One of these exploits a sophisticated non-centred parametrisation whose benefits in a *Markov chain Monte Carlo* (MCMC) context have previously been limited by the lack of blockwise updates for the latent point process. We apply this algorithm to a Lévy-driven stochastic volatility model. Secondly, we devise novel Monte Carlo methods – based around pseudo-marginal and conditional SMC approaches – for performing optimisation in latent-variable models and more generally. To ease the explanation of the wide range of techniques employed in this work, we describe a generic importance-sampling framework which admits virtually all Monte Carlo methods, including SMC and MCMC methods, as special cases. Indeed, hierarchical combinations of different Monte Carlo schemes such as SMC within MCMC or SMC within SMC can be justified as repeated applications of this framework.

Declaration

This thesis is the result of my own work and research, except where otherwise indicated. Chapter 4 is a condensed version of the published journal article

> Finke, A., Johansen, A. M. & Spanò, D. (2014). Static-parameter estimation in piecewise deterministic processes using particle Gibbs samplers. *Annals of the Institute of Statistical Mathematics*, *66*(3), 577–609.

This thesis has not been submitted for examination to any other institution than the University of Warwick.

Axel Finke

Acknowledgements

There are many people without whom I could not have written this thesis or undertaken the research on which it is based.

First, I would like to thank my supervisors, Adam M. Johansen and Dario Spanò, for their patience, guidance, refreshing sarcasm, and constant support throughout the past three-and-a-half years. I consider myself extremely fortunate for the substantial amount of enthusiasm and intuition for Monte Carlo methods they have bestowed upon me.

For significantly improving the presentation of this thesis, in particular of Chapters 1, 3 and 6, I am indebted to my other proofreaders: Cyril Chimisov, Andreas L. Hetland, and Felipe Medina-Aguayo.

I have benefited immensely from stimulating discussions with my colleagues (in no particular order): Murray Pollock, Giacomo Zanella, and Kirsty Hey, for which I am deeply grateful. I would also like acknowledge all the other participants of the Feynman–Kac and Markov chain Monte Carlo reading groups, and the various other seminars on computational statistics at Warwick. By furthering my understanding of the topic, they have all, in some way, contributed to this thesis.

I would also like to thank Gareth O. Roberts and Nick Whiteley for taking the time to read and examine my thesis and for taking an interest in my research.

Furthermore, I wish to express my thanks to Professor Mark Trede for introducing me to computational statistics and for encouraging me to study in the UK.

This work was also generously supported by Engineering and Physical Sciences Research Council Doctoral Training Grant EP/J500586/1.

Finally, I would like to thank my family for encouragement and support, and in particular my brother Till, for invaluable typographical advice.

Contents

II Some Novel Monte Carlo Schemes

List of Figures

Introduction

Context

Since its invention in the 1940s, the idea of approximating integrals by random samples, known as the Monte Carlo method, has served as a vital tool for scientific discovery in a wide range of disciplines such as biology (Wilkinson, 2011), econometrics (Greenberg, 2012; Durbin & Koopman, 2012), engineering (Cappé, Moulines & Rydén, 2005), epidemiology (Gibson & Renshaw, 1998; O'Neill & Roberts, 1999), operations research (Fishman, 1996), physics (Spanier & Gelbard, 1969; Sokal, 1997; Lapeyre, Pardoux & Sentis, 2003), and political science (Gelman et al., 2013)

In addition, the Monte Carlo method has spurred technological progress appreciable in everyday life. For instance, it now aids the tracking and positioning of mobile robots (Dellaert, Fox, Burgard & Thrun, 1999), produces weather forecasts (Epstein, 1969; Leith, 1974), predicts elections ('FiveThirtyEight', 2015), prices complicated financial instruments (Glasserman, 2004), and generates visual effects in blockbuster movies from animation studios such as Pixar (Veach & Guibas, 1995; Lokovic & Veach, 2000) – even leading to a Technical Oscar for Thomas Lokovic and Eric Veach ('The 86th Scientific & Technical Awards', 2014).

This thesis is largely concerned with developing sophisticated instances of the Monte Carlo method tailored to particular challenging real-world problems. To that end, we combine, extend, and improve a number of existing algorithms. As a by-product, we provide a unifying Monte Carlo framework which admits new insight into the relationship between the vast array of complex Monte Carlo algorithms that exist today.

Throughout, we view the Monte Carlo method as a technique for approximating measures and, by extension, as a technique for approximating integrals. Hence, we make heavy use of measure-theoretic notation. We hope that the reader is not discouraged by this presentation. In fact, this thesis only uses basic undergraduate-level mathematical techniques.

Outline

This thesis is divided into two parts. Part I provides some background on various Monte Carlo algorithms. Novel methodology is mostly, but not exclusively, confined to Part II.

Part I. In the first part, we briefly review basic Monte Carlo methodology, such as *importance sampling* (IS), *sequential Monte Carlo* (SMC) methods, and *Markov chain Monte Carlo* (MCMC) methods. To provide the reader with a better intuition for such a plethora of techniques, we present a generic IS framework, best described as *marginalised one-sample importance sampling* (MOSIS), which admits essentially all instances of the Monte Carlo method, including those mentioned above, as special cases.

Chapter 1 reviews IS as a particularly useful interpretation of the Monte Carlo method. We also describe self-normalised IS as well as state-space extension and state-space reduction techniques. Combining these ideas, we then develop the generic MOSIS framework which forms the heart of any Monte Carlo algorithm. We also show that this framework can, for instance, be used to justify pseudo-marginal approaches.

Chapter 2 devises a generic SMC scheme and demonstrates that it admits essentially any SMC algorithm as a special case, including, for instance, the discrete particle filter which could hitherto not be viewed as a standard SMC algorithm. In turn, we show that the generic SMC algorithm is itself a special case of MOSIS. In addition, we generalise and improve existing schemes which approximate integrals by recycling all particles generated by an SMC sampler.

Chapter 3 shows that MCMC methods, too, can be viewed as MOSIS and that a repeated, hierarchical application of MOSIS forms the basis of all MCMC kernels, including pseudo-marginal, randomised, and ensemble MCMC kernels as well as multiple-proposal Metropolis–Hastings and conditional sequential Monte Carlo kernels. This circumvents the need for checking sufficient conditions such as detailed balance individually for each of these kernels. We also prove the hitherto unknown result that the variance-reduction techniques for conditional sequential Monte Carlo kernels: backward sampling and ancestor sampling share the same extended target distribution. Finally, we comment on the relationship between particle MCMC and ensemble MCMC methods which, to our knowledge, has also not been investigated in the literature.

Part II. In the second part, we develop novel Monte Carlo methodology for two problems. Firstly, we devise methods for filtering and static-parameter estimation in a class of discretely observed continuous-time piecewise deterministic processes. These may also be viewed as partially-observed point processes. Secondly, we construct efficient Monte Carlo algorithms for optimisation in latent-variable models and more generally.

Chapter 4 motivates the use of piecewise deterministic processes and reviews, analyses and improves an existing SMC-based filter for such models. Around it, we also devise a particle Gibbs sampler – with a novel auxiliary-variable rejuvenation step – to perform static-parameter estimation.

Chapter 5 considers static-parameter estimation in partially-observed piecewise deterministic processes driven by compound Poisson processes. To improve mixing of the particle Gibbs chain, we adopt a non-centred parametrisation. The resulting algorithm is applied to a particularly challenging Lévy-driven stochastic volatility model.

Chapter 6 develops a framework for performing optimisation, e.g. for maximum likelihood or maximum a-posteriori estimation in latent-variable models. Specifically, we devise generic SMC and MCMC optimisation schemes within which sophisticated Monte Carlo approaches such as pseudo-marginal methods or particle Gibbs samplers can be incorporated.

A detailed list of novel contributions can be found on Page 190.

Notation

It may be helpful to clarify some notational conventions used throughout this work although non-standard notation is also explained in the main text on the first use. For easy reference, we also provide a list of frequently used symbols on Page 201 along with a list of acronyms on Page 203.

Sets. We denote by $\mathbb{R} \supseteq \mathbb{Z} \supseteq \mathbb{N}$, respectively, the sets of real numbers, integers and positive integers. We often make use of the following subsets of the latter two: $\mathbb{Z}_{k,l} := \{z \in \mathbb{Z} \mid k \leq z \leq l\}$ and $\mathbb{N}_l := \mathbb{Z}_{1,l}$. Furthermore, $\#A$ denotes the cardinality of some countable set A. Finally, $A_{1:n}^{\times} := \bigtimes_{p=1}^{n} A_p := A_1 \times \dots \times A_n$ represents the Cartesian product of sets $A_1, \dots, A_n$.

Vectors. We write $x_t^{1:N} := (x_t^1, \ldots, x_t^N)$ and $x_{1:T}^n := (x_1^n, \ldots, x_T^n)$. To avoid ambiguity, we often use the bold face notation $\boldsymbol{x}_t := x_t^{1:N}$ when both sub- and superscripts need to be vector valued. In this case, we often let $\boldsymbol{x}_t^{-k} := (x_t^{1:k-1}, x_t^{k+1:N})$ be the vector $\boldsymbol{x}$ without its kth component. Finally, A^{T} denotes the transpose of some matrix A.

Measures. All measures considered in this work will be positive. We write $\mathcal{M}_\sigma(\mathsf{X}) \supseteq \mathcal{M}(\mathsf{X}) \supseteq \mathcal{M}_1(\mathsf{X})$ for the sets of (positive) σ-finite, finite, and probability measures on some measurable space $(\mathsf{X}, \mathcal{X})$. Whenever possible, we take, $\mathcal{X} =: \mathcal{B}(\mathsf{X})$, where $\mathcal{B}(\mathsf{X})$ is the Borel σ-algebra on X. In this case, we refer to elements of the above-mentioned sets as measures 'on X'. In particular, $\mathrm{Leb} \in \mathcal{M}_\sigma(\mathbb{R})$ denotes the Lebesgue measure on $\mathbb{R}$. For $\nu, \mu \in \mathcal{M}_\sigma(\mathsf{X})$, we write $\nu \ll \mu$ if ν absolutely continuous with respect to μ and in this case, $\mathrm{d}\nu/\mathrm{d}\mu$ denotes the corresponding Radon–Nikodým derivative, i.e. $\nu = [\mathrm{d}\nu/\mathrm{d}\mu]\mu$. It is sometimes convenient to abuse the notation for Radon–Nikodým derivatives and to alternatively write $[\mathrm{d}\nu/\mathrm{d}\mu](x) =: \nu(\mathrm{d}x)/\mu(\mathrm{d}x)$.

Functions. For measurable spaces $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$, endowed with suitable σ-algebras $\mathcal{X}$ and $\mathcal{Y}$, we define

$$\mathcal{F}(\mathsf{X}, \mathsf{Y}) := \bigl\{ f\colon \mathsf{X} \to \mathsf{Y} \bigm| f \text{ is } \mathcal{X}/\mathcal{Y}\text{-measurable} \bigr\}.$$

Furthermore, we let $\mathrm{id}_{\mathsf{X}} \in \mathcal{F}(\mathsf{X}, \mathsf{X})$ denote the identity function and let $\mathbb{1}_B \in \mathcal{F}(\mathsf{X}, \{0, 1\})$ represent the indicator function of $B \subseteq \mathsf{X}$, i.e.

$$\mathbb{1}_B(x) := \begin{cases} 1, & \text{if } x \in B, \\ 0, & \text{if } x \in \mathsf{X} \setminus B. \end{cases}$$

If $B = \mathsf{X}$, we set $\mathbb{1}_{\mathsf{X}} =: \mathbb{1}$. For any function $f\colon \mathsf{X} \to \mathsf{Y}$, and $B \subseteq \mathsf{Y}$, the preimage of B under f is denoted $f^{-1}(B) := \{ x \in \mathsf{X} \mid f(x) \in B \}$. For functions f and g with domain X and Y, respectively, we use the tensor-product notation $[f \otimes g](x, y) := (f(x), g(y))$, for $(x, y) \in \mathsf{X} \times \mathsf{Y}$. Furthermore, if $\mathsf{X} = \mathsf{Y}$, we write $fg(x) := f(x)g(x)$, for $x \in \mathsf{X}$, as usual.

Integrals. For any $\mu \in \mathcal{M}_\sigma(\mathsf{X})$ and any $p \in [0, \infty)$, we let

$$\mathcal{L}^p(\mu) := \bigl\{ f \in \mathcal{F}(\mathsf{X}, \mathbb{R}) \bigm| \mu(|f|^p) < \infty \bigr\}$$

denote the set of p-times μ-integrable real-valued functions with the convention that $\mathcal{L}^1(\mu) =: \mathcal{L}(\mu)$, and with the following convenient shorthand for integrals: $\mu(f) := \int_{\mathsf{X}} f \,\mathrm{d}\mu$.

Kernels. For measurable spaces $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$, a (positive) σ-finite kernel is a function $K\colon \mathsf{X} \times \mathcal{Y} \to [0, \infty)$ if it satisfies both the following properties:

$$\begin{aligned} \forall\, A \in \mathcal{Y}: &\quad K(\,\cdot\,, A) \in \mathcal{F}(\mathsf{X}, [0, \infty)), \\ \forall\, x \in \mathsf{X}: &\quad K(x, \,\cdot\,) \in \mathcal{M}_\sigma(\mathsf{Y}). \end{aligned}$$

We call K, *finite* if $K(x, \cdot) \in \mathcal{M}(\mathsf{Y})$, and *stochastic* if $K(x, \cdot) \in \mathcal{M}_1(\mathsf{Y})$, for any $x \in \mathsf{X}$. We denote by $\mathcal{K}_\sigma(\mathsf{X}, \mathsf{Y}) \supseteq \mathcal{K}(\mathsf{X}, \mathsf{Y}) \supseteq \mathcal{K}_1(\mathsf{X}, \mathsf{Y})$ the sets of σ-finite, finite, and stochastic transition kernels from $(\mathsf{X}, \mathcal{X})$ to $(\mathsf{Y}, \mathcal{Y})$. For suitable kernels $K \in \mathcal{K}_\sigma(\mathsf{X}, \mathsf{Y})$ and $L \in \mathcal{K}_\sigma(\mathsf{Y}, \mathsf{Z})$, we may define a kernel $K \otimes L \in \mathcal{K}_\sigma(\mathsf{X}, \mathsf{Y} \times \mathsf{Z})$ by

$$[K \otimes L](x, A \times B) := K(x, \mathbb{1}_A L(\,\cdot\,, \mathbb{1}_B)),$$

for all $(x, A, B) \in \mathsf{X} \times \mathsf{Y} \times \mathcal{Z}$, and define a kernel $KL \in \mathcal{K}_\sigma(\mathsf{X}, \mathsf{Z})$ by

$$KL(x, B) := [K \otimes L](x, \mathsf{Y} \times B),$$

for all $(x, B) \in \mathsf{X} \times \mathcal{Z}$.

By extension, we set $K_{1:n}^{\otimes} := \bigotimes_{p=1}^{n} K_p := K_1 \otimes \cdots \otimes K_n$, for suitable kernels $K_1, \ldots, K_n$. In particular, if $K_1 = \cdots = K_n = K$, we use the shorthand $K_{1:n}^{\otimes} =: K^{\otimes n}$.

The same conventions for (tensor)products apply to measures by viewing them as kernels which are constant in their first argument. Finally, especially in Part II, we write a stochastic kernel $K \in \mathcal{K}_1(\mathsf{X}, \mathsf{Y})$ as $K(x, \mathrm{d}y) = K(\mathrm{d}y|x)$, for $x \in \mathsf{X}$. In particular, the distribution $K(\mathrm{d}y|x)$ is then sometimes implicitly defined to be the full conditional distribution of the second component under the probability measure $K \in \mathcal{M}_1(\mathsf{X} \times \mathsf{Y})$.

Distributions. We generally work with some underlying probability space $(\Omega, \mathcal{A}, \mathbb{P})$ and let $\mathbb{E}_\mu, \mathbb{V}_\mu$ denote expectation and variance under some probability measure μ, with the convention that $\mathbb{E}_{\mathbb{P}} = \mathbb{E}$ and $\mathbb{V}_{\mathbb{P}} = \mathbb{V}$. Hence, for a *random variable* $X \in \mathcal{F}(\Omega, \mathbb{R}^d)$ with distribution $\mu := \mathbb{P} \circ X^{-1} \in \mathcal{M}_1(\mathsf{X})$, where $X^{-1}(A)$ is the preimage of A under X, and for $f \in \mathcal{L}(\mu)$, we have the usual identity $\mu(f) = \mathbb{E}_\mu[f] = \mathbb{E}[f(X)]$. For this work, important probability measures are $\mathrm{N}_{\mu,\Sigma}$, the normal distribution with mean μ and covariance matrix Σ, and δ_x, the Dirac measure or point mass centred at x, defined by $\delta_x(A) := \mathbb{1}_A(x)$. For a full list of standard distributions used in this work, see Page 201.

Part I

Generic Monte Carlo Framework

1 Elementary Monte Carlo Tools

1.1 Importance Sampling

1.1.1 Motivation

In this chapter, we describe tools that form the basis of all known Monte Carlo algorithms. In Sections 1.1 and 1.2, we justify importance sampling and self-normalised importance sampling. In Section 1.3, we describe state-space extension techniques and also ways of reducing the dimension of the state space (i.e. Rao–Blackwellisation). Finally, Section 1.4 combines the ideas from the preceding sections into a generic importance-sampling framework which admits essentially all known Monte Carlo schemes as a special case.

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be some probability space and let $\mathcal{M}(\mathsf{X})$ denote the set of finite positive measures on some measurable space $(\mathsf{X}, \mathcal{B}(\mathsf{X}))$. For $\gamma \in \mathcal{M}(\mathsf{X})$, assume that we want to calculate integrals

$$\gamma(f) := \int_{\mathsf{X}} f \, \mathrm{d}\gamma, \tag{1.1}$$

for all test functions $f \in \mathsf{F} \subseteq \mathcal{L}(\gamma) := \{\, f : \mathsf{X} \to \mathbb{R} \mid f \text{ is } \gamma\text{-integrable} \,\}$.

1.1 Remark. *Let $\mathcal{M}_\sigma(\mathsf{X})$ denote the set of σ-finite (positive) measures on $(\mathsf{X}, \mathcal{B}(\mathsf{X}))$. Note that any finite integral $\tilde{\gamma}(\tilde{f})$ with respect to $\tilde{\gamma} \in \mathcal{M}_\sigma(\mathsf{X})$ may be written in the form of Equation 1.1 by applying the change of measure $\gamma := \tilde{f}\tilde{\gamma}$ (where $\tilde{f}\tilde{\gamma}(A) := \tilde{\gamma}(\tilde{f}\mathbb{1}_A)$, for $A \in \mathcal{B}(\mathsf{X})$), and setting $f \equiv 1$.*

Analytical computation of such integrals is often too costly or even impossible. Instead, numerical integration methods such as quadrature rules may be used. Unfortunately, the error of these methods is typically of order $\mathcal{O}(N^{-c/d})$, where d is the dimension of the state space X, $c > 0$, and N is the number of grid points. The need for N to be exponentially large in d is known as the *curse of dimensionality* (Bellman, 1957). It prohibits the use of standard numerical integration methods in higher dimensions.

1.1.2 Change of Measure

To apply the methods developed in this work, we need to turn $\gamma(f)$ into an integral with respect some probability measure. This can always be achieved as follows. Let $\mathcal{M}_1(\mathsf{X})$ denote the set of probability measures on $(\mathsf{X}, \mathcal{B}(\mathsf{X}))$ and select $\psi \in \mathcal{M}_1(\mathsf{X})$ such that $\gamma \ll \psi$. In this case,

$$\gamma(f) = \psi(wf) = \mathbb{E}_{\psi}[wf],$$

where w denotes the Radon–Nikodým derivative $w := \mathrm{d}\gamma/\mathrm{d}\psi$.

1.2 Remark. *It suffices that $f\gamma \ll \psi$. However, we make the stronger requirement: $\gamma \ll \psi$, here because it is independent of the particular test function and we are usually concerned with approximating $\gamma(f)$ for a large class of test functions,* F.

1.1.3 Sampling-Based Approximation

Given a vector of *independent and identically distributed* (IID) draws $\boldsymbol{X} := X^{1:N}$ from ψ, (a suitable version of) the Glivenko–Cantelli theorem (Billingsley, 2012, Theorem 20.6) justifies using the empirical measure of these samples to approximate the *probability measure* ψ,

$$\psi^{\mathrm{MC},N} := \frac{1}{N}\sum_{n=1}^{N} \delta_{X^n}.$$

Here, $\delta_x \in \mathcal{M}_1(\mathsf{X})$ is the point mass (or *Dirac* measure) located at $x \in \mathsf{X}$. We may thus approximate the *integral* $\gamma(f) = \psi(wf) = \mathbb{E}_{\psi}[wf]$ by

$$\psi^{\mathrm{MC},N}(wf) = \int_{\mathsf{X}} wf \,\mathrm{d}\psi^{\mathrm{MC},N} = \frac{1}{N}\sum_{n=1}^{N} wf(X^n),$$

i.e. we estimate the expectation by the corresponding sample mean.

Approximating integrals with respect to some probability measure ψ in this way – usually by generating $\boldsymbol{X}$ on a computer using pseudo-random numbers – is known as the *Monte Carlo* method. It was developed during the 1940s by John Von Neumann, Stanislav M. Ulam, Nicholas C. Metropolis and other mathematicians and physicists working on the Manhattan

Project at the Los Alamos Scientific Laboratory, New Mexico. Some of the earliest available references include Goertzel and Kahn (1949), Metropolis and Ulam (1949). Historical accounts can be found in Metropolis (1987), Eckhard (1987), Rota (2008).

Note that we may also view the Monte Carlo method as a procedure for approximating the *measure* γ by the following (random) unnormalised *weighted* empirical measure,

$$\gamma^{\textsc{is},N} := \frac{1}{N}\sum_{n=1}^{N} w(X^n)\delta_{X^n},$$

where $w(X^n)$ is known as an unnormalised *importance weight.*

1.3 Remark. *Throughout this work, for simplicity, we refer to the unnormalised importance weights $w(X^n)$ simply as 'weights' or 'importance weights' and we refer to $\gamma^{\textsc{is},N}$ as a 'weighted empirical measure', even though $\sum_{n=1}^{N} w(X^n) \neq 1$, in general.*

This view of the Monte Carlo method is known as *importance sampling* (IS) (Goertzel & Kahn, 1949). Though, initially, it was also referred to as *quota sampling* (Goertzel, 1949). Plugging in the test function, it clearly leads to the same estimator for $\gamma(f)$ as above, i.e. $\gamma^{\textsc{is},N}(f) = \psi^{\textsc{mc},N}(wf)$. One of the goals of Part I of this work is to demonstrate that essentially every Monte Carlo algorithm can be seen as a special case of IS which, in turn, is merely a convenient re-interpretation of the Monte Carlo method.

The IS-interpretation of the Monte Carlo method is useful because we are usually concerned with approximating $\gamma(f)$ for a large class of test functions, F, but without selecting and sampling from a different *proposal distribution* ψ for each $f \in \mathsf{F}$ as this tends to be costly. The IS-interpretation helps separating the influence of the test function f from the influence of the proposal distribution ψ on the estimator $\gamma^{\textsc{is},N}(f)$ through the oscillations of w.

1.1.4 Theoretical Properties

It is easy to see that $\gamma^{\textsc{is},N}(f)$ is an unbiased estimator for $\gamma(f)$ and the *strong law of large numbers* (SLLN) (Billingsley, 2012, Theorem 22.1) ensures that it converges almost surely to $\gamma(f)$, as $N \to \infty$. Define the set $\mathcal{L}^i(\psi) := \{ f : \mathsf{X} \to \mathbb{R} \mid f^i \text{ is } \psi\text{-integrable} \}$. If $wf \in \mathcal{L}^2(\psi)$, so that the asymptotic variance $\sigma^2(f) := \mathbb{V}_\gamma[f]$ exists, a simple *central limit theorem* (CLT) (Billingsley, 2012, Theorem 27.1) guarantees that

$$\sqrt{N}\big[\gamma^{\mathrm{IS},N}(f) - \gamma(f)\big] \quad \overset{N\to\infty}{\longrightarrow} \quad Z \sim \mathrm{N}_{0,\sigma^2(f)},$$

in distribution. In particular, the Monte Carlo error $|\gamma^{\mathrm{IS},N}(f) - \gamma(f)|$ vanishes at a rate $\mathcal{O}_{\mathbb{P}}(N^{-1/2})$ which is independent of the dimension d.

1.2 Self-Normalised Importance Sampling

1.2.1 Motivation

Assume that we wish to approximate $\pi(f)$, for some probability measure $\pi \in \mathcal{M}_1(\mathsf{X})$ and $f \in \mathcal{L}(\pi)$. In many applications, the Radon–Nikodým derivative $\widetilde{w} := \mathrm{d}\pi/\mathrm{d}\psi$ can only be evaluated up to some unknown constant $\mathfrak{z} > 0$. That is, we can evaluate $w := \mathfrak{z}\widetilde{w}$ point-wise but not $\widetilde{w}$.

The intractability of $\mathfrak{z}$ renders (standard) IS inapplicable because we can then only approximate the measure $\gamma \in \mathcal{M}(\mathsf{X})$, defined by

$$\gamma := w\psi = \mathfrak{z}\pi$$

and $\gamma \neq \pi$, except in the trivial case that $\mathfrak{z} = 1$. *Self-normalised* IS exploits fact that if π is a probability measure then $\mathfrak{z} = \gamma(\mathbb{1})$. Based on this identity, we may use standard IS to separately approximate the numerator and denominator in the identity $\pi = \gamma/\mathfrak{z}$ as described in the next subsection.

We conclude this subsection by noting that the intractability of $\mathfrak{z}$ usually arises because the target distribution π or the proposal distribution ψ are constructed via non-linear transformations of some other measures on X. More precisely, for $\nu \in \mathcal{M}_\sigma(\mathsf{X})$ and $g \in \mathcal{L}(\nu)$, define the *Boltzmann–Gibbs transformation* of ν under g, Ψ_g, by

$$\Psi_g(\nu) := g\nu/\nu(g).$$

Assume that π is constructed via $\pi = \Psi_g(\varpi)$ and ψ is constructed via $\psi = \Psi_h(\eta)$, for some $\varpi, \eta \in \mathcal{M}_\sigma(\mathsf{X})$, $g \in \mathcal{L}(\varpi)$, and $h \in \mathcal{L}(\eta)$. In this case, we often have that $\mathfrak{z} = \varpi(g)/\eta(h)$ and this ratio is usually intractable.

1.4 Example (Bayesian posterior). *In Bayesian statistics, π may be the posterior distribution given some prior distribution ϖ and some data y with associated likelihood $\tilde{g}(\,\cdot\,, y) =: g = L$, i.e. $\pi = \Psi_L(\varpi)$. The 'model evidence' or 'marginal likelihood', $\varpi(L)$, is then typically intractable.*

1.2.2 Sampling-Based Approximation

The idea of self-normalised IS is to separately approximate the numerator and denominator on the right hand side in the identity $\pi = \gamma/\mathfrak{z}$ via standard IS. That is, we approximate the measure γ by

$$\gamma^{\mathrm{IS},N} = \frac{1}{N}\sum_{n=1}^{N} w(X^n)\delta_{X^n}$$

and the integral $\mathfrak{z} = \gamma(\mathbb{1})$ by

$$\mathfrak{z}^{\mathrm{IS},N} := \gamma^{\mathrm{IS},N}(\mathbb{1}) = \frac{1}{N}\sum_{n=1}^{N} w(X^n).$$

Let $W^n(\boldsymbol{X}) := w(X^n)/\mathfrak{z}^{\mathrm{IS},N}$ denote the nth *self-normalised importance weight.* Combining the previous two approximations then defines the self-normalised IS approximation of π,

$$\pi^{\mathrm{IS}\star,N} = \frac{\gamma^{\mathrm{IS},N}}{\mathfrak{z}^{\mathrm{IS},N}} = \sum_{n=1}^{N} W^n(\boldsymbol{X})\delta_{X^n}.$$

Again, we can approximate the integral $\pi(f)$ by

$$\pi^{\mathrm{IS}\star,N}(f) = \sum_{n=1}^{N} W^n(\boldsymbol{X}) f(X^n).$$

1.2.3 Theoretical Properties

The estimator $\pi^{\mathrm{IS}\star,N}(f)$ is biased but strongly consistent, i.e. $\pi^{\mathrm{IS}\star,N}(f)$ converges almost surely to $\pi(f)$, as $N \to \infty$. As shown by Geweke (1989), for instance, the estimator again satisfies a CLT, i.e.

$$\sqrt{N}\big[\pi^{\mathrm{IS}\star,N}(f) - \pi(f)\big] \quad \overset{N\to\infty}{\longrightarrow} \quad Z \sim \mathrm{N}_{0,\sigma^2(f)},$$

in distribution, if we assume that $w, wf \in \mathcal{L}^2(\psi)$ to ensure that the asymptotic variance $\sigma^2(f) := \pi(\tilde{w}[f - \pi(f)]^2)$ exists. More precisely, Liu (2001, p. 35) proves that

$$\begin{aligned} \mathbb{E}\big[\pi^{\text{IS}\star,N}(f)\big] &= \pi(f) + \frac{\pi(\tilde{w}[f - \pi(f)])}{N} + \mathcal{O}(N^{-2}), \\ \mathbb{V}\big[\pi^{\text{IS}\star,N}(f)\big] &= \frac{\pi(\tilde{w}[f - \pi(f)]^2)}{N} + \mathcal{O}(N^{-2}). \end{aligned}$$

In particular, $\mathfrak{z}^{\text{IS},N}$ is again an IS estimate of $\mathfrak{z}$ and thus unbiased.

Finally, even though $\pi^{\text{IS}\star,N}(f)$ is biased, its mean-square error can sometimes be smaller than that of a standard IS estimate $\pi^{\text{IS},N}(f)$ (assuming that $\tilde{w}$ can be evaluated). Intuitively, the former can obtain variance reductions by exploiting the fact that π is a (random) probability measure. Indeed, $\pi^{\text{IS}\star,N}$ is a (random) probability measure but $\pi^{\text{IS},N}$ is not, in general, because its weights do not sum to 1.

1.2.4 Effective Sample Size

From the expression for the asymptotic variance $\sigma^2(f)$ above, it is clear that the performance of self-normalised IS is determined by how closely the target distribution π resembles the proposal distribution ψ, at least if we neglect the contribution from the oscillations of f.

Several criteria have been proposed to measure the efficiency of IS approximations, such as the *effective sample size* (ESS), defined by Kong, Liu and Wong (1994) as

$$ESS := \frac{N}{\mathbb{V}_{\psi}(\tilde{w}) + 1} = \frac{N\mathfrak{z}}{\pi(w)}, \tag{1.2}$$

if w is π-integrable. The effective sample size takes values in $(0, N]$. If $\psi = \pi$ then $\tilde{w} \equiv 1$ so that $ESS = N$. On the other hand, ESS decreases the more ψ and π differ.

Unfortunately, the ESS cannot be calculated analytically because the difficulty of computing integrals of the form $\pi(w)$ is one of the reasons for turning to importance sampling in the first place. We thus have to resort to estimating it by

$$ESS^N := \frac{N\mathfrak{z}^{\text{IS},N}}{\pi^{\text{IS}\star,N}(w)} = \frac{[\sum_{n=1}^N w(X^n)]^2}{\sum_{m=1}^N [w(X^m)]^2} = \frac{1}{\sum_{n=1}^N [W^n(\boldsymbol{X})]^2}. \tag{1.3}$$

This estimate of the effective sample size ranges from 1 (all self-normalised importance weights are zero except one) to N (all importance weights are identical).

Care must be taken when interpreting this estimate. For finite N, it is easy to construct examples in which the self-normalised IS estimator has a high variance despite it being very likely that all self-normalised importance weights are roughly identical. This is the case if the proposal distribution is likely to miss high-probability regions under π.

1.3 State-Space Extension and Reduction

1.3.1 Enlarging the Space

Assume again that we are interested in approximating (integrals with respect to) a measure $\gamma \in \mathcal{M}(\mathsf{X})$ by IS, using some proposal distribution $\psi \in \mathcal{M}_1(\mathsf{X})$ such that $\gamma \ll \psi$. Sometimes, we cannot evaluate the Radon–Nikodým derivative $w := \mathrm{d}\gamma/\mathrm{d}\psi$ point-wise – not even up to some unknown proportionality constant. Thus, direct IS approximations of γ and, by extension, self-normalised IS approximations of a probability measure $\pi \propto \gamma$ are not available.

However, in some cases, the intractability of the importance weights can be circumvented by approximating a measure $\bar{\gamma}$ on an extended space which admits γ as a marginal. More precisely, assume that

(1) there exists a measure $\bar{\gamma} \in \mathcal{M}(\overline{\mathsf{X}})$ on some space $\overline{\mathsf{X}} := \mathsf{X} \times \mathsf{Z}$ which admits γ as a marginal, i.e. $\gamma(A) = \bar{\gamma}(A \times \mathsf{Z})$, for any $A \in \mathcal{B}(\mathsf{X})$,

(2) we can sample from a distribution $\bar{\psi} \in \mathcal{M}_1(\overline{\mathsf{X}})$ satisfying $\bar{\gamma} \ll \bar{\psi}$,

(3) we can evaluate $\bar{w} := \mathrm{d}\bar{\gamma}/\mathrm{d}\bar{\psi}$ point-wise.

In this case, we can simply construct an IS approximation $\bar{\gamma}^{\mathrm{IS},N}$ of $\bar{\gamma}$. As shown below, the relevant marginal of $\bar{\gamma}^{\mathrm{IS},N}$ then approximates γ.

1.5 Example (Bayesian posterior, continued). *Many models are specified through extra latent (unobserved) parameters Z so that π is a marginal of the joint posterior distribution $\bar{\pi} := \Psi_{\overline{L}}(\bar{\varpi})$ associated with the joint prior $\bar{\varpi} \in \mathcal{M}_1(\overline{\mathsf{X}})$ and the 'joint' likelihood for both parameters, $\tilde{g}((x,z),y) =: \overline{L}(x,z)$. IS on the marginal space X is then often impossible. In this case, we need to devise a suitable extended proposal distribution $\bar{\psi}$.*

1.6 Example (complex proposal distributions). *Even if $w = \mathrm{d}\gamma/\mathrm{d}\psi$ can be evaluated, it can often be desirable to work on some extended space $\overline{\mathsf{X}}$ on which a more efficient proposal distribution $\bar{\psi}$ can be constructed (and the additional auxiliary variables Z included in $\bar{\psi}$ cannot be integrated out). In this case, we need to devise a suitable extended measure $\bar{\gamma}$.*

1.3.2 Importance Sampling on the Joint Space

In the setting described above, we can perform self-normalised IS on the joint space $\overline{\mathsf{X}}$. Let $\overline{X}^n = (X^n, Z^n)$ where $\overline{X}^1, \ldots, \overline{X}^N$ are IID samples distributed according to $\bar{\psi}$. Given an IS approximation

$$\bar{\gamma}^{\text{IS},N} := \sum_{n=1}^{N} \bar{w}(\overline{X}^n)\delta_{\overline{X}^n}$$

of $\bar{\gamma}$, we then immediately obtain an approximation of the marginal measure γ in the form of

$$\gamma^N := \sum_{n=1}^{N} \bar{w}(\overline{X}^n)\delta_{X^n}.$$

The approximation γ^N of the marginal measure γ is sometimes referred to as *random-weight* IS (Fearnhead, Papaspiliopoulos, Roberts & Stuart, 2010). This is because the weights are still random even after conditioning on the sample points $X^1, \ldots, X^N$ which determine the location of the point masses used in the construction of γ^N. Special cases of this include *IS-squared* (Tran, Scharth, Pitt & Kohn, 2014), for instance.

Unfortunately, performing IS on an extended space $\overline{\mathsf{X}} = \mathsf{X} \times \mathsf{Z}$ is generally less efficient than working directly on the (smaller) marginal space X. However, the fact that working on the marginal space might be impossible (as in Example 1.5) or that we might be able to construct more efficient proposal distributions by working on an extended space (as in Example 1.6) often justifies this approach.

When working on an extended space, there is often a considerable degree of freedom in constructing $\bar{\gamma}$ and $\bar{\psi}$. Modern methodological research on Monte Carlo algorithms can be viewed as almost exclusively dealing with optimising the choice of these measures even though this is not always immediately obvious as the examples in Subsection 1.3.4 show.

1.3.3 Rao–Blackwellisation

In the preceding subsections we mentioned the utility of performing IS on an extended space. However, approximating distributions on large spaces comes at a cost. Even though the *order* of the Monte Carlo convergence rate, $\mathcal{O}_{\mathbb{P}}(N^{-1/2})$, is independent of the dimension of the state space, the number of sample points, N, usually needs to grow exponentially with its dimension in order to guarantee a constant Monte Carlo error.

As many components as possible (of the target measure γ) should therefore be integrated out analytically as advocated by Trotter and Tukey (1956). By performing Monte Carlo approximations only on a smaller space, substantial variance reductions can be attained.

More precisely, let $\gamma \in \mathcal{M}(\mathsf{X})$ be some finite measure on $\mathsf{X} := \widetilde{\mathsf{X}} \times \mathsf{Z}$ and let $f \in \mathcal{L}(\mathsf{X})$ be some test function with domain X. Assume that $\gamma^{\text{IS},N}$ is an IS approximation of γ based on IID samples $X^1, \dots, X^N$ which are drawn from some suitable proposal distribution and which can be decomposed as $X^n = (\widetilde{X}^n, Z^n)$, where $\widetilde{X}^n$ takes values in $\widetilde{\mathsf{X}}$ and Z^n takes values in Z.

It is then preferable to use the *Rao–Blackwellised* estimator

$$\mu_0(f) := \mathbb{E}\big[\gamma^{\text{IS},N}(f)\big|\widetilde{X}^{1:N}\big],$$

(if we can calculate this integral) rather than $\mu_1(f) := \gamma^{\text{IS},N}(f)$ itself. This is because by Jensen's inequality, the former is dominated by the latter in the convex order, i.e. for any convex function $\phi\colon \mathbb{R} \to \mathbb{R}$ such that the following integrals are well defined,

$$\mathbb{E}[\phi(\mu_0(f))] \leq \mathbb{E}[\phi(\mu_1(f))].$$

Note that this result implies $\mathbb{V}[\mu_0(f)] \leq \mathbb{V}[\mu_1(f)]$ and the same ordering holds for the mean-square error since $\mathbb{E}[\mu_0(f)] = \mathbb{E}[\mu_1(f)]$.

This result does not generally carry over to self-normalised IS approximations. That is, $\mu_0(f)/\mu_0(\mathbb{1})$ is not necessarily dominated by $\mu_1(f)/\mu_1(\mathbb{1})$ in the convex order (Liu, 2001, p. 38).

1.7 Remark. *Note that $\mu_0(f)$ is just a standard IS estimate of the integral $\tilde{\gamma}(\mathbb{1})$, where $\tilde{\gamma}(A) := \gamma(f\, \mathbb{1}_{A\times \mathsf{Z}})$, for $A \in \mathcal{B}(\widetilde{\mathsf{X}})$, is the target measure (which is lower-dimensional than γ). This justifies using the term 'marginalisation' as a synonym for 'Rao–Blackwellisation' in the next section.*

1.3.4 Examples

A main thread throughout this work is that most seemingly-complicated Monte Carlo algorithms can be viewed as special cases of IS on a suitably extended space (and thus as instances of *the* Monte Carlo method). For example, it is well known that *sequential importance sampling* (the idea of which dates at least as far back as (Hammersley & Morton, 1954; M. N. Rosenbluth & Rosenbluth, 1955) and special cases of it such as *annealed importance sampling* (Jarzynski, 1997b, 1997a; Neal, 2001) may be viewed as standard IS.

In this subsection, we show, by example, that this also applies to many other algorithms. First, as shown in Example 1.8, *rejection sampling* (also referred to as *accept–reject method*) first described in Kahn (1949), may be viewed as a special case of (self-normalised) IS on an extended space. This was pointed out in Y. Chen (2005), for instance.

1.8 Example (rejection sampling). *Rejection sampling is usually viewed as generating a random number of IID samples from some distribution* $\pi \in \mathcal{M}_1(\mathsf{X})$*, as follows.*

(1) Propose IID samples $X^1, \dots, X^N$ *from some distribution* $\psi \in \mathcal{M}_1(\mathsf{X})$ *satisfying* $\pi \ll \psi$*.*

(2) Assume there exists $\mathfrak{z} > 0$ *such that* $w := \mathfrak{z}\mathrm{d}\pi/\mathrm{d}\psi \leq 1$ *and such that* w *can be evaluated.*

(3) For $n \in \mathbb{N}_N := \{n \in \mathbb{N} \mid n \leq N\}$*, independently 'accept'* X^n *with probability* $w(X^n)$ *and set* $K := \{n \in \mathbb{N}_N \mid X^n \text{ is 'accepted'}\}$*. Then marginally,* $(X^n)_{n \in K}$ *is an IID sample from* π*.*

Rejection sampling thus entails generating IID samples $\bar{X}^1, \dots, \bar{X}^N$ *from an extended proposal distribution*

$$\bar{\psi} := \psi \otimes \mathrm{Unif}_{[0,1]} \in \mathcal{M}_1(\bar{\mathsf{X}}),$$

where $\bar{\mathsf{X}} := \mathsf{X} \times [0, 1]$*. These proposals are then used to form an IS approximation* $\bar{\gamma}^{\mathrm{IS},N}$ *of the extended measure*

$$\bar{\gamma} := \psi \otimes L \in \mathcal{M}(\bar{\mathsf{X}}),$$

where $L(x, \mathrm{d}z) := \mathbb{1}_{[0,w(x)]}(z)\mathrm{d}z$*. The importance weights are therefore defined by* $\bar{w}(\bar{x}) := [\mathrm{d}\bar{\gamma}/\mathrm{d}\bar{\psi}](x, z) = \mathbb{1}_{[0,w(x)]}(z)$*. The resulting marginal* *self-normalised* IS *approximation of* π *can then easily be seen to be*

$$\pi^N := (\#K)^{-1} \sum_{n \in K} \delta_{X^n},$$

where $\#K$ *denotes the cardinality of the set* K*. In particular,* $\bar{\gamma}^{\textsc{is},N}(\mathbb{1})$ *is an unbiased estimate of the marginal acceptance probability,* $\mathfrak{z}$*.*

Finally, a standard IS *approximation of* $\gamma = \mathfrak{z}\pi$ *on the marginal space* X *(with proposal distribution* ψ*) can be viewed as a Rao–Blackwellisation of the rejection-sampling approximation. That is,*

$$\gamma^{\textsc{is},N}(f) = \frac{1}{N} \sum_{n=1}^{N} w f(X^n) = \mathbb{E}\bigl[\bar{\gamma}^{\textsc{is},N}(f \otimes \mathbb{1}_{[0,1]}) \big| \boldsymbol{X}\bigr].$$

Note that we are fixing the number of proposed *samples,* N*, in the rejection-sampling scheme. A Rao–Blackwellisation in the case where rejection sampling is performed until a certain number of* accepted *samples has been obtained was developed in Casella and Robert (1996).*

Many other algorithms that seem to be generalisations of IS, at a first glance, can actually also be viewed as standard IS on an extended space, as shown in Examples 1.9 and 1.10.

1.9 Example (generalised importance sampling). *Let* $\bar{\varPsi} \in \mathcal{K}_1(\mathsf{X}, \mathsf{X})$ *be some* γ*-invariant stochastic kernel, i.e. such that* $\gamma\bar{\varPsi} = \gamma$*. As shown in MacEachern, Clyde and Liu (1999, Theorem 6.1) it is possible to apply such a kernel to the weighted sample used to construct an* IS *approximation of* γ *without having to adjust the weights.*

Even though this procedure is sometimes referred to as 'generalised' importance sampling (e.g. Robert & Casella, 2004, Section 14.2), as pointed out in Doucet and Johansen (2011) (see also Del Moral, Doucet & Jasra, 2006b), it may be viewed as standard importance sampling on the extended space $\bar{\mathsf{X}} := \mathsf{X}^2$ *(i.e.* $\mathsf{Z} = \mathsf{X}$ *in the notation of this section), with extended proposal distribution* $\bar{\psi} := \psi \otimes \varPsi$ *and extended target measure* $\bar{\gamma} := \gamma \otimes \varPi$*, where*

$$\varPi(x', \mathrm{d}x) := \frac{\mathrm{d}\varPsi(x, \cdot)}{\mathrm{d}\gamma\varPsi}(x')\gamma(\mathrm{d}x) = \frac{\mathrm{d}\varPsi(x, \cdot)}{\mathrm{d}\gamma}(x')\gamma(\mathrm{d}x)$$

represents the time-reversal kernel *of* $\varPsi$ *associated with* γ*. Indeed writing* $\bar{X} = (X, X')$*, the weights* $\bar{w}(\bar{x}) = [\mathrm{d}\bar{\gamma}/\mathrm{d}\bar{\psi}](\bar{x}) = [\mathrm{d}\gamma/\mathrm{d}\psi](x)$ *do not depend on the second component.*

1.10 Example (dynamic weighting). *The* dynamically weighted Monte Carlo-*framework (Wong & Liang, 1997; Liu, Liang & Wong, 2001; Liang, 2002) designs an extended measure*

$$\tilde{\gamma}(\mathrm{d}x \times \mathrm{d}v) := vg(\mathrm{d}x, \mathrm{d}v),$$

on $\widetilde{X} := \mathsf{X} \times [0, \infty)$. *Here,* $g \in \mathcal{P}(\widetilde{\mathsf{X}})$, $\widetilde{\mathsf{X}} := \mathsf{X} \times [0, \infty)$, *is said to be* correctly weighted with respect to γ *if* $\tilde{\gamma}$ *admits* γ *as a marginal, i.e. if* $\gamma(A) = \tilde{\gamma}(A \times [0, \infty))$ *for any* $A \in \mathcal{B}(\mathsf{X})$.

In this case, an IID *sample*

$$\widetilde{X}^1, \ldots, \widetilde{X}^N \sim \tilde{\psi} := g,$$

where $\widetilde{X}^n = (X^n, V^n)$, *can be used to approximate* γ *by standard* IS.

The method is called 'dynamic' importance sampling because the n*th importance weight,* $\widetilde{w}(\widetilde{X}^n) = V^n$, *is not necessarily deterministic given* X^n. *It is also referred to as 'generalised' importance sampling in Liu (2001, pp. 36–37), Liang (2002) because taking*

$$g(\mathrm{d}x \times \mathrm{d}v) := \psi(\mathrm{d}x)\delta_{w(x)}(\mathrm{d}v),$$

where $w := \mathrm{d}\gamma/\mathrm{d}\psi$, *leads back to a direct* IS *approximation of the marginal* γ *using* ψ *as a proposal distribution.*

However, this approach is clearly no more than standard IS *on an extended space. Indeed, let* $\bar{\gamma} \in \mathcal{M}(\overline{\mathsf{X}})$ *be some other extended measure on a space* $\overline{\mathsf{X}} = \mathsf{X} \times \mathsf{Z}$ *such that (1)* $\bar{\gamma}$ *admits* γ *as a marginal, (2)* $\bar{\gamma}$ *has a density* $\overline{w}$ *with respect to some probability measure* $\bar{\psi} \in \mathcal{M}_1(\overline{\mathsf{X}})$. *Using the proposal distribution* $\bar{\psi}$, *we can then construct an* IS *approximation*

$$\bar{\gamma}^{\mathrm{IS},N} := \frac{1}{N}\sum_{n=1}^{N} \overline{w}(\bar{X}^n)\delta_{\bar{X}^n}$$

of $\bar{\gamma}$ *and hence obtain an approximation* γ^N *of the marginal* γ.

Note, however, that Z *and thus* $\overline{\mathsf{X}} = \mathsf{X} \times \mathsf{Z}$ *is often a high-dimensional space which can render the preceding* IS *scheme inefficient. Since we are only interested in the marginal measure* γ, *the key insight here is that we can turn this* IS *scheme into an* IS *scheme on the potentially lower-dimensional space* $\widetilde{\mathsf{X}} = \mathsf{X} \times \mathsf{V}$, *with* $\mathsf{V} := [0, \infty)$, *with extended target* $\tilde{\gamma}$ *and proposal* $\tilde{\psi} = g$ *as follows.*

Define $\phi := \mathrm{id}_{\mathsf{X}} \otimes \bar{w}$, $\tilde{\psi} = g := \bar{\psi} \circ \phi^{-1}$, then we can see that the marginal approximations of γ based on performing IS using the pair $(\tilde{\gamma}, \tilde{\psi})$ and using the pair $(\bar{\gamma}, \bar{\psi})$ coincide, i.e.

$$\gamma^N = \frac{1}{N} \sum_{n=1}^{N} \bar{w}(\bar{X}^n) \delta_{X^n} = \frac{1}{N} \sum_{n=1}^{N} \tilde{w}(\tilde{X}^n) \delta_{X^n} = \frac{1}{N} \sum_{n=1}^{N} V^n \delta_{X^n},$$

where $\bar{X}^n = (X^n, Z^n)$.

The transformation ϕ used in Example 1.10 to interpret the marginal of an IS approximation of a measure $\bar{\gamma}$ on a potentially high-dimensional space $\bar{\mathsf{X}} \times \mathsf{Z}$ as the marginal of an IS approximation of a measure $\tilde{\gamma}$ on the potentially lower-dimensional space $\tilde{\mathsf{X}}$ is also the main justification of *pseudo-marginal* Monte Carlo approaches (Andrieu & Roberts, 2009) the general idea of which is described in Subsection 1.4.3.

1.4 Marginalised One-Sample Importance Sampling

1.4.1 Extended Target Measure

In this section, we present a generic extended measure which admits the target measure, γ, as a marginal. It is based on the IS framework introduced in Andrieu and Roberts (2009), Andrieu, Doucet and Holenstein (2010) which was also extensively analysed in Lee (2011).

As shown in the next two chapters, virtually all known Monte Carlo schemes, e.g. *Markov chain Monte Carlo* (MCMC) methods, *sequential Monte Carlo* (SMC) methods, and even generalisations of the latter such as *Divide-&-Conquer SMC* (Lindsten, Johansen et al., 2014), can be regarded as Rao–Blackwellised IS approximations – based on a single sample – targeting this measure (or as self-normalised versions thereof).

A repeated, hierarchical application of this framework justifies employing one such Monte Carlo scheme into another, e.g. using standard IS within standard IS (Tran et al., 2014), SMC within MCMC (Andrieu et al., 2010), MCMC within SMC (Gilks & Berzuini, 2001; Del Moral et al., 2006b), or SMC within SMC (Johansen, Whiteley & Doucet, 2012; Chopin, Jacob & Papaspiliopoulos, 2013; Vergé, Dubarry, Del Moral & Moulines, 2013).

As before, we want to approximate an integral $\gamma(f)$, where $\gamma \in \mathcal{M}(\mathsf{X})$ is some finite measure and $f \in \mathsf{F}$ is some γ-integrable test function.

To that end, we define an extended target measure $\bar{\gamma} \in \mathcal{M}(\bar{\mathbf{X}})$ and an extended proposal distribution $\bar{\psi} \in \mathcal{M}_1(\bar{\mathbf{X}})$ such that $\bar{w} := \mathrm{d}\bar{\gamma}/\mathrm{d}\bar{\psi}$ exists and can be evaluated (point-wise). Here, $\bar{\mathbf{X}} := \mathsf{X} \times \bar{\mathbf{Z}}$ is an extended space such that if $\bar{\boldsymbol{X}} = (X, \bar{\boldsymbol{Z}}) \sim \bar{\psi}$, then $\bar{\boldsymbol{Z}}$ can be decomposed as $\bar{\boldsymbol{Z}} = (K, \boldsymbol{Z}) = (K, \boldsymbol{X}, Y)$, where

- K is some discrete index taking values in a finite space K,
- $\boldsymbol{X}$ represents the elements of a 'pool' of candidates for each of the components of the vector $X \sim \pi := \gamma/\gamma(\mathbb{1})$ such that the set of candidate components indexed by K, denoted X^K, takes values in X (see Remark 1.11 below for more details),
- Y is some set of other auxiliary variables taking values in a space Y.

1.11 Remark. *The definition of $\bar{\boldsymbol{X}} = (X, K, \boldsymbol{Z})$ is left deliberately vague so that the framework covers a wide range of Monte Carlo schemes.*

(1) To simplify the notation and without loss of generality, we restrict our exposition in this section to the case: $\mathsf{K} = \mathbb{N}_N$ and $\mathbf{X} = \mathsf{X}^N$, for some $N \in \mathbb{N} \setminus \{1\}$. In other words, X^K is the Kth element out of a pool of N candidates, $\boldsymbol{X} = X^{1:N}$, for X.

(2) More generally, let $X = X_{1:t}$ have t components for each of which we have a pool of N' candidates. We could then consider an index vector $K_{1:t}$ taking values in $(\mathbb{N}_{N'})^t$ such that $(X_1^{K_1}, \dots, X_t^{K_t})$ takes values in X. This is the case in SMC algorithms outlined in the next chapter. However, by applying a suitable reparametrisation and by introducing some additional conditionally degenerate copies of the components in the pool, we can always reduce such a seemingly more complex setting to the case in which we only have a one-dimensional index K taking values in $\mathbb{N}_N$, here $N = (N')^t$.

(3) We could also consider a random number of candidates, e.g. SMC algorithms with random numbers of particles (Crisan, Del Moral & Lyons, 1998). However, we refrain from doing so in order to work on a product space $\bar{\mathbf{X}} = \mathsf{X} \times \mathsf{K} \times \mathbf{Z}$ which greatly simplifies the notation.

We are now ready to define the extended target measure $\bar{\gamma}$. Take some stochastic kernel $\bar{\Pi} \in \mathcal{K}_1(\mathsf{X}, \bar{\mathbf{Z}})$. The distribution $\bar{\Pi}(x, \cdot)$ can then be used to define the full conditional distribution of $\bar{\boldsymbol{Z}}$ under the extended target distribution $\bar{\pi} := \bar{\gamma}/\bar{\gamma}(\mathbb{1})$, where we call

$$\bar{\gamma}(\mathrm{d}\bar{\boldsymbol{x}}) := \gamma(\mathrm{d}x)\bar{\Pi}(x, \mathrm{d}\bar{\boldsymbol{z}})$$

the *extended target measure.* Clearly, $\bar{\gamma}$ admits γ as a marginal. In addition to admitting γ as a marginal, we make the following minimal assumption on this extended target measure.

1.12 Assumption. *The stochastic kernel $\bar{\Pi} \in \mathcal{K}_1(\mathsf{X}, \bar{\mathbf{Z}})$ is such that*

$$\bar{\boldsymbol{X}} \sim \bar{\pi} \quad \Rightarrow \quad X^K = X, \text{ almost everywhere.}$$

Similarly, define an *extended proposal distribution*

$$\bar{\psi}(\mathrm{d}\bar{\boldsymbol{x}}) := \psi(\mathrm{d}\boldsymbol{z})\xi(\boldsymbol{z}, \mathrm{d}k)\delta_{x^k}(\mathrm{d}x),$$

where the probability measure $\psi \in \mathcal{M}_1(\mathbf{Z})$ and the stochastic kernel $\xi \in \mathcal{K}_1(\mathbf{Z}, \mathsf{K})$ are chosen such that $\bar{\gamma} \ll \bar{\psi}$.

1.4.2 Generic Estimator

Let $\bar{w} := \mathrm{d}\bar{\pi}/\mathrm{d}\bar{\psi}$. Using $\bar{\boldsymbol{X}} = (X, \bar{\boldsymbol{Z}}) = (X, K, \boldsymbol{Z})$ drawn from $\bar{\psi}$, we may approximate the extended measure $\bar{\gamma}$ by $\bar{\gamma}^{\textsc{is},1} := \bar{w}(\bar{\boldsymbol{X}})\delta_{\bar{\boldsymbol{X}}}$. This represents an IS approximation of $\bar{\gamma}$ based on a single sample. Define the kth 'weight'

$$w^k(\boldsymbol{Z}) := \mathbb{E}\big[\bar{w}(\bar{\boldsymbol{X}})\,\mathbb{1}_{\{k\}}(K)\big|\boldsymbol{Z}\big],$$

then we may analytically integrate out ('marginalise out') some subvector of $\bar{\boldsymbol{X}}$ which includes (X, K) – for simplicity, we only integrate out (X, K), here – to yield the following *marginalised one-sample importance sampling* (MOSIS) approximation of the marginal measure γ,

$$\begin{aligned}\gamma^{\textsc{mosis},N}(A) &:= \mathbb{E}\big[\bar{\gamma}^{\textsc{is},1}(A \times \bar{\mathbf{Z}})\big|\boldsymbol{Z}\big]\\ &= \sum_{k \in \mathsf{K}} w^k(\boldsymbol{Z})\delta_{X^k}(A),\end{aligned}$$

for any $A \in \mathscr{B}(\mathsf{X})$. If desired, a self-normalised IS approximation of π may then be constructed as

$$\pi^{\text{MOSIS}\star,N} := \frac{\gamma^{\text{MOSIS},N}}{\mathfrak{z}^{\text{MOSIS},N}} = \sum_{k \in \mathsf{K}} W^k(\boldsymbol{Z})\delta_{X^k},$$

where $W^k(\boldsymbol{Z}) := w^k(\boldsymbol{Z})/\mathfrak{z}^{\text{MOSIS},N}$ will be called the kth self-normalised weight and $\mathfrak{z}^{\text{MOSIS},N} := \gamma^{\text{MOSIS},N}(\mathbb{1})$ is a standard IS estimate of the normalising constant, $\mathfrak{z}$, and is therefore clearly unbiased.

The estimate of the normalising constant is a key quantity and its variance is strongly connected with the efficiency of the MOSIS scheme due to the following result.

1.13 Proposition. *Let $\xi(\boldsymbol{z}, \{n\}) = W^n(\boldsymbol{z})$, for any $(n, \boldsymbol{z}) \in \mathsf{K} \times \mathbf{Z}$, then*

$$\bar{w}(\bar{\boldsymbol{x}}) = \mathfrak{z}^{\text{MOSIS},N}, \quad \textit{for any } \bar{\boldsymbol{x}} \in \bar{\mathbf{X}}.$$

Proof. This follows immediately from the definition of $w^n(\boldsymbol{z})$. □

We stress that while this framework ensures unbiasedness, it does not necessarily ensure consistency because for any $N \in \mathbb{N}$, we are still performing IS with only one sample point. The following trivial counter example demonstrates this problem.

1.14 Example. *For some distribution $q \in \mathcal{M}_1(\mathsf{X})$ with $\gamma \ll q$, set*

$$\bar{\Pi}(x, \mathrm{d}\bar{z}) := \mathrm{Unif}_{\mathsf{K}}(\mathrm{d}k)\delta_x(\mathrm{d}x^k)\prod_{n \in \mathsf{K}\setminus\{k\}} \delta_{x^k}(\mathrm{d}x^n),$$

$$\bar{\psi}(\mathrm{d}\boldsymbol{x}) := q(\mathrm{d}x^1)\prod_{n=2}^{N} \delta_{x^1}(\mathrm{d}x^n),$$

and $\xi(\boldsymbol{z}, \mathrm{d}k) := \mathrm{Unif}_{\mathsf{K}}(\mathrm{d}k)$. Then, writing $w := \mathrm{d}\gamma/\mathrm{d}q$, the estimator $\gamma^{\text{MOSIS},N}(f)$ is almost surely equal to $wf(X^1)$, for any $N \in \mathbb{N}$. It is therefore unbiased but clearly not consistent.

Effective Sample Size. Finally, we may use this framework to obtain an approximation of the ESS, which, in this case, is defined according to Equation 1.2 as $ESS = N\mathfrak{z}^2/\bar{\gamma}(\bar{w})$. An approximation of ESS is thus

$$ESS^N = \frac{N(\mathbb{E}[\bar{\gamma}^{\text{IS},1}(\mathbb{1})|\boldsymbol{Z}])^2}{\mathbb{E}[\bar{\gamma}^{\text{IS},1}(\bar{w})|\boldsymbol{Z}]} = \frac{N[\sum_{k \in \mathsf{K}} w^k(\boldsymbol{Z})]^2}{\sum_{k \in \mathsf{K}}[w^k(\boldsymbol{Z})]^2/\xi(\boldsymbol{Z}, \{k\})}.$$

If $\xi(z, \cdot) = \mathrm{Unif}_{\mathbb{N}_N}$, for any $z \in \mathbf{Z}$, then this reduces to

$$ESS^N = \frac{1}{\sum_{k \in \mathrm{K}} [W^k(\boldsymbol{Z})]^2}.$$

1.4.3 Pseudo-Marginal Interpretation

Assume now that the target measure, extended target measure and extended proposal distribution depend on some parameter $\theta \in \Theta$. We indicate this by writing them as suitable kernels, i.e. by writing $\gamma(\theta, \cdot)$, $\bar{\gamma}(\theta, \cdot)$, $\bar{\psi}(\theta, \cdot)$ and $\mathrm{d}\bar{\gamma}(\theta, \cdot)/\mathrm{d}\bar{\psi}(\theta, \cdot) = \bar{w}^\theta$. Furthermore, let $\varpi \in \mathcal{M}(\Theta)$ be some finite measure.

Suppose that we want to approximate the following 'marginal' measure under the measure $\varpi \otimes \gamma$, defined by

$$\gamma^\star(A) := \varpi(\mathbb{1}_A \gamma(\cdot, \mathbb{1})) = [\varpi \otimes \gamma](A \times \mathrm{X}),$$

for all $A \in \mathcal{B}(\Theta)$. Unfortunately, the function $\gamma(\cdot, \mathbb{1})$ is often intractable. We must therefore resort to approximating it via MOSIS (often within some other Monte Carlo scheme). More precisely, we use some other Monte Carlo scheme to target the extended measure $\varpi \otimes \bar{\gamma}$ (which admits $\gamma^\star$ as a marginal) or a normalised version thereof.

1.15 Example (Bayesian posterior, continued). *If π is the (marginal) posterior distribution of some parameter Θ, then $L(\theta) = \gamma(\theta, \mathbb{1})$ is its (marginal) likelihood. This is often intractable if the model is specified through additional latent variables, X, which need to be integrated out to obtain $\gamma(\theta, \mathbb{1})$. However, in order to approximate $\pi = \Psi_L(\varpi) = L\varpi/\varpi(L)$ it usually suffices to approximate the (unnormalised) measure $L\varpi = \gamma^\star$. This approximation suffices for a self-normalised IS approximation of π. Or, within MCMC schemes, the normalising constant $\varpi(L) = \gamma^\star(\mathbb{1})$ cancels out in the 'acceptance probabilities' (see Chapter 3).*

Write $\widetilde{T}(\theta, \cdot) := \bar{\psi}(\theta, \cdot) \circ (\bar{w}^\theta)^{-1}$, where $(\bar{w}^\theta)^{-1}$ denotes the preimage under $\bar{w}^\theta$, then, for all $A \in \mathcal{B}(\Theta)$, we have the identity

$$\begin{aligned}\gamma^\star(A) &= [\varpi \otimes \bar{\gamma}](A \times \overline{\mathbf{X}})\\ &= [\varpi \otimes \widetilde{T}](\mathbb{1}_A \otimes \mathrm{id}_{[0,\infty)})\end{aligned}$$

$$= \int_{A\times \mathsf{V}} \varpi(\mathrm{d}\theta)\gamma(\theta,\mathbb{1})\widetilde{T}(\theta,\mathrm{d}v)\frac{v}{\gamma(\theta,\mathbb{1})}$$

$$= \int_{A} \gamma^\star(\mathrm{d}\theta)\,\mathbb{E}\left[\frac{V}{\gamma(\theta,\mathbb{1})}\right],$$

where $V \sim \widetilde{T}(\theta,\cdot)$ and $\mathsf{V} := [0,\infty)$. Note that for any $\theta \in \Theta$, the random variable V is an (unbiased) IS estimate of $\gamma(\theta,\mathbb{1})$ and hence

$$\mathbb{E}\left[\frac{V}{\gamma(\theta,\mathbb{1})}\right] = 1.$$

We may thus use some other Monte Carlo scheme to approximate $\varpi \otimes \bar{\gamma}$ (or its normalised version), based on the proposal distribution $q \otimes \bar{\psi}$ for some $q \in \mathcal{M}_1(\Theta)$ satisfying $\gamma^\star \ll q$. Ideally, we would like work on the marginal space and approximate the marginal $\gamma^\star$ using samples from q. However, this is impossible here because the 'marginal' Radon–Nikodým derivative $[\mathrm{d}\gamma^\star/\mathrm{d}q](\theta)$ involves $\gamma(\theta,\mathbb{1})$ and is therefore intractable.

Instead, we work on the extended space $\Theta \times \bar{\mathbf{X}}$ and target $\varpi \otimes \bar{\gamma}$. Then any realisation $(\theta, \bar{\boldsymbol{x}})$ of $(\varTheta, \bar{\boldsymbol{X}}) \sim q \otimes \bar{\psi}$ implies a corresponding realisation $v = \bar{w}^\theta(\bar{\boldsymbol{x}})$ of $V \sim \widetilde{T}(\theta,\cdot)$. As a result,

$$\frac{\mathrm{d}\varpi}{\mathrm{d}q}(\theta)\bar{w}^\theta(\bar{\boldsymbol{x}}) = \frac{\mathrm{d}\gamma^\star}{\mathrm{d}q}(\theta)\frac{v}{\gamma(\theta,\mathbb{1})}$$

can be viewed as a 'noisy' but often tractable (because $\bar{w}^\theta$ is tractable) evaluation of the intractable Radon–Nikodým derivative $[\mathrm{d}\gamma^\star/\mathrm{d}q](\theta)$.

The interpretation as a noisy evaluation of an intractable marginal density has led to such constructions being termed *pseudo-marginal* methods. They were introduced by Beaumont (2003), Andrieu and Roberts (2009), extended by Andrieu et al. (2010) and are usually applied within MCMC methods. However, the pseudo-marginal target measure $\varpi \otimes \bar{\gamma}$ can be approximated by other types of MOSIS schemes, too. For instance, some pseudo-marginal *SMC* algorithms are mentioned in Subsection 2.3.5.

As mentioned in Example 1.10, an approximation of the marginal measure $\gamma^\star$ obtained from performing IS on the potentially high-dimensional space $\Theta \times \bar{\mathbf{X}}$ (with proposal distribution $q \otimes \bar{\psi}$) can be interpreted as an approximation obtained from performing IS on the potentially lower-dimensional space $\Theta \times \mathsf{V}$.

1.4.4 Application to Standard Importance Sampling

By construction, MOSIS is obviously a special case of (Rao–Blackwellised) IS on the extended space $\bar{\mathsf{X}}$. However, to demonstrate the power of this approach, this subsection shows that MOSIS may also be viewed as a generalisation of IS on the original space, X.

Let $N \in \mathbb{N}$, $\mathsf{K} = \mathbb{N}_N$, $\mathbf{X} = \mathsf{X}^N$, and write $\bar{\boldsymbol{Z}} = (K, \boldsymbol{X})$ with the pool of candidates $\boldsymbol{X} = X^{1:N}$. That is, we do not use further auxiliary variables, Y, here. Take

$$\bar{\Pi}(x, \mathrm{d}\bar{z}) := \Lambda(\mathrm{d}k)\delta_x(\mathrm{d}x^k)q^{\otimes(N-1)}(\mathrm{d}\boldsymbol{x}^{-k}), \tag{1.4}$$

where $\boldsymbol{X}^{-n} := (X^{1:n-1}, X^{n+1:N})$ denotes the pool of candidates from which the nth element has been removed, and set $\Lambda := \mathrm{Unif}_{\mathsf{K}}$. For $\gamma \ll q \in \mathcal{M}_1(\mathsf{X})$, define the extended proposal distribution via $\psi := q^{\otimes N}$ and $\xi(\boldsymbol{z}, \cdot) := \mathrm{Unif}_{\mathsf{K}}$, for any $\boldsymbol{z} \in \mathbf{Z}$.

The importance weight is then given by $\bar{w}(\bar{\boldsymbol{x}}) = [\mathrm{d}\gamma/\mathrm{d}q](x) =: w(x)$. Thus $\bar{\gamma}^{\text{IS},1} := \bar{w}(\bar{\boldsymbol{X}})\delta_{\bar{\boldsymbol{X}}}$ represents an IS approximation of $\bar{\gamma}$ based on a single sample point $\bar{\boldsymbol{X}} \sim \bar{\psi}$. However, we are only interested in approximating the marginal measure γ. Noting that $w^k(\boldsymbol{z}) = w(x^k)/N$, we may analytically integrate out (X, K) to obtain a Rao–Blackwellised estimator, for any $A \in \mathcal{B}(\mathsf{X})$ defined by

$$\begin{aligned}
\gamma^{\text{MOSIS},N}(A) &= \mathbb{E}\big[\bar{\gamma}^{\text{IS},1}(A \times \mathsf{Z})\big|\boldsymbol{Z}\big] \\
&= \sum_{k \in \mathsf{K}} w^k(\boldsymbol{Z})\delta_{X^k}(A) \\
&= \frac{1}{N}\sum_{n=1}^{N} w(X^n)\delta_{X^n}(A) \\
&= \gamma^{\text{IS},N}(A).
\end{aligned}$$

Hence, IS can be viewed as a special case of MOSIS. In particular, the approximation of the ESS reduces to the expression in Equation 1.3,

$$ESS^N = \frac{[\sum_{k\in\mathsf{K}} w^k(\boldsymbol{Z})]^2}{\sum_{k\in\mathsf{K}}[w^k(\boldsymbol{Z})]^2} = \frac{[\sum_{n=1}^{N} w(X^n)]^2}{\sum_{m=1}^{N}[w(X^m)]^2}.$$

1.5 Summary

Of course, the MOSIS-framework presented in this chapter is not necessary for justifying a standard IS approximation as that given above.

Its power resides in the fact that it still guarantees unbiased estimates of $\gamma(f)$ when 'generalising' standard IS to settings in which

(1) the candidates $X^1, \dots, X^N$ are not necessarily independently or identically proposed, i.e. $\psi \neq q^{\otimes N}$ for any distribution q,

(2) we can evaluate $\bar{w} = \mathrm{d}\bar{\gamma}/\mathrm{d}\bar{\psi}$ but not necessarily the Radon–Nikodým derivative of γ with respect to a suitable marginal under the joint proposal distribution ψ.

1.16 Remark. *Equation* 1.4 *shows that in the example considered in this subsection, the extended target measure, $\bar{\gamma}$, is constructed by extending the actual target measure, γ, using the full conditional distribution of the $N-1$ candidates $\boldsymbol{X}^{-k}$ under the joint proposal distribution ψ. The importance weights therefore force us to evaluate (densities with respect to) the* marginal *distribution of the kth candidate under ψ. This makes it difficult to use complex joint proposal distributions, e.g. joint proposal distributions under which the candidates are dependent.*

The major innovation due to Andrieu and Roberts (2009) – which has received surprisingly little attention and is rarely exploited outside of particle MCMC methods – is the realisation that the extended target measure can also be constructed by extending γ differently. This permits a much more flexible choice of joint proposal distribution. More precisely, following Andrieu and Roberts (2009), we can construct $\bar{\gamma}$ in such a way that the importance weights only require us to evaluate (densities with respect to) the conditional *distribution of the kth candidate under ψ.*

Combined with the introduction of the auxiliary variable Y in Andrieu et al. (2010), this realisation turns the MOSIS approach into an extremely powerful instance of the Monte Carlo method.

Other instances of the MOSIS framework – more complicated than the standard IS approximation described in the previous subsection – will be described in the next two chapters. In particular, we show that SMC (Chapter 2) and MCMC (Chapter 3) methods may be viewed as special cases of MOSIS. One possible way of viewing the relationship between IS, MOSIS and their special cases is outlined in Figure 1.1.

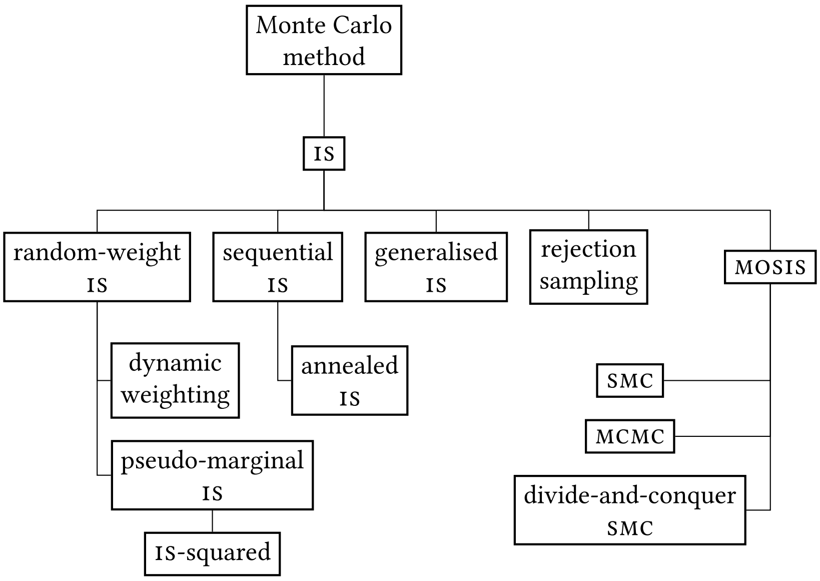

Figure 1.1 Relationship between various instances of the Monte Carlo method mentioned in Chapters 1, 2, and 3.

2 Sequential Monte Carlo Methods

2.1 Introduction

2.1.1 Motivation

In this chapter, we describe sequential Monte Carlo methods. Section 2.1 outlines a generic sequential Monte Carlo algorithm which admits essentially all known sequential Monte Carlo algorithms as a special case. In Section 2.2, we show that this generic algorithm can itself be viewed as a special case of the marginalised one-sample importance sampling framework introduced in Chapter 1 and thus as importance sampling. This was already established in Andrieu et al. (2010) and extended to non-exchangeable resampling schemes in Lee, Murray and Johansen (in prep.). We extend the latter construction to also allow for biased resampling schemes. In Section 2.3, we interpret a number of sequential Monte Carlo algorithms, such as the discrete particle filter, as special cases of this framework. In Subsection 2.3.3 we generalise and improve existing schemes for re-using all particles to approximate integrals. Finally, in Section 2.4, we show that forward filtering–backward smoothing, too, is a special case of importance sampling.

Sequential Monte Carlo (SMC) methods are a class of Monte Carlo schemes suitable for approximating a sequence of related measures. Dating at least as far back as Stewart and McCarty Jr (1992), Gordon, Salmond and Smith (1993), Del Moral (1995), SMC methods were originally developed to approximate the optimal filtering problem in discrete-time target tracking applications, e.g. in non-linear or non-Gaussian *(general state-space) hidden Markov models* (HMMs) (sometimes called *state-space models*). In this particular setting, they are often called 'particle filters'.

Today, it is well known – and has been formalised in Del Moral et al. (2006b) – that SMC methods are more widely applicable. They have been used for estimating (ratios of) normalising constants (Neal, 2001), for inference in 'static' models (Chopin, 2002), for rare-event estimation

(Cérou, LeGland, Del Moral & Lezaud, 2005), for inference in continuous-time models (Fearnhead et al., 2010), for optimisation (Johansen, Doucet & Davy, 2008), for model selection (Peters, 2005; Jasra, Doucet, Stephens & Holmes, 2008; Yan Zhou, Johansen & Aston, 2013), and for approximately solving inverse problems (Kantas, Beskos & Jasra, 2014).

A tutorial-style introduction to the area can be found in Doucet and Johansen (2011). A book-length treatment of their application to HMMs can be found in Cappé et al. (2005). A comprehensive theoretical framework was developed in the monographs Del Moral (2004, 2013) – see also, Del Moral and Doucet (2014) for a gentle introduction to this framework in the case of finite state spaces.

Various kinds of SMC algorithms have been developed, each tailored to individual problems. We detail some of these in Section 2.3. Essentially any such algorithm can be viewed as special cases of the generic SMC algorithm presented in the next subsection. In turn, as shown in Section 2.2, the generic SMC algorithm can itself be viewed as no more than a special case of the *marginalised one-sample importance sampling* (MOSIS) scheme described in Chapter 1.

The basic idea of SMC methods is as follows. Assume that we want to approximate (integrals with respect to) a family of positive finite measures $(\tilde{\gamma}_t)_{t \in \mathsf{T}}$; usually, $\mathsf{T} = \mathbb{N}_T$, for $T \in \mathbb{N}$ or $\mathsf{T} = \mathbb{N}$. In this case, we can define a sequence of extended measures $(\gamma_t)_{t \in \mathsf{T}}$, where $\gamma_t \in \mathcal{M}(\mathsf{X}_{1:t}^{\times})$, such that γ_t admits $\tilde{\gamma}_t$ as a marginal. As described in Section 1.3 in the previous chapter, working on such a product space $\mathsf{X}_{1:t}^{\times} := \bigtimes_{s=1}^{t} \mathsf{X}_s$, which typically includes all random variables generated over the course of the algorithm, is usually necessary to circumvent the calculation of intractable integrals related to the importance weights.

At Step $t-1$, the algorithm approximates γ_{t-1}, and thus $\tilde{\gamma}_{t-1}$, by weighted samples. To obtain an approximation of γ_t, and thus $\tilde{\gamma}_t$, SMC methods can be thought of as extending and re-weighting an existing collection of sample points, often called ‘particles’.

2.1 Remark. *To reduce the notational burden, we assume here that the number of particles generated at Step t, $N_t \in \mathsf{T}$, is deterministic. However, all developments in this chapter still hold if it was made a (non-degenerate) random variable as, for instance, in Crisan et al. (1998), Jasra, Lee, Yau and Zhang (2013), Lee, Andrieu and Doucet (in prep.).*

2.1.2 Particles and Parent Indices

Let $\boldsymbol{X}_t := X_t^{1:N_t}$ denote the collection of N_t particles which takes values in $\mathbf{X}_t := \mathsf{X}_t^{N_t}$. These are generated at Step t of an SMC algorithm targeting a measure $\gamma_t \in \mathcal{M}(\mathsf{X}_{1:t}^{\times})$, where $\mathsf{X}_{1:t}^{\times} := \bigtimes_{s=1}^{t} \mathsf{X}_s$. The nth particle at Step s, X_t^n, will be considered as the offspring of the A_{t-1}^nth particle generated at Step $t-1$. We therefore call A_{t-1}^n the nth parent index sampled at Step t. For simplicity, we collect the parent indices in vectors $\boldsymbol{A}_{t-1} := A_{t-1}^{1:N_t}$ taking values in $\mathbf{A}_{t-1} := \mathsf{K}_{t-1}^{N_t}$, where $\mathsf{K}_t := \mathbb{N}_{N_t}$.

To simplify the notation, we collect all random variables generated by the SMC algorithm at Step t in a vector $\boldsymbol{Z}_t$. That is, we write $\boldsymbol{Z}_1 := \boldsymbol{X}_1$ and $\boldsymbol{Z}_t := (O_{t-1}, \boldsymbol{A}_{t-1}, \boldsymbol{X}_t)$, for $t > 1$. These take values in the spaces $\mathbf{Z}_1 := \mathbf{X}_1$ and $\mathbf{Z}_t := \mathsf{O}_{t-1} \times \mathbf{A}_{t-1} \times \mathbf{X}_t$, for $t > 1$. Here, O_{t-1} is an auxiliary variable taking values in some space O_{t-1}. It will parametrise the proposal and resampling kernels and hence allow us to formalise adaptive resampling schemes, for instance.

After Step $t-1$, we have already sampled $\boldsymbol{Z}_{1:t-1}$ based on which we have constructed an approximation of γ_{t-1} given by the weighted empirical measure

$$\gamma_{t-1}^{\mathrm{SMC},N_{1:t-1}} := \sum_{n=1}^{N_{t-1}} w_{t-1}^n(\boldsymbol{Z}_{1:t-1}) \delta_{X_{1:t-1}^{B_{1:t-1|t-1}^n}}.$$

Here, we have used the following notation.

- $X_{1:t}^{b_{1:t}} = (X_1^{b_1}, \ldots, X_t^{b_t})$ denotes the particle *path* or *trajectory* associated with some particle indices $b_{1:t}$.
- $B_{1:t|t}^n = (B_{1|t}^n, \ldots, B_{t|t}^n)$ represents the particle indices formed by tracing back the nth ancestral lineage at Step t (as determined by the parent indices $\boldsymbol{A}_{1:t-1} = (\boldsymbol{A}_1, \ldots, \boldsymbol{A}_{t-1})$), i.e. $B_{t|t}^n = n$ and

$$B_{s|t}^n = A_s^{B_{s+1|t}^n}, \quad \text{for } s < t.$$

- $w_{t-1}^n(z_{1:t-1}) \in [0, \infty)$ is a weight associated with the nth particle path at Step t. As before, we use this terminology even though these 'weights' do not sum to 1, in general. The corresponding *self-normalised* weights are denoted $W_{t-1}^n(z_{1:t-1}) := w_{t-1}^n(z_{1:t-1})/[\sum_{n=1}^{N_{t-1}} w_{t-1}^m(z_{1:t-1})]$.

At Step t, to obtain an approximation of γ_t,

$$\gamma_t^{\textsc{smc},N_{1:t}} := \sum_{n=1}^{N_t} w_t^n(\boldsymbol{Z}_{1:t}) \delta_{\boldsymbol{X}_{1:t}^{B^n_{1:t|t}}},$$

SMC algorithms sample additional particles, parent indices and potentially other auxiliary random variables, all collected in the ordered set $\boldsymbol{Z}_t$, from a particular stochastic kernel $\Psi_t \in \mathcal{K}_1(\mathbf{Z}^\times_{1:t-1}, \mathbf{Z}_t)$ defined below. The extended set of samples $\boldsymbol{Z}_{1:t}$ is then used to construct a new collection of weights, $(w_t^n(\boldsymbol{z}_{1:t}))_{n \in \mathsf{K}_t}$. In many cases, the computational cost of sampling the additional random variables and of computing the new weights is constant in t. This constant cost per step makes SMC methods particularly beneficial in settings in which sequences of measures need to be approximated under computational constraints, e.g. in real-time object-tracking applications.

The conditional distribution of the random variables generated at Step t is then given by the stochastic kernel

$$\begin{aligned}\Psi_t(\boldsymbol{z}_{1:t-1}, \mathrm{d}\boldsymbol{z}_t) := S_{t-1}(\boldsymbol{z}_{1:t-1}, \mathrm{d}o_{t-1}) R_{t-1}((\boldsymbol{z}_{1:t-1}, o_{t-1}), \mathrm{d}\boldsymbol{a}_{t-1}) \\ \times Q_t((\boldsymbol{z}_{1:t-1}, o_{t-1}, \boldsymbol{a}_{t-1}), \mathrm{d}\boldsymbol{x}_t).\end{aligned}$$

The individual components of this kernel are as follows. Some examples of these quantities are discussed in Section 2.3.

- $R_{t-1} \in \mathcal{K}_1(\mathbf{Z}^\times_{1:t-1} \times \mathrm{O}_{t-1}, \mathbf{A}_{t-1})$ generates the parent indices $\boldsymbol{A}_{t-1}$, a process usually known as *resampling* (see Remark 2.2 in the next subsection for a more precise explanation of the terminology).
- $Q_t \in \mathcal{K}_1(\mathbf{Z}^\times_{1:t-1} \times \mathrm{O}_{t-1} \times \mathbf{A}_{t-1}, \mathbf{X}_t)$, for $t > 1$, generates new particles at Step t. It is commonly referred to as the (particle) *proposal kernel.* At Step 1, $\boldsymbol{X}_1$ is sampled from some suitable proposal distribution $q_1 \in \mathcal{M}_1(\mathbf{X}_1)$.
- $S_{t-1} \in \mathcal{K}_1(\mathbf{Z}^\times_{1:t-1}, \mathrm{O}_{t-1})$ generates an auxiliary variable O_{t-1} which governs the type of resampling or proposal kernel chosen at Step t.

2.1.3 Resampling

2.2 Remark. *We use the convention that 'not resampling' at Step $(s+1)$ of an SMC algorithm refers to the case that $R_s((\boldsymbol{z}_{1:s}, o_s), \cdot) = \delta_{(1,\dots,N_s)}$*

(assuming that $N_s = N_{s+1}$). Otherwise, the non-degenerate distribution $\widetilde{R}_s(z_{1:s}, \cdot)$ from which the parent indices $\boldsymbol{A}_s$ are sampled will be called a resampling scheme. *The latter could also depend on the auxiliary variable O_s but we do not make this explicit to keep the notation concise.*

Some common resampling schemes, such as *multinomial, stratified,* and *systematic* resampling, are outlined in Appendix A and Cappé, Godsill and Moulines (2007) provide an overview. The following definition are given by (e.g. Andrieu et al., 2010, Equations 23 and 24).

2.3 Definition. *Let $\boldsymbol{A}_s \sim \widetilde{R}_s(\boldsymbol{Z}_{1:s}, \cdot)$.*

(1) The resampling scheme $\widetilde{R}_s$ is called unbiased *if*

$$\mathbb{E}\left[\sum_{n=1}^{N_{s+1}} \mathbb{1}_{\{m\}}(A_s^n) \middle| \boldsymbol{Z}_{1:s}, O_s\right] = N_{s+1} W_s^m(\boldsymbol{Z}_{1:s}),$$

for all $m \in \mathsf{K}_s$.

(2) The resampling scheme $\widetilde{R}_s$ is called exchangeable *if*

$$\mathbb{E}\big[\mathbb{1}_{\{m\}}(A_s^n) \big| \boldsymbol{Z}_{1:s}, O_s\big] = \mathbb{E}\big[\mathbb{1}_{\{m\}}(A_s^k) \big| \boldsymbol{Z}_{1:s}, O_s\big],$$

for all $(m, n, k) \in \mathsf{K}_s \times \mathsf{K}_{s+1}^2$.

2.4 Remark. *Informally, unbiased resampling schemes yield a new set of samples which still target the same measure as before but whose weights are all equal. In contrast, in our terminology, 'biased' resampling schemes do not lead to equally weighted samples. We stress that using 'biased' resampling schemes does not jeopardise the unbiasedness of SMC-based estimates of integrals of the form $\gamma_t(f)$ (as long as the post-resampling weights take the resampling scheme into account).*

Although unbiased resampling schemes are common in practice, the generic framework developed in this chapter shows that neither unbiasedness nor exchangeability is actually needed for valid SMC algorithms, i.e. needed for SMC algorithms that yield unbiased estimates of integrals of the form $\gamma_t(f)$. Indeed, unbiasedness may not even be desirable, as the following examples show.

2.5 Example ('biased' resampling schemes). *In the some situations, it can be beneficial to use biased resampling schemes (as stressed in Remark 2.4, their use does not introduce any bias into SMC-based estimates).*

(1) *The discrete particle filter from Fearnhead (1998) described in Subsection 2.3.4 employs a biased resampling scheme which is optimal for finite state spaces.*

(2) *A biased resampling scheme called 'chopthin' resampling was introduced in Gandy and Lau (2015) and is related to that of Fearnhead (1998). It appears to empirically outperform 'unbiased' resampling schemes such as systematic resampling in simple state-space models. Intuitively, the 'unbiasedness' property (i.e. yielding evenly-weighted samples) may be stronger than what is actually needed to ensure stability of the algorithm and may require the introduction of too much Monte Carlo error. Further theoretical investigation of this issue is clearly needed.*

(3) *Assume the number of particles is constant, i.e. $N_t = N$ for some $N \in \mathbb{N}$. If the weights at the end of Step t do not depend on $\boldsymbol{Z}_t$, then it can be beneficial to use a resampling scheme which is biased, in the sense that it is based on the particle weights from the current step, $W_t^n(\boldsymbol{Z}_{1:t})$, rather than those from the previous step, $W_{t-1}^n(\boldsymbol{Z}_{1:t-1})$, e.g. a resampling scheme such that for any $m \in \mathsf{K}_s$,*

$$\mathbb{E}\Biggl[\sum_{n=1}^{N} \mathbb{1}_{\{m\}}(A_{t-1}^n) \Biggm| \boldsymbol{Z}_{1:t-1}, O_{t-1}\Biggr] = N W_t^m(\boldsymbol{Z}_{1:t}).$$

This is often viewed as 'switching the order' of sampling and resampling.

(4) *More generally, biased resampling schemes permit incorporating 'future information' into the particle system (Wang, Chen & Guo, 2002; Lin, Chen & Liu, 2013). Other potential benefits are mentioned in Liu (2001, p. 73).*

2.1.4 Generic Algorithm

Let F_t be a collection of γ_t-integrable test functions and assume that we want to approximate $\gamma_t(f_t)$. In this subsection, in Algorithm 2.6, we summarise the generic SMC scheme developed above. As already mentioned, essentially any SMC algorithm can be viewed as a special case of this algorithm. The next section shows that this algorithm can itself be viewed as a special case of the MOSIS scheme from the previous chapter.

2.6 Algorithm (sequential Monte Carlo). *At Step $t \in \mathbb{N}$,*

(1) sample $\boldsymbol{Z}_t \sim \Psi_t(\boldsymbol{z}_{1:t-1}, \cdot)$,

(2) calculate the (updated) weights $(w_t^n(\boldsymbol{Z}_{1:t}))_{n \in \mathsf{K}_t}$,

(3) approximate $\gamma_t(f_t)$ by $\gamma_t^{\textsc{smc},N_{1:t}}(f_t)$, for $f_t \in \mathsf{F}_t$.

As in the previous chapter, the usual SMC approximation of the probability measure $\pi_t = \gamma_t/\gamma_t(\mathbb{1})$ results from taking the self-normalised version of $\gamma_t^{\textsc{smc},N_{1:t}}$ and hence approximating π_t by

$$\pi_t^{\textsc{smc}\star,N_{1:t}} := \frac{\gamma_t^{\textsc{smc},N_{1:t}}}{\gamma_t^{\textsc{smc},N_{1:t}}(\mathbb{1})} = \sum_{n=1}^{N_t} W_t^n(\boldsymbol{Z}_{1:t}) \delta_{X_{1:t}^{B^n_{1:t|t}}},$$

recalling that $W_t^n(\boldsymbol{z}_{1:t}) = w_t^n(\boldsymbol{z}_{1:t})/\mathfrak{z}_t^{\textsc{smc},N_{1:t}}$ is the nth self-normalised Step-t particle weight. Furthermore,

$$\mathfrak{z}_t^{\textsc{smc},N_{1:t}} := \gamma_t^{\textsc{smc},N_{1:t}}(\mathbb{1}) = \sum_{n=1}^{N_t} w_t^n(\boldsymbol{Z}_{1:t})$$

is the usual SMC estimate of the normalising constant. This estimate is unbiased – a famous result first proved by Del Moral (1996) using martingale techniques. However, this result also follows trivially from the interpretation of SMC as IS described in the next section.

In the remainder of this chapter, we show that a number of well-known SMC algorithms can be viewed as special cases of the generic algorithm presented in this section. First, however, we show that the generic SMC algorithm is itself a special case of MOSIS.

2.2 Interpretation as Marginalised One-Sample Importance Sampling

2.2.1 Extended Proposal Distribution

In this section, we introduce SMC methods as a special case of the MOSIS framework described in Chapter 1. Given a target measure $\gamma_t \in \mathcal{M}(\mathsf{X}_{1:t}^{\times})$, we construct an extended target measure $\bar{\gamma}_t \in \mathcal{M}(\bar{\mathbf{X}}_t)$ and extended proposal distributions $\bar{\psi}_t \in \mathcal{M}_1(\bar{\mathbf{X}}_t)$ such that

- $\bar{\gamma}_t$ admits γ_t (and hence $\tilde{\gamma}_t$) as a marginal,
- we can sample from $\bar{\psi}_t$,
- $\bar{w}_t := \mathrm{d}\bar{\gamma}_t / \mathrm{d}\bar{\psi}_t$ exists and can be evaluated point-wise.

The usual SMC approximation of γ_t, $\gamma_t^{\text{SMC},N_{1:t}}$, can then be obtained by Rao–Blackwellising $\bar{\gamma}_t^{\text{IS},1}$, where the latter is an IS approximation of $\bar{\gamma}_t$ based on a single sample point $\bar{\boldsymbol{X}}_t \sim \bar{\psi}_t$.

This interpretation of SMC methods as a special case of MOSIS was developed – though not stated explicitly – in the seminal work by Andrieu et al. (2010). The explicit construction here closely follows Lee, Murray and Johansen (in prep.) whose work allows for adaptive resampling schemes and removes the need for resampling schemes to be exchangeable. Here, we take an even more general approach which also abolishes the need for resampling schemes to be unbiased. For instance, this was suggested as a desirable extension by R. Chen (2010). We show that the approach preserves exactness of the algorithm in the sense of unbiasedly estimating integrals with respect to the unnormalised target distribution, i.e. integrals of the form $\gamma_t(f_t)$.

First, in this subsection, we construct the extended proposal distribution, $\bar{\psi}_t \in \mathcal{M}_1(\bar{\mathbf{X}})$, where $\bar{\mathbf{X}}_t := \mathsf{X}_{1:t}^{\times} \times \mathsf{K}_{1:t}^{\times} \times \mathbf{Z}_{1:t}^{\times}$. This extended proposal distribution will be such that sampling

$$\bar{\boldsymbol{X}}_t = (U_{1:t}, B_{1:t}, \boldsymbol{Z}_{1:t}) \sim \bar{\psi}_t$$

can be achieved by

(1) sampling $\boldsymbol{Z}_{1:t}$ by running an SMC algorithm up to Step t,

(2) sampling a Step-t particle index B_t and setting $B_{1:t-1} := B_{1:t-1|t}^{B_t}$,

(3) setting $U_{1:t}$ equal to the B_tth particle trajectory, i.e. $U_{1:t} := X_{1:t}^{B_{1:t}}$.

We first note that the distribution of all random variables generated by an SMC algorithm up to Step t is given by $\psi_t := q_1 \in \mathcal{M}_1(\mathbf{Z}_1)$, if $t = 1$, and, for $t > 1$,

$$\psi_t := \psi_{t-1} \otimes \Psi_t = \psi_1 \otimes \Psi_{2:t}^{\otimes} \in \mathcal{M}_1(\mathbf{Z}_{1:t}^{\times}). \tag{2.1}$$

The extended proposal distribution can then be defined as

$$\bar{\psi}_t := \psi_t \otimes \Xi_t \in \mathcal{M}_1(\bar{\mathbf{X}}_t).$$

Here, $\Xi_t \in \mathcal{K}_1(\mathbf{Z}_{1:t}^{\times}, \mathsf{X}_{1:t}^{\times} \times \mathsf{K}_{1:t}^{\times})$ is a stochastic kernel given by

$$\begin{aligned}\Xi_t(\boldsymbol{z}_{1:t}, \mathrm{d}u_{1:t} \times \mathrm{d}b_{1:t}) \\ := \xi_{t|t}(\boldsymbol{z}_{1:t}, \mathrm{d}b_t)\Biggl[\prod_{s=1}^{t-1} \delta_{a_s^{b_{s+1}}}(\mathrm{d}b_s)\Biggr]\delta_{x_{1:t}^{b_{1:t}}}(\mathrm{d}u_{1:t}),\end{aligned} \qquad (2.2)$$

where $\xi_{t|t} \in \mathcal{K}_1(\boldsymbol{Z}_{1:t}^{\times}, \mathsf{K}_t)$ is some stochastic kernel used for sampling the Step-t particle index, B_t.

As shown later in this section, the SMC approximation of γ_t is obtained through a Rao–Blackwellisation step which analytically integrates out $(B_{1:t}, U_{1:t})$ given $\boldsymbol{Z}_{1:t}$. By Equation 2.1 we therefore actually only need to sample $\boldsymbol{Z}_t \sim \Psi_t(\boldsymbol{z}_{1:t-1}, \cdot\,)$ at Step t and this typically has computational complexity $\mathcal{O}(N_{t-1} \vee N_t)$.

2.2.2 Extended Target Measure

In this subsection, we define an extended measure $\bar{\gamma}_t$ under a normalised version of which a collection of random variables $U_{1:t}$ – which coincides with the particle path with indices $B_{1:t}$ at Step t – is marginally distributed according to $\pi_t = \gamma_t/\gamma_t(\mathbb{1})$. Write $\bar{\boldsymbol{Z}}_t := (B_{1:t}, \boldsymbol{Z}_{1:t})$. With some abuse of notation pertaining to the order of the components of $\bar{\boldsymbol{X}}_t$, the extended target measure is

$$\bar{\gamma}_t(\mathrm{d}\bar{\boldsymbol{x}}_t) = \gamma_t(\mathrm{d}u_{1:t})\bar{\Pi}_t^{\textsc{csmc}}(u_{1:t}, \mathrm{d}\bar{\boldsymbol{z}}_t), \qquad (2.3)$$

where the stochastic kernel $\bar{\Pi}_t^{\textsc{csmc}} \in \mathcal{K}_1(\mathsf{X}_{1:t}^{\times}, \bar{\mathbf{Z}}_t)$, for $\bar{\mathbf{Z}}_t := \mathsf{K}_{1:t}^{\times} \times \mathbf{Z}_{1:t}^{\times}$, is given by

$$\begin{aligned}\bar{\Pi}_t^{\textsc{csmc}}(u_{1:t}, \mathrm{d}\bar{\boldsymbol{z}}_t) := \bar{\Pi}_{1|t}^{\textsc{csmc}}(u_{1:t}, \mathrm{d}b_1 \times \mathrm{d}\boldsymbol{z}_1) \\ \times \prod_{s=2}^{t} \bar{\Pi}_{s|t}^{\textsc{csmc}}((u_{1:t}, b_{1:s-1}, \boldsymbol{z}_{1:s-1}), \mathrm{d}b_s \times \mathrm{d}\boldsymbol{z}_s)\end{aligned} \qquad (2.4)$$

with

$$\begin{aligned}&\bar{\Pi}^{\mathrm{CSMC}}_{1|t}(u_{1:t}, \mathrm{d}b_1 \times \mathrm{d}\boldsymbol{z}_1)\\ &\qquad := \Lambda_1(u_1, \mathrm{d}b_1)\delta_{u_1}(\mathrm{d}x_1^{b_1})q_1^{\mathrm{c}}((b_1, x_1^{b_1}), \mathrm{d}\boldsymbol{x}_1^{-b_1})\end{aligned}$$

and

$$\begin{aligned}&\bar{\Pi}^{\mathrm{CSMC}}_{s|t}((u_{1:t}, b_{1:s-1}, \boldsymbol{z}_{1:s-1}), \mathrm{d}b_s \times \mathrm{d}\boldsymbol{z}_s)\\ &\qquad := S_{s-1}(\boldsymbol{z}_{1:s-1}, \mathrm{d}o_{s-1})\delta_{b_{s-1}}(\mathrm{d}a_{s-1}^{b_s})\delta_{u_s}(\mathrm{d}x_s^{b_s})\\ &\qquad\quad \times \Lambda_s((u_{1:s}, \boldsymbol{z}_{1:s-1}, o_{s-1}, a_{s-1}^{b_s}), \mathrm{d}b_s)\\ &\qquad\quad \times R_{s-1}^{\mathrm{c}}((\boldsymbol{z}_{1:s-1}, o_{s-1}, b_s, a_{s-1}^{b_s}), \mathrm{d}\boldsymbol{a}_{s-1}^{-b_s})\\ &\qquad\quad \times Q_s^{\mathrm{c}}((\boldsymbol{z}_{1:s-1}, o_{s-1}, \boldsymbol{a}_{s-1}, b_s, x_s^{b_s}), \mathrm{d}\boldsymbol{x}_s^{-b_s}).\end{aligned}$$

Here, we have defined the following quantities.

- $\Lambda_s \in \mathcal{K}_1(\mathsf{X}^{\times}_{1:s} \times \mathbf{Z}^{\times}_{1:s-1} \times \mathsf{O}_{s-1} \times \mathsf{A}_{s-1}, \mathsf{K}_s)$ induce a distribution over the particle indices $B_{1:t}$. A common, valid choice for these kernels is given in Assumption 2.9 in the next subsection.
- $R_{s-1}^{\mathrm{c}}((\boldsymbol{z}_{1:s-1}, o_{s-1}, n, a_{s-1}^n), \cdot\,)$ is the conditional distribution of the parent indices $\boldsymbol{A}_{s-1}^{-n} := (A_{s-1}^{1:n-1}, A_{s-1}^{n+1:N_s})$ under $R_{s-1}((\boldsymbol{z}_{1:s-1}, o_{s-1}), \cdot\,)$ given that the nth parent index equals a_{s-1}^n.
- $Q_s^{\mathrm{c}}((\boldsymbol{z}_{1:s-1}, o_{s-1}, \boldsymbol{a}_{s-1}, n, x_s^n), \cdot\,)$ is the conditional distribution of the particles $\boldsymbol{X}_s^{-n} := (X_s^{1:n-1}, X_s^{n+1:N_s})$ under $Q_s((\boldsymbol{z}_{1:s-1}, o_{s-1}, \boldsymbol{a}_{s-1}), \cdot\,)$ given that the nth particle is equal to x_s^n. The conditional distribution q_1^{c} is similarly defined.

2.7 Remark. *The kernel $\bar{\Pi}_t^{\mathrm{CSMC}}$ is a so-called* conditional sequential Monte Carlo *(CSMC) kernel introduced by Andrieu et al. (2010). Though, despite this name, neither $\bar{\Pi}_t^{\mathrm{CSMC}}(u_{1:t}, \cdot\,)$ nor any of its marginals are usually conditional distributions relative to ψ_t. Using this kernel frees us from having to evaluate (a density with respect to) the marginal proposal distribution of a single particle path – which is usually intractable – when evaluating the importance weights in the next subsection (see Remark 1.16).*

Sampling according to the kernel $\bar{\Pi}_t^{\mathrm{CSMC}}$ is necessary within particle Gibbs samplers described in Section 3.4. More importantly, though, Equation 2.3 demonstrates that $\bar{\Pi}_t^{\mathrm{CSMC}}$ is a crucial ingredient in the particular justification of SMC algorithms which we have presented here.

2.2.3 Importance Weights

For the moment, our aim is to approximate $\bar{\gamma}_t$ via IS using the proposal distribution $\bar{\psi}_t$. The Rao–Blackwellisation described in the next subsection then leads to the usual SMC approximation of γ_t. We must therefore ensure that the Radon–Nikodým derivative

$$\begin{aligned}
\overline{w}_t(\bar{x}_t) &:= \frac{\mathrm{d}\bar{\gamma}_t}{\mathrm{d}\bar{\psi}_t}(\bar{x}_t) \\
&= \mathbb{1}_{\{u_{1:t}\}}(x_{1:t}^{b_{1:t}}) \frac{\gamma_1(\mathrm{d}u_1) \mathit{\Lambda}_1(u_1, \{b_1\})}{\xi_{t|t}(\boldsymbol{z}_{1:t}, \{b_t\}) q_1^{\mathrm{M}}(b_1, \mathrm{d}u_1)} \\
&\quad \times \prod_{s=2}^{t} \frac{\mathit{\Lambda}_s((u_{1:s}, \boldsymbol{z}_{1:s-1}, o_{s-1}, b_{s-1}), \{b_s\}) \, \mathbb{1}_{\{b_{s-1}\}}(a_{s-t}^{b_s})}{R_{s-1}^{\mathrm{M}}((\boldsymbol{z}_{1:s-1}, o_{s-1}, b_s), \{b_{s-1}\})} \\
&\qquad \times \frac{\mathit{\Gamma}_s(u_{1:s-1}, \mathrm{d}u_s)}{Q_s^{\mathrm{M}}((\boldsymbol{z}_{1:s-1}, o_{s-1}, \boldsymbol{a}_{s-1}, b_s), \mathrm{d}u_s)}
\end{aligned} \tag{2.5}$$

is well defined. Here, we have slightly abused notation by writing Radon–Nikodým derivatives as $\mu(\mathrm{d}x)/\nu(\mathrm{d}x) := [\mathrm{d}\mu/\mathrm{d}\nu](x)$, for any two measures $\mu \ll \nu$, and where we have defined the following quantities.

- $\mathit{\Gamma}_s \in \mathcal{K}(\mathsf{X}_{1:s-1}^{\times}, \mathsf{X}_s)$ is a kernel which extends the Step-($s - 1$) target measure to the Step-s target measure, i.e. it satisfies $\gamma_{s-1} \otimes \mathit{\Gamma}_s = \gamma_s$.
- $R_{s-1}^{\mathrm{M}}((\boldsymbol{z}_{1:s-1}, o_{s-1}, n), \cdot\,)$ denotes the 'marginal' distribution of the nth parent index under $R_{s-1}((\boldsymbol{z}_{1:s-1}, o_{s-1}), \cdot\,)$.
- $Q_s^{\mathrm{M}}((\boldsymbol{z}_{1:s-1}, o_{s-1}, \boldsymbol{a}_{s-1}, n), \cdot\,)$ denotes the 'marginal' distribution of the nth particle under $Q_s((\boldsymbol{z}_{1:s-1}, o_{s-1}, \boldsymbol{a}_{s-1}), \cdot\,)$. The marginal Step-1 proposal distribution $q_1^{\mathrm{M}}(b_1, \cdot\,)$ is similarly defined.

2.8 Remark. *The kernels R_{s-1}^{M} and Q_s^{M} induce marginal distributions only in the sense that they do not condition on the other parent indices or particles generated at Step s. They generally still depend on all the auxiliary variables, parent indices and particles sampled at previous steps. Indeed, this is why the 'conditional' SMC kernel does not represent a (full) conditional distribution under the distribution induced by the SMC algorithm as pointed out in Remark 2.7 (see also Remark 1.16).*

We will comment on particular choices for the kernels and measures guaranteeing the existence of the above importance weight in Section 2.3.

Distribution Over Particle Indices. The kernels Λ_s introduced in the previous subsection need to be chosen carefully to preserve absolute continuity, especially if a non-exchangeable resampling scheme is used.

A generally applicable choice considered in Lee, Murray and Johansen (in prep.), which is also implicitly used by most SMC algorithms, is to let Λ_s be a time-reversal kernel of some stochastic kernel $\kappa_s \in \mathcal{K}_1(\mathsf{X}_{1:s}^{\times}, \mathsf{K}_s)$ (which defines a distribution over B_t) under R_{s-1}^{M}, as defined in Assumption 2.9.

2.9 Assumption. *$\Lambda_1 := \kappa_1$ and, for $s > 1$,*

$$\begin{aligned}\Lambda_s((u_{1:s}, z_{1:s-1}, o_{s-1}, a_{s-1}^{b_s}), \{b_s\}) \\ = \frac{R_{s-1}^{\mathrm{M}}((z_{1:s-1}, o_{s-1}, b_s), \{a_{s-1}^{b_s}\})\kappa_s(u_{1:s}, \{b_s\})}{\sum_{n=1}^{N_s} \kappa_s(u_{1:s}, \{n\}) R_{s-1}^{\mathrm{M}}((z_{1:s-1}, o_{s-1}, n), \{a_{s-1}^{n}\})}.\end{aligned} \tag{2.6}$$

It often suffices to let $\kappa_s(u_{1:s}, \cdot\,) \equiv \mathrm{Unif}_{\mathsf{K}_s}$. However, more complex kernels are sometimes needed to ensure absolute continuity in Equation 2.5. For instance, a more complex kernel κ_s is needed in the discrete particle filter summarised in Subsection 2.3.4.

The main advantage of the time-reversal kernel is that the importance weight in Equation 2.5 depends on the resampling distribution only through the denominator in Equation 2.6. Hence, it is usually not necessary to require the resampling scheme to be exchangeable – even if we cannot evaluate the distribution implied by R_{s-1}^{M}. For instance, if we use an unbiased resampling scheme and if $\kappa_s(u_{1:s}, \cdot\,) := \mathrm{Unif}_{\mathsf{K}_s}$, then R_{s-1}^{M} drops out in the importance weights from Equation 2.5 because

$$\begin{aligned}&\frac{\Lambda_s((u_{1:s}, z_{1:s-1}, o_{s-1}, b_{s-1}), \{b_s\})}{R_{s-1}^{\mathrm{M}}((z_{1:s-1}, o_{s-1}, b_s), \{b_{s-1}\})} \\ &\quad = \begin{cases} 1/W_{s-1}^{b_{s-1}}(z_{1:s-1}), & \text{if we resample at Step } s, \\ \mathbb{1}_{\{b_{s-1}\}}(b_s), & \text{otherwise.}\end{cases}\end{aligned}$$

However, note that sampling according to Λ_s (which depends on R_{s-1}^{M}) and R_{s-1}^{C} is still required when sampling from the CSMC kernel. For various common resampling schemes, R_{s-1}^{M} and R_{s-1}^{C} are derived in Lee, Murray and Johansen (in prep.) and for completeness, they are also stated in Appendix A of this work.

2.2.4 Rao–Blackwellisation

Let $\bar{\gamma}_t^{\text{IS},1} := \bar{w}_t(\bar{\boldsymbol{X}}_t)\delta_{\bar{\boldsymbol{X}}_t}$ be an IS approximation of the extended target measure $\bar{\gamma}_t$ based on a single sample $\bar{\boldsymbol{X}}_t = (U_{1:t}, B_{1:t}, \boldsymbol{Z}_{1:t}) \sim \bar{\psi}_t$.

Of course, we are only interested in approximating the marginal γ_t of $\bar{\gamma}_t$. The usual SMC approximation of this marginal measure, $\gamma_t^{\text{SMC},N_{1:t}}$, can be obtained by Rao–Blackwellising $\bar{\gamma}_t^{\text{IS},1}$ as described in Lee, Murray and Johansen (in prep.).

More precisely, note that

$$w_t^{b_{1:t}}(\boldsymbol{Z}_{1:t}) := \mathbb{E}[\bar{w}_t(\bar{\boldsymbol{X}}_t)\,\mathbb{1}_{\{b_{1:t}\}}(B_{1:t})|\boldsymbol{Z}_{1:t}]$$

is non-zero only if $b_{1:t}$ coincides with a particle lineage under the SMC algorithm, i.e. if $b_{1:t} = B_{1:t|t}^n$, for some $n \in \mathsf{K}_t$. We can therefore identify N_t (unnormalised) Step-t particle weights, for $n \in \mathsf{K}_t$, as

$$w_t^n(\boldsymbol{z}_{1:t}) := w_t^{b_{1:t|t}^n}(\boldsymbol{z}_{1:t}).$$

For any $A \in \mathscr{B}(\mathsf{X}_{1:t}^{\times})$, a MOSIS approximation of $\gamma_t(A)$ is thus given by

$$\begin{aligned}
\gamma_t^{\text{MOSIS},N_{1:t}}(A) &= \mathbb{E}\big[\bar{\gamma}_t^{\text{IS},1}(A \times \bar{\mathbf{Z}}_t)\big|\boldsymbol{Z}_{1:t}\big]\\
&= \sum_{b_{1:t}\in\mathsf{K}_{1:t}^{\times}} w_t^{b_{1:t}}(\boldsymbol{Z}_{1:t})\delta_{X_{1:t}^{b_{1:t}}}(A)\\
&= \sum_{n=1}^{N_t} w_t^n(\boldsymbol{Z}_{1:t})\delta_{X_{1:t}^{B_{1:t|t}^n}}(A)\\
&= \gamma_t^{\text{SMC},N_{1:t}}(A).
\end{aligned}$$

The above construction immediately implies that the SMC estimate of the normalising constant, $\mathfrak{z}_t^{\text{SMC},N_{1:t}} = \gamma_t^{\text{SMC},N_{1:t}}(\mathbb{1}) = \bar{\gamma}_t^{\text{IS},1}(\mathbb{1})$, is a (one-sample) IS estimate and is therefore unbiased. Nonetheless, we stress again that the unbiasedness property alone does not ensure estimates that are useful in practice, i.e. estimates whose error can be controlled. Conditions under which this is guaranteed are summarised in Subsection 2.2.5.

Finally, recall that $\bar{w}_t = \mathrm{d}\bar{\gamma}_t/\mathrm{d}\bar{\psi}_t$. For later reference, we state the following slight generalisation of Andrieu et al. (2010, Theorem 2) (but which is really just a special case of Proposition 1.13).

2.10 Proposition. *Assume that*

$$\xi_{t|t}(\boldsymbol{z}_{1:t}, \{n\}) = W_t^n(\boldsymbol{z}_{1:t}),$$

for any $(n, \boldsymbol{z}_{1:t}) \in \mathsf{K}_t \times \mathbf{Z}^{\times}_{1:t}$, *then*

$$\bar{w}_t(\bar{\boldsymbol{x}}_t) = \mathfrak{z}_t^{\textsc{smc}, N_{1:t}}, \quad \text{for any } \bar{\boldsymbol{x}}_t \in \bar{\mathbf{X}}_t.$$

Proof. This follows immediately from the definition of $w_t^n(\boldsymbol{z}_{1:t})$. □

2.2.5 Theoretical Results

In this subsection, we briefly summarise some of the available theoretical results for estimates obtained from SMC algorithms. Throughout, we assume that $N_1 = N_2 = \ldots = N$, for simplicity.

Specific Test Function. SMC methods are particularly suited to approximating integrals of the form $\pi_t(f_t)$, where $\pi_t := \gamma_t / \gamma_t(\mathbb{1})$ and

$$f_t := \mathbb{1}_{\mathsf{X}^{\times}_{1:t-1}} \otimes f_{t,t}.$$

Here, $f_{t,t}$ is some suitable real-valued test function with domain X_t. For such a test function (and for suitably ergodic kernels $\varGamma_s$) we are essentially performing IS on a space of constant dimension (if $\mathsf{X}_s = \mathsf{X}_t$, for $s, t \in \mathsf{T}$). We assume such a test function in our statements of the theoretical results below. Most of these results can be readily generalised to vector-valued functions, e.g. by Cramér–Wold type arguments.

Non-Asymptotic Error Bounds. For instance, for such test functions, and for various SMC algorithms, non-asymptotic (in N) uniform-in-t $\mathcal{L}^p$ error bounds have been established, i.e. if $f_{t,t}$ is bounded (and under further 'strong mixing assumptions' on the kernels $\varGamma_s$) Del Moral and Miclo (2000) and Del Moral (2004, Theorem 7.4.4) show that

$$\sup_{t \in \mathsf{T}} \mathbb{E}\Big[\big(\pi_t^{\textsc{smc}^\star, N_{1:t}}(f_t) - \pi_t(f_t)\big)^p\Big]^{1/p} \leq \frac{c(p)}{N},$$

where $c(p) > 0$ is some finite constant which does not depend on N.

Central Limit Theorems. Del Moral and Guionnet (1999), as well as Del Moral (2004, Section 9.4.2) and also Chopin (2004), Künsch (2005) establish CLTs for the asymptotic behaviour (as $N \to \infty$) of the error at Step t, i.e. they show that

$$\sqrt{N}\big[\pi_t^{\mathrm{SMC}\star,N_{1:t}}(f_t) - \pi_t(f_t)\big] \quad \overset{N\to\infty}{\longrightarrow} \quad Z \sim \mathrm{N}_{0,\sigma_t(f_t)},$$

in distribution, again under certain regularity conditions. They also provide explicit analytical expressions for the asymptotic variance, $\sigma_t(f_t)$.

In particular, Del Moral and Guionnet (2001), Chopin (2004), Douc and Moulines (2008) obtain uniform-in-t bounds on $\sigma_t(f_t)$. Finally, Beskos, Jasra and Thiéry (2014) extend such CLTs to a more general class of adaptive SMC algorithms.

Normalising-Constant Estimates. As indicated by Proposition 2.10, the properties of the SMC-based estimate of the normalising constant $\mathfrak{z}_t$, $\mathfrak{z}^{\mathrm{SMC},N_{1:t}}$, are strongly related to the performance of SMC methods. In particular, it is important to obtain precise estimates of the normalising constant when using SMC methods as part of a pseudo-marginal approach, e.g. when using one of the pseudo-marginal SMC methods mentioned in Subsection 2.3.5 or when employing the particle marginal Metropolis–Hastings algorithm mentioned in Subsection 3.3.4.

Again under strong mixing assumptions, Cérou, Del Moral and Guyader (2011) show that the non-asymptotic (in N) relative variance of the SMC-based estimate of normalising constant, i.e. the variance of $\mathfrak{z}_t^{\mathrm{SMC},N_{1:t}}/\mathfrak{z}$, grows linearly in the step number. More formally, Cérou et al. (2011, Theorem 5.1) shows that

$$N > c(t) \quad \Rightarrow \quad \mathbb{V}\left[\frac{\mathfrak{z}_t^{\mathrm{SMC},N_{1:t}}}{\mathfrak{z}_t}\right] = \mathbb{E}\left[\left(\frac{\mathfrak{z}_t^{\mathrm{SMC},N_{1:t}}}{\mathfrak{z}_t} - 1\right)^2\right] \leq \frac{c(t)}{N},$$

where $c(t) > 0$ is a constant which depends on the kernels $\mathit{\Gamma}_s$ and which, in many settings, can be shown to grow linearly in t.

Relaxed Assumptions. Most of the above-mentioned results are obtained under strong mixing assumptions which essentially require the state space X_t to be compact. Much research has recently been devoted to relaxing this assumption. See, for instance, Jasra and Doucet (2008), Whiteley (2012), Douc, Moulines and Olsson (2014), Whiteley (2013).

Extension to General Test Functions. For more general test functions than f_t defined above, CLTs typically still hold. This can easily be seen by working on the path space. However, uniform-in-t error bounds non-asymptotic error bounds or uniform-in-t bounds on the asymptotic variance can almost never be established unless the test function is constant in all but a uniformly bounded history of the spaces $\mathsf{X}_1, \dots, \mathsf{X}_t$.

2.3 Some Important SMC Algorithms

2.3.1 Simple SIR Algorithm

In this section, we discuss various standard SMC algorithms which can be obtained as special cases of Algorithm 2.6. First, in this subsection, we define a simple SMC algorithm which we call sequential importance resampling for the purpose of this work. Many widely used SMC algorithms – some of which are outlined in this section – are variants of this algorithm as pointed out in Doucet and Johansen (2011). However, there are also some SMC algorithms, such as the discrete particle filter described in Section 2.3.4, which are not.

2.11 Definition (sequential importance resampling). *An instance of the generic SMC algorithm is called* sequential importance resampling *(SIR) if for all $s \in \mathsf{T}$, $\boldsymbol{z}_{1:s} \in \mathbf{Z}_{1:s}^{\times}$, the following holds.*

(1) The auxiliary variable O_s takes values in $\mathsf{O}_s := \{0, 1\}$ and

$$\begin{aligned}
&S_s(\boldsymbol{z}_{1:s}, \mathrm{d}o_s) \\
&\qquad := [\mathbb{1}_{D_s}(h_{s-1}(\boldsymbol{z}_{1:s-1}))\delta_0 + \mathbb{1}_{\mathbb{R}\setminus D_s}(h_{s-1}(\boldsymbol{z}_{1:s-1}))\delta_1](\mathrm{d}o_s),
\end{aligned}$$

for some suitable $h_{s-1} \colon \mathbf{Z}_{1:s-1}^{\times} \to \mathbb{R}$ and $D_s \subseteq \mathbb{R}$.

(2) If $o_s = 0$ then $N_{s+1} = N_s$ (otherwise, the number of particles is allowed to change between SMC steps); $\kappa_s(u_{1:s}, \,\cdot\,) := \mathrm{Unif}_{\mathsf{K}_s}$, for $u_{1:s} \in \mathsf{X}_{1:s}^{\times}$, and

$$\begin{aligned}
&R_s((\boldsymbol{z}_{1:s}, o_s), \mathrm{d}\boldsymbol{a}_s) \\
&\qquad := \mathbb{1}_{\{0\}}(o_s)\delta_{(1,\dots,N_s)}(\mathrm{d}\boldsymbol{a}_s) + \mathbb{1}_{\{1\}}(o_s)\widetilde{R}_s(\boldsymbol{z}_{1:s}, \mathrm{d}\boldsymbol{a}_s),
\end{aligned}$$

where $\widetilde{R}_s(\boldsymbol{z}_{1:s}, \,\cdot\,)$ is an unbiased resampling scheme.

(3) *The marginal proposal kernels only depend on the history of the current particle path, i.e.*

$$Q_s((\boldsymbol{z}_{1:s-1}, o_{s-1}, \boldsymbol{a}_{s-1}), \mathrm{d}\boldsymbol{x}_s) = \prod_{n=1}^{N_s} P_s(x_{1:s-1}^{b^n_{1:s-1|s}}, \mathrm{d}x_s^n),$$

for some suitable stochastic kernel $P_s \in \mathcal{K}_1(\mathsf{X}^{\times}_{1:s-1}, \mathsf{X}_s)$, *for* $s > 1$, *and with* $P_1 \in \mathcal{M}_1(\mathsf{X}_1)$ *being some suitable proposal distribution at Step* 1.

As discussed in Example 2.13, the quantity $h_{s-1}(\boldsymbol{z}_{1:s-1})$ in Item 1 of Definition 2.11 is usually taken to be the ESS associated with the particle weights from the previous SMC step, commonly employed to obtain *adaptive* resampling schemes, i.e. to not necessarily resample at every SMC step.

The particle weights for SIR algorithms simplify as follows. Write

$$\begin{aligned}\bar{\mathsf{O}}_t &:= \bigl\{ s \in \mathbb{N}_{t-1} \bigm| o_s = 1 \bigr\},\\ \bar{l}_s &:= \sup\bigl\{ l \in \bar{\mathsf{O}}_t \cup \{0\} \bigm| l < s \bigr\} + 1,\end{aligned}$$

then if – as we assume throughout this work – the kernels Λ_s are taken to be the time-reversal kernels given in Assumption 2.9, the nth particle weight at Step t is given by

$$\begin{aligned}w_t^n(\boldsymbol{z}_{1:t}) &= \Biggl[\prod_{s\in\bar{\mathsf{O}}_t} \frac{1}{N_s} \sum_{m=1}^{N_s} \prod_{l=\bar{l}_s}^{s} G_l(x_{1:l}^{b^m_{1:l|s}})\Biggr] \prod_{l=\bar{l}_t}^{t} G_l(x_{1:l}^{b^n_{1:l|t}})\\ &= \begin{cases} G_t(x_{1:t}^{b^n_{1:t|t}}) w_{t-1}^{b^n_{t-1|t}}(\boldsymbol{z}_{1:t-1}), & \text{if } o_{t-1} = 0,\\ G_t(x_{1:t}^{b^n_{1:t|t}}) \dfrac{1}{N_{t-1}} \displaystyle\sum_{m=1}^{N_{t-1}} w_{t-1}^m(\boldsymbol{z}_{1:t-1}), & \text{if } o_{t-1} = 1.\end{cases}\end{aligned}$$

Here, the quantity $G_t(x_{1:t}^n)$, for any $x_{1:s} \in \mathsf{X}^{\times}_{1:s}$ defined by

$$G_s(x_{1:s}) := \frac{\mathrm{d}\Gamma_s(x_{1:s-1}, \cdot)}{\mathrm{d}P_s(x_{1:s-1}, \cdot)}(x_s),$$

is sometimes referred to as the nth 'incremental importance weight' at Step t because, conditional on $\boldsymbol{Z}_{1:t-1}$, the nth self-normalised particle weight $W_t^n(\boldsymbol{Z}_{1:t})$ used for constructing an approximation of $\pi_t = \gamma_t/\gamma_t(\mathbb{1})$ only depends on $(G_t(X_{1:t}^{B^n_{1:t|t}}))_{n\in\mathsf{K}_t}$.

The SIR estimate of the normalising constant takes the usual form

$$\mathcal{Z}_t^{\textsc{smc},N_{1:t}} = \sum_{n=1}^{N_t} w_t^n(\mathbf{z}_{1:t}) = \prod_{s \in \bar{\mathrm{O}}_t \cup \{t\}} \frac{1}{N_s} \sum_{n=1}^{N_s} \prod_{l=\bar{l}_s}^{s} G_l(x_{1:l}^{b_{1:l|s}^n}).$$

2.12 Example (bootstrap particle filter). *If we resample at every step of the SIR algorithm, i.e. if $O_s \equiv 1$, and if $\widetilde{R}_s(\mathbf{z}_{1:s}, \cdot)$ represents multinomial resmpling, then the SIR algorithm reduces to the simple* bootstrap particle filter *introduced in Stewart and McCarty Jr (1992), Gordon et al. (1993).*

2.13 Example (adaptive resampling). *It is well known that resampling, i.e. sampling the parent indices from some non-degenerate distribution – see Remark 2.2 – is wasteful and should only be performed when necessary. A common approach, known as* adaptive resampling *which is theoretically justified in Del Moral, Doucet and Jasra (2012), is to only resample whenever the following estimate of the ESS: $ESS_s = N_s \mathcal{Z}_s^2 / \bar{\gamma}_s(\bar{w}_s)$, given by*

$$\begin{aligned} ESS_s^{N_{1:s}} &:= \frac{N_s(\mathbb{E}[\bar{\gamma}_s^{\textsc{is},1}(\mathbb{1})|\mathbf{Z}_{1:s}])^2}{\mathbb{E}[\bar{\gamma}_s^{\textsc{is},1}(\bar{w}_s)|\mathbf{Z}_{1:s}]} \\ &= \frac{N_s[\sum_{n=1}^{N_s} w_s^n(\mathbf{Z}_{1:s})]^2}{\sum_{m=1}^{N_s}[w_s^m(\mathbf{Z}_{1:s})]^2/\xi_{s|s}(\mathbf{Z}_{1:s}, \{m\})}, \end{aligned}$$

falls below a threshold εN_s for some $\varepsilon \in (0,1)$. This can be reconciled with Definition 2.11. by setting $h_s(\mathbf{Z}_{1:s}) := ESS_s^{N_{1:s}}$ and $D_s := (\varepsilon N_s, N_s]$.

2.14 Example (sequential importance sampling). *If we do not resample in the SIR algorithm, i.e. if $O_s \equiv 0$, and $N_s = N \in \mathbb{N}$ for any $s \in \mathsf{T}$,*

$$w_t^n(\mathbf{z}_{1:t}) = \frac{1}{N} \prod_{s=1}^{t} G_s(x_{1:s}^n) = \frac{1}{N} \frac{\mathrm{d}\gamma_t}{\mathrm{d}P_{1:t}^{\otimes}}(x_{1:t}^n),$$

is the nth Step-t weight obtained from standard N-sample 'sequential' IS.

Other special cases of sequential IS, e.g. annealed importance sampling *(Jarzynski, 1997b, 1997a; Neal, 2001) can thus also be viewed as special cases of SIR and hence as special cases of SMC algorithms.*

2.3.2 SMC Samplers

We now deal with the situation in which the target measures, now denoted $\tilde{\gamma}_t$, are defined on a sequence of arbitrary spaces $\widetilde{\mathsf{X}}_t$. Hence, the SMC scheme introduced above is not applicable – not directly, at least. To approximate $\tilde{\gamma}_t$, we can use an SMC scheme targeting an artificially extended target measure γ_t on $\mathsf{X}^{\times}_{1:t}$, where $\mathsf{X}_t := \widetilde{\mathsf{X}}_t$ which admits $\tilde{\gamma}_t$ as a marginal. By working on the product space $\mathsf{X}^{\times}_{1:t}$, we circumvent the need for calculating certain integrals, e.g. marginal proposal densities, which would be required for evaluating the importance weights.

As a generic way of constructing γ_t, Del Moral et al. (2006b, 2007) (see also Peters, 2005) introduce a sequence of 'backward' Markov kernels $L_t \in \mathcal{K}_1(\mathsf{X}_{t+1}, \mathsf{X}_t)$ and set

$$\gamma_t := \tilde{\gamma}_t \otimes L^{\otimes}_{t-1:1} \in \mathcal{M}(\mathsf{X}^{\times}_{1:t}).$$

The SMC samplers from Del Moral et al. (2006b) are then simply SMC algorithms targeting this particular sequence of measures. To simplify the expressions for the importance weights, we assume in the following that the SMC algorithm is a SIR algorithm as specified in Definition 2.11 and that the marginal proposal kernel, P_t, is Markov. However, neither of these assumptions is strictly necessary.

Clearly, γ_t admits $\tilde{\gamma}_t$ as a marginal. Assuming that the backward Markov kernels are such that $\tilde{\gamma}_t \otimes L_{t-1} \ll \tilde{\gamma}_{t-1} \otimes P_t$ (with some abuse of the tensor-product notation pertaining to the order of the components), the following incremental weights are well defined:

$$\begin{aligned} G_t(x_{1:t}) &= \frac{\mathrm{d}[\tilde{\gamma}_t \otimes L^{\otimes}_{t-1:1}]}{\mathrm{d}[\tilde{\gamma}_{t-1} \otimes L^{\otimes}_{t-2:1} \otimes P_t]}(x_{1:t}) \\ &= \frac{\mathrm{d}[\tilde{\gamma}_t \otimes L_{t-1}]}{\mathrm{d}[\tilde{\gamma}_{t-1} \otimes P_t]}(x_{t-1}, x_t). \end{aligned} \tag{2.7}$$

Optimal Backward Kernels. Equation 2.7 shows that the efficiency of SMC samplers crucially depends on the choice of backward Markov kernels. As shown in Del Moral et al. (2006b), the *optimal backward kernel* in the sense of minimising the conditional variance of the incremental importance weights is given by

$$L^{\text{OPT}}_{t-1}(x_t, \mathrm{d}x_{t-1}) := \frac{\mathrm{d}P_t(x_{t-1}, \cdot)}{\mathrm{d}\tilde{\gamma}_{t-1}P_t}(x_t)\tilde{\gamma}_{t-1}(\mathrm{d}x_{t-1}). \tag{2.8}$$

That is, L^{OPT}_{t-1} is the reversal kernel associated with $\tilde{\gamma}_{t-1}$ and P_t. Employing this kernel effectively takes us back to performing IS on the marginal space, $\widetilde{\mathsf{X}}_t$, because it yields the following expression for the incremental importance weights:

$$\begin{aligned} G_t(x_{1:t}) &= \frac{\mathrm{d}[\tilde{\gamma}_t \otimes L^{\mathrm{OPT}}_{t-1}]}{\mathrm{d}[\tilde{\gamma}_{t-1} \otimes P_t]}(x_{t-1}, x_t) \\ &= \frac{\mathrm{d}[\tilde{\gamma}_t \otimes \tilde{\gamma}_{t-1}]}{\mathrm{d}[\tilde{\gamma}_{t-1} \otimes P_t]}(x_{t-1}, x_t) \frac{\mathrm{d}P_t(x_{t-1}, \cdot\,)}{\mathrm{d}\tilde{\gamma}_{t-1} P_t}(x_t) \\ &= \frac{\mathrm{d}\tilde{\gamma}_t}{\mathrm{d}\tilde{\gamma}_{t-1} P_t}(x_t), \end{aligned} \tag{2.9}$$

where we have used the definition of L^{OPT}_{t-1} in the second step; in the third step, we have replaced $\tilde{\gamma}_t \otimes \tilde{\gamma}_{t-1}$ by $\tilde{\gamma}_{t-1} \otimes \tilde{\gamma}_t$ by to correct for some abuse of the tensor-product notation regarding the order of the components.

In practice, the optimal backward kernel is usually intractable because evaluating (densities with respect to the) measure $\tilde{\gamma}_{t-1} P_t$ is infeasible. Instead, Del Moral et al. (2006b) stress that it is extremely important to select L_t to be some (tractable) approximation of L^{OPT}_t. A common choice, albeit one which introduces finite-sample bias, is to replace $\tilde{\gamma}_{t-1} P_t(\mathrm{d}x_t)$ by $\gamma^{\mathrm{SMC}, N_{1:t-1}}_{t-1}(\mathbb{1}_{\mathsf{X}^{\times}_{1:t-2}} \otimes P_t(\,\cdot\,, \mathrm{d}x_t))$ in the denominator in Equation 2.9. This backward kernel is used in *population Monte Carlo* methods (Cappé, Guillin, Marin & Robert, 2004), for instance.

2.15 Example (MCMC kernels). *The kernel P_t is often taken to be $\tilde{\gamma}_t$-invariant. The construction of such $\tilde{\gamma}_t$-invariant kernels, called MCMC kernels, is the subject of Chapter 3.*

If $\tilde{\gamma}_t$ is similar to $\tilde{\gamma}_{t-1}$ in some suitable sense, so that $\tilde{\gamma}_t$ is reasonably close to $\tilde{\gamma}_{t-1} P_t$, then Equation 2.8 suggests approximating $L^{\mathrm{OPT}}_{t-1}(x_t, \mathrm{d}x_{t-1})$ by the time-reversal kernel of $\tilde{\gamma}_t$ under P_t, given by

$$L_{t-1}(x_t, \mathrm{d}x_{t-1}) := \frac{\mathrm{d}P_t(x_{t-1}, \cdot\,)}{\mathrm{d}\tilde{\gamma}_t}(x_t) \tilde{\gamma}_t(\mathrm{d}x_{t-1}).$$

Using this backward kernel leads to the following simple expression for the incremental importance weights.

$$G_t(x_{1:t}) = \frac{\mathrm{d}\tilde{\gamma}_t}{\mathrm{d}\tilde{\gamma}_{t-1}}(x_{t-1}).$$

Note that the incremental importance weights at Step t do not depend on $\boldsymbol{Z}_t$. As mentioned in Example 2.5, it can therefore be beneficial to 'switch the order' of sampling and resampling.

Mixture Kernels. The SMC-sampler framework also allows the use of a mixture of forward kernels by including the index of the Step-t mixture component, M_t, into X_t and instead targeting an even further extended measure γ_t on $\mathsf{X}_{1:t}^{\times} := \prod_{s=1}^{t} \mathsf{X}_s$. Here, $\mathsf{X}_s := \widetilde{\mathsf{X}}_s \times \mathsf{M}_s$, where M_s is the countable set of all mixture component indices. Writing $X_s = (\widetilde{X}_s, M_s)$, the unnormalised version of this further extended distribution is

$$\gamma_t(\mathrm{d}x_{1:t}) := \tilde{\gamma}_t(\mathrm{d}\tilde{x}_t)\beta_0(\tilde{x}_1, \mathrm{d}m_1) \prod_{s=1}^{t-1} L_s(\tilde{x}_{s+1}, \mathrm{d}\tilde{x}_s \times \mathrm{d}m_{s+1}). \quad (2.10)$$

We need to employ backward mixture kernels

$$L_s(\tilde{x}_{s+1}, \mathrm{d}\tilde{x}_s \times \mathrm{d}m_{s+1}) = \beta_s(\tilde{x}_{s+1}, \mathrm{d}m_{s+1})\widetilde{L}_s((m_{s+1}, \tilde{x}_{s+1}), \mathrm{d}\tilde{x}_s),$$

if the (forward) proposal kernels are mixture kernels of the form

$$P_s(x_{s-1}, \mathrm{d}x_s) = \alpha_s(\tilde{x}_{s-1}, \mathrm{d}m_s)\widetilde{P}_s((m_s, \tilde{x}_{s-1}), \mathrm{d}\tilde{x}_s).$$

Here, the stochastic kernels $\alpha_s \in \mathcal{K}_1(\widetilde{\mathsf{X}}_{s-1}\mathsf{M}_s)$ and $\beta_{s-1} \in \mathcal{K}_1(\widetilde{\mathsf{X}}_s, \mathsf{M}_s)$ define forward and backward kernel mixture weights at Step s. At Step 1, it is common not to sample from a mixture so that $\beta_0(\mathrm{d}\tilde{x}_1, \mathrm{d}m_1) = \alpha_1(\mathrm{d}m_1) = \delta_m(\mathrm{d}m_1)$ and $\mathsf{M}_1 = \{m\}$. With the above-mentioned abuse of notation, the incremental importance weights then simplify to

$$G_t(x_{1:t}) = \frac{\mathrm{d}[\tilde{\gamma}_t \otimes \widetilde{L}_{t-1} \otimes \beta_{t-1}]}{\mathrm{d}[\tilde{\gamma}_{t-1} \otimes \alpha_t \otimes \widetilde{P}_t]}(x_{t-1}, x_t).$$

These are only well defined if $\tilde{\gamma}_t \otimes \widetilde{L}_{t-1} \otimes \beta_{t-1} \ll \tilde{\gamma}_{t-1} \otimes \alpha_t \otimes \widetilde{P}_t$. In particular, the backward mixture kernel weights β_s need to be chosen carefully if some forward mixture kernel components do not cover the entire support of $\tilde{\gamma}_s$. That is, if there exists a set A in the support of $\tilde{\gamma}_s$ and a mixture component index $m_s \in \mathsf{M}_s$ with $\tilde{\gamma}_{s-1}(\alpha_s(\,\cdot\,, \{m_s\})) > 0$ such that $\tilde{\gamma}_{s-1}(\widetilde{P}_s((m_s, \cdot\,), A)) = 0$. An algorithm for which this is a concern is considered in Chapter 4.

Examples. In this subsection, we have introduced SMC samplers as a special case of the SIR algorithm. In turn, many well-known SMC algorithms may be viewed as special cases of SMC samplers.

For instance, as pointed out in Del Moral et al. (2006b, 2007) and as already mentioned above, *population Monte Carlo* methods (Iba, 2000; Cappé et al., 2004) can be viewed as special cases of this framework. The same is true for *block sampling* (Doucet, Briers & Sénécal, 2006), for the *resample–move* algorithm Gilks and Berzuini (2001), and for various special cases of the latter such as the SMC algorithm for 'static' models (Chopin, 2002), and also SMC-squared (Chopin et al., 2013).

By working on the path space and employing forward and backward kernels that are mostly degenerate, we can actually also view any SIR algorithm as an SMC sampler. Though we will not pursue this interpretation further in this work.

2.3.3 Re-Using All Particles

Motivation. Assume that $\mathsf{X}_1 = \dots = \mathsf{X}_T = \mathsf{X}$. Usually, the SMC sampler from the previous subsection is run for $T \in \mathbb{N}$ steps and we are actually only interested in calculating integrals with respect to the final marginal measure, $\tilde{\gamma} := \tilde{\gamma}_T$, i.e. integrals of the form $\tilde{\gamma}_T(\tilde{f})$, where $\tilde{f} \colon \mathsf{X} \to \mathbb{R}$ is some integrable test function. The target measures $\tilde{\gamma}_1, \dots, \tilde{\gamma}_{T-1}$ are then only employed to interpolate between some easy-to-approximate initial target measure, $\tilde{\gamma}_1$, and the measure that is actually of interest, $\tilde{\gamma}$. Throughout this subsection, we assume that $\tilde{\gamma} \ll \tilde{\gamma}_t$, for any $t \in \mathsf{T}$. The standard SMC approximation of the integral $\tilde{\gamma}(\tilde{f})$ is only based on the particles generated at the Tth step. This is wasteful, especially if $\tilde{\gamma}_t$ is similar (in some suitable sense) to $\tilde{\gamma}$, for some $t < T$, as is usually the case in SMC samplers. It is wasteful because some of the remaining samples could be exploited to achieve variance reductions.

In this subsection, we describe a way of re-using all the particles to approximate integrals of the form $\tilde{\gamma}(\tilde{f})$. To that end, we slightly generalise the *importance tempering* approach from Gramacy, Samworth and King (2010). The resulting estimator is a (doubly) Rao–Blackwellised version of the estimator from Nguyen, Septier, Peters and Delignon (2014). Alternative approaches for combining IS estimates are discussed in Veach and Guibas (1995), Madras and Piccioni (1999), Owen and Zhou (2000).

Algorithm. Our proposed combination scheme is summarised in Algorithm 2.16. In the remainder of this subsection, we present formal (MOSIS-type) extended state-space justification of this scheme and comment further on its relationship with the approaches from and Gramacy et al. (2010) Nguyen et al. (2014).

2.16 Algorithm. *For each $t \in \mathsf{T}$, let $(X_t^n, w_t^n(\boldsymbol{Z}_{1:t}))_{n \in \mathsf{K}_t}$ be a set of (not necessarily IID) samples weighted to target the measure $\tilde{\gamma}_t$. Then, setting $\tilde{v}_t := \mathrm{d}\tilde{\gamma}/\mathrm{d}\tilde{\gamma}_t$, we may approximate $\tilde{\gamma}(f)$ by*

$$\sum_{t \in \mathsf{T}} \eta_T(\{t\}) \sum_{n \in \mathsf{K}_t} \tilde{v}_t(x_t^n) w_t^n(\boldsymbol{Z}_{1:t}) \tilde{f}(X_t^n),$$

where η_T is a probability measure on T such that $\eta_T(\{t\})$ is proportional to

$$\frac{N_t [\sum_{n \in \mathsf{K}_t} \tilde{v}_t(X_t^n) W_t^n(\boldsymbol{Z}_{1:t})]^2}{\sum_{m \in \mathsf{K}_t} \tilde{v}_t^2(X_t^m) W_t^n(\boldsymbol{Z}_{1:t})}.$$

Extended Target Measure. To justify our novel estimator, we devise a slightly different instance of the generic MOSIS target measure from Section 1.4 (compared to that of SMC methods). Writing $\bar{\boldsymbol{X}}'_T := (X, K, \boldsymbol{Z}_{1:T})$, where $X = (U, U_{1:\tau})$ and $K = (\tau, B_{1:\tau})$, this target measure is given by

$$\begin{aligned}\overline{\gamma}'_T(\mathrm{d}\bar{\boldsymbol{x}}'_T) := {} & \eta_T(\mathrm{d}\tau)\tilde{\gamma}(\mathrm{d}u)\delta_u(\mathrm{d}u_\tau) L^{\otimes}_{\tau-1:1}(u_\tau, \mathrm{d}u_{\tau-1} \times \cdots \times \mathrm{d}u_1) \\ & \times \overline{\Pi}_\tau^{\mathrm{CSMC}}(u_{1:\tau}, \mathrm{d}k \times \mathrm{d}\boldsymbol{z}_{1:T}) \Psi^{\otimes}_{\tau+1:T}(\boldsymbol{z}_{1:\tau}, \mathrm{d}\boldsymbol{z}_{\tau+1:T}),\end{aligned}$$

where $\eta_T \in \mathcal{M}_1(\mathsf{T})$ with $\mathsf{T} := \mathbb{N}_T$. The extended measure is defined on the space $\bar{\mathbf{X}}'_T := \mathsf{X} \times \mathsf{C} \times \mathbf{Z}^{\times}_{1:T}$, where $\mathsf{C} := \bigcup_{t \in \mathsf{T}}(\{t\} \times \mathsf{X}^t \times \mathsf{K}^{\times}_{1:t})$.

In other words, to sample from the normalised version of this extended measure, we would first sample the random variable τ which indexes a particular generation (or step) of the SMC algorithm. We then sample a single particle $U_\tau = U$ from the actual target distribution, $\tilde{\gamma}/\tilde{\gamma}(\mathbb{1})$, sample $U_{1:\tau-1}$ according to the backward kernels and draw $\bar{\boldsymbol{Z}}_\tau = (K, \boldsymbol{Z}_{1:\tau})$ from the standard CSMC kernel (up to Step τ). We then sample additional variables $\boldsymbol{Z}_{\tau+1:T}$ via Steps $\tau + 1$ to T of the SMC sampler. Note that under the normalised version of this extended measure, $X_\tau^{B_\tau} \sim \tilde{\gamma}/\tilde{\gamma}(\mathbb{1})$.

Extended Proposal Distribution. The associated extended proposal distribution can be defined as

$$\bar{\psi}'_T(\mathrm{d}\bar{\boldsymbol{x}}'_T) := \psi_T(\mathrm{d}\boldsymbol{z}_{1:T}) \,\mathrm{Unif}_{\mathsf{T}}(\mathrm{d}\tau) \Xi_\tau(\boldsymbol{z}_{1:\tau}, \mathrm{d}u_{1:\tau} \times \mathrm{d}b_{1:\tau}) \delta_{u_\tau}(\mathrm{d}u),$$

where Ξ_t is as defined in Equation 2.2.

Importance Weights. Having defined the Radon–Nikodým derivatives $\bar{w}'_T := \mathrm{d}\bar{\gamma}'_T / \mathrm{d}\bar{\psi}'_T$ and $\tilde{v}_t := \mathrm{d}\tilde{\gamma}/\mathrm{d}\tilde{\gamma}_t$, we can calculate 'weights'

$$\begin{aligned} w_T^{t,n}(\boldsymbol{Z}_{1:T}) &:= \mathbb{E}[\bar{w}'_T(\bar{\boldsymbol{X}}'_T)\,\mathbb{1}_{\{(t,n)\}}(\tau, B_\tau)|\boldsymbol{Z}_{1:T}] \\ &= \eta_T(\{t\})\tilde{v}_t(x_t^n)w_t^n(\boldsymbol{Z}_{1:t}), \end{aligned}$$

where $w_t^n(\boldsymbol{Z}_{1:t})$ is the usual nth Step-t weight from the SMC algorithm. A one-sample IS approximation of the extended target measure is given by $\bar{\gamma}_T'^{\text{IS},1} := \bar{w}'_T(\bar{\boldsymbol{X}}'_T)\delta_{\bar{\boldsymbol{X}}'_T}$, where $\bar{\boldsymbol{X}}'_T = (U, U_{1:\tau}, \tau, B_{1:\tau}, \boldsymbol{Z}_{1:T}) \sim \bar{\psi}'_T$.

One-Sample IS Interpretation. Let $A \in \mathcal{B}(\mathsf{X})$. With a Rao–Blackwellisation as in Section 1.4, an approximation of $\tilde{\gamma}(A)$ which makes use of all particles generated over the course of the SMC sampler is then given by

$$\begin{aligned} \tilde{\gamma}^{\text{MOSIS},N_{1:T}}(A) &:= \mathbb{E}[\bar{\gamma}_T'^{\text{IS},1}(A \times \mathsf{C} \times \mathsf{Z}_{1:T}^{\times})|\boldsymbol{Z}_{1:T}] \\ &= \sum_{t \in \mathsf{T}} \sum_{k_t \in \mathsf{K}_t} w_T^{t,k_t}(\boldsymbol{Z}_{1:T})\delta_{X_t^{k_t}}(A) \\ &= \sum_{t \in \mathsf{T}} \eta_T(\{t\})\tilde{\gamma}_t^\star(A), \end{aligned} \tag{2.11}$$

where we have defined the following unbiased estimate of $\tilde{\gamma}(A)$, which is based on the particles generated at the tth step of the SMC sampler:

$$\tilde{\gamma}_t^\star(A) := \sum_{n \in \mathsf{K}_t} \tilde{v}_t(x_t^n)w_t^n(\boldsymbol{Z}_{1:t})\delta_{X_t^n}(A). \tag{2.12}$$

The approximation from Equation 2.12 is a Rao–Blackwellised version of the following estimator used by Nguyen et al. (2014) within the estimator $\tilde{\gamma}^{\text{MOSIS},N_{1:T}}(A) = \sum_{t \in \mathsf{T}} \eta_T(\{t\})\tilde{\gamma}_t^\star(A)$:

$$\tilde{\gamma}_t^\star(A) := \sum_{m \in \mathsf{M}_t} \tilde{v}_t(\widehat{X}_t^m)\biggl[\sum_{n \in \mathsf{K}_t} w_t^n(\boldsymbol{Z}_{1:t})\biggr]\delta_{\widehat{X}_t^m}(A),$$

Here, $\mathsf{M}_t := \mathbb{N}_{M_t}$ and $\widehat{X}_t^{1:M_t}$ is obtained by choosing M_t elements from among $X_t^{1:N_t}$ via multinomial resampling according to the standard self-normalised SMC weights $(W_t^n(\boldsymbol{Z}_{1:t}))_{n \in \mathsf{K}_t}$. In other words, Nguyen et al. (2014) first resample the particles from each step of the SMC sampler so that $\widehat{X}_t^{1:M_t}$ is a set of unweighted particles whose empirical measure approximates $\tilde{\pi}_t \propto \tilde{\gamma}_t$. They then weight these (now unweighted) particles so that their weighted empirical measure approximates $\tilde{\pi} \propto \tilde{\gamma}$.

Approximately Optimal Weighting Scheme. Note that the choice of the distribution η_T is critical in both estimators. On the one hand, taking $\eta = \delta_T$ takes us back to only using the Step-T particles to approximate $\tilde{\gamma}(\tilde{f})$. On the other hand, taking $\eta = \mathrm{Unif}_{\mathsf{T}}$ assigns equal weights to the individual estimators $\tilde{\gamma}_t^\star(\tilde{f})$ associated with different SMC steps. As pointed out in Gramacy et al. (2010), this usually leads to a high variance of the estimator from Equation 2.11 because $\tilde{\gamma}_1^\star(\tilde{f}), \dots, \tilde{\gamma}_T^\star(\tilde{f})$ have different variances. Ideally, we would weight each of these estimators inversely proportional to its variance, i.e. we would like to set

$$\eta_T(\{t\}) = \frac{1/\mathbb{V}[\tilde{\gamma}_t^\star(\tilde{f})]}{\sum_{s \in \mathsf{T}} 1/\mathbb{V}[\tilde{\gamma}_s^\star(\tilde{f})]}. \tag{2.13}$$

Unfortunately, these variances are usually intractable and furthermore, we are often interested in a large class of test functions.

Instead, we generalise the approach from Gramacy et al. (2010) and take $\eta_T := \varepsilon / \sum_{s \in \mathsf{T}} \varepsilon(\{s\})$ where $\varepsilon(\{t\})$ is the following proxy for the tth inverse variance in Equation 2.13: for $\bar{\boldsymbol{X}}_T' = (U, U_{1:\tau}, \tau, B_{1:\tau}, \boldsymbol{Z}_{1:T}) \sim \bar{\psi}_T'$,

$$\begin{aligned}\varepsilon(\{t\}) &:= \frac{N_t(\mathbb{E}[\tilde{v}_\tau(U) | \boldsymbol{Z}_{1:T}, \tau = t])^2}{\mathbb{E}[\tilde{v}_\tau^2(U) | \boldsymbol{Z}_{1:T}, \tau = t]} \\ &= \frac{N_t[\sum_{n \in \mathsf{K}_t} \tilde{v}_t(X_t^n) W_t^n(\boldsymbol{Z}_{1:t})]^2}{\sum_{m \in \mathsf{K}_t} \tilde{v}_t^2(X_t^m) W_t^n(\boldsymbol{Z}_{1:t})}.\end{aligned} \tag{2.14}$$

This is an approximation of $ESS_t := N_t \tilde{\mathfrak{z}}_t^2 / \tilde{\pi}_t(\tilde{v}_t^2)$ and can be seen as the *conditional* ESS associated with the self-normalised version of $\tilde{\gamma}_t^\star$. Here, we have set $\tilde{\mathfrak{z}}_t := \mathfrak{z}_T / \mathfrak{z}_t$. The conditional ESS was introduced in Yan Zhou et al. (2013) (albeit in a different context). Note that using the same set of samples to construct the estimators $\tilde{\gamma}_t^\star(\tilde{f})$ and to determine ε introduces a slight bias. Of course, if it is necessary to obtain unbiased estimates, this particular bias can be avoided by using two different sets of samples, i.e. by basing the construction of ε on samples generated in some pilot run.

Note that Equation 2.14 does not take into account the particular form of the test function. This is sensible whenever (1) we are interested in a large class of test functions and want to avoid the cost of re-calculating η_T for each test function, (2) the oscillations of $\tilde{f}$ have little effect on the integral $\tilde{\gamma}(\tilde{f})$. However, in some instances, it might be desirable to take the form of a particular test function into account. For instance, this is the case when performing rare-event estimation, i.e. when $\tilde{f} := \mathbb{1}_A$, where $A \in \mathscr{B}(\mathsf{X})$ is such that $\tilde{\gamma}(A)$ is very small compared to $\tilde{\gamma}(\mathsf{X})$. In this case, we can easily incorporate the test function into the above framework if we replace the pair $(\tilde{\gamma}, \tilde{f})$ by the pair $(\tilde{f}\tilde{\gamma}, \mathbb{1})$.

Finally, we note that we have presented our approach in the context of SMC samplers. However, it can be used in any other case in which we have sets of samples $X_t^{1:N_t}$ weighted to target a distribution $\tilde{\pi}_t$, for $t \in \mathsf{T}$.

Relationship With Previous Approaches. We conclude this subsection by summarising the relationship between our estimator and those in Gramacy et al. (2010) and Nguyen et al. (2014).

Our estimator can be seen as a generalisation of the approach from Gramacy et al. (2010). Indeed, our estimator reduces to that of Gramacy et al. (2010) in the case that $w_t^n(\boldsymbol{Z}_{1:t}) = w_t^m(\boldsymbol{Z}_{1:t}) = 1/N_t$, for any $(n, m) \in \mathsf{K}_t^2$ and any $t \in \mathsf{T}$ and $n_t \in \mathsf{K}_t$. This special case occurs when the 'particles' $X_t^{1:N_t}$ are unweighted draws from the distribution proportional to $\tilde{\gamma}_t$. For instance, in Gramacy et al. (2010), all the samples are unweighted because they are obtained from simulated tempering so that $X_t^{1:N_t}$ represent the samples associated with the tth temperature.

The approach from Nguyen et al. (2014) obtains such unweighted draws via the additional resampling step described above. Our construction shows that this extra resampling step is not necessary – neither for constructing the estimators $\tilde{\gamma}_t^\star(A)$ nor for designing ε (and hence η_T). Indeed, the extra resampling step introduces additional Monte Carlo variance and increases the computational cost. This insight is in the spirit of Johansen and Doucet (2008) who showed the redundancy of the second resampling step in the auxiliary particle filter from Pitt and Shephard (1999).

Indeed, our choice of the (random) measure ε in Equation 2.14 is obtained by Rao–Blackwellising the numerator and denominator of the following random measure employed by Nguyen et al. (2014):

$$\varepsilon(\{t\}) := \frac{[\sum_{n \in \mathsf{M}_t} \tilde{v}_t(\widehat{X}_t^n)]^2}{\sum_{m \in \mathsf{M}_t} \tilde{v}_t^2(\widehat{X}_t^m)}.$$

Hence, our approximation can be interpreted as a twice Rao–Blackwellised version of the estimator from Nguyen et al. (2014).

2.3.4 Discrete Particle Filter

If the number of elements in X_t is finite (and sufficiently small), SIR algorithms are wasteful because there is usually a positive probability that two available particle paths at Step t are identical, i.e. for any $m, n \in \mathsf{K}_t$,

$$\mathbb{P}\bigl[\bigl\{X_{1:t}^{B_{1:t|t}^n} = X_{1:t}^{B_{1:t|t}^m}\bigr\}\bigr] > 0. \tag{2.15}$$

The *discrete particle filter* (DPF) introduced by (Fearnhead, 1998; Fearnhead & Clifford, 2003) tackles this problem by propagating particles in a way that reduces the probability on the left hand side in Equation 2.15 to zero. This is done by extending each available particle trajectory once in every possible direction at Step t. To keep the computational cost from growing exponentially in t, the resulting trajectories are then stochastically pruned in a way that is optimal in the sense that it minimises the variance of the sum of the self-normalised importance weights.

In this subsection, we show that by using particular choices of the kernels S_{t-1}, R_{t-1}, Q_t, and κ_t, the DPF can be viewed as a special case of the generic SMC algorithm. To our knowledge, this is a new result. It immediately implies the validity of CSMC algorithms, backward sampling, or ancestor sampling (see Section 3.4 in the next chapter), as described in Whiteley, Andrieu and Doucet (2010), for the DPF. However, the DPF cannot be viewed as a special case of SIR due to the dependence in the proposal kernels and the use of a biased resampling scheme. Here, we recall that in the terminology of Definition 2.3, a resampling scheme is termed 'biased' if it does not lead to an evenly weighted (i.e. unweighted) set of particles after resampling. We reiterate that any estimates of integrals of the form $\gamma_t(f_t)$ will still be unbiased as long as the resampled particles are suitably weighted.

Without loss of generality, assume a finite state space $\mathsf{X}_t = \mathbb{N}_K$, for any $t \in \mathsf{T}$ for some (usually not too large) $K \in \mathbb{N}$. At the tth step of the algorithm, we have $N_t := MK \wedge K^t$ particles, where $M \in \mathbb{N}$ can be chosen to control the computational cost of the algorithm. As described below, at Step t, we select $M_t := N_t/K$ particle trajectories from the previous step and extend each of them in all K possible directions.

Resampling Scheme. In this case, we do not make use of the auxiliary variables O_{s-1} and therefore drop them from the notation along with the kernels S_{s-1}. The kernel $R_{s-1} = \widetilde{R}_{s-1}$ is then given by

$$\begin{aligned}
&\widetilde{R}_{s-1}(\boldsymbol{z}_{1:s-1}, \mathrm{d}\boldsymbol{a}_{s-1}) \\
&\qquad = \widetilde{R}^{\star}_{s-1}(\boldsymbol{z}_{1:s-1}, \mathrm{d}a^{1:M_s}_{s-1}) \prod_{n=2}^{K} \delta_{a^{1:M_s}_{s-1}}(\mathrm{d}a^{(n-1)M_s+1:nM_s}_{s-1}),
\end{aligned}$$

where $\widetilde{R}^{\star}_{s-1}(\boldsymbol{z}_{1:s-1}, \mathrm{d}a^{1:M_s}_{s-1})$ denotes the resampling scheme (for M_s offspring) developed in Fearnhead (1998) which summarised in the following. A more formal description of the entire kernel $\widetilde{R}_{s-1}$ can be found in Section A.5 of the appendix, for completeness.

At Step s, we use Fearnhead (1998, Algorithm 5.2) to solve

$$\sum_{n=1}^{N_{s-1}} \bigl[1 \wedge C_{s-1} W^n_{s-1}(\boldsymbol{z}_{1:s-1})\bigr] = M_s,$$

for $C_{s-1} > 0$. The idea is that particles whose self-normalised weights exceed the threshold $1/C_{s-1}$ get exactly one offspring. The remaining particles have at most one offspring.

Collect the indices of the former particles in the set

$$L_s := \#\bigl\{n \in \mathsf{K}_{s-1} \bigm| W^n_{s-1}(\boldsymbol{z}_{1:s-1}) > 1/C_{s-1}\bigr\}$$

and let $l_s \colon \{1, \dots, \#L_s\} \to L_s$ be the function which maps n to the nth largest element in L_s. We then set the first $\#L_s$ parent indices deterministically via $A^{1:\#L_s}_{s-1} := (l_s(1), \dots, l_s(\#L_s))$. The remaining $M_s - \#L_s$ parent indices take values in $\mathsf{K}_{s-1} \setminus L_s$. They are generated using systematic resampling based on the weights $(W^n_{s-1}(\boldsymbol{z}_{1:s-1}))_{n \in \mathsf{K}_{s-1} \setminus L_s}$, after these have been re-normalised to sum to 1.

Note that $M_s = N_{s-1}$ implies $C_{s-1} \geq 1/[\max_{n \in \mathsf{K}_{s-1}} W^n_{s-1}(\boldsymbol{z}_{1:s-1})]$ and thus $L_s = \mathbb{N}_{M_s}$, i.e. in this case, we propagate all existing particle paths without any pruning.

Proposal Kernel. The proposal kernels are completely deterministic, i.e. $q_1(\mathrm{d}\boldsymbol{x}_1) = \delta_{(1,\dots,K)}(\mathrm{d}\boldsymbol{x}_1)$, and

$$Q_s((\boldsymbol{z}_{1:s-1}, \boldsymbol{a}_{s-1}), \mathrm{d}\boldsymbol{x}_s) = \delta_{(1\iota_{M_s}, 2\iota_{M_s}, \dots, K\iota_{M_s})}(\mathrm{d}\boldsymbol{x}_s),$$

where ι_m denotes an m-component vector of 1s. In other words, each of the M_s particle trajectories chosen as parents by the resampling distribution has exactly K offspring – one for each element of X_s.

Importance Weights. To ensure that the Radon–Nikodým derivative $\overline{w}_t$ exists, let $\kappa_1(u_1, \cdot) := \delta_{u_1}$ and, for $s > 1$, let $\kappa_s(u_{1:s}, \cdot)$ be the uniform distribution on $\mathbb{Z}_{(u_s-1)M_s+1, u_s M_s} =: D_s^{u_s}$. If $\varLambda_s$ is the time-reversal kernel from Assumption 2.9, then by the properties of the resampling scheme employed here (see Section A.5 of the appendix),

$$\begin{aligned}
&\varLambda_s((u_{1:s}, \boldsymbol{z}_{1:s-1}, b_{s-1}), \{b_s\})\delta_{b_{s-1}}(\{a_{s-1}^{b_s}\}) \\
&\quad = \frac{R_{s-1}^{\mathrm{M}}((\boldsymbol{z}_{1:s-1}, b_s), \{a_{s-1}^{b_s}\})}{1 \wedge C_{s-1} W_{s-1}^{b_{s-1}}(\boldsymbol{z}_{1:s-1})} \mathbb{1}_{\{a_{s-1}^{b_s}\}}(b_{s-1}) \, \mathbb{1}_{D_s^{u_s}}(b_s).
\end{aligned}$$

Hence, the nth Step-t particle weight, $w_t^n(\boldsymbol{z}_{1:t})$, can be written as

$$\begin{aligned}
w_t^n(\boldsymbol{z}_{1:t}) &= \frac{\gamma_t(\{x_{1:t}^{b_{1:t|t}^n}\})}{\prod_{s=2}^t [1 \wedge C_{s-1} W_{s-1}^{b_{s-1|t}^n}(\boldsymbol{z}_{1:s-1})]} \\
&= w_{t-1}^{b_{t-1|t}^n}(\boldsymbol{z}_{1:t-1}) \frac{\varGamma_t(x_{1:t-1}^{b_{1:t-1|t}^n}, \{x_t^n\})}{1 \wedge C_{t-1} W_{t-1}^{b_{t-1|t}^n}(\boldsymbol{z}_{1:t-1})},
\end{aligned}$$

for any $\boldsymbol{z}_{1:t}$ in the support of $\tilde{\psi}_t$ and $w_t^n(\boldsymbol{z}_{1:t}) = 0$, otherwise.

2.3.5 Other SMC Algorithms

In this subsection, we briefly mention how other well-known SMC algorithms fit into the framework developed in this chapter.

Look-Ahead Algorithms. As pointed out in Heine (2005), Johansen and Doucet (2008), *auxiliary particle filters* (Pitt & Shephard, 1999) can be viewed as a special case of the SIR algorithm targeting a slightly altered sequence of measures. *Block sampling* (Doucet et al., 2006), *piloting* (Wang et al., 2002; Zhang & Liu, 2002) and the other look-ahead strategies surveyed in Lin et al. (2013) can also be seen as standard SIR algorithms on suitably extended spaces. *Twisted particle filters* (Whiteley & Lee, 2014) propose one particle at each step from a different proposal kernel than the others. They can therefore not be seen as SIR algorithms but remain a special case of the generic SMC framework introduced in this chapter.

Hierarchical Algorithms. The framework developed in this chapter can also be applied hierarchically. That is, the target measure γ_t can itself be an extended target measure of a 'lower-level' SMC algorithm (i.e. it is of the form given in Equation 2.3).

Using this construction, we can justify a wide range of hierarchical SMC algorithms such as *exactly-approximated Rao–Blackwellised particle filters* (Johansen et al., 2012), *island particle filters* (Vergé et al., 2013), *SMC-squared* (Chopin et al., 2013) and other interacting SMC algorithms by Jasra et al. (2008), Beskos, Crisan, Jasra, Kamatani and Zhou (2014). Finally, Johansen and Doucet (in prep.) use SMC algorithms at the lower level to mimic the behaviour of the intractable optimal block sampling strategy from Doucet et al. (2006).

These algorithms are also *pseudo-marginal* SMC algorithms because in the sense of Subsection 1.4.3, they (exactly) approximate a usually intractable marginal SMC algorithm. On the marginal space, their behaviour approaches that of the marginal algorithm as the number of particles in the lower-level SMC algorithm tends to infinity. More generally, at the lower level, we may target the extended measure associated with *any* MOSIS scheme. This justifies other pseudo-marginal SMC algorithms such as *random-weight* particle filters (Fearnhead et al., 2010). Indeed, the IS-squared approach later developed by Tran et al. (2014) is a simple special case this hierarchical idea. Alternatively, it may also be justified as a simple case of (random-weight) IS.

2.4 Sample Impoverishment and Remedies

2.4.1 Particle-Path Coalescence

In this section, we discuss the sample-impoverishment 'problem' and a partial remedy in the form of forward filtering–backward smoothing as well as the sampling-approximation of the latter known as forward filtering–backward sampling.

As noted in Subsection 2.2.5, SMC methods are particularly suited (often *only* suited) to approximating integrals with respect to the final marginal of π_t, i.e. integrals with respect to $\pi_{t,t}$ given by $\pi_{t,t}(A) := \pi_t(\mathsf{X}_{1:t-1}^{\times} \times A)$, for $A \in \mathcal{B}(\mathsf{X}_t)$. Fundamentally, this is due to the fact that for sufficiently ergodic models, such estimators behave approximately as IS estimators on a much smaller space.

However, this does not hold for more general test functions, i.e. test functions which are not constant on $\mathsf{X}^{\times}_{1:t-1}$. For the latter, the error associated with SMC estimates tends to grow in t. This is unsurprising because we are estimating integrals with respect to π_t and are thus performing IS on an ever increasing space.

In practice, this manifests itself in the *sample-impoverishment* problem, i.e. in the fact that all particle trajectories share a common ancestor if t is sufficiently large. More formally, let

$$\tau_t := \sup\{ s \in \mathbb{N}_t \mid B^1_{1:s|t} = \cdots = B^{N_t}_{1:s|t} \}$$

be the step number associated with the most recent common ancestor of all Step-t particles. Jacob, Murray and Rubenthaler (2013) show that the expected 'time' to the most recent common ancestor satisfies

$$t - \mathbb{E}[\tau_t] = \mathcal{O}(N \log(N)),$$

assuming that $N_1 = N_2 = \ldots = N$. However, as stressed by Doucet and Johansen (2011), while this coalescence is a result of performing resampling within an SMC algorithm, resampling is not fundamentally causing the error associated with SMC estimates to grow with t (for general test functions). Rather, the need for resampling is merely another symptom of the fact that we are performing IS on an ever increasing space. In fact, resampling is actually the crucial ingredient for guaranteeing uniform-in-t error bounds and other stability properties of estimates of integrals with respect to $\pi_{t,t}$.

2.4.2 Forward Filtering–Backward Smoothing

In order to still obtain estimates of path integrals $\pi_t(f_t)$ (i.e. for more general test functions), it has been suggested to make use of all the particles generated throughout the SMC algorithm (and not just those that form part of the N_t particle lineages at Step t. This procedure, known as *forward filtering–backward smoothing* (FFBS), leads to a different kind of MOSIS approximation based on a further extended measure which we describe in this subsection.

Up to now, the MOSIS interpretation of SMC methods was based around effectively taking the index K to be one of the particle lineages, i.e. loosely speaking, we have $K = B^n_{1:t|t}$, for some $n \in \mathsf{K}_t$. The aim of the further extended measure constructed in this subsection is to let $K = C_{1:t}$, where $C_{1:t}$ are particle indices which need not coincide with any particle lineage generated under the SMC algorithm. A normalised version of this measure is also at the heart of iterated conditional sequential Monte Carlo algorithms which use backward sampling or ancestor sampling and which are discussed in Section 3.4.

Backward Sampling Weights. Hereafter, we assume some fixed 'time' horizon, $T \in \mathbb{N}$ and set $\mathsf{T} := \mathbb{N}_T$. Before defining the further extended target measure, we need some additional notation.

For $t \in \mathsf{T}$, define the following kernels in $\mathcal{K}(\mathbf{Z}^\times_{1:t} \times \mathsf{X}^\times_{t+1:T}, \mathsf{K}_t)$, termed *backward sampling weights*,

$$w^k_{t|T}(z_{1:t}, v_{t+1:T}) := w^k_t(z_{1:t}) \frac{\mathrm{d}\Gamma^{\otimes}_{t+1:T}(x_{1:t}^{b^k_{1:t|t}}, \cdot)}{\mathrm{d}\lambda^{\otimes}_{t+1:T}(x_{1:t}^{b^k_{1:t|t}}, \cdot)}(v_{t+1:T}),$$

for $k \in \mathsf{K}_t$, where the kernels $\lambda_t \in \mathcal{K}_\sigma(\mathsf{X}^\times_{1:t-1}\mathsf{X}_t)$ define some suitable dominating measure. We also define the self-normalised versions

$$W^k_{t|T}(z_{1:t}, v_{t+1:T}) := \frac{w^k_{t|T}(z_{1:t}, v_{t+1:T})}{\sum_{n=1}^{N_t} w^n_{t|T}(z_{1:t}, v_{t+1:T})}.$$

Note that $w^k_{T|T}(z_{1:T}) = w^k_T(z_{1:T})$.

Further Extended Target Measure. We now construct the further extended target measure, $\tilde{\gamma}_T$. To that end, we include an additional set of particles, $V_{1:T}$, and particle indices, $C_{1:T}$, into the state space. Both of these will be such that $V_{1:T}$ coincides with the particles with indices $C_{1:T}$ generated under the SMC algorithm, i.e. $V_{1:T} = X^{C_{1:T}}_{1:T}$. However, we may have $C_{1:T} \neq B^n_{1:T|T}$, for all $n \in \mathsf{K}_T$.

Write $\widetilde{X}_T := (\overline{X}_T, V_{1:T}, C_{1:T})$, then the further extended measure on the extended space $\widetilde{\mathbf{X}}_T := \overline{\mathbf{X}}_T \times \mathsf{X}^\times_{1:T} \times \mathsf{K}^\times_{1:T}$ is defined as

$$\tilde{\gamma}_T(\mathrm{d}\tilde{x}_T) := \bar{\gamma}_T(\mathrm{d}\bar{x}_T) \varXi^{\mathrm{BS}}_T(z_{1:T}, \mathrm{d}v_{1:T} \times \mathrm{d}c_{1:T}),$$

where $\Xi_T^{\text{BS}} \in \mathcal{K}_1(\boldsymbol{Z}_{1:T}^{\times}, \mathsf{X}_{1:T}^{\times} \times \mathsf{K}_{1:T}^{\times})$ is a stochastic kernel given by

$$\begin{aligned}
&\Xi_T^{\text{BS}}(\boldsymbol{z}_{1:T}, \mathrm{d}v_{1:T} \times \mathrm{d}c_{1:T}) \\
&\qquad := \prod_{t=1}^{T} \xi_{t|T}((\boldsymbol{z}_{1:t}, v_{t+1:T}, c_{t+1:T}), \mathrm{d}c_t)\delta_{x_t^{c_t}}(\mathrm{d}v_t).
\end{aligned}$$

For $t < T$, we have defined the *backward sampling/smoothing kernels*

$$\begin{aligned}
&\xi_{t|T}((\boldsymbol{z}_{1:t}, v_{t+1:T}, c_{t+1:T}), \mathrm{d}c_t) \\
&\qquad := \begin{cases} \delta_{a_t^{c_{t+1}}}(\mathrm{d}c_t), & \text{if } \varrho_t(o_t) = 0, \\ W_{t|T}^{c_t}(\boldsymbol{z}_{1:t}, v_{t+1:T}), & \text{if } \varrho_t(o_t) = 1, \end{cases}
\end{aligned}$$

Here, $\varrho_t \colon \mathsf{O}_t \to \{0, 1\}$ are suitable functions for interpolating between 'full' backward smoothing ($\varrho_1 = \cdots = \varrho_T \equiv 1$) and no backward smoothing ($\varrho_1 = \cdots = \varrho_T \equiv 0$). This may be desirable for reducing the computational cost of calculating the backward smoothing weights.

Further Extended Proposal Distribution. To obtain a further extended proposal distribution, $\tilde{\psi}_T$, with respect to which $\tilde{\gamma}_T$ is absolutely continuous, we simply extend the usual SMC proposal distribution by the same kernel, i.e.

$$\tilde{\psi}_T := \bar{\psi}_T \otimes \Xi_T^{\text{BS}}.$$

Importance Weights. Since Ξ_T^{BS} extends both the target measure and proposal distribution, we have $\widetilde{w}_T(\tilde{\boldsymbol{x}}_T) = [\mathrm{d}\tilde{\gamma}_T / \mathrm{d}\tilde{\psi}_T](\tilde{\boldsymbol{x}}_T) = \overline{w}_T(\bar{\boldsymbol{x}}_T)$. Assuming that $\xi_{T|T}(\boldsymbol{z}_{1:T}, \{k\}) = W_T^k(\boldsymbol{z}_{1:T})$, Proposition 2.10 then implies the following weight for the particles indexed by $c_{1:T}$,

$$\begin{aligned}
\widetilde{w}_T^{c_{1:T}}(\boldsymbol{Z}_{1:T}) &:= \mathbb{E}\big[\widetilde{w}_T(\widetilde{\boldsymbol{X}}_T)\, \mathbb{1}_{\{c_{1:T}\}}(C_{1:T}) \big| \boldsymbol{Z}_{1:T}\big] \\
&= \Xi_T^{\text{BS}}(\boldsymbol{Z}_{1:T}, \mathsf{X}_{1:T}^{\times} \times \{c_{1:T}\}) \sum_{n=1}^{N_t} w_T^n(\boldsymbol{Z}_{1:T}). \qquad (2.16)
\end{aligned}$$

Rao–Blackwellisation. Let $\widetilde{\boldsymbol{X}}_T = (\overline{\boldsymbol{X}}_T, V_{1:T}, C_{1:T}) \sim \tilde{\psi}_T$, then a one-sample standard IS approximation of $\tilde{\gamma}_T$ is given by $\tilde{\gamma}_T^{\text{IS},1} := \widetilde{w}_T(\widetilde{\boldsymbol{X}}_T)\delta_{\widetilde{\boldsymbol{X}}_T}$.

The FFBS approximation of γ_T is then obtained in the usual manner by integrating out the indices $K = C_{1:T}$ and the set of particles $X = V_{1:T}$

along with the quantities $(U_{1:T}, B_{1:T})$. In contrast to the standard MOSIS interpretation of SMC methods, the latter quantities are now considered to be auxiliary variables which are part of Y, along with the parent indices $\boldsymbol{A}_{1:T-1}$ and $O_{1:T}$. For any $A \in \mathcal{B}(\mathsf{X}^{\times}_{1:T})$, by Equation 2.16,

$$\begin{aligned}\gamma_T^{\text{MOSIS},N_{1:T}}(A) &= \mathbb{E}\big[\tilde{\gamma}_T^{\text{IS},1}(\overline{\mathbf{X}}_T \times A \times \mathsf{K}^{\times}_{1:T})\big|\boldsymbol{Z}_{1:T}\big]\\ &= \sum_{c_{1:T} \in \mathsf{K}^{\times}_{1:T}} \widetilde{w}_T^{c_{1:T}}(\boldsymbol{Z}_{1:T})\delta_{X_{1:T}^{c_{1:T}}}(A)\\ &=: \gamma_T^{\text{FFBS},N_{1:T}}(A).\end{aligned}$$

Note that we have not yet shown that $\tilde{\gamma}_T$ admits γ_T as a marginal in the $v_{1:T}$-component, i.e. that

$$\tilde{\gamma}_T(\overline{\mathbf{X}}_T \times A \times \mathsf{K}^{\times}_{1:T}) = \gamma_T(A), \tag{2.17}$$

for any $A \in \mathcal{B}(\mathsf{X}^{\times}_{1:T})$. Hence, it is not obvious that the FFBS approximation $\gamma_T^{\text{FFBS},N_{1:T}}(f_T)$ should be an unbiased estimate of $\gamma_T(f_T)$. We postpone the proof of Equation 2.17 until Subsection 3.4.3. As shown therein, Equation 2.17 is essentially equivalent to showing that conditional sequential Monte Carlo kernels with backward sampling and ancestor sampling both share the same extended target distribution.

Finally, note that by Equation 2.16, FFBS cannot lead to any improvement over the standard SMC approximation if f_T is constant because

$$\mathfrak{z}_T^{\text{FFBS},N_{1:T}} := \sum_{c_{1:T} \in \mathsf{K}^{\times}_{1:T}} \widetilde{w}_T^{c_{1:T}}(z_{1:T}) = \sum_{n=1}^{N_T} w_T^n(z_{1:T}) = \mathfrak{z}_T^{\text{SMC},N_{1:T}}.$$

Thus, in particular, FFBS does not improve estimates of the normalising constant compared to the usual SMC estimate.

Additive Functionals. Using the FFBS approximation tends to be infeasible, in practice, because the cost of evaluating $\widetilde{w}_T^{c_{1:T}}(z_{1:T})$ is $\mathcal{O}(N^T)$ if we assume, for simplicity, that $N_1 = N_2 = \ldots = N_T = N$. However, the computational cost can be brought down to $\mathcal{O}(TN^2)$ via standard forward–backward recursions (Rauch, Striebel & Tung, 1965; Baum, Petrie, Soules & Weiss, 1970) if the test function admits the following additive decomposition

$$f_T(x_{1:T}) = f_{1,T}(x_1) + \sum_{t=2}^{T} f_{t,T}(x_{t-1}, x_t), \tag{2.18}$$

where $f_{t,T} \colon \mathsf{X}^{\times}_{t-1:t} \to \mathbb{R}$ are suitable functions. In particular, the FFBS approximation of the integral $\gamma_T(f_T)$ can then be calculated *without* an explicit backward recursion. This is known as *forward smoothing* (Del Moral, Doucet & Singh, 2010).

2.4.3 Forward Filtering–Backward Sampling

Instead of analytically integrating out $(V_{1:T}, C_{1;T})$, Doucet, Godsill and West (2000), Godsill, Doucet and West (2004) propose to sample M 'backward trajectories' $(V_{1:T}^m, C_{1:T}^m)_{m \in \mathbb{N}_M}$ (conditionally) independently from the distribution $\Xi_T^{\mathrm{BS}}(\boldsymbol{Z}_{1:T}, \cdot)$, given the variables $\boldsymbol{Z}_{1:T} \sim \psi_T$ which have been generated by a single run of the SMC algorithm. This has been termed *forward filtering–backward sampling*.

The backward-sampling idea represents a simple sampling-based approximation of FFBS (conditional on $\boldsymbol{Z}_{1:T}$) and compared to the latter, it reduces the computational cost to the more manageable $\mathcal{O}(MTN)$. This rate can formally be further improved by rejection-sampling ideas (Douc, Garivier, Moulines & Olsson, 2011) under some additional conditions on the target measure γ_T. Again, forward filtering–backward sampling does not improve estimates of the normalising constant, as noted, for instance, in Olsson and Rydén (2011).

Finally, in the case that the test function f_T admits an additive decomposition as in Equation 2.18, a sampling-based approximation of the above-mentioned forward-smoothing recursions has been developed by Olsson and Westerborn (2014).

2.5 Summary

In this chapter, we have described a generic SMC algorithm and have shown that it can be viewed as a special case of the MOSIS scheme described in the previous chapter. One way of interpreting the relationship between some well-known SMC algorithms mentioned in this chapter is outlined in Figure 2.1.

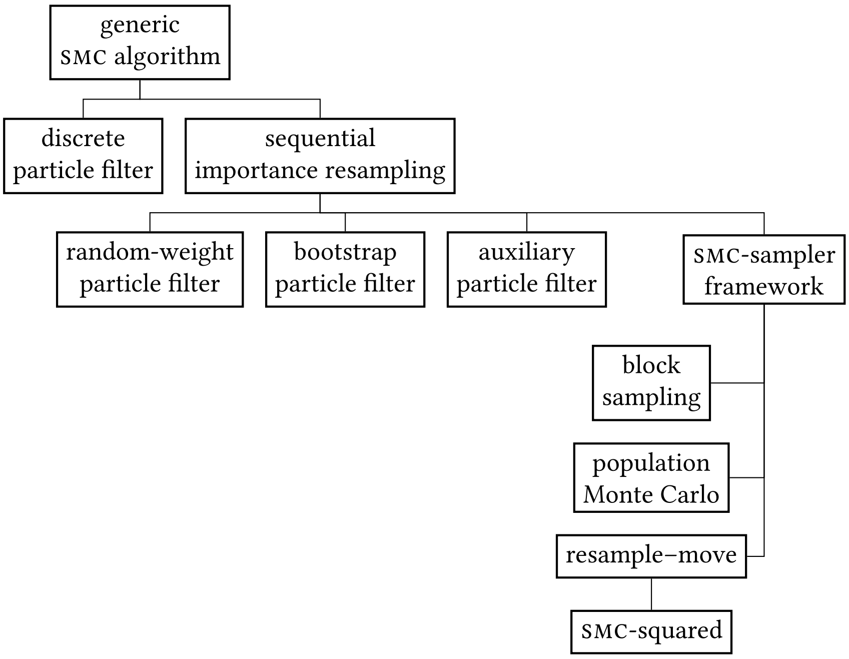

Figure 2.1 Some instances of the generic SMC algorithm mentioned in this chapter.

3 Markov Chain Monte Carlo Methods

3.1 Introduction

3.1.1 Motivation

In this chapter, we introduce Markov chain Monte Carlo methods. Section 3.1 outlines the main idea behind this class of Monte Carlo schemes and shows that they can be viewed as (an approximation to) a special case of the marginalised one-sample importance sampling scheme presented in Chapter 1. Using the same ideas again at a lower level, we construct a generic kernel which admits essentially every known Markov chain Monte Carlo kernel as a special case. This is done in Section 3.2. In Section 3.3, we demonstrate that multiple-proposal and 'randomised' Metropolis–Hastings kernels, pseudo-marginal kernels, and ensemble Markov chain Monte Carlo kernels can all be viewed as instances of the generic kernel. Other special cases, conditional sequential Monte Carlo kernels, which play a major rôle in Part II of this work, are detailed in Section 3.4. In particular, we show that the variance-reduction techniques: backward sampling and ancestor sampling share the same extended target distribution. To our knowledge, this is a new result.

In this chapter, we specifically assume that the measure with respect to which we want to calculate integrals is a probability measure. That is, throughout, we assume that we want to calculate integrals of the form $\pi(f)$ for some probability measure $\pi \in \mathcal{M}_1(\mathsf{X})$ and some test function $f \in \mathsf{F} \subseteq \mathcal{L}(\pi)$.

3.1 Remark. *Even though the methods described in this chapter only approximate* probability *measures, it is still generally possible to use these methods in the presence of intractable normalising constants, i.e. if* $\pi = \gamma/\mathfrak{z}$ *with some unknown normalising constant* $\mathfrak{z} = \gamma(\mathbb{1})$.

The main idea of *Markov chain Monte Carlo* (MCMC) methods is to approximate the measure π by a collection of samples $\boldsymbol{X} = X^{1:N}$ which are marginally (approximately) distributed according to π.

However, taking $\boldsymbol{X}$ to be *independent* (and thus IID) samples is impossible in realistic problems. Instead, MCMC methods generate *dependent* samples by iteratively sampling from some Markov kernel $P \in \mathcal{K}_1(\mathsf{X}, \mathsf{X})$ which is π-invariant, i.e. which satisfies

$$\pi P = \pi. \tag{3.1}$$

Assume for the moment that $X = x \sim \pi$. Then by Equation 3.1, any draw $Y \sim P(x, \cdot)$ is also distributed according to π. The samples X and Y are usually dependent but the main advantage of this idea is that sampling from $P(x, \cdot)$ is often feasible in situations in which generating IID samples from π is not.

The first MCMC algorithm, known as the Metropolis algorithm, was developed in the seminal work by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller (1953). In Hastings (1970), it was improved and generalised to a framework which includes the Metropolis–Hastings kernel as well as another MCMC kernel due to Barker (1965). Work by Peskun (1973) (and later extensions by Tierney (1998), Mira (1998)) showed that the Metropolis kernel dominates Barker's kernel in terms of resulting in a lower asymptotic variance.

The scope of MCMC methods was further expanded in the works by S. Geman and Geman (1984), Tanner and Wong (1987) and finally brought in to the mainstream of statistical computing by Gelfand and Smith (1990), Tierney (1994), Chib and Greenberg (1995). A thorough summary as well as further references may be found in Roberts and Rosenthal (2004), Robert and Casella (2004).

3.1.2 Note on Ergodicity

The estimator discussed in this chapter is 'valid' – in the sense of yielding unbiased estimates of integrals of the form $\pi(f)$ – as soon as (1) P is π-invariant, (2) the Markov chain $(X^n)_{n \in \mathbb{N}}$ has initial distribution π. However, as in the previous chapters, further conditions are needed to obtain consistent estimators. Even worse, initialising the Markov chain from π is generally impossible.

In order to obtain consistent estimates – and, as described in the next subsection, to permit non-stationary initial distributions $\mu \neq \pi$ – we require P to also be suitably *ergodic.* That is, we at least need

$$\mu P^N(f) \overset{N\to\infty}{\longrightarrow} \pi(f), \quad \text{for all } (\mu, f) \in \mathscr{P}(\mathsf{X}) \times \mathscr{L}(\pi),$$

where $\mathscr{P}(\mathsf{X})$ is some appropriate class of initial distributions – ideally, $\mathscr{P}(\mathsf{X}) = \mathscr{M}_1(\mathsf{X})$ is the set of all probability measures on X.

A discussion of various (stronger) notions of ergodicity and associated convergence rates is beyond the scope of this work. We refer the reader to Roberts and Rosenthal (2004), for an overview, and Meyn and Tweedie (2009), for a thorough theoretical treatment.

3.1.3 Generic Algorithm

Let μ be some probability measure on X from which we can sample and which will be the initial distribution of a Markov chain $(X^n)_{n\in\mathbb{N}}$.

The remainder of this section is concerned with constructing the following MCMC approximation of the probability measure π,

$$\pi_\mu^{\text{MCMC},N} := \sum_{n=1}^{N} w^n(\boldsymbol{Z})\delta_{X^n}.$$

Here, as in the previous chapters, $\boldsymbol{Z} = \boldsymbol{X} = X^{1:N}$ is a set of samples while $(w^n(\boldsymbol{Z}))_{n\in\mathsf{K}}$, for $\mathsf{K} := \mathbb{N}_N$, is a collection of non-negative weights. However, in contrast to the previous chapters, the nth weight will only depend on n (and possibly on X^n). This notation may seem unnecessarily complicated. However, in the next Subsection, it helps interpreting MCMC methods as a special case of the MOSIS approximation from Section 1.4.

As discussed below, the weights $(w^n(\boldsymbol{Z}))_{n\in\mathsf{K}}$ play an important rôle in discounting the effect of initialising the Markov chain from some distribution μ with $\mu \neq \pi$ but they may also be used for thinning.

A generic MCMC algorithm is outlined in Algorithm 3.2.

3.2 Algorithm (Markov chain Monte Carlo). *Given $X^1 \sim \mu$.*

- *For $n \in \mathbb{Z}_{2,N}$, sample $X^n \sim P(x^{n-1}, \cdot)$.*
- *Approximate $\pi(f)$ by $\pi_\mu^{\text{MCMC},N}(f)$.*

Clearly, $\pi_\mu^{\textsc{mcmc},N}(f)$ can only be an unbiased estimate of $\pi(f)$ if the Markov chain $(X^n)_{n\in\mathbb{N}}$ is initialised from the target distribution. Unfortunately, taking $\mu = \pi$ is impossible in realistic problems.

Instead, by appealing to the ergodicity properties of P, we can construct the weights $w^n(\boldsymbol{Z})$ in a way that largely removes the effect of sampling $X^1 \sim \mu \neq \pi$ from the estimator. This is commonly done by making sure that $w^n(\boldsymbol{Z}) = 0$, for any n in the *burn-in* period comprising the first $R < N$ iterations. The length of this period, R, needs to be large enough such that $\|\mu P^R - \pi\|$ is sufficiently small and hence that

$$\left\|\pi_\mu^{\textsc{mcmc},N} - \pi_\pi^{\textsc{mcmc},N}\right\| \tag{3.2}$$

is sufficiently small, where $\|\cdot\|$ denotes some suitable norm.

3.3 Remark. *For the rest of this chapter, we assume that $\mu = \pi$ and write $\pi_\pi^{\textsc{mcmc},N} =: \pi^{\textsc{mcmc},N}$, for simplicity. However, we stress that this is an unrealistic assumption and that in practice the kernel P needs to be sufficiently ergodic and the weights constructed in such a way that the distance in Equation* 3.2 *is sufficiently small.*

More complicated weights can also be used to accommodate *thinning*, i.e. for some suitable $M \in \mathbb{N}$, we may specify the weights in such a way that $w^m(z) = 0$, for any m satisfying $m \bmod M \neq 0$. If X^n and X^{n+m} are highly correlated for all $m \in \mathbb{N}_{M-1}$, then this choice has little effect on the variance of the estimate $\pi^{\textsc{mcmc},N}(f)$ but it dramatically reduces the memory cost because only every Mth value of the chain has to be stored. Valid ways of constructing the weights are implied by the IS interpretation of MCMC methods which is detailed in the next subsection.

3.1.4 Interpretation as Marginalised One-Sample Importance Sampling

In this subsection, we show that MCMC methods can be viewed as a special case of the MOSIS framework introduced in Chapter 1. To that end, we again introduce an extended proposal distribution, $\bar{\psi} \in \mathcal{M}_1(\overline{\mathbf{X}})$, and an extended target distribution $\bar{\pi} \in \mathcal{M}_1(\overline{\mathbf{X}})$ such that (1) $\bar{\pi}$ admits π as a marginal, (2) we can sample from $\bar{\psi}$, (3) $\bar{w} := \mathrm{d}\bar{\pi}/\mathrm{d}\bar{\psi}$ exists and can be evaluated point-wise. We then show that the usual MCMC approximation of π, $\pi^{\textsc{mcmc},N}$, coincides with a Rao–Blackwellised IS approximation of $\bar{\pi}$, based on a single sample.

Extended Proposal Distribution. As before, we include an index K into the state space which takes values in some finite space $\mathsf{K} := \mathbb{N}_N$. The collection of random variables generated by the MCMC algorithm, $\boldsymbol{Z} = \boldsymbol{X} = X^{1:N}$, takes values in the product space $\mathbf{Z} := \mathbf{X} := \mathsf{X}^N$. Finally, we let $\bar{\mathbf{X}} := \mathsf{X} \times \mathsf{K} \times \mathbf{Z}$ be the space on which we define the extended proposal distribution and accordingly write $\bar{\boldsymbol{X}} := (X, K, \boldsymbol{Z})$ for a random vector drawn from this distribution.

Assuming that we initialise the Markov chain from π (see Remark 3.3), the extended proposal distribution may be given by

$$\bar{\psi}(\mathrm{d}\bar{\boldsymbol{x}}) := \pi(\mathrm{d}x^1) P^{\otimes(N-1)}(x^1, \mathrm{d}x^{2:N}) \xi(\boldsymbol{x}, \mathrm{d}k) \delta_{x^k}(\mathrm{d}x),$$

where the kernel $\xi \in \mathcal{K}_1(\mathbf{X}, \mathsf{K})$ defines a distribution over the index K.

Extended Target Distribution. A corresponding extended target distribution can be constructed as

$$\begin{aligned}\bar{\pi}(\mathrm{d}\bar{\boldsymbol{x}}) := {} & \pi(\mathrm{d}x) \varLambda(x, \mathrm{d}k) \delta_x(\mathrm{d}x^k) P^{\otimes(N-k)}(x^k, \mathrm{d}x^{k+1:N}) \\ & \times L^{\otimes(k-1)}(x^k, \mathrm{d}x^{k-1} \times \cdots \times \mathrm{d}x^1).\end{aligned}$$

Above, the stochastic kernel $\varLambda \in \mathcal{K}_1(\mathsf{X}, \mathsf{K})$ defines a distribution over K and $L \in \mathcal{K}_1(\mathsf{X}, \mathsf{X})$ is the time-reversal kernel of π under P, i.e. L is defined by $L(x', \mathrm{d}x) = [\mathrm{d}P(x, \cdot)/\mathrm{d}\pi](x')\pi(\mathrm{d}x)$.

Rao–Blackwellisation. Assuming that ξ is chosen such that $\bar{\pi} \ll \bar{\psi}$,

$$\bar{w}(\bar{\boldsymbol{x}}) := \frac{\mathrm{d}\bar{\pi}}{\mathrm{d}\bar{\psi}}(\bar{\boldsymbol{x}}) = \frac{\varLambda(x^k, \{k\})}{\xi(\boldsymbol{x}, \{k\})} \mathbb{1}_{\{x^k\}}(x). \tag{3.3}$$

Given $\bar{\boldsymbol{X}} \sim \bar{\psi}$, a one-sample IS approximation of $\bar{\pi}$ is given by the random measure $\bar{\pi}^{\mathrm{IS},1} := \bar{w}(\bar{\boldsymbol{X}})\delta_{\bar{\boldsymbol{X}}}$. If we write

$$w^k(\boldsymbol{Z}) := \mathbb{E}\big[\bar{w}(\bar{\boldsymbol{X}}) \mathbb{1}_{\{k\}}(K) \big| \boldsymbol{Z}\big] = \varLambda(x^k, \{k\}),$$

then for any $A \in \mathcal{B}(\mathsf{X})$, Rao–Blackwellising the estimate $\bar{\pi}^{\mathrm{IS},1}(A \times \mathsf{K} \times \mathbf{Z})$, shows that the usual MCMC approximation of $\pi(A)$ is an instance of the MOSIS approximation, i.e.

$$\begin{aligned}\pi^{\mathrm{MOSIS},N}(A) &= \mathbb{E}\big[\bar{\pi}^{\mathrm{IS},1}(A \times \mathsf{K} \times \mathbf{Z}) \big| \boldsymbol{Z}\big] \\ &= \sum_{n=1}^{N} w^n(\boldsymbol{Z}) \delta_{X^n}(A) \\ &= \pi^{\mathrm{MCMC},N}(A).\end{aligned}$$

Hence, the MCMC approximation of π can be seen as a special case of IS, more precisely as a special case of the MOSIS construction from Chapter 1. Note that this immediately implies that MCMC may be viewed as a special case of the Monte Carlo method. We recognise that this is somewhat at odds with the often-held view that the Monte Carlo method is 'a special case of MCMC' (Geyer, 2011, p. 6).

Rôle of Time-Reversal Kernels. Using the time-reversal kernel to incorporate a corresponding π-invariant kernel into an IS scheme is a common approach employed, for instance, in the generalised IS approach (MacEachern et al., 1999) as described in Example 1.9, in SMC samplers (Del Moral et al., 2006b) as described in Example 2.15, as well as in Storvik (2011). Indeed, it is often the only feasible way of using the MCMC kernels described in the next section within IS. Differently constructed extended target distributions lead to intractable Radon–Nikodým derivatives as stressed by (Del Moral, Doucet & Jasra, 2006a).

The drawback, however, is that this way of simplifying the Radon–Nikodým derivative in Equation 3.3 requires the MCMC chain to be initialised from π (though, in practice, this can be relaxed the number of burn-in samples, R is sufficiently large). Ideally, we would like to initialise the chain from some other distribution, $\mu \neq \pi$ and use the Radon–Nikodým derivative to compensate for the fact that samples are not drawn (marginally) from π. Unfortunately, this is generally infeasible.

3.2 Generic MCMC Kernel

3.2.1 Elementary Kernels

In this section, we construct a generic MCMC kernel which admits essentially all known MCMC kernels as a special case. Its construction is based around the IS framework from Andrieu and Roberts (2009), Andrieu et al. (2010) and is thus closely linked to the extended measure $\bar{\gamma}$ from the MOSIS approach in Chapter 1. However, instead of obtaining an IS approximation of (a normalised version of) this measure by averaging over all candidates $X = X^{1:N}$, we sample one of these candidates, X^K, which is then marginally distributed according to π.

Related generic MCMC frameworks have appeared in Tjelmeland (2004), Storvik (2011) though without exploiting the fundamental insight from Andrieu and Roberts (2009) as described in Remark 1.16 and without using the extra auxiliary variables required for more sophisticated MCMC kernels. Our framework is similar to that of Lee (2011) around which Lee, Andrieu and Doucet (in prep.) also develop a number of novel extensions.

In particular in this subsection, we describe an elementary MCMC kernel. Essentially all MCMC kernels can be decomposed into repeated application of such kernels to an extended target distribution which is constructed in such a way that sampling from the elementary kernels is possible. In an MCMC context, such extended state-space constructions are often called *data augmentation* (Tanner & Wong, 1987).

Assume that the state space can be decomposed as $\mathsf{X} = \mathsf{X}_0 \times \mathsf{X}_1$ and let $X = (X_0, X_1) \sim \pi$. Let $\Pi \in \mathcal{K}_1(\mathsf{X}_0, \mathsf{X}_1)$ be the stochastic kernel defining the full conditional distribution of X_1 under π then the *elementary kernel*

$$P(x, \mathrm{d}x') := \delta_{x_0}(\mathrm{d}x_0')\Pi(x_0', \mathrm{d}x_1')$$

is clearly π-invariant. We also refer to these elementary kernels as *Gibbs* kernels in this work since their concatenation leads to MCMC algorithms known as Gibbs samplers. These are described in the next subsection.

3.2.2 Combinations of Kernels

More complicated MCMC kernels are constructed by combining elementary MCMC kernels. Indeed, let $(\widetilde{P}_m)_{m\in\mathsf{M}}$ be a countable collection of π-invariant kernels with $\widetilde{P}_m = \widetilde{P}((m, \cdot), \cdot) \in \mathcal{K}_1(\mathsf{X}, \mathsf{X})$, for $m \in \mathsf{M} := \mathbb{N}_M$. Additionally, let $\beta \in \mathcal{K}_1(\mathsf{M}, \mathsf{M})$ be another stochastic kernel which will be used to determine the choice of kernel P_m.

In this case, we can include the indices M into the state space to justify kernels of the form

$$P((m, x), \mathrm{d}m' \times \mathrm{d}x') := \beta(m, \mathrm{d}m')\widetilde{P}((m', x), \mathrm{d}x').$$

This implies a *mixture* of (elementary) kernels on the marginal space X. As a special case, we can justify *compositions* of kernels (on the marginal space), $P = P_1 \cdots P_M$, by setting

$$\beta((m, x), \mathrm{d}m') := \mathbb{1}_{\{M\}}(m)\delta_1(\mathrm{d}m') + \mathbb{1}_{\{1,\ldots,M-1\}}(m)\delta_{m+1}(\mathrm{d}m'). \quad (3.4)$$

3.4 Example (Gibbs sampler). *Assume that we can decompose the state space as* $\mathsf{X} = \mathsf{X}_{1:J}^{\times}$ *for some* $J \in \mathbb{N}$ *and write* $X = X_{1:J}$. *Define Sets* $J_1, \dots, J_M \subseteq \mathbb{N}_J$ *such that* $\bigcup_{m \in \mathsf{M}} J_m = \mathbb{N}_J$. *Furthermore, define the* m*th Gibbs kernel via*

$$\widetilde{P}((m, x), \mathrm{d}y) := \delta_{x_{J_m}}(\mathrm{d}y_{J_m}) \mathit{\Pi}_m(y_{-J_m}, \mathrm{d}y_{J_m}),$$

where $X_{J_m} := (X_j)_{j \in J_m}$, $X_{-J_m} := (X_j)_{j \in J \setminus J_m}$ *and where* $\mathit{\Pi}_m(x_{-J_m}, \cdot)$ *denotes the full conditional distribution of* X_{J_m} *under* π. *In this case, the Markov chain induced by* $P = \beta \otimes \widetilde{P}$ *is called a* Gibbs sampler *(S. Geman & Geman, 1984). In particular, it is called a*

- random-scan *Gibbs sampler if* $\beta(m, \cdot)$ *is non-degenerate,*
- deterministic-scan *Gibbs sampler if* β *is as given in Equation* 3.4.

3.5 Example (partially-collapsed Gibbs sampler). *Deterministic-scan Gibbs samplers are often presented as updating every component of* X *exactly once per iteration, i.e. taking* $(J_m)_{m \in \mathsf{M}}$ *to be a partition of* $\mathbb{N}_J$. *However, large reductions in the asymptotic variance are achievable if* $J_m \cap J_{m+1} \neq \emptyset$, *and in this case, the Gibbs sampler is often referred to as a* partially-collapsed *Gibbs sampler (Van Dyk & Park, 2008).*

3.2.3 Generic MCMC Kernel

Clearly, sampling from a tractable distribution on a sufficiently small, finite state space is feasible and we would not need MCMC algorithms for this task. However, the only MCMC kernel capable of sampling (approximately) from a higher-dimensional distribution π on general state space is the Gibbs-sampling kernel presented in Example 3.4. Unfortunately, this kernel requires sampling from full conditional distributions under π which are often intractable.

In this subsection, we construct a generic π-invariant MCMC kernel which does not require sampling from full conditional distributions under π. Unsurprisingly, this generic kernel is based around a state-space extension pioneered by Tjelmeland (2004). That is, it can be viewed as being invariant with respect to a distribution on an extended space which (1) admits π as a marginal, (2) has full conditional distributions from which we *can* sample. The resulting MCMC algorithm can thus be viewed as a Gibbs sampler targeting this extended distribution. As we shall see, this generic kernel admits most known MCMC kernels as a special case.

The generic MCMC kernel is again based around (a normalised version of) the extended measure $\bar{\pi} = \bar{\gamma}/\bar{\gamma}(\mathbb{1}) \in \mathcal{M}_1(\bar{\mathbf{X}})$ from Chapter 1. Recall that $\bar{\mathbf{X}} = \mathsf{X} \times \mathsf{K} \times \mathbf{Z}$, where $\mathsf{K} = \mathbb{N}_N$ and $\mathbf{Z} = \mathsf{X}^N \times \mathsf{Y}$, where Y is the set of values taken by some auxiliary variables, Y.

Note that in the previous section, we used the framework from Chapter 1 to interpret the entire MCMC algorithm as a special case of IS. Here, we employ the same framework again but at a lower level to construct the π-invariant kernel, P.

Further Extended Target Distribution. Recall that the extended distribution can be seen as a distribution over some auxiliary variables Y, a pool of N candidates for X, $\boldsymbol{X} = X^{1:N}$, and an index K. It is constructed in such a way that if $\bar{\boldsymbol{X}} = (X, K, \boldsymbol{Z}) = (X, K, X^{1:N}, Y) \sim \bar{\pi}$ then $X^K = X \sim \pi$ (Assumption 1.12). Approximations of π were previously constructed by integrating out (X, K) and thus averaging over all possible candidates in an IS scheme.

Here, instead of integrating (X, K) out, we generate a new value for the index K, denoted $\widetilde{K}$, and a 'new' candidate, $\widetilde{X} = X^{\widetilde{K}}$, in such a way that again, $\widetilde{X} \sim \pi$ under the extended distribution defined below. More precisely, writing $\widetilde{\boldsymbol{X}} := (\bar{\boldsymbol{X}}, \widetilde{K}, \widetilde{X})$ and recalling that $\bar{\boldsymbol{Z}} = (K, \boldsymbol{Z})$, the generic MCMC kernel targets the extended distribution

$$\begin{aligned}\tilde{\pi}(\mathrm{d}\tilde{\boldsymbol{x}}) &:= \bar{\pi}(\mathrm{d}\bar{\boldsymbol{x}})\varXi(\bar{\boldsymbol{x}}, \mathrm{d}\tilde{k} \times \mathrm{d}\tilde{x})\\ &= \pi(\mathrm{d}x)\bar{\varPi}(x, \mathrm{d}\bar{\boldsymbol{z}})\varXi(\bar{\boldsymbol{x}}, \mathrm{d}\tilde{k} \times \mathrm{d}\tilde{x})\end{aligned}$$

on $\widetilde{\mathbf{X}} := \bar{\mathbf{X}} \times \mathsf{K} \times \mathsf{X}$. Here, $\varXi \in \mathcal{K}_1(\bar{\mathbf{X}}, \mathsf{K} \times \mathsf{X})$ is chosen such that

$$\widetilde{\boldsymbol{X}} \sim \tilde{\pi} \quad \Rightarrow \quad X^{\widetilde{K}} = \widetilde{X} \sim \pi. \tag{3.5}$$

Note that such a kernel exists because we can always (and will often) take $\varXi(\bar{\boldsymbol{x}}, \mathrm{d}\tilde{k} \times \mathrm{d}\tilde{x}) = \bar{\pi}^{\mathrm{c}}(\boldsymbol{z}, \mathrm{d}\tilde{k} \times \mathrm{d}\tilde{x})$, where $\bar{\pi}^{\mathrm{c}}(\boldsymbol{z}, \,\cdot\,)$ is the full conditional distribution of (K, X) under $\bar{\pi}$.

Dominating Measure. Throughout the remainder of this chapter, we assume that $\bar{\pi}$ has a density $\overline{w}$ with respect to a suitable dominating measure $\bar{\psi}$ which admits the factorisation

$$\bar{\psi}(\mathrm{d}\bar{\boldsymbol{x}}) := \psi(\mathrm{d}\boldsymbol{z})\xi(\boldsymbol{z}, \mathrm{d}k)\delta_{x^k}(\mathrm{d}x),$$

where $\psi \in \mathcal{M}_\sigma(\mathbf{Z})$ and $\xi \in \mathcal{K}_\sigma(\mathbf{Z}, \mathsf{K})$. This factorisation is required to obtain the following explicit representation of $\bar{\pi}^{\mathrm{c}}$. For $\bar{\boldsymbol{x}} = (x, k, \boldsymbol{z})$, write $w^k(\boldsymbol{z}) := \bar{w}(\bar{x})\xi(\boldsymbol{z}, \{k\})$. We can then represent the full conditional distribution of (X, K) under $\bar{\pi}$ as $\bar{\pi}^{\mathrm{c}}(\boldsymbol{z}, \mathrm{d}k \times \mathrm{d}x) = \bar{\pi}_{\boldsymbol{z}}(\mathrm{d}k)\delta_{x^k}(\mathrm{d}x)$, where

$$\bar{\pi}_{\boldsymbol{z}}(\{k\}) := \frac{w^k(\boldsymbol{z})}{\sum_{n=1}^N w^n(\boldsymbol{z})}.$$

Finally, note the measure ψ does not depend on k. It therefore generalises the symmetric dominating measure required in Green (1995) as discussed in the Subsection 3.3.2,

Summary. By Equation 3.5, a π-invariant kernel is given by

$$P(x, A) := \int_{\bar{\mathsf{Z}} \times \mathsf{K} \times A} \bar{\Pi}(x, \mathrm{d}\bar{z}) \Xi(\bar{\boldsymbol{x}}, \mathrm{d}\tilde{k} \times \mathrm{d}\tilde{x}),$$

for all $(x, A) \in \mathsf{X} \times \mathcal{B}(\mathsf{X})$. It is summarised in Algorithm 3.6.

3.6 Algorithm (generic MCMC kernel). *Given $X \sim \pi$,*

(1) sample $\bar{\boldsymbol{Z}} \sim \bar{\Pi}(x, \cdot)$ and set $\bar{\boldsymbol{X}} := (X, \bar{\boldsymbol{Z}})$,

(2) sample $(\tilde{K}, \tilde{X}) \sim \Xi(\bar{\boldsymbol{x}}, \cdot)$,

(3) output $X := \tilde{X}$ $(= X^{\tilde{K}})$.

Note that the generic MCMC kernel is just a Gibbs kernel targeting the extended distribution $\tilde{\pi}$. The advantage of this state-space extension is that the kernels $\bar{\Pi}$ and Ξ can often be chosen in such a way that sampling from them – and thus sampling from this Gibbs kernel – is feasible even if sampling from full conditional distributions under π is not.

Finally, we note that the generic MCMC kernel is π-invariant by construction. In particular, this insight immediately shows that all the MCMC kernels mentioned in this chapter are π-invariant (since they are all special cases of this approach) without appealing (explicitly) to sufficient conditions such as *detailed balance*. However, we reiterate the fact that π-invariance is not sufficient for obtaining useful MCMC kernels. Indeed, taking $q := \pi$ in Example 1.14 can be viewed as initialising the Markov chain from π and then applying the trivial kernel $P(x, \cdot) := \delta_x$. This kernel is π-invariant but not ergodic and thus clearly useless.

For later use, we define the following terminology, borrowed from Nicholls, Fox and Watt (2012), which will be justified in the next section.

3.7 Definition (randomised MCMC kernel). *Any instance of the generic MCMC kernel is called 'randomised' if it makes use of further auxiliary variables, i.e. if* $\mathsf{Y} \neq \emptyset$.

3.2.4 Finite State-Space Kernels

In this section, we show that finite-state space MCMC kernels such as the Metropolis kernel (Metropolis et al., 1953) and the modified discrete-state Gibbs sampler kernel from Liu (1996) can be viewed as special cases of the generic kernel introduced in the previous subsection. These kernels are mainly of interest because they themselves serve as building blocks for the general-state space MCMC kernels described in the next section. Namely, they will be used to construct the kernel $\varXi$ in Algorithm 3.6 when the state space X is infinite.

Throughout this subsection, we do not make use of the auxiliary variables, Y, and therefore set $\mathsf{Y} := \emptyset$, for simplicity. However, there is no principal difficulty with including extra auxiliary variables and thus obtaining 'randomised' versions of the kernels described here.

Assume a finite state space, without loss of generality $\mathsf{X} = \mathbb{N}_N$, and assume that $N \in \mathbb{N}$ is sufficiently small so that any normalising constant $\mathfrak{z} = \gamma(\mathbb{1}) = \sum_{x \in \mathsf{X}} \gamma(x)$ of $\pi = \gamma / \mathfrak{z}$ is tractable which in turn implies that we can sample directly from π. Throughout, we write $\gamma(\{x\}) = \gamma(x)$.

General Construction. Define the extended target distribution through

$$\bar{\varPi}(x, \mathrm{d}\bar{\boldsymbol{z}}) := \delta_x(\mathrm{d}k)\delta_x(\mathrm{d}x^k)\delta_{(1,\dots,k-1,k+1,\dots,N)}(\mathrm{d}\boldsymbol{x}^{-k}), \quad (3.6)$$

$$\varXi(\bar{\boldsymbol{x}}, \{\tilde{k}\} \times \mathrm{d}\tilde{x}) := A(x^k, x^{\tilde{k}})\delta_{x^{\tilde{k}}}(\mathrm{d}\tilde{x}). \quad (3.7)$$

Here $A(x, y) := s(x, y)\pi(y)$, where, following Hastings (1970), we take $s \colon \mathsf{X}^2 \to [0, \infty)$ to be some non-negative symmetric function chosen such that $\varXi$ is a stochastic kernel. In this case, if $\widetilde{\boldsymbol{X}} \sim \tilde{\pi}$, we clearly have that $X^K \sim \pi$ and by symmetry, $X^{\widetilde{K}} \sim \pi$.

Examples. The remainder of this subsection details how some well-known finite state-space MCMC kernels can be viewed as special cases of this framework.

3.8 Example (IID sampling). *Generating IID samples from π can be interpreted as running an MCMC algorithm which uses the kernel P based on Equations* 3.6 *and* 3.7 *with $s \equiv 1$, i.e. $A(x, y) = \pi(y)$.*

The following example defines an alternative MCMC kernel which dominates IID sampling in the Peskun sense (Peskun, 1973).

3.9 Example (modified discrete-state Gibbs sampler). *Another valid choice for the symmetric function s is given by*

$$s(x, y) := \begin{cases} [1 - \pi(x) \vee 1 - \pi(y)]^{-1}, & \text{if } x \neq y, \\ \left[1 - \sum_{y \in \mathsf{X} \setminus \{x\}} s(x, y)\pi(y)\right] \dfrac{1}{\pi(x)}, & \text{otherwise.} \end{cases} \tag{3.8}$$

In this case, Ξ defined according to Equation 3.7 *is indeed a stochastic kernel. To see this, define the sets $\mathsf{X}_0 := \{\, y \in \mathsf{X} \setminus \{x\} \mid \pi(x) \leq \pi(y) \,\}$ and $\mathsf{X}_1 := \{\, y \in \mathsf{X} \setminus \{x\} \mid \pi(x) > \pi(y) \,\}$, then*

$$\begin{aligned} \sum_{y \in \mathsf{X} \setminus \{x\}} A(x, y) &= \sum_{y \in \mathsf{X}_0} \frac{\pi(y)}{1 - \pi(x)} + \sum_{y \in \mathsf{X}_1} \frac{\pi(y)}{1 - \pi(y)} \\ &\leq \frac{1}{1 - \pi(x)} \sum_{y \in \mathsf{X} \setminus \{x\}} \pi(y) \\ &= 1. \end{aligned}$$

The kernel P based on this symmetric function can be recognised as the modified discrete-state Gibbs kernel from (Liu, 1996).

3.10 Example (Metropolis). *If $N = 2$, then the kernel defined by Equations* 3.6 *and* 3.8 *is the famous* Metropolis *kernel introduced in (Metropolis et al., 1953). Indeed, in this case*

$$A(x, y) = \frac{\pi(y)}{\pi(y) \vee \pi(x)} = 1 \wedge \alpha(x, y),$$

for $x \neq y$ with $\alpha(x, y) := \gamma(y)/\gamma(x)$. This the usual Metropolis 'acceptance probability', so called because in some of the general state-space kernels presented in the next section, it can be viewed as the probability of accepting a proposed move from $x = x^k$ to $y = \tilde{x}^k$.

3.3 General State-Space Kernels

3.3.1 Barker, Forced-Move, Metropolis–Hastings

In this subsection, we show that many classical MCMC kernels for general state spaces such as the Metropolis kernel (Metropolis et al., 1953), Barker's kernel (Barker, 1965), the Metropolis–Hastings kernel (Hastings, 1970) as well as generalisations of the latter two to $N - 1 > 1$ proposals (Tjelmeland, 2004) can be viewed as special cases of the generic kernel introduced in Subsection 3.2.3.

Assume now that X is some general state space, e.g. continuous or finite but large enough such that the normalising constant of $\pi = \gamma/\mathfrak{z}$, given by $\mathfrak{z} = \gamma(\mathbb{1})$, is intractable.

General Idea. To ease the explanation, we recall that the main intuition behind the generic kernel can be summarised as follows. Given some value $X \sim \pi$, we propose N possible candidates with the Kth candidate being set equal to X. Of course, we cannot generally propose these candidates from π. Instead, we propose them from another (conditional) distribution and then pick the next value of the MCMC chain, $\widetilde{X} = X^{\widetilde{K}}$, by sampling a new index $\widetilde{K}$. If $\widetilde{K} = K$, then the Markov chain chain induced by the generic kernel does not move to a different value. If $N = 2$, then this amounts to 'rejecting' the proposed candidate $\boldsymbol{X}^{-K}$.

The extended target distribution from Chapter 1, $\bar{\pi}$, involved in the construction of the generic kernel is essentially the distribution targeted by the *pseudo-prior* approach from Carlin and Chib (1995) (if all models are defined on the same space). Though in the pseudo-prior case, the index K actually has some physical interpretation; namely, it indexes one out of N possible models. Because this framework allows for $N - 1 > 1$ proposals to be made (of which one, the $\widetilde{K}$th, is 'accepted') it has recently received renewed attention with a view to harnessing the benefits of parallelisation within MCMC algorithms (Calderhead, 2014).

Simplification. To obtain the concrete expressions for 'acceptance probabilities' and other familiar quantities, we often assume in this section that the extended target distribution factorises according to Assumption 3.11. However, we stress that this factorisation is only employed for illustrative purposes and is not necessary for constructing π-invariant kernels.

3.11 Assumption (simplified target distribution). *We take*

$$\bar{\Pi}(x, \mathrm{d}\bar{\boldsymbol{x}}) = \mathrm{Unif}_{\mathsf{K}}(\mathrm{d}k)\delta_x(\mathrm{d}x^k)Q_k(x, \mathrm{d}\boldsymbol{x}^{-k})R_k(\boldsymbol{x}, \mathrm{d}y),$$

where $Q_k(x, \cdot) := Q((x,k), \cdot)$, $R_k(\boldsymbol{x}, \cdot) := R((\boldsymbol{x},k), \cdot)$, *and*

(1) $Q \in \mathcal{K}_1(\mathsf{X} \times \mathsf{K}, \mathsf{X}^{N-1})$ *proposes the* $N-1$ *remaining candidates,*

(2) $R \in \mathcal{K}_1(\mathbf{X} \times \mathsf{K}, \mathsf{Y})$ *defines a distribution over the auxiliary variable* Y*.*

In this subsection, we take $\Xi(\bar{\boldsymbol{x}}, \mathrm{d}\tilde{k} \times \mathrm{d}\tilde{x}) := \widetilde{P}_{\boldsymbol{z}}(k, \mathrm{d}\tilde{k})\delta_{x^{\tilde{k}}}(\mathrm{d}\tilde{x})$, where $\widetilde{P}_{\boldsymbol{z}}$ will be one of the finite state-space MCMC kernels – invariant with respect to $\bar{\pi}_{\boldsymbol{z}}$ – constructed in the previous subsection.

In addition in this subsection, we assume a 'non-randomised' algorithm, i.e. we do not make use of the further auxiliary variables and take $\mathsf{Y} = \emptyset$ so that $\boldsymbol{Z} = \boldsymbol{X} = X^{1:N}$. We then obtain the classical Metropolis–Hastings algorithm and Barker's algorithms as instances of the generic kernel. We present slightly generalised versions of both which allow for $N-1 > 1$, not necessarily (conditionally) independent 'proposals'.

Barker's Kernel. If we take $\widetilde{P}_{\boldsymbol{x}}(k, \cdot) := \bar{\pi}_{\boldsymbol{x}}$ to be the IID-sampling kernel from Example 3.8, then the generic kernel P simplifies to *Barker's kernel* (Barker, 1965), slightly generalised to allow for $N-1$ (not necessarily independent) proposals. In turn, Barker's kernel admits the *frozen Carlin-&-Chib* kernel from Douc, Maire and Olsson (2014) as a special case (again, the index K then has a 'physical' interpretation: it indexes a particular component in an N-component mixture model).

In particular, under Assumption 3.11, a single application of Barker's kernel is outlined in Algorithm 3.12.

3.12 Algorithm (Barker). *Given* $X \sim \pi$*,*

(1) sample $K \sim \mathrm{Unif}_{\mathsf{K}}$, $\boldsymbol{X}^{-k} \sim Q_k(x^k, \cdot)$, *and set* $X^k := X$*,*

(2) set $\widetilde{K} = l$ *with probability*

$$\widetilde{P}_{\boldsymbol{x}}(k, \{l\}) = \bar{\pi}_{\boldsymbol{x}}(\{l\}) = \frac{\gamma(\mathrm{d}x^l)Q_l(x^l, \mathrm{d}\boldsymbol{x}^{-l})}{\sum_{n=1}^{N}\gamma(\mathrm{d}x^n)Q_n(x^n, \mathrm{d}\boldsymbol{x}^{-n})},$$

(3) output $\widetilde{X} = X^{\tilde{k}}$*.*

Note that in this simplified case (due to the exchangeability implied by the uniform distribution for K in Assumption 3.11), it suffices to set $K := 1$, say, in Step 1 of Algorithm 3.12.

Forced-Move Kernels. If we take $\widetilde{P}_{\boldsymbol{x}}(k, \cdot)$ to be the modified discrete-state Gibbs kernel from Example 3.9 in the previous section, with target distribution $\bar{\pi}_{\boldsymbol{x}}$, then the generic kernel P simplifies to the *forced-move* kernel described in Chopin and Singh (2013) (though, in that work, the kernel also makes use of the auxiliary variables, Y, which we have left out here, for simplicity). As shown in that work, the forced-move kernel dominates (the above-mentioned generalisation of) Barker's kernel in terms of Peskun ordering (Peskun, 1973).

In particular, under Assumption 3.11, a single application of the forced-move kernel is summarised in Algorithm 3.13, where we set

$$\beta_{\boldsymbol{x}}(k,l) := \frac{\bar{\pi}_{\boldsymbol{x}}(\{l\})}{1 - \bar{\pi}_{\boldsymbol{x}}(\{k\}) \vee 1 - \bar{\pi}_{\boldsymbol{x}}(\{l\})}.$$

3.13 Algorithm (forced move). *Given $X \sim \pi$,*

(1) sample $K \sim \mathrm{Unif}_{\mathsf{K}}$, $\boldsymbol{X}^{-k} \sim Q_k(x^k, \cdot)$, and set $X^k := X$,

(2) set $\widetilde{K} = l$ with probability

$$\widetilde{P}_{\boldsymbol{x}}(k, \{l\}) := \begin{cases} \beta_{\boldsymbol{x}}(k,l), & \text{if } l \neq k, \\ 1 - \sum_{l \in \mathsf{K} \setminus \{k\}} \beta_{\boldsymbol{x}}(k,l), & \text{if } l = k, \end{cases}$$

(3) output $\widetilde{X} = X^{\tilde{k}}$.

Again in the simplified case (due to the exchangeability implied by Assumption 3.11), it suffices to set $K := 1$, say, in Step 2 of Algorithm 3.13. Note that setting $\widetilde{K} = k$ in Step 2 is usually interpreted as 'rejecting' all the $N - 1$ proposed candidates $\boldsymbol{X}^{-k}$.

Metropolis–Hastings Kernels. If $N = 2$, then the forced-move kernel simplifies to the classic *Metropolis–Hastings* (MH) kernel introduced by Hastings (1970). Note that in this case, $\widetilde{P}_{\boldsymbol{x}}$ reduces to the Metropolis kernel from Example 3.10 with target distribution $\bar{\pi}_{\boldsymbol{x}}$.

In particular, under Assumption 3.11, a single application of the MH kernel takes the familiar form outlined in Algorithm 3.14, where we set

$$\alpha_{\boldsymbol{x}}(k,l) := \frac{\bar{\pi}_{\boldsymbol{x}}(\{l\})}{\bar{\pi}_{\boldsymbol{x}}(\{k\})} = \frac{\gamma(\mathrm{d}x^l) Q_l(x^l, \mathrm{d}\boldsymbol{x}^{-l})}{\gamma(\mathrm{d}x^k) Q_k(x^k, \mathrm{d}\boldsymbol{x}^{-k})}.$$

3.14 Algorithm (Metropolis–Hastings). *Given $X \sim \pi$,*

(1) sample $K \sim \mathrm{Unif}_{\mathrm{K}}$, $\boldsymbol{X}^{-k} \sim Q_k(x^k, \cdot)$, and set $X^k := X$,

(2) set $\widetilde{K} = l$ with probability

$$\widetilde{P}_{\boldsymbol{x}}(k, \{l\}) := \begin{cases} 1 \wedge \alpha_{\boldsymbol{x}}(k, l), & \mathit{if}\, l \neq k, \\ 1 - 1 \wedge \alpha_{\boldsymbol{x}}(k, l), & \mathit{if}\, l = k, \end{cases}$$

(3) output $\widetilde{X} = X^{\tilde{k}}$.

As before, in this simplified case (due to the exchangeability implied by Assumption 3.11), it suffices to set $K := 1$, say, in Step 1. In particular, setting $\widetilde{K} = l \neq k$ in Step 1 is usually interpreted as 'accepting' the proposed candidate $\boldsymbol{X}^{-k}$ and $\widetilde{P}_{\boldsymbol{x}}(k, \{l\}) = 1 \wedge \alpha_{\boldsymbol{x}}(k, l)$ is the corresponding 'acceptance probability'.

It is well known that many MCMC kernels can themselves be viewed as special cases of the MH kernel on a suitably extended space:

Multiple-try Metropolis kernels (Liu, Liang & Wong, 2000) are shown to be MH kernels e.g. in Maire, Douc and Olsson (2014, Appendix B.2).

Delayed-acceptance kernels (Christen & Fox, 2005) are MH kernels whose proposal kernel is itself an MH kernel targeting another distribution,

Reversible-jump kernels (Green, 1995) (see Subsection 3.3.2) are MH kernels targeting a distribution on a particular countable union of spaces.

Pseudo-marginal MH kernels (Beaumont, 2003; Andrieu & Roberts, 2009) (see Subsection 3.3.4) are MH kernels targeting a version of the pseudo-marginal target distribution from Subsection 1.4.3.

Note that these kernels are not inherently reliant on $N = 2$ and can easily be generalised to using $N - 1 > 1$ proposals. In addition, they could also use of some set of auxiliary variables, Y, e.g. as in Subsection 3.3.3.

We conclude this subsection by noting that Gibbs kernels are often viewed as (one-proposal) MH kernels in which the acceptance probability is equal to 1 (Gelman, 1992). The above interpretation shows that the opposite interpretation is also possible. That is, MH kernels are fundamentally Gibbs kernels on a suitably extended space. Of course, this is unsurprising because any implementable Monte Carlo algorithm must necessarily decompose into steps that require sampling from tractable (conditional) distributions.

3.3.2 Reversible-Jump MCMC

In some cases, for instance in the context of model selection problems, the target distribution π is defined on a particular countable union of spaces,

$$\mathsf{X} := \bigcup_{m \in \mathsf{M}} (\{m\} \times \mathsf{X}_m), \tag{3.9}$$

for some sequence of essentially arbitrary spaces $(\mathsf{X}_m)_{m \in \mathsf{M}}$. MH kernels targeting such distributions are then known as *reversible-jump Markov chain Monte Carlo* (RJMCMC) kernels (Green, 1995) though there is no fundamental difficulty with targeting π using other (more general) instances of the generic MCMC kernel. Indeed, Extensions and modifications of the RJMCMC kernel can be found in Brooks, Giudici and Roberts (2003).

Often, RJMCMC kernels induce additional degenerate dependencies in the kernel $\bar{\Pi}$ because the candidates $\boldsymbol{X}^{-k}$ are constructed, at least in part, from deterministic transformations of X^k. As a result, the dominating measure ψ can no longer be the otherwise common choice: Lebesgue measure, counting measure or combination of the two.

In this case, constructing a suitable dominating measure ψ will often be difficult. In the specific case: $N = 2$, Green (1995) proposes way of constructing the 'symmetric' dominating measure ψ. Evaluating (densities with respect to) this measure typically requires calculating Jacobian determinants related to the deterministic transformation. This requirement is simply a product of making such (partially) deterministic proposals and is entirely unrelated to the specific structure of the state space X.

As shown by Besag (1997), Godsill (2001), RJMCMC kernels and other MCMC kernels targeting distributions on a space as in Equation 3.9 can actually be viewed as being embedded in (and thus being a special case of) the pseudo-prior approach from Carlin and Chib (1995). The latter uses standard MCMC algorithms to target a distribution on the product space

$$\mathsf{X} = \mathsf{M} \times \bigtimes_{m \in \mathsf{M}} \mathsf{X}_m.$$

Again, if $\mathsf{X}_m = \mathsf{X}_n$, for $n, m \in \mathsf{M}$, then this extended distribution can in turn be interpreted as a special case of the extended target distribution $\bar{\pi}$ (but one in which the index $M = K$ taking values in $\mathsf{M} = \mathsf{K}$ has some 'physical' interpretation, for instance as the model index).

3.3.3 Randomised MCMC

Recall that in Definition 3.7 we defined instances of the generic MCMC kernel as 'randomised' if they make use of additional auxiliary variables, Y, which are included in $\boldsymbol{Z} = (\boldsymbol{X}, Y)$. In this subsection, we motivate this terminology and give some examples of such kernels.

Randomised Metropolis–Hastings Kernels. Note that the full conditional distribution of (X, K) under $\bar{\pi}$ now also depends on the auxiliary variable, Y. Assume that the extended target distribution has the particular structure from Assumption 3.11 and that $N = 2$.

In particular, under Assumption 3.11, a single application of such a randomised MH kernel is outlined in Algorithm 3.15, where we write

$$\hat{\alpha}_{\boldsymbol{z}}(k, l) := \frac{\bar{\pi}_{\boldsymbol{z}}(\{l\})}{\bar{\pi}_{\boldsymbol{z}}(\{k\})} = \alpha_{\boldsymbol{x}}(k, l) \frac{\mathrm{d}R_l(\boldsymbol{x}, \cdot)}{\mathrm{d}R_k(\boldsymbol{x}, \cdot)}(y).$$

Here, $\alpha_{\boldsymbol{x}}(k, l)$ is as in the standard MH kernel from Algorithm 3.14.

3.15 Algorithm (randomised Metropolis–Hastings). *Given $X \sim \pi$,*

(1) sample $K \sim \mathrm{Unif}_{\mathrm{K}}$, $\boldsymbol{X}^{-k} \sim Q_k(x^k, \cdot)$, and set $X^k := X$,

(2) sample $Y \sim R_k(\boldsymbol{x}, \cdot)$,

(3) set $\widetilde{K} = l$ with probability

$$\widetilde{P}_{\boldsymbol{z}}(k, \{l\}) := \begin{cases} 1 \wedge \hat{\alpha}_{\boldsymbol{z}}(k, l), & \mathit{if}\, l \neq k, \\ 1 - 1 \wedge \hat{\alpha}_{\boldsymbol{z}}(k, l), & \mathit{if}\, l = k, \end{cases}$$

(4) output $\widetilde{X} = X^{\widetilde{k}}$.

As before, in this simplified case (due to the exchangeability implied by Assumption 3.11), it suffices to set $K := 1$, say, in Step 1.

Randomised MH kernels have for instance been studied in Ceperley and Dewing (1999) (the *penalty method*), in I. Murray, Ghahramani and MacKay (2006) (the *single-variable exchange algorithm*) and in Nicholls et al. (2012), Alquier, Friel, Everitt and Boland (2014). They are usually motivated by the fact that the 'exact' acceptance probability based on $\alpha_{\boldsymbol{x}}(k, l)$ is intractable but that introducing the auxiliary variables Y allows the evaluation of $\hat{\alpha}_{(\boldsymbol{x}, y)}(k, l)$. However, as shown in Andrieu and Vihola (2014), these kernels cannot have a smaller asymptotic variance than the corresponding non-randomised kernels.

Interpretation as Randomised Acceptance Probability. The terminology 'randomised' MCMC (also called 'perturbed' MCMC in Andrieu & Vihola, 2014) is motivated by the fact that the term $\hat{\alpha}_{(\boldsymbol{x},Y)}(k,l)$ in the acceptance probability can be viewed as a randomised version of the term $\alpha_{\boldsymbol{x}}(k,l)$ in the acceptance probability from the standard MH kernel in Algorithm 3.14. That is, using the reparametrisation

$$\phi_{\boldsymbol{x},k,l}(y) := \frac{\mathrm{d}R_l(\boldsymbol{x},\,\cdot\,)}{\mathrm{d}R_k(\boldsymbol{x},\,\cdot\,)}(y),$$

we may write $\hat{\alpha}_{(\boldsymbol{x},Y)}(k,l) = \alpha_{\boldsymbol{x}}(k,l)V$, where V is a random variable, taking values in $\mathsf{V} := [0,\infty)$, which is distributed according to

$$\zeta_{k,l}(\boldsymbol{x},\,\cdot\,) := R_k(\boldsymbol{x},\,\cdot\,) \circ \phi_{\boldsymbol{x},k,l}^{-1}.$$

Conditional on $\boldsymbol{X} = \boldsymbol{x}$, $K = k$ and $L = l$, this random variable has expectation 1 which implies $\mathbb{E}[\hat{\alpha}_{(\boldsymbol{x},Y)}(k,l)] = \alpha_{\boldsymbol{x}}(k,l)$. Thus, V is interpreted as noise perturbing the acceptance probability.

The fact that the randomised acceptance probability in these algorithms is equivalent to including the auxiliary variables Y into the state space is pointed out in Lee, Andrieu and Doucet (in prep.).

Reversibility. We stress again that viewing these kernels as special cases of the generic kernel from the previous section immediately shows that they are π-invariant – without appealing to (and having to check!) sufficient conditions such as detailed balance.

The following simple condition for π-reversibility is given by O'Neill, Balding, Becker, Eerola and Mollison (2000, Equation 3.10).

3.16 Proposition (O'Neill et al., 2000). *Let $\alpha_{\boldsymbol{x}}(k,l)$ in Algorithm 3.14 be replaced by $\alpha_{\boldsymbol{x}}(k,l)V$, where $V \sim \rho_{k,l}(\boldsymbol{x},\,\cdot\,)$ for some distribution $\rho_{k,l}(\boldsymbol{x},\,\cdot\,) \in \mathcal{M}_1(\mathsf{V})$. Then the resulting kernel is π-reversible if*

$$\frac{\mathbb{E}[1 \wedge \alpha_{\boldsymbol{x}}(k,l)V]}{\mathbb{E}[1 \wedge \alpha_{\boldsymbol{x}}(l,k)W]} = \alpha_{\boldsymbol{x}}(k,l), \tag{3.10}$$

for any $k,l \in \{1,2\}$ with $k \neq l$, and where $W \sim \rho_{l,k}(\boldsymbol{x},\,\cdot\,)$.

Proof. Note that $\alpha_{\boldsymbol{x}}(k,l) = 1/\alpha_{\boldsymbol{x}}(l,k)$. Hence, by Equation 3.10,

$$\bar{\pi}_{\boldsymbol{x}}(\{k\})\,\mathbb{E}[1 \wedge \alpha_{\boldsymbol{x}}(k,l)V] = \bar{\pi}_{\boldsymbol{x}}(\{l\})\,\mathbb{E}[1 \wedge \alpha_{\boldsymbol{x}}(l,k)W].$$

Thus, the randomised MH kernel is π-reversible. □

A version of the following sufficient condition for Equation 3.10 can be found in Andrieu and Vihola (2014, Lemma 6).

3.17 Proposition (Andrieu & Vihola, 2014). *Using the notation from Proposition 3.16, Equation* 3.10 *holds if*

$$\int_A \rho_{k,l}(\boldsymbol{x}, \mathrm{d}v)v = \int_{\mathsf{V}} \rho_{l,k}(\boldsymbol{x}, \mathrm{d}w)\, \mathbb{1}_{(1/\operatorname{id}_{\mathsf{V}})^{-1}(A)}(w), \tag{3.11}$$

for any $A \in \mathscr{B}(\mathsf{Y})$.

Proof. Again, let $V \sim \rho_{k,l}(\boldsymbol{x}, \cdot)$ and $W \sim \rho_{l,k}(\boldsymbol{x}, \cdot)$. Using that $\alpha_{\boldsymbol{x}}(k,l) = 1/\alpha_{\boldsymbol{x}}(l,k)$ and decomposing the domain of integration, the left hand side in Equation 3.10 can be written as

$$\frac{\mathbb{E}[V\, \mathbb{1}_{(\alpha_{\boldsymbol{x}}(k,l),\infty)}(V)] + \alpha_{\boldsymbol{x}}(l,k)\, \mathbb{E}[\mathbb{1}_{[0,\alpha_{\boldsymbol{x}}(k,l)]}(V)]}{\mathbb{E}[\mathbb{1}_{[0,\alpha_{\boldsymbol{x}}(l,k)]}(W)] + \alpha_{\boldsymbol{x}}(l,k)\, \mathbb{E}[W\, \mathbb{1}_{(\alpha_{\boldsymbol{x}}(l,k),\infty)}(W)]}\alpha_{\boldsymbol{x}}(k,l).$$

Note that $[a,b] = (1/\mathrm{id})^{-1}([1/b, 1/a])$, for $0 < a \leq b$, then by Equation 3.11, the first and second expectation in the denominator are equal to $\mathbb{E}[V\, \mathbb{1}_{(\alpha_{\boldsymbol{x}}(k,l),\infty)}(V)]$ and $\mathbb{E}[\mathbb{1}_{[0,\alpha_{\boldsymbol{x}}(k,l)]}(V)]$, respectively. Hence, Equation 3.10 is satisfied. □

In turn, our auxiliary-variable construction immediately implies the sufficient condition from Equation 3.11. To see this, write $\rho_{k,l} = \zeta_{k,l}$ and use the changes-of-variables $v = \phi_{\boldsymbol{x},k,l}(y)$ and $w = \phi_{\boldsymbol{x},l,k}(y)$. For any $(\boldsymbol{x}, A) \in \mathbf{X} \times \mathscr{B}(\mathsf{V})$ and any $k \neq l$, we then have

$$\begin{aligned}
\int_A \zeta_{k,l}(\boldsymbol{x}, \mathrm{d}v)v &= \int_A R_k(\boldsymbol{x}, \cdot) \circ \phi^{-1}_{\boldsymbol{x},k,l}(\mathrm{d}v)v \\
&= \int_{\mathsf{Y}} \frac{\mathrm{d}R_l(\boldsymbol{x}, \cdot)}{\mathrm{d}R_k(\boldsymbol{x}, \cdot)}(y) R_k(\boldsymbol{x}, \mathrm{d}y)\, \mathbb{1}_{\phi^{-1}_{\boldsymbol{x},k,l}(A)}(y) \\
&= \int_{\mathsf{V}} R_l(\boldsymbol{x}, \cdot) \circ \phi^{-1}_{\boldsymbol{x},l,k}(\mathrm{d}w)\, \mathbb{1}_{\phi_{\boldsymbol{x},l,k} \circ \phi^{-1}_{\boldsymbol{x},k,l}(A)}(w) \\
&= \int_{\mathsf{V}} R_l(\boldsymbol{x}, \cdot) \circ \phi^{-1}_{\boldsymbol{x},l,k}(\mathrm{d}w)\, \mathbb{1}_{(1/\operatorname{id}_{\mathsf{V}})^{-1}(A)}(w) \\
&= \int_A \zeta_{l,k}(\boldsymbol{x}, \mathrm{d}w)\, \mathbb{1}_{(1/\operatorname{id}_{\mathsf{V}})^{-1}(A)}(w).
\end{aligned}$$

Here, in the fourth step, we have used that for any suitable surjective function $f : \mathsf{X}_1 \to \mathsf{X}_2$, and any set $A \subseteq \mathsf{X}_2$, $f \circ (1/f)^{-1}(A) = (1/\operatorname{id}_{\mathsf{X}_2})^{-1}(A)$.

3.3.4 Pseudo-Marginal MCMC

As a *pseudo-marginal* kernel, we refer to any MCMC kernel used to approximate a 'marginal' distribution $\pi^\star \propto \gamma^\star \in \mathcal{M}(\Theta)$, where

$$\gamma^\star(\mathrm{d}\theta) := \varpi(\mathrm{d}\theta)\gamma(\theta, \mathbb{1}),$$

by targeting an extended distribution

$$\pi(\mathrm{d}\theta \times \mathrm{d}v) := \gamma^\star(\mathrm{d}\theta)\widetilde{T}(\theta, \mathrm{d}v)v,$$

as defined in Subsection 1.4.3.

Note that this entails using the generic extended target measure from Chapter 1 three times. Once for justifying the overall MCMC approximation (Subsection 3.1.4), once for constructing the particular MCMC kernel P (Subsection 3.2.3) and now a third time for building the distribution π which with respect to which the kernel P is invariant.

Illustration. To give a more concrete description of the resulting MCMC kernel, assume again the particular factorisation of the extended target distribution (associated with the MCMC kernel) given in Assumption 3.11. For simplicity, assume that $\mathsf{Y} = \emptyset$. In addition, set $\mathsf{X} := \Theta \times \mathsf{V}$, where $\mathsf{V} := [0, \infty)$, and write $X := (\varTheta, V)$, $X^k = (\varTheta^k, V^k)$ and $\boldsymbol{\varTheta} = \varTheta^{1:N}$ as well as $\boldsymbol{V} = V^{1:N}$. The crucial ingredient of pseudo-marginal MCMC kernels is then a proposal kernel with the following factorisation:

$$Q_k(x, \mathrm{d}\boldsymbol{x}^{-k}) := S_k(\theta, \mathrm{d}\boldsymbol{\theta}^{-k}) \prod_{n \in \mathsf{K}\setminus\{k\}} \widetilde{T}(\theta^k, \mathrm{d}v^k),$$

where $S \in \mathcal{K}_1(\mathsf{X} \times \mathsf{K}, \Theta^N)$ is some suitable proposal kernel for $\varTheta$ and where we write $S((\theta, k), \cdot\,) =: S_k(\theta, \cdot\,)$, for simplicity.

The conditional distribution of (X, K) under $\bar{\pi}$ then simplifies to

$$\begin{aligned}\bar{\pi}_{\boldsymbol{x}}(\{k\}) &= \frac{\gamma(\mathrm{d}x^k)Q_k(x^k, \mathrm{d}\boldsymbol{x}^{-k})}{\sum_{n=1}^N \gamma(\mathrm{d}x^n)Q_n(x^n, \mathrm{d}\boldsymbol{x}^{-n})}\\ &= \frac{\gamma^\star(\mathrm{d}\theta^k)S_k(\theta^k, \mathrm{d}\boldsymbol{\theta}^{-k})v^k}{\sum_{n=1}^N \gamma^\star(\mathrm{d}\theta^n)S_n(\theta^n, \mathrm{d}\boldsymbol{\theta}^{-n})v^n}. \end{aligned} \tag{3.12}$$

In other words, if $V \equiv 1$, then the probability in Equation 3.12 reduces to that obtained from some MCMC algorithm which directly works on the marginal space, Θ and which at each iteration, proposes $N-1$ candidates according to the kernel S to target the marginal distribution $\pi^\star$. In the particular context of MCMC methods, this property can be powerful because the mixing properties of algorithms targeting this (usually intractable) marginal distribution – and whose properties the pseudo-marginal algorithm seeks to mimic by controlling the oscillations of V – can be substantially better than those of MCMC algorithms targeting some (tractable) extended distribution.

Pseudo-Marginal MH Kernels. Pseudo-marginal methods were introduced by Beaumont (2003), Andrieu and Roberts (2009) and – in the context of MCMC methods – they are usually implemented by targeting the pseudo-marginal distribution using a standard MH kernel.

More precisely, assume in particular that that $N = 2$. Then under Assumption 3.11, the MH algorithm (Algorithm 3.14) can be restated as in Algorithm 3.18, where the term $\alpha_{\boldsymbol{x}}(k,l)$ in the acceptance probability can be expressed as

$$\alpha_{\boldsymbol{x}}(k,l) = \frac{\bar{\pi}_{\boldsymbol{x}}(\{l\})}{\bar{\pi}_{\boldsymbol{x}}(\{k\})} = \frac{\gamma^\star(\mathrm{d}\theta^l) S_l(\theta^l, \mathrm{d}\boldsymbol{\theta}^{-l})}{\gamma^\star(\mathrm{d}\theta^k) S_k(\theta^k, \mathrm{d}\boldsymbol{\theta}^{-k})} \frac{v^l}{v^k}. \tag{3.13}$$

3.18 Algorithm (pseudo-marginal MH). *Given $X \sim \pi$,*

(1) sample $K \sim \mathrm{Unif}_{\mathrm{K}}$ and $\boldsymbol{\Theta}^{-k} \sim S_k(\theta, \cdot)$,

(2) sample $\boldsymbol{V}^{-k} \sim \widetilde{T}(\boldsymbol{\theta}^{-k}, \cdot)$ and set $X^k := X$,

(3) set $\widetilde{K} = l$ with probability

$$\widetilde{P}_{\boldsymbol{z}}(k,\{l\}) := \begin{cases} 1 \wedge \alpha_{\boldsymbol{x}}(k,l), & if l \neq k, \\ 1 - 1 \wedge \alpha_{\boldsymbol{x}}(k,l), & if l = k, \end{cases}$$

(4) output $\widetilde{X} = X^{\tilde{k}}$.

As before, in this simplified case (due to the exchangeability implied by Assumption 3.11), it suffices to set $K := 1$, say, in Step 1.

A thorough analysis of pseudo-marginal MH kernels has been undertaken by Doucet, Pitt, Deligiannidis and Kohn (2015), Andrieu and Vihola

(2015, 2014), Sherlock, Thiéry, Roberts and Rosenthal (2015) who also derive guidelines for the optimal value of tuning parameters which govern the oscillations of the 'noise', V, in this context.

3.19 Example (particle marginal MH). *The* particle marginal Metropolis–Hastings *(PMMH) kernel from Andrieu et al. (2010) is immediately justified by recalling that SMC methods can be viewed as a special case of the MOSIS framework from Chapter 1. More precisely, the PMMH kernel is a special case of Algorithm 3.18 with*

$$\widetilde{T}(\theta, \cdot) := \bar{\psi}(\theta, \cdot) \circ (\bar{w}^\theta)^{-1},$$

where $\bar{\psi}(\theta, \cdot)$ and $\bar{w}^\theta$ represent the extended proposal distribution and the Radon–Nikodým derivative from Chapter 2. These are now indexed by θ. In other words, at every iteration, the PMMH kernel employs an SMC algorithm which, conditionally on the proposed value of Θ, proposes the 'weight' V^l in the acceptance probability in Equation 3.13. *More precisely, the weight V^l is simply the usual SMC-based estimate of the normalsing constant $\gamma(\theta, \mathbb{1})$ of the measure $\gamma(\theta, \cdot)$ which is marginally targeted by the SMC algorithm.*

3.20 Example (random refreshment). *The* random refreshment *algorithm from Maire et al. (2014, Algorithm 3) can be viewed as applying the pseudo-marginal MH kernel twice but taking $S_k(\theta, \cdot) := \delta_\theta$ in the first application. This kernel obviously dominates a single application of the pseudo-marginal MH kernel in terms of not inducing a larger asymptotic variance.*

Other Pseudo-Marginal MCMC Kernels. The pseudo-marginal distribution can also be targeted by other (more general) instances of the generic MCMC kernel from Subsection 3.2.3. That is, the pseudo-marginal idea can still be used when the kernel includes extra auxiliary variables Y (leading to 'randomised' pseudo-marginal MH kernels) or if it uses $N > 2$ candidates (leading to multiple-proposal pseudo-marginal kernels).

For instance, using (iterated) conditional sequential Monte Carlo kernels – these are described in the next section – based around pseudo-marginal SMC algorithms (see Subsection 2.3.5) can be viewed as more complicated (non-MH) pseudo-marginal MCMC kernels. The latter are also 'randomised' pseudo-marginal kernels due to the use of a a number of further auxiliary variables (e.g. parent indices) within SMC algorithms.

3.3.5 Ensemble MCMC

The *ensemble MCMC* methods from Neal (2003, 2011), Shestopaloff and Neal (2013) are based around a particular instance of the generic MOSIS target measure $\bar{\gamma}$ from Section 1.4 which we describe below. Specifically, they describe two approaches.

Single-sequence ensemble MCMC kernels, described in Neal (2011) and also called *embedded HMM* methods in Neal (2003), can be viewed as building a multiple-proposal MCMC kernel around $\bar{\gamma}$. This leads to a special case of the generic MCMC kernel from Section 3.2.3. We show that this method bears some resemblance – upon which we further elaborate below – to the conditional sequential Monte Carlo kernel with backward sampling which we describe in the next section.

Pseudo-marginal ensemble MCMC kernels, described in Shestopaloff and Neal (2013, 2014), are special cases of the pseudo-marginal MH kernel described in the previous subsection. They employ the particular instance of the measure $\bar{\gamma}$ at a lower level to construct a particular kind of pseudo-marginal target distribution which is then targeted using a standard MH kernel. We show that this method can be interpreted as a particular PMMH kernel in which all the parent indices are integrated out analytically.

Setting. Let $\gamma_T \in \mathcal{M}(\mathsf{X}_{1:T}^{\times})$ be some target measure, let $\pi_T := \gamma_T/\gamma_T(\mathbb{1})$ denote the target distribution, and let $\Gamma_t \in \mathcal{K}(\mathsf{X}_{1:t-1}^{\times}, \mathsf{X}_t)$ be finite kernels such that $\gamma_T = \Gamma_{1:T}^{\otimes}$.

Ensemble MCMC methods propose a pool of N_t possible candidates for the tth component of this target measure. In contrast to SMC methods, the candidates for different components are proposed independently. Given the pool of candidates, single-sequence ensemble MCMC kernels select one possible candidate for each component. In contrast, pseudo-marginal ensemble MCMC kernels average over all available candidates.

Extended Target Distribution. We describe both types of ensemble MCMC approaches in the following. First, we discuss the particular instance of the generic extended measure $\bar{\gamma}$ around which they are based. We let $\mathsf{K}_t := \mathbb{N}_{N_t}$, for some $N_t \in \mathbb{N}$, $N_t > 1$, and set $\mathsf{Y} := \emptyset$ (i.e. we do not use any extra auxiliary variables).

The extended target measure employed by ensemble MCMC methods, and termed 'ensemble density' in Neal (2011), is then given by the usual factorisation

$$\bar{\gamma}_T(\mathrm{d}x_{1:T}) := \gamma_T(\mathrm{d}x_{1:T})\bar{\Pi}_T(x_{1:T}, \mathrm{d}\bar{z}_T),$$

where

$$\begin{aligned}\bar{\Pi}_T(x_{1:T}, \mathrm{d}\bar{z}_T) := {} & \mathrm{Unif}_{\mathsf{K}_{1:T}^{\times}}(\mathrm{d}k_{1:T})\delta_{x_{1:T}}(\mathrm{d}x_{1:T}^{k_{1:T}}) \\ & \times \prod_{t=1}^{T} \zeta_t^{\mathrm{C}}((k_t, x_t^{k_t}), \mathrm{d}\boldsymbol{x}_t^{-k_t}). \end{aligned} \tag{3.14}$$

Above, for each $t \in \mathbb{N}_T =: \mathsf{T}$, $\zeta_t^{\mathrm{C}} \in \mathcal{K}_1(\mathsf{K}_t \times \mathsf{X}_t, \mathsf{X}_t^{N_t - 1})$ is some stochastic kernel which defines a distribution over the $N_t - 1$ remaining candidates in the 'pool' $\boldsymbol{Z}_t := \boldsymbol{X}_t = X_t^{1:N_t}$.

For later use, let $\zeta_t \in \mathcal{M}_1(\mathbf{X}_t)$, be a probability measure on the space $\mathbf{Z}_t := \mathbf{X}_t := \mathsf{X}_t^{N_t}$, which is such that $\zeta_t^{\mathrm{C}}((k, x), \cdot)$ is the conditional distribution of $\boldsymbol{X}_t^{-k}$ under ζ_t given that the kth component takes the value x. Similarly, let $\zeta_t^{\mathrm{M}}(k, \cdot)$ denote the marginal distribution of the kth component under ζ_t.

3.21 Example. *For instance, Neal (2003) proposes to take $\zeta_t^{\mathrm{M}}(k, \cdot) = \rho_t$ to be some probability measure which is constant in k and sets*

$$\zeta_t^{\mathrm{C}}((k, x), \cdot) := P_t^{\otimes(N_t - k)}(x, \mathrm{d}x_t^{k+1:N_t})L_t^{\otimes(k-1)}(x, \mathrm{d}x_t^{k-1} \times \cdots \times \mathrm{d}x_t^1),$$

where P_t is some ρ_t-invariant MCMC kernel and L_t is the associated time-reversal kernel.

Dominating Measure. Assuming that $\gamma_T \ll \zeta_{1:T}^{\mathrm{M},\otimes}(k_{1:T}, \cdot)$, for any $k_{1:T} \in \mathsf{K}_{1:T}^{\times}$, the dominating measure employed by ensemble MCMC methods exists and can be written as

$$\bar{\psi}_T(\mathrm{d}\bar{\boldsymbol{x}}_T) := \psi_T(\mathrm{d}\boldsymbol{z}_{1:T})\xi_T(\boldsymbol{z}_{1:T}, \mathrm{d}k_{1:T})\delta_{x_{1:T}^{k_{1:T}}}(\mathrm{d}x_{1:T}), \tag{3.15}$$

where $\xi_T \in \mathcal{K}_1(\mathbf{Z}_{1:T}^{\times}, \mathsf{K}_{1:T}^{\times})$ is some suitable stochastic kernel (whose support is a large-enough subset of $\mathsf{K}_{1:T}^{\times}$) and where $\psi_T := \zeta_{1:T}^{\otimes}$ was termed 'ensemble base measure' in Neal (2011).

Radon–Nikodým Derivative. For the above-mentioned extended target measure and associated dominating measure, we obtain the following simple Radon–Nikodým derivative $\bar{w}_T := \mathrm{d}\bar{\gamma}_T / \mathrm{d}\bar{\psi}_T$, given by

$$\begin{aligned} \bar{w}_T(\bar{\boldsymbol{x}}_T) = {} & \mathbb{1}_{\{x_{1:T}\}}(x_{1:T}^{k_{1:T}}) \\ & \times \frac{\mathrm{dUnif}_{\mathsf{K}_{1:T}^{\times}}}{\mathrm{d}\xi_T(\boldsymbol{z}_{1:T}, \cdot)}(k_{1:T}) \frac{\mathrm{d}\gamma_T}{\mathrm{d}\zeta_{1:T}^{\mathrm{M},\otimes}(k_{1:T}, \cdot)}(x_{1:T}^{k_{1:T}}). \end{aligned} \tag{3.16}$$

For $\bar{\boldsymbol{X}}_T \sim \bar{\psi}_T$, if we again write

$$\begin{aligned} w^{k_{1:T}}(\boldsymbol{Z}_{1:T}) & := \mathbb{E}\big[\bar{w}_T(\bar{\boldsymbol{X}}_T)\, \mathbb{1}_{\{k_{1:T}\}}(K_{1:T}) \big| \boldsymbol{Z}_{1:T}\big] \\ & = \left[\prod_{t=1}^{T} N_t\right]^{-1} \frac{\mathrm{d}\gamma_T}{\mathrm{d}\zeta_{1:T}^{\mathrm{M},\otimes}(k_{1:T}, \cdot)}(X_{1:T}^{k_{1:T}}), \end{aligned}$$

then the full conditional distribution of $(X_{1:T}, K_{1:T})$ under $\bar{\pi}_T$ can be written as $\bar{\pi}_T^{\mathrm{C}}(\boldsymbol{z}_{1:T}, \mathrm{d}k_{1:T} \times \mathrm{d}x_{1:T}) = \bar{\pi}_{T,\boldsymbol{z}_{1:T}}(\mathrm{d}k_{1:T})\delta_{x_{1:T}^{k_{1:T}}}(\mathrm{d}x_{1:T})$, where

$$\bar{\pi}_{T,\boldsymbol{z}_{1:T}}(\{k_{1:T}\}) := \frac{w^{k_{1:T}}(\boldsymbol{z}_{1:T})}{\sum_{n_{1:T} \in \mathsf{K}_{1:T}^{\times}} w^{n_{1:T}}(\boldsymbol{z}_{1:T})}.$$

Single-Sequence Ensemble MCMC. The single-sequence ensemble or embedded HMM method from Neal (2003, 2011) is an instance of the generic MCMC kernel from Subsection 3.2.3. More specifically, by adding a large number of degenerate copies of the candidates and by using a suitable reparametrisation, it can be viewed as an instance of Barker's kernel with $N = \prod_{t=1}^{T} N_t$ candidates. In algorithmic form, it may be summarised as follows.

3.22 Algorithm (single-sequence ensemble MCMC). *For* $X_{1:T} \sim \pi_T$,

(1) *sample* $K_{1:T} \sim \mathrm{Unif}_{\mathsf{K}_{1:T}^{\times}}$ *and set* $X_{1:T}^{k_{1:T}} := X_{1:T}$,

(2) *for* $t \in \mathsf{T}$, *sample* $\boldsymbol{X}_t^{-k_t} \sim \zeta_t^{\mathrm{C}}((k_t, x_t^{k_t}), \cdot)$,

(3) *sample* $\widetilde{K}_{1:T} = l_{1:t}$ *with probability* $\bar{\pi}_{T,\boldsymbol{z}_{1:T}}(\{l_{1:T}\})$,

(4) *output* $\widetilde{X}_{1:T} = X_{1:T}^{\tilde{k}_{1:T}}$.

Note that the computational cost of sampling $\widetilde{K}_{1:T}$ in Step 3 of this Algorithm will generally grow exponentially in T. However, in the setting studied in Neal (2003, 2011), the kernels $\varGamma_t$ are Markov, i.e. $\varGamma_t(x_{1:t-1}, \cdot)$ is constant in $x_{1:t-2}$. As a result, the usual forward–backward recursions (Rauch et al., 1965; Baum et al., 1970) can reduce the computational complexity of Step 3 to $\mathcal{O}(TN^2)$ (assuming $N_1 = \ldots = N_T = N$).

Pseudo-Marginal Ensemble MCMC. We now turn to the ensemble MCMC approach from Shestopaloff and Neal (2013) which may be viewed as a special case of the pseudo-marginal MH kernel described in the previous subsection.

More precisely, the extended target measure defined by Equation 3.14 is now used to construct a particular instance of the pseudo-marginal target measure from Subsection 1.4.3 and hence to define a particular distribution π with respect to which the MH kernel is invariant. Note that this is in contrast to the single-sequence method in which the extended target measure was used at a 'higher level' to construct the π-invariant MCMC kernel P.

As in Subsection 3.3.4, we assume that we actually want to approximate some 'marginal' distribution $\pi_T^\star \propto \gamma_T^\star \in \mathcal{M}(\Theta)$, where

$$\gamma_T^\star(\mathrm{d}\theta) := \varpi(\mathrm{d}\theta)\gamma_T(\theta, \mathbb{1}).$$

In this case, $\gamma_T(\theta, \cdot)$ is the target distribution used for the single-sequence ensemble MCMC method but which may now depend on $\varTheta$. The probability measure $\varpi \in \mathcal{M}_1(\Theta)$ can often be viewed as a prior distribution on the parameter $\varTheta$.

As before, to circumvent intractabilities in the acceptance probability, the MH algorithm may be viewed as targeting the extended distribution π_T, proportional to the extended measure $\varpi(\mathrm{d}\theta)\widetilde{T}(\theta, \mathrm{d}v)v$. Here, $\widetilde{T}(\theta, \cdot) := \bar{\psi}_T(\theta, \cdot) \circ (\overline{w}_T^\theta)^{-1}$ where $\bar{\psi}_T(\theta, \cdot)$ and $\overline{w}_T^\theta$ are the dominating measure and the Radon–Nikodým derivative from Equations 3.15 and 3.16 but which may now depend on $\varTheta$.

By Proposition 1.13, taking $\xi_T(\boldsymbol{z}_{1:T}, \{k_{1:T}\}) := \bar{\pi}_{T,\boldsymbol{z}_{1:T}}(\{k_{1:T}\})$ means that the random variable governing the 'noise' in the noisily evaluated target density, $V^\theta \sim \widetilde{T}(\theta, \cdot)$ is given by the usual estimate of the normalising constant, i.e. by

$$V^\theta = \overline{w}^\theta(\bar{\boldsymbol{X}}_T) = \sum_{k_{1:T} \in \mathsf{K}_{1:T}^\times} w_T^{k_{1:T}}(\theta, \boldsymbol{Z}_{1:T}).$$

Here, the notation $w_T^{k_{1:T}}(\theta, \boldsymbol{Z}_{1:T})$ is used to indicate dependence on $\varTheta$.

Comparison With Particle MCMC. We conclude this section by offering a comparison between ensemble MCMC methods and the particle MCMC methods introduced by Andrieu et al. (2010). Such a formal comparison has not been undertaken in the literature.

Let ψ_T be the distribution of all random variables generated by an SMC algorithm up to Step T, as defined in Chapter 2. Specifically, assume that the SMC algorithm is as follows. For any $t \in \mathsf{T} := \mathbb{N}_T$,

- $\widetilde{R}_{t-1}(\boldsymbol{z}_{1:t-1}, \cdot) = \mathrm{Unif}_{\mathsf{K}_{t-1}^{N_t}}$, i.e. the resampling distribution is uniform, and, in addition, $O_t \equiv 1$, i.e. we resample at every step,
- $Q_t((\boldsymbol{z}_{1:t-1}, \boldsymbol{a}_{t-1}), \cdot) = \zeta_t$, i.e. the particles are proposed independently of the particles generated at previous steps (and of the parent indices).

In this case, the single-sequence ensemble MCMC method can be seen as an instance of the CSMC algorithm which we describe in the next section, but with a non-standard backward-sampling recursion. This recursion may be viewed as performing backward sampling after having analytically integrated out the parent indices.

Similarly, the pseudo-marginal ensemble MCMC method can be viewed as a PMMH algorithm (again based around the particular kind of SMC algorithm specified above) but one in which all the parent indices are integrated out when forming the estimate of the normalising constant.

Note that ensemble MCMC methods do not make use of the fundamental insight from Andrieu and Roberts (2009) described in Remark 1.16. More precisely, the extended target measure is constructed by extending the measure γ_T using the stochastic kernel $\zeta_{1:T}^{\mathrm{c},\otimes}((k_{1:T}, x_{1:T}), \cdot)$ which is a full conditional distribution under the joint proposal distribution (the 'ensemble base measure') ψ_T. As a result, evaluating the importance weights requires evaluating (a density with respect to) the marginal proposal distribution, $\zeta_{1:T}^{\mathrm{M},\otimes}((k_{1:T}), \cdot)$ (see Remark 1.16).

The need for the marginal proposal distribution to be tractable is exactly why the ensemble base measure, ψ_T, does not allow for dependence between $\boldsymbol{X}_t$ and $\boldsymbol{X}_{t+1}$. Unfortunately, this impedes the efficiency of ensemble MCMC methods whenever the tth and $(t+1)$th component are highly correlated under the target measure, γ_T. Nonetheless, Shestopaloff and Neal (2013) specifically stress the benefits of ensemble MCMC methods over other MCMC methods for models which exhibit such a strong dependence structure.

In contrast, as pointed out in Remarks 2.7 and 2.8, SMC methods only require evaluations of (densities with respect to) a certain conditional distribution under the joint proposal distribution. As a result, they are able to employ a proposal distribution with a much more complicated dependence structure. For instance, particles proposed at successive steps are almost always correlated under the law of the SMC algorithm, ψ_T, even without resampling. Indeed, it is precisely this complicated dependence structure which can lead to efficient proposal distributions on high-dimensional spaces.

Meanwhile, ensemble MCMC methods are only justified by assuming 'computational short-cuts'. More precisely, for the single-sequence method, let Θ be some other parameter which may be updated using another MCMC kernel. Neal (2011) then justifies these methods by assuming that $X_t^{1:N}$ are 'fast' variables so that sampling (and evaluating densities associated with) $(\Theta, X_t^{1:N})$ can be computationally faster than N times sampling (and evaluating densities associated with) the pair (Θ, X_t^1).

3.4 Conditional SMC Kernels

3.4.1 Iterated CSMC Kernel

In this section, we describe MCMC kernels based around the *conditional sequential Monte Carlo* (CSMC) kernel from Equation 2.4. In particular, we provide a unified framework for variance-reduction techniques for CSMC-based algorithms termed backward sampling (Whiteley, 2010) and ancestor sampling (Lindsten, Jordan & Schön, 2014): we show that both target the same extended distribution, $\tilde{\pi}_T := \tilde{\gamma}_T / \tilde{\gamma}_T(\mathbb{1})$. To our knowledge, this is a new result. Finally, we comment on the use of CSMC kernels within particle Gibbs samplers.

Throughout this section, we assume that we are running an SMC algorithm up to $T \in \mathbb{N}$ steps. To construct $\tilde{\pi}_T$, we include an additional set of particles $\widetilde{X} = V_{1:T}$ and particle indices $\widetilde{K} = C_{1:T}$ into the state space. Both of these will be such that $V_{1:T}$ coincides with the particles with indices $C_{1:T}$ generated under the SMC algorithm, i.e. $V_{1:T} = X_{1:T}^{C_{1:T}}$. Throughout this section, we re-use a substantial amount of notation from Subsection 2.4.2. This is deliberate because in the case of CSMC with backward sampling, discussed in the next subsection, the extended measure $\tilde{\gamma}_T$ is exactly the extended target measure associated with the forward filtering–backward smoothing approximation from Subsection 2.4.2.

In particular, in this subsection, we present the basic iterated CSMC algorithm from Andrieu et al. (2010) which forces the particles $V_{1:T}$ to coincide with a particle lineage under the SMC algorithm. That is, we have $C_{1:T} = B^n_{1:T|T}$, for some $n \in \mathsf{K}_T$. The variance-reduction techniques described in the next subsection are based around relaxing this condition.

To construct a π_T-invariant MCMC kernel based on the CSMC kernel $\bar{\Pi}_T^{\text{CSMC}} \in \mathcal{K}_1(\mathsf{X}^{\times}_{1:T}, \mathbf{Z}^{\times}_{1:T})$ from Equation 2.4, we let

$$\begin{aligned}
&\Xi_T(\boldsymbol{z}_{1:T}, \mathrm{d}x_{1:T} \times \mathrm{d}k_{1:T}) \\
&\qquad = \xi_{T|T}(\boldsymbol{z}_{1:T}, \mathrm{d}k_T)\left[\prod_{t=1}^{T-1} \delta_{a_t^{k_{t+1}}}(\mathrm{d}k_t)\right]\delta_{x_{1:T}^{k_{1:T}}}(\mathrm{d}x_{1:T})
\end{aligned}$$

be the stochastic kernel from Equation 2.2.

Throughout this section, $\xi_{T|T}(\boldsymbol{z}_{1:T}, \{k\}) := W_T^k(\boldsymbol{z}_{1:T})$, for $k \in \mathsf{K}_T$. Take $\bar{\pi}_T := \bar{\gamma}_T/\bar{\gamma}_T(\mathbb{1})$ to be the extended target distribution associated with the generic SMC algorithm from Chapter 2. Then by Proposition 2.10,

$$\begin{aligned}
\tilde{\pi}_T(\mathrm{d}\tilde{\boldsymbol{x}}_T) &:= \bar{\pi}_T(\mathrm{d}\boldsymbol{z}_{1:T} \times \mathrm{d}u_{1:T} \times \mathrm{d}b_{1:T})\Xi_T(\boldsymbol{z}_{1:T}, \mathrm{d}v_{1:T} \times \mathrm{d}c_{1:T}) \\
&= \psi_T(\mathrm{d}\boldsymbol{z}_{1:T})\mathfrak{z}_T^{\text{SMC},N_{1:T}}\,\Xi_T(\boldsymbol{z}_{1:T}, \mathrm{d}u_{1:T} \times \mathrm{d}b_{1:T}) \\
&\quad \times \Xi_T(\boldsymbol{z}_{1:T}, \mathrm{d}v_{1:T} \times \mathrm{d}c_{1:T}) \\
&= \bar{\pi}_T(\mathrm{d}\boldsymbol{z}_{1:T} \times \mathrm{d}v_{1:T} \times \mathrm{d}c_{1:T})\Xi_T(\boldsymbol{z}_{1:T}, \mathrm{d}u_{1:T} \times \mathrm{d}b_{1:T}).
\end{aligned}$$

The further extended distribution $\tilde{\pi}_T$ therefore satisfies Equation 3.5, i.e. if $\widetilde{\boldsymbol{X}} \sim \tilde{\pi}_T$, then $V_{1:T} \sim \pi_T$. As a result, the following algorithm induces a π_T-invariant kernel, called the *iterated CSMC* kernel.

3.23 Algorithm (iterated CSMC). *Given $X = U_{1:T} \sim \pi_T$,*

(1) sample $(B_{1:T}, \boldsymbol{Z}_{1:T}) \sim \bar{\Pi}_T^{\text{CSMC}}(u_{1:T}, \cdot)$,

(2) sample $(V_{1:T}, C_{1:T}) \sim \Xi_T(\boldsymbol{z}_{1:T}, \cdot)$,

(3) return $X := V_{1:T}$.

Unfortunately, as discussed by Fearnhead (2010), Whiteley (2010), the sample-impoverishment phenomenon discussed in Subsection 2.4.1 can lead to slow mixing of this kernel. That is, all N_T particle lineages under ψ_T will often share a common ancestor, i.e. there is some $t < T$ such that $B^n_{1:t|T} = U_{1:t}$, for all $n \in \mathsf{K}_T$.

In this case (at least) the first t particles of the path $U_{1:T}$ coincide with the first t particles of the path $V_{1:T}$. It is well known (Andrieu, Lee & Vihola, 2013; Lindsten, Douc & Moulines, 2015) and has been observed empirically (e.g. Fearnhead, 2010) that, in sufficiently ergodic models, the number of particles needs to scale at least linearly in T to control the probability of such a coalescence events and hence the asymptotic variance associated with the MCMC approximation.

3.4.2 Variance-Reduction Techniques

In this subsection, we describe two variance-reduction techniques for iterated CSMC algorithms known as backward sampling and ancestor sampling. The main idea of these techniques is to allow $C_{1:T} \neq B^n_{1:T|T}$, for any $n \in \mathsf{K}_T$ so that $V_{1:T}$ does not need to coincide with any particle lineage at Step T.

Backward and Ancestor Sampling Weights. Before describing the variance-reduction techniques we recall some notation from Subsection 2.4.2

For $t \in \mathsf{T} := \mathbb{N}_T$, we again define the following kernels, termed *backward sampling weights* or *ancestor sampling weights* in this section,

$$w^k_{t|T}(\boldsymbol{z}_{1:t}, v_{t+1:T}) := w^k_t(\boldsymbol{z}_{1:t}) \frac{\mathrm{d}\varGamma^{\otimes}_{t+1:T}(x^{b^k_{1:t|t}}_{1:t}, \cdot)}{\mathrm{d}\lambda^{\otimes}_{t+1:T}(x^{b^k_{1:t|t}}_{1:t}, \cdot)}(v_{t+1:T}), \tag{3.17}$$

for $k \in \mathsf{K}_t$, where the kernels $\lambda_t \in \mathcal{K}_\sigma(\mathsf{X}^{\times}_{1:t-1}\mathsf{X}_t)$ define some suitable dominating measure. We also define the self-normalised versions

$$W^k_{t|T}(\boldsymbol{z}_{1:t}, v_{t+1:T}) := \frac{w^k_{t|T}(\boldsymbol{z}_{1:t}, v_{t+1:T})}{\sum_{n=1}^{N_t} w^n_{t|T}(\boldsymbol{z}_{1:t}, v_{t+1:T})}.$$

We again note that $w^k_{T|T}(\boldsymbol{z}_{1:T}) = w^k_T(\boldsymbol{z}_{1:T})$.

Backward Sampling. First introduced by Whiteley (2010), *backward sampling* (BS) replaces the standard kernel Ξ_T by the slightly more general kernel $\Xi_T^{\mathrm{BS}} \in \mathcal{K}_1(\boldsymbol{Z}_{1:T}^{\times}, \mathsf{X}_{1:T}^{\times} \times \mathsf{K}_{1:T}^{\times})$ from Subsection 2.4.2 which we recall was given by

$$\begin{aligned}
&\Xi_T^{\mathrm{BS}}(\boldsymbol{z}_{1:T}, \mathrm{d}v_{1:T} \times \mathrm{d}c_{1:T}) \\
&\quad := \prod_{t=1}^{T} \xi_{t|T}((\boldsymbol{z}_{1:t}, v_{t+1:T}, c_{t+1:T}), \mathrm{d}c_t)\delta_{x_t^{c_t}}(\mathrm{d}v_t),
\end{aligned}$$

where, for any $t \in \mathsf{T} \setminus \{T\}$,

$$\begin{aligned}
&\xi_{t|T}((\boldsymbol{z}_{1:t}, v_{t+1:T}, c_{t+1:T}), \{c_t\}) \\
&\quad := \begin{cases} \delta_{a_t^{c_{t+1}}}(\{c_t\}), & \text{if } \varrho_t(o_t) = 0, \\ W_{t|T}^{c_t}(\boldsymbol{z}_{1:t}, v_{t+1:T}), & \text{if } \varrho_t(o_t) = 1, \end{cases}
\end{aligned} \tag{3.18}$$

Here, $\varrho_t \colon \mathsf{O}_t \to \{0, 1\}$ is again some suitable function for interpolating between the 'plain' iterated CSMC kernel from the previous subsection and an iterated CSMC kernel with 'full' backward sampling.

Iterated CSMC with BS is summarised in Algorithm 3.24. The proof that this procedure induces a kernel that is π_T-invariant is postponed to the next subsection.

3.24 Algorithm (iterated CSMC with BS). *Given $X = U_{1:T} \sim \pi_T$,*

(1) sample $(B_{1:T}, \boldsymbol{Z}_{1:T}) \sim \bar{\Pi}_T^{\mathrm{CSMC}}(u_{1:T}, \cdot)$,

(2) sample $(V_{1:T}, C_{1:T}) \sim \Xi_T^{\mathrm{BS}}((b_{1:T}, u_{1:T}, \boldsymbol{z}_{1:T}), \cdot)$,

(3) return $X := V_{1:T}$.

That is, to sample according to the kernel induced by iterated CSMC with BS, we first sample $\bar{\boldsymbol{Z}}_T = (U_{1:T}, B_{1:T}, \boldsymbol{Z}_{1:T})$ according to the standard CSMC kernel. Note that this conditions on particles $U_{1:T} = u_{1:T}$ which are allocated to indices $B_{1:T}$ in such a way that the B_Tth particle lineage at Step T (implied by $\boldsymbol{Z}_{1:T}$) takes values $u_{1:T}$. We then sample a sequence of particle indices $C_{1:T}$ 'backwards'. The collection of particles thus indexed is then labelled $V_{1:t}$. In general, it does not need to coincide with any of the Step-T particle lineages implied by $\boldsymbol{Z}_{1:T}$.

Ancestor Sampling. Introduced by Lindsten, Jordan and Schön (2012, 2014), *ancestor sampling* (AS) keeps the kernel $\varXi_T$ from the plain iterated CSMC algorithm but replaces the standard CSMC kernel, $\bar{\varPi}_T^{\mathrm{CSMC}}$, by $\bar{\varPi}_T^{\mathrm{AS}} \in \mathcal{K}_1(\mathsf{X}_{1:T}^{\times}, \mathbf{Z}_{1:T}^{\times} \times \mathsf{K}_{1:T}^{\times})$, given by

$$
\begin{aligned}
&\bar{\varPi}_T^{\mathrm{AS}}(v_{1:T}, \mathrm{d}\boldsymbol{z}_{1:T} \times \mathrm{d}c_{1:T}) \\
&\quad := \bar{\varPi}_{1|T}^{\mathrm{AS}}(v_{1:T}, \mathrm{d}c_1 \times \mathrm{d}\boldsymbol{z}_1) \\
&\qquad \times \prod_{t=2}^{T} \bar{\varPi}_{t|T}^{\mathrm{AS}}((v_{1:T}, c_{1:t-1}, \boldsymbol{z}_{1:t-1}), \mathrm{d}c_t \times \mathrm{d}\boldsymbol{z}_t),
\end{aligned}
$$

with

$$
\begin{aligned}
&\bar{\varPi}_{1|T}^{\mathrm{AS}}(v_{1:T}, \mathrm{d}c_1 \times \mathrm{d}\boldsymbol{z}_1) \\
&\quad := \varLambda_1(v_1, \mathrm{d}c_1)\delta_{v_1}(\mathrm{d}x_1^{c_1}) q_1^{\mathrm{C}}((c_1, x_1^{c_1}), \mathrm{d}\boldsymbol{x}_1^{-c_1})
\end{aligned}
$$

and

$$
\begin{aligned}
&\bar{\varPi}_{t|T}^{\mathrm{AS}}((v_{1:T}, c_{1:t-1}, \boldsymbol{z}_{1:t-1}), \mathrm{d}c_t \times \mathrm{d}\boldsymbol{z}_t) \\
&\quad := S_{t-1}(\boldsymbol{z}_{1:t-1}, \mathrm{d}o_{t-1})\delta_{v_t}(\mathrm{d}x_t^{c_t}) \\
&\qquad \times \rho_{t-1|t}(v_{1:t}, \boldsymbol{z}_{1:t-1}, o_{t-1}, c_{1:t-1}), \mathrm{d}a_{t-1}^{c_t}) \\
&\qquad \times \varLambda_t((v_{1:t}, \boldsymbol{z}_{1:t-1}, o_{t-1}, a_{t-1}^{c_t}), \mathrm{d}c_t) \\
&\qquad \times R_{t-1}^{\mathrm{C}}((\boldsymbol{z}_{1:t-1}, o_{t-1}, c_t, a_{t-1}^{c_t}), \mathrm{d}\boldsymbol{a}_{t-1}^{-c_t}) \\
&\qquad \times Q_t^{\mathrm{C}}((\boldsymbol{z}_{1:t-1}, o_{t-1}, \boldsymbol{a}_{t-1}, c_t, x_t^{c_t}), \mathrm{d}\boldsymbol{x}_t^{-c_t}).
\end{aligned}
$$

The only difference to the standard CSMC kernel is that the parent index associated with the particle on which we condition at Step $t+1$, $A_t^{C_{t+1}}$, is not deterministically set to C_t but rather sampled from some distribution determined by the stochastic kernel for any $t \in \mathsf{T} \setminus \{T\}$ defined by

$$
\begin{aligned}
&\rho_{t|T}((\boldsymbol{z}_{1:t}, o_t, v_{t+1:T}, c_{1:t}), \{a_t^{c_{t+1}}\}) \\
&\quad = \begin{cases} \delta_{c_t}(\{a_t^{c_{t+1}}\}), & \text{if } \varrho_t(o_t) = 0, \\ W_{t|T}^{a_t^{c_{t+1}}}(\boldsymbol{z}_{1:t}, v_{t+1:T}), & \text{if } \varrho_t(o_t) = 1. \end{cases}
\end{aligned}
$$

As a result, $V_{1:T}$ does not necessarily coincide with any particle lineage under the SMC algorithm. The function ϱ_t is the same is in the case of BS.

More precisely, the only difference between iterated CSMC with AS and 'plain' iterated CSMC is that the former potentially sets $A_{t-1}^{C_t} \neq C_{t-1}$ in Step 2 of Algorithm 3.25 which details the steps entailed in sampling from $\bar{\Pi}_T^{\mathrm{AS}}(v_{1:T}, \cdot)$.

3.25 Algorithm. *At Step* 1*, sample* $C_1 \sim \Lambda_1(v_1, \cdot)$*, set* $x_1^{c_1} := v_1$ *and sample* $\boldsymbol{X}_1^{-C_1} \sim q_1^{\mathrm{c}}((c_1, x_1^{c_1}), \cdot)$. *At Step* t, $t \in \mathbb{Z}_{2,T}$,

(1) sample $O_{t-1} \sim S_{t-1}(\boldsymbol{z}_{1:t-1}, \cdot)$,

(2) sample $A_{t-1}^{C_t} \sim \rho_{t-1|t}(v_{1:t}, \boldsymbol{z}_{1:t-1}, o_{t-1}, c_{1:t-1}), \cdot)$,

(3) sample $C_t \sim \Lambda_t((v_{1:t}, \boldsymbol{z}_{1:t-1}, o_{t-1}, a_{t-1}^{c_t}), \cdot)$ *and set* $x_t^{c_t} = v_t$,

(4) sample $\boldsymbol{A}_{t-1}^{-C_t} \sim R_{t-1}^{\mathrm{c}}((\boldsymbol{z}_{1:t-1}, o_{t-1}, c_t, a_{t-1}^{c_t}), \cdot)$,

(5) sample $\boldsymbol{X}_t^{-C_t} \sim Q_t^{\mathrm{c}}((\boldsymbol{z}_{1:t-1}, o_{t-1}, \boldsymbol{a}_{t-1}, c_t, x_t^{c_t}), \cdot)$.

The complete set of sampling steps needed for performing one iteration of the iterated-CSMC-with-AS kernel is summarised in Algorithm 3.26. Again, the proof that this procedure induces a kernel that is π_T-invariant is postponed to the next subsection.

3.26 Algorithm (iterated CSMC with AS). *Given* $X = V_{1:T} \sim \pi_T$,

(1) sample $(C_{1:T}, \boldsymbol{Z}_{1:T}) \sim \bar{\Pi}_T^{\mathrm{AS}}(v_{1:T}, \cdot)$ *via Algorithm 3.25,*

(2) sample $(U_{1:T}, B_{1:T}) \sim \Xi_T(\boldsymbol{z}_{1:T}, \cdot)$,

(3) return $X := U_{1:T}$.

3.4.3 Duality of Backward and Ancestor Sampling

By construction, iterated CSMC with BS targets the extended distribution

$$\begin{aligned}\tilde{\pi}_T^{\mathrm{BS}}(\mathrm{d}\tilde{\boldsymbol{x}}_T) := {} & \pi_T(\mathrm{d}u_{1:T})\bar{\Pi}_T^{\mathrm{CSMC}}(u_{1:T}, \mathrm{d}\boldsymbol{z}_{1:T} \times \mathrm{d}b_{1:T}) \\ & \times \Xi_T^{\mathrm{BS}}(\boldsymbol{z}_{1:T}, \mathrm{d}c_{1:T} \times \mathrm{d}v_{1:T}).\end{aligned}$$

Similarly, iterated CSMC with AS targets the extended distribution

$$\begin{aligned}\tilde{\pi}_T^{\mathrm{AS}}(\mathrm{d}\tilde{\boldsymbol{x}}_T) := {} & \pi_T(\mathrm{d}v_{1:T})\bar{\Pi}_T^{\mathrm{AS}}(v_{1:T}, \mathrm{d}\boldsymbol{z}_{1:T} \times \mathrm{d}c_{1:T}) \\ & \times \Xi_T(\boldsymbol{z}_{1:T}, \mathrm{d}b_{1:T} \times \mathrm{d}u_{1:T}).\end{aligned}$$

The main focus of this subsection is the following result whose proof can be found at the end of this subsection.

3.27 Proposition. *Iterated conditional sequential Monte Carlo algorithms with backward sampling and ancestor sampling target the same extended distribution, i.e.* $\tilde{\pi}_T := \tilde{\pi}_T^{\text{BS}} = \tilde{\pi}_T^{\text{AS}}$.

Recalling that forward filtering–backward smoothing (as well as its sampling approximation: forward filtering–backward sampling) targets an extended measure whose normalised version coincides with $\tilde{\pi}_T^{\text{BS}}$, Proposition 3.27 implies that it leads to unbiased estimates of integrals of the form $\gamma_T(f_T)$. More importantly, Proposition 3.27 immediately guarantees that iterated CSMC kernels with BS and AS both leave π_T invariant, as formalised in the following corollary.

3.28 Corollary. *If* $\widetilde{X}_T \sim \tilde{\pi}_T$, *then* $V_{1:T} \sim \pi_T$ *and* $U_{1:T} \sim \pi_T$. □

3.29 Remark. *Corollary 3.28 might also be useful for analysing and perhaps comparing the convergence properties of iterated CSMC algorithms with BS and with AS. This could be done by analysing the expectation of the usual normalising-constant estimate under a 'doubly-conditional' SMC algorithm to which Corollary 3.28 gives rise. This 'doubly-conditional' SMC algorithm is slightly more general than the one from Andrieu et al. (2013).*

To summarise, recall that $U_{1:T}$ forms a particle lineage under the SMC algorithm but $V_{1:T}$ does not (necessarily). The above-mentioned CSMC kernels with BS and AS can thus be interpreted as follows.

Iterated CSMC with BS samples from the conditional distribution of all random variables under $\tilde{\pi}_T$ given $V_{1:T} = v_{1:T}$. The procedure yields a new set of particles $U_{1:T}$.

Iterated CSMC with AS samples from the conditional distribution of all random variables under $\tilde{\pi}_T$ given $U_{1:T} = u_{1:T}$. The procedure yields a new set of particles $V_{1:T}$.

The remainder of this subsection is devoted to proving Proposition 3.27. To that end, we first state the following technical lemma.

3.30 Lemma. *For* $v_{1:T} \in \mathsf{X}_{1:T}^{\times}$, *use the convention that for any integer* k,

$$w_{0|T}^{k}(v_{1:T}) := \frac{\mathrm{d}\gamma_T}{\mathrm{d}\lambda_{1:T}^{\otimes}}(v_{1:T}),$$

where $\lambda^{\otimes}_{1:T} \in \mathcal{M}_\sigma(\mathsf{X}^{\times}_{1:T})$ is the same dominating measure used in the definition of the BS/AS *weights. With some abuse of notation, for $t > 1$, write $\beta_t(k, l)$ as a shorthand for*

$$\frac{\Lambda_t((\boldsymbol{z}_{1:t-1}, o_{t-1}, k), \{l\})}{R^{\mathrm{M}}_{t-1}((\boldsymbol{z}_{1:t-1}, o_{t-1}, l), \{k\}) Q^{\mathrm{M}}_t((\boldsymbol{z}_{1:t-1}, o_{t-1}, \boldsymbol{a}_{t-1}, l), \mathrm{d}x^l_t)}.$$

Likewise, for $t = 1$ and for any k, write

$$\beta_1(k, l) := \frac{\Lambda_1(\{l\})}{q^{\mathrm{M}}_1(l, \mathrm{d}x^l_1)}.$$

Let $c_{1:T} \in \mathsf{K}^{\times}_{1:T}$ and let $t \in \mathsf{T}$. If there exists $p \in \mathbb{N}_{t-1}$ such that

$$\forall s \in \mathbb{Z}_{p,t-1} : \; a^{c_{s+1}}_s = c_s,$$

(i.e. so that $c_{p:t} = b^{c_t}_{p:t|t}$) then, writing $v_{1:T} = x^{c_{1:T}}_{1:T}$,

$$w^{c_t}_{t|T}(\boldsymbol{z}_{1:t}, v_{t+1:T}) = w^{a^{c_p}_{p-1}}_{p-1|T}(\boldsymbol{z}_{1:p-1}, v_{p:T}) \prod_{s=p}^{t} \beta_s(a^{c_s}_{s-1}, c_s).$$

Proof. This follows from the definition of the BS/AS weights after some tedious but simple algebraic manipulations. □

Proof (of Proposition 3.27). To simplify the notation throughout this proof, we omit all degenerate dependencies and write $b^{b_T}_{t|T} = b_t$, $u_t = x^{b_t}_t$ and $v_t = x^{c_t}_t$, for $t \in \mathsf{T}$. We also define β_t as in Lemma 3.30.

As we assume that $\xi_{T|T}(\boldsymbol{z}_{1:T}, \{k\}) = W^k_T(\boldsymbol{z}_{1:T})$, it suffices to show that

$$\begin{aligned}
\left[\sum_{n=1}^{N_T} w^n_T(\boldsymbol{z}_{1:T})\right] \prod_{t=1}^{T} \xi_{t|T}((\boldsymbol{z}_{1:t}, v_{t+1:T}, c_{t+1:T}), \{c_t\}) \\
= \frac{\mathrm{d}\gamma_T}{\mathrm{d}\lambda^{\otimes}_{1:T}}(v_{1:T}) \left[\prod_{t=1}^{T} \beta_t(a^{c_t}_{t-1}, c_t)\right] \\
\times \prod_{t=1}^{T-1} \rho_{t|T}((\boldsymbol{z}_{1:t}, v_{t+1:T}, c_{1:t}), \{a^{c_{t+1}}_t\}).
\end{aligned} \tag{3.19}$$

Define

$$\bar{\mathsf{O}} := \left\{t \in \mathsf{T} \;\middle|\; \varrho_t(o_t) = 1\right\} \cup \{T\}$$

to be the indices of the CSMC steps for which BS is performed (to which we also count the final step). Using the definition of the kernels $\xi_{t|T}$ from Equation 3.18, left hand side in Equation 3.19 then equals

$$\left[w_T^{c_T}(\boldsymbol{z}_{1:T}) \prod_{t \in \bar{\mathsf{O}} \setminus \{T\}} W_{t|T}^{c_t}(\boldsymbol{z}_{1:t}, v_{t+1:T}) \right] \prod_{t \in \mathsf{T} \setminus \bar{\mathsf{O}}} \delta_{c_t}(\{a_t^{c_{t+1}}\}). \tag{3.20}$$

Let $L := \#\bar{\mathsf{O}}$ be the number of CSMC steps at which backward sampling is performed (to which we again count the final step) and let $t_{1:L}$ denote the indices of these CSMC steps in increasing order. In particular, therefore, $t_L = T$. Additionally, we use the conventions $t_0 := 1$.

By Lemma 3.30 and the conventions defined therein, in particular with the convention that $w_{0|T}^k = \mathrm{d}\gamma_T / \mathrm{d}\lambda_{1:T}^{\otimes}$, for any integer k, the term in the square brackets in Equation 3.20 can then be written as

$$\begin{aligned} &\left[\prod_{l=0}^{L-1} w_{t_l|T}^{a_{t_l}^{c_{t_l+1}}}(\boldsymbol{z}_{1:t_l}, v_{t_l+1:T}) \prod_{t=t_l+1}^{t_{l+1}} \beta_t(a_{t-1}^{c_t}, c_t) \right] \\ &\qquad \times \prod_{t \in \bar{\mathsf{O}} \setminus \{T\}} \left[\sum_{n \in \mathsf{K}_t} w_{t|T}^n(\boldsymbol{z}_{1:t}, v_{t+1:T}) \right]^{-1} \\ &\quad = \frac{\mathrm{d}\gamma_T}{\mathrm{d}\lambda_{1:T}^{\otimes}}(v_{1:T}) \left[\prod_{t=1}^{T} \beta_t(a_{t-1}^{c_t}, c_t) \right] \prod_{t \in \bar{\mathsf{O}} \setminus \{T\}} W_{t|T}^{a_t^{c_{t+1}}}(\boldsymbol{z}_{1:t}, v_{t+1:T}). \end{aligned}$$

This completes the proof. □

3.4.4 Application to Particle Gibbs Samplers

The CSMC kernels described above are usually employed as only one ingredient in a composite MCMC kernel.

That is, we usually want to approximate some 'marginal' distribution $\pi^\star := \gamma^\star / \gamma^\star(\mathbb{1}) \in \mathcal{M}_1(\Theta)$, where

$$\gamma^\star(\mathrm{d}\theta) := \varpi(\mathrm{d}\theta) \gamma_T(\theta, \mathbb{1}).$$

Here, $\varpi \in \mathcal{M}_1(\Theta)$ and, for any $\theta \in \Theta$, $\gamma_T(\theta, \cdot\,) \in \mathcal{M}(\mathsf{X}_{1:T}^{\times})$ is the measure employed throughout this section but is now indexed by θ. However, to avoid calculating the integral $\gamma_T(\theta, \mathbb{1})$, which is often intractable, we devise an MCMC kernel which targets an extended distribution $\pi := \gamma/\gamma(\mathbb{1}) \in \mathcal{M}_1(\Theta \times \mathsf{X}_{1:T}^{\times})$, where

$$\gamma(\mathrm{d}\theta \times \mathrm{d}x_{1:T}) := \varpi(\mathrm{d}\theta)\gamma_T(\theta, \mathrm{d}x_{1:T}).$$

3.31 Example (Bayesian posterior, continued). *In Bayesian statistics, π is often the joint posterior distribution of parameters $(\Theta, X_{1:T})$. In this case, letting $M(\theta, \cdot)$ be a conditional prior distribution of the parameters $X_{1:T}$, the joint prior distribution is given by $\overline{\varpi} := \varpi \otimes M$. Furthermore, $\overline{L} := \mathrm{d}\gamma/\mathrm{d}\overline{\varpi}$ represents the likelihood of $(\Theta, X_{1:T})$ given some observations. Such joint posterior distributions are often more tractable than the marginal posterior distribution given by $\pi^\star(A) := \pi(A \times \mathsf{X}_{1:T}^{\times})$.*

Particle Gibbs Sampler. *Particle Gibbs* (PG) samplers target the distribution π using an MCMC kernel P which is formed by first applying an MCMC kernel which is invariant with respect to the full conditional distribution of Θ under π, with some abuse of notation denoted $\pi(\mathrm{d}\theta|x_{1:T})$, and then applying one of the above-mentioned iterated CSMC kernels (which are invariant with respect to the full conditional distribution of $X_{1:T}$ under π, again denoted with some abuse of notation as $\pi(\mathrm{d}x_{1:T}|\theta) := \gamma_T(\theta, \mathrm{d}x_{1:T})/\gamma_T(\theta, \mathbb{1})$). Algorithm 3.32 presents a single iteration of the PG sampler.

3.32 Algorithm (particle Gibbs). *Given $(\Theta, X_{1:T}) \sim \pi$,*

(1) sample $\widetilde{\Theta}$ from some $\pi(\mathrm{d}\theta|x_{1:T})$-invariant MCMC kernel,

(2) sample $\widetilde{X}_{1:T}$ using Alg. 3.23, 3.24 or 3.26 (targeting $\pi(\mathrm{d}x_{1:T}|\tilde{\theta})$),

(3) return $(\Theta, X_{1:T}) := (\widetilde{\Theta}, \widetilde{X}_{1:T})$.

Novel Auxiliary-Variable Rejuvenation Step. The distribution π targeted by the PG sampler is sometimes itself an 'artificially' extended distribution in the sense that there is some one-to-one reparametrisation

$$(\Theta, X_{1:T}) \longleftrightarrow (\Theta, Z, Y). \tag{3.21}$$

Here, Z is often some collection of random variables which can be interpreted as latent parameters in the model of interest. In contrast, Y is often some collection of random variables which have only been introduced to allow a more flexible (conditional) SMC algorithm to be used.

We let $\hat{\pi}$ be the distribution of the random variables under the parametrisation on the right hand side in Equation 3.21. More specifically,

$$\hat{\pi}(\mathrm{d}\theta \times \mathrm{d}z \times \mathrm{d}y) = \hat{\pi}^{\mathrm{M}}(\mathrm{d}\theta \times \mathrm{d}z) L((\theta, z), \mathrm{d}y).$$

Here $\hat{\pi}^{\mathrm{M}} \in \mathcal{M}_1(\Theta \times \mathrm{Z})$ and $L \in \mathcal{K}_1(\Theta \times \mathrm{Z}, \mathrm{Y})$ while Z and Y take values in Z and Y, respectively. Crucially, we assume that sampling from $L((\theta, z), \cdot)$ is feasible.

3.33 Example. *In the SMC-sampler framework from 2.3.2, $\tilde{\pi}^{\mathrm{M}}$ may the distribution that is actually of interest, $Z = X_T$, $Y = X_{1:T-1}$, and $L = L^{\otimes}_{T-1:1}$ is the tensor product of the backward Markov kernels. In practice, it is often possible to sample from these.*

3.34 Remark. *As mentioned in Example 3.33, the auxiliary variables take a simple form in the case of the SMC-sampler framework. However, the auxiliary-variable rejuvenation idea is more widely applicable. Indeed, in Chapter 4 we apply it to an SMC algorithm which cannot be viewed as a (trivial) SMC sampler.*

Conditioning on the auxiliary variables Y when sampling Θ in Step 1 of the PG sweep described in Algorithm 3.32 can become computationally expensive and can induce slow mixing as soon as $L((\theta, z), \cdot)$ is not constant in θ. In Finke et al. (2014), we proposed an additional PG step that overcomes these potential difficulties. It is summarised in Algorithm 3.35, where we slightly abuse notation by letting $\hat{\pi}^{\mathrm{M}}(\mathrm{d}\theta|z)$ denote the full conditional distribution of Z under $\hat{\pi}^{\mathrm{M}}$.

3.35 Algorithm (PG with auxiliary-variable rejuvenation). *Given a draw $(\Theta, X_{1:T}) \sim \pi$,*

(1) reparametrise $(\Theta, Z, Y) \leftarrow (\Theta, X_{1:T})$,

(2) sample $\widetilde{\Theta}$ from some $\hat{\pi}^{\mathrm{M}}(\mathrm{d}\theta|z)$-invariant MCMC kernel,

(3) sample $\widetilde{Y} \sim L((\theta, z), \cdot)$ and reparametrise $(\widetilde{\Theta}, X_{1:T}) \leftarrow (\widetilde{\Theta}, Z, \widetilde{Y})$,

(4) sample $\widetilde{X}_{1:T}$ using Alg. 3.23, 3.24 or 3.26 (targeting $\pi(\mathrm{d}x_{1:T}|\tilde{\theta})$),

(5) return $(\Theta, X_{1:T}) := (\widetilde{\Theta}, \widetilde{X}_{1:T})$.

This algorithm is justified since the combination of steps Steps 1 to 3 leaves π invariant. Indeed, these steps represent a partially collapsed Gibbs step as described in Example 3.5. It offers three advantages.

(1) By only conditioning on Z, Step 1 can take larger steps in the θ-direction compared to Step 1 in the standard PG algorithm.

(2) Under some SMC algorithms, there is a strong interaction between Z and Y, so that rejuvenating the auxiliary variables Y outside of the CSMC kernel can dramatically improve mixing of the PG sampler. We give an example for this in Chapter 4. Therein, $X_{1:T} = (\phi_{1:T}, m_{1:T})$, where ϕ_t is some parameter and m_t is some finite index. Then, Z includes some subvector of $\phi_{1:T}$, whose choice depends on $m_{1:T}$, and Y denotes the remaining elements of $\phi_{1:T}$ and some other auxiliary variables. In this case, updating Y is beneficial because some of its components may be 'chosen to be part of Z' by the CSMC kernel.

(3) Finally, Algorithm 3.35 comes at little or no extra computational cost. It can even offer computational savings compared to the standard PG scheme, e.g. when each PG sweep updates Θ using the m-fold convolution of a Metropolis–Hastings kernel (as is often done in practice since these updates tend to be relatively inexpensive). Algorithm 3.35 then avoids $m + 1$ evaluations of (densities of) L at the cost of generating one sample from $L((\theta, z), \cdot)$.

3.5 Summary

In this chapter, we have shown that MCMC algorithms can be viewed as a special case of the MOSIS framework described in Chapter 1. Using the same framework at a lower level, we have constructed a generic MCMC kernel. One way of interpreting the relationship between some well-known MCMC kernels for general state spaces mentioned in this chapter is outlined in Figure 3.1.

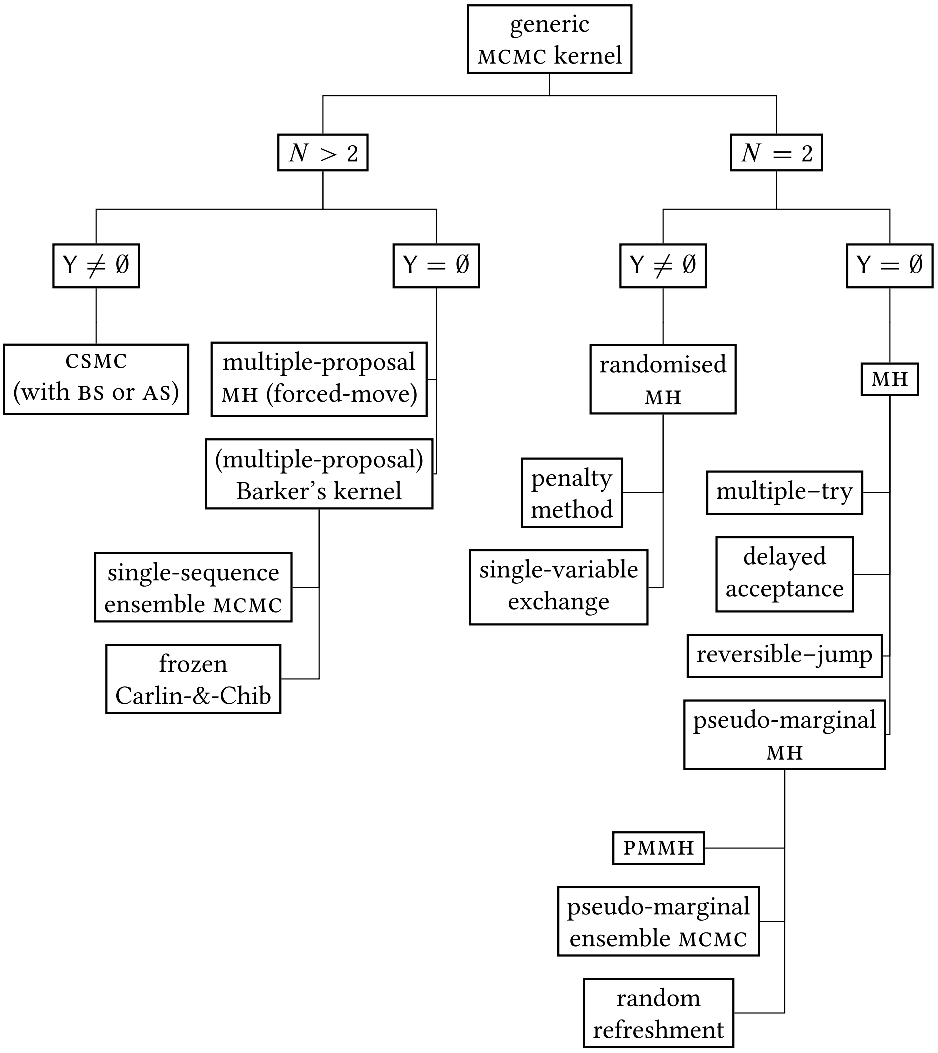

Figure 3.1 Particular instances of the generic MCMC kernel described in this chapter and their relation to the generic MCMC kernel.

Part II

Some Novel Monte Carlo Schemes

4 Inference in Piecewise Deterministic Processes

4.1 Introduction

4.1.1 Motivation

In this chapter, we develop algorithms for conducting inference in discretely observed piecewise deterministic processes. This class of models is defined in Section 4.2 where we also provide motivating examples. Section 4.3 describes an existing sequential Monte Carlo sampler for these models and investigates some of its properties. Section 4.4 derives a novel representation for this algorithm. In addition to ensuring the existence of the importance weights, this representation permits the use of backward sampling and ancestor sampling within particle Gibbs samplers and also allows the use of forward filtering–backward sampling schemes. Section 4.5 provides simulation results and comments on the utility of the novel particle Gibbs step which was presented in Subsection 3.4.4. An extended version of the work presented in this chapter was published as Finke et al. (2014).

A *piecewise deterministic process* (PDP) is a stochastic process that jumps randomly at a countable number of stopping times but otherwise evolves deterministically in continuous time. In this chapter, we employ *sequential Monte Carlo* (SMC)-based methods to conduct inference in PDPs which are observed only partially, noisily and in discrete time. Such models are more general than state-space models and inference for them is often more difficult.

Simple particle filters for PDPs, termed *variable-rate particle filters* (VRPFs), were introduced by Godsill and Vermaak (2004) and a corresponding smoothing algorithm for non-degenerate models was suggested by Bunch and Godsill (2013). To apply more sophisticated particle filtering techniques to these models, an SMC filter for PDPs, based on the SMC-sampler framework from Del Moral et al. (2006b) (see Subsection 2.3.2), was introduced in Whiteley, Johansen and Godsill (2011).

However, methods for efficiently estimating the static parameters in such models still need to be developed. A few approaches have been proposed in the literature. A stochastic expectation–maximisation algorithm based on a *reversible-jump Markov chain Monte Carlo* (RJMCMC) sampler (Green, 1995) was introduced by Centanni and Minozzo (2006a, 2006b). A simple SMC sampler was attempted in Del Moral et al. (2007) to which some improvements were made in Martin, Jasra and McCoy (2013). In addition, Rao and Teh (2013) developed a Gibbs sampler for the special case in which the state space is discrete.

4.1.2 Contribution

We employ a *particle Gibbs* (PG) sampler (Andrieu et al., 2010), based around the SMC filter for PDPs from Whiteley et al. (2011), to estimate the static parameters. Our methodological contributions are as follows.

(1) We provide new insight into the approximation induced by the SMC filter for PDPs and by related algorithms used in Del Moral et al. (2006b, 2007), Martin et al. (2013). We also suggest a way of ensuring the existence of the importance weights.

(2) We derive a new representation of the algorithm that – for non-degenerate models – permits the use of backward sampling and ancestor sampling (Whiteley, 2010; Whiteley et al., 2010; Lindsten et al., 2014) within PG samplers and also allows the use of forward filtering–backward sampling schemes.

(3) We apply the novel PG step for rejuvenating the potentially large number of auxiliary variables used in the SMC filter which was introduced in Subsection 3.4.4. This reduces the impact of these auxiliary variables on the overall mixing of the PG chain at virtually no extra computational cost, often even resulting in computational savings.

We demonstrate our method on two challenging examples. Our simulations indicate that it can compete with both a VRPF-based PG sampler and a RJMCMC sampler, at a potentially lower computational cost. We also empirically investigate the impact of the approximation mentioned in Item 1, above.

4.2 Piecewise Deterministic Processes

4.2.1 Definition

In this section, we introduce discretely-observed piecewise deterministic processes, the class of models with which the remainder of this article is concerned. They are stochastic processes that jump randomly at an almost surely countable number of random times but otherwise evolve deterministically in continuous time. Their description here follows Whiteley et al. (2011). We also provide motivating examples.

Notation. First, we clarify some notational conventions used throughout this chapter. For some spaces Θ and X, and some positive finite kernel $R \in \mathcal{K}(\Theta, \mathsf{X})$, we usually write $R(\theta, \cdot\,) =: R^\theta(\,\cdot\,)$, for any $\theta \in \Theta$. We use the same notation, $R^\theta(x)$, to denote a density of this measure with respect to some σ-finite measure on X, $\mathrm{d}x$, e.g. with respect to a suitable version of the Lebesgue or counting measure, evaluated at some point x. For vectors $x = (x_1, \dots, x_n)$ and some vector of indices $a = (a_1, \dots, a_k)$, where $\{a_1, \dots, a_k\} \subseteq \mathbb{N}_n$, we write $x_a := (x_{a_1}, \dots, x_{a_k})$. Finally, $x_{-a} = x \setminus x_a$ denotes the vector that is identical to x, except that the components $x_{a_1}, \dots, x_{a_k}$ have been removed.

Prior Distribution. Let $(\tau_j, \phi_j)_{j \in \mathbb{N} \cup \{0\}}$ be a stochastic process such that $0 \equiv \tau_0$, such that $\tau_j < \tau_{j+1}$, and such that each ϕ_j takes a value in some non-empty set Φ. In addition, take a (deterministic) function $F^\theta \colon [0, \infty)^2 \times \Phi \to \Phi$ that satisfies $F^\theta(t, t, \cdot\,) = \mathrm{id}$ for any $t \geq 0$. A *piecewise deterministic process* (PDP) is then a continuous-time stochastic process $\zeta := (\zeta_t)_{t \geq 0}$ with initial condition $\zeta_0 = \phi_0$ and with

$$\zeta_t = F^\theta(t, \tau_{\nu_t}, \phi_{\nu_t}),$$

for $t > 0$. Here, we have defined $\nu_t := \sup\{\, j \in \mathbb{N} \cup \{0\} \mid \tau_j \leq t \,\}$, so that τ_{ν_t} represents the time of the last jump before (and including) time t. In other words, after time τ_{j-1}, the PDP evolves deterministically in continuous time according to the function F^θ until it reaches the next *jump time* τ_j, at which point the process randomly jumps to a new value given by the *jump size* ϕ_j. Here and throughout, θ denotes the ordered set of all static parameters present in the model. That is, θ contains all the parameters that do not change over time and which therefore cannot be estimated via standard (particle) filtering methods.

Let $0 = t_0 < t_1 < t_2 < \dots$ be strictly increasing (non-random) times and let $K_n := \nu_{t_n}$ denote the number of jumps before time t_n, with realisations k_n and convention $k_0 = 0$. The process $\zeta_{[0,t_n]} := (\zeta_t)_{t \in [0,t_n]}$ is then completely determined by $(K_n, \tau_{1:K_n}, \phi_{0:K_n})$ (and θ). For simplicity, as in Whiteley et al. (2011), we assume the following Markovian prior on the number, times and sizes of jumps in the interval $[0, t_n]$ for any $n \in \mathbb{N}$,

$$\begin{aligned} p_n^\theta(k_n, \tau_{1:k_n}, \phi_{0:k_n}) = S^\theta(t_n, \tau_{k_n}) q_0^\theta(\phi_0)\, \mathbb{1}_{(0,t_n]}(\tau_{k_n}) \\ \times \prod_{j=1}^{k_n} q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1}) f^\theta(\tau_j | \tau_{j-1}), \end{aligned}$$

where $q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1}) f^\theta(\tau_j | \tau_{j-1})$ forms the Step-j transition kernel of $(\tau_j, \phi_j)_{j \in \mathbb{N} \cup \{0\}}$ with the support of $f^\theta(\tau_j | \tau_{j-1})$ being (τ_{j-1}, ∞). Furthermore, $q_0^\theta(\phi_0)$ is the distribution of the initial jump size, and finally, $S^\theta(t, \tau) := 1 - \int_\tau^t f^\theta(\mathrm{d}s | \tau)$ denotes the probability of no jump occurring in the interval $(\tau, t]$ (for $\tau \leq t$).

Posterior Distribution. Inference for such models becomes necessary if we assume that ζ can be observed only partially, at discrete times, and subject to some measurement error. Observations may be recorded at fixed or random times. Let $y_{(s,t]}$ denote the vector of all observations in the interval $(s, t]$ for some $0 < s < t < \infty$, a density of which is represented by $g^\theta(y_{(s,t]} | \zeta_{(s,t]})$. Again for simplicity, we assume the observations in disjoint time intervals to be conditionally independent given the PDP, though this assumption could easily be relaxed. Hence,

$$\begin{aligned} & g^\theta(y_{(0,t_n]} | \zeta_{(0,t_n]}) \\ & \quad = g^\theta(y_{[\tau_{k_n}, t_n]} | \tau_{k_n}, \phi_{k_n}) \prod_{j=1}^{k_n} g^\theta(y_{[\tau_{j-1}, \tau_j)} | \tau_{j-1}, \phi_{j-1}), \end{aligned}$$

where we sometimes use the notation

$$g^\theta(y_{[\tau_{j-1}, \tau_j)} | \tau_{j-1}, \phi_{j-1}) = g^\theta(y_{[\tau_{j-1}, \tau_j)} | \zeta_{[\tau_{j-1}, \tau_j)})$$

to stress that $\zeta_{[\tau_{j-1}, \tau_j)}$ is conditionally independent of all the other jump times, jump sizes, and the total number of jumps (up to time τ_j), given $(\tau_j, \tau_{j-1}, \phi_{j-1})$ (and given θ).

The conditional-independence property of the observations is reminiscent of state-space models. However, as mentioned earlier, PDPs can be seen as being more general than state-space models. Indeed, state-space models may be viewed as PDPs in which $f^\theta(\tau_j|\tau_{j-1})$ is degenerate, i.e. in which the number of jumps and the jump times are known. Hence, for the remainder of this work, we assume that $f^\theta(\tau_j|\tau_{j-1})$ is non-degenerate.

The conditional posterior distribution of the jumps up to time t_n (as well as their number) may then be defined by the density (with respect to a suitable dominating measure)

$$\begin{aligned}
&\tilde{\pi}_n^\theta(k_n, \tau_{1:k_n}, \phi_{0:k_n}) \\
&\quad = \tilde{\gamma}_n^\theta(k_n, \tau_{1:k_n}, \phi_{0:k_n})/\mathfrak{z}_n^\theta \\
&\quad := p_n^\theta(k_n, \tau_{1:k_n}, \phi_{0:k_n}) g^\theta(y_{(0,t_n]}|\zeta_{(0,t_n]})/\mathfrak{z}_n^\theta, \qquad (4.1)
\end{aligned}$$

where $\mathfrak{z}_n^\theta > 0$ is the normalising constant which is typically unknown.

Variable-Dimension Interpretation. Explicitly including the dimensionality parameter K_n into the state space ensures that for all $n \in \mathbb{N}$, the Step-n posterior distributions are defined on increasing subsets of the same space, i.e. the support of $\tilde{\pi}_n^\theta$ is a subset of

$$\widetilde{\mathsf{E}}_n := \bigcup_{k=0}^{\infty}(\{k\} \times \mathsf{T}_{(0,t_n],k} \times \Phi^{k+1}),$$

where $\mathsf{T}_{(s,t],k} := \{ (\tau_1, \ldots, \tau_k) \in (0,\infty)^k \mid s < \tau_1 < \cdots < \tau_k \leq t \}$. This representation makes the unknown number of jumps in any interval of time, $(0, t_n]$, explicit.

4.2.2 Elementary Change-Point Example

This subsection introduces an elementary change-point model as a first example of a PDP. We assume that the interjump times, $\tau_j - \tau_{j-1}$, are distributed according to some parametric family indexed by a parameter vector θ_τ. Conditional on the jump times, the jump sizes follow a first-order Gaussian autoregressive process, i.e.

$$q^\theta(\mathrm{d}\phi_j|\phi_{j-1}, \tau_j, \tau_{j-1}) = \mathrm{N}_{\rho\phi_{j-1},\sigma_\phi^2}(\mathrm{d}\phi_j),$$

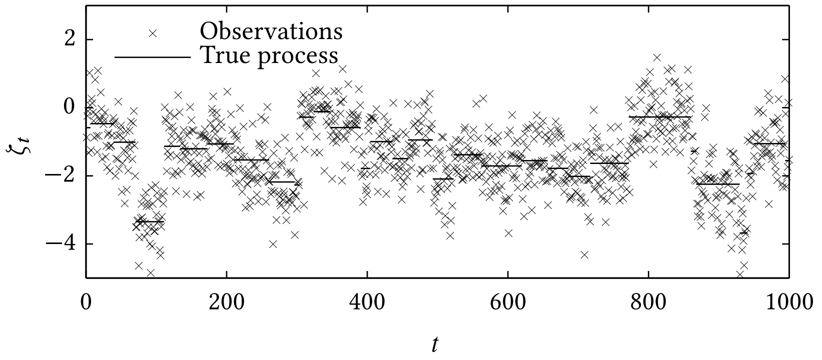

Figure 4.1 PDP and observations simulated from the elementary change-point model.

for some $\rho \in \mathbb{R}$. The deterministic function is taken to be piecewise constant and is given by $F^{\theta}(t, \tau, \phi) := \phi$. Observations are recorded at regular intervals of length Δ and are formed by adding Gaussian noise with mean 0 and variance σ_y^2 to the PDP.

Figure 4.1 shows data simulated from the model over a horizon of $T = 1{,}000$ time units with $\Delta = 1$, $\rho = 0.9$, $\sigma_\phi^2 = 1$ and $\sigma_y^2 = 0.5$ using gamma-distributed interjump times with shape and scale parameters $\theta_\tau := (\alpha, \beta) = (4, 10)$.

As the PDP ζ is only discretely and noisily observed, (particle) filtering methods are generally needed to conduct inference about the jump times and jump sizes. In addition, $\theta := (\rho, \sigma_\phi^2, \sigma_y^2, \theta_\tau) \in \mathbb{R} \times (0, \infty)^4$ are static parameters which often also have to be estimated.

4.2.3 Shot-Noise Cox-Process Example

A second example of a PDP is a shot-noise Cox-process model described in Whiteley et al. (2011). The model assumes that observations are taken on a Cox process (also known as a doubly-stochastic Poisson process) with piecewise deterministic shot-noise intensity, $\zeta = (\zeta_t)_{t \geq 0}$.

Application in Finance. Such models have applications in finance, as described in Centanni and Minozzo (2006a, 2006b): in the modelling of ultra-high-frequency financial data, observations are two-dimensional, comprising the time and size of the price movements of a stock. That is, the stock price process (which can be fully observed) is piecewise constant since the quoted price is only updated at a countable collection of random times. The times at which the stock price changes are realisations of a Cox process with unobserved shot-noise intensity ζ.

The latent intensity process ζ (i.e. the PDP) then has the following interpretation. The jth stopping time, τ_j, corresponds to the arrival of the jth news item at the market. This causes a positive jump in the intensity process, whose size, $\phi_j > 0$, depends on the 'importance' of the news item. Between τ_j and τ_{j+1}, the intensity gradually decays as the news item is absorbed by the market. The intensity process thus governs the amount of activity in the market: each jump leads to an increase in the trading activity as measured by the number of subsequent change points in the (observed) price process.

Application in Insurance. Such models are also used to price catastrophe insurance derivatives as described in Dassios and Jang (2003). In this context, the observations are only one-dimensional and represent the times at which claims are being recorded. In other words, the claim arrival process is a Cox process with intensity process ζ. The jth jump in the intensity process (at time τ_j) thus corresponds to a catastrophic event. The associated jump size, ϕ_j, characterises the event's severity.

More precisely, we have $q_0^\theta(\phi_0) = \lambda_\phi \exp(-\lambda_\phi \phi_0)\, \mathbb{1}_{[0,\infty)}(\phi_0)$, and

$$\begin{aligned} f^\theta(\tau_j | \tau_{j-1}) &:= \lambda_\tau \exp(-\lambda_\tau(\tau_j - \tau_{j-1}))\, \mathbb{1}_{(\tau_{j-1},\infty)}(\tau_j), \\ q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1}) &:= \lambda_\phi \exp(-\lambda_\phi(\phi_j - \zeta_{\tau_j}^-))\, \mathbb{1}_{(\zeta_{\tau_j}^-,\infty)}(\phi_j), \end{aligned}$$

where $\zeta_{\tau_j}^- := \phi_{j-1} \exp(-\kappa(\tau_j - \tau_{j-1}))$ is the intensity immediately before the jth jump. At any time t, the intensity is a deterministic function of t as well as of the most recent jump time and jump size as follows,

$$\zeta_t = F^\theta(t, \tau_{\nu_t}, \phi_{\nu_t}) := \phi_{\nu_t} \exp(-\kappa(t - \tau_{\nu_t})).$$

In addition, the likelihood of the observations recorded in the time interval $(t_{n-1}, t_n]$ (i.e. the times at which claims are recorded in this interval), denoted $y_{(t_{n-1},t_n]}$, is given by

$$g^\theta(y_{(t_{n-1},t_n]} | \zeta_{(t_{n-1},t_n]}) \propto \exp\biggl(-\int_{t_{n-1}}^{t_n} \zeta_s \,\mathrm{d}s\biggr) \prod_{i : y_i \in (t_{n-1},t_n]} \zeta_{y_i}.$$

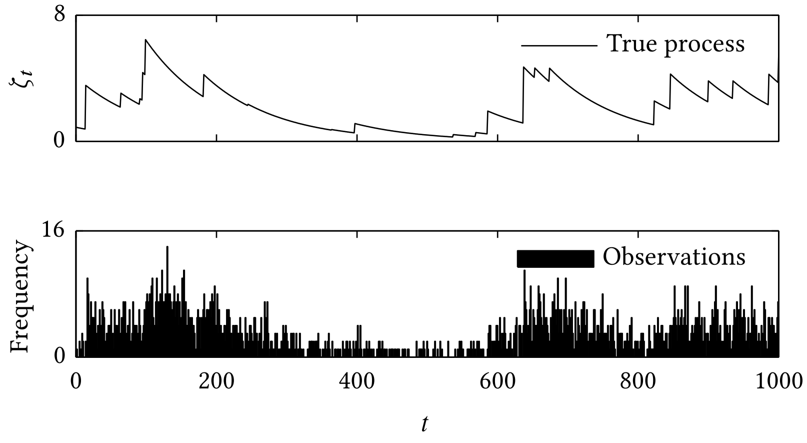

Figure 4.2 Data simulated from the shot-noise Cox-process model. *Top:* intensity process. *Bottom:* histogram of the observations using a bin width of 2.5.

Figure 4.2 shows an example trajectory and observations simulated from the model with $\kappa = 1/100$, $\lambda_\tau = 1/40$, $\lambda_\phi = 2/3$.

As the process ζ is not directly observed, (particle) filtering methods are needed to conduct inference about the jump times and jump sizes in the intensity process. The static parameters $\theta := (\kappa, \lambda_\tau, \lambda_\phi) \in (0, \infty)^3$ must often also be estimated.

4.2.4 Object-Tracking Example

This subsection briefly mentions, as a third example of a PDP, a model for tracking fighter aircraft from Whiteley et al. (2011).

In this model, the PDP represents the evolution of position, speed, and velocity of the aircraft. The assumption is that the pilot accelerates or decelerates at a countable collection of random times which correspond to the jumps in the PDP. Between jumps, the aircraft's location and speed are deterministic functions, given by the standard equations of motion, of the location, speed, and acceleration at the most recent jump time as well as of the time elapsed since the most recent jump. However, only countably many noisy observations on the aircraft's location are available.

While filtering for this model was shown to be feasible in Whiteley et al. (2011), it exhibits a characteristic that makes static-parameter estimation difficult: the transitions $q^\theta(\phi_j|\phi_{j-1}, \tau_j, \tau_{j-1})$ have degenerate components because the location and speed components of the PDP evolve continuously, i.e. they only have trivial jumps.

We note here that the variance-reduction techniques for conditional sequential Monte Carlo kernels described in Subsection 3.4.2 cannot be applied to such degenerate problems which makes PG sampler-based inference impractical. However, RJMCMC-based algorithms, such as those from Centanni and Minozzo (2006a, 2006b), Del Moral et al. (2007), Martin et al. (2013) will not be practical for such models either because the conditional posterior distribution of almost any individual jump is degenerate. The problem remains generally unsolved.

4.3 Existing SMC Algorithms

4.3.1 Variable-Rate Particle Filter

In this section, we describe filtering for PDPs via *sequential Monte Carlo* (SMC) methods. All three algorithms presented in this section may be viewed as special cases of the generic SIR algorithm from Section 2.3.1. Hence, we always use the same symbols $X_{1:n}$, π_n^θ, γ_n^θ, P_n^θ and G_n^θ to refer to the 'states', normalised and unnormalised extended target measures, proposal kernels and unnormalised incremental weights even though the particular form of these quantities may change between the next three subsections. The actual (marginal) target measure, $\tilde{\gamma}_n^\theta \in \mathcal{M}(\widetilde{\mathsf{E}}_n)$, and its normalising constant, $\mathfrak{z}_n^\theta$, are defined as in Subsection 4.2.1.

The first particle filter for PDPs, termed *variable-rate particle filter* (VRPF), was proposed by Godsill and Vermaak (2004). The VRPF can be viewed as an application of the generic SMC algorithm to a slightly reparametrised model described in the following.

Let $0 = t_0 < t_1, < t_2 < \ldots$ be a sequence of strictly increasing (non-random) times, where t_p, for $p > 1$, represents the time of the pth SMC step. Moreover, let $(\tau_{p,k}, \phi_{p,k})$ denote the kth jump time in the interval $(t_{p-1}, t_p]$ and its associated jump size. Let $k_p \geq 0$ be the total number of jumps in this interval and define the 'states' $X_{1:n}$, where

- $X_n := (k_n, \tau_{n,1:k_n}, \phi_{n,1:k_n})$, for $n > 1$, takes values in (a subset of)

$$\mathsf{E}_n := \bigcup_{k=0}^{\infty} (\{k\} \times \mathsf{T}_{(t_{n-1},t_n],k} \times \Phi^k),$$

- $X_1 := (k_1, \tau_{1,1:k_1}, \phi_{1,0:k_1})$ takes values in (a subset of)

$$\mathsf{E}_1 := \bigcup_{k=0}^{\infty} (\{k\} \times \mathsf{T}_{(0,t_1],k} \times \Phi^{k+1}).$$

Let $\nu(n) := \sup\{\, m \in \mathbb{N}_{n-1} \mid k_m > 0 \,\}$ be the index of the last interval of the form $(t_{p-1}, t_p]$ before $(t_{n-1}, t_n]$ in which the PDP has had a jump, with the convention that if $\nu(n) = -\infty$, then we set $\tau_{\nu(n),k_{\nu(n)}} = \tau_0 = 0$ and $\phi_{\nu(n),k_{\nu(n)}} = \phi_0$. The target distribution is then given by $\pi_n^\theta := \gamma_n^\theta / \mathcal{Z}_n^\theta$ on $\mathsf{E}_{1:n}^{\times} := \bigtimes_{p=1}^{n} \mathsf{E}_p$, where

$$\begin{aligned}
&\gamma_n^\theta(\mathrm{d}x_{1:n}) \\
&\quad := S^\theta(t_n, \tau_{\nu(n),k_{\nu(n)}}) q_0^\theta(\mathrm{d}\phi_0) g^\theta(y_{(0,t_n]} | \zeta_{(0,t_n]}) \\
&\qquad \times \prod_{p \in \widetilde{D}_n} \mathbb{1}_{\mathsf{T}_{(t_{p-1},t_p],k_p}}(\tau_{p,1:k_p}) \\
&\qquad\quad \times q^\theta(\mathrm{d}\phi_{p,1} | \phi_{\nu(p),k_{\nu(p)}}, \tau_{p,1}, \tau_{\nu(p),k_{\nu(p)}}) \\
&\qquad\quad \times f^\theta(\mathrm{d}\tau_{p,1} | \tau_{\nu(p),k_{\nu(p)}}) \\
&\qquad\quad \times \prod_{j=2}^{k_p} q^\theta(\mathrm{d}\phi_{p,j} | \phi_{p,j-1}, \tau_{p,j}, \tau_{p,j-1}) f^\theta(\mathrm{d}\tau_{p,j} | \tau_{p,j-1}).
\end{aligned}$$

Here, $\widetilde{D}_n := \{\, p \in \mathbb{N}_n \mid k_p > 0 \,\}$ is the collection of indices of intervals of the form $(t_{p-1}, t_p]$ that contain at least one jump. For $t \in (t_{n-1}, t_n]$, the PDP is then defined by

$$\zeta_t := \begin{cases} F^\theta(t, \tau_{n,j}, \phi_{n,j}), & \text{if } k_n > 0 \text{ and } t \in [\tau_{n,j}, \tau_{n,j+1}), \\ F^\theta(t, \tau_{\nu(n),k_{\nu(n)}}, \phi_{\nu(n),k_{\nu(n)}}), & \text{otherwise,} \end{cases}$$

with the convention that $\tau_{n,k_n+1} = t_n$. The extended distribution π_n^θ then admits $\tilde{\pi}_n^\theta$ as a marginal.

At Step n, the algorithm generates a particle X_n from the proposal kernel

$$\begin{aligned} P_n^\theta(\mathrm{d}x_n|x_{1:n-1}) := {} & P_{n,1}^\theta(\mathrm{d}k_n|x_{1:n-1}) \\ & \times P_{n,2}^\theta(\mathrm{d}\tau_{n,1:k_n} \times \mathrm{d}\phi_{n,1:k_n}|k_n, x_{1:n-1}). \end{aligned}$$

In the above equation, the kernels on the right hand side are selected in such a way that the usual absolute-continuity conditions are satisfied. At Step 1, the kernel $P_{1,2}^\theta$ also samples a value for ϕ_0.

The unnormalised incremental weight at Step n is then given by the following expressions. If $k_n = 0$, then with some abuse of the notation for Radon–Nikodým derivatives,

$$\begin{aligned} G_n^\theta(x_{1:n}) = {} & \frac{S^\theta(t_n, \tau_{\nu(n),k_{\nu(n)}})}{S^\theta(t_{n-1}, \tau_{\nu(n),k_{\nu(n)}})} \\ & \times \frac{g^\theta(y_{(t_{n-1},t_n]}|\phi_{\nu(n),k_{\nu(n)}}, \tau_{\nu(n),k_{\nu(n)}})}{P_{n,1}^\theta(k_n|x_{1:n-1})}. \end{aligned}$$

If $k_n \geq 1$, then again with some abuse of notation,

$$\begin{aligned} & G_n^\theta(x_{1:n}) \\ & \quad = \frac{S^\theta(t_n, \tau_{n,k_n})}{S^\theta(t_{n-1}, \tau_{\nu(n),k_{\nu(n)}})} \frac{g^\theta(y_{(t_{n-1},\tau_{n,1})}|\phi_{\nu(n),k_{\nu(n)}}, \tau_{\nu(n),k_{\nu(n)}})}{P_n^\theta(x_n|x_{1:n-1})} \\ & \quad \times g^\theta(y_{(\tau_{n,k_n},t_n]}|\phi_{n,k_n}, \tau_{n,k_n}) \\ & \quad \times q^\theta(\phi_{n,1}|\phi_{\nu(n),k_{\nu(n)}}, \tau_{n,1}, \tau_{\nu(n),k_{\nu(n)}}) f^\theta(\tau_{n,1}|\tau_{\nu(n),k_{\nu(n)}}) \\ & \quad \times \prod_{j=2}^{k_n} g^\theta(y_{(\tau_{n,j-1},\tau_{n,j})}|\phi_{n,j-1}, \tau_{n,j-1}) \\ & \qquad \times q^\theta(\phi_{n,j}|\phi_{n,j-1}, \tau_{n,j}, \tau_{n,j-1}) f^\theta(\tau_{n,j}|\tau_{n,j-1}). \end{aligned}$$

As shown in Whiteley et al. (2011), the VRPF can suffer from severe sample impoverishment. This is because at Step n, jumps are only proposed in the interval $(t_{n-1}, t_n]$ and only based on information available up to time t_n. If subsequent observations are highly informative about jumps in $(t_{n-1}, t_n]$, as they usually are in PDPs, then at later steps, this information can only be incorporated by reweighting particle paths. This can increase the variance of the particle weights which in turn aggravates the sample-impoverishment problem outlined in Subsection 2.4.1.

The SMC filter from Whiteley et al. (2011), outlined below, can reduce sample impoverishment because – even in its simplest form – it allows new jumps to be sampled anywhere after the most recent jump and also allows previously generated jumps to be adjusted.

4.3.2 SMC Filter for PDPs

The SMC filter for PDPs from Whiteley et al. (2011) is based on the SMC-sampler framework described in Subsection 2.3.2. That is, it is a 'standard' SMC algorithm that targets a sequence of artificially extended distributions $\pi_n^\theta := \gamma_n^\theta / \mathfrak{z}_n^\theta$ (as in Equation 2.10) by means of mixture proposal kernels. These extended distributions are defined on (product) spaces $\mathsf{E}_{1:n}^{\times} := \bigtimes_{p=1}^{n} \mathsf{E}_p$, with $\mathsf{E}_p := (\mathsf{M} \times \widetilde{\mathsf{E}}_p)$. Here, M is again the set of (proposal-kernel) mixture component indices.

We now add an additional subscript to the model parameters to account for the fact that for any particle, the jth jump time or jump size at the nth step of the algorithm may be different from the jth jump time or jump size at Step $n-1$. Thus, we hereafter write $X_n := (M_n, K_n, \tau_{n,1:k_n}, \phi_{n,0:k_n})$ for a particle at Step n. To ease the notational burden, we often write $Z_n := X_n \setminus M_n = (K_n, \tau_{n,1:k_n}, \phi_{n,0:k_n})$.

Proposal Kernels. In the most basic form presented in this work, we employ a mixture Kernel,

$$P_n^\theta(\mathrm{d}x_n | x_{n-1}) := \alpha_n^\theta(\mathrm{d}m_n | z_{n-1}) P_{n,m_n}^\theta(\mathrm{d}z_n | z_{n-1}),$$

with just two mixture components, i.e. $m_n \in \mathsf{M} = \{a, b\}$. At Step n, an *adjustment* move ($M_n = a$),

$$\begin{aligned} P_{n,a}^\theta(\mathrm{d}z_n | z_{n-1}) = {} & \delta_{k_{n-1}}(\mathrm{d}k_n) \delta_{\tau_{n-1,1:k_{n-1}-1}}(\mathrm{d}\tau_{n,1:k_n-1}) \\ & \times \delta_{\phi_{n-1,0:k_{n-1}-1}}(\mathrm{d}\phi_{n,0:k_n-1}) \\ & \times \rho_{n,a}^\theta(\mathrm{d}\tau_{n,k_n} | z_{n-1}) \eta_{n,a}^\theta(\mathrm{d}\phi_{n,k_n} | \tau_{n,k_n}, z_{n-1}), \end{aligned}$$

moves the most recent stopping time to a new location according to a distribution $\rho^{\theta}_{n,a}(\,\cdot\,|z_{n-1})$ with support $(\tau_{n-1,k_{n-1}-1}, t_n]$ and samples a new value for the corresponding jump size from a distribution $\eta^{\theta}_{n,a}(\,\cdot\,|\tau_{n,k_n}, z_{n-1})$ on Φ. A *birth move* $(M_n = b)$,

$$\begin{aligned} P^{\theta}_{n,b}(\mathrm{d}z_n|z_{n-1}) = {} & \delta_{k_{n-1}+1}(\mathrm{d}k_n)\delta_{\tau_{n-1,1:k_{n-1}}}(\mathrm{d}\tau_{n,1:k_n-1}) \\ & \times \delta_{\phi_{n-1,0:k_{n-1}}}(\mathrm{d}\phi_{n,0:k_n-1}) \\ & \times \rho^{\theta}_{n,b}(\mathrm{d}\tau_{n,k_n}|z_{n-1})\eta^{\theta}_{n,b}(\mathrm{d}\phi_{n,k_n}|\tau_{n,k_n}, z_{n-1}), \end{aligned}$$

adds a new stopping time by sampling it from a distribution $\rho^{\theta}_{n,b}(\,\cdot\,|z_{n-1})$ with support $(\tau_{n-1,k_{n-1}}, t_n]$. Additionally, a new jump-size parameter is sampled from a distribution $\eta^{\theta}_{n,b}(\,\cdot\,|\tau_{n,k_n}, z_{n-1})$ on Φ.

As in Whiteley et al. (2011), the forward mixture weights are set to $\alpha^{\theta}_n(a|z_{n-1}) := S^{\theta}(t_n, \tau_{n-1,k_{n-1}})$ as well as $\alpha^{\theta}_n(b|z_{n-1}) := 1 - \alpha^{\theta}_n(a|z_{n-1})$, i.e. the probability of a birth move grows as the time to the last jump increases. At Step 1, a birth move is enforced for each particle so that $\alpha^{\theta}_1(\mathrm{d}m_1) = \delta_b(\mathrm{d}m_1)$.

Backward Kernels. The backward Markov kernels

$$L^{\theta}_{n-1}(\mathrm{d}m_n \times \mathrm{d}z_{n-1}|z_n) := \beta^{\theta}_{n-1}(\mathrm{d}m_n|z_n)L^{\theta}_{n-1,m_n}(\mathrm{d}z_{n-1}|z_n),$$

have a similar 'mixture' structure. The component corresponding to adjustment moves is

$$\begin{aligned} L^{\theta}_{n-1,a}(\mathrm{d}z_{n-1}|z_n) := {} & \delta_{k_n}(\mathrm{d}k_{n-1})\delta_{\tau_{n,1:k_n-1}}(\mathrm{d}\tau_{n-1,1:k_{n-1}-1}) \\ & \times \delta_{\phi_{n,0:k_n-1}}(\mathrm{d}\phi_{n-1,0:k_{n-1}-1}) \\ & \times Q^{\theta}_{n-1,a}(\mathrm{d}\tau_{n-1,k_{n-1}} \times \mathrm{d}\phi_{n-1,k_{n-1}}|z_n), \end{aligned}$$

where $Q^{\theta}_{n-1,a}(\,\cdot\,|z_n)$ is a distribution whose support is a subset of the space $(\tau_{n-1,k_{n-1}-1}, t_{n-1}] \times \Phi$. For a birth move, the corresponding backward kernel component is

$$\begin{aligned} L^{\theta}_{n-1,b}(\mathrm{d}z_{n-1}|z_n) = {} & \delta_{k_n-1}(\mathrm{d}k_{n-1})\delta_{\tau_{n,1:k_n-1}}(\mathrm{d}\tau_{n-1,1:k_{n-1}}) \\ & \times \delta_{\phi_{n,0:k_n-1}}(\mathrm{d}\phi_{n-1,0:k_{n-1}}). \end{aligned}$$

The adjustment- and birth-move kernels only affect the most recent jump time and jump size. This is a reasonable approach as SMC filters can only be expected to work for ergodic models and for these, this strategy should be adequate. Nonetheless, other moves – and even a larger number of moves – could easily be incorporated. For instance, the second, say, most recent jump time or jump size may also be modified.

Indeed, as noted by Whiteley et al. (2011), a kernel for multiple-birth-moves should be included because otherwise, the above choice of forward/backward kernels induces an approximation. However, the probability of such moves is typically so small that this leads to computationally the same algorithm. To keep the presentation simple, we refrain from including such moves here (as was done in Del Moral et al., 2006b, 2007), although there is no technical difficulty with so doing.

4.3.3 Theoretical Analysis

Actually Targeted Distribution. In the following, we characterise the approximation induced by the above choice of forward and backward kernels, i.e. with only single-birth or single-adjustment moves.

First, we define some notation.

- $b_j := \inf\{\, q \in \mathbb{N} \mid \sum_{l=1}^{q} \mathbb{1}_{\{b\}}(m_l) = j \,\}$ denotes the index of the SMC step at which the jth birth move occurs.
- $s(\tau) := \inf\{\, q \in \mathbb{N} \mid t_q \geq \tau \,\}$ denotes the index of the first SMC step at which a jump with jump time τ could have been proposed.
- $\tilde{s}(\tau_{1:j}) := \sup\{\, s(\tau_{j-l+1}) + l - 1 \mid l \in \mathbb{N}_j \,\}$ denotes the minimum number of SMC steps needed to propose jumps with jump times $\tau_{1:j}$.

The product of the particle proposal kernels up to Step n,

$$P_1^\theta(\mathrm{d}x_1) \prod_{p=2}^{n} P_p^\theta(\mathrm{d}x_p|x_{1:p-1}), \tag{4.2}$$

then has support

$$\mathsf{E}_n^{\mathrm{r}} := \left\{ x_{1:n} \in \mathsf{E}_{1:n}^{\times} \;\middle|\; \begin{array}{l} b_1 = 1 \text{ and} \\ \forall j \in \mathbb{Z}_{2,k_n} : \tilde{s}(\tau_{n,1:k_j-1}) < b_j \leq n \end{array} \right\}.$$

In particular, the marginal distribution of X_n under the distribution in Equation 4.2 then has support

$$\widetilde{\mathsf{E}}_n^{\mathrm{r}} := \left\{ (k_n, \tau_{1:k_n}, \phi_{0:k_n}) \in \widetilde{\mathsf{E}}_n \;\middle|\; \forall j \in \mathbb{N}_{k_n} : s(\tau_j) \leq n - k_n + j \right\}.$$

Recall that we write $z_n = x_n \setminus m_n = (k_n, \tau_{1:k_n}, \phi_{0:k_n})$. To ensure that the importance weights exist, the algorithm can therefore only target, as a marginal, the distribution

$$\tilde{\pi}_n^{\mathrm{r},\theta}(\mathrm{d}z_n) \propto \tilde{\gamma}_n^{\mathrm{r},\theta}(\mathrm{d}z_n) = \tilde{\gamma}_n^{\theta}(\mathrm{d}z_n)\, \mathbb{1}_{\tilde{\mathsf{E}}_n^{\mathrm{r}}}(z_n).$$

If the time between successive SMC steps, $t_n - t_{n-1}$, is short compared to the average time between jumps, the difference between the distribution in Equation 4.1 and the marginally targeted distribution $\tilde{\pi}_n^{\mathrm{r},\theta}$ should be negligible. We will consider the influence of this approximation in Section 4.5.

The extended distribution actually targeted by the algorithm is

$$\begin{aligned}\pi_n^{\mathrm{r},\theta}(\mathrm{d}x_{1:n}) \propto \tilde{\gamma}_n^{\mathrm{r},\theta}(\mathrm{d}z_n)&\beta_0^{\theta}(\mathrm{d}m_1|z_1)\\ &\times \prod_{p=1}^{n-1} \beta_p^{\theta}(\mathrm{d}m_{p+1}|z_{p+1}) L_{p,m_{p+1}}^{\theta}(\mathrm{d}z_p|z_{p+1}),\end{aligned}$$

where we assume, for the moment, that the backward mixture weights $\beta_p^{\theta}(\,\cdot\,|z_{p+1})$ can be chosen such that this extended distribution does not have probability mass outside of $\mathsf{E}_n^{\mathrm{r}}$ to ensure that the importance weights exist. A detailed discussion of the choice of backward mixture weights is given below.

We conclude this subsection by describing some implementation issues regarding the above-mentioned SMC algorithm. To our knowledge, they have not been pointed out in the literature. The point we wish to stress here is that backward and proposal kernels need to be chosen carefully and in such a way that they are consistent with each other in order to avoid introducing biases resulting from a loss of absolute continuity. Such biases may be small in the case of filtering (i.e. if the static parameters are known). However, if the static parameters are to be estimated alongside the jump times and jump sizes, even small biases in the filter can induce large biases in the estimates of the static parameters.

Choice of Jump-Size Proposal Kernels. It was advocated in Whiteley et al. (2011) to sample the jump sizes from their full conditional posterior distribution. However, given the structure of the algorithm, this posterior distribution will often be based on observations ‘too far’ into the future.

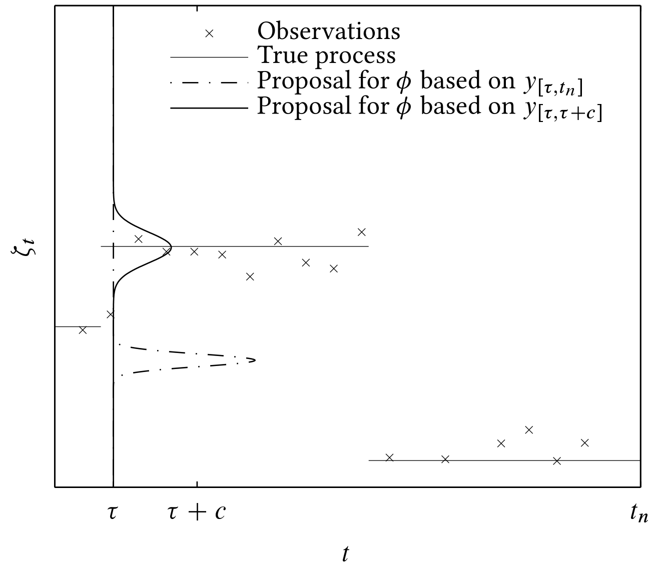

Figure 4.3 Proposal distributions for the jump size $\phi = \phi_{n,k_n}$ associated with the jump time $\tau = \tau_{n,k_n}$ sampled at the nth SMC step in the elementary change-point model (as part of one of the particles). *Dashed line:* full conditional distribution of ϕ given all the data up to time t_n. *Solid line:* full conditional distribution of ϕ given the data up to time $\tau + c$.

More formally, assume that the jump time $\tau = \tau_{n,k_n}$ proposed at Step n is much smaller than t_n. In this case, the full conditional posterior distribution of the jump size $\phi = \phi_{n,k_n}$ conditions on there not being another jump in the interval $(\tau, t_n]$. Due to conditioning on a potentially large number of observations, this full conditional posterior distribution can then be highly concentrated.

However, not having another jump in the interval $(\tau, t_n]$ can have little probability mass under the joint posterior distribution. If, at Step $(n + 1)$, a further jump is added in this interval, then the jump size ϕ might be located in a region that has little probability mass under the full conditional posterior distribution (which then also conditions on the additional jump). This can lead to a high variance in the particle weights. The problem is illustrated in Figure 4.3.

To circumvent this problem while still retaining the benefit of incorporating observations into the proposal kernels, we recommend to only take observations from some interval $[\tau, \tau + c])$ into account when sampling ϕ. For instance, we may take $c = (\tau + \lambda\tilde{\mu}) \wedge t_n$, $\lambda \in (0, 1)$ and $\tilde{\mu}$ may be the mean interjump time or a quantile of the interjump-time distribution.

Choice of Backward Mixture Weights. As previously mentioned, the backward mixture weights must be chosen such that the extended target distribution does not have probability mass outside of $\mathsf{E}_n^{\mathrm{r}}$. The most obvious problem with a poor choice of backward mixture weights is that the extended target distribution does not actually admit the right marginal (in addition to having ill-defined importance weights).

There is a one-to-one correspondence between $\sum_{p=1}^{n} \mathbb{1}_{\{b\}}(m_p)$, the number of birth moves, and k_n, the number of jumps in the proposal distribution and hence in the support of the truncated target distribution $\tilde{\pi}_n^{\mathrm{r},\theta}$. Therefore we cannot specify their distributions independently. The target already specifies a distribution over k_n; if the backward mixture weights $\beta_p^\theta(\,\cdot\,|z_{p+1})$ do not depend on k_{p+1}, then they implicitly specify a second distribution over k_n and the marginal distribution of this quantity under the target distribution will not be what is intended.

For instance, consider setting the backward mixture kernel weights to a uniform distribution over $\mathsf{M} = \{a, b\}$ – a popular choice. Write $\mathsf{A}_n := \{\, p \in \mathbb{N}_{n-1} \mid m_{p+1} = a \,\}$, $\mathsf{B}_n := \mathbb{N}_{n-1} \setminus \mathsf{A}_n$, and $\mathsf{C}_n := \{b\} \times \mathsf{M}^{n-1}$. The algorithm targets as a marginal the distribution defined by

$$
\begin{aligned}
&\int_{\mathsf{E}^{\times}_{1:n}} \pi_n^{\mathrm{r},\theta}(\mathrm{d}x_{1:n})\, \mathbb{1}_D(z_n) \\
&\qquad \propto \int_{\widetilde{\mathsf{E}}^{\times}_{1:n}} \tilde{\pi}_n^{\mathrm{r},\theta}(\mathrm{d}z_n)\, \mathbb{1}_D(z_n) \\
&\qquad\quad \times \sum_{m_{1:n} \in \mathsf{C}_n} \Bigg(\prod_{p \in A_n} L^{\theta}_{p,a}(\mathrm{d}z_p|z_{p+1}) \Bigg) \prod_{p \in B_n} L^{\theta}_{p,b}(\mathrm{d}z_p|z_{p+1}) \\
&\qquad = \int_D \tilde{\pi}_n^{\mathrm{r},\theta}(\mathrm{d}z_n)\, \#\Bigg\{ m_{1:n} \in \mathsf{C}_n \;\Bigg|\; \textstyle\sum_{p=1}^{n} \mathbb{1}_{\{b\}}(m_p) = k_n \Bigg\} \\
&\qquad = \int_D \tilde{\pi}_n^{\mathrm{r},\theta}(\mathrm{d}z_n) \binom{n-1}{k_n - 1},
\end{aligned}
$$

for any measurable set $D \subseteq \widetilde{\mathsf{E}}_n$.

For regular proposal kernels, a possible choice of backward kernels restricting the support of the extended target distribution to $\mathsf{E}^{\mathrm{r}}_n$ may be induced by setting the backward mixture weights to

$$\beta^{\theta}_{p-1}(b|z_p) = \begin{cases} 0, & \text{if } k_p = 1 \text{ and } p > 1, \\ 1, & \text{if } p \in \{1, k_p, \tilde{s}(\tau_{p,1:k_p-1}) + 1\}, \\ q_p(z_p), & \text{otherwise,} \end{cases} \tag{4.3}$$

for some probability $q_p(z_p) \in (0, 1)$ which may depend on z_p.

Local Adjustment Moves. Ideally, adjustment moves should direct the jumps towards regions of higher posterior probability. If such moves cannot be devised, it is preferable to use *local* adjustment moves, e.g. small-scale Gaussian kernels centred around the current location of the jump. This reduces the risk of moving jumps away from regions of high posterior probability, which would add to sample impoverishment. However, such local adjustment moves are unlikely to move a jump currently contained in $(t_{p-1}, t_p]$ out of such an interval. Therefore, even using Equation 4.3 could result in importance weights with infinite variance.

A simple remedy is to employ *restricted* adjustment moves, i.e. local moves that are limited to the particular interval $(t_{p-1}, t_p]$ currently containing the jump. More formally, recall that $s(\tau) = \inf\{\, q \in \mathbb{N} \mid t_q \geq \tau \,\}$ is the first SMC step at which a jump with jump time τ could have been proposed. For restricted adjustment moves, $\rho^{\theta}_{n,a}(\,\cdot\,|z_{n-1})$ then has support $((\tau_{n-1,k_{n-1}-1} \vee t_{s_{n-1}-1}), t_{s_{n-1}}]$, where $s_{n-1} := s(\tau_{n-1,k_{n-1}})$, rather than having support $(\tau_{n-1,k_{n-1}-1}, t_n]$.

Also recalling that $\tilde{s}(\tau_{1:j}) = \sup\{\, s(\tau_{j-l+1}) + l - 1 \mid l \in \mathbb{N}_j \,\}$ represents the minimum number of SMC steps needed to propose jumps with jump times $\tau_{1:j}$, the support of the joint proposal distribution from Equation 4.2 is then given by

$$\mathsf{E}^{\mathrm{r}}_n = \left\{ x_{1:n} \in \mathsf{E}^{\times}_{1:n} \;\middle|\; \begin{array}{l} b_1 = \tilde{s}(\tau_{n,1}) = 1 \text{ and} \\ \forall j \in \mathbb{Z}_{2,k_n} : \tilde{s}(\tau_{n,1:k_j}) \leq b_j \leq n \end{array} \right\}.$$

To ensure that the target distribution does not have probability mass outside of $\mathsf{E}^{\mathrm{r}}_n$, the distribution of $\tau_{n-1,k_{n-1}}$ under $Q^{\theta}_{n-1,a}(\,\cdot\,|z_n)$ must have support $((\tau_{n,k_n-1} \vee t_{s_n-1}), t_{s_n}]$, where $s_n := s(\tau_{n,k_n})$. In addition, the backward mixture weights might take the form presented in Equation 4.3 but with $\tilde{s}(\tau_{p,1:k_p-1})$ replaced by $\tilde{s}(\tau_{p,1:k_p}) - 1$.

4.4 Reformulation of the SMC Filter

4.4.1 General Idea

One problem with the SMC filter for PDPs from the previous subsection – henceforth referred to as the 'original' SMC filter – is that it induces unnecessary degeneracy in the transitions at Step n because most jump times and jump sizes in X_{n-1} coincide with jump times and jump sizes in X_n. In other words, the algorithm works explicitly on the path space by embedding a 'standard' SMC filter within the SMC-sampler framework using (mostly) trivial degenerate backward transitions.

Unfortunately, degenerate backward transitions prevent the use of backward-simulation methods such as forward filtering–backward sampling and they also prevent the use of the essential variance-reduction techniques for PG samplers: backward sampling and ancestor sampling, which were described in Subsection 3.4.2. At the same time, the algorithm does not gain any benefit from the path-space representation.

Below, we present a novel representation of the algorithm whose alternative Step-n extended target distribution also admits $\tilde{\pi}_n^{\mathrm{r},\theta}$ as a marginal but whose 'states' do not have degenerate transitions (unless $q^\theta(\phi_j|\phi_{j-1}, \tau_j, \tau_{j-1})$ is degenerate). In addition, this algorithm makes it easier to ensure the existence of the importance weights as it circumvents the problem of choosing sensible backward mixture weights. Our representation may be viewed as a way of extracting the 'standard' SMC filter embedded in the SMC sampler from Whiteley et al. (2011). The structure of its extended target distribution is reminiscent of the *product-space* formulation from Carlin and Chib (1995) (see also Godsill, 2001).

4.4.2 Extended Target Distribution

The algorithm presented in this subsection targets an extended distribution (defined further below) that contains all the 'states' $X_{1:n}$, where

- $X_n := (M_n, \tau_n, \phi_n)$, for $n > 1$, takes values in (a subset of)

$$\mathsf{E}_n := \mathsf{M} \times (0, t_n] \times \Phi,$$

- $X_1 := (M_1, \tau_1, \phi_1, \phi_0)$ takes values in (a subset of)

$$\mathsf{E}_1 := \mathsf{M} \times (0, t_n] \times \Phi^2.$$

In this subsection, τ_n and ϕ_n are the jump time and associated jump size sampled at the nth step of the SMC algorithm as part of a birth move or as part of an adjustment move. As before, M_n indicates an adjustment move ($M_n = a$) or birth move ($M_n = b$) at Step n. Again, these mixture component indices are added to the state space as auxiliary variables.

The main idea is to use the mixture component indices $M_{1:n}$ to keep track of which jumps affect the marginal target distribution $\tilde{\pi}_n^{\mathrm{r},\theta}$. These are the jumps sampled in SMC Steps $p \in H_n$, where

$$H_n := \{\, j \in \mathbb{N}_{n-1} \mid m_{j+1} = b \,\} \cup \{n\}.$$

That is, H_n contains the indices of all jumps which have been sampled (up to Step n) immediately before a birth move. We also define the set of indices of the remaining jumps, $V_n := \mathbb{N}_n \setminus H_n$. For easier reference, we collect all elements of the set H_n in the vector

$$h_n = (h_n(1) \ldots h_n(\#H_n)),$$

in increasing order and all the elements of the set V_n in the vector

$$v_n = (v_n(1) \ldots v_n(\#V_n)),$$

again in increasing order.

Extended Target Distribution. In the following, we present the extended target distribution of the algorithm. To show that it admits the right marginal, some reparametrisation is required: write $k_n := \#H_n$ for the total number of birth moves in the first n steps and let $i_{1:k_n}$ denote the SMC steps at which these birth moves occur, i.e. $i_1 := 1$ and $i_j := h_n(j-1) + 1$ for $j \in \mathbb{Z}_{2,k_n}$. Re-label $(\tau_{h_n}, \phi_{h_n}) =: (\tau'_{1:k_n}, \phi'_{1:k_n})$ and $(\tau_{v_n}, \phi_{v_n}) =: (\tau^\star_{1:n-k_n}, \phi^\star_{1:n-k_n})$. This permits the one-to-one transformation

$$\begin{aligned}&\big[m_{1:n}, (\tau_{h_n}, \phi_{h_n}), (\tau_{v_n}, \phi_{v_n})\big]\\ &\qquad \longleftrightarrow \big[(n, k_n, i_{1:k_n}), (\tau'_{1:k_n}, \phi'_{1:k_n}), (\tau^\star_{1:n-k_n}, \phi^\star_{1:n-k_n})\big],\end{aligned} \tag{4.4}$$

where we have implicitly used that H_n, V_n, h_n and v_n can be equivalently defined in terms of $m_{1:n}$ or in terms of $(n, k_n, i_{1:k_n})$.

Let $\mu_n^\theta(\mathrm{d}m_{1:n}|k, \tau_{h_n}, \phi_{h_n}, \phi_0)$ be some distribution on the indices $M_{1:n}$ conditional on $\sum_{p=1}^n \mathbb{1}_{\{b\}}(M_p) = k$.

4.1 Remark. *With some abuse of notation, Equation 4.4 allows us to use the same symbol μ_n^θ to define a distribution on the indices $I_{1:K_n}$, i.e. $\mu_n^\theta(\mathrm{d}i_{1:k_n} | k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)$. The choice of μ_n^θ is discussed below.*

The alternative extended target distribution for the SMC filter introduced in this subsection is defined as $\pi_n^\theta = \gamma_n^\theta / \mathfrak{z}_n^\theta$, where

$$\begin{aligned}
&\gamma_n^\theta(x_{1:n}) \\
&\quad := \tilde{\gamma}_n^{\mathrm{r},\theta}(\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)\mu_n^\theta(m_{1:n} | \#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0) \\
&\qquad \times \prod_{j \in V_n} Q_{j,a}^\theta(\tau_j, \phi_j | \#H_{j+1}, \tau_{h_{j+1}}, \phi_{h_{j+1}}, \phi_0) \\
&\quad = \tilde{\gamma}_n^{\mathrm{r},\theta}(k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)\mu_n^\theta(i_{1:k_n} | k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0) \\
&\qquad \times \left[\prod_{j \in D_n} Q_{v_n(j),a}^\theta\big(\tau_j^\star, \phi_j^\star \big| \bar{k}_n(j), \bar{\tau}_n(j), \bar{\phi}_n(j), \phi_0\big) \right] \\
&\qquad \times \prod_{j \in \mathbb{N}_{n-k_n} \setminus D_n} Q_{v_n(j),a}^\theta\big(\tau_j^\star, \phi_j^\star \big| \bar{k}_n(j), \tau'_{1:\bar{k}_n(j)}, \phi'_{1:\bar{k}_n(j)}, \phi_0\big).
\end{aligned} \tag{4.5}$$

The second line shows that this distribution can be written explicitly as a distribution over the variables on the right hand side in Equation 4.4. Here, we have defined the following symbols.

- The quantity $\bar{k}_n(j) := \sup\{ p \in \mathbb{N}_n \mid i_p \leq v_n(j) \}$ is the number of birth moves up to (and including) the $v_n(j)$th step of the SMC algorithm.
- Recall that by the definition of V_n and v_n, the particle trajectory $X_{1:n}$ has an adjustment move at Step $v_n(j)+1$, for any $j \in \mathbb{N}_{\#V_n}$. As a result, under the extended target distribution at Step n, the jump sampled at Step $v_n(j)$ is not distributed according to a marginal under the 'actual' target distribution, $\tilde{\pi}_n^{\mathrm{r},\theta}$. In the extended distribution above, we then need to distinguish two cases. First, the set

  $$D_n := \left\{ j \in \mathbb{N}_{n-k_n} \;\middle|\; \begin{array}{l} v_n(j) < n-1 \text{ and} \\ \forall l \in \mathbb{N}_{k_n} : v_n(j) + 2 \neq i_l \end{array} \right\},$$

  comprises those components $v_n(j)$ of v_n which are such that there is also an adjustment move at Step $v_n(j)+2$. Hence, under the extended target distribution at Step n, the jump sampled at Step $v_n(j)+1$ is also not distributed according to a marginal under $\tilde{\pi}_n^{\mathrm{r},\theta}$. Conversely, the set $\mathbb{N}_{n-k_n} \setminus D_n$ comprises those components $v_n(j)$ of v_n which are such that, under the extended target distribution at Step n, the jump sampled at Step $v_n(j)+1$ *is* distributed according to a marginal under $\tilde{\pi}_n^{\mathrm{r},\theta}$ (because of a birth move at Step $v_n(j)+2$).

- The symbols

$$\begin{aligned}\bar{\tau}_n(j) &:= \bigl(\tau'_{1:\bar{k}_n(j)-1}, \tau^\star_{j+1}\bigr),\\ \bar{\phi}_n(j) &:= \bigl(\phi'_{1:\bar{k}_n(j)-1}, \phi^\star_{j+1}\bigr),\end{aligned}$$

represent the vectors of jump times and jump sizes which, under the extended target distribution at Step $v_n(j)+1$, are distributed according to a suitable marginal of $\tilde{\pi}^{\mathrm{r},\theta}_{v_n(j)+1}$, but whose last component is not distributed according to a suitable marginal of $\tilde{\pi}_n^{\mathrm{r},\theta}$ because of an adjustment move at Step $v_n(j)+2$.

In summary, we have $j \in D_n$ if and only if the jumps sampled at Steps $v_n(j)$ and $v_n(j)+1$ are distributed according to a suitable marginal of $\tilde{\pi}_n^{\mathrm{r},\theta}$. Similarly, $j \in \mathbb{N}_{n-k_n} \setminus D_n$ if and only if the jump sampled at Step $v_n(j)$ is not distributed according to a suitable marginal of $\tilde{\pi}_n^{\mathrm{r},\theta}$, but the jump sampled at Step $v_n(j)+1$ is and it is denoted

$$\bigl(\tau'_{\bar{k}_n(j)}, \phi'_{\bar{k}_n(j)}\bigr).$$

Finally, $Q^\theta_{p,a}$ is defined as in the previous section.

The second line in Equation 4.5 shows that the extended target distribution admits $\tilde{\pi}_n^{\mathrm{r},\theta}$ as a marginal. In addition, under this extended target distribution, the transitions from $X_{1:n-1}$ to X_n will be free of degenerate components as long as $q^\theta(\phi_j|\phi_{j-1}, \tau_j, \tau_{j-1})$ is non-degenerate.

4.4.3 Extended Proposal Distribution

We use the proposal kernels

$$P_n^\theta(\mathrm{d}x_n|x_{1:n-1}) = \alpha_n(\mathrm{d}m_n|x_{n-1}) P^\theta_{n,m_n}(\mathrm{d}[x_n \setminus m_n]|x_{1:n-1})$$

with birth and adjustment moves that are similar to the ones used in the original formulation of the SMC filter for PDPs, except that they do not share the degenerate components. A birth move,

$$P^{\theta}_{n,b}(\mathrm{d}\tau_n \times \mathrm{d}\phi_n | x_{1:n-1}) := \rho^{\theta}_{n,b}(\mathrm{d}\tau_n | \#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0) \\ \times \eta^{\theta}_{n,b}(\mathrm{d}\phi_n | \tau_n, \#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0),$$

adds a new stopping time in $(\tau_{n-1}, t_n]$ and samples a new jump size. Similarly, an adjustment move,

$$P^{\theta}_{n,a}(\mathrm{d}\tau_n \times \mathrm{d}\phi_n | x_{1:n-1}) := \rho^{\theta}_{n,a}(\mathrm{d}\tau_n | \#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0) \\ \times \eta^{\theta}_{n,a}(\mathrm{d}\phi_n | \tau_n, \#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0),$$

shifts the most recent stopping time to a new location in $(\tau_{h_n(\#H_n-1)}, t_n]$ and also samples a new value for the corresponding jump size. The kernels ρ^{θ}_{n,m_n} and η^{θ}_{n,m_n} are defined as in the previous section and we again define the forward 'mixture weights' by $\alpha^{\theta}_n(a|x_{n-1}) := S^{\theta}(t_n, \tau_{n-1})$.

4.4.4 Distribution Over Birth-Move Locations

The support of the target distribution in Equation 4.5 must be included in the support of the proposal distribution. To ensure this, we propose to use the following distribution over the locations of the birth moves,

$$\mu^{\theta}_n(\mathrm{d}i_{1:k_n} | k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0) \\ := \nu_{k_n}(\mathrm{d}i_{k_n} | n+1, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)\delta_1(\mathrm{d}i_1) \\ \times \prod_{j=2}^{k_n-1} \nu_j(\mathrm{d}i_j | i_{j+1}, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0).$$

Here, $\nu_j(i_j | l, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)$ is a distribution that has support

$$\{\tilde{s}(\tau'_{1:j-1}) + 1, \ldots, l-1\},$$

where we recall the definition $\tilde{s}(\tau_{1:j}) = \sup\{\, s(\tau_{j-l+1}) + l - 1 \mid l \in \mathbb{N}_j \,\}$ with $s(\tau) = \inf\{\, q \in \mathbb{N} \mid t_q \geq \tau \,\}$.

If only local adjustment moves are used (see Subsection 4.3.2) then the support of $\nu_j(i_j | l, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)$ must be limited to

$$\{\tilde{s}(\tau'_{1:j}), \ldots, l-1\}.$$

In the applications presented in Section 4.5 we employ such restricted adjustment moves. Consequently, we may take $\nu_j(i_j|l, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)$ to be a geometric distribution truncated to $\{\tilde{s}(\tau'_{1:j}), \dots, l-1\}$ or simply a uniform distribution on this set. Such a choice also ensures that the computational cost (per SMC step) of computing the importance weights remains constant.

4.4.5 Incremental and Backward-Sampling Weights

Incremental Weights. For easier reference, we will hereafter refer to the algorithm presented in this subsection as the *reformulated sequential Monte Carlo* (RSMC) filter. Again abusing the notation for Radon–Nikodým derivatives, the incremental weights of this algorithm, denoted $G_n^\theta(x_{1:n}) = \gamma_n^\theta(x_{1:n})/[\gamma_{n-1}^\theta(x_{1:n-1})P_n^\theta(x_n|x_{1:n-1})]$,are computed as follows. For a birth move, i.e. $m_n = b$,

$$
\begin{aligned}
G_n^\theta(x_{1:n}) = {} & \frac{S^\theta(t_n, \tau_n)}{S^\theta(t_{n-1}, \tau_{n-1})} \frac{f^\theta(\tau_n|\tau_{n-1})q^\theta(\phi_n|\phi_{n-1}, \tau_n, \tau_{n-1})}{P_{n,b}^\theta(\tau_n, \phi_n|x_{1:n-1})} \\
& \times \frac{\mu_n^\theta(m_{1:n}|\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)}{\mu_{n-1}^\theta(m_{1:n-1}|\#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0)\alpha_n^\theta(b|x_{n-1})} \\
& \times \frac{g^\theta(y_{[\tau_n, t_n]}|\tau_n, \phi_n)}{g^\theta(y_{[\tau_n, t_{n-1}]}|\tau_{n-1}, \phi_{n-1})}.
\end{aligned}
$$

For an adjustment move, i.e. $m_n = a$,

$$
\begin{aligned}
G_n^\theta(x_{1:n}) = {} & \frac{S^\theta(t_n, \tau_n)}{S^\theta(t_{n-1}, \tau_{n-1})} \frac{Q_{n-1,a}^\theta(\tau_{n-1}, \phi_{n-1}|\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)}{P_{n,a}^\theta(\tau_n, \phi_n|x_{1:n-1})} \\
& \times \frac{\mu_n^\theta(m_{1:n}|\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)}{\mu_{n-1}^\theta(m_{1:n-1}|\#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0)\alpha_n^\theta(a|x_{n-1})} \\
& \times \frac{f^\theta(\tau_n|\tau_{h_n(\#H_n-1)})}{f^\theta(\tau_{n-1}|\tau_{h_n(\#H_n-1)})} \\
& \times \frac{q^\theta(\phi_n|\phi_{h_n(\#H_n-1)}, \tau_n, \tau_{h_n(\#H_n-1)})}{q^\theta(\phi_{n-1}|\phi_{h_n(\#H_n-1)}, \tau_{n-1}, \tau_{h_n(\#H_n-1)})} \\
& \times \frac{g^\theta(y_{[\tau_{n-1}\wedge\tau_n, \tau_n)}|\tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)})}{g^\theta(y_{[\tau_{n-1}\wedge\tau_n, \tau_{n-1})}|\tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)})} \\
& \times \frac{g^\theta(y_{[\tau_n, t_n]}|\tau_n, \phi_n)}{g^\theta(y_{[\tau_{n-1}, t_{n-1}]}|\tau_{n-1}, \phi_{n-1})}.
\end{aligned}
$$

Here, we use the convention $g^\theta(y_{[s,t)}|\tau_j, \phi_j) := 1$ if $s \geq t$. To actually compute these weights, it is preferable to switch to the parametrisation from the right hand side in Equation 4.4.

Backward-/Ancestor-Sampling Weights. We conclude this subsection by deriving the probabilities $G^\theta_{n|P}(x_{1:P}) := \gamma^\theta_P(x_{1:P})/\gamma^\theta_n(x_{1:n})$ needed for the computation of the backward or ancestor sampling weights in Equation 3.17 for the VRPF and the RSMC filter.

For the VRPF, using the notation from Subsection 4.3.1, assuming that $n < P$ and let

$$\mu(n) := \inf\bigl\{\, m \in \{n+1, \dots, P\} \bigm| k_m > 0 \,\bigr\}.$$

If $\sum_{p=n+1}^{P} k_p = 0$ and $k_n > 0$, then, recalling that $t_P = T$,

$$G^\theta_{n|P}(x_{1:P}) \propto S^\theta(T, \tau_{n,k_n}) g^\theta(y_{(t_n,T]}|\phi_{n,k_n}, \tau_{n,k_n}) / S^\theta(t_n, \tau_{n,k_n}).$$

If $\sum_{p=n+1}^{P} k_p > 0$ and $k_n > 0$ then

$$\begin{aligned} G^\theta_{n|P}(x_{1:P}) \propto{} & g^\theta(y_{(t_n,\tau_{\mu(n),1})}|\phi_{n,k_n}, \tau_{n,k_n}) f^\theta(\tau_{\mu(n),1}|\tau_{n,k_n}) \\ & \times q^\theta(\phi_{\mu(n),1}|\phi_{n,k_n}, \tau_{\mu(n),1}, \tau_{n,k_n}) / S^\theta(t_n, \tau_{n,k_n}). \end{aligned}$$

In the case that $k_n = 0$, the quantity $(\tau_{n,k_n}, \phi_{n,k_n})$ in the above equations is replaced by $(\tau_{\nu(n),k_{\nu(n)}}, \phi_{\nu(n),k_{\nu(n)}})$.

For the RSMC filter, using the notation from this section and assuming $n < P$, let $n_b := \inf\{\, p \in \mathbb{Z}_{n+2,P} \mid m_p = b \,\}$ denote the iteration with the first birth move after step $n+1$, with the convention that $n_b := P+1$ if there is no further jump at steps $n+2, \dots, P$. If $m_{n+1} = b$,

$$\begin{aligned} G^\theta_{n|P}(x_{1:P}) \propto{} & \frac{\mu^\theta_P(m_{1:P}|\#H_P, \tau_{h_P}, \phi_{h_P}, \phi_0)}{\mu^\theta_n(m_{1:n}|\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)} \\ & \times \frac{f^\theta(\tau_{n_b-1}|\tau_n) q^\theta(\phi_{n_b-1}|\phi_n, \tau_{n_b-1}, \tau_n)}{S^\theta(t_n, \tau_n)} \\ & \times \frac{g^\theta(y_{(t_n,\tau_{n_b-1})}|\tau_n, \phi_n)}{g^\theta(y_{[\tau_{n_b-1},t_n]}|\tau_n, \phi_n)} \\ & \times \prod_{j=n+2}^{n_b-1} Q^\theta_{j-1,a}(\tau_{j-1}, \phi_{j-1}|\#H_j, \tau_{h_j}, \phi_{h_j}, \phi_0). \end{aligned}$$

If $m_{n+1} = a$,

$$\begin{aligned}G^{\theta}_{n|P}(x_{1:P}) \propto\ & \frac{\mu^{\theta}_{P}(m_{1:P}|\#H_P, \tau_{h_P}, \phi_{h_P}, \phi_0)}{\mu^{\theta}_{n}(m_{1:n}|\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)} \\ & \times \frac{f^{\theta}(\tau_{n_b-1}|\tau_{h_n(\#H_n-1)})}{S^{\theta}(t_n, \tau_n) f^{\theta}(\tau_n|\tau_{h_n(\#H_n-1)})} \\ & \times \frac{q^{\theta}(\phi_{n_b-1}|\phi_{h_n(\#H_n-1)}, \tau_{n_b-1}, \tau_{h_n(\#H_n-1)})}{q^{\theta}(\phi_n|\phi_{h_n(\#H_n-1)}, \tau_n, \tau_{h_n(\#H_n-1)})} \\ & \times \frac{g^{\theta}(y_{[\tau_n \wedge \tau_{n_b-1}, \tau_{n_b-1})}|\tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)})}{g^{\theta}(y_{[\tau_n \wedge \tau_{n_b-1}, \tau_n)}|\tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)})} \\ & \times \frac{1}{g^{\theta}(y_{[\tau_n, t_n]}|\tau_n, \phi_n)} \\ & \times \prod_{j=n+1}^{n_b-1} Q^{\theta}_{j-1,a}(\tau_{j-1}, \phi_{j-1}|\#H_j, \tau_{h_j}, \phi_{h_j}, \phi_0).\end{aligned}$$

Again, we use the convention that $g^{\theta}(y_I|\tau_j, \phi_j) := 1$, if $I = \emptyset$. To compute these weights, it is again preferable to switch to the parametrisation from the right hand side in Equation 4.4.

4.4.6 The Algorithm

In this subsection, we summarise the RSMC-based PG sampler and with the auxiliary-variable rejuvenation scheme outlined in Subsection 3.4.4 and comment on the efficiency of *backward sampling* (BS) and *ancestor sampling* (AS), in this context.

More precisely, the algorithm is a special case of Algorithm 3.35 in which we take, with some abuse of notation, $T = P$, as well as

- $X_{1:P} = (M_{1:P}, \tau_{1:P}, \phi_{1:P})$,
- $Y = (I_{1:k_P}, \tau^{\star}_{1:P-k_P}, \phi^{\star}_{1:P-k_n})$,
- $Z = (k_P, \tau'_{1:k_P}, \phi'_{1:k_P})$,
- $\pi(\theta, x_{1:n}) \propto p(\theta)\gamma^{\theta}_{P}(x_{1:n})$, where $p(\theta)$ is some prior density for the parameters Θ, and where γ^{θ}_{P} represents the extended target measure associated with the RSMC algorithm up to Step P as defined in Equation 4.5,

- $\hat{\pi}^{\mathrm{M}}(\theta|z) \propto p(\theta)\tilde{\pi}_P^\theta(k_P, \tau'_{1:k_P}, \phi'_{1:k_P})$ (from which we sample in Step 2 of Algorithm 3.35), where $\tilde{\pi}_P^\theta$ denotes the posterior distribution of the jump times, jump sizes as well as their number up to Time t_P, as defined in Equation 4.1,
- and finally,

$$\begin{aligned} &L((\theta, z)), y) \\ &\quad = \mu_n^\theta(i_{1:k_n}|k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0) \\ &\qquad \times \left[\prod_{j \in D_n} Q^\theta_{v_n(j),a}\big(\tau_j^\star, \phi_j^\star \big| \bar{k}_n(j), \bar{\tau}_n(j), \bar{\phi}_n(j), \phi_0\big)\right] \\ &\qquad \times \prod_{j \in \mathbb{N}_{n-k_n} \setminus D_n} Q^\theta_{v_n(j),a}\big(\tau_j^\star, \phi_j^\star \big| \bar{k}_n(j), \tau'_{1:\bar{k}_n(j)}, \phi'_{1:\bar{k}_n(j)}, \phi_0\big). \end{aligned}$$

Efficiency of Backward/Ancestor Sampling. We conclude this section by commenting on the efficiency of BS (and, similarly, AS) when using a *conditional sequential Monte Carlo* (CSMC) algorithm based around the RSMC algorithm within a PG sampler. Assume that we are trying to 'connect' a particle path segment $X_{1:n}$ with a particle path segment $X_{n+1:P}$ via BS or AS.

(1) If $m_{n+1} = a$, $G^\theta_{n|P}(x_{1:P})$ is necessarily equal to zero if $\tau_j \leq \tau_{h_n(\#H_n - 1)}$, for any $j \in \mathbb{Z}_{n+1, n_b - 1}$. This reflects the fact that under the (extended) proposal distribution, any subsequent adjustment moves cannot move the most recent jump to a location before the second most recent jump at Step n. Furthermore, if only local adjustment moves for the jump time and jump size are employed, the term $G^\theta_{n|P}(x_{1:P})$ will necessarily be very small unless the distances $|\tau_n - \tau_{n+1}|$ and $|\phi_n - \phi_{n+1}|$ are sufficiently small. This reflects the fact that in this case, adjustment moves only slightly perturb an existing jump.

(2) If $m_{n+1} = b$, $G^\theta_{n|P}(x_{1:P})$ is necessarily equal to zero if $\tau_j \leq \tau_n$, for any $j \in \mathbb{Z}_{n+1, n_b - 1}$, where n_b is defined as in the previous subsection. This reflects the fact that under the (extended) proposal distribution, the birth move at Step $(n + 1)$ cannot propose a jump in $(0, \tau_n]$ and subsequent adjustment moves are similiarly unable to move the newly-born jump to any location in this interval.

In some models, $G^{\theta}_{n|P}(x_{1:P})$ may also be zero (or very small) due to restrictions imposed on the jump sizes under the model. For instance, in the Cox-process model from Subsection 4.2.3, the PDP must have a non-negative jump at the jth jump time, i.e. we must have $\zeta^{-}_{\tau_j} \leq \phi_j = \zeta_{\tau_j}$, where $\zeta^{-}_{\tau_j}$ denotes value of the PDP immediately Time τ_j.

Whenever $G^{\theta}_{n|P}(x_{1:P})$ is zero (or sufficiently small), it is impossible (or difficult) to 'connect' the particle path $X_{1:n}$ with the particle path $X_{n+1:P}$ via BS or AS. This hampers mixing of the PG kernel. However, the auxiliary-variable rejuvenation step alleviates this problem. More specifically, in our simulations (presented in the next Section), the extra Gibbs step (Step 3) of Algorithm 3.35 appeared to be crucial to the performance of the RSMC-based PG sampler: without it, the algorithm could get stuck in local modes.

We conjecture that with only local adjustment moves and without Step 3 of Algorithm 3.35, the PG sampler can get stuck because AS (or similarly BS) is relatively ineffective: it rarely changes the distinguished path (i.e. the particles $U_{1:P}$ in the notation from the previous chapter) in Situation 1, above, and mixing thus relies on replacing the distinguished path in Situation 2. Here, however, if τ_{n+1} is much smaller than t_n, the most recent jump in all other particle paths is likely to be located in the interval $(\tau_{n+1}, t_n]$ so that they have BS/AS weights equal to zero.

Step 3 of Algorithm 3.35 can circumvent the latter problem because it can change the SMC step of the birth move which is associated with a particular jump.

This reasoning might also explain why we observed that the RSMC-based PG sampler could get stuck in the shot-noise Cox-process example when it was initialised in a region with P jumps: if the distinguished path has only birth moves then Step 3 of Algorithm 3.35 cannot change the SMC step of the birth move associated with any jump.

4.5 Simulation study

4.5.1 General Setup

In this section, we apply a PG sampler with AS and with the auxiliary-variable rejuvenation step from Subsection 3.4.4 – based on the RSMC filter from Subsection 4.4.2 – to the elementary change-point model from Subsection 4.2.2 and to the shot-noise Cox-process model from Subsection 4.2.3. For easier reference, this algorithm is hereafter called RSMC-based PG sampler. We compare its performance with that of a VRPF-based PG sampler also using AS and additionally with a RJMCMC algorithm.

Reformulated SMC Filter. For the RSMC filter, a birth move at Step n samples a new jump time τ_n uniformly in $(\tau_{h_n(\#H_n-1)}, t_n]$. The jump size, ϕ_n, is then sampled from its full conditional posterior distribution given the observations up to time $\tau_n + \tilde{\mu}/4 \wedge t_n$, with $\tilde{\mu}$ being the prior mean interjump time. We use restricted jump-time adjustment moves, i.e. we use Gaussian kernels with variance 10^{-4}, centred around τ_{n-1} and truncated to $((\tau_{h_n(\#H_n-1)} \vee t_{s_{n-1}-1}), t_{s_{n-1}}]$ where $s_{n-1} := s(\tau_{n-1})$. Gaussian kernels with this variance, centred around ϕ_{n-1}, are also used for the jump-size adjustments. In the Cox-process example, these Gaussian kernels are truncated to $(F^\theta(\tau_n, \tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)}), \infty)$. Likewise, the kernel $Q^\theta_{n-1,a}$ is a product of independent Gaussians, each with variance 10^{-4}.

- The first component is centred around τ_n and truncated to the interval $((\tau_{h_n(\#H_n-1)} \vee t_{s_n-1}), t_{s_n}]$, where $s_n := s(\tau_n)$.
- The second component is centred around ϕ_n. In the Cox-process example, its support is restricted to $(F^\theta(\tau_{n-1}, \tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)}), \infty)$.

Finally, the conditional distribution of $i_{1:k_n}$ is taken to be a truncated geometric distribution with parameter 0.3 and with support (for restricted adjustment moves) as given in Subsection 4.4.2. Throughout, we use (conditional) systematic resampling and resample only when the effective sample size falls below $0.8N$, where N is the (constant) number of particles at every step.

Variable-Rate Particle Filter. For the VRPF, we propose the number of jumps in $(t_{n-1}, t_n]$ from a Poisson distribution with mean $(t_n - t_{n-1})/\tilde{\mu}$. The jump times are subsequently sampled independently from a uniform distribution on $(t_{n-1}, t_n]$ and are then ordered. The corresponding jump sizes are proposed from their full conditional time-t_n posterior distribution. The step size in both SMC algorithms is set to $t_n - t_{n-1} = 10$. Again, we use (conditional) systematic resampling and resample only when the effective sample size falls below $0.8N$, where N is again the constant number of particles at every step.

Reversible-Jump MCMC Updates. The moves that update jumps in the RJMCMC algorithm are those used in Centanni and Minozzo (2006b) except that in the elementary change-point model, the jump sizes are always sampled from their full conditional posterior distributions).

Static-Parameter Updates. Within all three algorithms, a new value for the vector of static parameters is proposed using the m-fold convolution of a Gaussian random-walk Metropolis–Hastings kernel with the same covariance matrix across algorithms. More sophisticated updates for the static parameters could be constructed but we choose not to do so since this chapter's focus is on updating the time-varying parameters.

In what follows, a single 'iteration' or 'sweep' of one of these algorithms refers to first updating the static parameters (followed by the auxiliary-variable rejuvenation step for the RSMC-based PG sampler algorithm) and then updating the jumps using either a conditional SMC update or l RJMCMC updates. For the first example, we used $m = l = 500$ and for the second, we used $m = l = 1{,}000$.

Initial Values. Initial values for the static parameters are sampled from the prior. For the second example, we then divide the first two static parameters by 100 to avoid starting in a region with a very large number of jumps. This is done to reduce the computational cost for the first iterations in the VRPF-based PG sampler and RJMCMC algorithms and also because we have observed that the RSMC-based PG sampler can get stuck if started in a region with close to P jumps. A possible explanation of the latter phenomenon is provided in Subsection 4.4.6.

Implementation. The algorithms are implemented in Matlab (The MathWorks, Inc., 2015) on a single 2.66 GHz Intel 'Westmere' core using 4 gigabytes of RAM. In each case, the presented results are based on 60,000 iterations of which the first 10,000 are discarded as burn-in. We note that this is just a proof-of-concept implementation: significant speed-ups should be attainable using parallelisation techniques to which particle methods are amenable (Lee, Yau, Giles, Doucet & Holmes, 2010).

4.5.2 Elementary Change-Point Model

For the elementary change-point model, we used the simulated data shown in Figure 4.1. We chose a Gaussian prior on the static parameters, with covariance matrix $\mathrm{diag}(10^2, 10^2, 10, 10^3, 10^4)$ and truncated to $\mathbb{R} \times (0, \infty)^4$.

As shown in Figure 4.4, all three algorithms yielded comparable estimates for the marginal posterior distributions of the static parameters even when using only 25 particles. The bivariate correlation structure and sample autocorrelations were also similar but are omitted due to limited space. However, we encountered RJMCMC chains that seemed to get stuck in local modes for a considerable number of iterations. Such a chain is represented by the dashed line in the bottom row of Figure 4.4 and the corresponding trace plot for the parameter β is shown in Figure 4.5. We stress that this did not occur in all runs of the RJMCMC algorithm. This behaviour may be the result of the sampler finding it difficult to add, remove, or modify individual jumps in particular regions of the space. Such single-site updates are particularly inefficient in this case due to the gamma prior on the interjump times. We did not encounter such a behaviour in any of the PG samplers as they allow for a blocked update of (large parts of) the entire set of jumps.

The computing time for the auxiliary-variable rejuvenation and conditional SMC update (with AS) in the RSMC-based PG sampler was around 1.6 seconds on average, the conditional SMC update (with AS) for the VRPF-based PG sampler took around 2.5 seconds and $l = 500$ individual moves for the RJMCMC algorithm took around 2 seconds. The difference can partially be explained by the fact that the RSMC sampler is more amenable to code vectorisation than the VRPF because at each SMC step, it generates the same number of random variables for each particle.

4.5.3 Shot-Noise Cox-Process Model

For the shot-noise Cox-process example, we used the simulated data set shown in Figure 4.2. We chose a Gaussian prior for the vector of static parameters, with covariance matrix $\mathrm{diag}(10, 10, 10^2)$ and truncated to $(0, \infty)^3$. For the static-parameter updates we switched to a partially non-centred parametrisation of the jump sizes to improve mixing of the decay parameter κ.

As shown in Figure 4.6, the estimated marginal posterior densities from all three algorithms have similar modes. However, those obtained from the RSMC-based PG sampler are more concentrated. This difference is possibly due to the approximation described in Subsection 4.3.2 which restricts the number of jumps in any particular interval. In this model, it produces visibly different results because the exponential prior on the

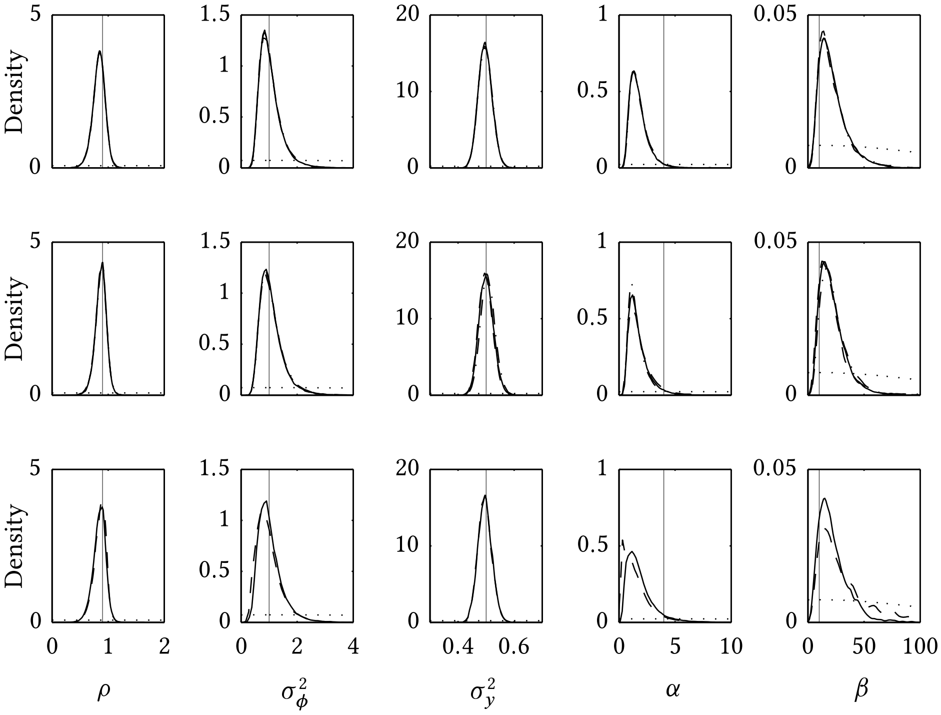

Figure 4.4 Kernel density estimates for the marginal posterior densities of the static parameters in the elementary change-point model. *Top row:* RSMC-based PG sampler algorithm with 100 particles (solid line), 50 particles (dashed line), 25 particles (dash-dotted line). *Middle row:* VRPF-based PG sampler with 100 particles (solid line), 50 particles (dashed line), 25 particles (dash-dotted line). *Bottom row:* two RJMCMC chains. Vertical lines indicate the true parameters; dotted lines show the prior densities.

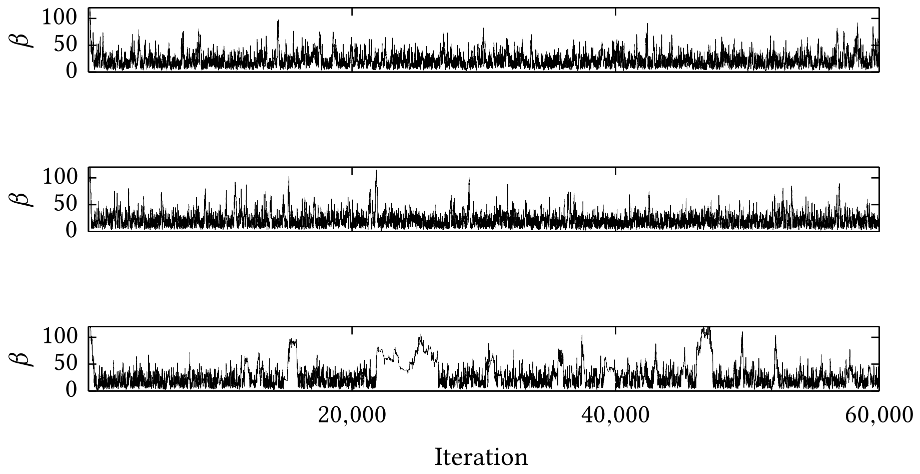

Figure 4.5 Trace plots for the scale-parameter estimates in the elementary change-point model. *Top:* RSMC-based PG sampler with 100 particles. *Middle:* VRPF-based PG sampler with 100 particles. *Bottom:* RJMCMC sampler.

interjump times allows large numbers of jumps to be placed close to each other with non-negligible probability. Thus, the posterior distribution of this model has tail regions with large numbers of jumps which the RSMC-based PG sampler algorithm rarely enters. This could also contribute to the differences in the autocorrelations in Figure 4.6. Note that the effect of this approximation can be reduced by decreasing the step size $t_n - t_{n-1}$.

4.6 Summary

In this chapter, we have demonstrated that PG samplers can be applied to piecewise deterministic processes and have presented a number of methodological developments in doing so. Numerical studies provide a comparative illustration of the performance of the proposed methods.

One of the methodological developments presented in this work involves a novel representation of the SMC sampler from Whiteley et al. (2011). This kind of representation, which embeds a 'variable-dimension' problem within a 'fixed-dimension' problem, may be useful more generally, e.g. for applying quasi-SMC methods (Gerber & Chopin, 2015) to variable-dimension problems. An extension of this representation to allow for multiple-birth-move kernels is left for future research.

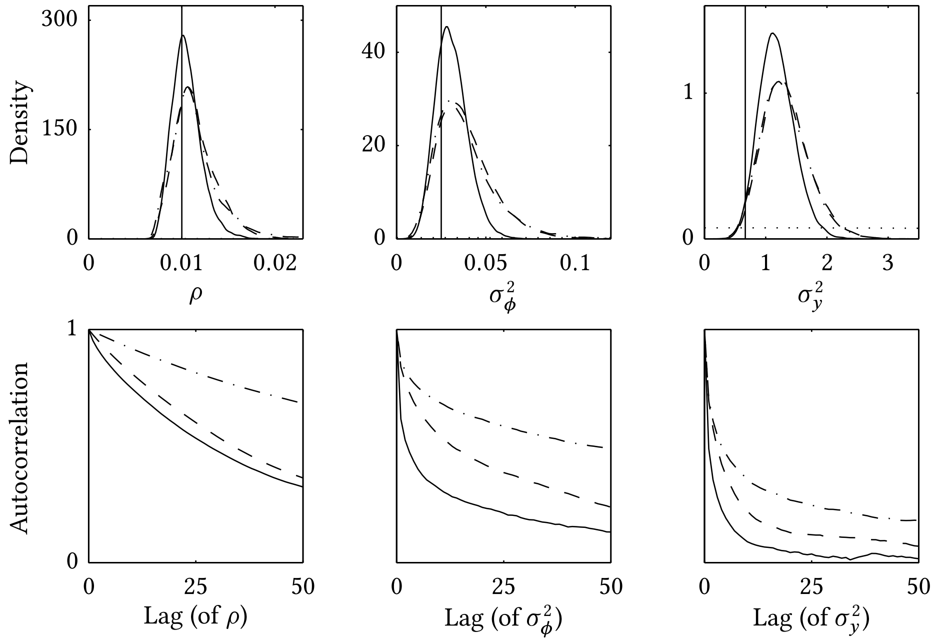

Figure 4.6 Static-parameter estimates for the shot-noise Cox-process example. Based on the RSMC-based PG sampler algorithm with 100 particles (solid line), the VRPF-based PG sampler with 100 particles (dashed line), and an RJMCMC sampler (dash-dotted line). *Top row:* kernel density estimates of the marginal posterior densities. Vertical lines indicate true parameters; dotted lines represent prior densities. *Bottom row:* autocorrelations.

5 Non-Centred Particle Gibbs Samplers for Compound Poisson-Process Models

5.1 Introduction

5.1.1 Motivation

In this chapter, we devise a particle Gibbs sampler for static-parameter estimation in partially or noisily observed compound Poisson processes. The algorithm is based on a non-centred parametrisation, described in Section 5.2, in order to reduce the impact of the correlation between the latent point process and the parameters on the mixing of the Gibbs sampler. Some modifications for enhancing the efficiency of the (conditional) sequential Monte Carlo algorithm at the heart of the particle Gibbs sampler are mentioned in Section 5.3. Finally, in Section 5.4, we provide illustrative results demonstrating the performance of the algorithm on a challenging Lévy-driven stochastic volatility model.

This paper considers a class of statistical models which are based around a compound Poisson process $L = (L_t)_{t \in [0,T]}$, for some $T \in (0, \infty)$. Such a process can be represented as

$$L_t := \sum_{j=1}^{K} E_j \, \mathbb{1}_{[0,t]}(S_j),$$

for $t \in [0, T] =: \mathsf{T}$. Here, the *number of jumps,* K, and the ordered *jump times,* $0 < S_1 < S_2 < \ldots < S_K$, are generated by a Poisson process on T with finite intensity $\lambda = l(\theta)$ which we assume to be constant, for simplicity. Here, $l \colon \Theta \to (0, \infty)$ is some known function which determines the intensity, and the *jump sizes* $E_1, E_2, \ldots, E_K$ are IID random variables distributed according to some distribution ν^θ on $\mathsf{E} \subseteq \mathbb{R}$. For convenience, we hereafter write $\Psi := (K, S_{1:K}, E_{1:K})$.

Throughout, we assume that the process is latent, i.e. Ψ can only be partially or noisily observed. The aim is to conduct inference about the parameters Θ that parametrise both the compound Poisson process and the likelihood function of the observations, $g^{\theta}(\,\cdot\,|\psi)$. In a Bayesian framework, this entails computing or at least approximating the *marginal* posterior distribution of Θ.

Usually, *Markov chain Monte Carlo* (MCMC) algorithms are used to approximate such posterior distributions. In addition, to circumvent intractable integrals, we usually have to work on an extended space and approximate the *joint* posterior distribution of Θ and Ψ, denoted $\pi(\mathrm{d}\theta \times \mathrm{d}\psi)$. Due to the difficulty of constructing efficient global updates for (Θ, Ψ), the MCMC transitions are almost always based around a convolution of local, component-wise updates, resulting in so called Gibbs samplers, Metropolis-within-Gibbs algorithms, or combinations of the two. Within these algorithms, the components of Ψ are usually updated individually using a particular type of MCMC update known as *reversible-jump Markov chain Monte Carlo* (RJMCMC) kernel (Green, 1995). A single sweep of such a sampler is outlined in Algorithm 5.1, where $\pi(\mathrm{d}\psi|\theta)$ and $\pi(\mathrm{d}\theta|\psi)$ denote, respectively, the full conditional distributions of Ψ and of Θ under π.

5.1 Algorithm (centred Metropolis-within-Gibbs).

(1) Update (components of) Ψ using a $\pi(\mathrm{d}\psi|\theta)$-invariant MCMC kernel.

(2) Update Θ using a $\pi(\mathrm{d}\theta|\psi)$-invariant MCMC kernel.

Unfortunately, as is well known, such component-wise updates impede mixing of the MCMC chain whenever components are highly correlated. To reduce the impact of correlation between Θ and Ψ on the mixing of the MCMC chain, Roberts, Papaspiliopoulos and Dellaportas (2004) propose Metropolis-Within-Gibbs samplers with single-site RJMCMC updates for Ψ based on various *non-centred parametrisations* (NCPs). Roughly speaking, NCPs are parametrisations under which certain subsets of parameters/latent variables are independent a-priori.

One of these NCPs is based on a representation by Ferguson and Klass (1972) which was also used by Griffin and Steel (2006). It was found to be highly effective in Roberts et al. (2004) as soon as the latent point process Ψ could be updated as a single block.

> [W]hen multiple updates of Ψ are performed for every update of the parameters, the [non-centred algorithm] that is

> based on the Ferguson–Klass representation is much more efficient than the others … [B]ig steps in parameter space can be achieved with this scheme. However, the mixing of Ψ is slow, since only small moves in the point process space are attempted … Multiple updates of Ψ are computationally extremely expensive and should be avoided. We have experimented extensively with more sophisticated approaches for updating Ψ, without significant success. However, *if more efficient methods for this updating step could be found, the [non-centred algorithm] that is based on the Ferguson–Klass representation would become very attractive* [emphasis added]. (Roberts et al., 2004, pp. 387–388)

5.1.2 Contribution

With the introduction of *particle Gibbs* (PG) samplers in Andrieu et al. (2010) it has become possible to jointly update all the components of Ψ and thus potentially reduce the impact of correlation *between the components of* Ψ on the mixing of the MCMC chain. Hence, it is only natural to devise a PG sampler that adopts the NCP based on the Ferguson–Klass representation to simultaneously reduce the impact of the correlation *between* Θ *and* Ψ. This is the focus of this work. We also apply the resulting non-centred PG sampler to a particularly challenging Lévy-driven stochastic volatility model for which we also point out some identifiability issues that have not been recognised in the literature. We note that the utility of reparametrisations in PG samplers has also recently been recognised by Fearnhead and Meligkotsidou (2014).

5.2 Non-Centred Metropolis-Within-Gibbs Algorithm

5.2.1 Actual Target Distribution

A generic statistical model based around a compound Poisson process L has a posterior distribution – the *target distribution* – of the form

$$\pi(\mathrm{d}\theta \times \mathrm{d}\psi) \propto \gamma(\mathrm{d}\theta \times \mathrm{d}\psi)$$

on $\Theta \times \Psi$, where

$$\gamma(\mathrm{d}\theta \times \mathrm{d}\psi) := \varpi(\mathrm{d}\theta)\Pi^{\theta}(\mathrm{d}\psi)g^{\theta}(y_{\mathsf{T}}|\psi) \tag{5.1}$$

is some positive measure with $\varpi(\mathrm{d}\theta)$ being some (prior) distribution for a set of static parameters, Θ, whose support is (a subset of) Θ. Furthermore, $\Pi^{\theta}(\mathrm{d}\psi)$ is a distribution on

$$\Psi := \bigcup_{k=0}^{\infty}(\{k\} \times \mathsf{T}_k \times \mathsf{E}^k),$$

where $\mathsf{T}_k := \{s_{1:k} \in \mathsf{T}^k | s_1 < s_2 < \ldots < s_k\}$ is the support of k ordered jump times in T. For the moment, we take $\mathsf{E} = \mathbb{R}$. The generalistion to $\mathsf{E} = \mathbb{R}^d$ is discussed Section 5.5.

The probability measure $\Pi^{\theta}(\mathrm{d}\psi)$ is the distribution of points – ordered according to the first component – which have been generated by a *Poisson point process* (PPP) on $\mathsf{T} \times \mathsf{E}$ with intensity measure

$$\eta^{\theta}(\mathrm{d}s \times \mathrm{d}e) := \lambda\, [\mathrm{Leb}|_{\mathsf{T}} \otimes \nu^{\theta}](\mathrm{d}s \times \mathrm{d}e),$$

where we recall that ν^{θ} is a probability measure on E and the intensity parameter $\lambda = l(\theta)$ is a function of the static parameters. Here and throughout, $\mu|_C$ denotes the restriction of a measure μ to a measurable set C and Leb denotes the Lebesgue measure on $\mathbb{R}$. To simplify the presentation, we assume that the intensity measure η^{θ} is assumed to be absolutely continuous with respect to the Lebesgue measure (on $\mathsf{T} \times \mathsf{E}$).

Finally, $g^{\theta}(y_{\mathsf{T}}|\psi)$ is the likelihood of the collection of observations $y_{\mathsf{T}} := (y_1, \ldots, y_P)$, which are recorded at times $0 < \tilde{t}_1 < \ldots < \tilde{t}_P \leq T$. For simplicity, we will subsequently assume that $\tilde{t}_P = T$. Moreover, we sometimes let $y_{(s,t]}$ denote the subset of observations that have been recorded in the interval $(s, t]$.

5.2.2 Non-Centred Parametrisation

The reparametrisation adopted in this paper is based on a representation derived in Ferguson and Klass (1972). The use of this representation to derive a (partially) non-centred parametrisation for compound Poisson processes was suggested by Roberts et al. (2004) and such reparametrisations were also extensively used by Griffin and Steel (2006) in the context of Lévy-driven stochastic volatility models. Alternative reparametrisations were suggested in Roberts et al. (2004) but were found to be inferior in the presence of efficient updates of the latent point process.

The basic idea is to take points distributed according to a unit-intensity PPP on $\mathsf{T} \times [0, \bar{\lambda})$, denoted $\widetilde{\Psi}$, where $\bar{\lambda}$ is some value in $[\lambda, \infty)$. First, we discard those points whose second components exceeds λ and divide the second component of the remaining points by λ. This is sometimes called *thinning* and leaves a set of points distributed according to a PPP on $\mathsf{T} \times [0, 1]$ with intensity measure $\lambda \operatorname{Leb}|_{\mathsf{T}\times[0,1]}$. These points are then transformed into realisations of points $\Psi \sim \Pi^{\theta}$ by applying the inverse *cumulative distribution function* (CDF)-method to the second component. For completeness and to set up some notation, we outline the formal justification of this reparametrisation below.

Define the space

$$\widetilde{\Psi} := \bigcup_{k=0}^{\infty} (\{k\} \times \mathsf{T}_k \times [0, \bar{\lambda}))$$

and let $\widetilde{\Pi}(\mathrm{d}\widetilde{\psi})$ be the distribution of points $\widetilde{\Psi} = (\widetilde{K}, \widetilde{S}_{1:\widetilde{K}}, \widetilde{E}_{1:\widetilde{K}})$ which are generated by a unit-intensity PPP on $\mathsf{T} \times (0, \bar{\lambda}]$. Again these points are taken to be ordered according to their first components. In the following, we describe how $\widetilde{\Psi}$ is first thinned and then transformed to obtain the desired points $\Psi \sim \Pi^{\theta}$.

Write $\mathbb{N}_n := \{ k \in \mathbb{N} \mid k \leq n \}$. Let

$$H := \left\{ j \in \mathbb{N}_{\widetilde{K}} \;\middle|\; \widetilde{E}_j \leq \lambda \right\}$$

be the indices of points in $\widetilde{\Psi}$ whose second component does not exceed $\lambda = l(\theta)$ and set $K := \# H$. Similarly, let

$$\widehat{H} := \left\{ j \in \mathbb{N}_{\widetilde{K}} \;\middle|\; \widetilde{E}_j > \lambda \right\} = \mathbb{N}_{\widetilde{K}} \setminus H$$

be the set of the remaining indices and set $\widehat{K} := \# \widehat{H}$. Collect the elements of H and $\widehat{H}$ in vectors $h = h_{1:K}$ and $\hat{h} = \hat{h}_{1:\widehat{K}}$, in increasing order. Note that $H, \widehat{H}, h$ and $\hat{h}$ depend on Θ through λ.

Points $\Psi \sim \Pi^\theta$ are then the result of the one-to-one reparametrisation

$$(\Theta, \widetilde{\Psi}) \longleftrightarrow (\Theta, \Psi, \widehat{\Psi}), \tag{5.2}$$

where the left hand side represents an NCP and the right hand side represents a *centred parametrisation* (CP), i.e.

(1) $\widetilde{\Psi} := (\widetilde{K}, \widetilde{S}_{1:\widetilde{K}}, \widetilde{E}_{1:\widetilde{K}})$ represents all the points under the NCP,

(2) $\Psi = (K, S_{1:K}, E_{1:K})$ is a sample from the desired PPP under the CP,

(3) $\widehat{\Psi} := (\widehat{K}, \widehat{S}_{1:\widehat{K}}, \widehat{E}_{1:\widehat{K}})$ are 'artificial' points added to Ψ under the CP.

In the following, we specify the one-to-one transformation involved in Equation 5.2.

- For $j \in \mathbb{N}_{\widehat{K}}$, the points in $\widehat{\Psi}$ are related to those under the NCP via

$$(\widehat{S}_j, \widehat{E}_j) := \big(\widetilde{S}_{\hat{h}_j}, \widetilde{E}_{\hat{h}_j}\big).$$

- For $j \in \mathbb{N}_K$, the points in Ψ are related to those under the NCP via

$$(S_j, E_j) = \phi^\theta\big(\widetilde{S}_{h_j}, \widetilde{E}_{h_j}\big).$$

Above, letting $\overline{F}^\theta$ denote the (generalised) inverse of the CDF F^θ associated with ν^θ, the function $\phi^\theta : \mathsf{T} \times (0, \lambda] \to \mathsf{T} \times \mathsf{E}$ is defined by

$$(\tilde{s}, \tilde{e}) \mapsto \phi^\theta(\tilde{s}, \tilde{e}) := [\tilde{s}, \overline{F}^\theta(1 - \tilde{e}/\lambda)].$$

This function transforms the thinned points from the unit-intensity PPP $\widetilde{\Psi}$ (under the NCP) – specifically, those points whose second component does not exceed $\bar{\lambda}$ – into the desired points Ψ which make up the compound Poisson process (under the CP). Since $\overline{F}^\theta$ is the (generalised) inverse of the CDF associated with ν^θ, the function ϕ^θ can be interpreted as applying the inverse CDF method.

Letting Ψ and $\widetilde{\Psi}$ be related as in Equation 5.2, we obtain the following proposition, stated here for completeness.

5.2 Proposition. *Let $\theta \in \Theta$ and $\widetilde{\Psi} \sim \widetilde{\Pi}$, then $\Psi \sim \Pi^\theta$.*

Proof. The fact that Ψ is the set of ordered points distributed according to a PPP on $\mathsf{T} \times \mathsf{E}$ follows by the independence property and the Mapping Theorem for PPPs (Kingman, 1992, p. 18). In particular, for $A \in \mathcal{B}(\mathsf{T} \times \mathsf{E})$, letting $\mathrm{D}\,\phi^{\theta}$ denote the Jacobian of ϕ^{θ},

$$\begin{aligned}
&[\mathrm{Leb}^{\otimes 2}\,|_{\mathsf{T}\times(0,\lambda)} \circ (\phi^{\theta})^{-1}](A) \\
&\quad = \int_A |\det(\mathrm{D}\,\phi^{\theta})((\phi^{\theta})^{-1}(s,e))|^{-1} \mathrm{d}s\, \mathrm{d}e \\
&\quad = \lambda\, [\mathrm{Leb}|_{\mathsf{T}} \otimes \nu^{\theta}](A) \\
&\quad = \eta^{\theta}(A).
\end{aligned}$$

Thus, the Ψ is the set of ordered points distributed according to a PPP on $\mathsf{T} \times \mathsf{E}$ with intensity measure η^{θ}. □

5.2.3 Extended Target Distribution

In this subsection, we specify an extended distribution $\tilde{\pi}(\mathrm{d}\theta \times \mathrm{d}\psi)$. First, this distribution admits the distribution of interest, $\pi(\mathrm{d}\theta \times \mathrm{d}\psi)$, as a marginal. Second, a Gibbs sampler or Metropolis-within-Gibbs algorithm targeting this distribution makes use of an NCP based on the representation outlined above.

Centred Parametrisation. The original formulation of the model in Equation 5.1 uses a CP, i.e. it implies prior dependence between Θ and Ψ. To permit the reparametrisation from Equation 5.2, we augment the target distribution $\pi(\mathrm{d}\theta \times \mathrm{d}\psi)$ with

$$\hat{\Psi} \sim \hat{\Pi}^{\theta} := \tilde{\Pi}|_{\mathsf{T}\times(\lambda,\bar{\lambda}]}\,.$$

Here, $\tilde{\Pi}|_{\mathsf{T}\times(\lambda,\bar{\lambda}]}$ is the distribution of points – again ordered according to the first component – that have been generated by a unit-intensity PPP on $\mathsf{T} \times (\lambda, \bar{\lambda}]$. We thus obtain a distribution $\hat{\pi} \propto \hat{\gamma}$ over the parameters on the right hand side in Equation 5.2, defined by

$$\hat{\gamma}(\mathrm{d}\theta \times \mathrm{d}\psi \times \mathrm{d}\hat{\psi}) := \gamma(\mathrm{d}\theta \times \mathrm{d}\psi)\hat{\Pi}^{\theta}(\mathrm{d}\hat{\psi}).$$

Non-Centred Parametrisation. Recall that y_T represents the collection of all available observations (in the interval $\mathsf{T} = [0, T]$). With the abuse of notation induced by writing

$$g^\theta(y_\mathsf{T}|\tilde{\psi}) = g^\theta(y_\mathsf{T}|\psi)$$

whenever Ψ and $\widetilde{\Psi}$ are related according to Equation 5.2), we can equivalently define an extended target distribution $\tilde{\pi} \propto \tilde{\gamma}$ parametrised via the left hand side in Equation 5.2, i.e. by

$$\tilde{\gamma}(\mathrm{d}\theta \times \mathrm{d}\tilde{\psi}) = \varpi(\mathrm{d}\theta)\widetilde{\Pi}(\mathrm{d}\tilde{\psi})g^\theta(y_\mathsf{T}|\tilde{\psi}).$$

This parametrisation represents an NCP because Θ and $\widetilde{\Psi}$ are independent a-priori. If the observations are not too informative, then the dependence between Θ and $\widetilde{\Psi}$ under $\tilde{\pi}$ should be smaller than the dependence between Θ and Ψ under $\hat{\pi}$. Hence, such a parametrisation is often beneficial in the context of Gibbs sampling.

5.2.4 The Algorithm

A single iteration of the non-centred MCMC algorithm proposed by Roberts et al. (2004) is given in Algorithm 5.3.

5.3 Algorithm (non-centred Metropolis-within-Gibbs).

```
(1) Update $\widetilde{\Psi}$ by sampling                                  // using the CP
      (i) (components of) $\Psi$ using a $\pi(\mathrm{d}\psi|\theta)$-invariant MCMC kernel,
      (ii) $\widehat{\Psi} \sim \hat{\pi}(\mathrm{d}\hat{\psi}|\theta, \psi) = \widetilde{\Pi}|_{\mathsf{T}\times(\lambda,\bar{\lambda}]}(\mathrm{d}\hat{\psi})$.
(2) Update $\Theta$ using a $\tilde{\pi}(\mathrm{d}\theta|\tilde{\psi})$-invariant MCMC kernel.  // using the NCP
```

5.4 Remark. *Note that Step 1 is independent of the previously sampled points in $\widehat{\Psi}$ as these are sampled again from their full conditional distribution. Furthermore, assuming that an* Metropolis–Hastings *(MH) kernel is used to update the parameters Θ, Step 2 is independent of those points in $\widehat{\Psi}$ that lie in the set*

$$\mathsf{T} \times (\lambda \vee \lambda^\star, \bar{\lambda}].$$

Here, $\lambda^\star = l(\theta^\star)$, where $\theta^\star$ is the value for Θ proposed as part of the MH kernel applied in Step 2. Hence, we need not actually determine $\bar{\lambda}$ nor do *points in* $\mathsf{T} \times (\lambda \vee \lambda^\star, \bar{\lambda}]$ *(under the* NCP*) actually have to be sampled. Thus, the algorithm can also deal with models for which* $\lambda = l(\Theta)$ *is not bounded but only almost-surely finite.*

The key reason for why this sampling scheme is often preferable to Algorithm 5.1 is that in Step 1, we are not fixing the points Ψ under our actual target distribution π. Instead, Ψ is a function of Θ and $\widetilde{\Psi}$. This will often allow greater movement in the θ-direction in Step 2.

Even more so as pointed out by Roberts et al. (2004), Griffin and Steel (2006), the particular NCP outlined above has another advantage: a decrease in $\lambda = l(\theta)$ coincides with the removal of those points from Ψ that are associated with the smallest jump sizes. Similarly, an increase in λ coincides with adding points to Ψ that have relatively small jumps. The above construction is therefore termed *dependent thinning* in Griffin and Steel (2006). Usually, adding or removing a single small jump has little impact on the posterior density. This property can further facilitate movement in the θ-direction compared the other NCPs from Roberts et al. (2004) which add or remove jumps with arbitrary jump size.

5.3 Non-Centred Particle Gibbs Sampler

5.3.1 Motivation

The NCP discussed in the previous section can help reduce the impact of correlation *between* Θ *and* Ψ on the efficiency of Algorithm 5.3 but it does not alleviate inefficiencies resulting from the correlation *between individual components of* Ψ if these are still updated one-at-a-time.

A strategy for systematically updating Ψ in one block is offered by the *conditional sequential Monte Carlo* (CSMC) kernels introduced by Andrieu et al. (2010) and described in Section 3.4 of this work. Simple *sequential Monte Carlo* (SMC) algorithms have been applied to latent point processes in Godsill and Vermaak (2004), Chopin et al. (2013). More sophisticated SMC algorithms based around the SMC-sampler framework (Del Moral et al., 2006b) have been developed in Del Moral et al. (2007), Whiteley et al. (2011), Martin et al. (2013) and in Chapter 4 of this work. As pointed out in Whiteley et al. (2011) and further analysed in Chapter 4, the latter class of SMC algorithms can introduce a substantial bias in the case of exponentially-distributed interjump times (as is the case here).

We therefore employ simple SMC algorithms even though these are potentially very inefficient (in the sense that sample impoverishment is severe). Our SMC algorithm slightly differs from that described in Chopin et al. (2013) in two ways. Firstly, following Chopin (2002), we allow for more than one observation to be included per SMC step in order to speed up the algorithm. Secondly, we employ a slightly different parametrisation which permits the use of the variance-reduction techniques: *backward sampling* (BS) and *ancestor sampling* (AS) (Whiteley, 2010; Lindsten et al., 2012) within PG samplers. These were described in Section 3.4.

Alternatives. There are, of course, alternatives to PG samplers for conducting inference in the models described here. For instance, we could use SMC-based pseudo-marginal MH algorithms known as *particle marginal Metropolis–Hastings* (PMMH) algorithms (Andrieu et al., 2010) or pseudo-marginal SMC algorithms based around PMMH updates known as *SMC-squared* (Chopin et al., 2013).

By construction, these methods are robust to strong correlation of Θ and Ψ under π. However, these methods tend to require large numbers of particles. For instance, Chopin et al. (2013) report the need for around 500 to 3,000 particles for a moderately-long time series in the simplest version of the Lévy-driven stochastic volatility model discussed in Section 5.4. We have found such numbers of particles to be prohibitively high for implementations in high-level programming languages such as R (R Development Core Team, 2014) or Matlab (The MathWorks, Inc., 2015). With smaller numbers of particles, pseudo-marginal MH kernels are well known to suffer from the so called 'stickiness' problem, i.e. from long periods of high rejection rates.

5.3.2 Conditional SMC Kernel

In this subsection, we describe some of the details of the (conditional) SMC algorithms which are needed to deal with the specific class of models analysed here (and in the previous chapter).

Step Size. For $I \in \mathbb{N}$ we let $0 = t_0 < t_1 < \ldots < t_I = T$. Here, $(t_{i-1}, t_i]$ is the time window of the latent compound Poisson process targeted at the

ith SMC step. That is, at the ith SMC step, we both assimilate observations and propose jumps in the interval $(t_{i-1}, t_i]$.

Without loss of generality and to simplify the presentation, we assume that $(t_i)_{i \in \mathbb{N}_I}$ is a subsequence of $(\tilde{t}_p)_{p \in \mathbb{N}_P}$. Note that the commonly-used strategy of assimilating one observation per SMC step corresponds to the special case $(t_i)_{i \in \mathbb{N}_I} = (\tilde{t}_p)_{p \in \mathbb{N}_P}$.

If the weights do not deteriorate too quickly over SMC steps, i.e. if the effective sample size does not decrease too steeply after a single SMC step, it can be preferable to increase this step size to reduce the computational cost of the algorithm (Chopin, 2002).

Reparametrisation. For the CSMC kernel, we need to apply a further reparametrisation to ensure that the computational cost of performing a single step of BS or AS does not grow with T (on average). This can be achieved by parametrising the compound Poisson process not in terms of jump sizes but in terms of the values of the process at the jump times. The latter coincides with the representation used in the previous chapter.

Recall that the compound Poisson process is denoted $L = (L_t)_{t \in \mathrm{T}}$. We can apply another one-to-one reparametrisation of the form

$$(\Theta, \Psi) \longleftrightarrow (\Theta, \dot{\Psi}_{1:I}), \tag{5.3}$$

where

$$\dot{\Psi}_i := (K_i, S_{i,1:K_i}, \widetilde{L}_{i,1:K_i})$$

denotes the points (as well as their number) of the latent compound Poisson process whose first component (the jump time) falls in the interval $(t_{i-1}, t_i]$. These points are again ordered according to their first component, i.e. $t_{i-1} < S_{i,1} < \ldots < S_{i,K_i} \leq t_i$. The second components, $\widetilde{L}_{i,1:K_i}$, no longer represent the actual jump sizes but are now taken to be the values of the compound Poisson process L at the jump times. That is, $\widetilde{L}_{i,j} := L_{S_{i,j}}$, for any $j \in \mathbb{N}_{K_i}$. This is corresponds to the terminology 'jump size' used in the previous chapter.

With this reparametrisation, we may write the distribution targeted by the (conditional) SMC algorithm as $\pi^\theta(\mathrm{d}\dot{\psi}_{1:I}) \propto \gamma^\theta(\mathrm{d}\dot{\psi}_{1:I})$, where

$$\gamma^\theta(\mathrm{d}\dot{\psi}_{1:I}) := \prod_{i=1}^{I} \dot{\Pi}_i^\theta(\mathrm{d}\dot{\psi}_i | \dot{\psi}_{1:i-1}) g^\theta(y_{(t_{i-1}, t_i]} | \dot{\psi}_{1:i}, y_{(t_0, t_{i-1}]}).$$

Here, $\dot{\Pi}_i^\theta(\mathrm{d}\dot{\psi}_i|\dot{\psi}_{1:i-1})$ denotes the conditional prior distribution of the points in the interval $(t_{i-1}, t_i]$, $\dot{\Psi}_i$, and we again slightly abuse notation by writing $g^\theta(y_\mathrm{T}|\dot{\psi}_{1:I}) = g^\theta(y_\mathrm{T}|\psi)$ if Ψ and $\dot{\Psi}$ are related as in Equation 5.3. Note that the observations taken in disjoint intervals are not necessarily assumed to be independent given the PPP and given Θ. Indeed, in the example considered in Section 5.4, we analytically integrate out a subset of the static parameters which means that the observations in disjoint intervals are no longer conditionally independent given the PPP and given the remaining parameters.

The SMC algorithm then targets $\pi^\theta(\mathrm{d}\dot{\psi})$ using a sequence of intermediate distributions

$$\pi_i^\theta(\mathrm{d}\dot{\psi}_{1:i}) \propto \gamma_i^\theta(\mathrm{d}\dot{\psi}_{1:i}),$$

where

$$\gamma_i^\theta(\mathrm{d}\dot{\psi}_{1:i}) := \prod_{j=1}^{i} \dot{\Pi}_j^\theta(\mathrm{d}\dot{\psi}_j|\dot{\psi}_{1:j-1}) g^\theta(y_{(t_{j-1},t_j]}|\dot{\psi}_{1:j}, y_{(t_0,t_{j-1}]}).$$

5.3.3 Full Algorithm

The full PG sampler is outlined in Algorithm 5.5. Note that the comments made in Remark 5.4 fully apply to this algorithm, too. That is, $\lambda = l(\Theta)$ only needs to be almost-surely bounded.

5.5 Algorithm (non-centred particle Gibbs).

(1) Update $\widetilde{\Psi}$ by *// using the CP*

(i) reparametrising $(\Theta, \Psi) \to (\Theta, \dot{\Psi}_{1:I})$,

(ii) updating $\dot{\Psi}$ using a CSMC kernel (with BS/AS, if possible),

(iii) reparametrising $(\Theta, \dot{\Psi}_{1:I}) \to (\Theta, \Psi)$,

(iv) sampling $\widehat{\Psi} \sim \tilde{\pi}(\mathrm{d}\hat{\psi}|\theta, \psi) = \widetilde{\Pi}|_{\mathrm{T}\times(\lambda,\bar{\lambda}]}(\mathrm{d}\hat{\psi})$.

(2) Update Θ using a $\tilde{\pi}(\mathrm{d}\theta|\tilde{\psi})$-invariant MCMC kernel. *// using the NCP*

5.4 Application to Lévy-Driven Stochastic Volatility Models

5.4.1 Model Description

In this section, we apply the above-mentioned PG sampler to a Lévy-driven stochastic volatility model and compare the algorithm's performance with that of a non-centred RJMCMC algorithm.

Inference for the particular model considered here, which was introduced by Barndorff-Nielsen and Shephard (2001), has previously been performed via (non-SMC based) MCMC methods (Roberts et al., 2004; Griffin & Steel, 2006), particle MCMC methods (Andrieu et al., 2010) and via hierarchical (or pseudo-marginal) SMC samplers (Chopin et al., 2013).

Log-Asset Price Process. Under the model, the log-price of some asset evolves according to the stochastic differential equation

$$\mathrm{d}X_t = \left[\mu + \sum_{m=1}^{M} \beta^m \sqrt{V_t^m}\right]\mathrm{d}t + \sqrt{V_t}\mathrm{d}B_t + \sum_{m=1}^{M} \rho^m \mathrm{d}\bar{L}_t^m. \tag{5.4}$$

Here, $B := (B_t)_{t\geq 0}$ denotes standard Brownian motion, $L^m := (L_t^m)_{t\geq 0}$, for $m \in \mathbb{N}_M$, are independent Lévy processes, and $\bar{L}^m := (\bar{L}_t^m)_{t\geq 0}$, with $\bar{L}_t^m := L_t^m - \mathbb{E}[L_t^m]$, represents the compensated process associated with L^m. In addition, $\beta^m, \rho^m \in \mathbb{R}$ are the risk-premium and linear-leverage parameters for the mth component process.

Latent Volatility Process. The (instantaneous) volatility process, denoted $V := (V_t)_{t\geq 0}$, is given by

$$V_t := \sum_{m=1}^{M} V_t^m,$$

where the components $V^1, \dots, V^M$ are independent processes satisfying the Ornstein–Uhlenbeck stochastic differential equation

$$\mathrm{d}V_t^m = -\kappa^m V_t^m \mathrm{d}t + \mathrm{d}L_t^m.$$

Here, $\kappa^m > 0$ denotes the decay rate of the mth component-volatility process. As in Roberts et al. (2004), Griffin and Steel (2006) we assume, for simplicity, that the marginally, V_t^m is gamma-distributed. This implies that L^m reduces to a compound Poisson process with some rate $\lambda^m > 0$ and exponential jump-size distribution with rate $\zeta > 0$, written Exp_ζ.

The stochastic differential equation admits the closed-form solution

$$V_t^m = V_0^m \exp(-\kappa^m t) + \sum_{k=1}^{K^m} E_k^m \exp(-\kappa^m (t - S_k^m)) \, \mathbb{1}_{[0,t]}(S_k^m),$$

where $(K^m, S^m_{1:K^m}, E^m_{1:K^m}) =: \Psi^m$ comprises the number of jumps, the jump times and the jump sizes associated with the compound Poisson process driving the mth component-volatility process.

Observed Aggregate Log-Returns. Let Y_p denote the pth observation, i.e. the aggregate log-return over the time interval $(\tilde{t}_{p-1}, \tilde{t}_p]$. By the properties of Brownian motion, Equation 5.4 implies that conditional on the component processes up to time $\tilde{t}_p$, the pth aggregate log-return Y_p is distributed according to a normal distribution with mean $m_{(\tilde{t}_{p-1}, \tilde{t}_p]}$ and variance $V^\star_{(\tilde{t}_{p-1}, \tilde{t}_p]}$, which are defined by

$$\begin{aligned} m_{(s,t]} &:= (t - s)\mu + \sum_{m=1}^{M} \beta^m V^{m,\star}_{(s,t]} + \rho^m \overline{L}^{m,\star}_{(s,t]}, \\ V^\star_{(s,t]} &:= \sum_{m=1}^{M} V^{m,\star}_{(s,t]}, \end{aligned}$$

where $V^{m,\star}_{(s,t]} := V^{m,\star}_{[0,t]} - V^{m,\star}_{[0,s]}$, $\overline{L}^{m,\star}_{(s,t]} := \overline{L}^{m,\star}_{[0,t]} - \overline{L}^{m,\star}_{[0,s]}$, and with

$$\begin{aligned} V^{m,\star}_{[0,t]} &:= \int_0^t \mathrm{d}V_s^m = \frac{1}{\kappa^m}(L_t^m + V_0^m - V_t^m), \\ \overline{L}^{m,\star}_{[0,t]} &:= \int_0^t \mathrm{d}\overline{L}_s^m = \int_0^t \mathrm{d}L_s^m - \mathbb{E}[L_t^m] = L_t^m - \frac{t\lambda^m}{\zeta}. \end{aligned}$$

5.4.2 Choice of Priors

Intuitively, as argued in Roberts et al. (2004), κ^m and λ^m should be highly correlated, a-posteriori. To improve mixing of the MCMC chain, we follow Griffin and Steel (2006) in reparametrising according to

$$\big(\kappa^{1:M}, \lambda^{1:M}, \zeta\big) \longleftrightarrow \big(\Delta_\kappa^{1:M}, w^{1:M}, \xi, \zeta^{-1}\big),$$

where, defining $\varepsilon^m := \lambda^m / \kappa^m$ and $\varepsilon := \sum_{m=1}^M \varepsilon^m$,

- $\Delta_\kappa^m := \kappa^m - \sum_{l=1}^{m-1} \kappa^l$ is the difference between the decay rates in the mth and $(m-1)$th component volatility processes, V^m and V^{m-1},
- $w^m := \varepsilon^m / \varepsilon$ is a weight which governs the influence of the mth component volatility process, V^m, on the overall volatility process, V,
- $\xi := \varepsilon/\zeta$ and $\omega^2 := \varepsilon/\zeta^2$ represent the stationary mean and variance of the overall volatility process, V.

Let $\mathrm{Gam}_{\alpha,\beta}$ be the gamma distribution with shape parameter α and scale parameter β. If we assume that $V_0^m \sim \mathrm{Gam}_{w^m \xi^2/\omega^2, \omega^2/\xi}$, then the latter distribution is the marginal distribution of V_t^m, $t \geq 0$, and hence the stationary distribution of the process V^m. This implies that $\mathbb{E}[V_t^m] = w^m \xi$ and $\mathbb{V}[V_t^m] = w^m \omega^2$, for any $t \geq 0$. For the same reason, $V_t \sim \mathrm{Gam}_{\xi^2/\omega^2, \omega^2/\xi}$ and hence $\mathbb{E}[V_t] = \xi$ as well as $\mathbb{V}[V_t] = \omega^2$, for any $t \geq 0$.

As in Griffin and Steel (2006), we use a weakly informative prior on the static parameters, $(\Theta, \widetilde{\Theta})$, where

$$\begin{aligned} \Theta &:= (\Delta_\kappa^{1:M}, w^{1:M}, \xi, \zeta^{-1}), \\ \widetilde{\Theta} &:= (\mu, \beta^{1:M}, \rho^{1:M}). \end{aligned}$$

A-priori, all parameters except $w^{1:M}$ are independent. The differences in the component-specific decay-rate parameters are given the prior $\Delta_\kappa^1, \dots, \Delta_\kappa^M \sim \mathrm{Gam}_{1,1}$. This choice of priors imposes the identifiability constraint $\kappa^{m+1} > \kappa^m$ on the component processes to circumvent the so-called label-switching problem. In addition, $w^{1:M} \sim \mathrm{Dir}_{\iota_M}$, where Dir_α denotes the Dirichlet distribution with parameter vector α and ι_m denotes a vector of ones of length m. Finally, a-priori, $\xi, \zeta^{-1} \sim \mathrm{Gam}_{2,5}$.

The parameters of the observation equation (Equation 5.4) are given normal priors, i.e. , a-priori, we have $\widetilde{\Theta} \sim \mathrm{N}_{\tilde{\mu}_0, \widetilde{\Sigma}_0}$, where $\tilde{\mu}_0 := 0\iota_{2M+1}$ and where $\widetilde{\Sigma}_0 := \mathrm{diag}(100\iota_{2M+1})$. The parameters in $\widetilde{\Theta}$ can then be integrated out and need not be sampled in the MCMC algorithm.

5.4.3 Algorithm Details

Particle Weight Updates. In this subsection, we derive the incremental importance weights and the BS/AS weights for the Lévy-driven stochastic volatility model.

To simplify the notation, we write the mth integrated component volatility process and mth integrated compensated driving compound Poisson process as

$$
\begin{aligned}
V_p^{m,\star} &:= V_{(\tilde{t}_{p-1},\tilde{t}_p]}^{m,\star},\\
\overline{L}_p^{m,\star} &:= \overline{L}_{(\tilde{t}_{p-1},\tilde{t}_p]}^{m,\star}.
\end{aligned}
$$

Furthermore, letting A^{T} denotes the transpose of some matrix A, we write

$$
Z_p := \bigl(\tilde{t}_p - \tilde{t}_{p-1}, V_p^{1,\star}, \dots, V_p^{M,\star}, \overline{L}_p^{1,\star}, \dots, \overline{L}_p^{M,\star}\bigr)^{\mathrm{T}},
$$

Recall that $\widetilde{\Theta} = (\mu, \beta^{1:M}, \rho^{1:M})$ are those static parameters that we want to integrate out while Θ denotes the remaining static parameters. Let i_r be the index such that $t_{i_r} = \tilde{t}_r$, and let $\widetilde{\varpi}(\mathrm{d}\tilde{\theta})$ denote the marginal prior distribution of $\widetilde{\Theta}$. Letting

$$
\begin{aligned}
\mu_p(\tilde{\theta}) &:= Z_p^{\mathrm{T}}\tilde{\theta},\\
\Sigma_p &:= \sum_{m=1}^{M} V_p^{m,\star},
\end{aligned}
$$

denote the mean and variance of the pth observation (conditional on the volatility processes and conditional on the static parameters $(\Theta, \widetilde{\Theta})$, the incremental particle weights at the i_rth SMC step are then given by

$$
\begin{aligned}
&g^{\theta}(y_{q:r} | \dot{\psi}_{1:i_r}^{1:M}, y_{1:q-1})\\
&\quad= \int_{\widetilde{\Theta}} \Biggl[\prod_{p=q}^{r} \mathrm{N}_{\mu_p(\tilde{\theta}),\Sigma_p}(y_p)\Biggr] \mathrm{N}_{\tilde{\mu}_{q-1},\widetilde{\Sigma}_{q-1}}(\mathrm{d}\tilde{\theta})\\
&\quad\propto \Biggl[\frac{\det(\widetilde{\Sigma}_{q-1})}{\det(\widetilde{\Sigma}_r)} \prod_{p=q}^{r} \Sigma_p\Biggr]^{-1/2}\\
&\qquad\times \exp\Biggl(-\frac{1}{2}\Biggl[\sum_{p=q}^{r} \frac{y_p^2}{\Sigma_p} - \tilde{\mu}_r^{\mathrm{T}} \widetilde{\Sigma}_r^{-1} \tilde{\mu}_r + \tilde{\mu}_{q-1}^{\mathrm{T}} \widetilde{\Sigma}_{q-1}^{-1} \tilde{\mu}_{q-1}\Biggr]\Biggr).
\end{aligned}
$$

Here, $\tilde{\mu}_0$ and $\widetilde{\Sigma}_0$ are mean and variance of the Gaussian prior on $\widetilde{\Theta}$. Furthermore, we define

$$
\widetilde{\Sigma}_r := \Biggl[\sum_{p=q}^{r} \frac{Z_p Z_p^{\mathrm{T}}}{\Sigma_p} + \widetilde{\Sigma}_{q-1}^{-1}\Biggr]^{-1} = \Biggl[\sum_{p=1}^{r} \frac{Z_p Z_p^{\mathrm{T}}}{\Sigma_p} + \widetilde{\Sigma}_0^{-1}\Biggr]^{-1},
$$

and

$$\begin{aligned}\tilde{\mu}_r &:= \widetilde{\Sigma}_r\biggl[\sum_{p=q}^{r}\frac{Z_p y_p}{\Sigma_p} + \widetilde{\Sigma}_{q-1}^{-1}\tilde{\mu}_{q-1}\biggr]\\ &= \widetilde{\Sigma}_r\biggl[\sum_{p=1}^{r}\frac{Z_p y_p}{\Sigma_p} + \widetilde{\Sigma}_{0}^{-1}\tilde{\mu}_{0}\biggr].\end{aligned}$$

Backward and Ancestor Sampling Weights. Using the notation from above, the BS/AS weights at the i_rth SMC step are given by the product of the above-mentioned particle weight at Step i_r multiplied by

$$\begin{aligned}&\frac{g^{\theta}(y_{1:P}|\dot{\psi}_{1:I}^{1:M})}{g^{\theta}(y_{1:r}|\dot{\psi}_{1:i_r}^{1:M})}\\ &\quad\propto \biggl[\frac{\det(\widetilde{\Sigma}_r)}{\det(\widetilde{\Sigma}_P)}\prod_{p=r+1}^{l_r}\Sigma_p\biggr]^{-1/2}\\ &\qquad\times \exp\biggl(-\frac{1}{2}\biggl[\sum_{p=r+1}^{l_r}\frac{y_p^2}{\Sigma_p} - \tilde{\mu}_P^{\mathrm{T}}\widetilde{\Sigma}_P^{-1}\tilde{\mu}_P + \tilde{\mu}_r^{\mathrm{T}}\widetilde{\Sigma}_r^{-1}\tilde{\mu}_r\biggr]\biggr),\end{aligned}$$

where

$$\begin{aligned}l_r^m &:= \min\bigl\{\, p \in \mathbb{N}_P \bigm| \dot{\psi}_{i_r+1:I}^{m} \text{ has a jump in the interval } (\tilde{t}_r, \tilde{t}_p] \,\bigr\},\\ l_r &:= \max\bigl\{\, l_r^m \bigm| m \in \mathbb{N}_M \,\bigr\}.\end{aligned}$$

We note that while $\tilde{\mu}_P$ and $\widetilde{\Sigma}_P$ depend on the entire 'future' of the latent volatility processes, i.e. on $\dot{\psi}_{i_r+1:I}^{m}$, $V_p^{m,\star}$ and $\overline{L}_p^{m,\star}$ are identical for all particles if $p > l_r^m + 1$ and so the latter quantities need only be calculated once per MCMC iteration. To reduce the computation time, we only perform AS at the ith SMC step if the algorithm also resamples at that step. Recall that such adaptive BS or AS schemes are justified via an appropriate choice of the function ϱ_t in Section 3.4.

5.4.4 Simulation Study

One-Component Model. In this section, we first present a simulation study for a one-component Lévy-driven stochastic volatility model (i.e.

with $M = 1$) based on 500 synthetic observations. The parameters used for generating the data were $\mu = 0$, $\beta = 0.05$, $\rho = -1/2$, $\xi = 1$, and $\zeta = 0.5$. The initial values for the static parameters (except for those that are integrated out) were sampled from the prior distribution. For each algorithm, the initial points of the PPP were generated by an SMC algorithm.

Both algorithms used 100 Gaussian random-walk MH updates per iteration with standard deviations $(0.1, 0.25, 0.05)$ for the (log-transformed) components $(\log \kappa, \log \xi, \log \zeta^{-1})$. The NCP was only used with probability 0.5. Otherwise, the CP was used. Such strategies for 'interweaving' NCPs and CPs have been studied in Yu and Meng (2011).

The algorithms differ only in the way that the points of the PPP were updated in each MCMC iteration. The first algorithm – the non-centred PG sampler – used a CSMC kernel with $N = 25$, $N = 50$ and $N = 100$ particles and ancestor sampling for this task. The particles where proposed from the prior; this type of 'bootstrap particle filter' was termed variable-rate particle filter in Godsill and Vermaak (2004). Resampling took place whenever the effective sample size dropped below $N/2$. The SMC step size was set to $t_i - t_{i-1} = 5$. On average, a single CSMC run took around 2 seconds.

The second algorithm updated the points of the PPP using 400 RJMCMC moves per iteration. These can be (a) an update of the initial value V_0, (b) a birth move, adding a new jump uniformly at random in T and sampling the associated jump size from the prior, (c) a death move randomly deleting a jump, (d) a move for adjusting a randomly chosen jump time by sampling a new location uniformly at random between the previous and the next jump time, (e) a move for adjusting a randomly chosen jump size by sampling a new value from the prior. The probabilities for selecting one of the Moves a to e were $(0.05, 0.25, 0.25, 0.225, 0.225)$ if the PPP contained at least one jump and $(0.05, 0.95, 0, 0, 0)$, otherwise. On average, 400 such RJMCMC moves took around 2 seconds.

The algorithms were implemented in Matlab (The MathWorks, Inc., 2015) and run on a single core of an Intel®Core™i7-5820 CPU. Performance of the SMC algorithm in Matlab is rather poor compared to SMC algorithms for state-space models, for instance. This is because at each iteration, different particles comprise different numbers of jumps which hinders vectorisation. Alternative SMC algorithms, such as those developed in Whiteley et al. (2011), which were also improved in Chapter 4 of this work, would be more amenable to vectorisation and parallelisation (at the cost of introducing a bias).

Marginal kernel density estimates and sample autocorrelations based on 100,000 iterations (of which the first 5,000 have been discarded as burn-in) are displayed in Figures 5.1 and 5.2, respectively. Both algorithms yield comparable results. The PG sampler also exhibits a slightly lower sample autocorrelation.

Two-Component Model. We conclude this section by applying the non-centred PG sampler to a two-component Lévy-driven stochastic volatility model and again compare its performance to that of a RJMCMC-based algorithm.

Both algorithms are designed as in the one-component model except that the SMC algorithm now uses the step size $t_i - t_{i-1} = 2$ and employs $N = 50$ particles while the RJMCMC-based scheme attempts 500 updates of the PPP per iteration. Furthermore, both algorithms now attempt 200 static-parameter updates per iteration. These are again Gaussian random-walk MH kernels each updating one of the blocks $\log \kappa^{1:2}$, w^1 or $(\log \xi, \log \zeta^{-1})$. The MH proposal kernels are parametrised through the diagonal covariance matrices $\operatorname{diag}([0.25, 1])$, 0.01, and $\operatorname{diag}([0.25, 0.25])$. Here $\operatorname{diag}(v)$ denotes a diagonal matrix which has a vector v as its diagonal.

We study two different settings. In both, we assume that the risk premium $\beta := \beta^1 = \beta^2$ is the same for both component processes. In the first setting, we additionally assume that the leverage parameters $\rho^{1:2}$ are known (and equal to 0). Studying models with such a slightly reduced number of parameters was motivated by the fact that we observed poor mixing of both algorithms unless extremely large number of observations (around 2000) were used.

For shorter time series, this poor mixing seems to be due to the fact that some of the static parameters are only weakly identifiable. In particular, as shown in Figure 5.3, the marginal posterior distribution of the static parameters is bimodal and these modes correspond to the component weights $w^{1:2} = (1, 0)$ or $w^{1:2} = (0, 1)$. This leads to the particularly high autocorrelation of the MCMC chain in this component as shown in Figure 5.4.

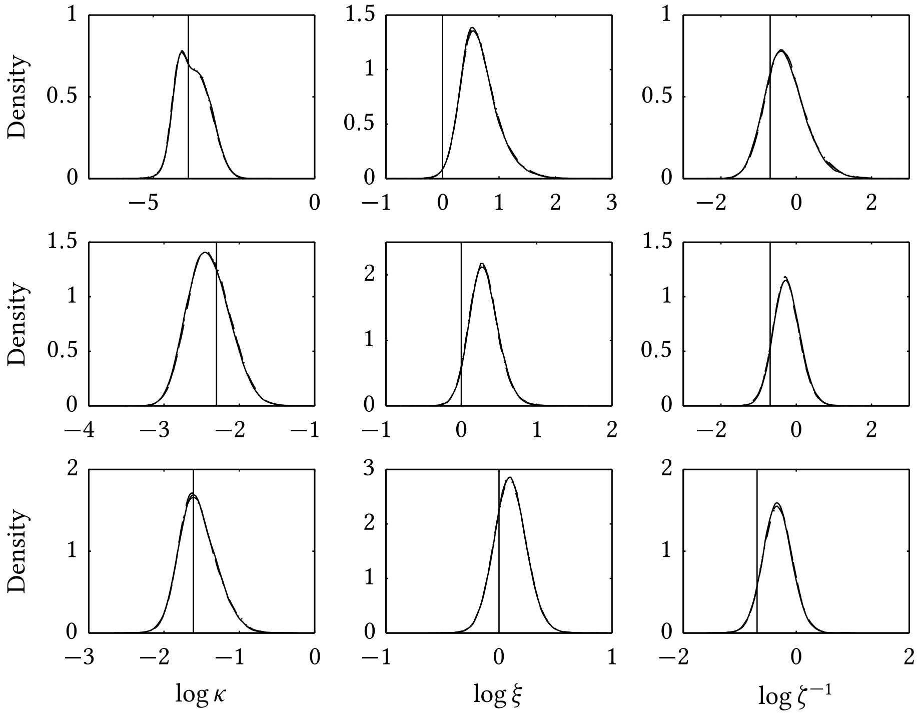

Figure 5.1 Estimated marginal posterior densities of the static parameters in the one-component Lévy-driven stochastic volatility model. Based on 500 observations taken every $\tilde{t}_p - \tilde{t}_{p-1} = 1$ time units and generated using true parameter values $(\xi, \zeta, \mu, \beta, \rho) = (1, 0.5, 0, 0.05, -0.5)$. *Top row:* $\kappa = 0.02$. *Middle row:* $\kappa = 0.1$. *Bottom row:* $\kappa = 0.2$. The estimates are obtained from 100,000 iterations of the non-centred PG sampler with 25 particles (dotted line), 50 particles (dash-dotted line), and 100 particles (solid line), as well as from a non-centred RJMCMC algorithm with 400 attempted updates of the latent process in between static-parameter updates (dashed line). Vertical lines represent the true parameter values.

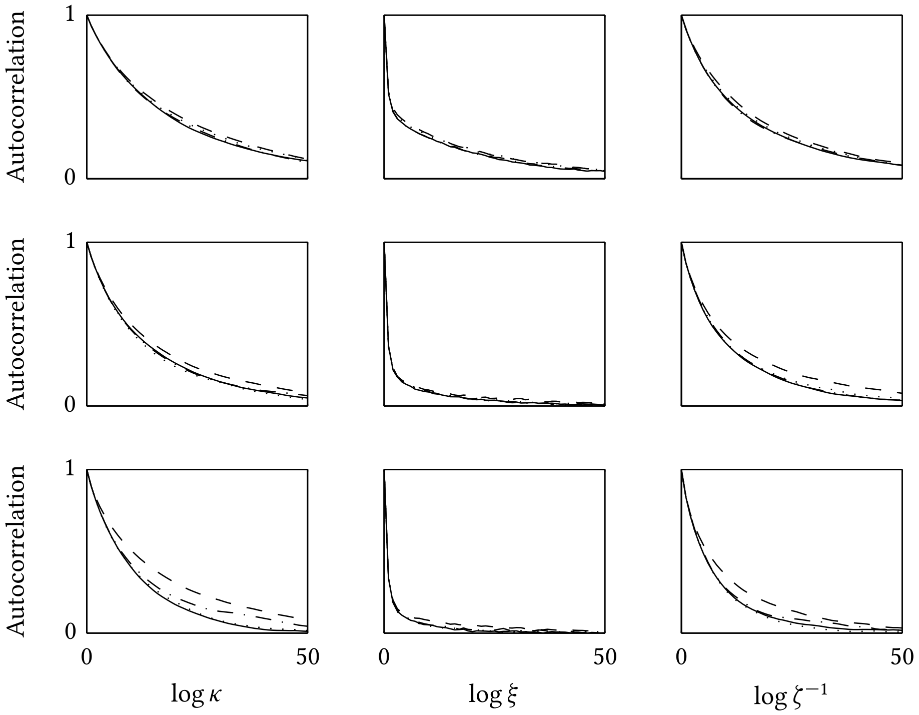

Figure 5.2 Autocorrelation of the static-parameter estimates in the one-component Lévy-driven stochastic volatility model. Based on 500 observations taken every $\tilde{t}_p - \tilde{t}_{p-1} = 1$ time units and generated using true parameter values $(\xi, \zeta, \mu, \beta, \rho) = (1, 0.5, 0, 0.05, -0.5)$. *Top row:* $\kappa = 0.02$. *Middle row:* $\kappa = 0.1$. *Bottom row:* $\kappa = 0.2$. The estimates are obtained from 100,000 iterations of the non-centred PG sampler with 25 particles (dotted line), 50 particles (dash-dotted line), and 100 particles (solid line), as well as from a non-centred RJMCMC algorithm with 400 attempted updates of the latent process in between static-parameter updates (dashed line).

Since $\lambda^m = w^m \kappa^m / \sum_{n=1}^{M} \lambda^n / \kappa^n$ these modes essentially imply no jumps in the second or first component process, respectively. This is a particular concern in the context of PG samplers because having fewer jumps increases the cost of computing the AS weights (as these then depend on the behaviour of the process much further into the future). To reduce the computational cost, we switch to RJMCMC updates whenever the total number of jumps in one of the component processes falls below 5. Nonetheless, we stress that these identifiability issues are not an artefact of our algorithm. Rather, they appear to be inherent in this class of models and, to our knowledge, they have not yet been pointed out in the literature.

5.5 Summary

Performance. In this chapter, we have combined particle Gibbs samplers with sophisticated NCPs to perform static-parameter estimation in statistical models based around compound Poisson processes. We have also applied our algorithm to a Lévy-driven stochastic volatility model. Somewhat surprisingly, in this model, the PG sampler does not appear significantly more efficient than a large number of single-site updates as part of a more conventional MCMC kernel composed of RJMCMC updates. In addition, these models appears to be overparametrised unless a particularly large number of observations is available. To our knowledge, this has not been pointed out in the literature. A more formal analysis of these identifiability issues is left for future research.

Extensions. The approach presented in this chapter can be extended to more general settings. As noted in Roberts et al. (2004), an extension to a time-dependent rate parameter $\lambda_t = l(t, \theta)$ is straightforward but notationally cumbersome. The extension to non-continuous mark distributions on ν^θ is also straightforward as long as the generalised inverse of the CDF can be evaluated point-wise. For d-dimensional mark distributions the reparametrisation proceeds as above but takes $\bar{F}^\theta$ to be the inverse CDF of a one-dimensional marginal distribution of ν^θ. The remaining mark components can then be generated one-at-a-time by inversion sampling from the inverse CDF of the relevant conditional distribution. This is sometimes called the *Rosenblatt transformation.*

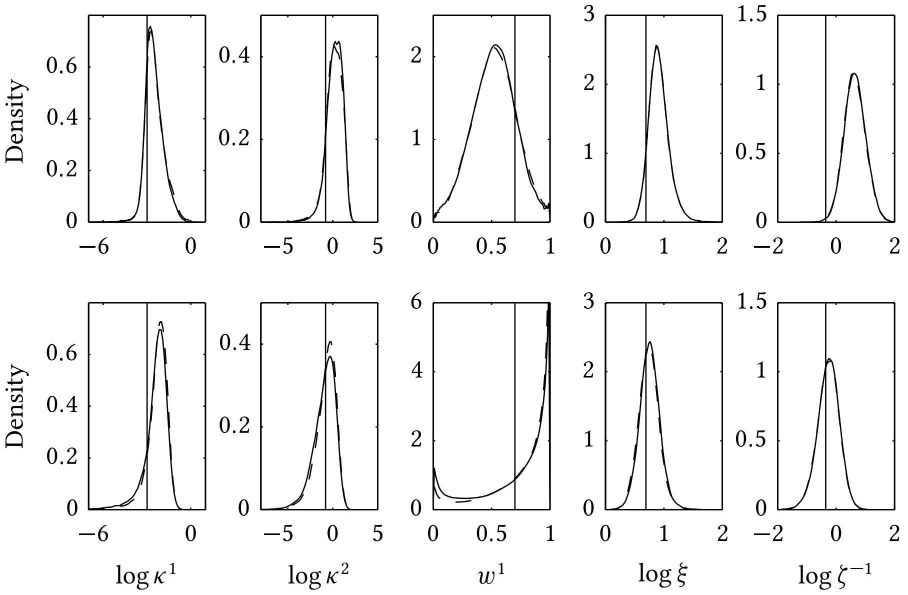

Figure 5.3 Estimated marginal posterior densities of the static parameters in two-component Lévy-driven stochastic volatility models. Based on 500 observations taken every $\tilde{t}_p - \tilde{t}_{p-1} = 1$ time units and generated using true parameter values $(\kappa^{1:2}, w^{1:2}, \xi, \zeta, \mu) = (0.05, 0.5, 0.7, 0.3, 2, 0.7, 0)$. *Top row:* simplified model with $\beta := \beta^1 = \beta^2$ and assuming that $\rho^1 = \rho^2 = 0$ is known. *Bottom row:* simplified model with $\beta := \beta^1 = \beta^2$ but now with the true parameters $\rho^{1:2} = (-0.5, -0.5)$ assumed to be unknown. The estimates are obtained from 100,000 iterations of the non-centred PG sampler with 50 particles (solid line) and from a non-centred RJMCMC algorithm with 500 attempted updates of the latent process in between static-parameter updates (dashed line). Vertical lines represent the true parameter values.

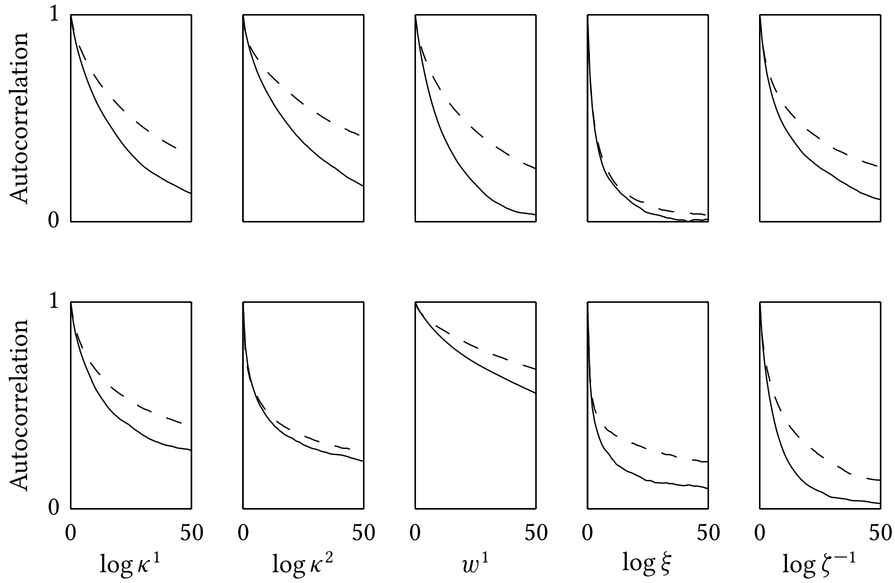

Figure 5.4 Autocorrelation of the static-parameter estimates in two-component Lévy-driven stochastic volatility models. Based on 500 observations taken every $\tilde{t}_p - \tilde{t}_{p-1} = 1$ time units and generated using true parameter values $(\kappa^{1:2}, w^{1:2}, \xi, \zeta, \mu) = (0.05, 0.5, 0.7, 0.3, 2, 0.7, 0)$. *Top row:* simplified model with $\beta := \beta^1 = \beta^2$ and assuming that $\rho^1 = \rho^2 = 0$ is known. *Bottom row:* simplified model with $\beta := \beta^1 = \beta^2$ but now with the true parameters $\rho^{1:2} = (-0.5, -0.5)$ assumed to be unknown. The estimates are obtained from 100,000 iterations of the non-centred PG sampler with 50 particles (solid line) and from a non-centred RJMCMC algorithm with 500 attempted updates of the latent process in between static-parameter updates (dashed line).

6 Pseudo-Marginal Monte Carlo Optimisation

6.1 Introduction

6.1.1 Motivation

In this chapter, we perform optimisation such as (marginal) maximum likelihood or maximum a-posteriori estimation in latent variable models. To that end, we present a flexible framework for combining the basic idea of the state augmentation for marginal estimation algorithm from Doucet, Godsill and Robert (2002), outlined in Section 6.2, with state-of-the-art Markov chain Monte Carlo kernels, such as pseudo-marginal Metropolis–Hastings (Subsection 6.3.2) or particle Gibbs kernels (Subsection 6.3.1). These ideas are also incorporated into population-based approaches in the form of sequential Monte Carlo samplers, as explained in Subsection 6.3.3. Finally, in Section 6.4, we demonstrate the effectiveness of our methods by performing maximum likelihood estimation in a number of challenging models.

Let Θ be some non-empty set, e.g. some subset of $\mathbb{R}^d$. In this chapter, we assume that we want to maximise some function

$$h\colon \Theta \to \mathbb{R},$$

i.e. we assume that we want to find the set of (equivalent) global maxima

$$\Theta_h := \left\{ \theta' \in \Theta \;\middle|\; \forall\, \theta \in \Theta : h(\theta') \geq h(\theta) \right\},$$

or at least a subset thereof, assuming that Θ_h is well defined and in particular, assuming that the cardinality of Θ_h, $\#\Theta_h$, is finite.

As usual, minimising some function $\tilde{h}$ can be treated as a maximisation problem by considering the function

$$h := -\tilde{h},$$

instead.

6.1 Example (ML/MAP estimation). *Given some data, let $L(\theta)$ denote the (marginal) likelihood of a parameter $\theta \in \Theta$ under some statistical model. In the Bayesian paradigm, let $\varpi \in \mathcal{M}_1(\Theta)$ be a prior distribution with density $\tilde{p}$ with respect to a suitable dominating measure, μ. Thus,*

(1) if $h(\theta) := L(\theta)$, then Θ_h is the maximum likelihood *(ML) estimate;*

(2) if $h(\theta) := L(\theta)\tilde{p}(\theta)$, then Θ_h is the (marginal) maximum a-posteriori *(MAP) estimate under the chosen parametrisation, determined by the decomposition of the prior distribution into the pair $(\mu, \tilde{p})$.*

Numerical Optimisation. For a sufficiently complicated objective function h, Θ_h cannot be found analytically and is usually approximated numerically. Unfortunately, numerical methods are not always reliable. They require strong regularity conditions on h and Θ which are often difficult to verify. In practice, they are prone to get stuck in local maxima.

Monte Carlo Optimisation. For complicated optimisation problems – in particular, in the presence of multi-modality – approaches based around the Monte Carlo method can be more robust than (gradient-based) numerical methods. This robustness often justifies the usually higher computational cost associated with the former.

Let Z_p be a potentially large collection of auxiliary variables taking values in Z_p. The idea of Monte Carlo optimisation is to define a sequence of finite measures $(\gamma_p)_{p \in \mathbb{N}}$, with $\gamma_p \in \mathcal{M}(\Theta \times \mathsf{Z}_p)$ such that the relevant marginal of γ_p becomes increasingly concentrated around the points in Θ_h. That is, for any $A \in \mathcal{B}(\Theta)$, the probability measures $\pi_p \propto \gamma_p$ satisfy

$$\pi_p(A \times \mathsf{Z}_p) \overset{p \to \infty}{\longrightarrow} \sum_{\theta \in \Theta_h} w_h(\theta) \delta_\theta(A), \tag{6.1}$$

where $w_h \colon \Theta_h \to (0, 1]$ is a weighting function with $\sum_{\theta \in \Theta_h} w_h(\theta) = 1$.

We then use Monte Carlo algorithms to sample approximately from π_p. For sufficiently small p, the distribution π_p should be dispersed enough to allow *Markov chain Monte Carlo* (MCMC) algorithms or *sequential Monte Carlo* (SMC) samplers to traverse the state space and explore different modes. For sufficiently large p, the samples should be concentrated around one point in Θ_h when using MCMC methods or around (some of) the points in Θ_h when using SMC or other population-based methods.

6.1.2 Contribution

Throughout this chapter, we consider different instances of the generic measures γ_p all of which are chosen to concentrate around Θ_h as $p \to \infty$. These measures are generically represented as

$$\gamma_p(\mathrm{d}\theta \times \mathrm{d}z_p) := \mu(\mathrm{d}\theta) M_p(\theta, \mathrm{d}z_p) H_p(\theta, z_p). \tag{6.2}$$

Here, $\mu \in \mathcal{M}_\sigma(\Theta)$ is some σ-finite measure on Θ which does not vanish in a neighbourhood of the points in Θ_h. Furthermore,

$$M_p \in \mathcal{K}_\sigma(\Theta, \mathsf{Z}_p)$$

is a suitable σ-finite kernel and

$$H_p \colon \Theta \times \mathsf{Z}_p \to [0, \infty)$$

is a suitable non-negative measurable function. All of these quantities will be specified such that Equation 6.1 holds.

Furthermore, we define an *annealing schedule* $(\beta_p)_{p \in \mathbb{N}}$. The values $\beta_p \in [0, \infty)$ are commonly interpreted as *inverse temperatures*. They are chosen to satisfy $\beta_p \leq \beta_{p+1}$, for $p \in \mathbb{N}$ and $\beta_p \to \infty$, as $p \to \infty$. This annealing schedule will be used to parametrise M_p and H_p in order to specify the speed at which the measures γ_p concentrate around Θ_h.

The remainder of this work discusses various ways of selecting suitable components M_p and H_p and devising efficient SMC algorithms to approximate the measures $(\gamma_p)_{p \in \mathbb{N}}$ and MCMC algorithms to (approximately) sample from their self-normalised versions $(\pi_p)_{p \in \mathbb{N}}$. A particular focus is on situations in which (suitable maximum-preserving transformations of) h cannot be evaluated point-wise. Our contributions are as follows.

(1) We construct a generic framework for pseudo-marginal MCMC-based optimisation algorithms. We also combine existing optimisation schemes with modern, sophisticated multiple-proposal kernels such as (iterated) *conditional sequential Monte Carlo* (CSMC) kernels.

(2) We incorporate both ideas into population-based approaches known as SMC samplers.

(3) We verify these methods on a number of tractable toy examples and further demonstrate their effectiveness on challenging problems.

6.2 Background

6.2.1 Simulated Annealing

Assume that h can be evaluated point-wise. In classical *simulated annealing* (SA) (Kirkpatrick, Gelatt & Vecchi, 1983), the target measures $\gamma_p^{\mathrm{SA}} \in \mathcal{M}(\Theta \times \mathsf{Z}_p)$ are defined according to Equation 6.2 with $\mathsf{Z}_p \equiv \emptyset$, $H_p(\theta) = \exp(\beta_p h(\theta))$. That is, the target measures of SA are defined by

$$\gamma_p^{\mathrm{SA}}(\mathrm{d}\theta) := \exp(\beta_p h(\theta))\mu(\mathrm{d}\theta).$$

In the particular case that $\mu = \mathrm{Leb}^{\otimes d}$ is the Lebesgue measure on $\Theta = \mathbb{R}^d$, h is three times continuously differentiable, and under further technical conditions on Θ_h, Hwang (1980) showed that this family of distributions concentrates around Θ_h as formalised in Equation 6.1. The idea is then to approximate the distributions $(\pi_p^{\mathrm{SA}})_{p\in\mathbb{N}}$, where $\pi_p^{\mathrm{SA}} \propto \gamma_p^{\mathrm{SA}}$, using an inhomogeneous MCMC algorithm, i.e. using an MCMC chain whose target distribution changes over iterations. Under strong regularity conditions, converge of SA was established in Hajek (1988), Winkler (2003), Andrieu, Breyer and Doucet (2001).

Finally, Rubenthaler, Rydén and Wiktorsson (2009) studied improvements upon the annealing schedule and target measures in SA.

6.2.2 State Augmentation for Marginal Estimation

Motivation. Algorithms such as SA require point-wise evaluations of (maximum-preserving transformations of) h. The rest of this work is concerned with situations in which such evaluations are impossible (to do efficiently) but in which we can find some space Z_p, some finite kernel M_p, and some measurable function H_p (which we *can* evaluate) such that the measures γ_p satisfy Equation 6.1.

More specifically, we assume that there exists a measurable function $H\colon \Theta \times \mathsf{X} \to [0,\infty)$ and a stochastic kernel $M \in \mathcal{K}_1(\Theta \times \mathsf{X})$ such that

$$\gamma(\theta, \mathrm{d}x) := H(\theta, x)M(\theta, \mathrm{d}x)$$

defines a finite kernel and such that – recalling that $\mathbb{1}$ is the unit function on an appropriate domain – the maxima of the integral

$$\gamma(\theta, \mathbb{1}) := \int_{\mathsf{X}} \gamma(\theta, \mathrm{d}x), \tag{6.3}$$

coincide with the maxima of h, i.e.

$$\Theta_h = \left\{ \theta' \in \Theta \,\middle|\, \forall\, \theta \in \Theta : \gamma(\theta', \mathbb{1}) \geq \gamma(\theta, \mathbb{1}) \right\}. \tag{6.4}$$

For later use, we define the stochastic kernel $\mathit{\Pi}_\beta$ by

$$\mathit{\Pi}_\beta(\theta, \mathrm{d}x) := \frac{H(\theta, x)^\beta M(\theta, \mathrm{d}x)}{\int_{\mathsf{X}} H(\theta, x)^\beta M(\theta, \mathrm{d}x)}. \tag{6.5}$$

6.2 Example (ML/MAP estimation, continued). *Many statistical models are specified through an additional set of latent variables X taking values in some set X. Under such a model, conditional on some $\theta \in \Theta$, X is distributed according to $M(\theta, \cdot)$. Let $G(\theta, x)$ denote the completed likelihood, i.e. the likelihood of some observed data given the parameters and latent variables, (θ, x). Unfortunately, the marginal likelihood*

$$L(\theta) = \int_{\mathsf{X}} G(\theta, x) M(\theta, \mathrm{d}x)$$

is often intractable. To still perform optimisation in this setting, we can take

(1) $H(\theta, x) := G(\theta, x)$ for the purpose of marginal ML estimation,

(2) $H(\theta, x) := G(\theta, x)\tilde{p}(\theta)$ for the purpose of marginal MAP estimation.

A MCMC algorithm for performing optimisation in such problems called *state augmentation for marginal estimation* (SAME) was introduced in an MCMC context by Doucet et al. (2002) (see also Gaetan & Yao, 2003; Jacquier, Johannes & Polson, 2007). Here, we describe the slightly more general construction developed by Johansen et al. (2008) which allows for non-integer inverse temperatures β_p. Again, $(\beta_p)_{p \in \mathbb{N}}$ is a sequence in $[0, \infty)$ such that $\beta_p \uparrow \infty$.

Extended Target Measure. At the pth iteration, the SAME algorithm augments the space with $\lceil \beta_p \rceil$ replicas of the random variable $X = X_p$, i.e. $Z_p := X^{1:\lceil \beta_p \rceil}$. The extended target distributions $\pi_p^{\text{SAME}} \propto \gamma_p^{\text{SAME}}$ on $\Theta \times \mathsf{Z}_p$ with $\mathsf{Z}_p := \mathsf{X}^{\lceil \beta_p \rceil}$, are then defined by letting γ_p^{SAME} be given by Equation 6.2 with

$$M_p(\theta, \mathrm{d}z_p) := \prod_{i=1}^{\lceil \beta_p \rceil} M(\theta, \mathrm{d}x_p^i),$$

and with

$$H_p(\theta, z_p) := H(\theta, x_p^{\lceil \beta_p \rceil})^{\beta_p^\sharp} \prod_{i=1}^{\lfloor \beta_p \rfloor} H(\theta, x_p^i),$$

where $\beta^\sharp := \beta - \lfloor \beta \rfloor$. Whenever $\beta_p \in \mathbb{N}$,

$$\int_{A \times \mathsf{Z}_p} \gamma_p^{\textsc{same}}(\mathrm{d}\theta \times \mathrm{d}z_p) = \int_A \gamma(\theta, \mathbb{1})^{\beta_p} \mu(\mathrm{d}\theta),$$

so that under suitable regularity conditions, by Equation 6.4, the θ-marginal of $\pi_p^{\textsc{same}}$ concentrates around Θ_h as $p \to \infty$.

Algorithm. The SAME algorithm is usually implemented as an inhomogeneous Gibbs sampler or inhomogeneous Metropolis-within-Gibbs algorithm as summarised in Algorithm 6.3. Therein, Q_p denotes some MCMC kernel which is invariant with respect to the full conditional distribution of Θ under $\pi_p^{\textsc{same}}$ and R_p is an MCMC kernel targeting

$$\begin{cases} \Pi_1^{\otimes \lceil \beta_p \rceil}(\theta, \cdot), & \text{if } \beta_p \in \mathbb{N}, \\ \Pi_1^{\otimes \lfloor \beta_p \rfloor} \otimes \Pi_{\beta_p^\sharp}(\theta, \cdot), & \text{otherwise,} \end{cases}$$

where Π_β was defined in Equation 6.5. Furthermore,

$$S_p \in \mathcal{K}_1\bigl(\Theta \times \mathsf{X}^{\lceil \beta_{p-1} \rceil}, \mathsf{X}^{\lceil \beta_p \rceil - \lceil \beta_{p-1} \rceil}\bigr)$$

is some suitable proposal kernel for augmenting the space with additional latent variables $X_{p-1}^{\lceil \beta_{p-1} \rceil + 1 : \lceil \beta_p \rceil}$, whenever $\lceil \beta_p \rceil > \lceil \beta_{p-1} \rceil$. To simplify the notation, we write the thus augmented vector of latent variables as

$$\widetilde{Z}_{p-1} := X_{p-1}^{1:\lceil \beta_p \rceil} = \bigl(Z_{p-1}, X_{p-1}^{\lceil \beta_{p-1} \rceil + 1 : \lceil \beta_p \rceil}\bigr).$$

This random vector takes values in the set $\widetilde{\mathsf{Z}}_{p-1} := \mathsf{X}^{\lceil \beta_p \rceil}$.

6.3 Algorithm (SAME). *At the nth iteration,*

(1) if $\lceil \beta_n \rceil > \lceil \beta_{n-1} \rceil$, sample $X_{n-1}^{\lceil \beta_{n-1} \rceil + 1 : \lceil \beta_n \rceil} \sim S_n((\theta_{n-1}, z_{n-1}), \cdot)$,

(2) sample $Z_n = X_n^{1:\lceil \beta_n \rceil} \sim R_n((\theta_{n-1}, \tilde{z}_{n-1}), \cdot)$,

(3) sample $\Theta_n \sim Q_n((\theta_{n-1}, z_n), \cdot)$.

Potential Drawbacks. Usually, the SAME algorithm is implemented as an inhomogeneous Gibbs sampler or Metropolis-within-Gibbs algorithm. This can lead to poor mixing of the MCMC chain for two reasons.

(1) It is often impossible to update $X_p^{1:\lceil \beta_p \rceil}$ using a Gibbs step and devising another efficient form for the MCMC kernel R_p can be difficult, especially if X is high-dimensional.

(2) If Θ and Z_p are highly correlated under π_p^{SAME}, then component-wise updates are known to be inefficient.

To alleviate the first potential drawback, we propose to combine the SAME approach with sophisticated multiple-proposal MCMC kernels such as (iterated) CSMC kernels (Andrieu et al., 2010; Whiteley, 2010). This is described in the next section. To alleviate the second potential drawback, we also present a sequence of further extended distributions. This permits the combination of the SAME approach with pseudo-marginal MCMC kernels (Beaumont, 2003; Andrieu & Roberts, 2009).

6.2.3 Optimisation Using SMC Samplers

Even if the objective function h (and also $\gamma(\,\cdot\,, \mathbb{1})$) is unimodal, the introduction of the latent variables $X_p^{1:\lceil \beta_p \rceil}$ often leads to an extended measure γ_p^{SAME} with varying degrees of multimodality. This can hamper mixing of the inhomogeneous MCMC algorithms targeting the distributions π_p^{SAME}. Furthermore, if Θ_h contains multiple maxima, a single SAME chain will eventually become trapped around one of them.

To improve robustness, Johansen et al. (2008) devise a population-based version of the SAME idea. More precisely, they incorporate the SAME approach into the SMC-sampler framework from Del Moral et al. (2006b, 2007), Peters (2005) which is summarised in Subsection 2.3.2. Such SMC samplers have already been successfully employed to realistic problems in air-traffic management (Kantas, Maciejowski & Lecchini-Visintini, 2009, 2010).

At the tth step, the SMC sampler uses (forward) proposal kernels which move the particles according to an MCMC kernel $\widetilde{P}_t$, given by

$$\begin{aligned}&\widetilde{P}_t((\theta_{t-1}, \tilde{z}_{t-1}), \mathrm{d}\theta_t \times \mathrm{d}z_t)\\ &\qquad := R_t((\theta_{t-1}, \tilde{z}_{t-1}), \mathrm{d}z_t) Q_t((\theta_{t-1}, z_t), \mathrm{d}\theta_t).\end{aligned}$$

This MCMC kernel is applied after having extended each particle via $S_t \in \mathcal{K}_1(\Theta \times \mathsf{Z}_{t-1}, \mathsf{X}^{\lceil \beta_t \rceil - \lceil \beta_{t-1} \rceil})$, in the case that $\lceil \beta_t \rceil > \lceil \beta_{t-1} \rceil$. In summary, recalling that $\widetilde{Z}_{p-1} = X_{p-1}^{1:\lceil \beta_p \rceil}$, the Step-$t$ proposal kernel is

$$\begin{aligned}
&P_t((\theta_{t-1}, z_{t-1}), \mathrm{d}x_{t-1}^{\lceil \beta_{t-1} \rceil + 1:\lceil \beta_t \rceil} \times \mathrm{d}\theta_t \times \mathrm{d}z_t) \\
&\quad := S_t(\theta_{t-1}, z_{t-1}), \mathrm{d}x_{t-1}^{\lceil \beta_{t-1} \rceil + 1:\lceil \beta_t \rceil}) \\
&\qquad \times \widetilde{P}_t((\theta_{t-1}, \tilde{z}_{t-1}), \mathrm{d}\theta_t \times \mathrm{d}z_t).
\end{aligned}$$

As stressed in Johansen et al. (2008), this is not the most general SMC implementation of the SAME idea; letting $\widetilde{P}_t$ be an MCMC kernel is often convenient but not actually necessary.

The SMC sampler marginally targets the measure γ_t^{SAME} by targeting an extended measure constructed via backward Markov kernels,

$$\begin{aligned}
&L_{t-1}((\theta_t, z_t), \mathrm{d}\theta_{t-1} \times \mathrm{d}\tilde{z}_{t-1}) \\
&\quad := \frac{\widetilde{P}_t((\theta_{t-1}, \tilde{z}_{t-1}), \,\cdot\,)}{\mathrm{d}\gamma_t^{\text{SAME}}}(\theta_t, z_t)\gamma_t^{\text{SAME}}(\mathrm{d}\theta_{t-1} \times \mathrm{d}\tilde{z}_{t-1}).
\end{aligned}$$

In other words, this is the usual approximation to the optimal backward kernel described in Example 2.15. This backward kernel is commonly employed whenever MCMC kernels are used to move the particles within an SMC sampler.

With these backward kernels, and with $U_t := (\Theta_t, Z_t, X_{t-1}^{\lceil \beta_{t-1} \rceil + 1:\lceil \beta_t \rceil})$, the incremental importance weights at Step t are defined by

$$G_t(u_{1:t}) := \frac{\tilde{\gamma}_t}{\tilde{\gamma}_{t-1} \otimes S_t}(\theta_{t-1}, \tilde{z}_{t-1}).$$

This may also be derived by viewing the single step of the SMC sampler as two consecutive SMC steps. The first step then changes the extended target distribution (and potentially adds additional variables using the proposal kernel S_t), leading to the above-mentioned incremental weight. The second step simply applies the MCMC kernel $\widetilde{P}_t$ to the particles using time-reversal backward kernels. As noted in Example 1.9, this second step does not modify the weights.

6.3 Novel Methodology

6.3.1 Pseudo Gibbs Samplers

In this section, we present two approaches which can be interpreted as extensions of SAME. Both of these approaches employ an even larger number of replicas of the latent variable X than the standard SAME algorithm. Whilst this increases the number of random variables which need to be sampled at each iteration, these approaches are even more amenable to parallelisation than the standard SAME algorithm for whom this was already exploited in Zhao, Jiang and Canny (2014).

First, in this subsection, we assume that Gibbs sampling-type implementations of the SAME idea are desirable because correlation of Θ and Z_p under π_p^{SAME} is not too severe or because it can be circumvented using some reparametrisation, e.g. along the lines of the non-centred parametrisation from Papaspiliopoulos (2003), Papaspiliopoulos, Roberts and Sköld (2007).

Unfortunately, it is often impossible to sample directly from $\Pi_1(\theta, \cdot)$ (or $\Pi_{\beta_p^\sharp}(\theta, \cdot)$) and letting R_p be a concatenation of Metropolis-within-Gibbs steps can be inefficient, in particular if X is high-dimensional. Below, we present a scheme that mimics the intractable Gibbs step (Step 2 in Gibbs-sampling implementations of Algorithm 6.3) by employing multiple-proposal MCMC kernels. Particularly useful instances of such multiple-proposal kernels are given by the (iterated) CSMC kernels from (Andrieu et al., 2010, 2013) which were described in Section 3.4

Conditional SMC-based optimisation. We target the standard SAME target distribution but rather than sampling from $\Pi_1(\theta, \cdot)$ (or $\Pi_{\beta_p^\sharp}(\theta, \cdot)$), which is assumed to be infeasible, we apply some (multiple-proposal) instance of the generic MCMC kernel from Subsection 3.2.3 within Step 2 of Algorithm 6.3. Algorithm 6.5 summarises a single iteration of the resulting procedure if the specific instance of the generic MCMC kernel is one of the (iterated) CSMC kernels from Section 3.4, potentially with backward sampling or ancestor sampling. In this case, $X = X_p$ is to be understood as an entire particle trajectory.

6.4 Example (hidden Markov models). *Though applicable much more widely, (conditional) SMC methods are often used to propose values for* *the latent states in* (general state-space) hidden Markov models *(HMMs). Assume that we have observations up to some time $T \in \mathbb{N}$. In this case, $M(\theta, \cdot)$ represents the conditional prior distribution of the vector of latent states up to Time T, denoted X, given the static parameters Θ. Likewise, $G(\theta, x)$ is the likelihood of the observations given the static parameters and given the latent states.*

We set $R_p := R_{p,1:\lceil\beta_p\rceil}^{\otimes}$, where $R_{p,l}$ is the plain (iterated) CSMC kernel (induced by Algorithm 3.23), the (iterated) CSMC kernel with backward sampling (induced by Algorithm 3.24) or the (iterated) CSMC kernel with ancestor sampling (induced by Algorithm 3.26). In each case, this kernel targets the probability measure

$$\begin{cases} \Pi_1(\theta, \cdot), & \text{if } l \leq \lfloor\beta_p\rfloor, \\ \Pi_{\beta_p^\sharp}(\theta, \cdot), & \text{if } l = \lceil\beta_p\rceil > \lfloor\beta_p\rfloor. \end{cases}$$

In addition, the space is augmented with extra replicas of the latent variable by sampling from the kernel $S_p := S_{p,1:\lceil\beta_p\rceil-\lceil\beta_{p-1}\rceil}^{\otimes}$ whenever $\lceil\beta_p\rceil > \lceil\beta_{p-1}\rceil$. Here, $S_{p,k}(\theta, \cdot)$ denotes the marginal distribution of one particle trajectory under the extended target distribution

$$\bar{\pi}_T(\theta, \cdot) \propto \bar{\gamma}_T(\theta, \cdot)$$

associated with an SMC algorithm described in Chapter 2 (which may now depend on θ), assuming that the SMC algorithm targets a measure proportional to

$$\begin{cases} \Pi_1(\theta, \cdot), & \text{if } k \leq \lfloor\beta_p\rfloor - \lceil\beta_{p-1}\rceil, \\ \Pi_{\beta_p^\sharp}(\theta, \cdot), & \text{if } k = \lceil\beta_p\rceil - \lceil\beta_{p-1}\rceil \text{ and } \beta_p \notin \mathbb{N}. \end{cases}$$

6.5 Algorithm (CSMC-based SAME). *At the nth iteration,*

(1) if $\lceil\beta_n\rceil > \lceil\beta_{n-1}\rceil$, sample $X_{n-1}^{\lceil\beta_{n-1}\rceil+1:\lceil\beta_n\rceil} \sim S_n((\theta_{n-1}, z_{n-1}), \cdot)$,

(2) sample $Z_n = X_n^{1:\lceil\beta_n\rceil} \sim R_n((\theta_{n-1}, \tilde{z}_{n-1}), \cdot)$,

(3) sample $\Theta_n \sim Q_n((\theta_{n-1}, z_n), \cdot)$.

We stress that (iterated) CSMC kernels are only one possible instance of the generic MCMC kernel from Subsection 3.2.3. In many situations, it may be desirable to employ other instances of the generic MCMC kernel from Subsection 3.2.3 in Step 2.

6.3.2 Pseudo-Marginal Optimisation

Motivation. Component-wise updates within the SAME algorithm can induce poor mixing of the MCMC chain if Θ and X are correlated under $\pi \propto \mu \otimes \gamma$ because then Θ and Z_p are correlated under π_p. However, recall that X has only been introduced because the integral in Equation 6.3 is intractable and this prohibits the use of some 'ideal' marginal SA algorithm targeting distributions on the marginal space Θ, e.g. targeting $\pi_p(\mathrm{d}\theta) \propto \gamma(\theta, \mathbb{1})^{\beta_p}\mu(\mathrm{d}\theta)$ at Iteration p. It is well known that Monte Carlo methods are typically more efficient on a smaller space (unless better proposal distributions can be constructed on an extended space).

To mimic the behaviour of such an intractable marginal SA algorithm, we propose to adopt the pseudo-marginal framework (Beaumont, 2003; Andrieu & Roberts, 2009). To that end, we construct a different instance of the extended measure $\gamma_p(\mathrm{d}\theta \times \mathrm{d}z_p) = \mu(\mathrm{d}\theta)M_p(\theta, \mathrm{d}z_p)H_p(\theta, z_p)$ from Equation 6.2, denoted γ_p^{PM}.

Generic Extended Target Measure. Recall that we want to find the values $\theta \in \Theta$ that maximise $\gamma(\theta, \mathbb{1})$. To employ pseudo-marginal ideas, we augment the space with extra auxiliary variables (compared to the SAME approach). More specifically, let $\bar{\boldsymbol{X}} := (X, K, \boldsymbol{X}, Y)$ so that

$$\bar{\gamma}(\theta, \mathrm{d}\bar{\boldsymbol{x}}) = \gamma(\theta, \mathrm{d}x)\bar{\Pi}((\theta, x), \mathrm{d}k \times \mathrm{d}\boldsymbol{x} \times \mathrm{d}y),$$

is some instance of the generic MOSIS target measure from Section 1.4 (but which is now indexed by θ). This generic target measure is thus obtained by extending the measure $\gamma(\theta, \mathrm{d}x) = H(\theta, x)M(\theta, \mathrm{d}x)$ using a stochastic kernel $\bar{\Pi}$ as defined in Section 1.4. We recall that the normalised version of the thus extended measure, $\bar{\pi}(\theta, \cdot\,) \propto \bar{\gamma}(\theta, \cdot\,)$, satisfies

$$\bar{\boldsymbol{X}} \sim \bar{\pi}(\theta, \cdot\,) \quad \Rightarrow \quad X^K \sim \pi(\theta, \cdot\,) \propto \gamma(\theta, \cdot\,).$$

Also as in Section 1.4, we let

$$\bar{\psi}(\theta, \mathrm{d}\bar{\boldsymbol{x}}) = \psi(\theta, \mathrm{d}\boldsymbol{x} \times \mathrm{d}y)\xi((\theta, \boldsymbol{x}, y), \mathrm{d}k)\delta_{x^k}(\mathrm{d}x)$$

be some extended proposal distribution but whose components may now depend on θ, too. Furthermore, assuming that $\bar{\gamma}(\theta, \cdot\,) \ll \bar{\psi}(\theta, \cdot\,)$, we define the Radon–Nikodým derivative

$$\bar{w}^\theta := \frac{\mathrm{d}\bar{\gamma}(\theta, \cdot\,)}{\mathrm{d}\bar{\psi}(\theta, \cdot\,)}.$$

The extended target distribution of the pseudo-marginal SAME algorithm is then written as $\pi_p^{\text{PM}} \propto \gamma_p^{\text{PM}}$, where γ_p^{PM} is a measure on $\Theta \times \mathsf{Z}_p$ with $Z_p := \bar{\boldsymbol{X}}_p^{1:\lceil\beta_p\rceil}$. Writing $\mathsf{Z}_p := \bar{\mathbf{X}}^{\lceil\beta_p\rceil}$, where $\bar{\mathbf{X}} := \mathsf{X} \times \mathsf{K} \times \mathbf{X} \times \mathbf{Y}$, the measure γ_p^{PM} is defined by Equation 6.2 with

$$M_p(\theta, \mathrm{d}z_p) := \prod_{i=1}^{\lceil\beta_p\rceil} \bar{\psi}(\theta, \mathrm{d}\bar{\boldsymbol{x}}_p^i),$$
$$H_p(\theta, z_p) := \bar{w}^\theta(\bar{\boldsymbol{x}}_p^{\lceil\beta_p\rceil})^{\beta_p^\sharp} \prod_{i=1}^{\lfloor\beta_p\rfloor} \bar{w}^\theta(\bar{\boldsymbol{x}}_p^i).$$

Above, we have again set $\beta^\sharp := \beta - \lfloor\beta\rfloor$.

Whenever $\beta_p \in \mathbb{N}$, since $\bar{\gamma}(\theta,\,\cdot\,)$ admits $\gamma(\theta,\,\cdot\,)$ as a marginal,

$$\int\limits_{A \times \mathsf{Z}_p} \gamma_p^{\text{PM}}(\mathrm{d}\theta \times \mathrm{d}z_p) = \int\limits_{A \times \mathsf{X}^{\lceil\beta_p\rceil}} \gamma_p^{\text{SAME}}(\mathrm{d}\theta \times \mathrm{d}x_p^{1:\lceil\beta_p\rceil})$$
$$= \int\limits_A \gamma(\theta, \mathbb{1})^{\beta_p} \mu(\mathrm{d}\theta).$$

Hence, under suitable regularity conditions, by Equation 6.4, the marginal of π_p^{PM} in the θ-component concentrates around Θ_h as $p \to \infty$.

Implementation. As in Subsection 1.4.3, we may again set

$$\tilde{T}(\theta,\,\cdot\,) := \bar{\psi}(\theta,\,\cdot\,) \circ (\bar{w}^\theta)^{-1}.$$

This reparametrisation allows us to turn the potentially high-dimensional target measure γ_p^{PM} into a measure $\tilde{\gamma}_p^{\text{PM}}$ on the often lower-dimensional space $\Theta \times \mathsf{V}^{\lceil\beta_p\rceil}$, where $\mathsf{V} := [0, \infty)$. The measure $\tilde{\gamma}_p^{\text{PM}}$ is given by

$$\tilde{\gamma}_p^{\text{PM}}(\mathrm{d}\theta \times \mathrm{d}v^{1:\lceil\beta_p\rceil})$$
$$:= \mu(\mathrm{d}\theta)(v^{\lceil\beta_p\rceil})^{\beta_p^\sharp}\left[\prod_{i=1}^{\lfloor\beta_p\rfloor} v^i\right] \prod_{i=1}^{\lceil\beta_p\rceil} \tilde{T}(\theta, \mathrm{d}v^i).$$

Pseudo-marginal optimisation schemes then target $(\tilde{\pi}_p^{\text{PM}})_{p\in\mathbb{N}}$ using some instance of the generic MCMC kernel from Subsection 3.2.3.

In particular, if we target these distributions using MH kernels, we obtain the pseudo-marginal MH-based optimisation scheme which is outlined in Algorithm 6.6, where $Q_p \in \mathcal{K}_1(\Theta, \Theta)$ is some suitable proposal kernel for $\mathit{\Theta}$.

6.6 Algorithm (pseudo-marginal MH-based SAME). *At Step n,*

(1) If $\lceil \beta_n \rceil > \lceil \beta_{n-1} \rceil$, sample $V_n^{\lceil \beta_{n-1} \rceil + 1 : \lceil \beta_n \rceil} \sim \widetilde{T}^{\otimes(\lceil \beta_n \rceil - \lceil \beta_{n-1} \rceil)}(\theta_{n-1}, \cdot)$.

(2) Propose $\vartheta \sim Q_n(\theta_{n-1}, \cdot)$ and $W^{1:\lceil \beta_n \rceil} \sim \widetilde{T}^{\otimes \lceil \beta_n \rceil}(\vartheta, \cdot)$.

(3) Set $(\mathit{\Theta}_n, V_n^{1:\lceil \beta_n \rceil}) := (\vartheta, W^{1:\lceil \beta_n \rceil})$ with probability

$$1 \wedge \frac{\mu(\mathrm{d}\vartheta) Q_n(\vartheta, \mathrm{d}\theta_{n-1})}{\mu(\mathrm{d}\theta_{n-1}) Q_n(\theta, \mathrm{d}\vartheta)} \Biggl[\prod_{i=1}^{\lfloor \beta_n \rfloor} \frac{w^i}{v_{n-1}^i} \Biggr] \Biggl[\frac{w^{\lceil \beta_n \rceil}}{v_{n-1}^{\lceil \beta_n \rceil}} \Biggr]^{\beta_n^\sharp},$$

otherwise, set $(\mathit{\Theta}_n, V_n^{1:\lceil \beta_n \rceil}) := (\mathit{\Theta}_{n-1}, V_{n-1}^{1:\lceil \beta_n \rceil})$.

6.7 Example (hidden Markov models, continued). *Assume that the goal is to perform ML estimation in an HMM. Then $\gamma(\theta, \cdot)$ is proportional to the conditional posterior distribution of the latent states given some data and given the parameter θ. Similarly, $\mu \otimes \gamma$ can be interpreted as (being proportional to) the joint posterior distribution of the parameters and the latent states. If $\beta_n = 1$ and if $\bar{\gamma}(\theta, \cdot)$ and $\bar{\psi}(\theta, \cdot)$ represent the extended target measure and extended proposal distribution of the generic SMC algorithm presented in Chapter 2 (but now indexed by θ), the pseudo-marginal SAME kernel from Algorithm 6.6 reduces to a standard* particle marginal Metropolis–Hastings *(PMMH) kernel. In particular, this means that W^1 is equal to the normalising-constant estimate obtained from the SMC algorithm. In constrast, if $\beta_n > 1$, then the pseudo-marginal SAME kernel employs $\lceil \beta_n \rceil$ SMC algorithms and W^i is equal to the normalising-constant estimate obtained from the i th SMC algorithm.*

Pseudo-marginal MH kernels are well known to suffer from the so-called ‘stickiness’ problem. That is, a high variance of V^i can lead to long periods during which the algorithm rejects the proposed values at every iteration. This happens when one or more of the V^is in the denominator of the above acceptance probability takes a particularly large value.

To improve mixing, Beaumont (2003) proposes the *Monte Carlo within Metropolis* (MCWM) algorithm (so named in O'Neill et al. (2000)) which samples new values for the V^is at every iteration. While this introduces bias, it can sometimes improve mixing of the algorithm (Medina-Aguayo, Lee & Roberts, 2015). Moreover, the bias may be less problematic here because we are essentially only trying to find the mode. A single iteration of an optimisation scheme incorporating the MCWM-idea is summarised in Algorithm 6.8.

6.8 Algorithm (MCWM-based SAME). *At Step n,*

(1) propose $\vartheta \sim Q_n(\theta_{n-1}, \cdot)$,

(2) Sample $V^{1:\lceil \beta_n \rceil} \sim \widetilde{T}^{\otimes \lceil \beta_n \rceil}(\theta_{n-1}, \cdot)$ *and* $W^{1:\lceil \beta_n \rceil} \sim \widetilde{T}^{\otimes \lceil \beta_n \rceil}(\vartheta, \cdot)$.

(3) Set $\Theta_n := \vartheta$ *with probability*

$$1 \wedge \frac{\mu(\mathrm{d}\vartheta) Q_n(\vartheta, \mathrm{d}\theta_{n-1})}{\mu(\mathrm{d}\theta_{n-1}) Q_n(\theta, \mathrm{d}\vartheta)} \left[\prod_{i=1}^{\lfloor \beta_n \rfloor} \frac{w^i}{v^i}\right] \left[\frac{w^{\lceil \beta_n \rceil}}{v^{\lceil \beta_n \rceil}}\right]^{\beta_n^\sharp},$$

otherwise, set $\Theta_n := \Theta_{n-1}$.

Finally, as shown in Part I of this work, particular instances of the generic measure $\bar{\gamma}(\theta, \cdot)$ allow us to use a wide range of proposal distributions $\widetilde{T}(\theta, \cdot)$ within this framework, e.g. we may use SMC algorithms sample the variables V^i. In this case, Algorithm 6.6 is essentially an inhomogeneous PMMH algorithm with multiple independent sets of particles.

We can also employ other, more general instances of the MOSIS approach to devise the pseudo-marginal target distribution π_p^{PM} (or, equivalently, $\tilde{\pi}_p^{\text{PM}}$). Additionally, it is not necessary to target this distribution via MH kernels. Any other instance of the generic MCMC kernel from Subsection 3.2.3 can be used instead.

Relation With SAME. While we have constructed the pseudo-marginal MCMC optimisation schemes as a generalisation of the SAME approach (on the space $\Theta \times \mathsf{X}$), its extended target distribution may also be viewed as a special case of distribution targeted by the SAME algorithm (on the extended space $\Theta \times \bar{\mathbf{X}}$), i.e. as a SAME algorithm with latent variable $X = \bar{\boldsymbol{X}}$, and with

$$\begin{aligned} H(\theta, \bar{\boldsymbol{x}}) &= \bar{w}^\theta(\bar{\boldsymbol{x}}), \\ M(\theta, \cdot) &= \bar{\psi}(\theta, \cdot). \end{aligned}$$

6.3.3 Incorporation Into SMC Samplers

In this subsection, we briefly describe how the MCMC kernels devised in this section can be incorporated into SMC samplers.

Incorporating the CSMC-based SAME algorithm (Algorithm 6.5) into an SMC sampler can be achieved exactly as described in Subsection 6.2.3 but with the particular choice of the kernels Q_t, R_t and S_t specified in Subsection 6.3.1.

Incorporating pseudo-marginal SAME (Algorithm 6.6) into an SMC sampler proceeds again as in Subsection 6.2.3 but now with $\widetilde{P}_t$ being defined by Steps 2 and 3 of Algorithm 6.6 (with $n = t$). Furthermore, in this case, the kernel for augmenting the space with additional latent variables, $S_t \in \mathcal{K}_1(\Theta, \mathsf{X}^{\lceil \beta_t \rceil - \lceil \beta_{t-1} \rceil})$ is defined by

$$S_t(\theta_{t-1}, \cdot) := \widetilde{T}^{\otimes(\lceil \beta_t \rceil - \lceil \beta_{t-1} \rceil)}(\theta_{t-1}, \cdot).$$

6.4 Applications

6.4.1 Student-t Toy Model

As a first application of the pseudo-marginal SAME idea, we perform ML estimation the Student-t toy model from Gaetan and Yao (2003).

Model. In this case, the objective function is the (marginal) likelihood associated with observations $y = y_{1:4} = (-20, 1, 2, 3)$ which are assumed to have been generated from a non-central Student-t distribution with (known) $\nu := 0.05$ degrees of freedom and unknown location parameter $\theta \in \Theta := \mathbb{R}$. This distribution is denoted $\mathrm{t}_{\nu,\theta}$ and we use the same symbol for its Lebesgue-density. That is, the objective function is

$$h(\theta) = \prod_{t=1}^{4} \mathrm{t}_{\nu,\theta}(y_t) \propto \prod_{t=1}^{4} \bigl[1 + (y_t - \theta)^2/\nu\bigr]^{-(\nu+1)/2}.$$

Let $\mathrm{Gam}_{\alpha,\beta}$ denote the gamma distribution with shape parameters α and scale parameter β and let N_{m,v^2} denote the normal distribution with mean $m \in \mathbb{R}$ and variance $v^2 > 0$. Again, we use the same symbols for Lebesgue-densities of these distributions. It is well known and can easily

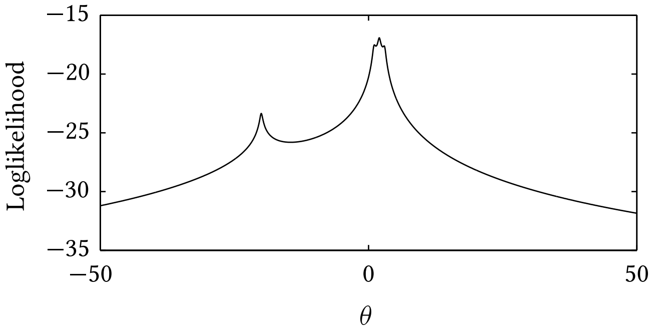

Figure 6.1 Log-objective function, i.e. loglikelihood, in the Student-t toy model.

be checked that the Student-t distribution arises if a gamma prior is placed over the precision (the reciprocal variance) of a normal distribution and if the precision-parameter is then integrated out, i.e.

$$\mathrm{t}_{\nu,\theta}(y_t) = \int_{\mathbb{R}} \mathrm{N}_{\theta,1/\tilde{x}_t}(y_t)\, \mathrm{Gam}_{\nu/2,2/\nu}(\mathrm{d}\tilde{x}_t).$$

Recall that we want to approximate the ML estimate of θ. Thus, in the notation from this chapter, $h(\theta) = \gamma(\theta, \mathbb{1}) = \int_{\mathsf{X}} H(\theta, x) M(\theta, \mathrm{d}x)$, where $X = \widetilde{X}_{1:4}$ takes values in $\mathsf{X} = \mathbb{R}^4$, and with

$$\begin{aligned} H(\theta, x) &:= \prod_{t=1}^{4} \mathrm{N}_{\theta,1/\tilde{x}_t}(y_t), \\ M(\theta, \mathrm{d}x) &:= \prod_{t=1}^{4} \mathrm{Gam}_{\nu/2,2/\nu}(\mathrm{d}\tilde{x}_t). \end{aligned}$$

In this case, due to the conjugacy outlined above, the objective function h can actually be evaluated point-wise. As illustrated in Figure 6.1, it has a global maximum at around 1.997 and has further local maxima at around −19.993, 1.086, and 2.906. However, to test the methods developed in the previous section, we pretend that we cannot evaluate h point-wise, i.e. we pretend that we cannot solve the integral $h(\theta) = \gamma(\theta, \mathbb{1})$ analytically.

Algorithms. We compare a number of algorithms which are listed below. Within the pseudo-marginal type algorithms, $\boldsymbol{X}_p^i := (\widetilde{X}_{p,1:4}^{i,1}, \dots, \widetilde{X}_{p,1:4}^{i,N})$ denotes the all the latent variables associated with the ith SAME replica at

Iteration p, i.e. $i \in \mathbb{N}_{\lceil \beta_p \rceil}$. Furthermore, $N \in \mathbb{N}$ is the number candidates in a multiple-proposal MCMC kernel or the number of pseudo-marginal replicas, i.e. the number of candidates involved in the construction of the ith weight in the acceptance probability, V_p^i.

(i) The first algorithm is a pseudo-Gibbs sampling implementation of SAME which updates the latent variables using the forced-move kernel from Subsection 3.3.1.

(ii) The second algorithm is a pseudo-marginal MH version of SAME.

(iii) The third algorithm is an MCWM-type version of SAME.

(iv) The fourth algorithm is an 'idealised' Gibbs-sampling implementation of SAME whose performance Algorithm i seeks to mimic.

(v) The fifth algorithm is the 'idealised' marginal SA algorithm whose performance Algorithms ii and iii seek to mimic.

For Algorithms i to iii, we compare the performance of two different proposal distributions for the latent variables.

(a) The first proposal simply entails sampling the latent variables from their prior distribution, i.e.

$$\psi(\theta, \mathrm{d}\boldsymbol{x}_p^i) := \prod_{t=1}^{4} \prod_{n=1}^{N} \mathrm{Gam}_{\nu/2, 2/\nu}(\mathrm{d}\tilde{x}_{p,t}^{i,n}).$$

Intuitively, this proposal distribution is inefficient because it takes neither the current value of θ nor the observations into account.

(b) Into the second proposal, we incorporate information about θ and about the observations. More precisely, letting Exp_λ denote the exponential distribution with rate $\lambda > 0$, we instead take

$$\psi(\theta, \mathrm{d}\boldsymbol{x}_p^i) := \prod_{t=1}^{4} \prod_{n=1}^{N} \mathrm{Exp}_{1/(1+|y_t - \theta|)}(\mathrm{d}\tilde{x}_{p,t}^{i,n}).$$

The algorithms are all initialised by sampling values for θ from the distribution $\mu \in \mathcal{M}_1(\Theta)$ which we take to be the uniform distribution on $[-50, 50]$ and which could be interpreted as the prior in the Bayesian paradigm. The specific form of this distribution on the estimates of Θ_h should diminish as $p \to \infty$.

We use 5,000 iterations. The inverse temperature increases linearly from $\beta_1 = 0.1$ to $\beta_{4{,}000} = 5$ and then remains constant for the last 1,000 iterations. At the nth iteration, θ is updated using a Gaussian random-walk MH kernel with variance $10/\beta_n$.

Results. We run each of these algorithms 200 times. Figure 6.2 shows the behaviour of the first three algorithms as the number of forced-move candidates/pseudo-marginal replicas, N, increases. Increasing N is clearly beneficial in each algorithm. Also beneficial is the incorporation of information about θ and about the observations into the proposal distribution for the latent variables (which is done in Algorithms ib and vb).

Figure 6.3 shows the estimates obtained from Algorithms i to iii. Interestingly, for large N and when using an efficient proposal distribution for the latent variables, the MCWM-type algorithm outperforms the 'exact' pseudo-marginal MH-type algorithms.

The Gibbs-sampling implementation of SAME (Algorithm iv) and the marginal SA algorithm (Algorithm v) perform even better. However, we stress that these algorithms rely on structure which is not available in more complicated scenarios. That is, SA requires point-wise evaluation of (a maximum-preserving) transformation of the objective function h while the Gibbs-sampling implementation of SAME requires the full conditional distributions of X under π_p^{SAME} to be tractable. These algorithms are only included here as a benchmark because they represent, in some sense, idealised algorithms whose behaviour Algorithms i to iii seek to mimic.

6.4.2 Linear Gaussian State-Space Model

Model. In this subsection, we apply the algorithms developed in the previous section to perform ML estimation in a linear Gaussian HMM, given by $X_0 \sim \mathrm{N}_{0,1}$ and for $t \in \mathbb{N}$,

$$\begin{aligned}\widetilde{X}_t &= A\widetilde{X}_{t-1} + B\varepsilon_t,\\ Y_t &= C\widetilde{X}_t + D\eta_t,\end{aligned}$$

where $\varepsilon_t, \eta_t \sim \mathrm{N}_{0,1}$ are IID and where we set $C = 1$ to ensure identifiability. We assume that we have obtained observations $Y_{1:T} = y_{1:T}$.

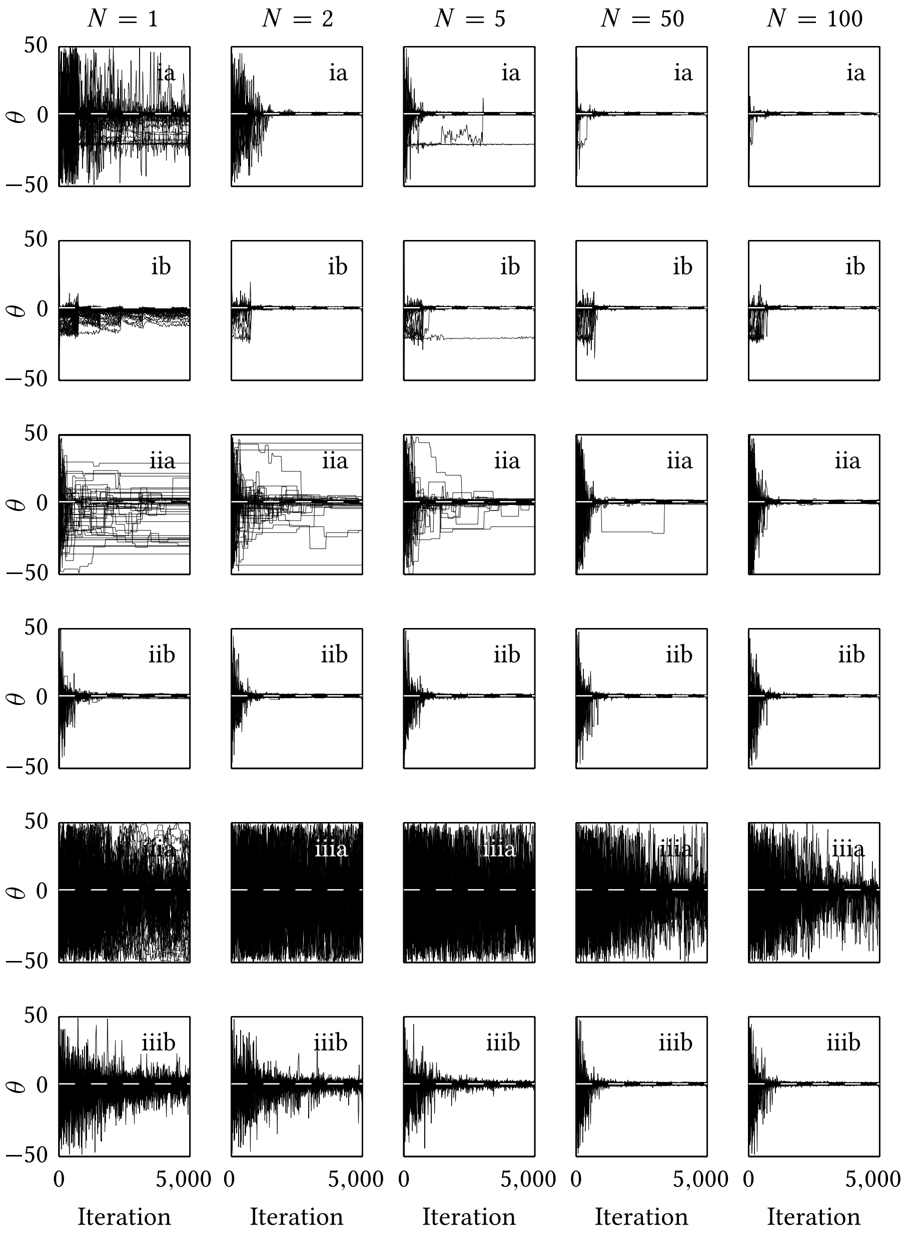

Figure 6.2 Traces of the first 50 runs of Algorithms i to iii in the Student-t toy model with varying numbers of forced-move candidates/pseudo-marginal replicas, N. The dashed line represents the location of the global maximum.

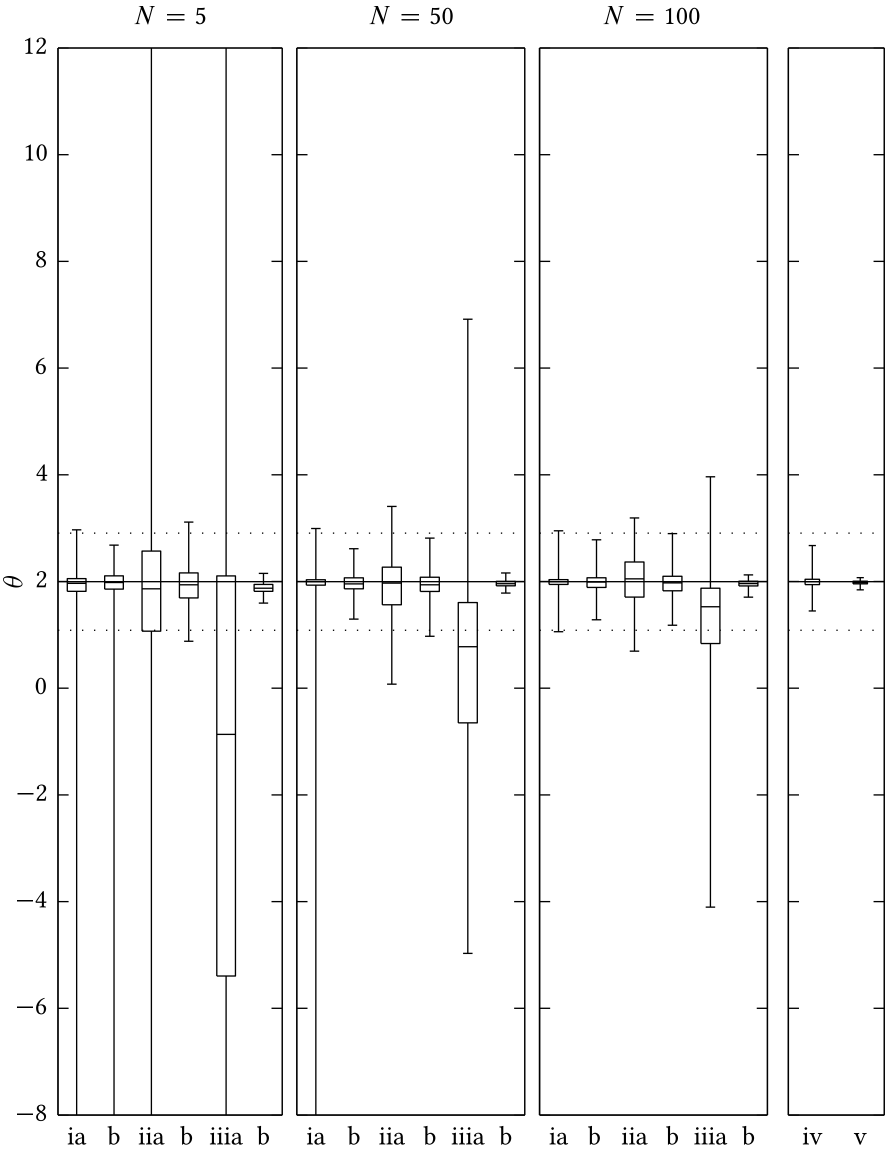

Figure 6.3 ML estimates from 200 runs of Algorithms i to iii with varying numbers of forced-move replicas/pseudo-marginal replicas, N, and from 200 runs of Algorithms iv and v, in the Student-t toy model. For each MCMC chain, an estimate is obtained by averaging over the last 1,000 iterations. The horizontal solid line and dotted lines represent the location of the global maximum and of local maxima, respectively.

We want to find $\theta := \theta_{1:3} := (A, \log B, \log D) \in \Theta := \mathbb{R}^3$ which maximises the (marginal) likelihood $h(\theta) = \gamma(\theta, \mathbb{1}) = \int_{\mathsf{X}} H(\theta, x) M(\theta, \mathrm{d}x)$, where the latent states $X := \widetilde{X}_{0:T}$ take values in $\mathsf{X} := \mathbb{R}^{T+1}$ and where

$$\begin{aligned} H(\theta, x) &:= \prod_{t=1}^{T} \mathrm{N}_{\tilde{x}_t, D^2}(y_t), \\ M(\theta, \mathrm{d}x) &:= \mathrm{N}_{0,1}(\mathrm{d}\tilde{x}_0) \prod_{t=1}^{T} \mathrm{N}_{A\tilde{x}_{t-1}, B^2}(\mathrm{d}\tilde{x}_t). \end{aligned}$$

Finally, we define the dominating measure $\mu \in \mathcal{M}_1(\Theta)$ as follows. Under μ, the components of $\varTheta$ are independent and we take $1/B^2 \sim \mathrm{Gam}_{1.5,1}$ as well as $1/D^2 \sim \mathrm{Gam}_{1.5,1}$. Additionally, we assume that the Markov chain $(X_t)_{t \in \mathbb{N}}$ is non-explosive (i.e. $|A| < 1$) and therefore restrict the support of A to $(-1, 1)$ by taking $A \sim \mathrm{Unif}_{(-1,1)}$. Again, the effect of μ on the estimates of Θ_h should diminish as $p \to \infty$.

Algorithms. We compare the following five algorithms based around SMC methods with a constant number of particles, $N := N_1 = \ldots = N_T$.

(i) The first algorithm mimics an 'idealised' Gibbs-sampling implementation of SAME by updating the latent variables using an iterated CSMC algorithm with *ancestor sampling* (AS) and using $N = 100$ particles.

(ii) The second algorithm is a PMMH version of SAME in which we propose the latent variables by running an SMC algorithm with $N = 1{,}00$ particles. It seeks to mimic an 'idealised' marginal SA algorithm.

(iii) The third algorithm is an MCWM-type version of Algorithm ii but with only $N = 500$ particles to account for the fact that this algorithm requires running twice as many SMC algorithms.

(iv) The fourth algorithm is the 'idealised' Gibbs-sampling implementation of SAME which samples the latent states $\widetilde{X}_{1:T}$ from their full conditional posterior distribution via standard forward–backward recursions (Rauch et al., 1965).

(v) The fifth algorithm is the 'idealised' marginal SA algorithm which exploits the fact that the latent variables can actually be integrated out analytically in this model. The recursions needed to calculate these integrals are known as the *Kalman filter* (Kalman, 1960).

In Algorithms i to iii, the SMC algorithm used to generate the latent variables propose the Step-t particles from the conditional prior distribution of the tth state, $\widetilde{X}_t$, given $\widetilde{X}_{1:t-1} = \tilde{x}_{1:t-1}$, i.e. from $\mathrm{N}_{A\tilde{x}_{t-1},B^2}$. The resulting algorithm is often called a 'bootstrap' particle filter.

The algorithms are all initialised by sampling values for θ from the 'prior' distribution μ. We use 5,000 iterations. The inverse temperature increases linearly from $\beta_1 = 1$ to $\beta_{4,000} = 10$ and then remains constant for the last 1,000 iterations.

In each algorithm, at the nth iteration, the parameters θ are updated using a mixture Gaussian random-walk MH kernel. The underlying normal distribution has diagonal covariance matrix $\operatorname{diag}([1, 1, 1])/(50\beta_n)$, with probability 0.8 and has covariance matrix $\operatorname{diag}([1, 1, 1])/50$, with probability 0.2. Here, $\operatorname{diag}(v)$ denotes a diagonal matrix whose diagonal is equal to the vector v. The support of the first component in the proposal kernels is restricted to $(-1, 1)$. In Algorithms i and Algorithms iv, we perform 100 such θ-updates at each iteration as these are relatively cheap.

Results. We obtain $T = 200$ observations from the model with true parameters $A = 0.9$ and $B = D = 1$. However, for a finite number of observations, the ML estimate does generally not coincide with the true parameter value.

We run the algorithms 15 times. Trace plots of the resulting Markov chains are shown in Figure 6.4. In addition, Figure 6.5 illustrates the variability in the estimates obtained from these algorithms.

6.4.3 Simple Stochastic Volatility Model

Model. In this subsection, we perform ML estimation in a simple univariate stochastic volatility model (Jacquier, Polson & Rossi, 1994). This model is given by $\widetilde{X}_1 \sim \mathrm{N}_{-7,1}$ and, for $t > 1$, we have

$$\begin{aligned} \widetilde{X}_t &= \alpha + \delta \widetilde{X}_{t-1} + \sigma \varepsilon_t, \\ Y_t &= \exp(\widetilde{X}_t/2)\eta_t. \end{aligned}$$

where $\varepsilon_t, \eta_t \sim \mathrm{N}_{0,1}$ are again IID. We assume that we have obtained observations $Y_{1:T} = y_{1:T}$, for some $T \in \mathbb{N}$.

Assuming the model to be non-explosive, i.e. assuming that $|\delta| < 1$, we want to find $\theta := \theta_{1:3} := (\alpha, \delta, \sigma) \in \Theta := \mathbb{R} \times (-1, 1) \times (0, \infty)$ maximising the (marginal) likelihood $h(\theta) = \gamma(\theta, \mathbb{1}) = \int_{\mathsf{X}} H(\theta, x) M(\theta, \mathrm{d}x)$.

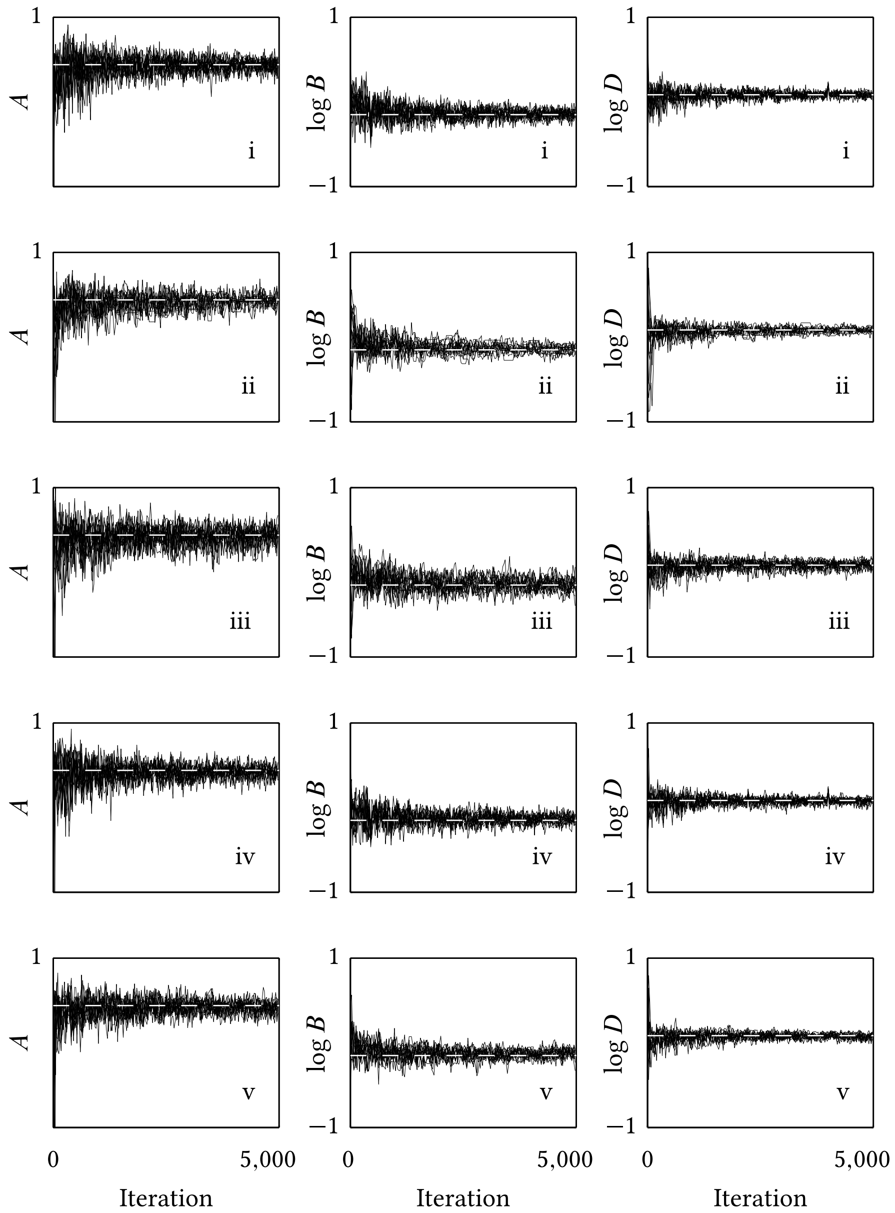

Figure 6.4 Traces obtained from 15 runs of Algorithms i to v in the linear Gaussian HMM. The dashed line represents the location of the global maximum.

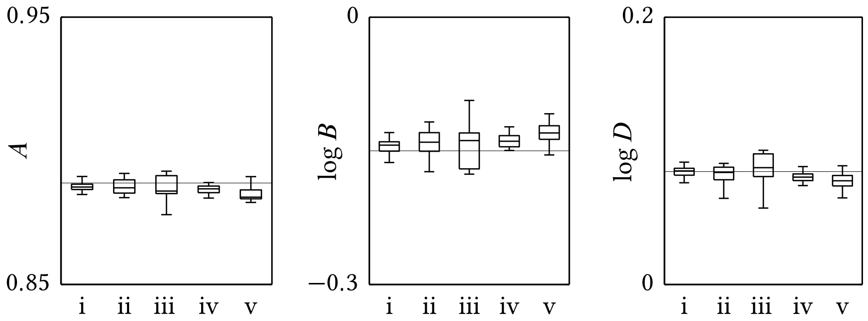

Figure 6.5 ML estimates obtained from 15 runs of Algorithms i to v in the linear Gaussian HMM. The horizontal line represents the location of the global maximum.

In this model, the states $X := \widetilde{X}_{1:T}$ take values in $\mathrm{X} := \mathbb{R}^T$ and

$$H(\theta, x) := \prod_{t=1}^{T} \mathrm{N}_{0,\exp(\tilde{x}_t)}(y_t),$$

$$M(\theta, \mathrm{d}x) := \mathrm{N}_{-7,1}(\mathrm{d}\tilde{x}_1) \prod_{t=2}^{T} \mathrm{N}_{\alpha+\delta\tilde{x}_{t-1},\sigma^2}(\mathrm{d}\tilde{x}_t).$$

Finally, we define the dominating measure $\mu \in \mathcal{M}_1(\Theta)$ following 'prior' distribution. Under μ, α, δ and σ are independent and $\alpha \sim \mathrm{N}_{0,100}$, $\delta \sim \mathrm{Unif}_{(-1,1)}$ and $1/\sigma \sim \mathrm{Gam}_{5,1}$.

Algorithms. We compare the following three algorithms, all based around a simple bootstrap particle filter with N particles.

(i) The first algorithm mimics an 'idealised' but intractable Gibbs-sampling implementation of SAME by updating the latent variables using an iterated CSMC algorithm with AS and using $N = 100$ particles.

(ii) The second algorithm is PMMH version of SAME using $N = 1{,}000$ particles. It aims to mimic an intractable 'idealised' marginal SA algorithm.

(iii) The third algorithm is an MCWM-type version of Algorithm ii with only $N = 500$ particles to account for the fact that this algorithms requires running twice as many SMC algorithms as the second.

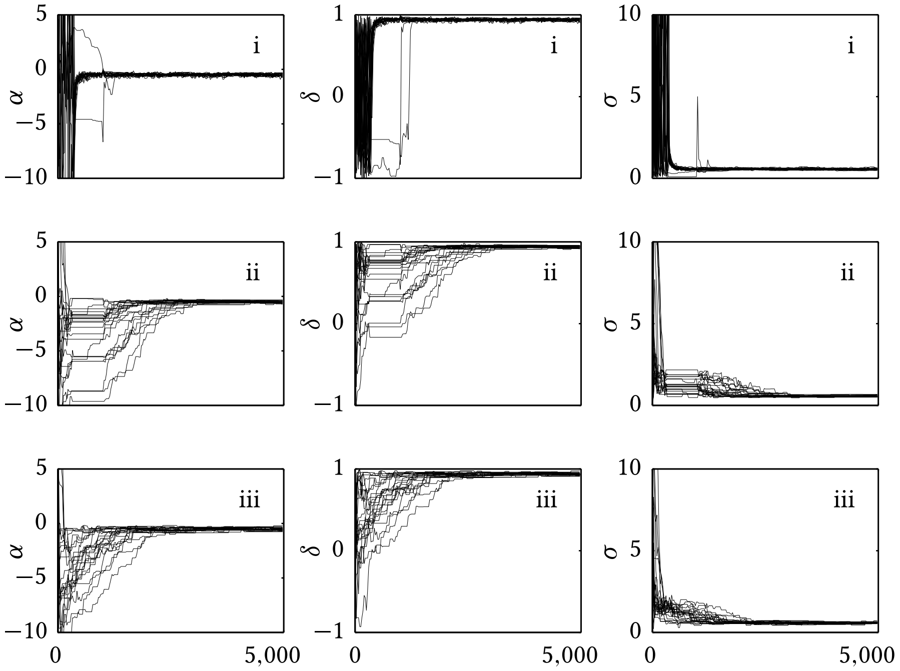

Figure 6.6 Traces obtained from 10 runs of Algorithms i to iii in the simple stochastic volatility model.

The algorithms are all initialised by sampling values for θ from the 'prior' distribution μ. We use 5,000 iterations. The inverse temperature increases linearly from $\beta_1 = 1$ to $\beta_{4,000} = 3$ and then remains constant for the last 1,000 iterations.

In each algorithm, at the nth iteration, the parameters θ are updated using a Gaussian random-walk MH kernel with diagonal covariance matrix $\mathrm{diag}([1, 1, 1])/(2\beta_n)$. In Algorithm i, we again perform 100 such updates at each iteration as these are comparatively cheap relative to the latent-variable updates.

Results. We obtain $T = 500$ observations from the model with true parameters $\alpha = -0.363$, $\delta = 0.95$, and $\sigma^2 = 0.26$. We run the algorithms 30 times. Trace plots θ-components of the resulting Markov chains are shown in Figure 6.6.

6.5 Discussion

In this chapter, we have presented methods for performing optimisation in latent-variable settings with the help of sophisticated modern Monte Carlo methods such as (iterated) CSMC and pseudo-marginal kernels.

Previously, Poyiadjis, Doucet and Singh (2011), Olsson, Cappé, Douc and Moulines (2008), Yıldırım, Singh and Doucet (2013), Nemeth, Fearnhead and Mihaylova (2015) have developed alternative algorithms for SMC-based optimisation. However, their methods are essentially gradient-based and it is not clear how they fare in settings in which the objective function is multimodal. Furthermore, these methods are restricted to parameter estimation in HMMs and then typically require point-wise evaluation of densities of the transition kernels associated with the latent states. Closed-form expressions for these densities are often unavailable in realistic models.

In contrast, the approaches presented here only require the ability to sample from the these transition kernels which is often feasible. A number of examples of such models can be found in Ionides, Bretó and King (2006). In addition, our methods can deal with much broader settings, e.g. with marginal ML or MAP estimation in non-Markovian models.

Work on the algorithms presented in this chapter is ongoing:

(1) Assume that the distribution of the individual weights W^i in Algorithm 6.6 does not depend on the parameter θ and that their variance $\mathbb{V}[W^i] =: \sigma_N^2$ *decreases* linearly in the parameter N which governs the number of 'candidates' in some underlying MOSIS scheme, e.g. N may be the number of particles when SMC algorithms are used to sample the weights. For simplicity, we assume that $\sigma_N^2 = \sigma^2/N$, for some $\sigma^2 > 0$. Meanwhile, assume that the computational cost of sampling W^i *increases* linearly in N (as is typically the case for standard IS or SMC algorithms).

In this setting, for $\beta_n = 1$, Sherlock et al. (2015), Doucet et al. (2015) show that for maximal efficiency of pseudo-marginal MH kernels, N should be chosen in such a way that σ_N^2 is around 1 (though more precisely, Andrieu and Vihola (2014) demonstrate that the variance of the weight is not always the right quantity by which to rank the efficiency of different pseudo-marginal MH kernels).

For inverse temperatures $\beta_n \in \mathbb{N} \setminus \{1\}$, the 'combined' weight in the pseudo-marginal SAME kernel is given by the product $\prod_{i=1}^{\beta_n} W^i$. For any $\theta \in \Theta$, its (conditional) variance is given by

$$\begin{aligned}\mathbb{V}\biggl[\prod_{i=1}^{\beta_n} W^i \biggm| \Theta = \theta\biggr] &= \mathbb{E}\biggl[\prod_{i=1}^{\beta_n} (W^i)^2\biggr] - \biggl(\mathbb{E}\biggl[\prod_{i=1}^{\beta_n} W^i\biggr]\biggr)^2 \\ &= \prod_{i=1}^{\beta_n} (\mathbb{V}[W^i] + \mathbb{E}[W^i]^2) - \prod_{i=1}^{\beta_n} \mathbb{E}[W^i]^2 \\ &= (\sigma^2/N + 1)^{\beta_n} - 1, \qquad (6.6)\end{aligned}$$

since the weights are conditionally independent given $\Theta = \theta$, since their (conditional) distribution is constant in θ, and since we can assume that $\mathbb{E}[W^i] = 1$ without loss of generality.

Solving Equation 6.6 for N suggests that we should take

$$N = \mathcal{O}\biggl(\frac{\sigma^2}{2^{1/\beta_n} - 1}\biggr),$$

i.e. effectively increase N linearly in β_n, to keep the variance of the 'combined' weight in the pseudo-marginal SAME kernel at around 1 as the inverse temperature increases. We are currently working on schemes for adaptively choosing N in this respect.

(2) We are currently also working towards extending the convergence analysis carried out in Andrieu et al. (2001) to the pseudo-marginal SAME setting.

(3) Finally, we are currently implementing the algorithms used in the examples in Section 6.4 within SMC samplers as described in Subsection 6.3.3. Furthermore, we are also currently applying our optimisation schemes to perform ML estimation in some of the more complicated Lévy-driven stochastic volatility models which were described in Chapter 5.

Conclusion

Summary

In this thesis, we have introduced a number of sophisticated non-standard Monte Carlo algorithms (1) for conducting inference in a challenging class of statistical models based around point processes, (2) for performing optimisation in latent-variable settings. These are based on a combination of *Markov chain Monte Carlo* (MCMC) methods, *sequential Monte Carlo* (SMC) methods and pseudo-marginal ideas.

To ease the explanation of the wide array of complex algorithms employed in this work, we have also presented a unifying Monte Carlo framework which is best described as *marginalised one-sample importance sampling* (MOSIS). Following Andrieu and Roberts (2009), Andrieu et al. (2010), Lee, Andrieu and Doucet (in prep.), Lee, Murray and Johansen (in prep.), this framework admits essentially any Monte Carlo scheme, including MCMC and SMC methods, as a special case.

Repeatedly applying MOSIS also justifies nesting SMC within MCMC or SMC within SMC (as in pseudo-marginal SMC algorithms). Furthermore, MOSIS forms the heart of a generic MCMC kernel which admits essentially any MCMC kernel as a special case, including (randomised) *Metropolis–Hastings* (MH) kernels and their extension to multiple proposals.

Further exploiting the MOSIS framework, this thesis has also presented new insights into the relationship between a number of modern Monte Carlo schemes. For instance, we have formally established the relationship

- between the discrete particle filter from Fearnhead (1998), Fearnhead and Clifford (2003) and other more conventional SMC methods,
- between the particle MCMC methods from Andrieu et al. (2010) and the ensemble MCMC methods from Neal (2011),
- between backward sampling (Whiteley, 2010) and ancestor sampling (Lindsten et al., 2012) for particle Gibbs samplers.

Contributions

Part I. Specifically, this thesis comprises the following novel contributions. In Part I, we have constructed the generic MOSIS framework which admits essentially all Monte Carlo schemes as special cases when viewing them on a suitably extended space. We have successfully applied this framework to analyse and enhance a number of existing algorithms.

Chapter 1 constructed the generic MOSIS framework. The structure of MOSIS estimators is not new. Indeed, they are based on the *importance sampling* (IS) schemes from Andrieu and Roberts (2009), Andrieu et al. (2010). However, our novel contribution in Part I has been to demonstrate that this framework represents the extended state-space justification of essentially all Monte Carlo schemes.

Chapter 2 presented a generic SMC algorithm and showed that it, too, represents a special case of MOSIS. Again, the justification of SMC methods as (one-sample) IS on an extended space is not new. It was already used in Andrieu et al. (2010) and extended to *non-exchangeable* and *adaptive* resampling schemes in Lee, Murray and Johansen (in prep.). In light of this, our contribution has been threefold.

- We have extended the construction from Lee, Murray and Johansen (in prep.) to also allow for *'biased'* resampling schemes.
- We have shown that the discrete particle filter from Fearnhead (1998) is a special case of the generic SMC algorithm.
- We have slightly generalised the 'importance tempering' approach from Gramacy et al. (2010) to re-use all the particles generated by an SMC sampler for approximating integrals. The resulting scheme may be viewed as a 'doubly' Rao–Blackwellised version of the recycling scheme from Nguyen et al. (2014).

Chapter 3 showed that MCMC algorithms, too, can be viewed as a special case of MOSIS. Employing the same framework at a lower level, we have also devised a generic MCMC kernel which admits essentially any MCMC kernel as a special case. Again, this generic kernel is not entirely new. It can be viewed as an extension of the approach from Tjelmeland (2004) and similar constructions can be found in Lee, Andrieu and Doucet (in prep.).

Our first contribution has therefore been pedagogical. We have shown that special cases of the generic MCMC kernel include MH kernels (Metropolis et al., 1953; Hastings, 1970), randomised MH kernels (Ceperley & Dewing, 1999; I. Murray et al., 2006), Barker's kernel (Barker, 1965), forced-move kernels (Chopin & Singh, 2013), ensemble MCMC kernels (Neal, 2011), and (iterated) *conditional sequential Monte Carlo* (CSMC) kernels (Andrieu et al., 2010). Our second contribution is to have established connections between various complex MCMC kernels. Specifically, we have shown the following.

- Iterated CSMC algorithms with *backward sampling* (BS) (Andrieu et al., 2010; Whiteley, 2010) and with *ancestor sampling* (AS) (Lindsten et al., 2012) share the same extended target distribution.
- The ensemble MCMC method from Shestopaloff and Neal (2013) is a pseudo-marginal MH kernel and more specifically, it can be viewed as a non-standard *particle marginal Metropolis–Hastings* (PMMH) kernel (Andrieu et al., 2010) in which all the parent indices are analytically integrated out in order to form the unbiased estimate of the normalising constant.

Part II. In Part II, we have combined a number of ideas from Part I to analyse, improve, and extend Monte Carlo schemes for a number of challenging problems.

Chapter 4 performed inference on the static parameters and latent variables in a class of piecewise deterministic processes via *particle Gibbs* (PG) samplers, based around a novel reformulation of the SMC filter from Whiteley et al. (2011). Specifically, our methodological contributions have been threefold.

- We have provided new insight into the approximation induced by this SMC filter and by related algorithms used in Del Moral et al. (2006b, 2007), Martin et al. (2013). We have also suggested a way of ensuring the existence of the importance weights.
- We have derived a new representation of this SMC filter which permits the use of BS and AS within PG samplers for piecewise deterministic processes.

- We have devised a novel PG step for rejuvenating a subset of the potentially large number of auxiliary variables used in an SMC filter. This reduces the impact of these auxiliary variables on the mixing of the PG chain while often also lowering the computational cost.

Chapter 5 performed PG-based inference on the static parameters (and latent variables) in a similar class of point-process models as Chapter 4. However, whereas Chapter 4 dealt with improving the underlying SMC algorithm, Chapter 5 focussed on reducing correlation in order to improve mixing of the (particle) Gibbs chain. Our contributions can be summarised as follows.

- We have combined a PG sampler with a non-centred parametrisation introduced by Roberts et al. (2004).
- We have applied the algorithms to a particularly challenging Lévy-driven stochastic volatility model.

Chapter 6 devised sophisticated Monte Carlo algorithms for performing optimisation in latent-variable settings, i.e. in situations in which the objective function cannot be evaluated point-wise. Such settings include marginal maximum likelihood and marginal maximum a-posteriori estimation. Specifically, we have extended the *state augmentation for marginal estimation* (SAME) algorithm from Doucet et al. (2002) in a number of directions which, among other benefits, allows SMC algorithms to be used as proposal distributions. Our contributions can be summarised as follows.

- We have combined SAME with multiple-proposal type MCMC kernels such as (iterated) CSMC kernels. The resulting algorithm can be viewed as mimicking an intractable 'idealised' Gibbs-sampling implementation of SAME.
- We have combined SAME with pseudo-marginal MCMC kernels such as PMMH kernels. The resulting algorithm can be viewed as mimicking an intractable 'idealised' marginal simulated annealing algorithm.
- We have proposed population-based versions of these approaches by incorporating them into SMC samplers.

Future Directions

Throughout Part I, we have shown that the generic MOSIS framework is particularly useful for structuring and comparing various Monte Carlo schemes. It also guarantees unbiased estimates of integrals with respect to a user-supplied target measure.

However, as we have stressed throughout this work, this construction is fundamentally importance sampling based on a single sample point. It therefore does not guarantee consistency. In particular, knowing that MCMC and SMC methods are a special case of the MOSIS framework does not abolish the need to check that MCMC kernels are ergodic or that SMC algorithms are stable in a suitable sense.

It would therefore be desirable to establish conditions under which a more general MOSIS-based estimator is consistent. This would be a step towards a unifying mathematical framework for analysing convergence properties of Monte Carlo approximations.

In addition, the work in this thesis suggests the following more specific avenues of further research.

- As mentioned in Subsection 3.4.3, the fact that (iterated) CSMC algorithms with BS and AS both target the same extended distribution could perhaps be exploited to analyse and compare the convergence properties of these algorithms. This could potentially be achieved by slightly generalising the approach taken in Andrieu et al. (2013).
- As pointed out in Section 4.6, it would be desirable to extend the reformulation of the SMC filter for piecewise deterministic processes to allow for multiple-birth-moves. Such an extension would reduce or even remove the bias introduced by using only single-birth moves.
- As discussed in Section 5.5, it would be useful to more formally characterise the identifiability issues in multiple-component Lévy-driven stochastic volatility models for which these properties we empirically observed in Section 5.4.4.
- As stressed in Section 6.5, it would be beneficial to analyse the convergence properties of the SAME algorithm and the various extensions to it proposed in Section 6.3. For instance, it would be of interest to gain some more formal insight into the relationship between the choice of temperature schedule and the number of particles and to choose the latter automatically and adaptively.

A Resampling Schemes

A.1 Overview

In this appendix, we summarise a number of widely used resampling schemes for SMC algorithms. For completeness, we also state the conditional versions of the resampling schemes as derived in Andrieu et al. (2010), Whiteley et al. (2010), Lee, Murray and Johansen (in prep.). These are required for sampling from CSMC kernels, e.g. within particle Gibbs kernels.

A full comparison of resampling schemes was undertaken in Douc, Cappé and Moulines (2005). We only present schemes with a fixed number of particles. Resampling schemes which induce a random number of particles have been developed in Crisan et al. (1998).

To simplify the presentation, we drop the dependence on $\boldsymbol{Z}_{1:t-1}$ from all subsequent notation. We note that when using an unbiased resampling schemes (such as multinomial, stratified, and systematic resampling) at Step t and if $\kappa_t(u_{1:t}, \cdot) = \mathrm{Unif}_{\mathbb{N}_{N_t}}$, then use of the time-reversal kernel from Assumption 2.9 implies

$$\Lambda_t((u_{1:t}, o_{t-1}, a_{t-1}^k), \{k\}) = \frac{\widetilde{R}_{t-1}^{\mathrm{M}}(k, \{a_{t-1}^k\})}{W_{t-1}^{a_{t-1}^k}}.$$

As a result, the 'marginal' resampling distribution $\widetilde{R}_{t-1}^{\mathrm{M}}(k, \{a_{t-1}^k\})$ cancels out in the expression of the particle weights in Equation 2.5 and does not actually have to be computed.

A.2 Multinomial Resampling

Multinomial resampling was employed within the first SMC algorithms (Stewart & McCarty Jr, 1992; Gordon et al., 1993). In this case, the joint resampling scheme factorises as

$$\widetilde{R}_{t-1} := \mathrm{Mul}_{\boldsymbol{W}_{t-1}}^{\otimes N_t},$$

where $\boldsymbol{W}_{t-1} := (W_{t-1}^1, \ldots, W_{t-1}^{N_{t-1}})$ denotes the vector of Step-$(t-1)$ self-normalised weights and Mul_p represents the multinomial distribution for some vector of probabilities, p.

By (conditional) independence, the marginal distribution associated with the parent index of the kth offspring is then defined by

$$\widetilde{R}_{t-1}^{\mathrm{M}}(k, \cdot) := \mathrm{Mul}_{\boldsymbol{W}_{t-1}}.$$

The conditional resampling distribution given the kth parent index is given by the product of the remaining $N_t - 1$ marginals.

A.3 Stratified Resampling

Stratified resampling (Kitagawa, 1996) generates the vector of parent indices according to

$$\widetilde{R}_{t-1}(\{\boldsymbol{a}_{t-1}\}) := \int_{[0,1]^{N_t}} \nu(\mathrm{d}\boldsymbol{u}) \prod_{n=1}^{N_t} \sum_{l=1}^{N_{t-1}} \mathbb{1}_{D^l}(u^n + n - 1)\delta_l(\{a_{t-1}^n\}),$$

where $\boldsymbol{U} := U^{1:N_t}$, $\nu := \mathrm{Unif}_{[0,1]}^{\otimes N_t}$, and

$$D^l := N_t\Bigg(\sum_{m=1}^{l-1} W_{t-1}^m, \sum_{m=1}^{l} W_{t-1}^m\Bigg].$$

Hence, by (conditional) independence of the elements in the vector $\boldsymbol{U}$, the marginal resampling distribution simplifies to

$$\begin{aligned}&\widetilde{R}_{t-1}^{\mathrm{M}}(k, \{a_{t-1}^k\})\\ &\quad= \int_{[0,1]} \mathrm{Unif}_{[0,1]}(\mathrm{d}u) \sum_{l=1}^{N_{t-1}} \mathbb{1}_{D^l}(u + k - 1)\delta_l(\{a_{t-1}^k\})\\ &\quad= \mathrm{Leb}(D^{a_{t-1}^k} \cap (k-1, k]).\end{aligned}$$

For the same reason, the conditional resampling distribution given the kth parent index is again the product of the remaining $N_t - 1$ marginals.

A.4 Systematic Resampling

Systematic resampling (Carpenter, Clifford & Fearnhead, 1999) induces a similar joint distribution over the parent indices as stratified resampling, except that we now take

$$\nu(\mathrm{d}\boldsymbol{u}) := \mathrm{Unif}_{[0,1]}(\mathrm{d}u^1) \prod_{n=2}^{N_t} \delta_{u^1}(\mathrm{d}u^n).$$

The marginal resampling distribution is then the same as in the case of stratified resampling.

However, in contrast to stratified resampling, the parent indices are not conditionally independent. To sample from the conditional resampling distribution given the kth parent index, A_{t-1}^k, we may simply extend the space to include U^1 in the marginal resampling distribution. Then the conditional distribution of U^1 under this distribution is

$$\begin{aligned}&\frac{\mathrm{Leb}|_{[0,1]}(\mathrm{d}u) \sum_{l=1}^{N_{t-1}} \mathbb{1}_{D^l}(u+k-1)\delta_l(\{a_{t-1}^k\})}{\mathrm{Leb}(D^{a_{t-1}^k} \cap (k-1,k])} \\ &\qquad = \mathrm{Unif}_{\{v \in [0,1] | v+k-1 \in D^{a_{t-1}^k} \cap (k-1,k]\}}(\mathrm{d}u).\end{aligned}$$

Having determined U^1 (and, by extension, $U^{2:N_t}$) the remaining parent indices $\boldsymbol{A}_{t-1}^{-k}$ are then determined.

A.5 Optimal Finite-State Resampling

In this section, we describe the resampling scheme for the discrete particle filter from Fearnhead (1998) (and whose conditional version was derived by Whiteley et al. (2010)) which is also briefly summarised in Subsection 2.3.4.

Having used Fearnhead (1998, Algorithm 5.2) to solve

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

for $C_{t-1} > 0$,

- define $L_s := \#\{n \in \mathsf{K}_{t-1} \mid W_{t-1}^n > 1/C_{t-1}\}$ and $H_t := \mathsf{K}_{t-1} \setminus L_t$,
- let $l_t \colon \mathbb{N}_{\#L_t} \to L_t$ and $h_t \colon \mathbb{N}_{\#H_t} \to H_t$ be the functions which map m to the mth largest element in L_t and H_t, respectively.

Writing $\boldsymbol{I} := (I_1, \ldots, I_{M_t - \# L_t})$, $\mathsf{I} := \mathbb{N}_{\# H_t}$, and $\boldsymbol{\mathsf{I}} := \mathsf{I}^{M_t - \# L_t}$, the joint resampling distribution can be represented as

$$\begin{aligned}\widetilde{R}_{t-1}(\{\boldsymbol{a}_{t-1}\}) := {} & \left[\prod_{n=1}^{\# L_t} \delta_{l_t(n)}(\{a_{t-1}^{n}\})\right] \\ & \times \left[\sum_{\boldsymbol{i} \in \boldsymbol{\mathsf{I}}} \bar{R}_{t-1}(\{\boldsymbol{i}\}) \prod_{n=1}^{M_t - \# L_t} \delta_{h_t(i_n)}(\{a_{t-1}^{\# L_t + n}\})\right] \\ & \times \prod_{n=2}^{K} \delta_{a_{t-1}^{1:M_t}}(\{a_{t-1}^{(n-1)M_t + 1:nM_t}\}), \qquad \text{(A.1)}\end{aligned}$$

where $\bar{R}_{t-1}$ represents a systematic resampling scheme for generating parent indices associated with $M_t - \# L_t$ offspring based on $\# H_t$ parents which are associated with the re-normalised weights $\bar{W}_{t-1}^{1:\# H_t}$, defined via

$$\bar{W}_{t-1}^{n} := \frac{W_{t-1}^{h_t(n)}}{\sum_{m \in H_t} W_{t-1}^{m}} = \frac{C_{t-1} W_{t-1}^{h_t(n)}}{M_t - \# L_t},$$

for $n \in \mathbb{N}_{\# H_t}$, where the last equality is due to the definition of C_{t-1}.

The marginal resampling distribution for generating the parent index for the kth offspring is thus given by $\widetilde{R}_{t-1}^{\mathrm{M}}(k, \cdot) = \delta_{l_t(k)}$, in the case that $k \bmod M_t \leq \# L_t$. Else, if $k \bmod M_t > \# L_t$, by the properties of systematic resampling,

$$\widetilde{R}_{t-1}^{\mathrm{M}}(k, \{a_{t-1}^{k}\}) = \operatorname{Leb}\big(\bar{D}^{h_t^{-1}(a_{t-1}^{k})} \cap D_k\big), \qquad \text{(A.2)}$$

where h_t^{-1} is the inverse h_t and

$$\begin{aligned} D_k &:= ((k \bmod M_t) - \# L_t - 1, (k \bmod M_t) - \# L_t], \\ \bar{D}^{l} &:= (M_t - \# L_t)\left(\sum_{m=1}^{l-1} \bar{W}_{t-1}^{m}, \sum_{m=1}^{l} \bar{W}_{t-1}^{m}\right]. \end{aligned}$$

As in Subsection 2.3.4, we take $\kappa_t(u_{1:t}, \cdot)$ to be the uniform distribution on $\mathbb{Z}_{(u_t - 1)M_t + 1, u_t M_t}$. By Assumption 2.9, we then have

$$\Lambda_t((u_{1:t}, o_{t-1}, a_{t-1}^{k}), \{k\}) = \frac{\widetilde{R}_{t-1}^{\mathrm{M}}(k, \{a_{t-1}^{k}\})}{1 \wedge C_{t-1} W_{t-1}^{a_{t-1}^{k}}}.$$

Note that only the denominator is therefore needed to calculate the particle weights in an SMC algorithm.

In a CSMC algorithm, to sample a particular particle index $B_t = K$ conditional on knowing the associated parent index A_{t-1}^{K} and to perform conditional resampling in order to generate the remaining the parent indices, we proceed as follows.

- If $C_{t-1}W_{t-1}^{a_{t-1}^{k}} > 1$, we set $k := l_t^{-1}(a_{t-1}^{k})$, where l_t^{-1} denotes the inverse of l_t, and sample the remaining parent indices according to the right hand side in Equation A.1 (ignoring the kth factor in the first product).
- If $C_{t-1}W_{t-1}^{a_{t-1}^{k}} < 1$, we sample $K = k$ according to the distribution proportional to right hand side in Equation A.2 (interpreted as a measure in k). We then only need to perform a conditional version of the systematic resampling scheme $\overline{R}_{t-1}$. This proceeds as described in Section A.4. The remaining parent indices $A_{t-1}^{M_t+1:N_t}$ are then also determined.

Notation

$\ll$	‘is absolutely continuous with respect to’
$:=, =:$	‘is defined as’, ‘defines’
$\sim, \propto$	‘is distributed according to’, ‘is proportional to’
$\min, \wedge; \max, \vee$	minimum; maximum
$\inf, \sup$	infimum, supremum
$\otimes$	tensor product
$\# A$	cardinality of a set A
$\Psi_g(\mu)$	Boltzmann–Gibbs transformation of a measure μ under g
$\mathbb{E}, \mathbb{V}; \mathbb{E}_\mu, \mathbb{V}_\mu$	expectation, variance under $\mathbb{P}$; under μ
id	identiy function
$\mathbb{1}_A, \mathbb{1}$	indicator function of a set A, unit function
$\mathcal{O}, \mathcal{O}_{\mathbb{P}}$	‘big O’, ‘big O in probability’ asymptotic notation
$\mathbb{N}, \mathbb{Z}, \mathbb{R}$	set of positive integers, integers, real numbers
$\mathbb{N}_k$	set of positive integers up to k ($\{n \in \mathbb{N} \mid n \leq k\}$)
$\mathbb{Z}_{k,l}$	set of integers from k to l ($\{z \in \mathbb{Z} \mid k \leq z \leq l\}$)
$\mathcal{B}(\mathsf{X})$	Borel σ-algebra on X
$\mathcal{F}(\mathsf{X}, \mathsf{Y})$	set of $\mathcal{B}(\mathsf{X})/\mathcal{B}(\mathsf{Y})$-measurable functions
$\mathcal{L}(\mu), \mathcal{L}^p(\mu)$	set of μ-integrable, p-times μ-integrable real functions
$\mathcal{M}_\sigma(\mathsf{X}), \mathcal{M}(\mathsf{X})$	set of positive σ-finite and positive finite measures on X
$\mathcal{M}_1(\mathsf{X})$	set of probability measures on X
$\mathcal{K}_\sigma(\mathsf{X}, \mathsf{Y}), \mathcal{K}(\mathsf{X}, \mathsf{Y})$	set of positive σ-finite, positive finite kernels from X to Y
$\mathcal{K}_1(\mathsf{X}, \mathsf{Y})$	set of stochastic kernels from X to Y
δ_x	Dirac measure/point mass centred at x
Exp_λ	exponential distribution with rate λ
Dir_α	Dirichlet distribution with parameter vector α
$\mathrm{Gam}_{\alpha,\beta}$	gamma distribution with shape, scale parameters α, β
Leb	Lebesgue measure on $\mathbb{R}$
Mul_p	multinomial distribution with probabilities p
$\mathrm{N}_{\mu,\Sigma}$	normal distribution with mean μ, covariance matrix Σ
$\mathrm{t}_{\nu,\mu,\sigma}$	non-central Student-t distribution with ν degrees of freedom
Unif_A	uniform distribution on a set A

Abbreviations

AS	ancestor sampling
BS	backward sampling
CDF	cumulative distribution function
CLT	central limit theorem
CP	centred parametrisation
CSMC	conditional sequential Monte Carlo
DPF	discrete particle filter
ESS	effective sample size
FFBS	forward filtering–backward smoothing
HMM	(general state-space) hidden Markov model
IID	independent and identically distributed
IS	importance sampling
MAP	maximum a-posteriori
MCMC	Markov chain Monte Carlo
MCWM	Monte Carlo within Metropolis
MH	Metropolis–Hastings
ML	maximum likelihood
MOSIS	marginalised one-sample importance sampling
NCP	non-centred parametrisation
PDP	piecewise deterministic process
PG	particle Gibbs
PMMH	particle marginal Metropolis–Hastings
PPP	Poisson point process
RJMCMC	reversible-jump Markov chain Monte Carlo
RSMC	reformulated sequential Monte Carlo
SA	simulated annealing
SAME	state augmentation for marginal estimation
SIR	sequential importance resampling
SLLN	strong law of large numbers
SMC	sequential Monte Carlo
VRPF	variable-rate particle filter

References

Alquier, P., Friel, N., Everitt, R. & Boland, A. (2014). Noisy Monte Carlo: convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, 1–19.

Andrieu, C., Breyer, L. A. & Doucet, A. (2001). Convergence of simulated annealing using Foster–Lyapunov criteria. *Journal of Applied Probability*, *38*(4), 975–994.

Andrieu, C., Doucet, A. & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 269–342. With discussion.

Andrieu, C., Lee, A. & Vihola, M. (2013, December 22). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. arXiv: 1312.6432v1

Andrieu, C. & Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, *37*(2), 697–725.

Andrieu, C. & Vihola, M. (2014, April 28). Establishing some order amongst exact approximations of MCMCs. arXiv: 1404.6909v1

Andrieu, C. & Vihola, M. (2015). Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms. *The Annals of Applied Probability*, *25*(2), 1030–1077.

Barker, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a proton–electron plasma. *Australian Journal of Physics*, *18*(2), 119–134.

Barndorff-Nielsen, O. E. & Shephard, N. (2001). Non-Gaussian Ornstein–Uhlenbeck-based models and some of their uses in financial economics. *63*(2), 167–241.

Baum, L. E., Petrie, T., Soules, G. & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 164–171.

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, *164*(3), 1139–1160.

Bellman, R. E. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Besag, J. (1997). Contribution to the discussion on 'On Bayesian analysis of mixtures with an unknown number of components' by Richardson, S. and Green, P. J. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *59*(4), 774.

Beskos, A., Crisan, D., Jasra, A., Kamatani, K. & Zhou, Y. (2014, December 11). A stable particle filter in high-dimensions. arXiv: 1412.3501v1

Beskos, A., Jasra, A. & Thiéry, A. H. (2014, February 6). On the convergence of adaptive sequential Monte Carlo methods. arXiv: 1306.6462v3

Billingsley, P. (2012). *Probability and measure* (Anniversary Edition). Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley.

Brooks, S. P., Giudici, P. & Roberts, G. O. (2003). Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *65*(1), 3–39.

Bunch, P. & Godsill, S. J. (2013). Particle smoothing algorithms for variable rate models. *IEEE Transactions on Signal Processing*, *61*(5-8), 1663–1675.

Calderhead, B. (2014). A general construction for parallelizing Metropolis–Hastings algorithms. *Proceedings of the National Academy of Sciences*, *111*(49), 17408–17413.

Cappé, O., Godsill, S. J. & Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, *95*(5), 899–924.

Cappé, O., Guillin, A., Marin, J.-M. & Robert, C. P. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics*, *13*(4).

Cappé, O., Moulines, E. & Rydén, T. (2005). *Inference in hidden Markov models*. Springer Series in Statistics. New York, NY: Springer.

Carlin, B. P. & Chib, S. (1995). Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 473–484.

Carpenter, J., Clifford, P. & Fearnhead, P. (1999). An improved particle filter for nonlinear problems. *IEE Proceedings – Radar, Sonar and Navigation*, *146*(1), 2–7.

Casella, G. & Robert, C. P. (1996). Rao–Blackwellisation of sampling schemes. *Biometrika*, *83*(1), 81–94.

Centanni, S. & Minozzo, M. (2006a). A Monte Carlo approach to filtering for a class of marked doubly stochastic Poisson processes. *Journal of the American Statistical Association*, *101*(476), 1582–1597.

Centanni, S. & Minozzo, M. (2006b). Estimation and filtering by reversible jump MCMC for a doubly stochastic Poisson model for ultra-high-frequency financial data. *Statistical Modelling*, *6*(2), 97–118.

Ceperley, D. M. & Dewing, M. (1999). The penalty method for random walks with uncertain energies. *The Journal of Chemical Physics*, *110*(20), 9812–9820.

Cérou, F., Del Moral, P. & Guyader, A. (2011). A nonasymptotic theorem for unnormalized Feynman–Kac particle models. *Annales de l'Institut Henri Poincaré, Probability and Statistics*, *47*(3), 629–649.

Cérou, F., LeGland, F., Del Moral, P. & Lezaud, P. (2005). Limit theorems for the multilevel splitting algorithm in the simulation of rare events. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong & J. A. Joines (Eds.), *Proceedings of the 2005 Winter Simulation Conference* (pp. 682–691). IEEE.

Chen, R. (2010). Contribution to the discussion on 'Particle Markov chain Monte Carlo methods' by Andrieu, C., Doucet, A., and Holenstein, R. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 305–306.

Chen, Y. (2005). Another look at rejection sampling through importance sampling. *Statistics & Probability Letters*, *72*(4), 277–283.

Chib, S. & Greenberg, E. (1995). Understanding the Metropolis–Hastings algorithm. *The American Statistician*, *49*(4), 327–335.

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., Jacob, P. E. & Papaspiliopoulos, O. (2013). SMC^2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *75*(3), 397–426.

Chopin, N. & Singh, S. S. (2013, April 6). On the particle Gibbs sampler. arXiv: 1304.1887v1

Christen, J. A. & Fox, C. (2005). Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics*, *14*(4), 795–810.

Crisan, D., Del Moral, P. & Lyons, T. (1998). *Discrete filtering using branching and interacting particle systems.*

Dassios, A. & Jang, J.-W. (2003). Pricing of catastrophe reinsurance and derivatives using the Cox process with shot noise intensity. *Finance and Stochastics*, *7*(1), 73–95.

Del Moral, P. (1995). Nonlinear filtering using random particles. *Theory of Probability & Its Applications*, *40*(4), 690–701.

Del Moral, P. (1996). Nonlinear filtering: interacting particle solution. *Markov Processes and Related Fields*, *2*(4), 555–580.

Del Moral, P. (2004). *Feynman-Kac formulae: genealogical and interacting particle systems with applications.* New York, NY: Springer.

Del Moral, P. (2013). *Mean field simulation for Monte Carlo integration.* Monographs on Statistics & Applied Probability. Boca Raton, FL: Chapman & Hall/CRC.

Del Moral, P. & Doucet, A. (2014). Particle methods: an introduction with applications. In *ESAIM proceedings* (Vol. 44, pp. 1–46). EDP Sciences.

Del Moral, P., Doucet, A. & Jasra, A. (2006a). A note on the use of Metropolis–Hastings kernels in importance sampling. Retrieved January 17, 2015, from Arnaud Doucet's homepage: http://www.cs.ubc.ca/~arnaud/note_smcsamplers.pdf

Del Moral, P., Doucet, A. & Jasra, A. (2006b). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(3), 411–436.

Del Moral, P., Doucet, A. & Jasra, A. (2007). Sequential Monte Carlo for Bayesian computation. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith & M. West (Eds.), *Bayesian statistics 8: Proceedings of the Eighth Valencia International Meeting*, 2nd–6th June 2006 (Vol. 8, pp. 115–148). Oxford University Press.

Del Moral, P., Doucet, A. & Jasra, A. (2012). On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, *18*(1), 252–278.

Del Moral, P., Doucet, A. & Singh, S. S. (2010, December 24). Forward smoothing using sequential Monte Carlo. arXiv: 1012.5390v1

Del Moral, P. & Guionnet, A. (1999). Central limit theorem for nonlinear filtering and interacting particle systems. *Annals of Applied Probability*, 275–297.

Del Moral, P. & Guionnet, A. (2001). On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincaré, Probability and Statistics*, *37*(2), 155–194.

Del Moral, P. & Miclo, L. (2000). Branching and interacting particle systems approximations of Feynman–Kac formulae with applications to non-linear filtering. In J. Azéma, M. Émery, M. Ledoux & M. Yor (Eds.), *Lecture notes in mathematics: Vol. 1729. Séminaire de probabilités XXXIV*. Berlin, Germany: Springer-Verlag.

Dellaert, F., Fox, D., Burgard, W. & Thrun, S. (1999). Monte Carlo localization for mobile robots. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation* (Vol. 2, pp. 1322–1328). IEEE.

Douc, R., Cappé, O. & Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis* (pp. 64–69). Zagreb, Croatia: IEEE.

Douc, R., Garivier, A., Moulines, E. & Olsson, J. (2011). Sequential Monte Carlo smoothing for general state space hidden Markov models. *The Annals of Applied Probability*, *21*(6), 2109–2145.

Douc, R., Maire, F. & Olsson, J. (2014). On the use of Markov chain Monte Carlo methods for the sampling of mixture models. *Statistics and Computing*, 1–16. To appear.

Douc, R. & Moulines, E. (2008). Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *The Annals of Statistics*, 2344–2376.

Douc, R., Moulines, E. & Olsson, J. (2014). Long-term stability of sequential Monte Carlo methods under verifiable conditions. *The Annals of Applied Probability*, *24*(5), 1767–1802.

Doucet, A., Briers, M. & Sénécal, S. (2006). Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, *15*(3), 693–711.

Doucet, A., Godsill, S. J. & Robert, C. P. (2002). Marginal maximum a posteriori estimation using Markov chain Monte Carlo. *Statistics and Computing*, *12*(1), 77–84.

Doucet, A., Godsill, S. J. & West, M. (2000). Monte Carlo filtering and smoothing with application to time-varying spectral estimation. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal processing* (Vol. 2, pp. 701–704). IEEE.

Doucet, A. & Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: fifteen years later. In D. Crisan & B. Rozovskii (Eds.), *The Oxford handbook of nonlinear filtering* (Chap. 24, pp. 656–704). Oxford Handbooks. Oxford, UK: Oxford University Press.

Doucet, A., Pitt, M., Deligiannidis, G. & Kohn, R. (2015). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, asu075.

Durbin, J. & Koopman, S. J. (2012). *Time series analysis by state space methods* (2nd ed.). Oxford Statistical Science Series. Oxford, UK: Oxford University Press.

Eckhard, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo method [Special issue]. *Los Alamos Science*, *15*, 131–137.

Epstein, E. S. (1969). Stochastic dynamic prediction. *Tellus*, *21*(6), 739–759.

Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory* (Doctoral dissertation, Department of Statistics, University of Oxford, UK).

Fearnhead, P. (2010). Contribution to the discussion on 'Particle Markov chain Monte Carlo methods' by Andrieu, C., Doucet, A., and Holenstein, R. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 302–304.

Fearnhead, P. & Clifford, P. (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *65*(4), 887–899.

Fearnhead, P. & Meligkotsidou, L. (2014, August 29). Augmentation schemes for particle MCMC. arXiv: 1408.6980v1

Fearnhead, P., Papaspiliopoulos, O., Roberts, G. O. & Stuart, A. (2010). Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(4), 497–512.

Ferguson, T. S. & Klass, M. J. (1972). A representation of independent increment processes without Gaussian components. *The Annals of Mathematical Statistics*, 1634–1643.

Finke, A., Johansen, A. M. & Spanò, D. (2014). Static-parameter estimation in piecewise deterministic processes using particle Gibbs samplers. *Annals of the Institute of Statistical Mathematics*, *66*(3), 577–609.

Fishman, G. S. (1996). *Monte Carlo: concepts, algorithms, and applications*. Springer Series in Operations Research. New York, NY: Springer.

Gaetan, C. & Yao, J.-F. (2003). A multiple-imputation Metropolis version of the EM algorithm. *Biometrika*, *90*(3), 643–654.

Gandy, A. & Lau, F. D.-H. (2015, April 20). The chopthin algorithm for resampling. arXiv: 1502.07532v3

Gelfand, A. E. & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, *85*(410), 398–409.

Gelman, A. (1992). Iterative and non-iterative simulation algorithms. *Computing Science and Statistics*, (24), 433–348.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. & Rubin, D. B. (2013). *Bayesian data analysis* (3rd ed.). Boca Raton, FL: CRC Press.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

Gerber, M. & Chopin, N. (2015). Sequential quasi Monte Carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *77*(3), 509–579.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, *57*(6), 1317–1339.

Geyer, C. J. (2011). Introduction to Markov chain Monte Carlo. In S. Brooks, A. Gelman, G. L. Jones & X.-L. Meng (Eds.), *Handbook of Markov chain Monte Carlo* (pp. 3–48). Handbooks of Modern Statistical Methods. Boca Raton, FL: Chapman & Hall/CRC.

Gibson, G. J. & Renshaw, E. (1998). Estimating parameters in stochastic compartmental models using Markov chain methods. *Mathematical Medicine and Biology*, *15*(1), 19–40.

Gilks, W. R. & 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*. New York, NY: Springer.

Godsill, S. J. (2001). On the relationship between Markov chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, (2), 230–248.

Godsill, S. J., Doucet, A. & West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, *99*(465).

Godsill, S. J. & Vermaak, J. (2004). Models and algorithms for tracking using trans-dimensional sequential Monte Carlo. In *Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 3, pp. 976–979). Montreal, Canada: IEEE.

Goertzel, G. (1949). *Quota sampling and importance functions in stochastic solution of particle problems* (Technical report No. 434). Oak Ridge National Laboratory.

Goertzel, G. & Kahn, H. (1949). *Monte Carlo methods for shield computation* (Technical report No. 429). Oak Ridge National Laboratory.

Gordon, N. J., Salmond, D. J. & Smith, A. F. M. (1993). Novel approach to nonlinear/-non-Gaussian Bayesian state estimation. *IEE Proceedings F, Radar and Signal Processing*, *140*(2), 107–113.

Gramacy, R., Samworth, R. & King, R. (2010). Importance tempering. *Statistics and Computing*, *20*(1), 1–7.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, *82*(4), 711–732.

Greenberg, E. (2012). *Introduction to Bayesian econometrics*. New York, NY: Cambridge University Press.

Griffin, J. E. & Steel, M. F. (2006). Inference with non-Gaussian Ornstein–Uhlenbeck processes for stochastic volatility. *Journal of Econometrics*, *134*(2), 605–644.

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of operations research*, *13*(2), 311–329.

Hammersley, J. M. & Morton, K. W. (1954). Poor man's Monte Carlo. *Journal of the Royal Statistical Society. Series B (Methodological)*, *16*(1), 23–38.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*(1), 97–109.

Heine, K. (2005). Unified framework for sampling/importance resampling algorithms. In *Proceedings of the 8th International Conference on Information Fusion* (Vol. 2, pp. 1459–1464). IEEE.

Hwang, C.-R. (1980). Laplace's method revisited: weak convergence of probability measures. *The Annals of Probability*, *8*(6), 1177–1182.

Iba, Y. (2000). Population Monte Carlo algorithms. *Transactions of the Japanese Society for Artificial Intelligence*, *16*, 279–286.

Ionides, E. L., Bretó, C. & King, A. A. (2006). Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, *103*(49), 18438–18443.

Jacob, P. E., Murray, L. M. & Rubenthaler, S. (2013). Path storage in the particle filter. *Statistics and Computing*, 1–10.

Jacquier, E., Johannes, M. & Polson, N. (2007). MCMC maximum likelihood for latent state models. *Journal of Econometrics*, *137*(2), 615–640.

Jacquier, E., Polson, N. G. & Rossi, P. E. (1994). Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, *12*(4), 69–87.

Jarzynski, C. (1997a). Equilibrium free-energy differences from nonequilibrium measurements: a master-equation approach. *Physical Review E*, *56*(5), 5018.

Jarzynski, C. (1997b). Nonequilibrium equality for free energy differences. *Physical Review Letters*, *78*(14), 2690.

Jasra, A. & Doucet, A. (2008). Stability of sequential Monte Carlo samplers via the Foster–Lyapunov condition. *Statistics & Probability Letters*, *78*(17), 3062–3069.

Jasra, A., Doucet, A., Stephens, D. A. & Holmes, C. C. (2008). Interacting sequential Monte Carlo samplers for trans-dimensional simulation. *Computational Statistics & Data Analysis*, *52*(4), 1765–1791.

Jasra, A., Lee, A., Yau, C. & Zhang, X. (2013, March 21). The alive particle filter. arXiv: 1304.0151v1

Johansen, A. M. & Doucet, A. (2008). A note on auxiliary particle filters. *Statistics & Probability Letters*, *78*(12), 1498–1504.

Johansen, A. M. & Doucet, A. (in prep.). *The hierarchical particle filter.*

Johansen, A. M., Doucet, A. & Davy, M. (2008). Particle methods for maximum likelihood estimation in latent variable models. *Statistics and Computing*, *18*(1), 47–57.

Johansen, A. M., Whiteley, N. & Doucet, A. (2012). Exact approximation of Rao–Blackwellised particle filters. *System Identification*, *16*(1), 488–493.

Kahn, H. (1949). *Stochastic (Monte Carlo) attenuation analysis* (Technical report No. P-88). RAND Corporation.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, *82*, 35–45.

Kantas, N., Beskos, A. & Jasra, A. (2014). Sequential monte carlo methods for high-dimensional inverse problems: a case study for the navier–stokes equations. *SIAM/ASA Journal on Uncertainty Quantification*, *2*(1), 464–489.

Kantas, N., Lecchini-Visintini, A. & Maciejowski, J. M. (2010). Simulation-based Bayesian optimal design of aircraft trajectories for air traffic management. *International Journal of Adaptive Control and Signal Processing*, *24*(10), 882–899.

Kantas, N., Maciejowski, J. M. & Lecchini-Visintini, A. (2009). Sequential Monte Carlo for model predictive control. In *Nonlinear model predictive control* (pp. 263–273). Springer.

Kingman, J. F. C. (1992). *Poisson processes.* Oxford, UK: Oxford University Press.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, *5*(1), 1–25.

Kong, A., Liu, J. S. & Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, *89*(425), 278–288.

Künsch, H. R. (2005). Recursive Monte Carlo filters: algorithms and theoretical analysis. *Annals of Statistics*, 1983–2021.

Lapeyre, B., Pardoux, É. & Sentis, R. (2003). *Introduction to Monte-Carlo methods for transport and diffusion equations* (A. Craig & F. Craig, Trans.). Oxford texts in applied and engineering mathematics. Oxford, UK: Oxford University Press.

Lee, A. (2011). *On auxiliary variables and many-core architectures in computational statistics* (Doctoral dissertation, Department of Statistics, University of Oxford, UK).

Lee, A., Andrieu, C. & Doucet, A. (in prep.). *Active particles and locally adaptive Markov chain Monte Carlo.*

Lee, A., Murray, L. M. & Johansen, A. M. (in prep.). *Resampling in conditional SMC algorithms.*

Lee, A., Yau, C., Giles, M. B., Doucet, A. & Holmes, C. C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of Computational and Graphical Statistics*, *19*(4), 769–789.

Leith, C. E. (1974). Theoretical skill of Monte Carlo forecasts. *Monthly Weather Review*, *102*(6), 409–418.

Liang, F. (2002). Dynamically weighted importance sampling in Monte Carlo computation. *Journal of the American Statistical Association*, *97*(459), 807–821.

Lin, M., Chen, R. & Liu, J. S. (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science*, *28*(1), 69–94.

Lindsten, F., Douc, R. & Moulines, E. (2015). Uniform ergodicity of the particle gibbs sampler. *Scandinavian Journal of Statistics.*

Lindsten, F., Johansen, A. M., Naesseth, C. A., Kirkpatrick, B., Schön, T. B., Aston, J. A. D. & Bouchard-Côté, A. (2014, June 19). Divide-and-Conquer with sequential Monte Carlo. arXiv: 1406.4993v1

Lindsten, F., Jordan, M. I. & Schön, T. B. (2012). Ancestor sampling for particle Gibbs. In *Proceedings of the 2012 Conference on Neural Information Processing Systems.* Lake Tahoe, NV.

Lindsten, F., Jordan, M. I. & Schön, T. B. (2014). Particle gibbs with ancestor sampling. *The Journal of Machine Learning Research*, *15*(1), 2145–2184.

Liu, J. S. (1996). Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika*, *83*(3).

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing.* Springer Series in Statistics. New York, NY: Springer.

Liu, J. S., Liang, F. & Wong, W. H. (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, *95*(449), 121–134.

Liu, J. S., Liang, F. & Wong, W. H. (2001). A theory for dynamic weighting in Monte Carlo computation. *Journal of the American Statistical Association*, *96*(454), 561–573.

Lokovic, T. & Veach, E. (2000). Deep shadow maps. In *Proceedings of the 27nd Annual Conference on Computer Graphics and Interactive Techniques* (pp. 385–392). ACM.

MacEachern, S. N., Clyde, M. & Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: the next generation. *Canadian Journal of Statistics*, *27*(2), 251–267.

Madras, N. & Piccioni, M. (1999). Importance sampling for families of distributions. *The Annals of Applied Probability*, *9*(4), 1202–1225.

Maire, F., Douc, R. & Olsson, J. (2014). Comparison of asymptotic variances of inhomogeneous Markov chains with application to Markov chain Monte Carlo methods. *The Annals of Statistics*, *42*(4), 1483–1510.

Martin, J. S., Jasra, A. & McCoy, E. (2013). Inference for a class of partially observed point process models. *Annals of the Institute of Statistical Mathematics*, *65*(3), 413–437.

Medina-Aguayo, F., Lee, A. & Roberts, G. O. (2015, March 24). Stability of noisy Metropolis–Hastings. arXiv: 1503.07066v1

Metropolis, N. (1987). The beginning of the Monte Carlo method [Special issue]. *Los Alamos Science*, *15*, 125–130.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*(6), 1087–1092.

Metropolis, N. & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, *44*(247), 335–341.

Meyn, S. P. & Tweedie, R. L. (2009). *Markov chains and stochastic stability* (2nd ed.). Cambridge, UK: Cambridge University Press.

Mira, A. (1998). *Ordering, slicing and splitting Monte Carlo Markov chains* (Doctoral dissertation, University of Minnesota, US).

Murray, I., Ghahramani, Z. & MacKay, D. J. C. (2006). MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence* (pp. 359–366). Cambridge, MA: AUAI Press.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, *11*(2), 125–139.

Neal, R. M. (2003, May 1). Markov chain sampling for non-linear state space models using embedded hidden Markov models. arXiv: math/0305039

Neal, R. M. (2011, January 2). MCMC using ensembles of states for problems with fast and slow variables such as Gaussian process regression. arXiv: 1101.0387v1

Nemeth, C., Fearnhead, P. & Mihaylova, L. (2015, February 3). Particle approximations of the score and observed information matrix for parameter estimation in state space models with linear computational cost. arXiv: 1306.0735v2

Nguyen, T. L. T., Septier, F., Peters, G. W. & Delignon, Y. (2014). Improving SMC sampler estimate by recycling all past simulated particles. (pp. 117–120). 2014 IEEE Workshop on Statistical Signal Processing. IEEE.

Nicholls, G. K., Fox, C. & Watt, A. M. (2012, May 30). Coupled MCMC with a randomized acceptance probability. arXiv: 1205.6857v1

Olsson, J., Cappé, O., Douc, R. & Moulines, E. (2008). Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, *14*(1), 155–179.

Olsson, J. & Rydén, T. (2011). Rao–Blackwellization of particle Markov chain Monte Carlo methods using forward filtering backward sampling. *IEEE Transactions on Signal Processing*, *59*(10), 4606–4619.

Olsson, J. & Westerborn, J. (2014, December 23). Efficient particle-based online smoothing in general hidden markov models: the PaRIS algorithm. arXiv: 1412.7550v1

O'Neill, P. D., Balding, D. J., Becker, N. G., Eerola, M. & Mollison, D. (2000). Analyses of infectious disease data from household outbreaks by Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *49*(4), 517–542.

O'Neill, P. D. & Roberts, G. O. (1999). Bayesian inference for partially observed stochastic epidemics. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *162*(1), 121–129.

The 86th Scientific & Technical Awards. (2014). Retrieved January 17, 2015, from Academy of Motion Picture Arts and Sciences website: http://www.oscars.org/sci-_tech/ceremonies/2014

FiveThirtyEight. (2015). Retrieved January 26, 2015, from http://fivethirtyeight.com

Owen, A. & Zhou, Y. [Yi]. (2000). Safe and effective importance sampling. *Journal of the American Statistical Association*, *95*(449), 135–143.

Papaspiliopoulos, O. (2003). *Non-centered parameterisations for data augmentation and hierarchical models* (Doctoral dissertation, Department of Statistics, University of Lancaster, UK).

Papaspiliopoulos, O., Roberts, G. O. & Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, 59–73.

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, *60*(3), 607–612.

Peters, G. W. (2005). *Topics in sequential Monte Carlo samplers* (Master's thesis, Department of Engineering, University of Cambridge, UK).

Pitt, M. K. & Shephard, N. (1999). Filtering via simulation: auxiliary particle filters. *Journal of the American statistical association*, *94*(446), 590–599.

Poyiadjis, G., Doucet, A. & Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, *98*(1), 65–80.

R Development Core Team. (2014). R: a language and environment for statistical computing (Version 3.1.2) [Computer software]. Vienna, Austria: R Foundation for Statistical Computing.

Rao, V. & Teh, Y. W. (2013). Fast MCMC sampling for Markov jump processes and extensions. *The Journal of Machine Learning Research*, *14*(1), 3295–3320.

Rauch, H. E., Striebel, C. T. & Tung, F. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, *3*(8), 1445–1450.

Robert, C. P. & Casella, G. (2004). *Monte Carlo statistical methods* (2nd ed.). Springer Texts in Statistics. New York, NY: Springer.

Roberts, G. O., Papaspiliopoulos, O. & Dellaportas, P. (2004). Bayesian inference for non-Gaussian Ornstein–Uhlenbeck stochastic volatility processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *66*(2), 369–393.

Roberts, G. O. & Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys*, *1*, 20–71.

Rosenbluth, M. N. & Rosenbluth, A. W. (1955). Monte Carlo calculation of the average extension of molecular chains. *The Journal of Chemical Physics*, *23*(2), 356–359.

Rota, G.-C. (2008). The lost café. *Indiscrete Thoughts*, 63–85.

Rubenthaler, S., Rydén, T. & 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.

Sherlock, C., Thiéry, A. H., Roberts, G. O. & Rosenthal, J. S. (2015). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Annals of statistics*, *43*(1), 238–275.

Shestopaloff, A. Y. & Neal, R. M. (2013, May 2). MCMC for non-linear state space models using ensembles of latent sequences. arXiv: 1305.0320v1

Shestopaloff, A. Y. & Neal, R. M. (2014, December 9). Efficient Bayesian inference for stochastic volatility models with ensemble MCMC methods. arXiv: 1412.3013v1

Sokal, A. (1997). Monte Carlo methods in statistical mechanics: foundations and new algorithms. In C. DeWitt-Morette, P. Cartier & A. Folacci (Eds.), *Functional integration* (Vol. 361, pp. 131–192). NATO ASI Series. New York, NY: Springer.

Spanier, J. & Gelbard, E. M. (1969). *Monte Carlo principles and neutron transport problems.* Reading, MA: Addison-Wesley.

Stewart, L. & McCarty Jr, P. (1992). Use of Bayesian belief networks to fuse continuous and discrete information for target recognition, tracking, and situation assessment. In *Aerospace sensing* (pp. 177–185). International Society for Optics and Photonics.

Storvik, G. (2011). On the flexibility of Metropolis–Hastings acceptance probabilities in auxiliary variable proposal generation. *Scandinavian Journal of Statistics*, *38*(2), 342–358.

Tanner, M. A. & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, *82*(398), 528–550.

The MathWorks, Inc. (2015). MATLAB and statistics toolbox (Version 2014b) [Computer software]. Natick, MA: The MathWorks, Inc.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, *22*(4), 1701–1728.

Tierney, L. (1998). A note on Metropolis–Hastings kernels for general state spaces. *Annals of Applied Probability*, 1–9.

Tjelmeland, H. (2004). *Using all Metropolis–Hastings proposals to estimate mean values* (preprint No. 4/2004). Norwegian University of Science and Technology, Trondheim, Norway.

Tran, M.-N., Scharth, M., Pitt, M. K. & Kohn, R. (2014, January 24). Importance sampling squared for Bayesian inference in latent variable models. arXiv: 1309.3339v3

Trotter, H. F. & Tukey, J. W. (1956). Conditional Monte Carlo for normal samples. In H. A. Meyer (Ed.), *Symposium on Monte Carlo methods* (pp. 64–79). New York, NY: Wiley.

Van Dyk, D. A. & Park, T. (2008). Partially collapsed Gibbs samplers: theory and methods. *Journal of the American Statistical Association*, *103*(482), 790–796.

Veach, E. & Guibas, L. J. (1995). Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (pp. 419–428). ACM.

Vergé, C., Dubarry, C., Del Moral, P. & Moulines, E. (2013). On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 1–18.

Wang, X., Chen, R. & Guo, D. (2002). Delayed-pilot sampling for mixture Kalman filter with application in fading channels. *IEEE Transactions on Signal Processing*, *50*(2), 241–254.

Whiteley, N. (2010). Contribution to the discussion on 'Particle Markov chain Monte Carlo methods' by Andrieu, C., Doucet, A., and Holenstein, R. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 306–307.

Whiteley, N. (2012). Sequential Monte Carlo samplers: error bounds and insensitivity to initial conditions. *Stochastic Analysis and Applications*, *30*(5), 774–798.

Whiteley, N. (2013). Stability properties of some particle filters. *The Annals of Applied Probability*, *23*(6), 2500–2537.

Whiteley, N., Andrieu, C. & Doucet, A. (2010, October 10). Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods. arXiv: 1011.2437v1

Whiteley, N., Johansen, A. M. & Godsill, S. J. (2011). Monte Carlo filtering of piecewise deterministic processes. *Journal of Computational and Graphical Statistics*, *20*(1), 119–139.

Whiteley, N. & Lee, A. (2014). Twisted particle filters. *The Annals of Statistics*, *42*(1), 115–141.

Wilkinson, D. J. (2011). *Stochastic modelling for systems biology* (2nd ed.). Boca Raton, FL: CRC Press.

Winkler, G. (2003). *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction.* New York, NY: Springer.

Wong, W. H. & Liang, F. (1997). Dynamic weighting in Monte Carlo and optimization. *Proceedings of the National Academy of Sciences*, *94*(26), 14220–14224.

Yıldırım, S., Singh, S. S. & Doucet, A. (2013). An online expectation–maximization algorithm for changepoint models. *Journal of Computational and Graphical Statistics*, *22*(4), 906–926.

Yu, Y. & Meng, X.-L. (2011). To center or not to center: that is not the question – an ancillarity–sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics*, *20*(3), 531–570.

Zhang, J. L. & Liu, J. S. (2002). A new sequential importance sampling method and its application to the two-dimensional hydrophobic–hydrophilic model. *The Journal of Chemical Physics*, *117*(7), 3492–3498.

Zhao, H., Jiang, B. & Canny, J. (2014, September 18). SAME but different: fast and high-quality Gibbs parameter estimation. arXiv: 1409.5402v1

Zhou, Y. [Yan], Johansen, A. M. & Aston, J. A. D. (2013, March 13). Towards automatic model comparison: an adaptive sequential Monte Carlo approach. arXiv: 1303.3123v1