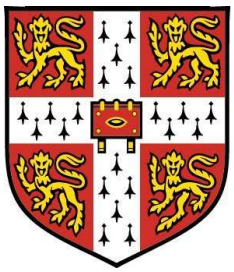
Maximum Likelihood Parameter Estimation in Time Series Models Using Sequential Monte Carlo



Sinan Yıldırım

Darwin College

Department of Pure Mathematics and Mathematical Statistics

University of Cambridge

A thesis submitted for the degree of

Doctor of Philosophy

To Selcan and İlhan...

Declaration

This dissertation is the result of work carried out by myself between October 2009 and October 2012. It includes nothing which is the outcome of work done in collaboration with others, except as specified in the text.

Signed:

Sinan Yıldırım

Acknowledgements

I would like to thank my supervisor, Dr. Sumeetpal S. Singh for his supervision, assistance, and friendship. Having greatly benefited from working with him for over three years, I am very glad to have been his PhD student.

I would like to express my gratitude to my advisor Prof. A. Philip Dawid, who has been extremely kind and supportive throughout my PhD.

Many thanks to my collaborators Dr. A. Taylan Cemgil, Dr. Tom Dean, Lan Jiang, and Prof. Arnaud Doucet for their invaluable contributions to the development of this thesis.

I must thank my proofreaders Ozan Aksoy and Peter Bunch, whose efforts have improved the presentation of this work significantly.

Also, many thanks to all my friends for providing the happy moments of this stressful period of time. Besides my special thanks to my 'mentor' Prof. Mihalis Dafermos, my housemate Yasemin Aslan, and my 'music-mate' Dr. Kyriacos Leptos, I owe a special word of appreciation to Selcan Deniz Kolağasıoğlu, who shared all the happiness, sorrow, excitement with me.

Finally, I would like to thank my parents and relatives, especially my dear brother İlhan and my best cousin and best friend Hasan İlkay Çelik, for their endless support and encouragement all these years.

Abstract

Time series models are used to characterise uncertainty in many real-world dynamical phenomena. A time series model typically contains a static variable, called parameter, which parametrizes the joint law of the random variables involved in the definition of the model. When a time series model is to be fitted to some sequentially observed data, it is essential to decide on the value of the parameter that describes the data best, a procedure generally called parameter estimation.

This thesis comprises novel contributions to the methodology on parameter estimation in time series models. Our primary interest is online estimation, although batch estimation is also considered. The developed methods are based on batch and online versions of expectation-maximisation (EM) and gradient ascent, two widely popular algorithms for maximum likelihood estimation (MLE). In the last two decades, the range of statistical models where parameter estimation can be performed has been significantly extended with the development of Monte Carlo methods. We provide contribution to the field in a similar manner, namely by combining EM and gradient ascent algorithms with sequential Monte Carlo (SMC) techniques. The time series models we investigate are widely used in statistical and engineering applications.

The original work of this thesis is organised in Chapters 4 to 7. Chapter 4 contains an online EM algorithm using SMC for MLE in changepoint models, which are widely used to model heterogeneity in sequential data. In Chapter 5, we present batch and online EM algorithms using SMC for MLE in linear Gaussian multiple target tracking models. Chapter 6 contains a novel methodology for implementing MLE in a hidden Markov model having intractable probability densities for its observations. Finally, in Chapter 7 we formulate the nonnegative matrix factorisation problem as MLE in a specific hidden Markov model and propose online EM algorithms using SMC to perform MLE.

Contents

List of Figures

List of Tables

List of Abbreviations

a.e.	almost everywhere
a.s.	almost surely
ABC	approximate Bayesian computation
AMPF	auxiliary marginal particle filter
CPHD	Cardinalised PHD
EM	expectation-maximisation
FFBS	forward filtering backward smoothing
GLSSM	Gaussian linear state-space model
HMM	hidden Markov model
i.i.d.	independently and identically distributed
JPDAF	joint probabilistic data association filter
MCEM	Monte Carlo EM
MCMC	Markov chain Monte Carlo
MCMC-DA	MCMC-data association
MHT	multiple hypothesis tracking
MLE	maximum likelihood estimation
MPF	marginal particle filter
MTT	multiple target tracking
NMF	nonnegative matrix factorisation
PHD	probability hypothesis density
PMCMC	particle Markov chain Monte Carlo
PMHT	probabilistic MHT
RBPF	Rao-Blackwellised particle filter
S-ABC MLE	smoothed ABC MLE
SAEM	stochastic approximation EM
SEM	stochastic EM
SIS	sequential importance sampling
SISR	sequential importance sampling resampling
SMC	sequential Monte Carlo
SN-ABC MLE	smoothed noisy ABC MLE
SVαR	stochastic volatility model with α-stable returns

Chapter 1

Introduction

1.1 Context

1.1.1 Time series models

In probability theory and statistics, stochastic processes are used to capture uncertainty in many real-world dynamical phenomena. A stochastic process can be thought to evolve in time either continuously or discretely; in this thesis we will only consider discrete time stochastic processes. In the literature, a large number of different discrete time stochastic processes can be represented under the family of generative dynamical models called *time series models*. A parametric time series model consists of random variables that describe the modelled process with adequate generality, and these random variables admit probability laws that are parametrised by a vector-valued static variable. This variable is generally denoted by θ and called the *static parameter*, or simply the *parameter*, of the model.

A time series model associated with a stochastic process is generative. That is, when simulated, the model produces a realisation of a sequence of observable random variables $\{Y_t\}_{t\geq 1}$ of the stochastic process over time. Typically $\{Y_t\}_{t\geq 1}$ are only a subset of the random variables that comprise the time series model; the rest of the random variables are called latent, or hidden, variables. In many cases, observable variables are considered to be somewhat noisy measurements of an underlying structure which is of primary interest. The power of a time series model is its ability to provide a rigorous mathematical formulation of this underlying structure as well as its relation to $\{Y_t\}_{t\geq 1}$ via its latent variables. This helps the scientist infer the latent variables from an observed time series in a principled way by employing well-established methods from statistics.

1.1.2 Sequential inference and Monte Carlo

In many time series models, the latent variables themselves are lumped together to form another random process $\{X_t\}_{t\geq 1}$. This process represents the hidden state of interest evolving dynamically, typically in a Markovian fashion. An example of this is a *hidden Markov model* (HMM), sometimes called a *state-space model*. In a HMM, $\{X_t\}_{t\geq 1}$ is a

Markov process and each Y_t is a conditionally independent observation generated by X_t, the evolving state at time t. (For a review of HMMs in a closely related context, see Cappé et al. [2005]).

In the literature, the problem of sequential Bayesian estimation of X_t based on the sequentially observed variables $Y_1, \ldots, Y_t$ is known as the *optimum Bayesian filtering* problem. When the time series model has linear and Gaussian dynamics, the exact solution of this problem is the Kalman filtering. However, in non-linear non-Gaussian models, numerical approximations must be used. *Sequential Monte Carlo* (SMC) methods, also known as *particle filters*, are the most popular numerical methods for approximate solutions of the optimum Bayesian filtering problem [Doucet et al., 2000b; Durbin and Koopman, 2000; Gordon et al., 1993; Kitagawa, 1996; Liu and Chen, 1998]. These methods are a special class of Monte Carlo methods, which rely on the basic idea of simulating from probability distributions when analytical evaluation of quantities that involve in these probability distributions cannot be performed [Metropolis and Ulam, 1949]. Although originally developed for HMMs, SMC methods can often easily be extended to more general time series models. A review of SMC methods is presented in Section 2.5, and their application to HMMs is reviewed in Section 3.3.

1.1.3 Online parameter estimation

For the case when the true value of the static parameter of the time series model, which we will denote by θ^* throughout the thesis, is known, numerous SMC methods have been proposed and successfully applied to the Bayesian optimal filtering problem over the last two decades. (See Cappé et al. [2007]; Doucet and Johansen [2009]; Fearnhead [2008] for recent reviews of the methodology.) However, in realistic applications θ^* is hardly ever known although its estimation is essential for accurate inference of the latent variables of the model. Therefore, developing efficient and accurate parameter estimation methods for time series model is of significant importance.

Classical methods used for parameter estimation process the observed data in a *batch* fashion, i.e. they require several iterative complete browses through the entire data set. In this thesis, we are primarily concerned with developing *online parameter estimation* algorithms. With the advancement of sensor and storage technologies, and with the significantly reduced costs of data acquisition, we are able to collect and record vast amounts of raw data. Arguably, the grand challenge facing computation in the 21st century is the effective handling of such large data sets. Unfortunately, classical batch processing methods fail with very large data sets due to memory restrictions and long computational time. For this reason, so called online methods have recently gained a popularity in the area. The main principle of these methods is that, a current estimate obtained using the data available so far could be updated when a new portion of data is

received. Based on this principle, online methods are promising in terms of reducing both memory and computation requirements; hence they are potentially a powerful alternative to batch methods.

1.1.4 Bayesian estimation vs maximum likelihood estimation

There are two different approaches for static parameter estimation, which is either Bayesian or maximum likelihood. Bayesian parameter estimation requires the assignment of a prior distribution for the unknown parameter θ. The objective is then to calculate the posterior distribution of θ given the observed data. When a point estimate of θ^* is required, some feature of this posterior distribution can be provided. The common Bayesian estimators are the posterior mean, posterior median, and the posterior mode, or the maximum *a posteriori* probability (MAP) estimate. There are several Monte Carlo based methods for Bayesian parameter estimation when exact calculation of the posterior distribution is not available. Alternatively, the maximum likelihood approach regards the likelihood of the observed data, which is a function of θ, to contain all relevant information for estimating θ^*. The point estimate of θ^* is the maximising argument of the likelihood. When maximum likelihood estimation (MLE) cannot be done analytically, iterative search-based algorithms such as *expectation-maximisation* (EM) and *gradient ascent* guarantee maximising the likelihood locally given certain regularity conditions on densities of the random variables involved. Also, Monte Carlo versions of these algorithms have been developed and applied to many time series models successfully. See Kantas et al. [2009] for a comprehensive review of SMC methods for Bayesian and maximum likelihood parameter estimation, or Section 3.4 for a more brief discussion.

Whether one should in principle use the Bayesian or maximum likelihood approach for estimating θ^* is a fundamental debate which we will not go into. There are indeed cases when these two approaches do produce dramatically different suggestions on what θ^* might be, especially when the observed data is of small size and a highly informative prior for Bayesian estimation is used. However, as data size tends to infinity, the likelihood of the data sweeps away the effect of the prior in the posterior distribution and the difference between the estimates of the two approaches vanishes (say when the MAP estimate is used for Bayesian estimation), provided that the prior is well-behaved (i.e. it does not assign zero density to any 'feasible' parameter value). Therefore, in an online estimation setting, where the data size is presumably very large, the two approaches are expected to give almost identical results if they could be implemented exactly. Thus, for the practitioner, the choice of online parameter estimation method depends on which has the most favourable properties in terms of computational costs and memory requirements rather than philosophical concerns that would matter when the outcomes of the Bayesian and maximum likelihood approaches differed significantly. Moreover, when a parameter

estimation method involves any sort of Monte Carlo approximation, this brings with it the additional requirement that the statistical properties of the method, such as bias and variance of its estimator, be added to consideration.

Given these concerns, one can argue that so far online MLE methods proposed in the literature are preferable over their Bayesian counterparts. Online Bayesian estimation methods, in one way or the other, are based on including the static parameter into the hidden state of the time series model and cast the online parameter estimation problem as a filtering one. Unfortunately, when the data size is large, these methods suffer from particle degeneracy which is inherent in SMC filtering, see e.g. Andrieu et al. [2005]; Olsson et al. [2008] for a discussion. There are certain techniques proposed to overcome the degeneracy problem, such as those based on Markov chain Monte Carlo (MCMC) moves for the parameter (e.g. Gilks and Berzuini [2001]; Polson et al. [2008]) or introducing artificial dynamics on the parameter (e.g. Campillo and Rossi [2009]; Higuchi [2001]; Kitagawa [1998]). But all these techniques either still suffer from particle degeneracy problem or come with the price of bias and tuning difficulties, or both; see Section 3.4 for more discussion or Kantas et al. [2009] for even more details. On the other hand, online MLE methods based on Monte Carlo are more promising due to their favourable stability properties and reasonable computational and memory requirements. Recently, SMC based online EM and online gradient ascent algorithms for hidden Markov models have been proposed and analysed in several works such as Cappé [2009]; Del Moral et al. [2009, 2011]; Poyiadjis et al. [2011]. It has been shown in these works that the variance of the estimators in these algorithms either remain constant over time or decay, depending on their SMC schemes. For these reasons, in this thesis we focus on online MLE methods for parameter estimation in this thesis.

1.2 Scope of the thesis

SMC based MLE methods are present in the literature, and they are successfully applied to many important time series models, especially to a large proportion of HMMs. However, there are still many important types of time series models for which the developed methods so far are not directly applicable. This thesis aims to develop MLE methods, especially online MLE methods, in some non-standard time series models using Monte Carlo. Below we list and summarise the topics we investigate in this thesis.

- **Changepoint models:** One example for a time series model is a changepoint model, which is commonly used to model heterogeneity of sequential data in a range of areas such as engineering, physical and biological sciences, and finance. Having a segmented structure introduced by changepoints, the model differs from

a HMM, which makes it both interesting and challenging to study its statistical aspects and to estimate its parameters.

- **Multiple target tracking models:** Another challenging problem in the areas of applied statistics and engineering is multiple target tracking (MTT). In MTT, the main objective is to simultaneously track several moving objects in a surveillance region under far from ideal conditions that introduce random mis-detection of targets and false measurements. The additional issues of time varying unknown number of targets and unknown association of targets to observation points make the problem even more challenging. In this thesis we restrict ourselves to the linear Gaussian MTT model. Statistical treatment for the dynamics of the MTT model is widely popular and Monte Carlo methods are available for estimation of latent states in the tracking problem. However, the problem of calibrating the MTT model by estimating its static parameters has largely been ignored.

- **HMMs with intractable observation densities:** An important computational problem studied in this thesis is that of implementing batch and online MLE in a HMM where the conditional law of observations is intractable, that is, its probability density is either analytically unavailable or prohibitive to calculate. Due to this intractability, the online MLE methods developed for HMM are not directly applicable since computation of quantities required by those methods becomes impossible. Approximate Bayesian computation (ABC) has become an increasingly popular strategy for confronting intractability in many statistical models. The adaptation of ABC to HMMs resulting in an SMC-ABC scheme has recently been demonstrated. Moreover, theoretical analysis on the properties of MLE based on this SMC-ABC scheme has been performed. However, methods for implementing MLE using SMC-ABC have not been discovered yet, and we believe that solving this implementation problem would be a valuable contribution to the literature.

- **Nonnegative matrix factorisation:** Another interesting problem where online statistical estimation methods are of use is the nonnegative matrix factorisation (NMF) problem, where a given non-negative matrix Y is to be approximated as a multiplication of two nonnegative matrices as BX. In many applications, B is considered as a matrix of 'basis' vectors and X is a 'gain' matrix determining which of the columns in B dominate the columns of Y. Our approach to the NMF problem is to consider it as a parameter estimation problem for a HMM whose latent and observed processes are the columns of X and Y, respectively. This approach is useful to handle the case where the columns of Y are generated sequentially in time, such as in audio signal processing. Usually very large number of columns in Y leads to the necessity of online algorithms to learn the model and make inference.

This thesis aims to contribute to the methodology on MLE, especially online MLE, in time series models within the contexts of the topics summarised above. We present novel EM and gradient ascent methods implemented with SMC. Statistical and computational aspects of the developed methods will be studied, mostly using numerical experiments.

1.3 Outline

The material presented in the rest of the thesis is organised in six main chapters, followed by a final chapter including a conclusion, as follows.

Chapter 2: Monte Carlo Methods for Statistical Inference

This chapter provides a survey of the Monte Carlo literature. We will review some basic Monte Carlo methods, such as rejection sampling, importance sampling, MCMC, SMC, ABC, etc.

Chapter 3: Hidden Markov Models and Parameter Estimation

We introduce hidden Markov models and review Monte Carlo methods for filtering and parameter estimation in hidden Markov models.

Chapter 4: An Online Expectation-Maximisation Algorithm for Changepoint Models:

We present a novel online EM algorithm using SMC for changepoint models. We also provide theoretical and numerical stability analysis for the developed algorithm.

Chapter 5: Estimating the static parameters of the linear Gaussian Multiple Target Tracking Model:

We present novel batch and online EM algorithms using SMC for linear Gaussian MTT models. The algorithms are based on the availability of exact EM algorithms in a single linear Gaussian state-space model, but involve SMC for tracking the unknown data association inherent in the model.

Chapter 6: Approximate Bayesian Computation for Maximum Likelihood Estimation in Hidden Markov Models

We present methods for implementing MLE in HMMs with intractable observation densities. An SMC based ABC is our main tool for dealing with the intractability inherent in these HMMs and batch and online gradient ascent algorithms using SMC are shown to be suitable for this scheme.

Chapter 7: An Online Expectation-Maximisation Algorithm for Nonnegative Matrix Factorisation Models

We formulate the nonnegative matrix factorisation (NMF) problem as a MLE problem for HMMs and propose online EM algorithms using SMC to estimate the NMF and the other unknown static parameters.

1.4 Notation

It will be useful to summarise the notation used throughout this thesis. The notation presented here will be used with consistency in the literature review part of the thesis (Chapters 2 and 3); however the particular requirements of Chapters 4, 5, 6, and 7 containing original work are such that there is inevitably some conflict with the desire to be consistent with standard usage within the literature. The reader will be notified whenever we deviate from the notation or any additional notation is introduced.

We use $\mathbb{N}$ and $\mathbb{R}$ to denote the set of natural numbers and real numbers. For a sequence $\{a_k\}_{k\geq 1}$ and integers i, j, we let $a_{i:j}$ denote the set $\{a_i, \ldots, a_j\}$, which is empty if $j < i$, and $a_{i:\infty} = \{a_i, a_{i+1}, \ldots\}$.

Probability measures, integrals, and random variables: Given a general measurable space $(\mathcal{X}, \mathcal{E})$, we refer to the set of all σ-finite measures on that space as $\mathcal{M}(\mathcal{X})$. The set of all probability measures on $(\mathcal{X}, \mathcal{E})$ is denoted $\mathcal{P}(\mathcal{X}) \subset \mathcal{M}(\mathcal{X})$. We use $\mathcal{B}_b(\mathcal{X})$ to denote the Banach space of bounded real valued measurable functions on $\mathcal{X}$.

The integration of a real-valued measurable function φ on $\mathcal{X}$ with respect to the measure $\mu \in \mathcal{M}(\mathcal{X})$ is denoted as

$$\mu(\varphi) = \int_{\mathcal{X}} \varphi(x)\mu(dx), \quad \forall \mu \in \mathcal{M}(\mathcal{X}).$$

Also, for $A \in \mathcal{X}$, $\mu(A) = \mu(\mathbb{I}_A)$, where $\mathbb{I}_A : \mathcal{X} \to \{0, 1\}$ is the indicator function so that $\mathbb{I}_A(x)$ is 1 if $x \in A$ and 0 otherwise. Finally, δ_x is the Dirac measure satisfying $\delta_x(A) = \mathbb{I}_A(x)$ for all $A \in \mathcal{X}$.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(\mathcal{X}, \mathcal{E})$ be a measurable space. A $\mathcal{E}/\mathcal{F}$ measurable function $X : \Omega \to \mathcal{X}$ is called a $(\mathcal{X}, \mathcal{E})$-valued *random variable*. The probability measure π on $(\mathcal{X}, \mathcal{E})$ corresponding to the law of X is given by $\mathbb{P} \circ X^{-1}$ so that

$$\pi(A) = \mathbb{P}\left[X^{-1}(A)\right], \quad \forall A \in \mathcal{E}.$$

We will denote the expectation of φ with respect to π as

$$\mathbb{E}_\pi[\varphi(X)] = \pi(\varphi), \quad \forall \pi \in \mathcal{P}(\mathcal{X}).$$

Both expressions on the left and the right sides of the equality will be used. If π is parametrised by a vector θ, we will denote it by π_θ and we will write $\mathbb{E}_\theta[\varphi(X)]$ to mean $\mathbb{E}_{\pi_\theta}[\varphi(X)]$. Finally, capital letters X, Y, Z, etc. will be used to denote random variables; whereas for their realisations corresponding small letters x, y, z, etc. will be used.

Let π be the law of X. We write $\pi \ll \lambda$ to mean that π is absolutely continuous with respect to the dominating measure λ, and we call the Radon-Nikodým derivative $\nu = \frac{d\pi}{d\lambda}$ the density of π (or the probability density of X) with respect to λ. Throughout the chapters of this thesis containing original work, λ will be either the Lebesgue measure or the counting measure and $\lambda(dx)$ will be replaced by dx for simplicity. To make explicit the law of X, we interchangeably use $X \sim \pi$ and $X \sim \nu$.

Markov kernels: Given two measurable spaces $(\mathcal{X}_1, \mathcal{E}_1)$ and $(\mathcal{X}_2, \mathcal{E}_2)$, we define a *Markov kernel* or *transition kernel* $K : \mathcal{X}_1 \to \mathcal{P}(\mathcal{X}_2)$ satisfying the following two conditions

- $\forall x \in \mathcal{X}_1$, $K(x, \cdot)$ is a probability measure in $\mathcal{P}(\mathcal{X}_2)$,
- $\forall A \in \mathcal{E}_2$, $K(\cdot, A)$ is a nonnegative measurable function with respect to $\mathcal{E}_1$ on $\mathcal{X}_1$.

A Markov kernel induces two operators, the first one on $\mathcal{M}(\mathcal{X}_2)$ and the second one on the bounded $\mathcal{E}_2$-measurable functions on $\mathcal{X}_2$:

$$\mu K(dy) = \int_{\mathcal{X}_1} \mu(dx) K(x, dy), \quad \forall \mu \in \mathcal{M}(\mathcal{X}_1),$$
$$K(\varphi)(x) = \int_{\mathcal{X}_2} \varphi(y) K(x, dy), \quad \forall \varphi \in \mathcal{B}_b(\mathcal{X}_2).$$

Using the first operation a probability measure $\pi \in \mathcal{P}(\mathcal{X}_1)$ is mapped by K to another probability measure $\mu K \in \mathcal{P}(\mathcal{X}_2)$. Also, when we wish to consider the joint distribution induced over $(\mathcal{X}_1 \times \mathcal{X}_2, \mathcal{E}_1 \times \mathcal{E}_2)$ by a measure π and a Markov kernel $K : \mathcal{X}_1 \to \mathcal{M}(\mathcal{X}_2)$, we use the notation $\pi \otimes K$, i.e.

$$\pi \otimes K(dx, dy) = \pi(dx) K(x, dy).$$

Moreover, given a sequence of measurable spaces $\{X_n, \mathcal{E}_n\}_{n \geq 1}$ and a sequence of Markov kernels $\{K_n : \mathcal{X}_{n-1} \to \mathcal{P}(\mathcal{X}_n)\}_{n \geq 2}$,

$$K_{p:q}(x_{p-1}, dx_{p:q}) = K_{p+1} \otimes \ldots \otimes K_q(x_p, dx_{p+1:q}) = \prod_{i=p+1}^{q} K_i(x_{i-1}, dx_i). \quad q \geq p \geq 1;$$

and we define the operator on the bounded $\mathcal{E}_{p+1} \otimes \ldots \otimes \mathcal{E}_q$-measurable functions on $\mathcal{X}_2$

$$K_{p:q}(\varphi)(x_p) = \int_{\mathcal{X}_{p+1}} \cdots \int_{\mathcal{X}_q} \varphi(x_{p+1:q}) \prod_{i=p+1}^{q} K_i(x_{i-1}, dx_i), \quad \forall \varphi \in \mathcal{B}_b(\mathcal{X}_{p+1} \times \ldots \times \mathcal{X}_q).$$

Some common probability distributions: We will use $\mathcal{N}(\mu, \sigma^2)$ to describe the normal distribution with mean μ and variance σ^2; $\mathcal{U}_A$ for the uniform distribution over the set A; $\mathcal{PO}(\lambda)$ for the Poisson distribution with rate λ; $\mathcal{G}(\alpha, \beta)$ for the gamma distribution with shape α and scale β; $\mathcal{IG}(\alpha, \beta)$ for the inverse-gamma distribution with shape α and (inverse) scale β; $\mathcal{BE}(p)$ for the Bernoulli distribution with success rate p; $\mathcal{N}\Gamma^{-1}(\zeta, \kappa, \alpha, \beta)$ for the normal-inverse gamma distribution such that $(X, Y) \sim \mathcal{N}\Gamma^{-1}(\xi, \kappa, \alpha, \beta)$ means $Y \sim \mathcal{IG}(\alpha, \beta)$ and $X \sim \mathcal{N}(\xi, \frac{Y}{\kappa})$; $\mathcal{A}(\alpha, \beta, \mu, \sigma)$ for the α-stable distribution with shape α, skewness β, location μ and scale σ; $\mathcal{M}(\alpha, \rho)$ for the multinomial distribution with α number of independent trials and the probability vector ρ. We also use these notations to express the corresponding probability densities. For example, $\mathcal{N}(x; \mu, \sigma^2)$ is the probability density of the normal distribution $\mathcal{N}(\mu, \sigma^2)$ evaluated at x.

Chapter 2

Monte Carlo Methods for Statistical Inference

Summary: *This chapter provides a survey of the Monte Carlo literature. We will review some basic Monte Carlo methods for statistical inference that are related to the main content of this thesis. These methods are rejection sampling, importance sampling, Markov chain Monte Carlo, sequential Monte Carlo, and approximate Bayesian computation.*

2.1 Introduction

Assume that we are given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and some random variable $X : \Omega \to \mathcal{X}$ which is $\mathcal{E}/\mathcal{F}$ measurable. We allow the probability measure π on $(\mathcal{X}, \mathcal{E})$ to describe the law of X so that $\pi = \mathbb{P} \circ X^{-1}$. We are interested in integrating a measurable function $\varphi : \mathcal{X} \to \mathbb{R}^{d_\varphi}$ with respect to the probability measure π, i.e.

$$\pi(\varphi) = \mathbb{E}_\pi \left[\varphi(X)\right] = \int_{\mathcal{X}} \varphi(x)\pi(dx). \tag{2.1}$$

When analytical evaluation of (2.1) is not possible, we have to use approximations. There are deterministic numerical integration techniques available; however these methods encounter the problem called the *curse of dimensionality* since the amount of computation grows exponentially with the dimension of $\mathcal{X}$ [Press, 2007]. Therefore, they are far from being practical and reliable unless they work in low dimensional problems. A powerful alternative to deterministic methods for integration problems is *Monte Carlo* integration, where random samples from some distribution are used to approximate the integral in (2.1). The term Monte Carlo was coined in the 1940s, see Metropolis and Ulam [1949] for a first use of the term, and Eckhardt [1987]; Metropolis [1987] for a historical review.

In this chapter we will review the Monte Carlo methodology. We first present the main methods in the literature that aim to evaluate the integral in (2.1). We then proceed to *sequential Monte Carlo* methods to approximate a sequence of integrals like in (2.1). We conclude the chapter with a review of *approximate Bayesian computation*, which is a name

attached to a wide class of popular Monte Carlo methods aiming to tackle integrations $\pi(\varphi)$ where π is a posterior distribution resulting from an intractable likelihood. Note that we restrict ourselves to the review of only those methods which are closely related to the work in this thesis. A book length review of general Monte Carlo methods can be found in Robert and Casella [2004], and for a detailed review of sequential Monte Carlo methods one can consult the books Doucet et al. [2001] and Del Moral [2004].

2.2 Perfect Monte Carlo

The term *perfect Monte Carlo* refers to those methods in which the distribution of interest π is approximated by $N > 0$ of independent, identically distributed (i.i.d.) samples from the distribution π and integration of φ with respect to π is approximated by using this approximation. The approximation to π using N i.i.d. samples $X^{(1)}, \ldots, X^{(N)}$ is given by

$$\pi^N_{MC}(dx) := \frac{1}{N}\sum_{i=1}^{N} \delta_{X^{(i)}}(dx).$$

Then, the perfect Monte Carlo approximation to $\pi(\varphi)$ is obtained by substituting π with π^N_{MC} in (2.1) as

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

It is this approach which was originally referred to as *the* Monte Carlo method in Metropolis and Ulam [1949], although the term has come to encompass a broader class of methods through the following years.

It is easy to show that $\pi^N_{MC}(\varphi)$ is an unbiased estimator of $\pi(\varphi)$ for any $N > 0$. Also, if $\pi(\varphi)$ is finite, the *strong law of large numbers* (e.g. Shiryaev [1995], p. 391) ensures almost sure (a.s.) convergence of $\pi^N_{MC}(\varphi)$ to $\pi(\varphi)$ as the number of i.i.d. samples tends to infinity,

$$\pi^N_{MC}(\varphi) \overset{a.s.}{\to} \pi(\varphi).$$

The variance of $\pi^N(\varphi)$ is given by

$$var\left[\pi^N_{MC}(\varphi)\right] = \frac{1}{N^2}\sum_{i=1}^{N} var_\pi\left[\varphi(X^{(i)})\right] = \frac{1}{N} var_\pi\left[\varphi(X)\right].$$

which indicates the improvement in the accuracy with increasing N, provided that $var_\pi[\varphi(X)]$ is finite. Note that this is true regardless of the dimension of $\mathcal{X}$; which makes Monte Carlo preferable over the deterministic numerical methods particularly for high dimensional integrations [Newman and Barkema, 1999]. Also, if $var_\pi[\varphi(X)]$ is finite, the distribution of the estimator is well behaved in the limit, which is ensured by

the *central limit theorem* (e.g. Shiryaev [1995], p. 335)

$$\sqrt{N}\left[\pi_{MC}^{N}(\varphi) - \pi(\varphi)\right] \xrightarrow{d} \mathcal{N}\left(0, var_{\pi}\left[\varphi(X)\right]\right).$$

The requirement of perfect Monte Carlo is the ability to obtain i.i.d. samples from π. There are several methods for obtaining i.i.d. samples from distributions. We shall cover the two most common ones in the following.

2.2.1 Inversion sampling

If π is a distribution on $\mathbb{R}$, then its cumulative distribution function can be defined as

$$F_{\pi} : \mathbb{R} \to [0,1], \quad F_{\pi}(x) = \pi((-\infty, x]).$$

If it is possible to invert F_{π}, then it is possible to sample from π by transforming a uniform sample U distributed over $(0,1)$ as

$$X = F_{\pi}^{-1}(U) := \inf\{x \in \mathcal{X} : F_{\pi}(x) \geq U\}.$$

This approach was considered by Ulam prior to 1947 [Eckhardt, 1987] and some extensions to the method are provided by Robert and Casella [2004].

2.2.2 Rejection sampling

Another common method of obtaining i.i.d. samples from π is *rejection sampling*, which is available when there exists an instrumental distribution μ such that $\pi \ll \mu$ with bounded Radon-Nikodým derivative $\frac{d\pi}{d\mu}$. Rejection sampling was first mentioned in a 1947 letter by Von Neumann [Eckhardt, 1987], it was also presented a few years later in von Neumann [1951]. The method for obtaining one sample from π can be implemented with any $M \geq \sup_x \frac{d\pi}{d\mu}(x)$ by (i) generating X from μ, (ii) accepting it with probability $\frac{1}{M}\frac{d\pi}{d\mu}(X)$, and otherwise repeating steps (i) and (ii) until acceptance. Letting $A = \{U \leq \frac{1}{M}\frac{d\pi}{d\mu}(X)\}$ be the event of acceptance in a single trial, its probability is given by

$$P(A) = \mathbb{E}_{\mu}\left[\frac{1}{M}\frac{d\pi}{d\mu}(X)\right] = \frac{1}{M}\mu\left(\frac{d\pi}{d\mu}\right) = \frac{1}{M}, \tag{2.2}$$

which is also the long term proportion of the number accepted samples over the number of trials. Therefore, taking μ as close to π as possible to avoid large Radon-Nikodým derivatives and taking $M = \sup_x \frac{d\pi}{d\mu}(x)$ are sensible choices to make the acceptance probability $P(A)$ as high as possible.

Algorithm 2.1. ***Rejection sampling:*** *Choose* $M \geq \sup_x \frac{d\pi}{d\mu}(x)$. *To generate a single sample,*

1. *Generate* $X \sim \mu$ *and* $U \sim \text{Unif}(0, 1)$.
2. *If* $U \leq \frac{1}{M}\frac{d\pi}{d\mu}(X)$, *accept* X; *else go to 1.*

The rejection sampling algorithm is given in Algorithm 2.1. The validity of this algorithm can be verified by considering the distribution of the accepted samples. Using Bayes' theorem,

$$P\left(X \in dx \,|A\right) = \frac{\mu(dx)P\left(A|x\right)}{P(A)} = \mu(dx)\frac{1}{M}\frac{d\pi}{d\mu}(x)/\frac{1}{M} = \pi(dx). \tag{2.3}$$

One advantage of rejection sampling is that we can implement it even when we know π and μ only up to some proportionality constants Z_π and Z_μ, that is, when $\pi = \frac{\widehat{\pi}}{Z_\pi}, \mu = \frac{\widehat{\mu}}{Z_\mu}$ and we only know $\widehat{\pi}$ and $\widehat{\mu}$. It is easy to check that one can perform the steps (i) and (ii) of rejection sampling method for any $M \geq \sup_x \frac{d\widehat{\pi}}{d\widehat{\mu}}(x)$ using $\frac{d\widehat{\pi}}{d\widehat{\mu}}$ instead of $\frac{d\pi}{d\mu}$, and justification of this modification would follow from similar steps to those in (2.3). Also, in that case, the acceptance probability would be $\frac{1}{M}\frac{Z_\pi}{Z_\mu}$. Finally, when π and μ have densities (denoted as π and μ also) with respect to a common dominating measure, then the Radon-Nikodým derivative $\frac{d\pi}{d\mu}(x)$ becomes equal to $\frac{\pi(x)}{\mu(x)}$.

The drawback of rejection sampling is that in practice a rejection based procedure is usually not viable when $\mathcal{X}$ is high-dimensional, since $P(A)$ gets smaller and more computation is required to evaluate acceptance probabilities as the dimension increases. In the literature there exist approaches to improve the computational efficiency of rejection sampling. For example, assuming the densities exist, when it is difficult to compute $\pi(x)$, tests like $u \leq \frac{1}{M}\frac{\pi(x)}{\mu(x)}$ can be slow to evaluate. In this case, one may use a squeezing function $s : \mathcal{X} \to [0, \infty)$ such that $\frac{s(x)}{\mu(x)}$ is cheap to evaluate and $\frac{s(x)}{\pi(x)}$ is tightly bounded from above by 1. For such an s, not only $u \leq \frac{1}{M}\frac{s(x)}{\mu(x)}$ would guarantee $u \leq \frac{1}{M}\frac{\pi(x)}{\mu(x)}$, hence acceptance, but also if $u \leq \frac{1}{M}\frac{\pi(x)}{\mu(x)}$ then $u \leq \frac{1}{M}\frac{s(x)}{\mu(x)}$ would hold with a high probability. Therefore, in case of acceptance evaluation of $\frac{\pi(x)}{\mu(x)}$ would largely be avoided by checking $u \leq \frac{1}{M}\frac{s(x)}{\mu(x)}$ first. In Marsaglia [1977], the author proposed to squeeze π from above and below by μ and s respectively, where μ is easy to sample from and s is easy to evaluate. There are also adaptive methods to squeeze π from both below and above; they involve an adaptive scheme to gradually modify μ and s from the samples that have already been obtained [Gilks, 1992; Gilks et al., 1995; Gilks and Wild, 1992].

2.3 Importance sampling

We saw that rejection sampling can be wasteful as it uses only about $1/M$ of generated random samples to construct an approximation to π. In contrast, *importance sampling* uses every sample but weights each one according to the degree of similarity between the target and instrumental distributions. The idea of importance sampling follows from the *importance sampling fundamental identity* [Robert and Casella, 2004]: if there is a probability measure μ such that $\pi \ll \mu$ with the Radon-Nikodým derivative $w = \frac{d\pi}{d\mu}$, then we have

$$\pi(\varphi) = \mu(\varphi w).$$

This identity can be used with a μ which is easy to sample from. Sampling $X^{(1)}, \ldots, X^{(N)}$ from μ, the integral $\pi(\varphi) = \mu(\varphi w)$ can be approximated by using perfect Monte Carlo as

$$\pi_{IS}^N(\varphi) := \frac{1}{N}\sum_{i=1}^{N} \varphi(X^{(i)})w(X^{(i)}). \tag{2.4}$$

Algorithm 2.2. ***Importance sampling:***

- *For $i = 1, \ldots, N$; generate $X^{(i)} \sim \mu$, calculate $w(X^{(i)}) = \frac{d\pi}{d\mu}(X^{(i)})$.*
- *Set $\pi_{IS}^N(\varphi) = \frac{1}{N}\sum_{i=1}^{N} w(X^{(i)})\varphi(X^{(i)})$.*

The importance sampling is summarised in Algorithm 2.2. The Radon-Nikodým derivatives $w(X^{(i)})$ are known as the *importance sampling weights*. Noting its equivalence to perfect Monte Carlo for $\mu(\varphi w)$, the estimator in (2.4) is unbiased and justified by the strong law of large numbers and the central limit theorem, provided that $\pi(\varphi)$ and $var_\mu[w(X)\varphi(X)]$ are finite. Moreover, as we have freedom to choose μ we can control the variance of importance sampling [Robert and Casella, 2004]

$$\begin{aligned} var\left[\pi_{IS}^N(\varphi)\right] &= \frac{1}{N} var_\mu[w(X)\varphi(X)] \\ &= \frac{1}{N}\left(\mu(w^2\varphi^2) - [\mu(w\varphi)]^2\right) \\ &= \frac{1}{N}\left(\mu(w^2\varphi^2) - [\pi(\varphi)]^2\right). \end{aligned}$$

Therefore, minimising $var\left[\pi_{IS}^N(\varphi)\right]$ is equivalent to minimising $\mu(w^2\varphi^2)$, which can be lower bounded as

$$\mu(w^2\varphi^2) \geq [\mu(w|\varphi|)]^2 = [\pi(|\varphi|)]^2$$

using the Jensen's inequality. Considering $\mu(w^2\varphi^2) = \pi(w\varphi^2)$, this bound is attainable if

we choose μ such that it satisfies

$$w(x) = \frac{d\pi}{d\mu}(x) = \frac{\pi(|\varphi|)}{|\varphi(x)|}, \quad x \in \mathcal{X}, \varphi(x) \neq 0.$$

This results in the optimum choice of μ to be

$$\mu(dx) = \pi(dx)\frac{|\varphi(x)|}{\pi(|\varphi|)}$$

for points $x \in \mathcal{X}$ such that $\varphi(x) \neq 0$, and the resulting minimum variance is given by

$$\min_{\mu} var\left[\pi_{IS}^{N}(\varphi)\right] = \frac{1}{N}\left([\pi(|\varphi|)]^2 - [\pi(\varphi)]^2\right).$$

Note that this minimum value is 0 if φ is nonnegative π-almost everywhere. Therefore, importance sampling in principle can achieve a lower variance than perfect Monte Carlo. Of course, if we can not already compute $\pi(\varphi)$, it is unlikely that we can compute $\pi(|\varphi|)$. Also, it will be rare that we can easily simulate from the optimal μ even if we can construct it. Instead, we are guided to seek a μ close to the optimal one, but from which it is easy to sample.

2.3.1 Self-normalised importance sampling

Like rejection sampling, the importance sampling method is available also when $\pi = \frac{\widehat{\pi}}{Z_\pi}, \mu = \frac{\widehat{\mu}}{Z_\mu}$ and we only have $\widehat{\pi}$ and $\widehat{\mu}$. This time, letting $w = \frac{d\widehat{\pi}}{d\widehat{\mu}}$ we write the importance sampling fundamental identity in terms of $\widehat{\pi}$ and $\widehat{\mu}$ as

$$\pi(\varphi) = \frac{\mu(\varphi w)}{Z_\pi / Z_\mu} = \frac{\mu(\varphi w)}{\mu(w)}.$$

The importance sampling method can be modified to approximate both the nominator (the unnormalised estimate) and the denominator (the normalisation constant) by using perfect Monte Carlo. Sampling $X^{(1)}, \ldots, X^{(N)}$ from μ, we have the approximation

$$\pi_{IS}^{N}(\varphi) = \frac{\frac{1}{N}\sum_{i=1}^{N}\varphi(X^{(i)})w(X^{(i)})}{\frac{1}{N}\sum_{i=1}^{N}w(X^{(i)})} = \sum_{i=1}^{N} W^{(i)}\varphi(X^{(i)}).$$

where $W^{(i)} = \frac{w(X^{(i)})}{\sum_{j=1}^{N} w(X^{(j)})}$ are called the *normalised importance weights* as they sum up to 1. Being the ratio of two unbiased estimators, estimator of the self-normalised importance sampling is biased for finite N. However, its consistency and stability are provided by a strong law of large numbers and a central limit theorem in Geweke [1989]. In the same work, the variance of the self normalised importance sampling estimator is analysed and

an approximation is provided, from which it reveals that it can provide lower variance estimates than the unnormalised importance sampling method. Therefore, this method can be preferable to its unnormalised version even if it is not the case that π and μ are known only up to proportionality constants.

Algorithm 2.3. ***Self-normalised importance sampling:***

- *For* $i = 1, \ldots, N$*; generate* $X^{(i)} \sim \mu$*, calculate* $w(X^{(i)}) = \frac{d\widehat{\pi}}{d\widehat{\mu}}(X^{(i)})$.
- *For* $i = 1, \ldots, N$*; set* $W^{(i)} = \frac{w(X^{(i)})}{\sum_{j=1}^{N} w(X^{(j)})}$.
- *Set* $\pi_{IS}^{N}(\varphi) = \sum_{i=1}^{N} W^{(i)} \varphi(X^{(i)})$.

Self-normalised importance sampling is also called Bayesian importance sampling in Geweke [1989], since in most Bayesian inference problems the normalising constant of posterior distribution is unknown.

One approximation to the variance of the self-normalised importance sampling estimator is proposed in Kong et al. [1994] to be

$$\begin{aligned} var\left[\pi_{IS}^{N}(\varphi)\right] &\approx \frac{1}{N} var_{\pi}\left[\varphi(X)\right]\left\{1 + var_{\mu}\left[w(X)\right]\right\} \\ &= var\left[\pi_{MC}^{N}(\varphi)\right]\left\{1 + var_{\mu}\left[w(X)\right]\right\}. \end{aligned}$$

This approximation might be confusing at the first instance since it suggests that the variance of self-normalised importance sampling is always greater than that of perfect Monte Carlo, which we have just seen is not the case. However, it is useful as it provides an easy way of monitoring the efficiency of the method. Consider the ratio of variances of the self-normalised importance sampling method with N particles and perfect Monte Carlo with N' particles, which is given according to this approximation by

$$\frac{var\left[\pi_{IS}^{N}(\varphi)\right]}{var\left[\pi_{MC}^{N'}(\varphi)\right]} \approx \frac{N'}{N}\{1 + var_{\mu}\left[w(X)\right]\}.$$

The number N' for which this ratio is 1 would suggest how many samples for perfect Monte Carlo would be equivalent to N samples for self-normalised importance sampling. For this reason this number is defined as the *effective sample size* [Kong et al., 1994; Liu, 1996] and it is given by

$$N_{\mathit{eff}} = \frac{N}{1 + var_{\mu}\left[w(X)\right]}.$$

Obviously, the term $var_{\mu}\left[w(X)\right]$ itself is usually estimated using the samples $X^{(1)}, \ldots, X^{(N)}$ with weights $w(X^{(i)}), \ldots, w(X^{(N)})$ obtained from the method.

2.4 Markov chain Monte Carlo

We have already discussed the difficulties of generating a large number of i.i.d. samples from π. One alternative was importance sampling which involved weighting every generated sample in order not to waste it, but it has its own drawbacks mostly due to issues related to controlling variance. Another alternative is to use *Markov chain Monte Carlo* (MCMC) methods [Gilks et al., 1996; Hastings, 1970; Metropolis et al., 1953; Robert and Casella, 2004]. These methods are based on design of a suitable ergodic Markov chain whose stationary distribution is π. The idea is that if one simulates such a Markov chain, after a long enough time the samples of the Markov chain will admit π. Although the samples generated from the Markov chain are not i.i.d., their use is justified by convergence results for dependent random variables in the literature. First examples of MCMC can be found in Metropolis et al. [1953]; Hastings [1970], and book length reviews are available in Gilks et al. [1996]; Robert and Casella [2004].

2.4.1 Discrete time Markov chains

In order to adequately summarise the MCMC methodology, we first need reference to the theory of discrete time Markov chains defined on general state spaces. Discrete time Markov chains also constitute an important part of this thesis. The review made here is limited by the relation of Markov chains to the topics of this thesis; for more details one can see Meyn and Tweedie [2009] or Shiryaev [1995]; a more related introduction to our area of interest in this this thesis is present in Robert and Casella [2004, Chapter 6] and Cappé et al. [2005, Chapter 14], Tierney [1994] and Gilks et al. [1996, Chapter 4].

Definition 2.1 (**Markov chain**). *Consider a sequence of measurable spaces* $\{\mathcal{X}_n, \mathcal{E}_n\}_{n\geq 1}$, *an initial distribution* η *and a sequence of Markov kernels* $\{M_n\}_{n\geq 2}$ *with each* $M_n : \mathcal{X}_{n-1} \to \mathcal{P}(\mathcal{X}_n)$, *where* $\mathcal{P}(\mathcal{X}_n)$ *denotes the set of probability measures on* $\mathcal{X}_n$. *Then, there exists a unique stochastic process* $\{X_n\}_{n\geq 1}$ *on the canonical space* $(\prod_{n=1}^{\infty} \mathcal{X}_n, \otimes_{n=1}^{\infty} \mathcal{E}_n)$ *and admits the following probability law* $\mathbb{P}_\eta$ *on* $\mathcal{F}$ *which is defined from the initial distribution* η *and the Markov kernels* $\{M_n\}_{n\geq 2}$ *by finite dimensional distributions as*

$$\mathbb{P}_\eta(X_1 \in A_1, X_2 \in A_2, \ldots, X_n \in A_n) = \int_{A_1} \int_{A_2} \ldots \int_{A_n} \eta(dx_1) M_2(x_1, dx_2) \ldots M_n(x_{n-1}, dx_n)$$

for all n *and* $\mathcal{E}_i$*-measurable* A_i, $i = 1, \ldots, n$.

This is the canonical definition of a Markov chain, which leads to the defining property of a Markov chain which is that the current state of the chain at time n depends only on the previous state at time $n-1$. More explicitly, for any n and $\mathcal{E}_n$-measurable set A, we

have

$$\begin{aligned}\mathbb{P}_\eta(X_n \in A | X_{1:n-1} = x_{1:n-1}) &= \mathbb{P}_\eta(X_n \in A | X_{n-1} = x_{n-1}) \\ &= M_n(x_{n-1}, A).\end{aligned}$$

This property is also referred to as the *weak Markov property*, which can be stated in a more general sense:

Proposition 2.1 (**weak Markov property**). *Given $X_1 = x_1, \ldots, X_m = x_m$, the process $\{X_{m+n}\}_{n\geq 0}$ is a Markov chain independent from $X_1, \ldots, X_m$ whose probability law is constructed from the initial distribution δ_{x_m} and the sequence of Markov kernels $\{M_{m+n}\}_{n\geq 1}$ in the same way as in Definition 2.1.*

From now on, we will consider time-homogenous Markov chains where $(\mathcal{X}_n, \mathcal{E}_n) = (\mathcal{X}, \mathcal{E})$ for all $n \geq 1$ and $M_n = M$ for all $n \geq 2$, and we will denote them as $Markov(\eta, M)$. Such Markov chains are sufficient for the purposes of considering MCMC methods and also the other methods investigated throughout this thesis.

For MCMC, we require the Markov chain to have a unique stationary distribution π and to converge to π. Before that we need to review some fundamental properties a of discrete time Markov chain to understand when stationarity and convergence are ensured.

Irreducibility: Informally, a Markov chain is *irreducible* if (almost) all its states communicate, that is, it is with a positive probability that the chain travels from any point in $\mathcal{X}$ to any set in $\mathcal{E}$. For discrete $\mathcal{X}$ it is possible to state this as

$$\forall x, x' \in \mathcal{X}, \exists n \geq 1 \text{ s.t. } \mathbb{P}_{\delta_x}(X_n = x') > 0.$$

For general state-spaces, we need to generalise the concept of irreducibility.

Definition 2.2 (**ϕ-irreducibility**). *The transition kernel M, or the Markov chain $\{X_n\}_{n\geq 1}$ with transition kernel M, is said to be ϕ-irreducible if there exists a measure ϕ on $(\mathcal{X}, \mathcal{E})$ such that for any $A \in \mathcal{E}$ with $\phi(A) > 0$, we have*

$$\forall x \in \mathcal{X}, \exists n \geq 1 \text{ s.t. } \mathbb{P}_{\delta_x}(X_n \in A) > 0.$$

Such a measure ϕ is called an irreducibility measure for M.

Recurrence and Transience: In the discrete state-space case, we say that a Markov chain is *recurrent* if every of its states is expected to be visited by the chain infinitely often, otherwise it is *transient*. In the general state-space case, instead of states we consider *accessible sets*. A set $A \in \mathcal{E}$ is accessible if $\mathbb{P}_{\delta_x}(X_n \in A \text{ for some } n) > 0$ for all

$x \in \mathcal{X}$. It is also useful to consider stronger recurrence properties, expressed in terms of return probabilities rather than expected number of visits.

Definition 2.3 (**recurrence**). *Let A be a set in $\mathcal{E}$. We say A is recurrent if for all $x \in A$*

$$\mathbb{E}_x \left[\sum_{n=1}^{\infty} \mathbb{I}_A(X_n) \right] = \infty.$$

Moreover, we say A is Harris recurrent if for all $x \in A$

$$\mathbb{P}_{\delta_x} \left(\sum_{n=1}^{\infty} \mathbb{I}_A(X_n) = \infty \right) = 1.$$

Finally, we say a ϕ-irreducible Markov chain is recurrent (Harris recurrent) if every accessible $A \in \mathcal{E}$ is recurrent (Harris recurrent).

Invariant measures: We call a σ-finite measure μ M-invariant if $\mu = \mu M$. If a *M-invariant* μ is a probability measure then μ is referred to as stationary. A Markov chain associated with a ϕ-irreducible M is called *positive* if there is a probability measure μ which is M-invariant. In order to state the conditions for existence of a unique invariant probability measure for a Markov chain, we need the definition of a *small set*.

Definition 2.4 (**small set**). *Let M and ν be a transition kernel and a probability measure, respectively, on $(\mathcal{X}, \mathcal{E})$, integer $m \geq 2$ and constant $\epsilon \in (0, 1]$. A set $C \in \mathcal{E}$ is called a (m, ϵ, ν)-small set for M, or simply a small set, if for all $x \in C$ and $A \in \mathcal{E}$,*

$$\mathbb{P}_{\delta_x}(X_m \in A) \geq \epsilon \nu(A).$$

Trivially, every point in $\mathcal{X}$ is a small set, so in discrete $\mathcal{X}$ every state is a small set. Now, we have the following theorem for the existence and uniqueness of an invariant probability measure.

Theorem 2.1. *Given a Markov kernel M and a Markov chain associated to M, the following hold*

- *M is ϕ-irreducible and recurrent if and only if it admits a unique (up to a multiplicative constant) invariant measure.*
- *If M admits an accessible small set C such that*

$$\sup_{x \in C} \mathbb{E}_{\mathbb{P}_{\delta_x}} \left[\inf\{n \geq 2 : X_n \in C\} \right] < \infty, \tag{2.5}$$

then the Markov chain is positive.

Note that while ϕ-irreducibility and recurrence ensure a unique (up to a multiplicative constant) invariant measure, existence of an accessible small set is required as well to have an invariant probability measure. In fact, the condition (2.5) is equivalent to the property of positive recurrence for Markov chains with discrete state-space which is necessary for the existence of a unique stationary distribution.

Reversibility and detailed balance: One useful way to verify the existence of an invariant probability measure for a Markov chain is to check for its *reversibility*, which is a sufficient (but not necessary) condition for existence of a stationary distribution.

Definition 2.5 (**reversibility**). *Let M be a transitional kernel having a stationary distribution and assume the associated Markov chain is started from π. We say that M is reversible if the reversed process $\{X_m = X_{n-m+1}\}_{1\leq m\leq n}$ is also $Markov(\pi, M)$ for all $n \geq 1$.*

A necessary and sufficient condition for reversibility of M is the detailed balance condition.

Proposition 2.2 (**detailed balance**). *We say a Markov kernel M is reversible with respect to a probability measure π if and only if the following condition, known as the detailed balance condition, holds: for all bounded measurable functions f on $\mathcal{X} \times \mathcal{X}$*

$$\int_{\mathcal{X}\times\mathcal{X}} f(x,y)\mu(dx)M(x,dy) = \int_{\mathcal{X}\times\mathcal{X}} f(x,y)\mu(dy)M(y,dx).$$

Also, then π is a stationary distribution for M.

Being a sufficient condition for stationarity, the detailed balance condition is quite useful for designing transition kernels for MCMC algorithms.

Ergodicity: We have shown the conditions for a unique stationary distribution of a Markov chain. The first ergodic theorem shows that these conditions are sufficient for establishing a strong law of large numbers.

Theorem 2.2. *If $\{X_n\}_{n\geq 1}$ is a positive, Harris recurrent Markov chain with invariant distribution π, then for all π-integrable functions φ,*

$$\frac{1}{n}\sum_{i=1}^{n} \varphi(X_i) \overset{a.s}{\to} \pi(\varphi).$$

Note that this ergodic theorem is about the convergence of the sample mean and it does not tell whether the chain will converge to its stationary distribution. For that to happen the Markov chain is required to be *aperiodic*, a property which restricts the chain

from getting trapped in *cycles*. In discrete state-space a cycle is defined as the greatest common divisor of the lengths of all routes of positive probability between two states, and if there exists no cycles of length greater than one, the chain is said to be aperiodic. In general state-spaces, a more detailed care is required to define a cycle. It is a theorem that there exists a (m, ϵ, ν)-small C set for a ϕ-irreducible Markov chain, which enables the following definition.

Definition 2.6 (**cycle and period**). *A ϕ-irreducible Markov chain associated to the Markov kernel M has a cycle of length d if for some accessible (m, ϵ, ν)-small set C d is the greatest common divisor of*

$$\{n - 1 : n \geq 2 : C \text{ is } (n, \epsilon_n, \nu_n)\text{-small for some } \epsilon_n > 0, \nu_n \in \mathcal{P}(\mathcal{X})\}$$

The period of the Markov chain is the largest possible cycle d for M. When the period is 1, the chain is called aperiodic.

We are now ready to state our second ergodic theorem which requires the ergodicity of the Markov chain.

Theorem 2.3. *If $\{X_n\}_{n\geq 1}$ is a positive, and aperiodic Markov chain with stationary distribution π, then for π-almost every $x \in \mathcal{X}$, and all sets $A \in \mathcal{E}$,*

$$\sup_{A\in\mathcal{E}} |\mathbb{P}_{\delta_x}(X_n \in A) - \pi(A)| \overset{a.s.}{\to} 0. \tag{2.6}$$

Moreover, if the chain is Harris recurrent with stationary distribution π, the above holds for every $x \in \mathcal{X}$

We call a chain *ergodic* if it satisfies (2.6) for π-almost all $x \in \mathcal{X}$; if (2.6) is satisfied for all $x \in \mathcal{X}$ then the chain is called *uniformly ergodic*. Hence we can define ergodicity in terms of the properties of the Markov chain.

Definition 2.7 (**ergodic Markov chain**). *A ϕ-irreducible Markov chain is called ergodic if it is positive and aperiodic; it is called uniformly ergodic if it is also Harris recurrent.*

2.4.2 Metropolis-Hastings

As previously stated, an MCMC method is based on a discrete-time Markov chain which has its stationary distribution as π. The most widely used MCMC algorithm up to date is the *Metropolis-Hastings* algorithm [Hastings, 1970; Metropolis et al., 1953]. In this algorithm, given the previous sample X_{n-1} a new value Y for X_n is proposed using an instrumental transitional kernel $K : \mathcal{X} \to \mathcal{P}(\mathcal{E})$. We assume for simplicity that the product measure $\pi(dx)K(x, dy)$ has a probability density $q(x, y)$ with respect to

a dominating symmetric measure $\zeta(dx, dy)$ (a situation where this is not the case will be visited in Section 2.4.3). The proposed sample Y is accepted with the acceptance probability $\alpha(X_{n-1}, Y)$, where the function $\alpha : \mathcal{X} \times \mathcal{X} \to [0, 1]$ is defined as

$$\alpha(x, y) = \min\left\{1, \frac{q(y, x)}{q(x, y)}\right\}, \quad x, y \in \mathcal{X}.$$

Algorithm 2.4. ***Metropolis-Hastings:*** *Begin with some $X_1 \in \mathcal{X}$. For $n = 2, 3, \ldots$*

- *Sample $Y \sim K(X_{n-1}, \cdot)$.*
- *Set $X_n = Y$ with probability $\alpha(X_{n-1}, Y)$; otherwise set $X_n = X_{n-1}$.*

According to Algorithm 2.4, the transition kernel M of the Markov chain from which the samples are obtained is such that for any bounded measurable function f defined on $\mathcal{X}$

$$M(x, f) = \int_{\mathcal{X}} K(x, dy)\alpha(x, y)f(y) + \left[1 - \int_{\mathcal{X}} K(x, dy)\alpha(x, y)\right] f(x).$$

where we can simplify the expression by substituting $p_r(x) = 1 - \int_{\mathcal{X}} K(x, dy)\alpha(x, y)$, the rejection probability of a proposed sample from $K(x, \cdot)$. We can check for the detailed balance condition to see why this Markov chain has π as its stationary distribution. For a bounded measurable f on $\mathcal{X} \times \mathcal{X}$, we have

$$\begin{aligned}\int_{\mathcal{X}\times\mathcal{X}} \pi(dx)M(x, dy)f(x, y) &= \int_{\mathcal{X}\times\mathcal{X}} q(x, y)\alpha(x, y)f(x, y)\zeta(dx, dy) + \int_{\mathcal{X}} \pi(dx)p_r(x)f(x, x) \\ &= \int_{\mathcal{X}\times\mathcal{X}} \min\{q(x, y), q(y, x)\}f(x, y)\zeta(dx, dy) + \pi(p_r g).\end{aligned}$$

where the function $g : \mathcal{X} \to \mathbb{R}$ satisfies $g(x) = f(x, x)$. Since the measure $\zeta(dx, dy)$ and the expression $\min\{q(x, y), q(y, x)\}$ are symmetric in (x, y), we can swap x and y in $f(x, y)$ in the last line, hence in the first line. This results the detailed balance condition being satisfied for M with π. Note that existence of π for M ensures the recurrence of M, and fortunately it is rare that a recurrent M is not Harris recurrent. There are also various sufficient conditions for the M in the Metropolis-Hastings algorithm to be ϕ-irreducible and aperiodic. For example, if K is π-irreducible and $\alpha(x, y) > 0$ for all $x, y \in \mathcal{X}$ then M is π-irreducible; if $P(X_n = X_{n-1}) > 0$ or K is aperiodic then M is aperiodic [Roberts and Smith, 1994]. More detailed results on the convergence of Metropolis-Hastings are also available, see e.g. Tierney [1994], Roberts and Tweedie [1996], and Mengersen and Tweedie [1996].

Historically, the original MCMC algorithm was introduced by Metropolis et al. [1953] for the purpose of optimisation on a discrete state-space. This algorithm, called the Metropolis algorithm, used symmetrical proposal kernels K. The Metropolis algorithm was later generalised by Hastings [1970] such that it permitted continuous state-spaces

and asymmetrical proposal kernels, preserving the Metropolis algorithm as a special case, and its use for statistical simulation was shown. A more historical survey is provided by Hitchcock [2003].

2.4.3 Gibbs sampling

The *Gibbs sampler* [Gelfand and Smith, 1990; Geman and Geman, 1984] is one of the most popular MCMC methods, which can be used when X has more than one dimension. If X has $d > 1$ components (of possibly different dimensions) such that $X = (X_1, \ldots, X_d)$, and one can sample from each of the full conditional distributions $\pi_k(\cdot|X_{1:k-1}, X_{k+1:d})$, then the Gibbs sampler produces a Markov chain by updating one component at a time using π_k's. One cycle of the Gibbs sampler successively samples from the conditional distributions $\pi_1, \ldots, \pi_d$ by conditioning on the most recent samples.

Algorithm 2.5. ***The Gibbs sampler:*** *Begin with some $X_1 \in \mathcal{X}$. For $n = 2, 3, \ldots$, generate for $k = 1, \ldots, d$*

$$X_{n,k} \sim \pi_i(\cdot|X_{n-1,1:k-1}, X_{n-1,k+1:d}).$$

For an $x \in \mathcal{X}$, let $x_{-k} = (x_{1:k-1}, x_{k+1:d})$ for $k = 1, \ldots, d$ denotes the components of x excluding x_k, and let us permit ourselves to write $x = (x_k, x_{-k})$. The corresponding MCMC kernel of the Gibbs sampler can be written as $M = M_1 M_2 \ldots M_d$, where each transition Kernel $M_k : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ for $k = 1, \ldots, d$ can be written as

$$M_k(x, dy) = \pi_k(dy_k|x_{-k})\delta_{x_{-k}}(dy_{-k})$$

The justification of the transitional kernel comes from the reversibility of each M_k with respect to π, which can be verified from the detailed balance condition as follows. For any bounded measurable function f on $\mathcal{X} \times \mathcal{X}$,

$$\begin{aligned}
\int \pi(dx)M_k(x, dy)f(x, y) &= \int \pi(dx)\pi_k(dy_k|x_{-k})\delta_{x_{-k}}(dy_{-k})f(x_k, x_{-k}, y_k, y_{-k}) \\
&= \int \pi(dx_{-k})\pi_k(dx_k|x_{-k})\pi_k(dy_k|x_{-k})f(x_k, x_{-k}, y_k, x_{-k}) \\
&= \int \pi(dy)\pi_k(dx_k|y_{-k})f(x_k, y_{-k}, y_k, y_{-k}) \\
&= \int \pi(dy)\pi_k(dx_k|y_{-k})\delta_{x_{-k}}(dy_{-k})f(x_k, x_{-k}, y_k, y_{-k}) \\
&= \int \pi(dy)M_k(y, dx)f(x, y), \qquad (2.7)
\end{aligned}$$

hence the detailed balance condition for M_k is satisfied with π. This leads to $\pi M_k = \pi$, hence $\pi M = \pi$, so π is indeed stationary for the Gibbs sampler. An insightful interpre-

tation of (2.7) is that each step of a cycle of the Gibbs sampler is a Metropolis-Hastings move whose MCMC kernel M is equal to its proposal kernel K i.e. the $\alpha(x, y) = 1$ uniformly. This also shows that the assumption that $\pi(dx)K(x, dy)$ has a density with respect to a symmetric measure $\zeta(x, y)$ is not a necessary condition for the Metropolis-Hastings algorithm. However, reversibility of each M_k with respect to π does not suffice to establish proper convergence of the Gibbs sampler, as none of the individual steps produces a ϕ-irreducible chain. Only the combination of the d moves in the complete cycle has a chance of producing a ϕ-irreducible chain. We refer to Roberts and Smith [1994] for some simple conditions for convergence of the classical Gibbs sampler. Note, also, that M is not reversible either, although this is not a necessary condition for convergence. A way of guaranteeing both ϕ-irreducibility and reversibility is to use a mixture of kernels

$$M_\beta = \sum_{k=1}^{d} \beta_k M_k, \quad \beta_k > 0, \quad k = 1, \ldots, d, \quad \sum_{k=1}^{d} \beta_k = 1.$$

provided that at least one M_k is irreducible and aperiodic. This choice of kernel leads to the *random scan Gibbs sampler algorithm*. We refer to Tierney [1994], Roberts and Tweedie [1996], and Robert and Casella [2004] for more detailed convergence results pertaining to these variants of the Gibbs sampler.

Having attractive computational properties, the Gibbs sampler is widely used. The requirement for easy-to-sample conditional distributions is the main restriction for the Gibbs sampler. Fortunately, though, replacing the exact simulation by a Metropolis-Hastings step in a general MCMC algorithm does not violate its validity as long as the Metropolis-Hastings step is associated with the correct stationary distribution. The most natural alternative to the Gibbs move in step k where sampling from the full conditional distribution $\pi_k(\cdot|x_{-k})$ is not directly feasible is to use one-step Metropolis-Hastings move that updates x_k by using a Metropolis-Hastings kernel $M : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ such that $\pi_k(\cdot|x_{-k})$ is M-invariant [Tierney, 1994].

2.5 Sequential Monte Carlo

Despite their versatility and success, it might be impractical to apply MCMC algorithms to sequential inference problems. This section discusses *sequential Monte Carlo* (SMC) methods, that can provide with approximation tools for a sequence of varying distributions. Good tutorials on the subject are available, see for example Doucet et al. [2000b] and Doucet et al. [2001] for a book length review. Also, Robert and Casella [2004] and Cappé et al. [2005] contain detailed summaries. Finally, the book Del Moral [2004] contains a more theoretical work on the subject in a more general framework, namely Feynman-Kac formulae.

2.5.1 Sequential importance sampling

Let $\{X_n\}_{n\geq 1}$ be a sequence of random variables where each X_n takes values at some measurable space $(\mathcal{X}_n, \mathcal{E}_n)$. Define the sequence of distributions $\{\pi_n\}_{n\geq 1}$ defined on the measurable space $(\mathscr{X}_n = \prod_{i=1}^n \mathcal{X}_i, \mathscr{E}_n = \otimes_{i=1}^n \mathcal{E}_i)$. Also, let $\{\varphi_n\}_{n\geq 1}$ be a sequence of functions where $\varphi_n : \mathscr{X}_n \to \mathbb{R}$ is a π_n-measurable real-valued function on $\mathscr{X}_n$. We are interested in sequential inference, i.e. approximating the following integrals sequentially in n

$$\pi_n(\varphi_n) = \mathbb{E}_{\pi_n}\left[\varphi_n(X_{1:n})\right], \quad n = 1, 2, \ldots$$

The first method which is usually considered a SMC method is *sequential importance sampling* (SIS), which is a sequential version of the importance sampling. First use of SIS can be recognised in works back in 1960s and 1970s such as Mayne [1966], Handschin and Mayne [1969], and Handschin [1970]; see Doucet et al. [2000b] for a general formulation of the method for Bayesian filtering. Consider the naive importance sampling approach to the sequential problem where we have a sequence of importance measures $\{q_n\}_{n\geq 1}$ with each q_n is a measure defined on $(\mathscr{X}_n, \mathscr{E}_n)$ such that $\pi_n \ll q_n$ with Radon-Nikodým derivative $w_n = \frac{d\pi_n}{dq_n}$. It is obvious that we can approximate $\pi_n(\varphi_n)$ by generating samples from q_n independently of samples generated from $q_1, \ldots, q_{n-1}$ and exploiting the relation

$$\pi_n(\varphi_n) = q_n\left(w_n \varphi_n\right).$$

This approach would require the design of a separate q_n and sampling the whole path $X_{1:n}$ at each n, which is obviously inefficient. An efficient alternative to this approach is SIS which can be used when it is possible to choose q_n to have the form

$$q_n(dx_{1:n}) = q_1(dx_1) \prod_{i=1}^{n} Q_i(dx_{1:i-1}, x_i), \tag{2.8}$$

where $Q_n : \mathcal{X}_{1:n-1} \to \mathcal{P}(\mathcal{E}_n)$ are some transitional kernels which are possible to sample from. This selection of q_n leads to the following useful relation recursion on the importance weights

$$w_n(x_{1:n}) = w_{n-1}(x_{1:n-1}) \frac{d\pi_n}{d(\pi_{n-1} \otimes Q_n)}(x_{1:n}). \tag{2.9}$$

In many applications of (2.9), the Radon-Nikodým derivative $\frac{d\pi_n}{d(\pi_{n-1}\otimes Q_n)}(x_{1:n})$ is function of x_{n-1} and x_n only. Hence, one can exploit this recursion by sampling only X_n using Q_n at time n and updating the weights with a small effort. More explicitly, assume a set of $N > 0$ samples, termed as particles, $X_{1:n-1}^{(i)}$ with weights $w_{n-1}^{(i)}$ for $i = 1, \ldots, N$ are available at time $n-1$. As long as self-normalised importance sampling is concerned, it

is practical to define the weighted empirical distribution

$$\pi_{n-1}^N(dx_{1:n-1}) = \sum_{i=1}^N W_{n-1}^{(i)} \delta_{X_{1:n-1}^{(i)}}(dx_{1:n-1}), \tag{2.10}$$

as an approximation to π_{n-1}, where $W_n^{(i)}$, $i = 1, \ldots, N$ are the self-normalised importance weights

$$W_{n-1}^{(i)} = \frac{w_{n-1}(X_{1:n-1}^{(i)})}{\sum_{i=1}^N w_{n-1}(X_{1:n-1}^{(i)})}. \tag{2.11}$$

The update from π_{n-1}^N to π_n^N can be performed by first sampling $X_n^{(i)} \sim Q(X_{1:n-1}^{(i)}, \cdot)$ and computing the weights w_n at points $X_{1:n}^{(i)} = (X_{1:n-1}^{(i)}, X_n^{(i)})$ using the update rule in (2.9), and finally obtain the normalised weights $W_n^{(i)}$ using (2.11). A SIS estimate of $\pi_n(\varphi_n)$ is, then, given by

$$\pi_n^N(\varphi_n) = \sum_{i=1}^N W_n^{(i)} \varphi_n(X_{1:n}^{(i)}).$$

Being a special case of importance sampling approximation, this approximation has almost sure convergence to $\pi_n^N(\varphi_n)$ for any n (under regular conditions) as the number of particles tends to infinity; it is also possible to have a central limit theorem for $\pi_n^N(\varphi_n)$ [Geweke, 1989]. The SIS method is summarised in Algorithm 2.6.

Algorithm 2.6. ***Sequential importance sampling (SIS)***
For $n = 1, 2, \ldots$;

- *for $i = 1, \ldots, N$,*
  - *if $n = 1$; sample $X_1^{(i)} \sim q_1$, calculate $w_1(X_1^{(i)}) = \frac{d\pi_1}{dq_1}(X_1^{(i)})$.*
  - *if $n \geq 2$; sample $X_n^{(i)} \sim Q_n(X_{1:n-1}^{(i)}, \cdot)$, set $X_{1:n}^{(i)} = (X_{1:n-1}^{(i)}, X_n^{(i)})$, and calculate*

$$w_n(X_{1:n}^{(i)}) = w_{n-1}(X_{1:n-1}^{(i)}) \frac{d\pi_n}{d(\pi_{n-1} \otimes Q_n)}(X_{1:n}^{(i)}).$$

- *for $i = 1, \ldots, N$, calculate*

$$W_n^{(i)} = \frac{w_n(X_{1:n}^{(i)})}{\sum_{i=1}^N w_n(X_{1:n}^{(i)})}.$$

As in the non-sequential case, it is important to choose $\{q_n\}_{n\geq 1}$ such that the variances of $\{\pi_n^N(\varphi_n)\}_{n\geq 1}$ are minimised. Recall that in the SIS algorithm we restrict ourselves to $\{q_n\}_{n\geq 1}$ satisfying (2.8), therefore selection of the optimal proposal distributions suggested in Section 2.3 may not be possible. Instead, a more general motivation for those

$\{q_n\}_{n\geq 0}$ satisfying (2.8) might be to minimise the variance of *incremental importance weights*

$$w_{n|n-1}(x_{1:n}) = \frac{d\pi_n}{d(\pi_{n-1} \otimes Q_n)}(x_{1:n}).$$

conditional upon $x_{1:n-1}$. Note that the objective of minimising the conditional variance of $w_{n|n-1}$ is more general in the sense that it is not specific to φ_n. It was shown in Doucet [1997] that the kernel Q_n^{opt} by which the variance is minimised is given by

$$Q_n^{opt}(x_{1:n-1}, dx_n) = \pi_n(dx_n|x_{1:n-1}). \tag{2.12}$$

Before Doucet [1997], the optimum kernel was used in several works for particular applications, see e.g. Kong et al. [1994], Liu and Chen [1995], and Chen and Liu [1996]. The optimum kernel leads to the optimum incremental weight

$$w_{n|n-1}^{opt}(x_{1:n-1}) = \frac{d\pi_n}{d\pi_{n-1}}(x_{1:n-1}). \tag{2.13}$$

which does not depend on the value of x_n. This is an interesting observation and it will be revisited in Section 2.5.3.

2.5.2 Sequential importance sampling resampling

The SIS method is an efficient way of implementing importance sampling sequentially. However; unless the proposal distribution is very close to the true distribution, the importance weight step will lead over a number of iterations to a small number of particles with very large weights compared to the rest of the particles. This will eventually result in one of the normalised weights to being 1 and the others being 0, effectively leading to a particle approximation with a single particle, see Kong et al. [1994] and Doucet et al. [2000b]. This problem is called the *weight degeneracy* problem.

In order to address the weight degeneracy problem, a *resampling* step is introduced at iterations of the SIS method, leading to the *sequential importance sampling resampling* (SISR) algorithm. Generally, we can describe resampling as a method by which a weighted empirical distribution is replaced with an equally weighted distribution, where the samples of the equally weighted distribution are drawn from the weighted empirical distribution. Here, resampling is applied to π_{n-1}^N before proceeding to approximate π_n. Assume, again, that π_{n-1} is approximated with N particles $X_{1:n-1}^{(1)}, \ldots, X_{1:n-1}^{(N)}$ with normalised weights $W_{n-1}^{(i)}$ as in equation (2.10). We draw N independent samples from π_{n-1}^N, namely $\widetilde{X}_{1:n-1}^{(i)}$, $i = 1, \ldots, N$ such that

$$P(\widetilde{X}_{1:n-1}^{(i)} = X_{1:n-1}^{(j)}) = W_{n-1}^{(j)}, \quad i, j = 1, \ldots, N.$$

Obviously, this corresponds to drawing N independent samples from a multinomial distribution, therefore this particular resampling scheme is called *multinomial resampling*. After resampling, for each $i = 1, \ldots, N$ we sample $X_n^{(i)}$ from $Q_n(\widetilde{X}_{1:n-1}^{(i)}, \cdot)$, weight the particles $X_{1:n}^{(i)} = (\widetilde{X}_{1:n-1}^{(i)}, X_n^{(i)})$ using

$$W_n^{(i)} \propto \frac{d\pi_n}{d(\pi_{n-1} \otimes Q_n)}(X_{1:n}^{(i)}), \quad \sum_{i=1}^{N} W_n^{(i)} = 1.$$

The SISR method, also known as the *particle filter*, is summarised in Algorithm 2.7.

Algorithm 2.7. ***Sequential importance sampling resampling (SISR)***
For $n = 1$; for $i = 1, \ldots, N$ sample $X_1^{(i)} \sim q_1$, set $W_1^{(i)} \propto \frac{d\pi_1}{dq_1}(X_1^{(i)})$.
For $n = 2, 3, \ldots$

- *Resample $\{X_{1:n-1}^{(i)}\}_{1 \leq i \leq N}$ according to the weights $\{W_{n-1}^{(i)}\}_{1 \leq i \leq N}$ to get resampled particles $\{\widetilde{X}_{1:n-1}^{(i)}\}_{1 \leq i \leq N}$ with weight $1/N$.*
- *For $i = 1, \ldots, N$; sample $X_n^{(i)} \sim Q_n(\widetilde{X}_{1:n-1}^{(i)}, \cdot)$, set $X_{1:n}^{(i)} = (\widetilde{X}_{1:n-1}^{(i)}, X_n^{(i)})$, and set*

$$W_n^{(i)} \propto \frac{d\pi_n}{d(\pi_{n-1} \otimes Q_n)}(X_{1:n}^{(i)}).$$

The importance of resampling in the context of SMC was first demonstrated by Gordon et al. [1993] based on the ideas of Rubin [1987]. Although the resampling step alleviates the weight degeneracy problem, it has two drawbacks. Firstly, since after successive resampling steps some of the distinct particles for $X_{1:n}$ are dropped in favour of more copies of highly-weighted particles. This leads to the impoverishment of particles such that for $k << n$, very few particles represent the marginal distribution of X_k under π_n [Andrieu et al., 2005; Del Moral and Doucet, 2003; Olsson et al., 2008]. Hence, whatever being the number of particles, $\pi_n(dx_{1:k})$ will eventually be approximated by a single unique particle for all (sufficiently large) n. As a result, any attempt to perform integrations over the path space will suffer from this form of degeneracy, which is called *path degeneracy*. The second drawback is the extra variance introduced by the resampling step. There are a few ways of reducing the effects of resampling.

- One way is adaptive resampling i.e. resampling only at iterations where the effective sample size drops below a certain proportion of N. For a practical implementation, the effective sample size at time n itself should be estimated from particles as well. One particle estimate of $N_{\mathit{eff},n}$ is given in Liu [2001, pp. 35-36]

$$\widetilde{N}_{\mathit{eff},n} = \frac{1}{\sum_{i=1}^{N} W_n^{(i)2}}.$$

- Another way to reduce the effects of resampling is to use alternative resampling methods to multinomial resampling. Let $I_n(i)$ is the number of times the i'th particle is drawn from π_n^N in a resampling scheme. A number of resampling methods have been proposed in the literature that satisfy $\mathbb{E}[I_n(i)] = NW_n^{(i)}$ but have different $var[I_n(i)]$. The idea behind $\mathbb{E}[I_n(i)] = NW_n^{(i)}$ is that the mean of the particle approximation to $\pi_n(\varphi_n)$ remains the same after resampling. Standard resampling schemes include multinomial resampling [Gordon et al., 1993], residual resampling [Liu and Chen, 1998; Whitley, 1994], stratified resampling [Kitagawa, 1996], and systematic resampling [Carpenter et al., 1999; Whitley, 1994]. There are also some non-standard resampling algorithms such that the particle size varies (randomly) after resampling (e.g. Crisan et al. [1999]; Fearnhead and Liu [2007]), or the weights are not constrained to be equal after resampling (e.g. Fearnhead and Clifford [2003]; Fearnhead and Liu [2007]).

- A third way of avoiding path degeneracy is provided by the *resample-move* algorithm [Gilks and Berzuini, 2001], where each resampled particle $\widetilde{X}_{1:n}^{(i)}$ is moved according to a MCMC kernel $K_n : \mathscr{X}_n \to \mathcal{P}(\mathscr{E}_n)$ whose invariant distribution is π_n. In fact we could have included this MCMC move step in Algorithm 2.7 to make the algorithm more generic. However, the resample-move algorithm is a useful degeneracy reduction technique usually in a much more general setting. Although possible in principle, it is computationally infeasible to apply a kernel to the path space on which current particles exist as the state space grows at evert iteration of SISR. The resample-move algorithm will be revisited in Section 2.5.4, where it is considered as a special case of a wide class of sequential sampling methods that operate on sequences of arbitrary spaces.

- The final method we will mention here that is used to reduce path degeneracy is *block sampling* [Doucet et al., 2006], where at time n one samples components $X_{n-L+1:n}$ for $L > 1$, and previously sampled values for $X_{n-L+1:n-1}$ are simply discarded. In return of the computational cost introduced by L, this procedure reduces the variance of weights and hence reduces the number of resampling steps (if an adaptive resampling strategy is used) dramatically. Therefore, path degeneracy is reduced.

2.5.3 Auxiliary particle filter

Recall that when the optimum proposal Q_n^{opt} is used to sample x_n the corresponding optimum incremental weight $w_{n|n-1}^{opt}$ does not depend on the value of x_n. Therefore, the optimum incremental weight indicates which particles are likely to represent π_n better even before proposing the new state x_n. This encourages for a sequential sampling

strategy where the optimum incremental weights are involved in deciding on with which particles the algorithm proceeds to the next time step, and this is the strategy on which the *auxiliary particle filter* [Pitt and Shephard, 1999] is based. To understand how we can implement this strategy, it is useful to see how target distributions at iterations are modified with the resampling step in the SISR algorithm. One can show that given π_{n-1}^N in (2.10) to be the SISR approximation to π_{n-1}, SISR targets the following distribution at time n (provided that resampling step is performed)

$$\bar{\pi}_n(dx_{1:n}) \propto \left[\sum_{i=1}^N W_{n-1}^{(i)} w_{n|n-1}^{opt}(X_{1:n-1}^{(i)}) \delta_{X_{1:n-1}^{(i)}}(dx_{1:n-1})\right] Q_n^{opt}(dx_n|x_{1:n-1}). \tag{2.14}$$

In the standard SISR algorithm, the following proposal distribution is used to implement importance sampling at time n

$$\bar{q}_n(dx_{1:n}) = \underbrace{\left[\sum_{i=1}^N W_{n-1}^{(i)} \delta_{X_{1:n-1}^{(i)}}(dx_{1:n-1})\right]}_{\text{resampling } X_{1:n-1}} \underbrace{Q_n(x_{1:n-1}, dx_n)}_{\text{proposing } X_n}$$

which does not fully exploit the structure in (2.14). As a result we have a well known drawback of SISR: if π_n varies significantly compared to π_{n-1}, the variance of the weights can be quite high. This results in an inefficient algorithm, and a large number of particles may be required for recovery.

Provided that one can calculate $w_n^{opt}(x_{1:n-1})$, a more sensible choice for $\bar{q}_n(dx_{1:n})$ could be

$$\bar{q}_n^{opt}(dx_{1:n}) = \left[\sum_{i=1}^N \overline{W}_{n-1}^{(i)} \delta_{X_{1:n-1}^{(i)}}(dx_{1:n-1})\right] Q_n(dx_n|x_{1:n-1}). \tag{2.15}$$

where $\overline{W}_{n-1}^{(i)} \propto W_{n-1}^{(i)} w_n^{opt}(X_{1:n-1}^{(i)})$ such that $\sum_{i=1}^N \overline{W}_{n-1}^{(i)} = 1$. Then, the importance weight for particle $X_{1:n}^{(i)} = \left(X_{1:n-1}^{(j)}, X_n^{(i)}\right)$ would be

$$W_n^{(i)} \propto \frac{d\bar{\pi}_n}{d\bar{q}_n^{opt}}(X_{1:n}^{(i)}) = \frac{dQ_n^{opt}(X_{1:n-1}^{(j)}, \cdot)}{dQ_n(X_{1:n-1}^{(j)}, \cdot)}(X_n^{(i)}).$$

This type of particle filter is called an auxiliary particle filter in the literature. The term 'auxiliary' is due to treating $X_{1:n-1}$ at time n as auxiliary; because in many cases where a particle filter is used, integration of functions on $\mathcal{X}_n$ with respect to the marginal distribution $\pi_n(dx_n)$ is the main interest and resampling of $X_{1:n-1}$ in this particular way helps the Monte Carlo approximation of such integrations improve.

One remarkable point here is that if one can use $Q_n = Q_n^{opt}$, then all the particles have equal weights. This shows how this sampling scheme can reduce weight degeneracy

effectively. (Notice also that $Q_n = Q_n^{opt}$ results in the regular SISR with optimum proposal, where the sampling and resampling steps are interchanged.) However, it may not be possible (or straightforward) to sample from Q_n^{opt} or calculate $w_n^{opt}(x_{1:n-1})$. This does not restrict the use of the idea behind the auxiliary particle filter, though. In fact, the auxiliary particle filter is more general: We can perform importance sampling for $\bar{\pi}_n$ by constructing a $\bar{q}_n$ which can be generically written as

$$\bar{q}_n^{aux}(dx_{1:n}) = \sum_{i=1}^{N} \alpha_{n-1}^{(i)} \delta_{X_{1:n-1}^{(i)}}(dx_{1:n-1}) Q_n(dx_n | x_{1:n-1}).$$

We have complete control over α_{n-1} and Q_n; however the idea is to be able to sample those particles $X_{1:n-1}^{(i)}$ which represents $\pi_n(x_{1:n-1})$ better, and sample X_n approximately from the optimal proposal distribution in order to have weights with low variance. Therefore, the rule of thumb is to make $\alpha_{n-1}^{(i)}$ and Q_n as close as possible to $\overline{W}_{n-1}^{(i)}$ and Q_n^{opt}. Indeed, the authors in Andrieu et al. [2001] propose an improved auxiliary particle filter scheme, where (2.15) or a suitable approximation to (2.15) is suggested to be used.

2.5.4 Sequential Monte Carlo samplers

Sequential Monte Carlo samplers [Del Moral et al., 2006] cover a very large class of SMC methods. Assume that we have a sequence of somehow related distributions $\pi_1, \ldots, \pi_p$ where each π_n is defined on an arbitrary measurable space $(\mathcal{X}_n, \mathcal{E}_n)$. There are many potential choices for $\pi_1, \ldots, \pi_p$ leading to various integration and optimisation algorithms; examples can be found in Chopin [2002] for static parameter estimation, Gelman and Meng [1998] and Neal [2001] for targeting a distribution through a sequence of intermediate distributions, Del Moral et al. [2006] for global optimisation, Johansen et al. [2005] and Del Moral et al. [2006] for rare event simulation and density estimation, and Del Moral et al. [2012] for approximate Bayesian computation. The problem of approximating these distributions sequentially using Monte Carlo is beyond the extend of the classical SIS or SISR methods, since these require the distributions to be defined on increasing spaces.

The first approach that comes to mind is to treat each π_n individually and perform importance sampling for each of them independently. Obviously, this approach has the difficulties of importance sampling: unless the distribution of interest is a standard low-dimensional one, importance sampling is almost never used when there are alternatives. The main reason for that is the difficulty of designing an good proposal. One reasonable way is to do importance sampling for π_n individually, but this time by designing the importance distributions sequentially using an initial distribution η_1 and a sequence of transition kernels $\{K_n : \mathcal{X}_{n-1} \to \mathcal{P}(\mathcal{E}_n)\}_{n \geq 1}$. The idea here is that if the distributions π_n varies slowly in n, then it is possible to obtain samples to approximate π_n effectively by

using K_n to slowly move the samples obtained to approximate π_{n-1}. Let us assume that we begin with sampling $X_1^{(1)}, \ldots, X_1^{(N)}$ from η_1 to approximate π_1. At times $n \geq 2$, we sample $X_n^{(i)}$ from $K_n(X_{n-1}^{(i)}, \cdot)$. The importance weight of $X_n^{(i)}$ is given by

$$w_n^{(i)} = \frac{d\pi_n}{d\eta_n}(X_n^{(i)}), \quad \eta_n(dx_n) = \eta_{n-1}K_n(dx_n).$$

The choice of K_n's are optional except the requirement that $\pi_n \ll \eta_{n-1}K_n$; however it is crucial for the the performance of this method. In the literature, several different types of moves are used, such as independent proposals [West, 1993], local random moves [Givens and Raftery, 1996], MCMC and Gibbs moves [Del Moral et al., 2006], etc.

This sequential implementation of importance sampling approach is attractive and optimal in some sense (we will see soon in what sense), however it has a quite restrictive limitation: in most cases it is impossible to calculate the importance distribution η_n. SMC samplers come into role at this point, circumventing the need for calculation of η_n. The main idea of the method is to construct the synthetic distributions $\widetilde{\pi}_n$ on the extended spaces $(\mathcal{X}_1 \times \ldots \times \mathcal{X}_n, \mathcal{E}_1 \otimes \ldots \otimes \mathcal{E}_n)$ as

$$\widetilde{\pi}_n(dx_{1:n}) = \pi_n(dx_n) \prod_{i=1}^{n-1} L_i(x_{i+1}, dx_i) \tag{2.16}$$

where each $L_n : \mathcal{X}_{n+1} \to \mathcal{P}(\mathcal{X}_n)$ is a backward Markov kernel. Since $\widetilde{\pi}_n$ admits π_n marginally by construction, importance sampling on $\widetilde{\pi}_n$ using the following proposal distribution

$$\widetilde{\eta}_n(dx_{1:n}) = \eta_1(dx_1) \prod_{i=2}^{n} K_i(x_{i-1}, dx_i).$$

can provide an approximation for π_n as well. Although, freedom to choose K_n's and L_n's contribute to the method's generality, the performance of the method crucially depends on the their choice. In fact, the central limit theorem presented in Del Moral et al. [2006] demonstrates that the variance of the estimator is strongly dependent upon the choice of these kernels. The importance weight for this method is given by

$$w_n(x_{1:n}) = \frac{d\widetilde{\pi}_n}{d\widetilde{\eta}_n}(x_{1:n}).$$

It was shown in Del Moral et al. [2006] that given K_n, the optimum backward kernel L_{n-1}^{opt} which minimises the variance of the importance weights satisfies the relation

$$\eta_n \otimes L_{n-1}^{opt} = \eta_{n-1} \otimes K_n.$$

It can be shown that the importance weights for the optimum backward kernel is

$$w_n^{opt}(x_{1:n}) = \frac{d\pi_n}{d\eta_n}(x_n).$$

This result reveals that the optimum backward kernel takes us back to the case where one performs importance sampling on the marginal space instead of the extended one. However, most of the time η_n cannot be calculated, hence other sub-optimal backward kernels must be used. It was shown in Del Moral et al. [2006] that when L_{n-1}^{opt} is not used, the variance of $w_n(x_{1:n})$ can not be stabilised. For that reason, resampling of the samples that are used for approximating π_{n-1} is necessary before moving to the approximation of π_n. Actually, this can be done thanks to the possibility of constructing $\widetilde{\pi}_n$ such that the importance weights can be expressed as a product of incremental weights. Assume that $\pi_n \ll K_n$ and $L_n \ll \pi_n$ for all n. Then it can be shown that for a bounded measurable function φ_n on $\mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ we have $\widetilde{\pi}_n(\varphi_n) = \widetilde{\eta}_n(\varphi_n w_n)$ where the importance weights w_n are given by

$$w_n(x_{1:n}) = \frac{d\pi_1}{d\eta_1}(x_1) \prod_{i=2}^{n} \frac{dL_{i-1}(x_i, \cdot)}{d\pi_{i-1}}(x_{i-1}) \frac{d\pi_i}{dK_i(x_{i-1}, \cdot)}(x_i). \tag{2.17}$$

Equation (2.17) admits a recursion in n as

$$w_n(x_{1:n}) = w_{n|n-1}(x_{n-1}, x_n) w_n(x_{1:n-1})$$

where the incremental weight $w_{n|n-1}(x_{n-1}, x_n)$ is given by

$$w_{n|n-1}(x_{n-1}, x_n) = \frac{d\pi_n}{dK_n(x_{n-1}, \cdot)}(x_n) \frac{dL_{n-1}(x_n, \cdot)}{d\pi_{n-1}}(x_{n-1}). \tag{2.18}$$

Note that the recursive form of the weights enables us to implement an SMC method for the synthetic distributions $\widetilde{\pi}_n$. Actually, when (2.17) exists, the SMC sampler for $\pi_1, \ldots, \pi_p$ is the SISR algorithm targeting $\widetilde{\pi}_1, \ldots, \widetilde{\pi}_p$ using the initial and transitional proposal distributions η_1 and K_n, $n = 2, \ldots, p$ respectively, and its incremental weights are given in (2.18).

Note that, in practice even if L_n is not absolutely continuous with respect to π_n, we can still obtain importance weights factorized into incremental weights by taking the restrictions of L_n's to the supports of π_n's. Note, also, that as in importance sampling, SIS, and SISR, even if we know $\widetilde{\pi}_n$'s and $\widetilde{\eta}_n$'s only up to some normalising constants we can still perform the SMC samplers algorithm to approximate the integrals π_n and to estimate the unknown normalising constants as well.

SMC samplers generalise many related works previously in the literature. For exam-

ple, the *annealed importance sampling* method, which corresponds to the SMC sampler without resampling where L_{n-1} satisfies

$$\pi_{n-1}K_n \otimes L_{n-1} = \pi_{n-1} \otimes K_n \tag{2.19}$$

and K_n is such that π_{n-1} is K_n-invariant, is proposed by Neal [2001] for sequences of slightly varying distributions. To deal with the variance problem for general cases, the equivalent choice of kernels are used in (among others) Chopin [2002] and Gilks and Berzuini [2001] with resample-move strategies, which actually corresponds to the SMC sampler algorithm with resampling. *Population Monte Carlo*, presented by Cappé et al. [2004] and Celeux et al. [2006] with an extension, is another special case of SMC samplers where the authors consider the homogeneous case where $\pi_n = \pi$ and $L_n(x, dx') = \pi(dx')$ and $K_n(x, dx') = K_n(dx')$. Finally Liang [2002] presents a related algorithm where $\pi_n = \pi$ and $K_n(x, x') = L_n(x, dx') = K(x, dx')$.

2.6 Approximate Bayesian computation

Assume that we have a random variable of interest X, taking values in $\mathcal{X}$. Its probability distribution $\pi(dx)$ has a density on $\mathcal{X}$ with respect to a dominating measure dx, which is abusively denoted as $\pi(x)$[1]. The value of X, denoted by x, is observed indirectly through an observation process generating values $Y \in \mathcal{Y}$ according to conditional observation probability distribution who also has a density on $\mathcal{Y}$ with respect to dy, which is denoted as $g(y|x)$. The density $g(y|x)$ is also called the *likelihood*. The posterior distribution of X given $Y = y$ has the following density which is given by Bayes' theorem

$$\pi(x|y) = \frac{\pi(x)g(y|x)}{\int_{\mathcal{X}} \pi(x')g(y|x')dx'}.$$

Approximate Bayesian computation (ABC) deals with the problem of Monte Carlo approximation to $\pi(x|y)$ when the likelihood $g(y|x)$ is *intractable*. By intractability it is meant either that the density does not have a close form expression or that it is prohibitive to calculate it. ABC methods try to approximate $\pi(x|y)$ without circumventing the calculation of $g(y|x)$ and for this reason they are also known as *likelihood-free* methods. The main idea behind ABC is simulating from the observation process and accepting simulated samples provided that they are close to the observed value y in some sense. ABC methods have appeared in the past ten years as one of the most satisfactory approach to intractable likelihood problems. This section is a brief and limited review of the main contributions to the ABC methodology, for a more detailed recent review, one

[1]It is simpler to describe the methodology in this section when we use densities instead of measures.

can see Marin et al. [2011].

The idea core to ABC is first mentioned in Rubin [1984]; but the first ABC method was proposed by Tavaré et al. [1997] as a special case of rejection sampling for discrete $\mathcal{Y}$. It proposes to sample (x, y) from $\pi(x)g(u|x)$ and consider only those samples for which $u = y$. It is not difficult to show that if the accepted samples are $(X^{(1)}, y), \ldots, (X^{(N)}, y)$, then $X^{(1)}, \ldots, X^{(N)}$ are samples from the posterior $\pi(x|y)$. Note that this a rejection sampling method for the distribution $\pi(x, u|y)$ on $\mathcal{X} \times \mathcal{Y}$, which is given by

$$\pi(x, u|y) \propto \pi(x)g(u|x)\mathbb{I}_y(u) \tag{2.20}$$

and when this density is integrated over u, we end up with $\pi(x|y)$. This method is exact in the sense that the obtained samples for X are drawn from $\pi(x|y)$. However, obviously $\mathbb{I}_y(u)$ would not work for continuous $\mathcal{Y}$, since the probability of hitting $\{y\}$ will be zero. The first genuine ABC method, proposed by Pritchard et al. [1999] as a rejection sampling method also, relaxes $\mathbb{I}_y(u)$ and replaces (2.20) with

$$\pi_\epsilon(x, u|y) \propto \pi(x)g(u|x)\mathbb{I}_{A_y^\epsilon}(u) \tag{2.21}$$

where A_y^ϵ is called the ABC set and defined based on some summary statistic $s : \mathcal{Y} \to \mathbb{R}^{d_s}$ and a distance metric $\rho : \mathbb{R}^{d_s} \times \mathbb{R}^{d_s} \to \mathbb{R}$ as

$$A_y^\epsilon = \{u \in \mathcal{Y} : \rho[s(u), s(y)] < \epsilon\}. \tag{2.22}$$

If s is sufficient with respect to x, one can show that as ϵ tends to zero, the marginal of density $\pi_\epsilon(x, u|y)$ with respect to x converges to the posterior $\pi(x|y)$. In most cases sufficient statistics are not available, hence the choice of summary statistics is of great importance. The ABC literature is rich in papers discussing on the selection of these sufficient statistics, see Fearnhead and Prangle [2012] for an example. The ABC method in Pritchard et al. [1999] is also a rejection sampling method, targeting $\pi_\epsilon(x, u|y)$: generate (X, U) from $\pi(x)g(u|x)$ and consider only those samples for which $U \in A_y^\epsilon$. This method is summarised in Algorithm 2.8.

Algorithm 2.8. ***Rejection sampling for ABC:*** *To generate a single sample from* $\pi_\epsilon(x, u|y)$*,*

1. *Generate* $(X, U) \sim \pi(x)g(u|x)$.
2. *If* $U \in A_y^\epsilon$*, accept* (X, U)*; else go to 1.*

Using simulations from the prior distribution $\pi(x)$ can be inefficient since this does not take neither the data nor the previously accepted samples into account when proposing a new x and thus fails to propose values located in high posterior probability regions.

To overcome this impracticality of rejection sampling, an MCMC based ABC method was developed by Marjoram et al. [2003]. The method is simply an MCMC algorithm targeting $\pi_\epsilon(x,u|y)$ which uses an instrumental kernel with density $q(x'|x)g(u'|x')$ to move samples (x,u) and takes either (x',u') or (x,u) as the next sample according to the corresponding acceptance probability

$$\alpha(x,u;x',u') = \min\left\{1, \frac{q(x|x')\pi(x')\mathbb{I}_{A_y^\epsilon}(u')}{q(x'|x)\pi(x)}\right\}, \quad x,x' \in \mathcal{X}, u,u' \in \mathcal{U}.$$

The MCMC-ABC method is given in Algorithm 2.9.

Algorithm 2.9. ***MCMC for ABC:*** *Begin with some* $(X_1,U_1) \in \mathcal{X}\times\mathcal{U}$. *For* $n = 2,3,\ldots$

- *Generate* $(X',U') \sim q(x'|x)g(u'|x')$.
- *Set* $(X_n,U_n) = (X',U')$ *with probability* $\alpha(X_{n-1},U_{n-1};X_n,U_n)$*; otherwise set* $(X_n,U_n) = (X_{n-1},U_{n-1})$.

It is useful to interpret the ABC posterior all in terms densities: We can consider $\mathbb{I}_{A_y^\epsilon}(u)$ as to which the density of the conditional distribution of Y given $U=u$, say $\kappa_\epsilon(y|u)$, is proportional to. Then we can rewrite (2.21) as

$$\pi_\epsilon(x,u|y) = \frac{\pi(x)g(u|x)\kappa_\epsilon(y|u)}{\int_{\mathcal{X}\times\mathcal{Y}}\pi(x')g(u'|x')\kappa_\epsilon(y|u')dx'du'}.$$

A useful generalisation of $\pi_\epsilon(x,u|y)$ can be made by taking $\kappa_\epsilon(y|u)$ some normalised kernel with bandwidth ϵ centred at u. In many applications, it is practical sometimes to take κ_ϵ proportional not to an indicator function but to a smooth kernel, such as a Gaussian kernel, to make calculations tractable or to avoid computational waste due to rejections. We will see a use of choosing a smooth kernel in Chapter 6.

Another use of kernels in defining the ABC posterior is to be able to express the difference between the ABC posterior and the real posterior in terms of model error. Note that the ABC suffers from model discrepancy since it corresponds to performing Bayesian inference for the case where the observation Y has the conditional probability density not being $g(y|x)$ but the following:

$$g_\epsilon(y|x) = \int_{\mathcal{Y}} g(u|x)\kappa_\epsilon(y|u)du.$$

Therefore, we say that the ABC posterior is not 'calibrated'. A way of rephrasing this is that if the model included an error term, characterised by κ_ϵ, then the ABC would target the true posterior hence the ABC posterior would be 'calibrated' [Wilkinson, 2008]. This

leads to the method of *noisy ABC* [Dean et al., 2011; Fearnhead and Prangle, 2012], which adds noise to the (summary statistic of) data itself to have $y^\epsilon \sim \kappa_\epsilon(\cdot|y)$, and then perform ABC for the modified data by targeting $\pi_\epsilon(x, u|y^\epsilon)$, which is calibrated.

Other than approximating the posterior distribution of a single random variable, the ABC approach also extends to sequential inference. Jasra et al. [2012] propose an ABC implementation scheme for approximating the densities like $\pi_n(x_{1:n}|y_{1:n})$ in hidden Markov models (HMM), where $\{X_t\}_{t\geq 1}$ is a Markov process and the distribution of the observables $\{Y_t\}_{t\geq 1}$ conditioned on the hidden process is intractable. Their approach is related to the convolution particle filter of Campillo and Rossi [2009]. Dean et al. [2011] discuss the ABC implementation for HMMs further and show that the model for which noisy ABC is exact is also a HMM; therefore they conclude that noisy ABC can be implemented for HMMs. We will see the sequential implementation of ABC as well as its use for static parameter estimation in more detail in Chapter 6.

While decreasing the value of ϵ obviously makes $\pi_\epsilon(x, u|y)$ close to the true posterior, the variance of its Monte Carlo becomes a more crucial issue. Therefore, it is important to keep the variance of the approximation at a reasonable level while making ϵ sufficiently small. For this reason, SMC samplers are used for approximating a sequence of distributions $\{\pi_k(x, u|y) = \pi_{\epsilon_k}(x, u|y)\}_{1\leq k\leq p}$, where $\epsilon_1 > \ldots > \epsilon_p = \epsilon$ and the difference between successive π_k's are small enough to make successive $\pi_k(x, u|y)$ varying sufficiently slowly. SMC samplers are used in the ABC context in Sisson et al. [2007] and the method there was improved in Beaumont et al. [2009]; Toni et al. [2009] and Sisson et al. [2009]. Del Moral et al. [2012] showed the relation of these works to SMC samplers explicitly. Other novelties of Del Moral et al. [2012] is that the authors rely on M repeated simulations of the pseudo-data u and benefit the variance reduction property of Monte Carlo averaging and they propose a scheme for adaptive selection of the sequence of tolerance levels $\{\epsilon_k\}_{1\leq k\leq n}$. The forward kernel at step k is chosen to leave π_{k-1} invariant and backward kernel is chosen to satisfy (2.19).

Other than inference of hidden variables, there is a lot of work for model selection using ABC methods. We will not review these methods as they are not of particular interest for this thesis; the interested reader may see Marin et al. [2011] for a review and the references therein for details.

Chapter 3

Hidden Markov Models and Parameter Estimation

Summary: *This chapter contains the second half of the literature survey. The main purpose of this chapter is to introduce hidden Markov models (HMM), which are also known as general state-space models, and review their use in the literature as a powerful framework for filtering and parameter estimation.*

3.1 Introduction

HMMs arguably constitute the widest class of time series models that are used for modelling stochastic dynamic systems. In Section 3.2, we will introduce HMMs using a formulation that is appropriate for filtering and parameter estimation problems. We will restrict ourselves to discrete time homogenous HMMs whose dynamics for their hidden states and observables admit conditional probability densities which are parametrised by vector valued static parameters. However, this is our only restriction; we keep our framework general enough to cover those models with non-linear non-Gaussian dynamics.

One of the main problems dealt within the framework of HMMs is *optimal Bayesian filtering*, which has many applications in signal processing and related areas such as speech processing [Rabiner, 1989], finance [Pitt and Shephard, 1999], robotics [Gordon et al., 1993], communications [Andrieu et al., 2001], etc. Due to the non-linearity and non-Gaussianity of most of models of interest in real life applications, approximate solutions are inevitable and SMC is the main computational tool used for this; see e.g. Doucet et al. [2001] for a wide selection of examples demonstrating use of SMC. SMC methods have already been presented in its general form in Section 2.5, we will present their application to HMMs for optimal Bayesian filtering in Section 3.3.

In practice, it is rare that the practitioner has complete knowledge on the static parameters of the time series model which she uses to perform optimal Bayesian filtering. This raises the necessity of 'calibrating the model', hence estimating its static parameters. Note also that estimating the static parameters of a HMM itself may be the main objective. Section 3.4 of this chapter contains a review of the methodology for static

parameter estimation in HMMs, in particular we will present some of the popular maximum likelihood estimation (MLE) algorithms. We will also show how to obtain SMC approximations of those MLE algorithms for HMMs.

Although we present optimal Bayesian filtering and statical parameter estimation methods and their SMC approximations within the framework of HMMs, we would like to stress that this thesis does contain time-series models which are *not* a HMM (at least in the way we deal with it). We will mention such models in Section 3.2.1. The reason why we restrict ourselves to HMMs is that the computational tools developed for them are generally applicable to more general time series models with some suitable modifications.

3.2 Hidden Markov models

We begin with the definition of a HMM. Let $\{X_n\}_{n\geq 1}$ be a homogenous Markov chain defined on $(\mathcal{X}, \mathcal{E}_{\mathcal{X}})$. Suppose that this process is observed as another process $\{Y_n\}_{n\geq 1}$ defined on $(\mathcal{Y}, \mathcal{E}_{\mathcal{Y}})$ such that the conditional distribution on Y_n given all the other random variables depends only on X_n. Then the bivariate process $\{X_n, Y_n\}_{n\geq 1}$ is called a HMM. We give below a more formal definition which is taken from Cappé et al. [2005]; we additionally assume that the HMM is parametrised by a vector valued static parameter.

Definition 3.1 (HMM). *Let $(\mathcal{X}, \mathcal{E}_{\mathcal{X}})$ and $(\mathcal{Y}, \mathcal{E}_{\mathcal{Y}})$ be two measurable spaces, $d_\theta > 0$, and Θ is a compact subset of $\mathbb{R}^{d_\theta}$. For any $\theta \in \Theta$, let μ_θ, F_θ, and G_θ denote, respectively, a probability measure on $(\mathcal{X}, \mathcal{E}_{\mathcal{X}})$, a Markov transition kernel on $(\mathcal{X}, \mathcal{E}_{\mathcal{X}})$, and a transitional kernel from $(\mathcal{X}, \mathcal{E}_{\mathcal{X}})$ to $(\mathcal{Y}, \mathcal{E}_{\mathcal{Y}})$. Consider the Markov transition kernel H_θ defined on the product space $(\mathcal{X} \times \mathcal{Y}, \mathcal{E}_{\mathcal{X}} \otimes \mathcal{E}_{\mathcal{Y}})$ such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $C \in \mathcal{E}_{\mathcal{X}} \otimes \mathcal{E}_{\mathcal{Y}}$*

$$H_\theta[(x, y), C] = \int_C F_\theta(x, dx')G_\theta(x', dy').$$

Then, the Markov chain $\{X_n, Y_n\}_{n\geq 1}$ with initial distribution $\mu_\theta \otimes G_\theta$, and with transitional kernel H_θ is called a hidden Markov model (HMM) parametrised by θ.

Although this definition concerns the joint process $\{X_n, Y_n\}_{n\geq 1}$, the term 'hidden' is justified when only $\{Y_n\}_{n\geq 1}$ is observable. We call $\{X_n\}_{n\geq 1}$ the hidden process and its states the hidden states, and $\{Y_n\}_{n\geq 1}$ is called the observed process, containing the observed states. We will deal with real valued vector processes, that is why we always take $\mathcal{X} \in \mathbb{R}^{d_x}$ and $\mathcal{Y} \in \mathbb{R}^{d_y}$. Note, also, that it is Definition 3.1 from which it follows that $\{X_n\}_{n\geq 1}$ is $Markov(\mu_\theta, F_\theta)$ and observations $\{Y_n\}_{n\geq 1}$ conditioned upon $\{X_n\}_{n\geq 1}$ are independent and have the conditional distributions $G_\theta(x_n, \cdot)$, i.e. for every $A \in \mathcal{E}_{\mathcal{X}}$

and $B \in \mathcal{E}_{\mathcal{Y}}$ we have

$$P_\theta(X_1 \in A) = \mu_\theta(A), \quad P_\theta\left(X_n \in A \,|X_{1:n-1} = x_{1:n-1}\right) = F_\theta(x_{n-1}, A), \tag{3.1}$$

$$P_\theta\left(Y_n \in B \,\middle|\, \{X_t\}_{t\geq 1} = \{x_t\}_{t\geq 1}, \{Y_t\}_{t\neq n} = \{y_t\}_{t\neq n}\right) = G_\theta(x_n, B). \tag{3.2}$$

In the time series literature, the term HMM has been widely associated with the case of $\mathcal{X}$ being finite [Rabiner, 1989] and those models with continuous $\mathcal{X}$ are often referred to as state-space models. Again, in some works the term 'state space models' refers to the case of linear Gaussian systems [Anderson and Moore, 1979]. We emphasise at this point that in this thesis we shall keep the framework as general as possible. We consider the general case of measurable spaces and we avoid making any restrictive assumptions on μ_θ, F_θ, and G_θ that impose a certain structure on the dynamics of the HMM. Also, we clarify that in contrast to previous restrictive use of terminology, we will use both terms 'HMM' and 'general state space model' to describe exactly the same thing as defined by Definition 3.1.

For the rest of the thesis, we will be dealing with *fully dominated HMMs*, where μ_θ, $F_\theta(x, \cdot)$ and $G_\theta(x, \cdot)$ have densities with respect to some dominating measures. We give a formal definition of a fully dominated HMM here.

Definition 3.2 (**fully dominated HMM**). *Consider the HMM in Definition 3.1. Suppose that there exists probability measures λ on $(\mathcal{X}, \mathcal{E}_{\mathcal{X}})$ and ν on $(\mathcal{Y}, \mathcal{E}_{\mathcal{Y}})$ such that (i) μ_θ is absolutely continuous with respect to λ (ii) for all $x \in \mathcal{X}$, $F_\theta(x, \cdot)$ is absolutely continuous with respect to λ with transition density function $f_\theta(\cdot|x)$ and (iii) for all $x \in \mathcal{X}$, $G_\theta(x, \cdot)$ is absolutely continuous with respect to ν with transition density function $g_\theta(\cdot|x)$. Then the HMM is said to be fully dominated and the joint Markov transition kernel H_θ is dominated by the product measure $\lambda \otimes \nu$ and admits the transition density function*

$$h_\theta(x', y'|x, y) = f_\theta(x'|x) g_\theta(y'|x').$$

Therefore, for a fully dominated HMM as in Definition 3.2, the joint probability density of $(X_{1:n}, Y_{1:n})$ exists and it is given by

$$p_\theta(x_{1:n}, y_{1:n}) = \mu_\theta(x_1) g_\theta(y_1|x_1) \prod_{t=2}^{n} f_\theta(x_t|x_{t-1}) g_\theta(y_t|x_t) \tag{3.3}$$

where, with abuse of notation, we have used μ also to denote the density of the probability measure μ. Note, the joint law of all the variables of the HMM up to time n is summarised in (3.3) from which we derive several probability densities of interest. One example is the

likelihood of the observations up to time n which can be derived as

$$p_\theta(y_{1:n}) = \int p_\theta(x_{1:n}, y_{1:n})\lambda(dx_{1:n}). \tag{3.4}$$

Maximisation of this quantity with respect to θ is the main interest of this thesis. Another important probability density, which will be pursued in detail, is the density of the posterior distribution of $X_{1:n}$ given $Y_{1:n} = y_{1:n}$, which is obtained by using the Bayes' theorem

$$p_\theta(x_{1:n}|y_{1:n}) = \frac{p_\theta(x_{1:n}, y_{1:n})}{p_\theta(y_{1:n})} \tag{3.5}$$

3.2.1 Extensions to HMMs

Although HMMs are the most common class of time series models in the literature, there are also many time series models which are not a HMM and are still of great importance. These models differ from HMMs mostly because they do not possess the conditional independency of observations. Here, we give two examples that we will also use in this thesis.

- In the first example of such models, the process $\{X_n\}_{n\geq 1}$ is still a Markov chain; however the conditional distribution of Y_n, given all past variables $X_{1:n}$ and $Y_{1:n-1}$, depends not only on the value of X_n but also on the values of past observations i.e. $Y_{1:n-1}$. If we denote the probability density of this conditional distribution $g_{\theta,n}(y_n|x_n, y_{1:n-1})$, the joint probability density of $(X_{1:n}, Y_{1:n})$ is

$$p_\theta(x_{1:n}, y_{1:n}) = \mu_\theta(x_1)g_\theta(y_1|x_1)\prod_{t=2}^{n} f_\theta(x_t|x_{t-1})g_{\theta,t}(y_t|x_t, y_{1:t-1}).$$

  If Y_n given X_n is independent of the past values of the observations prior to time $n-k$, then we can define a g_θ such that $g_{\theta,n}(y_n|x_n, y_{1:n-1}) = g_\theta(y_n|x_n, y_{n-k:n-1})$ for all n. One example of such models is a changepoint model e.g. see Fearnhead and Liu [2007]. We will encounter changepoint models in Chapter 4 of this thesis.

  The terminology regarding the type of models where we have $g_\theta(y_n|x_n, y_{n-k:n-1})$ is not fully standardised. One term that is used is *Markov switching models*; *Markov jump systems* is also used at least in cases where the hidden state space is finite [Cappé et al., 2005]. These models have much in common with basic HMMs in the sense that virtually identical computational tools may be used for both models. In the particular context of SMC, the similarity between these to types of models is more clearly exposed in Del Moral [2004] via the Feynman-Kac representation of SMC methods, where the conditional density of observation at time n is treated generally as a *potential function* of x_n.

- In another type of time series models that are not HMM the latent process $\{X_n\}_{n\geq 1}$ is, again, still a Markov chain; however observation at current time depends on all the past values, i.e. Y_n conditional on $(X_{1:n}, Y_{1:n-1})$ depends on all of these conditioned random variables. Actually, these models are usually the result of marginalising an extended HMM. Consider the HMM $\{(X_n, Z_n), Y_n\}_{n\geq 1}$, where the joint process $\{X_n, Z_n\}_{n\geq 1}$ is a Markov chain such that its transitional law admits the density f_θ with respect to the product measure $\lambda_1 \otimes \lambda_2$ which can be factorized as

$$f_\theta(x_n, z_n|x_{n-1}, z_{n-1}) = f_{\theta,1}(x_n|x_{n-1}) f_{\theta,2}(z_n|x_n, z_{n-1}).$$

and the observation Y_n depends only on X_n and Z_n given all the past random variables and admits the probability density $g_\theta(y_n|x_n, z_n)$. Now, the marginal bivariate process $\{X_n, Y_n\}_{n\geq 1}$ is not a HMM and we express the joint density of $(X_{1:n}, Y_{1:n})$ as

$$p_\theta(x_{1:n}, y_{1:n}) = \mu_\theta(x_1) p_{\theta,1}(y_1|x_1) \prod_{t=2}^{n} f_{\theta,1}(x_t|x_{t-1}) p_{\theta,t}(y_t|x_{1:t}, y_{1:t-1})$$

where the density $p_{\theta,n}(y_n|x_{1:n}, y_{1:n-1})$ is given by

$$p_{\theta,n}(y_n|x_{1:n}, y_{1:n-1}) = \int p_\theta(z_{1:n-1}|x_{1:n-1}, y_{1:n-1}) f_{\theta,2}(z_n|x_n, z_{n-1}) g_\theta(y_n|x_n, z_n) \lambda_2(dz_{1:n}). \tag{3.6}$$

The reason $\{X_n, Y_n\}_{n\geq 1}$ might be of interest is that the conditional laws of $Z_{1:n}$ may be available in close form and exact evaluation of the integral in (3.6) is available. In that case, it can be more effective to perform Monte Carlo approximation for the law of $X_{1:n}$ given observations $Y_{1:n}$, which leads to the so called *Rao-Blackwellised particle filters* in the literature [Doucet et al., 2000a].

The integration is indeed available in close form for some time series models. One example is the *linear Gaussian switching state space models* [Chen and Liu, 2000; Doucet et al., 2000a; Fearnhead and Clifford, 2003], where X_n takes values on a finite set whose elements are often called 'labels', and conditioned on $\{X_n\}_{n\geq 1}$, $\{Z_n, Y_n\}_{n\geq 1}$ is a linear Gaussian state-space model. A more sophisticated time series model of the same nature is *linear Gaussian multiple target tracking models*, which we will investigate in detail in Chapter 5.

Having stated that the interest of this thesis is on more general time series models than HMMs, we note that the computational tools developed for HMMs are generally applicable to a more general class of time series models with some suitable modifications. For this reason we carry on this chapter with review of SMC and parameter estimation methods for HMMs.

3.3 Sequential inference in HMMs

3.3.1 Bayesian optimal filtering

In a HMM, one is usually interested in sequential inference on the variables of the hidden process $\{X_t\}_{t\geq 1}$ given observations $Y_{1:n} = y_{1:n}$ up to time n. For example, one pursues for the sequence of posterior distributions $\{p_\theta(x_{1:n}|y_{1:n})\}_{n\geq 1}$, where $p_\theta(x_{1:n}|y_{1:n})$ is given in equation (3.5). It is also straightforward to generalise $p_\theta(x_{1:n}|y_{1:n})$ to the posterior distributions of $X_{1:n'}$ for any $n' \geq 1$. For $n' > n$ we have

$$p_\theta(x_{1:n'}|y_{1:n}) = p_\theta(x_{1:n}|y_{1:n}) \prod_{t=n+1}^{n'} f_\theta(x_t|x_{t-1});$$

whereas for $n' < n$ the density $p_\theta(x_{1:n'}|y_{1:n})$ can be obtained simply by integrating out the variables $x_{n'+1:n}$, i.e.

$$p_\theta(x_{1:n'}|y_{1:n}) = \int p_\theta(x_{1:n}|y_{1:n})\lambda(dx_{n'+1:n}).$$

It is possible to obtain a recursion for these posterior distributions as one receives observations sequentially. Equations (3.3) and (3.5) reveal that we can write $p_\theta(x_{1:n}|y_{1:n})$ in terms of $p_\theta(x_{1:n-1}|y_{1:n-1})$ as

$$p_\theta(x_{1:n}|y_{1:n}) = \frac{f_\theta(x_n|x_{n-1})g_\theta(y_n|x_n)}{p_\theta(y_n|y_{1:n-1})} p_\theta(x_{1:n-1}|y_{1:n-1}). \tag{3.7}$$

The normalising constant $p_\theta(y_n|y_{1:n-1})$ can be written in terms of the known densities as

$$p_\theta(y_n|y_{1:n-1}) = \int p_\theta(x_{1:n-1}|y_{1:n-1}) f_\theta(x_n|x_{n-1}) g_\theta(y_n|x_n)\lambda(dx_{1:n}). \tag{3.8}$$

Also, by convention $p_\theta(y_1|y_0) = p_\theta(y_1) = \int g_\theta(y_1|x_1)\mu_\theta(x_1)\lambda(dx_1)$. The recursion in (3.7) is essential since it enables efficient sequential approximation of the distributions $p_\theta(x_{1:n}|y_{1:n})$ as we will see in Section 3.3.2.

From a Bayesian point of view, the probability densities $p_\theta(x_{1:n'}|y_{1:n})$ are complete solutions to the inference problems as they contain all the information about the hidden states $X_{1:n'}$ given the observations $y_{1:n}$. For example, the expectation of a measurable function $\varphi_{n'} : \mathcal{X}^{n'} \to \mathbb{R}^{d_\varphi(n')}$ conditional upon the observations $y_{1:n}$ can be evaluated as

$$\mathbb{E}_\theta\left[\varphi_n(X_{1:n'})|y_{1:n}\right] = \int \varphi(x_{1:n'}) p_\theta(x_{1:n'}|y_{1:n})\lambda(dx_{1:n'}).$$

However, one can restrict her focus to a problem of smaller size, such as the marginal

distribution of the random variable X_k, $k \leq n'$, given $y_{1:n}$. The probability density of such a marginal posterior distribution $p_\theta(x_k|y_{1:n})$ is called a *smoothing, filtering or prediction* density if $k < n$, $k = n$ and $k > n$, respectively. Indeed, there are many cases where one is interested in calculating the expectations of functions $\varphi : \mathcal{X} \to \mathbb{R}^{d_\varphi}$ of X_k given $y_{1:n}$

$$\mathbb{E}_\theta \left[\varphi(X_k)|y_{1:n}\right] = \int \varphi(x_k) p_\theta(x_k|y_{1:n}) \lambda(dx_k).$$

Although one we have $p_\theta(x_{1:n'}|y_{1:n})$ for $n' \geq k$ the marginal density can directly be obtained by marginalization, the recursion in (3.7) may be intractable or too expensive to calculate. Therefore it is useful to use alternative recursion techniques to effectively evaluate the marginal densities sequentially. Here, we will cover the recursions for the filtering and one-step prediction densities. Given the filtering density $p_\theta(x_{n-1}|y_{1:n-1})$ at time $n-1$, the filtering density at time n is usually obtained recursively in two stages, which are called prediction and update. These are given as

$$p_\theta(x_n|y_{1:n-1}) = \int f_\theta(x_n|x_{n-1}) p_\theta(x_{n-1}|y_{1:n-1}) \lambda(dx_{n-1}), \tag{3.9}$$

$$p_\theta(x_n|y_{1:n}) = \frac{g_\theta(y_n|x_n) p_\theta(x_n|y_{1:n-1})}{p_\theta(y_n|y_{1:n-1})}. \tag{3.10}$$

where this time we write the normalising constant as

$$p_\theta(y_n|y_{1:n-1}) = \int p_\theta(x_n|y_{1:n-1}) g_\theta(y_n|x_n) \lambda(dx_n). \tag{3.11}$$

The problem of evaluating the recursion given by equations (3.9) and (3.10) is called the *Bayesian optimal filtering* (or shortly *optimum filtering*) problem in the literature. In the following, we will look at the SMC methodology in the context of HMMs and review how SMC methods have been used to provide approximate solutions to the optimal filtering problem.

3.3.2 Particle filters for optimal filtering

There are cases when the optimum filtering problem can be solved exactly. One such case is when $\mathcal{X}$ is a finite countable set [Rabiner, 1989]. Also, in linear Gaussian state-space models the densities in (3.9) and (3.10) are obtained by the *Kalman filter* [Kalman, 1960]. In general, however, these densities do not admit a close form expression and one has to use methods based on numerical approximations. One such approach is to use grid-based methods, where the continuous $\mathcal{X}$ is approximated by its finite discretised version and the update rules are used as in the case of finite state HMMs. Another approach is *extended Kalman filter* [Sorenson, 1985], which approximates a non-linear transition by a linear

one and performs the Kalman filter afterwards. The method fails if the nonlinearity in the HMM is substantial. An improved approach based on the Kalman filter is the *unscented Kalman filter* [Julier and Uhlmann, 1997], which is based on a deterministic selection of sigma-points from the support of the state distribution of interest such that the mean and the variance of the true distribution are preserved by the sample mean and covariance calculated at these selected sigma-points. All of these methods are deterministic and not capable of dealing with the most general state-space models; in particular they will fail when the dimensions or the nonlinearities increase.

Alternative to the deterministic approximation methods, Monte Carlo can provide a robust and efficient solution to the optimal filtering problem. SMC methods for optimal filtering, also known as *particle filters*, have been shown to produce more accurate estimates than the deterministic methods mentioned [Doucet et al., 2000b; Durbin and Koopman, 2000; Kitagawa, 1996; Liu and Chen, 1998]. Some of the good tutorials on SMC methods for filtering as well as smoothing in HMMs are Doucet et al. [2000b], Arulampalam et al. [2002], Cappé et al. [2007], Fearnhead [2008], and Doucet and Johansen [2009], from the earliest to the most recent. One can also see Doucet et al. [2001] as a reference book, although a bit outdated. Also, the book Del Moral [2004] contains a rigorous review of numerous theoretical aspects of the SMC methodology in a different framework where a SMC method is treated as an interacting particle system associated with the mean field interpretation of a Feynman-Kac flow.

With reference to the Monte Carlo methodology covered in Chapter 2, the filtering problem in state space models can be considered as a sequential inference problem for the sequence of probability distributions $\pi_{\theta,n}$ on the product measurable spaces $(\mathcal{X}_n = \mathcal{X}^n, \mathcal{E}_n = \mathcal{E}^{\otimes(n)})$

$$\pi_{\theta,n}(dx_{1:n}) := p_\theta(x_{1:n}|y_{1:n})\lambda(dx_{1:n}).$$

As we saw Section 2.5, we can perform SIS and SISR methods targeting $\{\pi_{\theta,n}\}_{n\geq 1}$. The SMC proposal distribution at time n, denoted as $q_{\theta,n}$, is designed conditional to the observations up to time n and state values up to time $n-1$; and in the most general case it can be written as

$$\begin{aligned} q_{\theta,n}(dx_{1:n}) &:= Q_{\theta,1}(y_1, dx_1) \prod_{t=2}^{n} Q_{\theta,t}\left[(x_{1:t-1}, y_{1:t}), dx_t\right] \\ &= q_{\theta,n-1}(dx_{1:n-1}) Q_{\theta,n}\left[(x_{1:n-1}, y_{1:n}), dx_n\right] \end{aligned} \tag{3.12}$$

In fact, most of the time the transition kernel $Q_{\theta,n}$ only depends only on the current observation and the previous state, hence we simplify (3.12) by defining $Q_\theta : \mathcal{X} \times \mathcal{Y} \to \mathcal{P}(\mathcal{E})$ and taking

$$Q_{\theta,n}\left[(x_{1:n-1}, y_{1:n}), dx_n\right] = Q_\theta\left[(x_{n-1}, y_n), x_n\right]$$

for all $n \geq 1$ with the convention $Q_\theta\left[(x_0, y_1), x_1\right] = Q_\theta(y_1, x_1)$. Suppose we design $Q_\theta\left[(x, y), \cdot\right]$ such that it is absolutely continuous with respect to λ with density $q_\theta(\cdot|x, y)$. Therefore, we can write

$$q_{\theta,n}(dx_{1:n}) = \left[q_\theta(x_1|y_1)\prod_{t=2}^{n} q_\theta(x_t|x_{t-1}, y_t)\right]\lambda(dx_{1:n}) \tag{3.13}$$

If we wanted to perform SMC using the target distribution $\pi_{\theta,n}$ directly, then we would have to calculate the following incremental weight at time n

$$\frac{d\pi_{\theta,n}}{d\pi_{\theta,n-1}\otimes Q_\theta}(x_{1:n}) = \frac{f_\theta(x_n|x_{n-1})g_\theta(y_n|x_n)}{p_\theta(y_n|y_{1:n-1})q_\theta(x_n|x_{n-1}, y_n)} \propto \frac{f_\theta(x_n|x_{n-1})g_\theta(y_n|x_n)}{q_\theta(x_n|x_{n-1}, y_n)}.$$

In most of the applications $p_\theta(y_n|y_{1:n-1})$ can not be calculated, hence $\frac{d\pi_{\theta,n}}{d\pi_{\theta,n-1}\otimes Q_\theta}(x_{1:n})$ is not available. For this reason, instead of $\pi_{\theta,n}$ SMC methods use the following unnormalised measure for importance sampling

$$\widehat{\pi}_{\theta,n}(dx_{1:n}) = p_\theta(x_{1:n}, y_{1:n})\lambda(dx_{1:n}),$$

where the normalising constant is $p_\theta(y_{1:n})$, the likelihood of observations up to time n. In that case, the importance weight for the whole path $X_{1:n}$ is given by

$$w_n(x_{1:n}) = w_{n-1}(x_{1:n-1})w_{n|n-1}(x_{n-1}, x_n),$$

where the incremental importance weight $w_{n|n-1}(x_{1:n})$ is

$$w_{n|n-1}(x_{n-1}, x_n) = \frac{f_\theta(x_n|x_{n-1})g_\theta(y_n|x_n)}{q_\theta(x_n|x_{n-1}, y_n)}.$$

Algorithm 3.1. ***SISR (Particle filter) for HMM***
For $n = 1$; for $i = 1, \ldots, N$ sample $X_1^{(i)} \sim q_\theta(\cdot|y_1)$, set $W_1^{(i)} \propto \frac{\mu_\theta(X_1^{(i)})g_\theta(y_1|X_1^{(i)})}{q_\theta(X_1^{(i)}|y_1)}$.
For $n = 2, 3, \ldots$

- *Resample $\{X_{1:n-1}^{(i)}\}_{1\leq i\leq N}$ according to the weights $\{W_{n-1}^{(i)}\}_{1\leq i\leq N}$ to get resampled particles $\{\widetilde{X}_{1:n-1}^{(i)}\}_{1\leq i\leq N}$ with weight $1/N$.*
- *For $i = 1, \ldots, N$; sample $X_n^{(i)} \sim q_\theta(\cdot|\widetilde{X}_{n-1}^{(i)}, y_n)$, set $X_{1:n}^{(i)} = (\widetilde{X}_{1:n-1}^{(i)}, X_n^{(i)})$, and set*

$$W_n^{(i)} \propto \frac{f_\theta(X_n^{(i)}|\widetilde{X}_{n-1}^{(i)})g_\theta(y_n|X_n^{(i)})}{q_\theta(X_n^{(i)}|\widetilde{X}_{n-1}^{(i)}, y_n)}.$$

We present the SISR algorithm, aka the particle filter, for general state-space models in Algorithm 3.1, reminding that SIS is a special type of SISR where there is no resampling.

In the following we list some of the aspects of the particle filter.

- As in the general SISR algorithm, we can use an optional resampling scheme, where we do resampling only when the estimated effective sampling size decreases below a threshold value.

- A by-product of the particle filter is that it can provide unbiased estimates for unknown normalising constants of the target distribution [Del Moral, 2004, Chapter 7]. For example, when SISR is used with an optional sampling scheme, if the last time prior to n when resampling was performed is k, an unbiased estimator of $p_\theta(y_{k+1:n}|y_{1:k})$ can be obtained as

$$p_\theta(y_{k+1:n}|y_{1:k}) \approx \frac{1}{N}\sum_{i=1}^{N}\prod_{t=k+1}^{n} w_{t|t-1}(X_{t-1}^{(i)}, X_t^{(i)}).$$

  We will come back to this aspect of the particle filter in Section 3.4.1.

- The choice of the kernel Q_θ for the proposal distribution in the particle filter is important to ensure effective SMC approximation. The first genuine particle filter in the literature, proposed by Gordon et al. [1993], involved proposing from the prior distribution of $X_{1:n}$, hence taking $q_\theta(x_n|x_{n-1}, y_n) = f_\theta(x_n|x_{n-1})$ and the resulting particle filter with this particular choice of Q_θ is called the *bootstrap filter*. Another interesting choice is to take $q_\theta(x_n|x_{n-1}, y_n) = q_\theta(x_n|y_n)$, which can be useful when observations provide significant information about the hidden state but the state dynamics are weak. This proposal was introduced in Lin et al. [2005] and the resulting particle filter was called *independent particle filter*. The optimal choice that minimises the variance of the incremental importance weights is, from equation (2.12),

$$q_\theta^{opt}(x_n|x_{n-1}, y_n) = p_\theta(x_n|x_{n-1}, y_n).$$

  This results in the optimal incremental weights to be $w_{n|n-1}^{opt}(x_{1:n}) = p_\theta(y_n|x_{n-1})$, which is independent from the value of x_n. First works where q_θ^{opt} was used include Kong et al. [1994]; Liu [1996]; Liu and Chen [1995].

- The auxiliary particle filter for optimal filtering [Pitt and Shephard, 1999] is implemented by sampling $X_{1:n-1}$ among the set of the particle paths up to time $n-1$ and a new X_n from $\mathcal{X}$ in order to target

$$\bar{\pi}_{\theta,n}(dx_{1:n}) = \left[\sum_{i=1}^{N} W_{n-1}^{(i)} w_{n|n-1}^{opt}(X_{1:n-1}^{(i)})\delta_{X_{1:n-1}^{(i)}}(dx_{1:n-1})\right] p_\theta(x_n|x_{n-1}, y_n)\lambda(dx_n).$$

  Note that when $p_\theta(y_n|x_{n-1})$ can be calculated and $p_\theta(x_n|x_{n-1}, y_n)$ is available to

sample from, then all the particles at time n will have equal weights. If this is not the case, the proposal distribution to sample from this target distribution can be written generally as

$$\bar{q}_{\theta,n}(dx_{1:n}) = \left[\sum_{i=1}^{N} \alpha_{n-1}^{(i)} \delta_{X_{1:n-1}^{(i)}}(dx_{1:n-1})\right] q_\theta(x_n|x_{n-1}, y_n)\lambda(dx_n)$$

where $\alpha_{n-1}(x_{n-1})$ and $q_\theta(x_n|x_{n-1}, y_n)$ is up to choice and should be close as possible to the ideal choice. One attempt to make $\alpha_{n-1}^{(i)}$ close to $W_{n-1}^{(i)} p_\theta(y_n|X_{n-1}^{(i)})$ (up to normalising), which was suggested in the original work Pitt and Shephard [1999] on the auxiliary particle filter, is to take $\alpha_{n-1}^{(i)} = g_\theta(y_n|x_n^{*(i)})$, where $x_n^{*(i)}$ is a prediction of X_n given $X_{n-1}^{(i)}$ based on the dynamics of the process, e.g. $x_n^* = \mathbb{E}_\theta[X_n|X_{n-1}^{(i)}]$.

- Although the particle filter we presented in Algorithm 3.1 targets the path filtering distributions $\pi_{\theta,n}(dx_{1:n}) = p_\theta(x_{1:n}|y_{1:n})\lambda(dx_{1:n})$; it can easily be modified, or used directly, to make inference on other distributions that might be of interest. For example, consider the one step path prediction distribution

$$\pi_{\theta,n}^p(dx_{1:n}) = p_\theta(x_{1:n}|y_{1:n-1})\lambda(dx_{1:n}).$$

There is the following relation between $\pi_{\theta,n}$ and $\pi_{\theta,n}^p$.

$$\pi_{\theta,n}^p(dx_{1:n}) = \pi_{\theta,n-1}(dx_{1:n-1}) f_\theta(x_n|x_{n-1})\lambda(dx_n), \quad \frac{d\pi_{\theta,n}}{d\pi_{\theta,n}^p}(x_{1:n}) = \frac{g_\theta(y_n|x_n)}{\pi_{\theta,n}^p(g_\theta(y_n|\cdot))}.$$

Therefore, it is easy to derive approximations to these distributions from each other: obtaining $\pi_{\theta,n}^{p,N}$ from π_{n-1}^N requires a simple extension of the path $X_{1:n-1}$ to $X_{1:n}$ through f_θ; this is done by sampling $X_n^{(i)}$ conditioned on the existing particles paths $X_{1:n-1}^{(i)}$, respectively for $i = 1, \ldots, N$. Whereas; obtaining $\pi_{\theta,n}^N$ from $\pi_{\theta,n}^{p,N}$ requires a simple reweighting of the measure (or the approximate measure) according to $g_\theta(y_n|\cdot)$. As a second example, the approximations to the marginal distributions $\pi_n^N(dx_k)$, $k \leq n$ (or $\pi_n^{p,N}(dx_k)$) are simply obtained from the k'th components of the particles, e.g.

$$\pi_n^N(dx_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}}(dx_{1:n}) \Rightarrow \pi_n^N(dx_k) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_k^{(i)}}(dx_k).$$

Note that the optimal filtering problem corresponds to the case $k = n$. Therefore, it may be sufficient to have a good approximation for the marginal posterior distribution of the current state X_n rather than the whole path $X_{1:n}$. This justifies the resampling step of the particle filter in practice, since resampling trades off accu-

racy for states X_k with $k \ll n$ for a good approximation for the marginal posterior distribution of X_n.

3.3.3 The marginal particle filter

Recall that the standard particle filter follows the recursion in (3.7). It estimates $\pi_{\theta,n}(dx_{1:n})$ by taking an estimate of $\pi_{\theta,n-1}(dx_{1:n-1})$ and augmenting it with x_n at time n. It involves a resampling step not to suffer from high variance which is a result of the sequential nature of the algorithm and that the dimension of the sampled paths is increased by the dimension of the state space at each time. When it is the filtering distribution $\pi_{\theta,n}(dx_n)$ that is desired, one can use a somewhat more principled approach. The *marginal particle filter* (MPF) [Klaas et al., 2005] follows the recursion in (3.9) and (3.10) and performs particle filtering for the marginal distribution $\pi_{\theta,n}(dx_n)$ instead of the joint distribution $\pi_{\theta,n}(dx_{1:n})$.

Assume $\{X_{n-1}^{(i)}, W_{n-1}^{(i)}\}_{1\leq i\leq N}$ is the set of particles and their weights obtained by the MPF for the approximation of $\pi_{\theta,n-1}(dx_{n-1})$. The MPF approximates the recursion in (3.9) and (3.10) by substituting the predictive density $p_\theta(x_n|y_{1:n-1})$ with its approximation $\sum_{i=1}^N W_{n-1}^{(i)} f(x_n|X_{n-1}^{(i)})$ in (3.9). Then it performs importance sampling for the following resulting approximation of the marginal density $p_\theta(x_n|y_{1:n})$

$$p_\theta^N(x_n|y_{1:n}) \propto g_\theta(y_n|x_n) \sum_{i=1}^N W_{n-1}^{(i)} f(x_n|X_{n-1}^{(i)}).$$

Although we have freedom to choose any proposal distribution $q_\theta(x_n|y_{1:n})$ that has appropriate support, the authors in Klaas et al. [2005] suggest a proposal which takes a similar form, namely

$$q_\theta(x_n|y_{1:n}) = \sum_{i=1}^N W_{n-1}^{(i)} q_\theta(x_n|X_{n-1}^{(i)}, y_n). \tag{3.14}$$

Note that the proposal in (3.14) suggests sampling $X_{t-1}^{(i)}$ from the particle estimate of $\pi_{\theta,n-1}(dx_{n-1})$ and then proposing the new component X_t. Instead, we may want to design a proposal that samples particles which will be in high-probability regions of the observation model. We can do this by re-weighting the particles at time $n-1$ to boost them in these regions, and this modification results in the *auxiliary marginal particle filter* (AMPF) [Klaas et al., 2005]. The AMPF is the general version of the MPF where the proposal distribution can be written more generally than (3.14) as

$$q_\theta(x_n|y_{1:n}) = \sum_{i=1}^N \alpha_{n-1}^{(i)} q_\theta(x_n|X_{n-1}^{(i)}, y_n). \tag{3.15}$$

Just as in the auxiliary particle filter in Section 3.3.2, one should ideally take

$$\alpha_{n-1}^{(i)} \propto W_{n-1}^{(i)} p_\theta(y_n|X_{n-1}^{(i)})$$

if calculation of $p_\theta(y_n|X_{n-1}^{(i)})$ is possible; otherwise a suitable approximation of $p_\theta(y_n|X_{n-1}^{(i)})$ should be used instead of $p_\theta(y_n|X_{n-1}^{(i)})$.

The pseudocode for the AMPF is given in Algorithm 3.2. The variance of the importance weights of the AMPF is less than or equal to the variance of the importance weights of the standard auxiliary particle filter. Although this improvement of the marginal particle filter comes with the cost of $\mathcal{O}(N^2)$ calculations per time compared to the $\mathcal{O}(N)$ calculations in standard particle filters; it is possible to reduce this cost to $\mathcal{O}(N \log N)$ with a small and controllable error [Klaas et al., 2005].

Algorithm 3.2. ***The auxiliary marginal particle filter:***
For $n = 1$; for $i = 1, \ldots, N$ sample $X_1^{(i)} \sim q_\theta(\cdot|y_1)$, set $W_1^{(i)} \propto \frac{\mu(X_1^{(i)}) g_\theta(y_1|X_1^{(i)})}{q_\theta(X_1^{(i)}|y_1)}$.
For $n = 2, 3, \ldots$

- *For $i = 1, \ldots, N$; sample $X_n^{(i)} \sim \sum_{i=1}^N \alpha_{n-1}^{(i)} q_\theta(x_n|X_{n-1}^{(i)}, y_n)$ where $\alpha^{(i)}$ is proportional to $W_{n-1}^{(i)} p_\theta(y_n|X_{n-1}^{(i)})$ or to an approximation of it.*
- *For $i = 1, \ldots, N$; set*

$$W_n^{(i)} \propto \frac{g_\theta(y_n|x_n) \sum_{i=1}^N W_{n-1}^{(i)} f_\theta(x_n|X_{n-1}^{(i)})}{\sum_{i=1}^N \alpha_{n-1}^{(i)} q_\theta(X_n^{(i)}|X_{n-1}^{(i)}, y_n)}.$$

Finally, we note that another $\mathcal{O}(N^2)$ particle filter can be found in Lin et al. [2005] as a special case of what the authors call the *independent particle filter*. The name 'independent' is due to their proposal distribution at time n being independent of x_{n-1}, and this allows multiple matching with the previous particles which makes their algorithm $\mathcal{O}(N^2)$ in case of complete matching. Moreover; a slight extension of their algorithm where the proposal distribution uses the past particles is also mentioned in their work, and the MPF or AMPF can be considered to be equivalent to special cases of this extension.

3.3.4 The Rao-Blackwellised particle filter

Assume we are given a HMM $\{(X_n, Z_n), Y_n\}_{n\geq 1}$ where this time the hidden state at time n is composed of two components X_n and Z_n. Suppose that the initial and transition distributions of the Markov chain $\{X_n, Z_n\}_{n\geq 1}$ have densities μ_θ and f_θ with respect to the product measure $\lambda_1 \otimes \lambda_2$ and they can be factorized as follows

$$\mu_\theta(x_1, z_1) = \mu_{\theta,1}(x_1)\mu_{\theta,2}(z_1|x_1), \quad f_\theta(x_n, z_n|x_{n-1}, z_{n-1}) = f_{\theta,1}(x_n|x_{n-1}) f_{\theta,2}(z_n|x_n, z_{n-1}).$$

Also, conditioned on (x_n, z_n) the distribution of observation Y_n admit a density $g_\theta(\cdot|x_n, z_n)$ with respect to ν. We are interested in the case where the posterior distribution

$$\pi_{\theta,n}(dx_{1:n}dz_{1:n}) = p_\theta(x_{1:n}, z_{1:n}|y_{1:n})\lambda_1(dx_{1:n})\lambda_2(dz_{1:n})$$

is analytically intractable and we are interested in approximating the expectations $\pi_{\theta,n}(\varphi_n) = \mathbb{E}_\theta\left[\varphi_n(X_{1:n}, Z_{1:n})|y_{1:n}\right]$ for bounded measurable functions $\varphi_n : \mathcal{X}^n \times \mathcal{Z}^n \to \mathbb{R}^{d_\varphi(n)}$. Obviously, one way to do this is to run an SMC filter for $\{\pi_{\theta,n}\}_{n\geq 1}$ which obtains the approximation $\pi^N_{\theta,n}$ at time n as

$$\pi^N_{\theta,n}(dx_{1:n}dz_{1:n}) = \sum_{i=1}^N W_n^{(i)}\delta_{(X_{1:n}^{(i)}, Z_{1:n}^{(i)})}(dx_{1:n}dz_{1:n}), \quad \sum_{i=1}^N W_n^{(i)} = 1.$$

However, if the conditional posterior probability distribution

$$\pi_{\theta,2,n}(dz_{1:n}|x_{1:n}) = p_\theta(z_{1:n}|x_{1:n}, y_{1:n})\lambda_2(dz_{1:n})$$

is analytically tractable, there is a better SMC scheme for approximating $\pi_{\theta,n}$ and estimating $\pi_{\theta,n}(\varphi_n)$. This SMC scheme is called the *Rao Blackwellised particle filter* (RBPF) [Doucet et al., 2000a]. Consider the following decomposition which follows from the chain rule

$$p_\theta(x_{1:n}, z_{1:n}|y_{1:n}) = p_\theta(x_{1:n}|y_{1:n})p_\theta(z_{1:n}|x_{1:n}, y_{1:n})$$

and define the marginal posterior distribution of $X_{1:n}$ conditioned on $y_{1:n}$ as

$$\pi_{\theta,1,n}(dx_{1:n}) = p_{\theta,1}(x_{1:n}|y_{1:n})\lambda_1(dx_{1:n}).$$

The RBPF is a particle filter for the sequence of marginal distributions $\{\pi_{\theta,1,n}\}_{n\geq 1}$ which produces at time n the approximation

$$\pi^N_{\theta,1,n}(dx_{1:n}) = \sum_{i=1}^N W_{1,n}^{(i)}\delta_{X_{1:n}^{(i)}}(dx_{1:n}), \quad \sum_{i=1}^N W_{1,n}^{(i)} = 1.$$

and the Rao-Blackwellised approximation the full posterior distribution involves the particle filter estimate $\pi^N_{\theta,1,n}$ and the exact distribution $\pi_{\theta,2,n}$

$$\pi^{\mathrm{RB},N}_{\theta,n}(dx_{1:n}dz_{1:n}) = \pi^N_{\theta,1,n}(dx_{1:n})\pi_{\theta,2,n}(dz_{1:n}|x_{1:n}).$$

Then, the estimator of the the RBPF for $\pi_{\theta,n}(\varphi_n)$ becomes

$$\pi^{\mathrm{RB},N}_{\theta,n}(\varphi_n) = \pi^N_{\theta,1,n}\left(\pi_{\theta,2,n}\left[\varphi_n(X_{1:n}, \cdot)\right]\right) = \sum_{i=1}^N W_{1,n}^{(i)}\pi_{\theta,2,n}[\varphi_n(X_{1:n}^{(i)}, \cdot)].$$

Assuming $q_\theta(x_{1:n}|y_{1:n}) = q_\theta(x_{1:n-1}|y_{1:n-1})q_\theta(x_n|x_{1:n-1}, y_{1:n})$ is used as the proposal distribution, the incremental importance weight for the RBPF is given by

$$w_{1,n|n-1}(x_{1:n}) = \frac{f_{\theta,1}(x_n|x_{n-1})p_\theta(y_n|x_{1:n}, y_{1:n-1})}{q_\theta(x_n|x_{1:n-1}, y_{1:n})}$$

where the density $p_\theta(y_n|x_{1:n}, y_{1:n-1})$ is given by

$$p_{\theta,n}(y_n|x_{1:n}, y_{1:n-1}) = \int p_\theta(z_{1:n-1}|x_{1:n-1}, y_{1:n-1}) f_{\theta,2}(z_n|x_n, z_{n-1}) g_\theta(y_n|x_n, z_n) \lambda_2(dz_{1:n}).$$

Also, the optimum importance density which reduces the variance of $w_{1,n|n-1}$ is when the incremental importance density $q_\theta(x_n|x_{1:n-1}, y_{1:n})$ is taken to be $p_\theta(x_n|x_{1:n-1}, y_{1:n})$ which results in $w_{1,n|n-1}(x_{1:n})$ being equal to $p_\theta(y_n|x_{1:n-1}, y_{1:n-1})$.

The use of the RBPF whenever it is possible is intuitively justified by the fact that we substitute particle approximation of some expectations with their exact values. Indeed, the theoretical analysis in Doucet et al. [2000a] and Chopin [2004, Proposition 3] revealed that the RBPF has better precision than the regular particle filter: the estimates of the RBPF never have larger variances. The favouring results for the RBPF are basically due to the Rao-Blackwell theorem (see e.g. Blackwell [1947]), after which the proposed particle filter gets its name.

The RBPF was formulated by Doucet et al. [2000a] and have been implemented in various settings by Andrieu and Doucet [2002]; Chen and Liu [2000]; Särkkä et al. [2004] among many. We will also use RBPFs in our works presented in Chapters 4, 5, and 7.

The use of Rao-Blackwellisation is not limited to marginalising out one of the components of the hidden state; it may be possible to use Rao-Blackwellisation in the intermediate steps of a particle filter. In some time series models, an exact sequential inference is not tractable but the exact one-step update of distributions conditioned on the approximations made prior to the current time is possible. For such models, one can calculate an expectation of interest using this exact one-step update that is available, and then continue by approximating this exact update with particles in order to be able to proceed to the next time step of the particle filter. For examples of such implementation of Rao-Blackwellisation, see Fearnhead and Clifford [2003, p. 890], Fearnhead and Liu [2007], and the Algorithm in Chapter 4 of this thesis.

3.3.5 Application of SMC to smoothing additive functionals

In this section, we provide an example for use of particle filters which is central to this thesis due to its relation to parameter estimation. We are interested in approximating smoothed estimates of additive functionals of state variables in a fully dominated HMM $\{X_n, Y_n\}_{n\geq 1}$ defined in Definition 3.2. Let us have a sequence of functions $s_t : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,

$t \geq 1$ and let $S_n : \mathcal{X}^n \to \mathbb{R}$, $n \geq 1$ be the corresponding sequence of *additive functionals* constructed from s_t as follows

$$S_n(x_{1:n}) = \sum_{t=1}^{n} s_t(x_{t-1}, x_t)$$

where, by convention, we take $s_1(x_0, x_1) = s_1(x_1)$. In many instances it is necessary to be able to compute the following expectations sequentially

$$S_n^\theta = \pi_{\theta,n}(S_n) = \mathbb{E}_\theta \left[S_n(X_{1:n}) | y_{1:n} \right] = \int S_n(x_{1:n}) p_\theta(x_{1:n}|y_{1:n}) \lambda(dx_{1:n}).$$

The expectation is to be computed with respect to the density $p_\theta(x_{1:n}|y_{1:n})$ and for this reason S_n^θ is referred to as a *smoothed additive functional*. Calculation of S_n^θ might be of interest for its own sake, it is also necessary for computing the filter derivative and the gradient of the log-likelihood of observations [Del Moral et al., 2011; Poyiadjis et al., 2011], the intermediate function of the expectation-maximisation algorithm (see e.g. Del Moral et al. [2009]), etc.

In most cases exact computation of S_n^θ is not available due to the unavailability of $p_\theta(x_{1:n}|y_{1:n})$, therefore one has to use Monte Carlo methods, specifically SMC. The first SMC method in the literature proposed to approximate S_n^θ uses the *path space approximation* of $\pi_{\theta,n}$ directly [Cappé, 2009]. Let the SMC approximation of $\pi_{\theta,n}$ be

$$\pi_{\theta,n}^N(dx_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}}(dx_{1:n}), \quad \sum_{i=1}^{N} W_n^{(i)} = 1. \tag{3.16}$$

Then, one obtains the path space approximation of the smoothed additive functional as

$$\widehat{S}_n^\theta = \pi_{\theta,n}^N(S_n) = \sum_{i=1}^{N} W_n^{(i)} S_n(X_{1:n}^{(i)}) \tag{3.17}$$

Observing $S_n(x_{1:n}) = S_{n-1}(x_{1:n-1}) + s_n(x_{n-1}, x_n)$, this approximation can be calculated online for n along with the particle filter, see Cappé [2009] for an application exploiting this fact. In this approximation, there is no need to store the entire ancestry of each particle and computational cost of calculation of $\widehat{S}_n^\theta$ is linear in the number of particles, i.e. $\mathcal{O}(N)$. However; this approximation relies on the approximation of the joint distribution $\pi_{\theta,n}(dx_{1:n})$ which, as already mentioned in Section 2.5.2, is well-known in the SMC literature to become progressively impoverished as n increases because of the successive resampling steps. Indeed, it was shown in Del Moral and Doucet [2003] that under favourable mixing assumptions, the authors established an upper bound on the $\mathbb{L}^p$ error in the path space estimate in (3.17) which is proportional to $n^2/\sqrt{N}$; and under

similar assumptions it was shown in Poyiadjis et al. [2011] that the asymptotic variance of the path space estimate increases at least quadratically with n.

An $\mathcal{O}(N)$ SMC approach that reduces the variance is *fixed-lag smoothing* [Kitagawa and Sato, 2001] which uses the following approximation

$$p_\theta(x_{1:k}|y_{1:n}) \approx p_\theta(x_{1:k}|y_{1:\min(n,k+\Delta)}), \quad \Delta > 0. \tag{3.18}$$

with the idea that for large enough Δ the error introduced by Δ will be negligible. The SMC implementation of this approximation prevents the particle filter from updating path $X_{1:k}$ beyond time $k+\Delta$ and hence reduces the variance resulting from path degeneracy. However; choosing the lag amount Δ is a difficult task, and this approach introduces a bias to the estimate of S_n^θ which does not vanish asymptotically in N, see Olsson et al. [2008].

3.3.5.1 Forward filtering backward smoothing

A standard alternative to computing S_n^θ is to use SMC approximations of fixed-interval smoothing techniques such as the *forward filtering backward smoothing* (FFBS) algorithm [Doucet et al., 2000b; Godsill et al., 2004]. Let us define the marginal smoothing distributions

$$\eta_{\theta,n,k}(dx_k) := \pi_{\theta,n}(dx_k) = p_\theta(x_k|y_{1:n})\lambda(dx_k)$$

and define the backward transition kernel $M_{\theta,n-1} : \mathcal{X} \to \mathcal{P}(\mathcal{E})$ such that

$$M_{\theta,n-1}(x_n, dx_{n-1}) = p_\theta(x_{n-1}|x_n, y_{1:n-1})\lambda(dx_{n-1}).$$

FFBS relies on the additivity of the functional S_n and that $p_\theta(x_{t-1}, x_t|y_{1:n})\lambda(dx_{t-1}dx_t) = \eta_{\theta,n,t}(dx_t)M_{\theta,t-1}(x_t, dx_{t-1})$ for $t \leq n$, which lead to

$$S_n^\theta = \sum_{t=1}^{n} \left[\eta_{\theta,n,t} \otimes M_{\theta,t-1}\right](s_t) = \sum_{t=1}^{n} \int \eta_{\theta,n,t}(dx_t)M_{\theta,t-1}(x_t, dx_{t-1})s_t(x_{t-1}, x_t).$$

Moreover, once $\pi_{\theta,1}, \ldots, \pi_{\theta,n}$ are obtained up to time n (forward filtering), $\eta_{\theta,n,1}, \ldots, \eta_{\theta,n,n}$ can be obtained with a backward recursion (backward smoothing) starting from $\eta_{\theta,n,n}(dx_n) = \pi_{\theta,n}(dx_n)$ and recursing back with

$$\eta_{\theta,n,t} = \eta_{\theta,n,t+1}M_{\theta,t}, \quad t = n-1, \ldots, 1.$$

The SMC implementation of FFBS [Doucet et al., 2000b], which we will call SMC-

FFBS, is based on the following alternative approximation to $\pi_{\theta,n}$

$$\pi_{\theta,n}^{*,N} = \eta_{\theta,n,n}^N \otimes M_{\theta,n-1}^N \otimes \ldots \otimes M_{\theta,1}^N \tag{3.19}$$

where the particle approximation to the backward kernels are

$$M_{\theta,n-1}^N(x_n, dx_{n-1}) = \eta_{\theta,n-1,n-1}^N(dx_{n-1}) \frac{f_\theta(x_n|x_{n-1})}{\int \eta_{\theta,n-1,n-1}^N(dx_{n-1}) f_\theta(x_n|x_{n-1}) \lambda(dx_n)}. \tag{3.20}$$

Therefore, once $\pi_{\theta,1}^N, \ldots, \pi_{\theta,n}^N$ are obtained up to time n (forward filtering), $\eta_{\theta,n,1}^N, \ldots, \eta_{\theta,n,n}^N$ can be obtained with a backward recursion (backward smoothing) starting from $\eta_{\theta,n,n}^N(dx_n) = \pi_{\theta,n}^N(dx_n)$ and recursing back with

$$\eta_{\theta,n,t}^N = \eta_{\theta,n,t+1}^N M_{\theta,t}^N, \quad t = n-1, \ldots, 1.$$

Then, the SMC approximation to FFBS leads to the following estimate of the smoothed functional

$$\widehat{S}_n^{*,\theta} = \sum_{t=1}^n \left[\eta_{\theta,n,t}^N \otimes M_{\theta,t-1}^N\right](s_t).$$

The SMC implementation of FFBS requires $\mathcal{O}(N^2)$ computations per time, compared to the $\mathcal{O}(N)$ path space approximation. As a return, the estimator has better properties over the estimator of the path space approximation. Douc et al. [2011] includes a central limit theorem for $\widehat{S}_n^{*,\theta}$ and time uniform deviation inequalities for the SMC-FFBS approximations of the marginals $\{\eta_{\theta,n,t}\}_{1\leq t\leq n}$. For alternative proofs to those in Douc et al. [2011], see Del Moral et al. [2010]. Additionally, it was shown in Del Moral et al. [2009] that under strong mixing conditions the asymptotic variance of $\widehat{S}_n^{*,\theta}$ as $N \to \infty$ is linear in n. More general but more complicated results on the variance of $\widehat{S}_n^{*,\theta}$ with weaker conditions can be found in Del Moral et al. [2010].

3.3.5.2 Forward-only smoothing

Filtering forwards and smoothing backwards, the FFBS algorithm is surely offline, unlike the path space approximation. Also, it may be demanding since it requires the SMC filters $\eta_{\theta,t,t}^N$ to be stored up to time n. To circumvent the need for the backward pass in the computation of S_n^θ, the following auxiliary function on $\mathcal{X}$ is introduced,

$$T_n^\theta(x_n) = M_{n-1} \otimes \ldots \otimes M_1 \left[S_n(\cdot, x_n)\right](x_n) = \mathbb{E}_\theta \left[S_n(X_{1:n})|X_n = x_n, y_{1:n-1}\right].$$

It is apparent that $S_n^\theta = \eta_{\theta,n,n}(T_n^\theta)$. A forward recursion to compute $\{T_n^\theta\}_{n\geq 1}$, hence $\{S_n^\theta\}_{n\geq 1}$, is established by

$$T_n^\theta(x_n) = M_{\theta,n-1}\left[T_{n-1}^\theta + s_n(\cdot, x_n)\right] = \mathbb{E}_\theta\left[T_{n-1}^\theta(X_{n-1}) + s_n(X_{n-1}, x_n)\middle| x_n, y_{1:n-1}\right]. \tag{3.21}$$

for $n \geq 2$, with the initial condition $T_n^\theta(x_1) = s_1(x_1)$. Note that online calculation of $T_n^\theta(x_n)$ requires only an integration with respect to the measure $M_{n-1}(x_n, \cdot)$, i.e. $p_\theta(x_{n-1}|x_n, y_{1:n-1})$. The recursion in (3.21) has been rediscovered independently several times (see e.g. Elliott and Krishnamurthy [1999]; Hernando et al. [2005]; Mongillo and Deneve [2008]) and it was called *forward smoothing recursion* in Del Moral et al. [2009].

A straightforward implementation of forward smoothing recursion would be by using $\pi_{\theta,n}^{*,N}$ in (3.19) so that $M_{\theta,n-1}$ in (3.21) is approximated by $M_{\theta,n-1}^N$ in (3.20). It can be shown that when this approximation is used, we calculate exactly the same quantity as SMC-FFBS. Therefore, the preferable statistical properties of SMC-FFBS is preserved. Moreover, although the online calculation still requires $\mathcal{O}(N^2)$ calculations per time, it does not need to store the SMC filters $\{\eta_{\theta,n,t}\}_{1\leq t\leq n}$. Being a forward implementation of SMC-FFBS, we call this implementation SMC-forward smoothing, or SMC-FS. SMC-FS proves to be a very useful tool for online parameter estimation, as we shall see in Section 3.4 and throughout this thesis.

Algorithm 3.3. ***SMC-FS: Forward only SMC computation of FFBS for smoothing additive functionals***

For $n = 1$;

- *compute the SMC approximation $\{X_1^{(i)}, W_1^{(i)}\}_{1\leq i\leq N}$ for $\eta_{\theta,1,1}$.*
- *For $i = 1, \ldots, N$, set $T_1^{(i)} = s_1(X_1^{(i)})$.*

For $n = 2, 3, \ldots$

- *Compute the SMC approximation $\{X_n^{(i)}, W_n^{(i)}\}_{1\leq i\leq N}$ for $\eta_{\theta,n,n}$.*
- *For $i = 1, \ldots, N$; set*

$$T_n^{(i)} = \frac{\sum_{j=1}^N \left[T_{n-1}^{(j)} + s_n(X_{n-1}^{(j)}, X_n^{(i)})\right] W_{n-1}^{(j)} f_\theta(X_n^{(i)}|X_{n-1}^{(j)})}{\sum_{j'=1}^N W_{n-1}^{(j')} f_\theta(X_n^{(i)}|X_{n-1}^{(j')})}.$$

- *Calculate $\widehat{S}_n^{*,\theta} = \sum_{i=1}^N W_n^{(i)} T_n^{\theta(i)}$.*

We present the SMC-FS algorithm in Algorithm 3.3. We note that; since SMC-FS relies on particle estimates of the filtering distributions $\{\eta_{\theta,n,n}\}_{n\geq 1}$ only, the marginal

particle filter in Section 3.3.3 can be used in Algorithm 3.3 the instead of the standard particle filter. Finally, note that the SMC implementation of the forward smoothing recursion by using the path space approximation is trivial in the sense that it reduces to the approximation given in (3.16).

3.4 Static parameter estimation in HMMs

One problem that is largely dealt in the literature is that of estimating the *true* static parameter θ^* of the HMM given observations $y_{1:n}$ up to time n. There are two main approaches to solving the parameter estimation problem, the Bayesian approach and the maximum likelihood approach. We briefly summarise Bayesian methods and then give a more detailed review on the maximum likelihood parameter estimation methods. We refer the interested reader to Kantas et al. [2009] for a comprehensive review of SMC methods that have been proposed for static parameter estimation in HMMs.

Bayesian parameter estimation: In the Bayesian approach, the static parameter is treated as a random variable taking values θ in Θ with a probability density $\eta(\theta)$ with respect to a dominating measure $d\theta$, and the aim is to evaluate the density of the posterior distribution of θ given $y_{1:n}$, which follows from Bayes' theorem as

$$\eta(\theta|y_{1:n}) = \frac{\eta(\theta)p_\theta(y_{1:n})}{\int p_\theta(y_{1:n})\eta(\theta)d\theta}. \tag{3.22}$$

When the likelihood $p_\theta(y_{1:n})$ is analytically available, one can simply apply a MCMC scheme for the posterior $\eta(\theta|y_{1:n})$. An MCMC algorithm can be inefficient when n is large; however online Bayesian methods are also available. For example, the method in Chopin [2002] is based on the SMC approximation of the sequence of distributions $\{\eta(d\theta|y_{1:t})\}_{1\leq t\leq n}$. This approach is equivalent to the resample-move algorithm described in Gilks and Berzuini [2001], which is a special SMC sampler.

More sophisticated techniques are required when $p_\theta(y_{1:n})$ cannot be computed, which is usually the case for a general HMM. The methods developed for this case consider

- the joint density $p(\theta, x_{1:n}|y_{1:n}) = \eta(\theta)p_\theta(x_{1:n}|y_{1:n})$ in the batch parameter estimation setting
- the sequence of posterior densities $\{p_\theta(\theta, x_{1:t}|y_{1:t})\}_{1\leq t\leq n}$ in the online parameter estimation setting.

One elegant method used in the batch estimation setting is *particle MCMC* (PMCMC) [Andrieu et al., 2010]. Notice that an ideal Metropolis-Hastings algorithm targeting $p_\theta(\theta, x_{1:n}|y_{1:n})$ is not feasible in general since it requires exact sampling from $p_\theta(x_{1:n}|y_{1:n})$

and exact calculation of $p_\theta(y_{1:n})$. A particle version of the Metropolis-Hastings algorithm, which was called PMMH in Andrieu et al. [2010], runs an SMC for $p_\theta(x_{1:n}|y_{1:n})$ with N particles and uses the SMC approximation of the unknown quantity $p_\theta(y_{1:n})$. The validity of this approach is not trivial to show; see, again, Andrieu et al. [2010] for a derivation. In the same work, a particle version of the Gibbs sampler was also developed.

In Andrieu et al. [2010] the variance of the acceptance rate of the PMMH algorithm was numerically shown to be proportional to n/N under favourable mixing conditions. This suggests that one needs to increase the number of particles linearly with n in order to keep the performance of the PMCMC algorithm at a certain level. Therefore, for large n PMCMC may not be practical and online parameter estimation methods may be required.

Although with possible modifications, all of the Bayesian methods for online parameter estimation rely on the SMC approximation of the sequence of distributions $p_\theta(\theta, x_{1:t}|y_{1:t})$, $1 \leq t \leq n$. At first sight, it seems easy to achieve this using standard SMC methods by introducing the extended state $\{\theta_n, X_n\}_{n\geq 1}$ with the initial distribution $\mu_{\theta_1}(x_1)\lambda(dx_1)\eta(\theta_1)d\theta$ and transitional distribution $f_{\theta_n}(x_n|x_{n-1})\lambda(dx_n)\delta_{\theta_{n-1}}(d\theta_n)$. This implies $\theta_n = \theta_{n-1}$; therefore an SMC algorithm explores the parameter space only at its initialisation. As a result of successive resampling steps, we will end up with only a single value for θ, which makes the approximation to the marginal distribution $\eta(d\theta|y_{1:n})$ clearly a bad approximation. Several methods have been proposed to avoid degeneracy of particles for the static parameter of the HMM. We briefly mention them below.

One approach to avoid degeneracy, proposed originally in Gilks and Berzuini [2001], is based on adding MCMC steps to re-introduce particle diversity. Assume that the SMC approximation to $p(\theta, x_{1:n}|y_{1:n})$ at time n contains particles $(\theta_n^{(i)}, X_{1:n}^{(i)})$, $i = 1, \ldots, N$, with equal weights. To add diversity in this population, a MCMC kernel K_n which leaves $p(d(\theta, x_{1:n})|y_{1:n})$ invariant is applied to each of the particles. One remarkable point is that the MCMC kernel need not be ergodic; indeed in practice one designs K_n so that it moves only $\theta^{(i)}$ and last L components of $X_{1:n}^{(i)}$. A first use of this method in an online Bayesian parameter estimation context is seen in Andrieu et al. [1999], K_n is taken to be a Gibbs move to update the parameter value only, i.e.

$$K_n\left[(x_{1:n}, \theta), d(x'_{1:n}, \theta')\right] = \delta_{x_{1:n}}(dx'_{1:n})p(\theta|x_{1:n}, y_{1:n})d\theta$$

Similar strategies were used in Fearnhead [2002] and Storvik [2002]. The use of MCMC within SMC steps is particularly elegant when $(x_{1:n}, y_{1:n})$ can be summarised by a set of fixed dimensional sufficient statistics; since then the memory and computational requirements for calculating densities such as $p_\theta(y_{1:n}|x_{1:n})$ or $p(\theta|x_{1:n}, y_{1:n})$ does not increase with time. Unfortunately; these MCMC-based methods suffer from the path degeneracy problem of the SMC approximation, since the error in the estimate of $p_\theta(x_{1:n}|y_{1:n})$ will

lead to an error in sufficient statistics to be used and these errors build up over time. This disadvantage was first noticed in Andrieu et al. [1999] and a convincing example was provided in Andrieu et al. [2005].

Another MCMC-based online Bayesian estimation method is called *practical filtering* [Polson et al., 2008], which relies on a fixed-lag approximation as in (3.18). As for all fixed-lag approaches, it is hard to tune the amount of lag and control the non-vanishing bias introduced by the approximation.

Alternative to MCMC-based methods to avoid degeneracy, another class of methods are based on introducing artificial dynamics for the parameter [Higuchi, 2001; Kitagawa, 1998]. More explicitly, it is assumed that

$$\theta_1 \sim \eta(\theta_1), \quad \theta_n = \theta_{n-1} + \epsilon_n, \quad n \geq 2,$$

where ϵ_n is a small artificial dynamic centred noise whose variance is decreasing with n. Obviously, SMC applied to approximate $\{p(\theta_n, x_{1:n}|y_{1:n})\}_{n\geq 1}$ under this assumption will have better properties than before in terms of degeneracy. This approach is closely related to the kernel density estimation method in Liu and West [2001], which proposes regularising smoothing the empirical measure of the posterior distribution of the parameter with a smooth kernel density, such as Gaussian or Epanechnikov. A more general approach where the kernel smoothing approach is also applied to the components of the HMM is given in Campillo and Rossi [2009]. All these methods who introduce artificial dynamics to the parameter require a significant amount of tuning and it suffers from bias which is hard to quantify.

Maximum likelihood parameter estimation: In the maximum likelihood approach to parameter estimation, one has a point estimate obtained by calculating the value of θ that maximises the likelihood $p_\theta(y_{1:n})$ over all the possible values of θ, i.e.

$$\theta_{\mathrm{ML}} = \arg\max_{\theta\in\Theta} p_\theta(y_{1:n}).$$

This procedure is called maximum likelihood estimation (MLE). In this thesis we will investigate methods for MLE applied to several time series models. In the following we present some of the MLE methods directly applicable to HMMs.

3.4.1 Direct maximisation of the likelihood

The traditional approach of ML is to try to calculate the maximiser of $p_\theta(y_{1:n})$ with respect to θ by direct calculation of $p_\theta(y_{1:n})$. Note that $p_\theta(y_{1:n})$ also satisfies the following

recursive form

$$p_\theta(y_{1:n}) = p_\theta(y_1)\prod_{t=2}^{n} p_\theta(y_t|y_{1:t-1}) = p_\theta(y_{1:n-1})p_\theta(y_n|y_{1:n-1}). \tag{3.23}$$

The incremental likelihood $p_\theta(y_n|y_{1:n-1})$ may be obtained by exploiting one the expressions for it, such as the one in (3.8) or (3.11), whichever is available. In practice, one uses the log-likelihood

$$l_\theta(y_{1:n}) = \log p_\theta(y_{1:n})$$

which is numerically better-behaved since this time the product in (3.23) is replaced by a sum.

It is rarely the case that the likelihood (or log-likelihood) is in closed form and can be maximised analytically. When it is not in closed form but it can be calculated, grid based methods, where the likelihood is calculated on a grid based representation of Θ with enough resolution, can be used. When even the likelihood can not even be calculated, SMC approximation can be applied. Let $\tau_1, \ldots, \tau_k$ be the times when the resampling step is applied in the particle filter in Algorithm 3.1 and let $\tau_0 = 0$ and $\tau_{k+1} = n$. It is shown in Del Moral [2004, Chapter 7] the following estimator of $p_\theta(y_{1:n})$ is unbiased

$$p_\theta^N(y_{1:n}) = \prod_{j=1}^{k+1} p_\theta^N(y_{\tau_{j-1}+1:\tau_j}|y_{1:\tau_{j-1}}), \quad p_\theta^N(y_{\tau_{j-1}+1:\tau_j}|y_{1:\tau_{j-1}}) = \sum_{i=1}^{N}\prod_{t=\tau_{j-1}+1}^{\tau_j} w_{t|t-1}^{(i)}$$

Based on this unbiased estimator, an estimate of $l_\theta(y_{1:n})$ is

$$l_\theta^N(y_{1:n}) = \sum_{j=1}^{k+1} \log p_\theta^N(y_{\tau_{j-1}+1:\tau_j}|y_{1:\tau_{j-1}})$$

which is obviously biased due to the non-linear transformation of the unbiased estimators. The bias can be reduced by using the following standard technique based on a Taylor series expansion, see Andrieu et al. [2004].

Direct maximisation of the likelihood by means of calculating it point-wise is not a practical approach unless Θ is a discrete space with small number of elements or a continuous space which can be well approximated by a grid. Unfortunately, these conditions do not hold in almost all cases mainly because θ is of large dimension. In the following we will review two alternative approaches that maximises $p_\theta(y_{1:n})$ (at least locally) indirectly without calculating it.

3.4.2 Gradient ascent maximum likelihood

Gradient based maximum likelihood methods work with the gradient of the log-likelihood rather than itself. The gradient ascent algorithm is an iterative procedure implemented as follows: We begin with $\theta^{(0)}$ and assume that we have the estimate $\theta^{(j-1)}$ at the end of the the $(j-1)$'th iteration. At the j'th iteration we update the parameter

$$\theta^{(j)} = \theta^{(j-1)} + \gamma_j \nabla_\theta l_\theta(y_{1:n})\big|_{\theta=\theta^{(j-1)}}.$$

The gradient term $\nabla_\theta l_\theta(y_{1:n})$ is also called the *score vector*. Here $\{\gamma_j\}_{j\geq 1}$ is the sequence of step sizes satisfying

$$\sum_{j\geq 0} \gamma_j = \infty, \quad \sum_{j\geq 0} \gamma_j^2 < \infty, \tag{3.24}$$

ensuring convergence of the algorithm when it is used with the Monte Carlo approximations $\nabla_\theta^N l_\theta(y_{1:n})$ of the score vectors. A common choice is $\gamma_n = n^{-a}$ for $0.5 < a \leq 1$.

One way to calculate the gradient term is to use Fisher's identity for the score vector as

$$\nabla_\theta l_\theta(y_{1:n}) = \int p_\theta(x_{1:n}, y_{1:n}) \log p_\theta(x_{1:n}, y_{1:n}) \lambda(dx_{1:n}). \tag{3.25}$$

i.e. the expectation of the complete data log-likelihood with respect to the posterior distribution of the latent variables. Equation (3.25) can be rewritten as

$$\nabla_\theta l_\theta(y_{1:n}) = \pi_{\theta,n}(S_{\theta,n}) \tag{3.26}$$

where $S_{\theta,n} : \mathcal{X}^n \to \mathbb{R}^{d_\theta}$ is the additive function of of $x_{1:n}$

$$\begin{aligned} S_{\theta,n}(x_{1:n}) &= \sum_{t=1}^{n} s_{\theta,t}(x_{t-1}, x_t), \\ s_{\theta,1}(x_0, x_1) &= s_{\theta,1}(x_1) = \nabla_\theta \log \mu_\theta(x_1) + \nabla_\theta \log g_\theta(y_1|x_1) \\ s_{\theta,t}(x_{t-1}, x_t) &= \nabla_\theta \log g_\theta(y_t|x_t) + \nabla_\theta \log f_\theta(x_t|x_{t-1}), \quad t \geq 2. \end{aligned} \tag{3.27}$$

Notice that since $S_{\theta,n}$ is in the additive form, the approximation to its expectation $\pi_{\theta,n}(S_{\theta,n})$ can be carried out with one of the Monte Carlo methods mentioned in Section 3.3.5 when exact calculation of $\pi_{\theta,n}(S_{\theta,n})$ is not available. An SMC estimate of the score vector using the $\mathcal{O}(N)$ path space approximation was provided in Andrieu et al. [2004]. However; it was shown in Poyiadjis et al. [2011] that the variance of this estimate increases typically quadratically with n. For this reason, Poyiadjis et al. [2011] proposed to use the $\mathcal{O}(N^2)$ method that is based on FFBS to estimate $\nabla_\theta l_\theta(y_{1:n})$, and it was shown in Del Moral et al. [2011] that this SMC estimate is stable.

An alternative to Fisher's identity to compute the score vector $\nabla_\theta l_\theta(y_{1:n})$ is a method

based on *infinitesimal perturbation analysis* which was proposed in Coquelin et al. [2009]. This method is also estimating the expectation with respect to $p_\theta(x_{1:n}|y_{1:n})$ of an additive functional of the form $\sum_{t=1}^{n} s_\theta(x_{t-1}, x_t)$; so all the SMC smoothing techniques described in Section 3.3.5 can also be applied to estimate this expectation.

3.4.2.1 Online gradient ascent

The batch gradient ascent MLE algorithm may be inefficient when n is large since each iteration requires a complete browse over the whole data sequence. An alternative to the batch algorithm is possible via online calculation of the score vector, leading to a recursive maximum likelihood algorithm which we will call *online gradient ascent*. An online gradient ascent algorithm can be implemented as follows [Del Moral et al., 2011; Poyiadjis et al., 2011]: Let θ_1 be the initial guess of θ^* before having made any observations and at time n and let $\theta_{1:n}$ be the sequence of parameter estimates of the online gradient ascent algorithm computed sequentially based on $y_{1:n-1}$. When y_n is received, we update the parameter

$$\theta_{n+1} = \theta_n + \gamma_n \nabla_\theta \log p_\theta(y_n|y_{1:n-1})\big|_{\theta=\theta_n}. \tag{3.28}$$

The incremental gradients $\nabla_\theta \log p_\theta(y_n|y_{1:n-1})$ can be calculated sequentially from the gradients $\nabla_\theta l_\theta(y_{1:n})$ using the relation

$$\begin{aligned} \nabla_\theta \log p_\theta(y_n|y_{1:n-1}) &= \nabla_\theta l_\theta(y_{1:n}) - \nabla_\theta l_\theta(y_{1:n-1}) \\ &= \pi_{\theta,n}(S_{\theta,n}) - \pi_{\theta,n-1}(S_{\theta,n-1}). \end{aligned} \tag{3.29}$$

However, since θ_n is changing over time, (3.29) hence (3.28) is impractical to calculate sequentially. In practice, the integrals $\pi_{\theta_n,n}(S_{\theta_n,n})$ are approximated by

$$\pi_{\theta_{1:n},n}\left(\sum_{t=1}^{n} s_{\theta_t}(x_{t-1}, x_t)\right),$$

where $\theta_{1:n}$ in $\pi_{\theta_{1:n},n}$ indicates that the distributions are calculated sequentially with varying θ's.

This approach has previously appeared in the literature for finite state-space HMMs, see e.g. Le Gland and Mevel [1997] and Collings and Ryden [1998]. The asymptotic properties of this algorithm, i.e. the behaviour of θ_n in the limit as n goes to infinity, has been studied by Titterington [1984] for i.i.d. hidden processes and by Le Gland and Mevel [1997] for finite state-space HMMs. It is shown in Le Gland and Mevel [1997] that under regularity conditions this algorithm converges towards a local maximum of the average log-likelihood and that this average log-likelihood is maximised at θ^*.

Algorithm 3.4. ***SMC-online gradient ascent algorithm***

Choose θ_1. *Set* $\mathcal{S}_0 = 0$. *For* $n = 1, 2, \ldots$;

- *If* $n = 1$,
  - *Compute the SMC approximation* $\{X_1^{(i)}, W_1^{(i)}\}_{1 \leq i \leq N}$ *for* $\eta_{\theta_1,1,1}$.
  - *For* $i = 1, \ldots, N$; *for* $k = 1, \ldots, r$ *set* $T_{\gamma,1,k}^{(i)} = \nabla_\theta \log \mu_\theta(X_1^{(i)}) + \nabla_\theta \log g_\theta(y_1 | X_1^{(i)})$.

  if $n \geq 2$,
  - *Compute the SMC approximation* $\{X_n^{(i)}, W_n^{(i)}\}_{1 \leq i \leq N}$ *for* $\eta_{\theta_{1:n},n,n}$.
  - *For* $i = 1, \ldots, N$ *set*

$$T_{\gamma,n}^{(i)} = \frac{\sum_{j=1}^N \left[(1-\gamma_n) T_{\gamma,n-1}^{(j)} + \gamma_n s_n(X_{n-1}^{(j)}, X_n^{(i)})\right] W_{n-1}^{(j)} f_{\theta_n}(X_n^{(i)} | X_{n-1}^{(j)})}{\sum_{j'=1}^N W_{n-1}^{(j')} f_{\theta_n}(X_n^{(i)} | X_{n-1}^{(j')})}.$$

where $s_n(X_{n-1}^{(j)}, X_n^{(i)}) = \nabla_{\theta_n} \log f_{\theta_n}(X_n^{(i)} | X_{n-1}^{(j)}) + \nabla_{\theta_n} \log g_{\theta_n}(y_n | X_n^{(i)})$

- *Calculate* $\mathcal{S}_n = \sum_{i=1}^N W_n^{(i)} T_{\gamma,n}^{(i)}$ *and set* $\theta_{n+1} = \theta_n + \gamma_n (\mathcal{S}_n - \mathcal{S}_{n-1})$.

A SMC online gradient ascent method, which can be seen as a particle version of the recursive maximum likelihood algorithm of Le Gland and Mevel [1997] is given in Algorithm 3.4. This algorithm is based on the $\mathcal{O}(N^2)$ SMC approximation of (3.29) and calculates

$$\theta_{n+1} = \theta_n + \gamma_n \left[\pi_{\theta_{1:n},n}^{*,N} \left(\sum_{t=1}^n s_{\theta_t}(x_{t-1}, x_t) \right) - \pi_{\theta_{1:n-1},n-1}^{*,N} \left(\sum_{t=1}^{n-1} s_{\theta_t}(x_{t-1}, x_t) \right) \right]. \quad (3.30)$$

In Poyiadjis et al. [2011], this algorithm is used with the MPF described in Section 3.3.3 in order to approximate the filtering distributions $\eta_{\theta,n,n}$. A very similar algorithm, which is equivalent to Algorithm 3.4 in principle, can be found in Del Moral et al. [2011]; the difference is that the authors include θ_n in the calculation of the second term in (3.29) by using the relation

$$\pi_{\theta,n-1}(S_{\theta,n-1}) = \widehat{\pi}_{\theta,n}(S_{\theta,n-1} + \nabla_\theta \log f_\theta(x_n | x_{n-1})).$$

We remind that $\widehat{\pi}_{\theta,n}$ is distribution of $X_{1:n}$ conditioned on $y_{1:n-1}$. Hence, (3.30) is replaced by

$$\theta_{n+1} = \theta_n + \gamma_n \left[\pi_{\theta_{1:n},n}^{*,N} \left(\sum_{t=1}^n s_{\theta_t}(x_{t-1}, x_t) \right) - \widehat{\pi}_{\theta_{1:n},n}^{*,N} \left(\nabla_{\theta_n} \log f_{\theta_n}(x_n | x_{n-1}) + \sum_{t=1}^{n-1} s_{\theta_t}(x_{t-1}, x_t) \right) \right].$$

and the online implementation of this update is derived using the filter derivative at time n. Similar to the $\mathcal{O}(N^2)$ particle approximation to $\pi_{\theta,n}(S_{\theta,n})$, the $\mathcal{O}(N^2)$ particle

approximation of $\widehat{\pi}_{\theta,n}(S_{\theta,n})$ can be performed by taking

$$\widehat{\pi}^{*,N}_{\theta,n} = \widehat{\eta}^{N}_{\theta,n,n} \otimes M^{N}_{\theta,n-1} \otimes \ldots \otimes M^{N}_{\theta,1}$$

where $\widehat{\eta}^{N}_{\theta,n,n}(dx_n) = \widehat{\pi}^{N}_{\theta,n}(dx_n)$ is the particle approximation to the one step prediction distribution obtained by marginalising the path particle approximation $\widehat{\pi}^{N}_{\theta,n}$.

3.4.3 Expectation-Maximisation

The *expectation-maximisation* (EM) algorithm [Dempster et al., 1977] is one of the most popular methods for MLE. Given $Y_{1:n} = y_{1:n}$, the EM algorithm for maximising $p_\theta(y_{1:n})$ is given by the following iterative procedure: if $\theta^{(j)}$ is the estimate of the EM algorithm at the jth iteration, then at iteration $j+1$ the estimate is updated by first calculating the following intermediate optimisation criterion, which is known as the expectation (E) step,

$$\begin{aligned} Q(\theta^{(j)}, \theta) &= \int \log p_\theta(x_{1:n}, y_{1:n}) p_{\theta^{(j)}}(x_{1:n}|y_{1:n}) \lambda(dx_{1:n}) \\ &= \mathbb{E}_{\theta^{(j)}} \left[\log p_\theta(X_{1:n}, y_{1:n}) | y_{1:n} \right]. \end{aligned} \tag{3.31}$$

The updated estimate is then computed in the maximisation (M) step

$$\theta^{(j+1)} = \arg\max_{\theta \in \Theta} Q(\theta^{(j)}, \theta) \tag{3.32}$$

The EM algorithm produces a sequence $\{\theta^{(j)}\}_{j \geq 1}$ such that $\{p_{\theta^{(j)}}(y_{1:n})\}_{j \geq 1}$ is non-decreasing, and under mild conditions this sequence is guaranteed to converge to a maximum point of $p_\theta(y_{1:n})$. In practice, the procedure in (3.31) and (3.32) is repeated until $\theta^{(j)}$ ceases to change significantly.

One important observation here is that the integrand in (3.31), which is the joint-log density of the complete data $(x_{1:n}, y_{1:n})$, has the following additive structure.

$$\log p_\theta(x_{1:n}, y_{1:n}) = \mu_\theta(x_1) + \log g_\theta(y_1|x_1) + \sum_{t=2}^{n} \log f_\theta(x_t|x_{t-1}) + \log g_\theta(y_t|x_t) \tag{3.33}$$

Moreover, equation (3.33) suggests that when $p_\theta(x_{1:n}, y_{1:n})$ belongs to the exponential family with respect to θ, then there exist an integer $r > 0$, functions $s_{i,t} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $i = 1, \ldots, r$, $t \geq 1$, such that the E-step and M-step of the EM algorithm reduce to calculating

$$S^{\theta^{(j)}}_{i,n} = \pi_{\theta^{(j)},n}(S_{i,n}) = \mathbb{E}_{\theta^{(j)}} \left[S_{i,n}(X_{1:n}) | \, y_{1:n} \right], \quad S_{i,n}(x_{1:n}) = \sum_{t=1}^{n} s_{i,t}(x_{t-1}, x_t), \quad i = 1, \ldots, r,$$

and applying a maximisation rule $\Lambda : \mathbb{R}^r \to \Theta$ to compute (3.32) such that

$$\theta^{(j+1)} = \Lambda\left(S_{1,n}^{\theta^{(j)}}, \ldots, S_{r,n}^{\theta^{(j)}}\right). \tag{3.34}$$

Functionals $S_{1,n}, \ldots, S_{r,n}$ are also called the sufficient statistics of the complete data $(x_{1:n}, y_{1:n})$.

3.4.3.1 Stochastic versions of EM

The intermediate function $Q(\theta^{(j)}, \theta)$ of the EM algorithm can be computed exactly only in few HMMs such as linear Gaussian HMMs or finite state-space HMMs. When $Q(\theta^{(j)}, \theta)$ cannot be computed exactly, Monte Carlo approximation must be used to numerically estimate it. The additive structure of $\log p_\theta(x_{1:n}, y_{1:n})$ allows us to use several SMC smoothing techniques for estimating $Q(\theta^{(j)}, \theta)$; see Andrieu et al. [2004] for the path space approximation, Olsson et al. [2008] for the fixed-lag approximation, Wills et al. [2008] for the FFBS approximation and Briers et al. [2010] for generalised *two-filter smoothing*.

Using Monte Carlo estimate of the intermediate function leads to the stochastic versions of the EM algorithm. There are three different main stochastic versions of the EM algorithm proposed in the literature, we will review them below.

- If we use a constant number N of particles for all iterations, the resulting algorithm is called the *stochastic EM algorithm* (SEM) [Celeux and Diebolt, 1985]. Since the Monte Carlo variance is never reduced over iterations, this algorithm will not converge to a point in Θ; however one expects to have an ergodic homogeneous Markov chain of estimates $\{\theta^{(j)}\}_{j\geq 0}$ whose stationary distribution is concentrated around θ_{ML} [Nielsen, 2000].

- The settlement of the Markov chain in the SEM algorithm to its equilibrium may take too much time. An alternative to SEM is introduced in Wei and Tanner [1990] and is called *Monte Carlo EM* (MCEM). In MCEM, the number of particles for Monte Carlo approximation increases with j in order to ensure convergence to the maximum likely parameter value θ_{ML} rather than convergence to a stationary distribution around it. The disadvantage of this approach is having to use an increasing amount of computational resource because of the increasing number of particles over iterations.

- Another stochastic version of the EM algorithm involves a stochastic approximation procedure for which it is called *stochastic approximation EM* (SAEM) [Delyon et al., 1999]. In SAEM, the E-step involves a weighted average of the approximations of the intermediate quantity of EM obtained in the current as well as in the previous iterations. Specifically, consider step size sequence $\{\gamma_j\}_{j\geq 0}$ satisfying the conditions

in (3.24). Then we calculate the weighted average of the estimates $Q^N(\theta^{(j)}, \theta)$ of the intermediate functions recursively as

$$Q_{\gamma,j}(\theta) = (1 - \gamma_j)\, Q_{\gamma,j-1}(\theta) + \gamma_j Q^N(\theta^{(j)}, \theta),$$

with the initialisation $Q_{\gamma,-1}(\theta) = 0$ and at the M-step at iteration j θ_{j+1} is set to be the maximiser of $Q_{\gamma,j}(\theta)$ with respect to θ. When $p_\theta(x_{1:n}, y_{1:n})$ is in the exponential family, the above recursion is in terms of the smoothed estimates of sufficient statistics; we will see a use of SAEM in this case in Chapter 5.

3.4.3.2 Online EM

The online EM algorithm [Cappé, 2009, 2011; Elliott et al., 2002; Kantas et al., 2009; Mongillo and Deneve, 2008] is a variation over the batch EM where, as in online gradient ascent algorithm, the parameter is re-estimated each time a new observation is collected. We assume that $p_\theta(x_{1:n}, y_{1:n})$ is in the exponential family and there exists sufficient statistics so that the M-step can be characterised by (3.34). In the online EM algorithm, running averages of $S^\theta_{i,n}$ are computed. Specifically, let $\gamma = \{\gamma_n\}_{n\geq 1}$, called the step-size sequence, be a positive decreasing sequence satisfying $\sum_{n\geq 1} \gamma_n = \infty$ and $\sum_{n\geq 1} \gamma_n^2 < \infty$. Let θ_1 be the initial guess of θ^* before having made any observations and at time n and let $\theta_{1:n}$ be the sequence of parameter estimates of the online EM algorithm computed sequentially based on $y_{1:n-1}$. When y_n is received, online EM computes for $i = 1, \ldots, r$

$$T_{\gamma,i,n}(x_n) = M_{\theta_{1:n},n-1}\left[(1-\gamma_n) T_{\gamma,i,n-1} + \gamma_n s_{i,n}(\cdot, x_n)\right](x_n), \tag{3.35}$$

$$\mathcal{S}_{i,n} = \eta_{\theta_{1:n},n,n}(T_{\gamma,i,n}) \tag{3.36}$$

and then sets

$$\theta_{n+1} = \Lambda\left(\mathcal{S}_{1,n}, \ldots, \mathcal{S}_{r,n}\right).$$

The subscript $\theta_{1:n}$ on $M_{\theta_{1:n},n-1}$ and $\eta_{\theta_{1:n},n,n}$ indicates that these laws are being computed sequentially using the parameter θ_t at time t, $t \leq n$. In practice, the maximisation step is not executed until a burn-in time n_b for added stability of the estimators as discussed in Cappé [2009].

The online EM algorithm can be implemented exactly for a linear Gaussian state-space model [Elliott et al., 2002] and for finite state-space HMM's. [Cappé, 2011; Mongillo and Deneve, 2008]. An exact implementation is not possible for state-space models in general, therefore SMC implementations of the online EM algorithm are used. Both the $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ approximations are used for the SMC implementation on online EM in the literature, we present both of them in Algorithms 3.5 and 3.6. The first SMC online

EM algorithm, proposed in Cappé [2009] uses the path space approximation to equations (3.35) and (3.36) resulting in Algorithm 3.5. The $\mathcal{O}(N^2)$ approximation was proposed in Del Moral et al. [2009], resulting in Algorithm 3.6.

Algorithm 3.5. ***SMC-online EM:*** $\mathcal{O}(N)$ ***implementation***
Choose θ_1. *For* $n = 1, 2, \ldots$;

- *If* $n = 1$,
  - *Compute the SMC approximation* $\{X_1^{(i)}, W_1^{(i)}\}_{1 \leq i \leq N}$ *for* $\pi_{\theta_1,1}$.
  - *For* $i = 1, \ldots, N$; *for* $k = 1, \ldots, r$ *set* $T_{\gamma,k,1}^{(i)} = s_{k,1}(X_1^{(i)})$.

  if $n \geq 2$,
  - *Compute the SMC approximation* $\{X_{1:n}^{(i)}, W_n^{(i)}\}_{1 \leq i \leq N}$ *for* $\pi_{\theta_{1:n},n}$. *Construct the* $N \times 1$ *vector* A *of resampling indexes such that* $X_{1:n}^{(i)} = (X_{1:n-1}^{(A(i))}, X_n^{(i)})$.
  - *For* $i = 1, \ldots, N$; *set* $j = A(i)$, *and compute for* $k = 1, \ldots, r$ *set*

$$T_{\gamma,k,n}^{(i)} = (1 - \gamma_n) T_{\gamma,k,n-1}^{(j)} + \gamma_n s_{k,n}(X_{n-1}^{(j)}, X_n^{(i)}).$$

- *If* $n \geq n_b$, *calculate* $\mathcal{S}_{k,n} = \sum_{i=1}^N W_n^{(i)} T_{\gamma,k,n}^{(i)}$ *for* $k = 1, \ldots, r$ *and set* $\theta_{n+1} = \Lambda(\mathcal{S}_{1,n}, \ldots, \mathcal{S}_{r,n})$. *Else, set* $\theta_{n+1} = \theta_n$.

Algorithm 3.6. ***SMC-online EM:*** $\mathcal{O}(N^2)$ ***implementation***
Choose θ_1. *For* $n = 1, 2, \ldots$;

- *If* $n = 1$,
  - *Compute the SMC approximation* $\{X_1^{(i)}, W_1^{(i)}\}_{1 \leq i \leq N}$ *for* $\eta_{\theta_1,1,1}$.
  - *For* $i = 1, \ldots, N$; *for* $k = 1, \ldots, r$ *set* $T_{\gamma,k,1}^{(i)} = s_{k,1}(X_1^{(i)})$.

  if $n \geq 2$,
  - *Compute the SMC approximation* $\{X_n^{(i)}, W_n^{(i)}\}_{1 \leq i \leq N}$ *for* $\eta_{\theta_{1:n},n,n}$.
  - *For* $i = 1, \ldots, N$; *for* $k = 1, \ldots, r$ *set*

$$T_{\gamma,k,n}^{(i)} = \frac{\sum_{j=1}^N \left[(1 - \gamma_n) T_{\gamma,k,n-1}^{(j)} + \gamma_n s_{k,n}(X_{n-1}^{(j)}, X_n^{(i)})\right] W_{n-1}^{(j)} f_{\theta_n}(X_n^{(i)} | X_{n-1}^{(j)})}{\sum_{j'=1}^N W_{n-1}^{(j')} f_{\theta_n}(X_n^{(i)} | X_{n-1}^{(j')})}.$$

- *If* $n \geq n_b$, *calculate* $\mathcal{S}_{k,n} = \sum_{i=1}^N W_n^{(i)} T_{\gamma,k,n}^{(i)}$ *for* $k = 1, \ldots, r$ *and set* $\theta_{n+1} = \Lambda(\mathcal{S}_{1,n}, \ldots, \mathcal{S}_{r,n})$. *Else, set* $\theta_{n+1} = \theta_n$.

3.4.4 Iterated filtering

As a batch MLE method for HMMs, iterated filtering Ionides et al. [2011] can be useful to non-linear state space dynamics. The iterated filtering algorithm works as follows. We begin with $\theta^{(0)}$ and assume that at the end of the $j-1$'th iteration we obtain the estimate $\theta^{(j)}$. Iterated filtering extends the HMM further as $\{X_t, \theta_t, Y_t\}_{t\geq 1}$ by introducing a slowly moving Markov chain for the static parameter as $\{\theta_t\}_{t\geq 1}$. At iteration j, the Markov chain for $\{\theta_t\}_{t\geq 1}$ is a random walk typically with Gaussians steps.

$$\theta_1 \sim \mathcal{N}(\theta^{(j)}, \tau_j^2 \Sigma), \quad \theta_k|\theta_{k-1} \sim \mathcal{N}(\theta_{k-1}, \sigma_j^2 \Sigma), \quad k \geq 2. \tag{3.37}$$

At iteration j, one runs an SMC filter for the $\{X_t, \theta_t, Y_t\}_{t\geq 1}$ with N_j particles, and calculates at every time step the mean and variance estimates for θ_t with respect to the filtering and prediction densities, respectively

$$m_t = \mathbb{E}_{\theta^{(j-1)}}\left[\theta_t|y_{1:t}\right], \quad V_k = var_{\theta^{(j-1)}}\left[\theta_t|y_{1:t-1}\right], \quad t = 1, \ldots, n. \tag{3.38}$$

Denoting the SMC estimates of these quantities as $\widetilde{m}_t$ and $\widetilde{V}_t$ and letting $\widetilde{m}_0 = 0$, the algorithm updates the parameter estimates by

$$\theta^{(j)} = \theta^{(j-1)} + \gamma_j \sum_{t=1}^{n} \widetilde{V}_t^{-1}(\widetilde{m}_t - \widetilde{m}_{t-1}) \tag{3.39}$$

Actually, the quantity $\sum_{t=1}^{n} \widetilde{V}_t^{-1}(\widetilde{m}_t - \widetilde{m}_{t-1})$ is an approximation to the gradient of the log likelihood, $\nabla_\theta l_\theta(y_{1:n})$ at $\theta = \theta^{(j-1)}$.

Here, the positive sequences $\{\tau_j\}_{j\geq 0}$ and $\{\sigma_j\}_{j\geq 0}$ satisfy the conditions $\lim_{j\to\infty} \tau_j = 0$ and $\lim_{j\to\infty} \sigma_j/\tau_j = 0$, which are the conditions leading to an annealing schedule. Moreover, the sequences of number of particles and step sizes must satisfy $N_j\tau_j \to \infty$ and $\sum_j \gamma_j^2 N_j^{-1} \tau_j^{-2} < \infty$, which are the conditions for convergence of the stochastic approximation for θ to a local maximum [Ionides et al., 2011].

3.4.5 Discussion of the MLE methods

One one hand, one might prefer a gradient ascent procedure over the EM algorithm for a number of reasons. Firstly, when $l_\theta(y_{1:n})$ is a concave function of θ, if γ_j is replaced by $-\gamma_j \Gamma_j^{-1}$ where Γ_j is the Hessian of $l_\theta(y_{1:n})$ evaluated at θ_j, then the rate of convergence is quadratic and thus faster than the EM which converges linearly. The Hessian matrix can be estimated using SMC techniques, see Poyiadjis et al. [2011]. Secondly, the gradient ascent algorithm is more general since it can be implemented in those cases where M-step of the EM cannot be solved in closed-form. On the other hand, scaling the gradients

might be quite hard. In addition, the EM needs less tuning and its M-step is typically numerically stable. Therefore, one might prefer an EM approach if the M-step can be computed analytically. Finally, both approaches have online versions, which makes them very powerful tools in dealing with large sequential data sets.

An advantage of iterative filtering over standard gradient and EM techniques is that it only requires being able to sample from $f_\theta(x'|x)$ and there is no explicit calculations of the derivative. However, it might require a bit of tuning when the parameter is high-dimensional. Another disadvantage of iterated filtering is the necessity to use increasing number of particles versus iterations in order to ensure convergence. Finally, iterated filtering does not have an online version hence can only be used in a batch setting.

Chapter 4

An Online Expectation-Maximisation Algorithm for Changepoint Models

Summary: *Changepoint models are widely used to model the heterogeneity of sequential data. We present a novel sequential Monte Carlo (SMC) online Expectation-Maximisation (EM) algorithm for estimating the static parameters of such models. The SMC online EM algorithm has a cost per time which is linear in the number of particles and could be particularly important when the data is representable as a long sequence of observations, since it drastically reduces the computational requirements for implementation. We present an asymptotic analysis for the stability of the SMC estimates used in the online EM algorithm and demonstrate the performance of this scheme using both simulated and real data originating from DNA analysis.*

The work done in this chapter is published in Yıldırım et al. [2012d]. The idea was initiated in a discussion Dr. Sumeetpal S. Singh had with Prof. Arnaud Doucet. I did all the work except that Section 4.4 was done in collaboration with Dr. Sumeetpal S. Singh.

4.1 Introduction

Consider a sequence of observations $\{y_1, y_2, \ldots\}$ collected sequentially in time. A changepoint model is a particular model for heterogeneity of sequential data that postulates the existence of a strictly increasing time sequence $t_1, t_2, \ldots$ with $t_1 = 1$, that partitions the data into disjoint segments

$$\{y_{t_1}, \ldots, y_{t_2-1}\}, \{y_{t_2}, \ldots, y_{t_3-1}\}, \ldots$$

and that the data is correlated within a segment but are otherwise independent across segments. The time instances $t_1, t_2, \ldots$ are known as the *changepoints* and constitute a random unobserved sequence. This segmental structure is both an intuitive and versatile model for heterogeneity and it is the reason why changepoint models have enjoyed a wide

appeal in a variety of disciplines such as Biological Science [Braun and Muller, 1998; Caron et al., 2011; Fearnhead and Vasileiou, 2009; Johnson et al., 2003], Physical Science [Lund and Reeves, 2002; Ó Ruanaidh and Fitzgerald, 1996] Signal Processing [Cemgil et al., 2006; Punskaya et al., 2002], and Finance [Dias and Embrechts, 2004].

In a Bayesian approach to inferring changepoints, one adopts a prior distribution on their locations and a likelihood function for the observed process given these changepoints. However, both of these laws typically depend on a finite dimensional real parameter vector $\theta \in \Theta$ where Θ denotes the set of permissible parameter vectors. In all realistic applications, the static parameter θ is unknown and needs to be estimated from the data as well. A fully Bayesian approach would assign a prior distribution to θ. However the resulting posterior distribution is intractable. Several Markov chain Monte Carlo (MCMC) schemes have been proposed in this context [Chib, 1998; Fearnhead, 2006; Lavielle and Lebarbier, 2001; Stephens, 1994]. Unfortunately these algorithms are far too computationally intensive when dealing with very large datasets. Alternative to an MCMC based full Bayesian analysis is sequential Monte Carlo (SMC); however, SMC methods to perform online Bayesian static parameter estimation suffer from the well-known particle path degeneracy problem and can provide unreliable estimates; see Andrieu et al. [2005], Olsson et al. [2008] for a discussion of this issue. This is why we focus here on estimating the parameter θ using a maximum likelihood approach; i.e. the Maximum Likelihood Estimate (MLE) of interest is the parameter vector from Θ that maximises the probability density of the observed data sequence $p_\theta(y_1, \ldots, y_n)$. This is a challenging problem as computing the likelihood $p_\theta(y_1, \ldots, y_n)$ requires a computational cost increasing super-linearly with n [Chopin, 2007; Fearnhead and Liu, 2007].

Our main contribution is a novel online EM algorithm to compute the MLE of the static parameter θ for changepoint models. We remark that standard batch EM algorithms for a restricted class of changepoint models have been proposed before, e.g. see Gales and Young [1993], Barbu and Limnios [2008], Fearnhead and Vasileiou [2009]. The main reason why an online algorithm is desirable is that huge computational and memory savings are possible. For a long data sequence, a standard EM algorithm requires a complete browse through the entire data set at each iteration to update the MLE of θ; and many such iterations are needed until the estimate of θ converges. This not only requires storing the entire data sequence but also the probability laws that are needed in the intermediate computations done in each EM iteration, which can be impractical. For this reason, there has been a strong interest in online methods which make parameter estimation possible by browsing through the data only once and hence circumventing the need to store it in its entirety (see Kantas et al. [2009] for a review). The only other work on computing the MLE of θ for a more restrictive class of changepoint models in an online manner that we are aware of is Caron et al. [2011], where the authors used

a recursive gradient algorithm. If the model permits an EM implementation then it is fair to say that the EM is generally preferred by practitioners as no algorithm tuning is required whereas it can be difficult to properly scale the components of the computed gradient vector.

For finite state-space Hidden Markov Models (HMM) [Cappé, 2011; Mongillo and Deneve, 2008] and linear Gaussian state-space models [Elliott et al., 2002], it is possible to implement exactly the online EM algorithm. A detailed study of this algorithm in the finite state-space case can be found in Cappé [2011]. For changepoint models, it is necessary to approximate numerically certain expectations sequentially over time with respect to (w.r.t.) the conditional law of the changepoints and other latent random variables of the model given the available observations up to that point in time. We present SMC estimates of these expectations and establish the stability (via the variance) of these estimates w.r.t. time n and the number of particles N both theoretically and with numerical examples. Stability of the SMC estimates of the expectations is important for assessing the performance and reliability of the EM algorithm and is not to be taken for granted because these expectations are computed w.r.t. a probability law whose dimension increases linearly with time n. We note that the computational cost of the proposed SMC online EM algorithm is $\mathcal{O}(N)$ per-time whereas a $\mathcal{O}(N^2)$ per-time algorithm is required to obtain similar stability results for general state-space HMMs [Del Moral et al., 2009]. Cappé [2011], remarked that "although the online EM algorithm resembles a classical stochastic approximation algorithm, it is sufficiently different to resist conventional 'analysis of convergence'. We believe that limited results similar to those discussed in Cappé [2011, Section 4] identifying the potential accumulation points of the online EM procedure could be established but this is beyond the scope of this work. In the numerical studies reported in this work, and indeed in all the ones we have conducted, the SMC online EM algorithm converges, and to a very close vicinity of the correct values when these are known, e.g. in synthetic examples. Moreover, we observed that online EM converged significantly quicker than the batch EM implementation.

The organisation of the chapter is as follows. In Section 4.2, we describe a general changepoint model. In Section 4.3, we present the associated online EM algorithm and its SMC implementation. Theoretical results on the stability of the SMC estimates used in the online EM algorithm are given in Section 4.4. In Section 4.5, we demonstrate the performance of the SMC online EM algorithm on both simulated and real data. We finish with a discussion in Section 4.6 and finally, some detailed model specific derivations as well as mathematical proofs are given in Appendix.

4.2 The changepoint model

In this work a changepoint model is defined to be comprised of two discrete-time stochastic processes which are $\{(X_k, Z_k)\}_{k\geq 1}$ and $\{Y_k\}_{k\geq 1}$. $\{(X_k, Z_k)\}_{k\geq 1}$ is an unobserved time-homogeneous Markov chain taking values in $\mathcal{X} \times \mathcal{Z}$ where $\mathcal{X} = \{1, 2, \ldots\} \times \{1, \ldots, R\}$ and $\mathcal{Z} \subseteq \mathbb{R}^p$. (While the definition of $\mathcal{X}$ in this manner is necessary for the resulting model to be a changepoint model, the definition of $\mathcal{Z}$ can change depending on the application domain.) We denote realisations of the first component of this chain by $x_k = (d_k, m_k)$. The variable m_k takes values in the index set $\{1, \ldots, R\}$ and indicates the (generative) model the chain is in at that time while d_k indicates the duration the chain has spent in model m_k. The transition law of $\{(X_k, Z_k)\}_{k\geq 1}$ is

$$X_1 \sim \mu, \qquad X_k \,|(x_{k-1} = (d, m), z_{k-1}) = \begin{cases} (d+1, m) & \text{w.p. } 1 - \lambda_{\theta,m}(d) \\ (1, m') & \text{w.p. } \lambda_{\theta,m}(d) \times P_\theta(m, m') \end{cases},$$
$$Z_k \,|(x_k = (d', m'), x_{k-1}, z_{k-1}) \sim \begin{cases} f_{\theta,m'}(z|z_{k-1})dz & \text{if } d' \neq 1 \\ \pi_{\theta,m'}(z)dz & \text{if } d' = 1 \end{cases}, \tag{4.1}$$

where $\lambda_{\theta,m}(d) \in [0, 1]$ for all $\theta \in \Theta$ and $(d, m) \in \mathcal{X}$; P_θ is an $R \times R$ row stochastic matrix; for each θ and m, $f_{\theta,m}(z|z_{k-1})$ is the density of a Markov transition kernel on $\mathcal{Z}$ w.r.t. a suitable dominating measure which is denoted by dz; and for each θ and m, $\pi_{\theta,m}$ is a probability density on $\mathcal{Z}$. The transition kernel of the Markov chain $\{(X_k, Z_k)\}_{k\geq 1}$ is assumed to be parametrised by the finite dimensional parameter $\theta \in \Theta$. Without loss of generality, it is assumed that the probability distribution of the initial state of the chain $\{X_k\}_{k\geq 1}$, denoted μ, has all its mass on $\{(1, 1), \ldots, (1, R)\}$, e.g. the uniform distribution on $\{(1, 1), \ldots, (1, R)\}$.

For a sequence $\{a_k\}_{k\geq 1}$ and integers i, j, let $a_{i:j}$ denote the set $\{a_i, a_{i+1}, ..., a_j\}$, which is empty if $j < i$, and $a_{i:\infty} = \{a_i, a_{i+1}, ...\}$. The process $\{Y_k\}_{k\geq 1}$ is a $\mathcal{Y}$-valued observed process which satisfies the following conditional independence property:

$$Y_k \left| \left(\{x_k, z_k\}_{k\geq 1}, y_{1:k-1}, y_{k+1:\infty} \right) \sim g_{\theta,m_k}(y|z_k)dy \right. \tag{4.2}$$

where for each θ and m, $g_{\theta,m}$ is a probability density on $\mathcal{Y}$ with respect to the dominating measure dy. In this work $\mathcal{Y} \subseteq \mathbb{R}^q$ although the definition of $\mathcal{Y}$ may be altered depending on the application. Equations (4.1) and (4.2), now define the law of $(X_{1:n}, Z_{1:n}, Y_{1:n})$.

Note that $\{X_k\}_{k\geq 1}$ itself is a Markov chain and we denote its transition matrix by $p_\theta(x_k | x_{k-1})$. Secondly, it is useful to visualise a realisation of $\{X_k\}_{k\geq 1}$ as a labelled contiguous partition of $\{1, 2, \ldots\}$, $\{[t_1, t_2), [t_2, t_3), \ldots\}$ and $t_{i+1} > t_i$, where each set $[t_i, t_{i+1})$ of the partition, which we call a *segment*, is accompanied by m_{t_i}, the model number during that segment. The variables t_i are the instances $\{X_k\}_{k\geq 1}$ visits the set $\{1\} \times \{1, \ldots, R\}$

and are called as the changepoints. As $\{Z_k\}_{k\geq 1}$ forgets its past at times of changepoints, within the segment $[t_i, t_{i+1})$, $\{(Z_k, Y_k)\}_{t_i \leq k < t_{i+1}}$ is a HMM with initial, state transition, and observation densities $\pi_{\theta,m_{t_i}}$, $f_{\theta,m_{t_i}}$, and $g_{\theta,m_{t_i}}$ respectively. In this sense, our model is general enough to encompass both *hidden semi-Markov models* ([Barbu and Limnios, 2008; Murphy, 2002] and *segmented hidden semi-Markov models* [Dong and He, 2007; Gales and Young, 1993]. Below, we give an example of a changepoint model, which we will use in our experiments throughout this chapter.

Example 4.1. *Consider the following changepoint model presented in Fearnhead and Vasileiou [2009], where* $Z_k = (Z_{k,1}, Z_{k,2}) \in \mathbb{R} \times \mathbb{R}^+$, *and* $\mathcal{Y} = \mathbb{R}$. *The model satisfies*

$$X_1 \sim \mathcal{U}_{\{1\}\times\{1,\ldots,R\}}, \qquad X_k \,|(x_{k-1} = (d,m)) = \begin{cases} (d+1, m) & w.p.\ (1-\lambda_m) \\ (1, m') & w.p.\ \lambda_m \times P(m,m') \end{cases},$$

$$Z_k \,|(x_k = (d', m'), z_{k-1}) \sim \begin{cases} \delta_{z_{k-1}} & \text{if } d' \neq 1 \\ \mathcal{N}\Gamma^{-1}(\xi_m, \kappa_m, \alpha, \beta) & \text{if } d' = 1 \end{cases},$$

$$Y_k \,|z_k \sim \mathcal{N}(z_{k,1}, z_{k,2}),$$

where $\mathcal{N}\Gamma^{-1}(\cdot)$ *denotes the normal-inverse gamma distribution and* $\mathcal{U}_A$ *is the uniform distribution over the set* A. *In relation to* (4.1) *and* (4.2), *we have* $\lambda_{\theta,m}(d) = \lambda_m$, $f_{\theta,m}(z|z_{k-1})dz = \delta_{z_{k-1}}(dz)$, $\pi_{\theta,m} = \mathcal{N}\Gamma^{-1}(\xi_m, \kappa_m, \alpha, \beta)$, *and* $g_\theta(y|z_k) = \mathcal{N}(y; z_{k,1}, z_{k,2})$. *Therefore, the parameters of interest are* $\theta = (\xi_{1:R}, \kappa_{1:R}, \lambda_{1:R}, \alpha, \beta, P)$. *In this model, the observations in each segment are i.i.d. Gaussian random variables whose mean and variance change from segment to segment and are drawn from the normal-inverse gamma distribution.*

The following important conditional independence property, which follows from (4.1) and (4.2), will be frequently used in the derivations to follow: for any $k' \geq k$,

$$p_\theta(y_k|x_{1:k'}, y_{1:k-1}) = p_\theta(y_k|x_k, y_{1:k-1}) = p_\theta(y_k|x_k, y_{k-d_k+1:k-1}).$$

(Recall that d_k is the first component of x_k.) This equation may be interpreted to mean that y_k only depends statistically on the past observations that are received since the most recent changepoint and not on the observations before that. For the models considered in this work we assume that $p_\theta(y_k|x_k, y_{1:k-1})$ can be evaluated for any x_k and $y_{1:k}$ (whenever the conditional law is well defined). This assumption is satisfied by some important models (e.g. Caron et al. [2011]; Fearnhead and Vasileiou [2009]; Whiteley et al. [2009]), and allows us to focus inference on $X_{1:n}$ and θ given $Y_{1:n}$ as $Z_{1:n}$ may be integrated out.

For a given realisation of observations $\{y_k\}_{k\geq 1}$, we define the potential function $G_{\theta,k}$:

$\mathcal{X} \to [0, \infty)$ as

$$G_{\theta,k}(x_k) = \frac{\int \pi_{\theta,m_k}(z_j) \prod_{i=j+1}^{k} f_{\theta,m_k}(z_i|z_{i-1}) \prod_{i=j}^{k} g_{\theta,m_k}(y_i|z_i) dz_{j:k}}{\int \pi_{\theta,m_k}(z_j) \prod_{i=j+1}^{k-1} f_{\theta,m_k}(z_i|z_{i-1}) \prod_{i=j}^{k-1} g_{\theta,m_k}(y_i|z_i) dz_{j:k-1}}, \quad j = \max(k-d_k+1, 1).$$

($G_{\theta,k}$ is introduced for brevity.) Note that $G_{\theta,k}(x_k)$ is precisely $p_\theta(y_k|x_k, y_{1:k-1})$ at values of x_k where the latter is well defined. We can now express the probability density of the observed process, or likelihood, succinctly as

$$p_\theta\left(y_{1:n}\right) = \mathbb{E}_\theta\left[\prod_{k=1}^{n} G_{\theta,k}(X_k)\right].$$

4.3 EM algorithms for changepoint models

Our main aim is to estimate the static parameter θ of the changepoint model in an online manner using the EM algorithm. We first introduce the batch EM algorithm and then explain how it can be modified to obtain the online EM version.

4.3.1 Batch EM

Given $Y_{1:n} = y_{1:n}$, the EM algorithm for maximising $p_\theta(y_{1:n})$ is given by the following iterative procedure: if θ_i is the estimate of the maximiser at the ith iteration, then at iteration $i+1$ we first calculate the following intermediate optimisation criterion,

$$\begin{aligned} Q(\theta_i, \theta) &= \mathbb{E}_{\theta_i}\left[\log p_\theta(y_{1:n}, Z_{1:n}, X_{1:n}) \middle| y_{1:n}\right] \\ &= \mathbb{E}_{\theta_i}\left[\log p_\theta(X_{1:n}) + \log p_\theta(y_{1:n}, Z_{1:n}|X_{1:n}) \middle| y_{1:n}\right] \\ &= \mathbb{E}_{\theta_i}\left[\log p_\theta(X_{1:n}) + \mathbb{E}_{\theta_i}\left\{\log p_\theta(y_{1:n}, Z_{1:n}|X_{1:n}) \middle| y_{1:n}, X_{1:n}\right\} \middle| y_{1:n}\right]. \end{aligned} \tag{4.3}$$

This step is known as the expectation (E) step. The inner expectation in (4.3) is w.r.t. the law of $Z_{1:n}$ conditioned on $y_{1:n}$ and $X_{1:n}$ under θ_i, that is $p_{\theta_i}\left(z_{1:n} \middle| y_{1:n}, x_{1:n}\right)$, whereas the outer expectation is w.r.t. the law of $X_{1:n}$ conditioned on $y_{1:n}$ under θ_i, that is $p_{\theta_i}\left(x_{1:n} \middle| y_{1:n}\right)$. The updated estimate is then computed in the maximisation (or M) step

$$\theta_{i+1} = \arg\max_\theta Q(\theta_i, \theta).$$

This procedure is repeated until θ_i converges (or ceases to change significantly).

Let us define the integrand of the outer expectation in (4.3) as the function $H_k : \mathcal{X}^k \times \mathcal{Y}^k \times \Theta^2 \to \mathbb{R}$, $k = 1, \ldots, n$,

$$H_k(x_{1:k}, y_{1:k}, \theta_i, \theta) := \log p_\theta(x_{1:k}) + \mathbb{E}_{\theta_i}\left[\log p_\theta(y_{1:k}, Z_{1:k}|x_{1:k}) \middle| y_{1:k}, x_{1:k}\right]$$

We can exploit the following three properties of H_k and $Q(\theta_i, \theta)$. Firstly, H_k has an additive structure (see Appendix 4.A.1 for a derivation):

$$H_k(x_{1:k}, y_{1:k}, \theta_i, \theta) = H_{k-1}(x_{1:k-1}, y_{1:k-1}, \theta_i, \theta) + h_k(x_{k-1}, x_k, y_{k-d_k+1:k}, \theta_i, \theta) \tag{4.4}$$

where the incremental term h_k is a function of $(x_{k-1}, x_k, y_{k-d_k+1}, \ldots, y_k, \theta_i, \theta)$. Secondly, when the transition laws of the changepoint model given in (4.1)-(4.2) belong to the exponential family then the incremental terms can be expressed as

$$h_k(x_{k-1}, x_k, y_{k-d_k+1:k}, \theta_i, \theta) = v_\theta^{\mathrm{T}} s_k(x_{k-1}, x_k, y_{k-d_k+1:k}, \theta_i) \tag{4.5}$$

where v_θ is a $r \times 1$ vector depending only on θ, s_k is a $r \times 1$ vector valued function of $(x_{k-1}, x_k, y_{k-d_k+1}, \ldots, y_k, \theta_i)$. (From now on, we omit the dependency of H_k, h_k, and s_k on $y_{1:k}$ for the sake of conciseness.) If (4.5) holds, $Q(\theta_i, \theta) = v_\theta^{\mathrm{T}} \mathbb{E}_{\theta_i}\left[S_n(X_{1:n}, \theta_i) \middle| y_{1:n}\right]$ where

$$S_n(x_{1:n}, \theta_i) = \sum_{j=1}^{n} s_j(x_{j-1}, x_j, \theta_i), \tag{4.6}$$

with $s_1(x_0, x_1, \theta) = s_1(x_1, \theta)$ by convention, and its maximiser is explicitly characterised by a function $\Lambda : \mathbb{R}^r \to \Theta$

$$\arg\max_{\theta \in \Theta} Q(\theta_i, \theta) = \Lambda\left(\mathbb{E}_{\theta_i}\left[S_n(X_{1:n}, \theta_i) \middle| y_{1:n}\right]\right). \tag{4.7}$$

Hence from a practical point of view, it is necessary to compute the expectation of additive functionals (4.6) w.r.t. $p_{\theta_i}\left(x_{1:n} \middle| y_{1:n}\right)$. As for a standard HMM, this can be achieved using a forward-backward type algorithm; see Gales and Young [1993], Barbu and Limnios [2008], Fearnhead and Vasileiou [2009]. However in a general scenario the computational complexity is quadratic in n and approximations are necessary when n is very large. In Fearnhead and Vasileiou [2009] a Monte Carlo EM (MCEM) algorithm was proposed for a specific changepoint model (see Section 4.5) where the expectations in the E-step are computed using a backward Monte Carlo sampling procedure.

4.3.2 Online EM

The development of an online version of the EM rests on the following key fact [Cappé, 2011; Del Moral et al., 2009]. The quantity $\mathbb{E}_\theta\left[S_n(X_{1:n}, \theta) \middle| y_{1:n}\right]$ when S_n has the additive structure in (4.6) can be evaluated sequentially with the following recursion which we

will refer to as the *forward smoothing recursion*:

$$\begin{aligned} T_n(x_n, \theta) &:= \sum_{x_{1:n-1} \in \mathcal{X}^{n-1}} S_n(x_{1:n}, \theta) p_\theta(x_{1:n-1} | y_{1:n-1}, x_n) \\ &= \sum_{x_{n-1} \in \mathcal{X}} \left[T_{n-1}(x_{n-1}, \theta) + s_n(x_{n-1}, x_n, \theta) \right] p_\theta(x_{n-1} | y_{1:n-1}, x_n) \end{aligned}$$

with $T_1(x_1, \theta) = s_1(x_1, \theta)$. The second line follows from (4.6) and the decomposition

$$p_\theta(x_{1:n-1} | y_{1:n-1}, x_n) = p_\theta(x_{1:n-2} | y_{1:n-2}, x_{n-1}) p_\theta(x_{n-1} | y_{1:n-1}, x_n) \tag{4.8}$$

due to the fact that given x_{n-1}, $x_{1:n-2}$ do not depend on $x_n, x_{n+1}, \ldots, y_{n-1}, y_n, \ldots$, which follows from (4.1) and (4.2). The function $T_n(\cdot, \theta) : \mathcal{X} \to \mathbb{R}^r$ can be computed in an online manner and hence so can

$$\mathbb{E}_\theta \left[S_n(X_{1:n}, \theta) \middle| y_{1:n} \right] = \sum_{x_n \in \mathcal{X}} T_n(x_n, \theta) p_\theta(x_n | y_{1:n}).$$

It is possible to use this recursion to implement the batch EM algorithm. Compared to the standard forward-backward implementation, this approach does not require a backward pass to compute the expectations of interest and hence requires far less memory to implement.

The online EM algorithm is a variation over the batch EM where the parameter is re-estimated each time a new observation is collected. In this approach running averages of $\mathbb{E}_\theta \left[S_n(X_{1:n}, \theta) \middle| y_{1:n} \right]$ are computed [Cappé, 2009, 2011; Elliott et al., 2002; Mongillo and Deneve, 2008], [Kantas et al., 2009, Section 3.2.]. Let $\gamma = \{\gamma_n\}_{n \geq 1}$, called the step-size sequence, be a positive decreasing sequence satisfying $\sum_{n \geq 1} \gamma_n = \infty$ and $\sum_{n \geq 1} \gamma_n^2 < \infty$. A common choice is $\gamma_n = n^{-a}$ for $0.5 < a \leq 1$. Let θ_1 be the initial guess of θ^* before having made any observations and let $\theta_{1:n}$ be the sequence of parameter estimates of the online EM algorithm computed sequentially based on $y_{1:n-1}$. When y_n is received, online EM computes

$$T_{\gamma,n}(x_n) = \sum_{x_{n-1} \in \mathcal{X}} \left[(1 - \gamma_n) T_{\gamma,n-1}(x_{n-1}) + \gamma_n s_n(x_{n-1}, x_n, \theta_n) \right] p_{\theta_{1:n}}(x_{n-1} | y_{1:n-1}, x_n), \tag{4.9}$$

$$\mathcal{S}_n = \sum_{x_n \in \mathcal{X}} T_{\gamma,n}(x_n) p_{\theta_{1:n}}(x_n | y_{1:n}) \tag{4.10}$$

and then sets $\theta_{n+1} = \Lambda(\mathcal{S}_n)$. The subscript $\theta_{1:n}$ on $p_{\theta_{1:n}}(x_{n-1} | y_{1:n-1}, x_n)$ and $p_{\theta_{1:n}}(x_n | y_{1:n})$ indicates that these laws are being computed sequentially using the parameter θ_k at time k, $k \leq n$. (See Algorithm 4.1 for details.) In practice, the maximisation step is not executed until a burn-in time n_b for added stability of the estimators as discussed in Cappé [2009].

The online EM algorithm can be implemented exactly for a linear Gaussian state-space model [Elliott et al., 2002] and for finite state-space HMM's. [Cappé, 2011; Mongillo and Deneve, 2008]. An exact implementation is not possible for changepoint models in general, therefore we now investigate SMC implementations of the online EM algorithm.

4.3.3 SMC implementations of the online EM algorithm

Let $\mathbb{Q}_{\theta,n}(x_{1:n}) = p_\theta(x_{1:n}|y_{1:n-1})$ denote the law of $X_{1:n}$ conditioned on the sequence of observed variables $y_{1:n-1}$, and let $\eta_{\theta,n}(x_n) = p_\theta(x_n|y_{1:n-1})$ denote the time n marginal of $\mathbb{Q}_{\theta,n}$. $\eta_{\theta,n}$ is also known as the predicted filter but we refer to it simply as the filter. In order to execute (4.9) and (4.10) at time n, we need to calculate the following probability distributions:

$$p_\theta(x_{n-1}|x_n, y_{1:n-1}) = \frac{\eta_{\theta,n-1}(x_{n-1})G_{\theta,n-1}(x_{n-1})p_\theta(x_n|x_{n-1})}{\sum_{x'_{n-1}} \eta_{\theta,n-1}(x'_{n-1})G_{\theta,n-1}(x'_{n-1})p_\theta(x_n|x'_{n-1})} \tag{4.11}$$

$$p_\theta(x_n|y_{1:n}) = \frac{\eta_{\theta,n}(x_n)G_{\theta,n}(x_n)}{\sum_{x'_n} \eta_{\theta,n}(x'_n)G_{\theta,n}(x'_n)} \tag{4.12}$$

Note that to calculate these probability distributions we only need $\eta_{\theta,n-1}$ and $\eta_{\theta,n}$ at time n. Besides, $\eta_{\theta,n}$ may be computed recursively using Bayes' formula:

$$\eta_{\theta,n}(x_n) = \frac{\sum_{x_{n-1}} \eta_{\theta,n-1}(x_{n-1})\, G_{\theta,n-1}(x_{n-1})\, p_\theta(x_n|\, x_{n-1})}{\sum_{x_{n-1}} \eta_{\theta,n-1}(x_{n-1})\, G_{\theta,n-1}(x_{n-1})}, \quad n > 1, \tag{4.13}$$

However, the computational cost of the filtering recursion in (4.13) at time n is $\mathcal{O}(nR)$; this follows since $p_\theta(x'|x)$ is non-zero for at most $R+1$ values of x'. For the analysis of large amounts of data, exact filtering is computationally infeasible and SMC methods have been introduced as a viable alternative [Chopin, 2007; Fearnhead and Liu, 2007].

One way to obtain the SMC approximation to $\eta_{\theta,n}$ is via the *path space* particle approximation of $\mathbb{Q}_{\theta,n}$. This is the empirical measure corresponding to a set of $N \geq 1$ random samples termed particles [Del Moral, 2004]:

$$\mathbb{Q}^{\mathrm{p},N}_{\theta,n}(x_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X^{(i)}_{1:n}}(x_{1:n}). \tag{4.14}$$

where $\delta_a(\cdot)$ is the probability mass function concentrated at a. These particles are then propagated in time using importance sampling and resampling steps; see Doucet et al. [2001] and Cappé et al. [2005] for a review of the literature. Specifically, $\mathbb{Q}^{\mathrm{p},N}_{\theta,n}$ is the

empirical measure constructed from N independent samples from

$$\frac{\mathbb{Q}_{\theta,n-1}^{\mathrm{p},N}\left(x_{1:n-1}\right) G_{\theta,n-1}\left(x_{n-1}\right) p_{\theta}\left(x_{n} \middle| x_{n-1}\right)}{\sum_{x_{1:n-1}} \mathbb{Q}_{\theta,n-1}^{\mathrm{p},N}\left(x_{1:n-1}\right) G_{\theta,n-1}\left(x_{n-1}\right)}. \tag{4.15}$$

The particle approximation of $\eta_{\theta,n}$ can now be obtained from $\mathbb{Q}_{\theta,n}^{\mathrm{p},N}$ by marginalization

$$\eta_{\theta,n}^{N}(x_n) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_n^{(i)}}\left(x_n\right). \tag{4.16}$$

Other than the one in (4.16), there are other ways to sequentially update $\eta_{\theta,n-1}^{N}$ so that $\eta_{\theta,n}$ is approximated at N distinct particles. Given $\eta_{\theta,n-1}^{N}$, at time n the distribution

$$\frac{\sum_{x_{n-1}} \eta_{\theta,n-1}^{N}\left(x_{n-1}\right) G_{\theta,n-1}\left(x_{n-1}\right) p_{\theta}\left(x_{n} \middle| x_{n-1}\right)}{\sum_{x_{n-1}} \eta_{\theta,n-1}^{N}\left(x_{n-1}\right) G_{\theta,n-1}\left(x_{n-1}\right)}$$

with support at $N+R$ points is calculated exactly and then $\eta_{\theta,n}^{N}$ is obtained by sampling this distribution independently N times (see Algorithm 4.1). Caron et al. [2011] propose truncating to the N support points with the highest weights. This deterministic resampling scheme introduces bias, but the authors report that this bias is negligible. Fearnhead and Liu [2007] propose an unbiased resampling scheme that retains the maximum number of unique particles in the reduced representation of size N. In the same work, and in Fearnhead and Vasileiou [2009], resampling schemes that allow changing number of particles in time are proposed.

The online EM algorithm in Section 4.3.2 can be approximated with $\mathcal{O}(N)$ cost per time using the SMC approximation of the densities in (4.11) and (4.12). The resulting algorithm, presented as Algorithm 4.1, will be referred to as the *SMC-FS online EM* algorithm.

Algorithm 4.1. ***SMC-FS online EM algorithm for changepoint models***

- ***E-step:*** *If $n = 1$, initialise θ_1; sample $\widetilde{X}_1^{(i)} \sim \mu$, set $\widetilde{T}_1^{(i)} = s_1(\widetilde{X}_1^{(i)}, \theta_1)$, $i = 1, \ldots, N$.*

  If $n \geq 2$

  - *For $i = 1, \ldots, N$, set $\widetilde{X}_n^{(i)} = (d_{n-1}^{(i)} + 1, m_{n-1}^{(i)})$, where $X_{n-1}^{(i)} = (d_{n-1}^{(i)}, m_{n-1}^{(i)})$*
  - *For $m = 1, \ldots, R$, set $\widetilde{X}_n^{(N+m)} = (1, m)$.*
  - *For $i = 1, \ldots, N+R$, compute $\widetilde{W}_n^{(i)} = \sum_{j=1}^{N} G_{\theta_{n-1},n}(X_{n-1}^{(j)}) p_{\theta_n}(\widetilde{X}_n^{(i)} | X_{n-1}^{(j)})$ and*

$$\widetilde{T}_n^{(i)} = \frac{1}{\widetilde{W}_n^{(i)}} \sum_{j=1}^{N} G_{\theta_{n-1},k}(X_{n-1}^{(j)}) p_{\theta_n}(\widetilde{X}_n^{(i)} | X_{n-1}^{(j)}) \left[(1-\gamma_n) T_{n-1}^{(j)} + \gamma_n s_n(X_{n-1}^{(j)}, \widetilde{X}_n^{(i)}, \theta_n)\right]$$

Resample $\{\widetilde{X}_n^{(i)}, \widetilde{T}_n^{(i)}\}_{i=1,\ldots,N+R}$ *according to the weights* $\{\widetilde{W}_n^{(i)}\}_{i=1,\ldots,N+R}$ *to get resampled particles* $\{X_n^{(i)}, T_n^{(i)}\}_{i=1,\ldots,N}$ *each with weight* $1/N$.

- ***M-step:*** *If* $n < n_b$, *set* $\theta_{n+1} = \theta_n$ *else, calculate using the particles before resampling*

$$\mathcal{S}_n = \frac{\sum_{i=1}^{N+R} \widetilde{T}_n^{(i)} \widetilde{W}_n^{(i)} G_{\theta_n,n}(\widetilde{X}_n^{(i)})}{\sum_{i=1}^{N+R} \widetilde{W}_n^{(i)} G_{\theta_n,n}(\widetilde{X}_n^{(i)})},$$

update the parameter $\theta_{n+1} = \Lambda(\mathcal{S}_n)$.

4.3.4 Comparison with the path space online EM

As shown in Section 4.3.1, the EM algorithm requires certain expectations w.r.t. the measure $\mathbb{Q}_{\theta,n}$, and the online EM algorithm in Section 4.3.2 relies on the running averages of these expectations. Consider the following backward representation of $\mathbb{Q}_{\theta,n}$

$$\mathbb{Q}_{\theta,n}(x_{1:n}) = \eta_{\theta,n}(x_n) \prod_{k=n}^{2} p_\theta(x_{k-1}|x_k, y_{1:k-1}).$$

Then a corresponding particle approximation, different from the path-space one, is given by

$$\mathbb{Q}_{\theta,n}^N(x_{1:n}) = \eta_{\theta,n}^N(x_n) \prod_{k=n}^{2} p_\theta^N(x_{k-1}|x_k, y_{1:k-1}). \tag{4.17}$$

where $p_\theta^N(x_{k-1}|x_k, y_{1:k-1})$ is (4.11) with $\eta_{\theta,k-1}$ replaced with $\eta_{\theta,k-1}^N$. One can then show that the online EM algorithm using the SMC approximation to the forward smoothing recursion relies on the particle approximation $\mathbb{Q}_{\theta,n}^N$ described above. More precisely, in Algorithm 4.1, if $\gamma_i = 1/i$, $n < n_b$ (see the M-step), $\theta_1 = \cdots = \theta_{n+1} = \theta$, and $s_{n+1}(x_n, x_{n+1}, \theta) = 0$, then

$$\mathcal{S}_{n+1} = \mathbb{Q}_{\theta,n+1}^N((n+1)^{-1} S_n).$$

This observation will be useful for analysing the stability properties of the sufficient statistics calculated SMC-FS online EM algorithm in Section 4.4.

As an alternative to SMC-FS online EM, we could have proposed an SMC online EM algorithm relying on the particle approximation $\mathbb{Q}_{\theta,n}^{\mathrm{p},N}$ defined in (4.14)-(4.15). In that case (using the short-hand notation in Algorithm 4.1) the approximation to (4.9) and (4.10) becomes

$$\widetilde{T}_n^{(i)} = (1 - \gamma_n) T_{n-1}^{(i)} + \gamma_n s_n(X_{n-1}^{(i)}, \widetilde{X}_n^{(i)}, \theta_n)$$

for each $i = 1, \ldots, N$, and then calculating the estimates of sufficient statistics as

$$\mathcal{S}_n = \frac{\sum_{i=1}^N \widetilde{T}_n^{(i)} G_{\theta_n,n}(\widetilde{X}_n^{(i)})}{\sum_{i=1}^N G_{\theta_n,n}(\widetilde{X}_n^{(i)})}.$$

Recall that each $\widetilde{X}_n^{(i)}$ is sampled from $p_{\theta_n}(x_n|X_{n-1}^{(i)})$. $\{\widetilde{X}_n^{(i)}, \widetilde{T}_n^{(i)}\}_{i=1,\ldots,N}$ are then resampled to obtain $\{X_n^{(i)}, T_n^{(i)}\}_{i=1,\ldots,N}$ according to the weights $\{G_{\theta_n,n}(\widetilde{X}_n^{(i)})\}_{i=1,\ldots,N}$. Based on the path space approximation, we will hereafter call this algorithm the *SMC-PS online EM* algorithm. In the context of general state-space HMM, this was proposed in Cappé [2009] and only requires $\mathcal{O}(N)$ computations per time step. However, it is a well-known fact that $\mathbb{Q}_{\theta,n}^{\mathrm{p},N}$ becomes progressively impoverished as n increases because of the successive resampling steps [Del Moral and Doucet, 2003; Olsson et al., 2008]. That is, the number of distinct particles representing the marginal $\mathbb{Q}_{\theta,n}^{\mathrm{p},N}(x_{1:k})$ for any fixed $k < n$ diminishes as n increases until it eventually collapses to a single particle – this is known as the *particle path degeneracy* problem. Whereas, in the backward particle approximation $\mathbb{Q}_{\theta,n}^N$, we do not have this problem since it relies on the SMC approximations to the filters $\eta_{\theta,n}$ only. Therefore, we expect that the resulting SMC estimates in the SMC-PS online EM algorithm have higher variances than those in the SMC-FS online EM algorithm [Del Moral et al., 2009]. For a numerical illustration of this fact, see Section 4.5.

4.4 Theoretical results

Recall that the M-step of the exact online EM algorithm applies a mapping Λ which maps expectations of sufficient statistics $\mathbb{Q}_{\theta,n+1}(n^{-1}S_n) = \mathbb{E}_\theta\left[n^{-1}S_n(X_{1:n})|y_{1:n}\right]$ to a parameter estimate in Θ; see (4.9) and (4.10) with $\gamma_n = n^{-1}$. It follows from the discussion in Section 4.3.4 that the reliability of the SMC online EM algorithm described in Section 4.3.2 depends on how stable the estimates of expectations of the type $\mathbb{Q}_{\theta,n}^N(S_n)$ are. One convenient way of assessing the stability is to check how the asymptotic (in particle number) variance of $\sqrt{N}\left(\mathbb{Q}_{\theta,n}^N - \mathbb{Q}_{\theta,n}\right)(S_n)$ changes with time n. The asymptotic analysis will give us an idea about what will happen when we use a large number of particles. We would like the order of the variance to grow less than quadratically in time n; since then the variance of $\sqrt{N}\left(\mathbb{Q}_{\theta,n}^N - \mathbb{Q}_{\theta,n}\right)(n^{-1}S_n)$, which is the variance of the estimates in the M-step, is not only time uniformly bounded but also vanishes. This should result in the variability of the EM's parameter update step to particle realisation also diminishing over time. Before proceeding further we shall make clear that our analysis is for the approximation $\mathbb{Q}_{\theta,n}^N$ defined in (4.17) for a fixed θ. That is, our results are *only* indicative of the stability of the sufficient statistics calculated in the SMC-FS online EM algorithm, which actually uses a changing sequence of θ's. In summary, our main result in this section establishes that (under certain assumptions) the asymptotic (in particle number)

variance of $\sqrt{N}\left(\mathbb{Q}_{\theta,n}^N - \mathbb{Q}_{\theta,n}\right)(S_n)$ is upper bounded by a term $\mathcal{O}(n)$ or $\mathcal{O}(n\log^2 n)$. The tighter $\mathcal{O}(n)$ bound is for finite duration models while the looser $\mathcal{O}(n\log^2 n)$ bound is for infinite duration models.

The results in this section are phrased for any fixed θ and any sequence of observations $y = \{y_n\}_{n\geq 1}$. Also, to keep the notation "light" θ is omitted from the subscripts. Some basic definitions are provided first. For a real valued functions $\varphi : \mathcal{X} \to \mathbb{R}$, let $\|\varphi\|_A = \sup_{x\in A}|\varphi(x)|$ for $A \subseteq \mathcal{X}$. Let $\mathcal{B}(\mathcal{X})$ denote the space of bounded real valued functions on $\mathcal{X}$. For a probability measure ν on $\mathcal{X}$, let $\nu(\varphi) = \sum_{x\in\mathcal{X}} \nu(x)\ \varphi(x)$, and for $A \in \mathcal{X}$, $\nu(A) = \nu(I_A)$ where I_A is the indicator function for the set A such that $I_A(x) = 1$ if $x \in A$, 0 otherwise. Denote the support of ν by $\mathrm{supp}(\nu) = \{x \in \mathcal{X} : \nu(x) > 0\}$. If $M(x, x')$ is a transition probability (from x to x') on $\mathcal{X}$, let $(M\varphi)(x) = M(\varphi)(x) = \sum_{x'} M(x, x')\varphi(x')$. For $\varphi \in \mathcal{B}(\mathcal{X})$ and $A \subseteq \mathcal{X}$, let $\mathrm{osc}_A(\varphi) = \sup_{x,x'\in A}|\varphi(x) - \varphi(x')|$ be the oscillation of the function over A and $\mathrm{osc}(\varphi) = \mathrm{osc}_{\mathcal{X}}(\varphi)$. The complement of a set A is $\overline{A}$.

We will require the following result concerning the asymptotic variance of particle smoothing [Del Moral et al., 2010].

Theorem 4.1. *Given $y = \{y_n\}_{n\geq 1}$, assume there exists finite constants c_n such that $c_n^{-1} \leq G_n \leq c_n$ for all n. For any $n \geq 1$, $F_n \in \mathcal{B}(\mathcal{X}^n)$, $\sqrt{N}\left(\mathbb{Q}_n^N - \mathbb{Q}_n\right)(F_n)$ converges in law, as $N \to \infty$, to a centered Gaussian random variable with variance*

$$\sum_{i=1}^{n} \eta_i\left(\left[G_{i,n}\ D_{i,n}(F_n - \mathbb{Q}_n(F_n))\right]^2\right). \tag{4.18}$$

where, for $1 \leq i \leq n$, the potential function $G_{i,n}$ and the bounded integral operator $D_{i,n}$ are

$$G_{i,n}(x_i) := \frac{p(y_{i:n-1}|x_i, y_{1:i-1})}{p(y_{i:n-1}|y_{1:i-1})}, \quad D_{i,n}(F_n)(x_i) := \mathbb{E}\left[F_n(X_{1:n})|\, y_{1:n-1}, x_i\right].$$

The assumption that the potentials G_n are uniformly bounded below by c_n^{-1} is not overly restrictive as it is satisfied when $g_m(y|z) > 0$ for all m, y and z. The latter is a typical assumption in the context of the analysis of particle filters to avoid the possibility of all the particles having weight zero [Del Moral, 2004].

In order to discuss the rate of growth of the asymptotic variance (4.18) as a function of time n, we need to quantify the sensitivity of the forward and backward smoothers to their initialisations. For a given sequence of observations $y_{1:n}$, the *forward smoother* is defined as the Markov chain on $\mathcal{X}$ with transition kernel $p(x_{k+1}|x_k, y_{1:n})$, $k = 1, \ldots, n-1$. Similarly, the *backward smoother* is the reverse time Markov chain with transition kernel $p(x_k|x_{k+1}, y_{1:n})$, $k = n-1, n-2, \ldots, 1$. Each term of the sum in (4.18) is an integral over $\mathcal{X}^n$ and will typically grow linearly with n unless both the forward and backward smoother forget their initialisations quick enough (e.g. with geometric rate) and the class of functions F_n is restricted. Indeed the E-step of the EM algorithm computes the

expectation for not an arbitrary F_n but one that has a specific additive structure; see Section 4.3.1, also Proposition 4.1. A definition of geometric rate is as follows. Given $\{y_i\}_{i\geq 1}$, if for some integer $L > 0$ there exists a finite constant $c(L) \geq 1$ such that for all $m - k \geq L$, $n \geq m$,

$$|\mathbb{E}\left[s(X_m)|\, x_k, y_{1:n}\right] - \mathbb{E}\left[s(X_m)|\, x'_k, y_{1:n}\right]| \leq \mathrm{osc}(s)(1 - c(L)^{-2})^{\left\lfloor \frac{m-k}{L} \right\rfloor} \tag{4.19}$$

irrespective of (x_k, x'_k) provided both conditional expectations are well defined, then the forward smoother is said to forget its initialisation with geometric rate. (A similar definition applies for the backward smoother; see (4.32)). Henceforth, when we say *forward forgetting* we mean that the forward smoother forgets its initial condition in the sense of (4.19) but without any specific reference to a rate. By backward forgetting, similarly, we will mean the insensitivity of the backward smoother to its initialisation.

A typical route to establish forward and backward forgetting is to exploit the fact that the Markov chain $\{X_k\}_{k\geq 1}$ satisfies a majorization and a minorization condition: that is there exists a probability measure $m(x)$, positive integer l and positive constant c such that $c^{-1}m(x_k) \leq p(x_k|x_{k-l}) \leq c\, m(x_k)$ for all $(x_{k-l}, x_k) \in \mathcal{X}^2$. When this condition is satisfied it may be shown that the backward and forward smoothers forget their initialisations at geometric rate, which is quick enough such each term of the sum (4.18) is uniformly bounded over time. For changepoint models however, the majorization-minorization condition is not satisfied in general. Consider the following example: let $R = 1$ (in which case we drop the variable m_k from x_k, i.e. $x_k = d_k$) and

$$X_k = \begin{cases} x_{k-1} + 1 & \text{w.p.} \quad 1 - \lambda \\ 1 & \text{w.p.} \quad \lambda \end{cases} \tag{4.20}$$

Furthermore, given $X_k = d$ then it must be that $X_{k-i} = d - i$ for $i < d$. Thus the distance between the probability distributions $\Pr(X_{k-i}|X_k = d)$ and $\Pr(X_{k-i}|X_k = d')$ will not decrease at geometric rate and the same cannot be expected for the backward smoother (which is essentially these laws but with additional conditioning on $y_{1:k-1}$.) In this work, we analyse the asymptotic variance for changepoint models using a slightly refined approach.

We analyse two types of changepoint models separately, namely *finite duration* changepoint models and *infinite duration* changepoint models. We distinguish between the two models as follows. In a finite duration changepoint model, for each $m \in \{1, \ldots, R\}$ there exists some finite $\bar{d}_m$ such that $\lambda_m(d) = 1$ for all $d \geq \bar{d}_m$, and smallest such $\bar{d}_m$ is the maximum duration length for model m. If, for at least one $m \in \{1, \ldots, R\}$, $\lambda_m(d) < 1$ for all $d > 0$, then the model is called an infinite duration model.

Given $\{y_n\}_{n\geq 1}$, for positive integers $k \geq 1$, (lag) l and set $A \subseteq \mathcal{X}$, let

$$c_{k,l}(A) = \sup_{\substack{x_{k+l}\in A,\\ x_k, x'_k \in \text{supp}(\eta_k)}} \frac{p(x_{k+l}, y_{k:k+l-1}|x_k, y_{1:k-1})}{p(x_{k+l}, y_{k:k+l-1}|x'_k, y_{1:k-1})} \tag{4.21}$$

where $c_{k,l}$ is taken to be infinity if the denominator can be made zero while the numerator is not. By convention $0/0 = 1$. The variables x_k and x'_k range over $\text{supp}(\eta_k)$ to ensure the conditional expectations in the numerator and denominator are well defined. Also, we abbreviate $c_{k,l}(\mathcal{X})$ to $c_{k,l}$. The variance result is now stated for additive functions of the form $S_k(x_{1:k}) = \sum_{i=1}^{k} s_i(x_i)$ and may be extended to the case where $S_k(x_{1:k}) = \sum_{i=1}^{k} s_i(x_{i-1}, x_i)$. The proof of the result is based on some supporting results and is given in Appendix 4.A.3.

Proposition 4.1. *Assume* $S_n(x_{1:n}) = \sum_{k=1}^{n} s_k(x_k)$ *where* $\text{osc}(s_k) \leq 1$.

- *If* $\{X_k\}$ *is a finite duration changepoint model which is irreducible and aperiodic; and there exists a finite constant* c *such that* $c^{-1} \leq G_n \leq c$ *for all* n*, then the asymptotic variance of* $\sqrt{N}\left(\mathbb{Q}_n^N - \mathbb{Q}_n\right)(S_n)$ *given in (4.18) is upper bounded by a term* $\mathcal{O}(n)$.

- *Assume* $\{X_k\}$ *is an infinite duration changepoint model whose forward smoother forgets its initialisation at geometric rate in the sense of (4.19). Furthermore, let* $A = \{1, \ldots, L\} \times \{1, \ldots, R\}$. *If there exist a finite positive constant* c *such that* $c^{-1} \leq G_n \leq c$ *for all* n *and finite positive constants* C, $\gamma \in (0, 1)$ *and* c' *such that for all* n *and* L

$$\sup_{i\geq 1} \eta_i(\overline{A}) \leq C\gamma^L, \qquad \textit{and} \qquad \sup_{i\geq 1} c_{i,L}(A) \leq c', \tag{4.22}$$

then the asymptotic variance of $\sqrt{N}\left(\mathbb{Q}_n^N - \mathbb{Q}_n\right)(S_n)$ *is upper bounded by* $\mathcal{O}(n \log^2 n)$.

The first condition in (4.22) is a uniform tightness condition on the probabilities η_i, whereas the second condition means that if a changepoint occurs between times k and $k + L$, the observations up to the last changepoint prior to time $k + L$ do not favour one x_k over another too much. Proposition 4.1 is now shown to be applicable to the infinite duration model in (4.20) with the following example whose verification is shown in Appendix 4.A.3.1.

Example 4.2. *For the infinite duration model in (4.20), recall that* Z_k *(see Section 4.2) is a Markov process that "resets" itself when* X_k *returns to state 1, i.e.*

$$Z_k \,|(x_{1:k}, z_{1:k-1}) \sim \begin{cases} \pi(z_k)dz_k & \text{if } x_k = 1 \\ f(z_k|z_{k-1})dz_k & \text{otherwise} \end{cases}$$

We will assume that the process $\{Z_k\}_{k\geq 1}$ *assumes values from a compact space and that there exists some positive constant* c *such that for all* (z_{k-1}, z_k)

$$c^{-1/2} \leq \pi(z_k) \leq c^{1/2}, \qquad c^{-1/2} \leq f(z_k|z_{k-1}) \leq c^{1/2}. \tag{4.23}$$

Furthermore, assume $g(y_k|z_k) > 0$ *for all* z_k, y_k. *For example, a changepoint model satisfying these assumptions could be the changepoint model in Example 4.1 in Section 4.2 with* $R = 1$ *and instead of a static* $\{Z_k\}_{k\geq 1}$ *process, a slowly moving one which is "mixing". Note that a slowly moving* $\{Z_k\}_{k\geq 1}$ *process permits a more parsimonious representation of the data.*

4.5 Numerical examples

4.5.1 Simulated experiments

For the experiments in this section, we will use the infinite duration changepoint model in Example 4.1 in Section 4.2, where $\theta = (\xi_{1:R}, \kappa_{1:R}, \lambda_{1:R}, \alpha, \beta, P)$. The constituent distributions of this model belong to the exponential family and so (4.5) holds; see Appendix 4.A.2 for details.

4.5.1.1 Online EM applied to long data sequence

We applied Algorithm 4.1 to a data sequence of length 500000 generated by the model in Example 4.1 with $R = 2$ and parameter values $\alpha = 10, \beta = 0.1, \xi_1 = 1.445, \xi_2 = -0.214, \kappa_1 = 1.588, \kappa_2 = 0.379, \lambda_1 = 0.12, \lambda_2 = 0.09, P_{ij} = 0.5, i, j = 1, 2$. The M-step was not executed for the first 2000 points (i.e. $n_b = 2000$). The step-size sequence was $\gamma_n = n^{-0.8}$. Figure 4.1 shows the trace parameter estimates over time. We observe that the algorithm converges towards the true values. We also did multiple runs to check that the algorithm would not only converge to a local maximum.

4.5.1.2 Comparison between online and batch EM for a short data sequence

Figure 4.1 also suggests that online EM requires a long data sequence for convergence. Therefore, for short data sequences the algorithm may not converge and its potential use is questionable. One can of course use the batch EM algorithm in such cases but another solution might be to apply online EM to the concatenated sequence $\{y_{1:K}, y_{1:K}, \ldots\}$. By doing so, the online EM solution is not 'online' anymore. However, it can still be significantly faster than the offline version as we demonstrate below. Figures 4.2 and 4.3 show results for such a scenario for 2000 data points. We used Algorithm 4.1 to obtain the results in Figure 4.2 by replicating $y_{1:K}$ 100 times and the SMC-FS batch EM

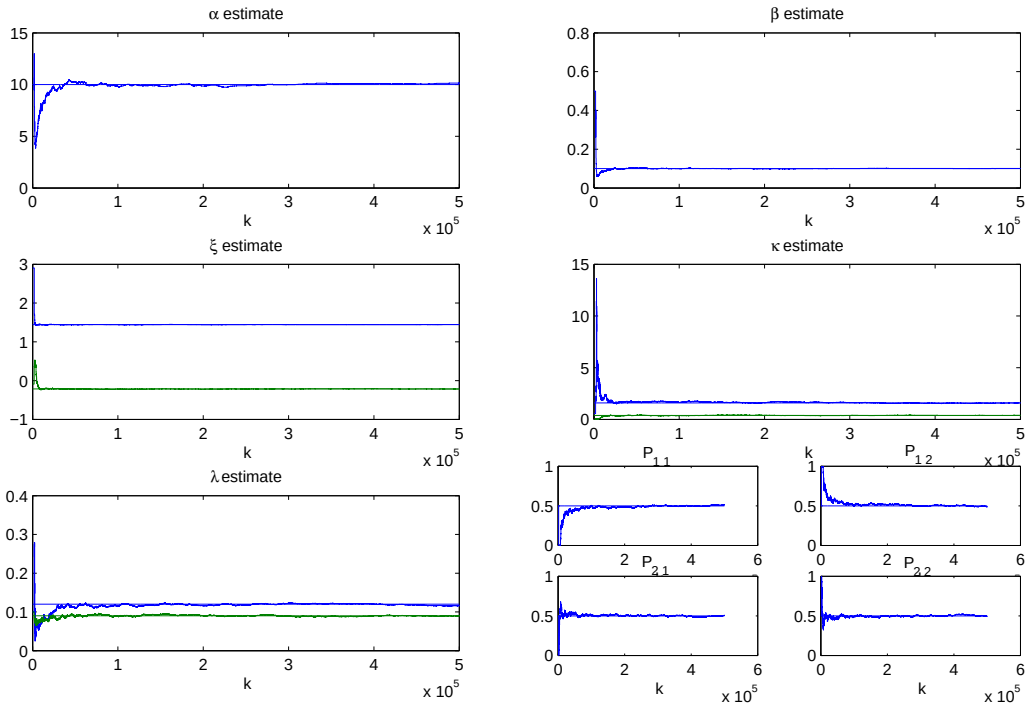


Figure 4.1: SMC-FS online EM estimates vs time for a long simulated data sequence. The true parameter values are indicated with a horizontal line.

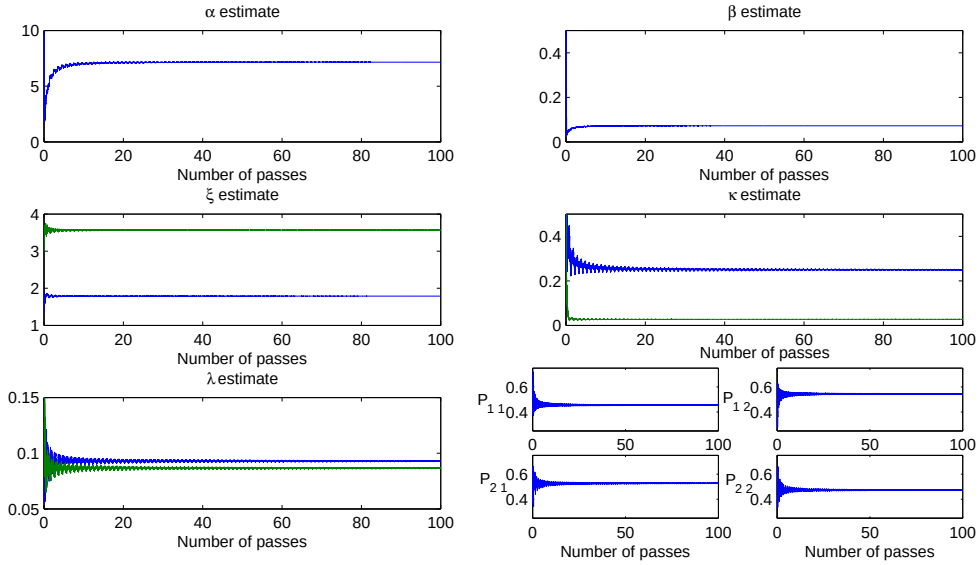


Figure 4.2: SMC-FS online EM estimates vs number of passes for the concatenated data set $\{y_{1:2000}, y_{1:2000}, \ldots\}$ where each pass is one complete browse of $y_{1:2000}$. The true parameter values: $\alpha = 10$, $\beta = 0.1$, $\xi_1 = 1.78, \xi_2 = 3.56, \kappa_1 = 0.30, \kappa_2 = 0.03, \lambda_1 = \lambda_2 = 0.1, P_{i,j} = 0.5$.

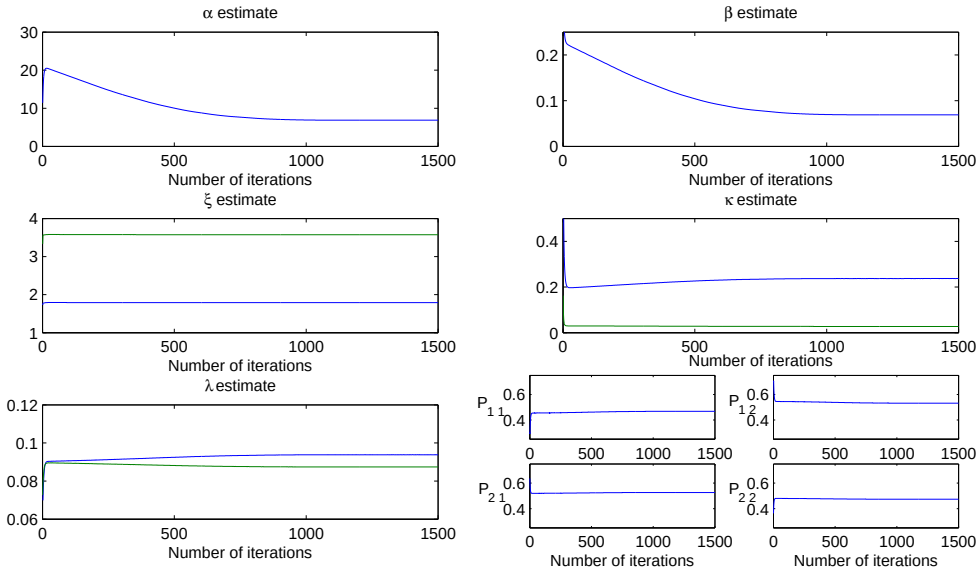


Figure 4.3: SMC-FS batch EM estimates vs number of iterations for for the same $y_{1:2000}$ used to produce the results in Figure 4.2.

algorithm (the batch version of SMC-FS online EM) to obtain the results in Figure 4.3. The true parameter values are $\alpha = 10$, $\beta = 0.1$, $\xi_1 = 1.78, \xi_2 = 3.56, \kappa_1 = 0.30, \kappa_2 = 0.03, \lambda_1 = \lambda_2 = 0.1, P_{i,j} = 0.5, i, j = 1, 2$.

There are two main outcomes to be stressed from the results in Figures 4.2 and 4.3. First, the online EM algorithm in this example is much faster since it converges after around 50 passes, whereas the batch EM algorithm needs over 1000 iterations for convergence. Notice that the computational cost of one pass over the data in the online case and one iteration in the batch case are almost the same and therefore the comparison makes sense. Second, the parameter estimates of both algorithms converge to almost the same points. This empirically validates the potential benefit of the online EM algorithm even in the offline setting.

4.5.1.3 Comparison with the path space method

As stated in Section 4.3.3, other than the SMC-FS online EM algorithm, it is possible to devise an online EM algorithm using $\mathbb{Q}_{\theta,n}^{p,N}$ (SMC-PS online EM), but it suffers from higher variance. In the following, we compare the performances of these two online EM algorithms.

In the first experiment, we compare the variability in the estimates of the sufficient statistics of the changepoint model defined above when the SMC-FS online EM algorithm and the SMC-PS online EM algorithm (see Section 4.3.4) are used with θ_n frozen to θ. We show the results for only one of the statistics, $S_{6,n}^1$, required for the EM algorithm (see Appendix 4.A.2) in Figure 4.4. The figures are obtained after running 100 Monte Carlo

simulations for the same sequence of observation data. For illustration purposes, while the box plots show the estimates up to time 10000, we show the relative variance along 100000 time steps. We can deduce from the box-plots and relative variance that there is much less variability in the estimates obtained by using forward smoothing and the SMC-FS method always outperforms the SMC-PS method in time and thus should be favoured. Note that, using a finite number of particles, these SMC estimates are biased and will result in a loss of accuracy in the EM algorithms. To assess this bias, studies in the context of Feynman-Kac formulae are helpful. For example, the result in Del Moral et al. [2009] suggests that the bias of SMC-FS estimate of S_n/n for finite duration models is bounded by a term $\mathcal{O}(1/N)$.

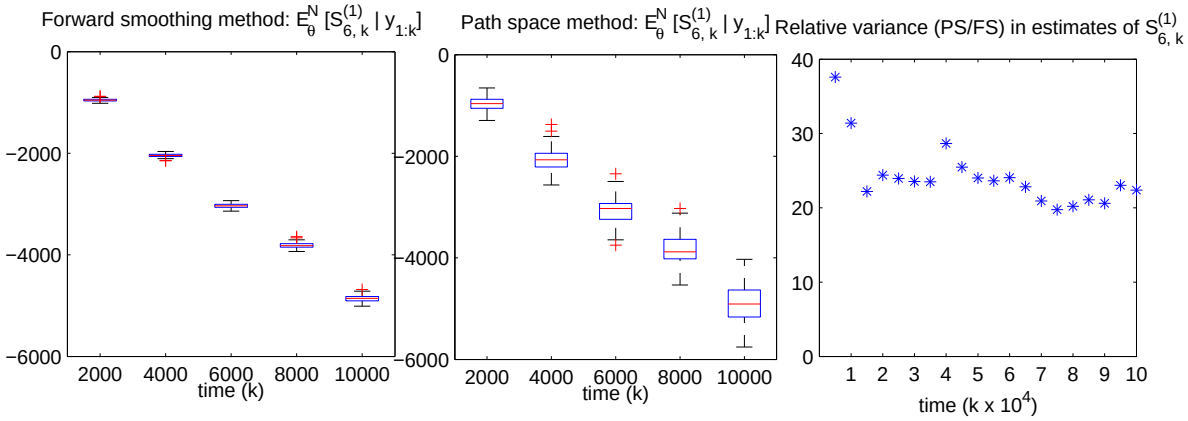


Figure 4.4: Comparison of the forward smoothing and the path space methods in terms of the variability in the estimates of $S_{6,n}^1$. The box plots and the relative variance plot are generated from 100 Monte Carlo simulations using the same observation data.

The second experiment compares the variability in the parameter estimates of the SMC-FS online EM and the SMC-PS online EM algorithms. Figure 4.5 shows the estimation results for the parameter λ_1 when the two algorithms are used. The results are obtained from 100 Monte Carlo simulations using the same sequence of observation data of length 10000. It is interesting to observe that the trends of estimates over time are similar for both algorithms; however, it is obvious from the box plots as well as the relative variance over time that the SMC-FS online EM estimates have less variance than the SMC-PS online EM estimates.

4.5.2 GC content in the DNA of Human Chromosome no. 2

We applied our online EM method to estimate the parameters of a changepoint model used for modelling the Guanine+Cytosine (GC) content along human chromosome. It appears that many features of the genome are correlated with GC content, such as gene density, repeat density, substitution rates, and recombination rates; see Fearnhead and

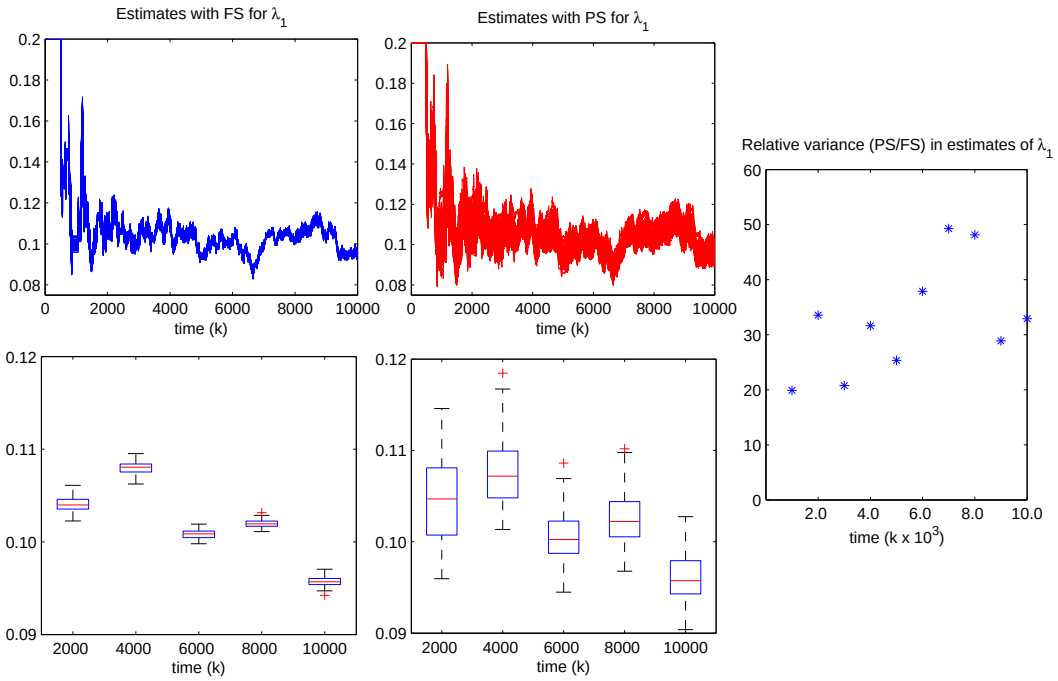


Figure 4.5: Comparison of SMC-FS online EM and SMC-PS online EM in terms of the variability in their estimates of $\lambda_1 = 0.1$. The two plots at the top generated by superimposing different estimates, the box plots, and the relative variance plot are generated from estimates out of 100 different Monte Carlo runs using the same observation data.

Vasileiou [2009] and the references therein for further explanation. It is assumed that the chromosome is separated into successive segments by changepoints and the GC content during each segment is constant. However, as the signal is obscured by small scale noise, a statistical approach may be used to uncover the sequence of changepoints. There is a commonly used binary segmentation approach implemented within the program IsoFinder [Oliver et al., 2004]. Fearnhead and Vasileiou [2009] proposed the changepoint model described in Example 4.1. Regarding the model variables, $Z_k = (Z_{k,1}, Z_{k,2})$ were interpreted as the mean and variance of the GC content during the segment at window k, and Y_k was taken to be the observed GC content of the k'th window. The authors estimated the model parameters by using a MCEM approach and their results outperformed the ones obtained using IsoFinder.

In our experiments we used human chromosome 2, which can be downloaded via the link `http://hgdownload.cse.ucsc.edu/goldenPath/hg17`. The raw data was preprocessed as follows. The raw data consists of a single contiguous stretch of DNA data containing only four different letters: A, C, G, and T. We summarised the DNA data by partitioning the 24 Megabase (Mb) region, which is nearly the whole data set, into 80000 windows, each 3.0 kb long, and for each window recording the proportion of letters within that window that are G or C. Some parts of the DNA sequence could not be measured leading to missing parts. The noisy GC content with missing parts, which

we use as the observation sequence, is shown in Figure 4.6. We assumed two generative

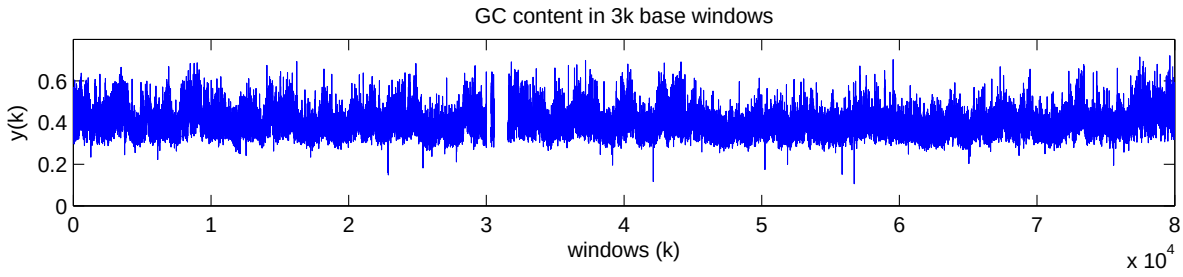


Figure 4.6: Noisy GC content over 3 kb windows in human DNA chromosome 2.

models ($R = 2$) to represent segments of high and low GC contents. The missing data problem is straightforward to handle, e.g. see Fearnhead and Vasileiou [2009]. Figure 4.7 shows the online EM parameter estimates versus number of passes over the data obtained with Algorithm 4.1. One can see that most of the parameter estimates converge after 10 passes, whereas for convergence of the rest 30 passes are enough.

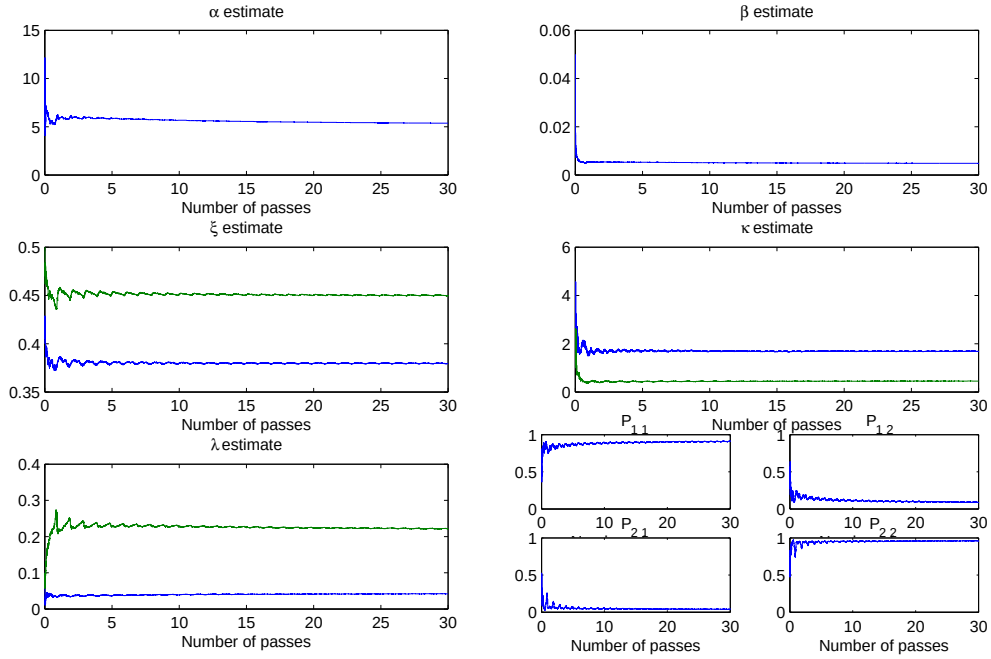


Figure 4.7: Online EM estimates vs number of passes over the data sequence in Figure 4.6.

4.6 Discussion

We have presented a novel SMC online EM algorithm for changepoint models and we have studied the stability of the associated SMC estimates. The proposed EM algorithm

does not require the filters to be stored and has memory requirements independent of the size of the dataset. We have shown that it is practical for very long data sequences, and it can outperform the batch EM even when the data length is not so long that batch EM is impractical (in terms of memory requirement to store the filters and the entire data set).

From a Monte Carlo point of view, our SMC implementation of the forward smoothing recursion at the core of the online EM algorithm is essentially an online implementation of the forward-filtering backward-smoothing algorithm of Doucet et al. [2000b] where the filtering densities are approximated using SMC and then backward smoothing is executed exactly. This method is more efficient than using the path space method as demonstrated in Section 4.5.1.3. Since we need only the SMC approximation of the filters, we could even use more effective SMC routines that are not applicable to a path space method; see for example the SMC algorithm in Fearnhead and Vasileiou [2009]. Besides, unlike the general state-space model case [Del Moral et al., 2009], the computational cost of our algorithm is of the same order as the cost of using a path space method in changepoint models.

Even though the numerical examples were presented for one specific changepoint model, our online EM algorithm is also applicable to the changepoint models studied in Whiteley et al. [2009] and Caron et al. [2011]. More generally, the proposed online EM algorithm is applicable when the constituent laws of the changepoint model given in (4.1)-(4.2) belong to the exponential family and the latent variable $\{Z_k\}_{k\geq 1}$ can be integrated out analytically.

4.A Appendix

4.A.1 Derivation of H_k in (4.4)

Given $\{x_k\}_{k\geq 1}$, consider the partition of $\{1, 2, \ldots\}$ $\{[t_1, t_2), [t_2, t_3), \ldots\}$ where t_i is the i'th time when $d_k = 1$. Each set $[t_n, t_{n+1})$ is called a segment. To emphasise the segmented structure of the changepoint model, we define $a_k = \sum_{i=1}^{k} I_{\{1\}}(d_k)$ to be the number of segments up to time k, $l_n = t_{n+1} - t_n$ to be the length of the n'th segment, and $\bar{m}_n = m_{t_n}$ to be the model number in the n'th segment. Also, we define $\bar{Z}_n = Z_{t_n:t_{n+1}-1}$ and $\bar{Y}_n = Y_{t_n:t_{n+1}-1}$ to group the variables Z_k and Y_k that belong to the same segment with shorthand notation. Recall that

$$H_k(x_{1:k}, y_{1:k}, \theta_i, \theta) = \log p_\theta(x_{1:k}) + \mathbb{E}_{\theta_i}\left[\log p_\theta(y_{1:k}, Z_{1:k}|x_{1:k})\middle|\, y_{1:k}, x_{1:k}\right]. \tag{4.24}$$

Proposition 4.2. *For any changepoint model defined as in Section 4.2, we have*

$$H_k(x_{1:k}, \theta', \theta) = H_{k-1}(x_{1:k-1}, \theta', \theta) + h_k(x_{k-1}, x_k, \theta', \theta)$$

Proof. Since $\{X_k\}_{k\geq 1}$ is a Markov chain, so $\log p_\theta(x_{1:k}) = \log p_\theta(x_{1:k-1}) + \log p_\theta(x_k|x_{k-1})$, and we are done for the first term in (4.24). For the second term in (4.24), due to the conditional independence of $(\bar{Z}_n, \bar{Y}_n)$ given the model number at the segment n, which is $\bar{m}_n$, we have

$$p_{\theta'}(z_{1:k}|y_{1:k}, x_{1:k}) = \left[\prod_{n=1}^{a_k-1} p_{\theta'}(\bar{z}_n|\bar{y}_n, \bar{m}_n)\right] p_{\theta'}(z_{k-d_k+1:k}|y_{k-d_k+1:k}, m_k) \tag{4.25}$$

$$\log p_\theta(y_{1:k}, z_{1:k}|x_{1:k}) = \left[\sum_{n=1}^{a_k-1} \log p_\theta(\bar{y}_n, \bar{z}_n|\bar{m}_n)\right] + \log p_\theta(y_{k-d_k+1:k}, z_{k-d_k+1:k}|m_k) \tag{4.26}$$

Combining (4.25) and (4.26), we have

$$H_k(x_{1:k}, \theta', \theta) = \log p_\theta(x_{1:k}) + \mathbb{E}_{\theta'}\left[\sum_{n=1}^{a_k-1} \log p_\theta(\bar{y}_n, \bar{Z}_n|\bar{m}_n)\middle| \bar{y}_n, \bar{m}_n\right] + \mathbb{E}_{\theta'}\left[\log p_\theta(y_{k-d_k+1:k}, Z_{k-d_k+1:k}|m_k)\middle| y_{k-d_k+1:k}, m_k\right]$$

Now consider H_{k-1}. Given d_{k-1}, there are two possibilities for d_k, either $d_k = 1$, $d_k = d_{k-1} + 1$.

- If $d_k = 1$, it means a new segment starts at time k. Therefore, $a_k = a_{k-1} + 1$ and the a_{k-1}'th segment ends at time $k-1$. This gives $H_{k-1}(x_{1:k-1}, \theta', \theta)$ being equal to
$$\log p_\theta(x_{1:k-1}) + \mathbb{E}_{\theta'}\left[\sum_{n=1}^{a_k-1} \log p_\theta(\bar{y}_n, \bar{Z}_n|\bar{m}_n)\middle| \bar{y}_n, \bar{m}_n\right]$$
- If $d_k = d_{k-1} + 1$, then we are still at the segment at which we were at time $k-1$. Therefore, we have $a_k = a_{k-1}$, $m_k = m_{k-1}$, and $H_{k-1}(x_{1:k-1}, \theta', \theta)$ is equal to
$$\log p_\theta(x_{1:k-1}) + \mathbb{E}_{\theta'}\left[\sum_{n=1}^{a_k-1} \log p_\theta(\bar{y}_n, \bar{Z}_n|\bar{m}_n)\middle| \bar{y}_n, \bar{m}_n\right] + \mathbb{E}_{\theta'}\left[\log p_\theta(y_{k-d_k:k-1}, Z_{k-d_k:k-1}|m_k)\middle| y_{k-d_k+1:k-1}, m_k\right]$$

Therefore, we have $H_k(x_{1:k}, \theta', \theta) = H_{k-1}(x_{1:k-1}, \theta', \theta) + h_k(x_{k-1}, x_k, \theta', \theta)$ where

$$h_k(x_{k-1}, x_k, \theta', \theta) = \log p_\theta(x_k|x_{k-1}) + \begin{cases} \mathbb{E}_{\theta'}\left[\log p_\theta(y_k, Z_k|m_k)\middle|\, y_k, m_k\right], & \text{if } d_k = 1 \\ \mathbb{E}_{\theta'}\left[\log p_\theta(y_{k-d_k+1:k}, Z_{k-d_k+1:k}|m_k)\middle|\, y_{k-d_k+1:k}, m_k\right] & \\ -\mathbb{E}_{\theta'}\left[\log p_\theta(y_{k-d_k+1:k-1}, Z_{k-d_k+1:k-1}|m_k)\middle|\, y_{k-d_k+1:k-1}, m_k\right], & \text{if } d_k = d_{k-1}+1 \end{cases}$$

which does not depend on the values of x_1 to x_{k-2}. □

4.A.2 Derivation of the EM algorithm for the model in Section 4.5

We write $(Z_1, Z_2) \sim \mathcal{N}\Gamma^{-1}(\xi, \kappa, \alpha, \beta)$ to mean $Z_2 \sim \Gamma^{-1}(\alpha, \beta)$ and $Z_1|z_2 \sim \mathcal{N}(\xi, \frac{z_2}{\kappa})$. If $Y_k|\,(z_1, z_2) \sim \mathcal{N}(z_1, z_2)$ for $k = 1, \ldots, n$, the marginal likelihood and the posterior are:

$$p(y_{1:n}) = \frac{\pi^{-n/2}\,(2\beta)^\alpha\,\Gamma\,(\alpha + n/2)}{\left(2\beta + \sum_{k=1}^n y_k^2 + \xi^2\kappa - \frac{\sum_{k=1}^n y_k + \xi^2\kappa}{n+\kappa}\right)^{n/2+\alpha}}$$

$$(Z_1, Z_2)\,|\,(y_{1:n}) \sim \mathcal{N}\Gamma^{-1}\left(\frac{\kappa\xi + n\bar{y}}{\kappa + n}, \kappa + n, \alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum_{k=1}^n (y_k - \bar{y})^2 + \frac{n\kappa}{n+\kappa}\frac{(\bar{y}^2 - \xi)^2}{2}\right)$$

where $\bar{y} = \frac{1}{n}\sum_{k=1}^n y_k$. Also, the required expectations are analytically available:

$$\mathbb{E}\,[1/Z_2] = \alpha/\beta,\ \mathbb{E}\,[Z_1/Z_2] = \xi\alpha/\beta,\ \mathbb{E}\left[Z_1^2/Z_2\right] = 1/\kappa + \xi^2\alpha/\beta,\ \mathbb{E}\,[\log Z_2] = \log\beta - \Psi(\alpha)$$

For the EM algorithm, we estimate the following functionals for $m, m_1, m_2 = 1, \ldots, R$:

$$S_{1,k}^m(x_{1:k}, \theta_i) = \sum_{n:\bar{m}_n = m}^{a_k} 1, \quad S_{2,k}^m(x_{1:k}, \theta_i) = \sum_{n:\bar{m}_n = m}^{a_k - 1} (l_n - 1) + I_{\{m\}}(m_k)\,(d_k - 1),$$

$$S_{3,k}^{m_1,m_2}(x_{1:k}, \theta_i) = \sum_{n:\bar{m}_n = m_1, \bar{m}_{n+1} = m_2}^{a_k - 1} 1$$

$$S_{4,k}^m(x_{1:k}, \theta_i) = \sum_{n:\bar{m}_n = m}^{a_k - 1} \mathbb{E}_{\theta_i}\left[\log Z_{t_n,2}\middle|\, \bar{y}_n, m\right] + I_{\{m\}}(m_k)\mathbb{E}_{\theta_i}\left[\log Z_{k,2}\middle|\, y_{k-d_k+1:k}, m\right],$$

$$S_{5,k}^m(x_{1:k}, \theta_i) = \sum_{n:\bar{m}_n = m}^{a_k - 1} \mathbb{E}_{\theta_i}\left[1/Z_{t_n,2}\middle|\, \bar{y}_n, m\right] + I_{\{m\}}(m_k)\mathbb{E}_{\theta_i}\left[1/Z_{t_n,2}\middle|\, y_{k-d_k+1:k}, m\right],$$

$$S_{6,k}^m(x_{1:k}, \theta_i) = \sum_{n:\bar{m}_n = m}^{a_k - 1} \mathbb{E}_{\theta_i}\left[Z_{t_n,1}/Z_{t_n,2}\middle|\, \bar{y}_n, m\right] + I_{\{m\}}(m_k)\mathbb{E}_{\theta_i}\left[Z_{t_n,1}/Z_{t_n,2}\middle|\, y_{k-d_k+1:k}, m\right],$$

$$S_{7,k}^m(x_{1:k}, \theta_i) = \sum_{n:\bar{m}_n = m}^{a_k - 1} \mathbb{E}_{\theta_i}\left[Z_{t_n,1}^2/Z_{t_n,2}\middle|\, \bar{y}_n, m\right] + I_{\{m\}}(m_k)\mathbb{E}_{\theta_i}\left[Z_{t_n,1}^2/Z_{t_n,2}\middle|\, y_{k-d_k+1:k}, m\right].$$

The corresponding additive functions are

$$
\begin{aligned}
&s^m_{1,k}(x_{k-1},x_k,\theta_i) = I_{\{m\}}(m_k)I_{\{1\}}(d_k) \quad s^m_{2,k}(x_{k-1},x_k,\theta_i) = I_{\{m\}}(m_k)I_{\{d_{k-1}+1\}}(d_k),\\
&s^{m_1,m_2}_{3,k}(x_{k-1},x_k,\theta_i) = I_{\{1\}}(d_k)I_{\{m_1\}}(m_{k-1})I_{\{m_2\}}(m_k),\\
&s^m_{4,k}(x_{k-1},x_k,\theta_i) = I_{\{m\}}(m_k)\left\{I_{\{1\}}(d_k)\mathbb{E}_{\theta_i}\left[\log Z_{k,2}\middle|\, y_k,m\right]\right.\\
&\qquad\left. + I_{\{d_{k-1}+1\}}(d_k)\left(\mathbb{E}_{\theta_i}\left[\log Z_k\middle|\, y_{k-d_k+1:k},m\right] - \mathbb{E}_{\theta_i}\left[\log Z_{k,2}\middle|\, y_{k-d_k+1:k-1},m\right]\right)\right\},\\
&s^m_{5,k}(x_{k-1},x_k,\theta_i) = I_{\{m\}}(m_k)\left\{I_{\{1\}}(d_k)\mathbb{E}_{\theta_i}\left[1/Z_{k,2}\middle|\, y_k,m\right]\right.\\
&\qquad\left. +I_{\{d_{k-1}+1\}}(d_k)\left(\mathbb{E}_{\theta_i}\left[1/Z_{k,2}\middle|\, y_{k-d_k+1:k},m\right] - \mathbb{E}_{\theta_i}\left[1/Z_{k,2}\middle|\, y_{k-d_k+1:k-1},m\right]\right)\right\},\\
&s^m_{6,k}(x_{k-1},x_k,\theta_i) = I_{\{m\}}(m_k)\left\{I_{\{1\}}(d_k)\mathbb{E}_{\theta_i}\left[Z_{k,1}/Z_{k,2}\middle|\, y_k,m\right]\right.\\
&\qquad\left. +I_{\{d_{k-1}+1\}}(d_k)\left(\mathbb{E}_{\theta_i}\left[Z_{k,1}/Z_{k,2}\middle|\, y_{k-d_k+1:k},m\right] - \mathbb{E}_{\theta_i}\left[Z_{k,1}/Z_{k,2}\middle|\, y_{k-d_k+1:k-1},m\right]\right)\right\},\\
&s^m_{7,k}(x_{k-1},x_k,\theta_i) = I_{\{m\}}(m_k)\left\{I_{\{1\}}(d_k)\mathbb{E}_{\theta_i}\left[Z^2_{k,1}/Z_{k,2}\middle|\, y_k,m\right]\right.\\
&\qquad\left. +I_{\{d_{k-1}+1\}}(d_k)\left(\mathbb{E}_{\theta_i}\left[Z^2_{k,1}/Z_{k,2}\middle|\, y_{k-d_k+1:k},m\right] - \mathbb{E}_{\theta_i}\left[Z^2_{k,1}/Z_{k,2}\middle|\, y_{k-d_k+1:k-1},m\right]\right)\right\}.
\end{aligned}
$$

The maximisation step is as follows: Letting $\widehat{S}^m_{j,n}(\theta) = \mathbb{E}_\theta\left[S^m_{j,n}(X_{1:n},\theta)\middle|\, y_{1:n}\right]$,

$$
\begin{aligned}
&\alpha^{(i+1)} = \Psi^{-1}\left(\frac{\log\beta^{(i)}\sum_{m=1}^R \widehat{S}^m_{1,n}(\theta_i) + \sum_{m=1}^R \widehat{S}^m_{4,n}(\theta_i)}{\sum_{m=1}^R \widehat{S}^m_{1,n}(\theta_i)}\right),\ \beta^{(i+1)} = \alpha^{(i+1)}\frac{\sum_{m=1}^R \widehat{S}^m_{1,n}(\theta_i)}{\sum_{m=1}^R \widehat{S}^m_{5,n}(\theta_i)}\\
&\xi^{(i+1)}_m = \widehat{S}^m_{6,n}(\theta_i)/\widehat{S}^m_{5,n}(\theta_i),\ \kappa^{(i+1)}_m = \widehat{S}^m_{1,n}(\theta_i)/\left(\widehat{S}^m_{7,n}(\theta_i) - 2\xi^{(i+1)}_m\widehat{S}^m_{6,n}(\theta_i) + \xi^{(i+1)2}_m\widehat{S}^m_{5,n}(\theta_i)\right)\\
&\lambda^{(i+1)}_m = \widehat{S}^m_{1,n}(\theta_i)/\left(\widehat{S}^m_{2,n}(\theta_i) + \widehat{S}^m_{1,n}(\theta_i)\right),\ P^{(i+1)}_{m_1,m_2} = \widehat{S}^{m_1,m_2}_{3,n}(\theta_i)/\sum_{m=1}^R \widehat{S}^{m_1,m}_{3,n}(\theta_i)
\end{aligned}
$$

where $\Psi(x) = d\log\Gamma(x)/dx$ is the derivative of the log-gamma function.

4.A.3 Proof of Proposition 4.1

We will first establish a weaker form of backward forgetting for the infinite duration model with the aid for the following lemma, whose proof is straightforward and is omitted.

Lemma 4.1. *Let $M(x,x')$ be a Markov transition kernel (from x to x') on $\mathcal{X}$, c a constant and m a probability measure on $\mathcal{X}$. If $c^{-1}\,m(x') \le M(x,x') \le c\,m(x')$ for all $x \in A$, where $A \subseteq \mathcal{X}$, then for any $B \subseteq \mathcal{X}$ and $\varphi \in \mathcal{B}(\mathcal{X})$ such that $\operatorname{osc}(\varphi) \le 1$,*

$$\operatorname{osc}_A(M(\varphi)) \le (1-c^{-1})\operatorname{osc}_B(\varphi) + 2c\,m\left(\overline{B}\right),$$

Corollary 4.1. *Assume $\{y_i\}_{i\ge1}$ is given with $p(y_{1:n}) > 0$ for all n. Let $\varphi_n(x_n) = \mathbb{E}\left[s(X_1)\middle|\, x_n, y_{1:n-1}\right]$. For any $L > 0$, $n - L > 0$, $\operatorname{osc}(s) \le 1$, $A \subseteq \operatorname{supp}(\eta_n)$, $B \subseteq \operatorname{supp}(\eta_{n-L})$*

$$\operatorname{osc}_A(\varphi_n) \le \left(1 - c_{n-L,L}(A)^{-1}\right)\operatorname{osc}_B(\varphi_{n-L}) + 2c_{n-L,L}(A)\eta_{n-L}(\overline{B}). \tag{4.27}$$

Furthermore, let $A = \{1, \ldots, L\} \times \{1, \ldots, R\}$. *If there exist finite positive constants* C, $\gamma \in (0, 1)$ *and* $c(L)$ *such that for all* L

$$\sup_{i \geq 1} \eta_i(\overline{A}) \leq C\gamma^L \qquad \textit{and} \qquad \sup_{i \geq 1} c_{i,L}(A) \leq c(L) \tag{4.28}$$

then for all L *large enough, for all* n,

$$\mathrm{osc}_{A \cap \mathrm{supp}(\eta_n)}(\varphi_n) \leq (1 - c(L)^{-1})^{\lfloor \frac{n-1}{L} \rfloor} + 2c(L)^2 C\gamma^L. \tag{4.29}$$

Proof. Substituting $l = L$ and $k = n - L$ in (4.21), it can be shown that

$$c_{n-L,L}(A)^{-1} p\left(x_{n-L} | y_{1:n-L-1}\right) \leq p\left(x_{n-L} | x_n, y_{1:n-1}\right) \leq c_{n-L,L}(A) p\left(x_{n-L} | y_{1:n-L-1}\right)$$

for all $x_n \in A$. The bound (4.27) now follows from Lemma 4.1 with $c = c_{n-L,L}(A)$, $m(x_{n-L}) = p\left(x_{n-L} | y_{1:n-L-1}\right)$, $M(x_n, x_{n-L}) = p(x_{n-L} | y_{1:n-1}, x_n)$, and $\varphi = \varphi_{n-L}$. The second bound (4.29) follows from (4.27) by iterating the backward kernels with $B = A \cap \mathrm{supp}(\eta_{n-L})$, and using the tail behaviour of the minorization measure in (4.28). □

The first condition in (4.28) is a uniform tightness condition on the probabilities η_i. This bound for the tail probabilities can be loosened but only at the expense of a weaker bound in Proposition 4.1. It is clear that (4.29) is weaker than backward forgetting at geometric rate.

Corollary 4.1 presents a weaker form of backward forgetting for the infinite duration model. The following lemma establishes that the finite duration models posses the geometric forward forgetting and geometric backward forgetting properties; both of which are necessary in order to establish linear growth of the variance.

Lemma 4.2. *For a finite duration changepoint model, let* $\bar{d}_m = \min\{d' : \lambda_m(d) = 1, d \geq d'\}$ *be the maximum duration length in model* m *and let* $\mathcal{X}_f = \bigcup_{m=1}^{R} \{(1, m), \ldots, (\bar{d}_m, m)\}$. *Assume that the transition matrix* $\{p(x_k | x_{k-1}) : x_k, x_{k-1} \in \mathcal{X}_f\}$ *is irreducible and aperiodic; and that for the given* $\{y_n\}_{n \geq 1}$ *there exist finite positive constants* c_n *such that* $c_n^{-1} \leq G_n \leq c_n$ *for all* n. *(i) Then there exists a positive integer* L *such that* $c_{k,l}$ *defined in (4.21) is finite for all* $l \geq L$, $k \geq 1$. *(ii) It now follows that for all* $l \geq L$, $n \geq k + l$, *and* $x_{k+l} \in \mathcal{X}_f$,

$$p(x_{k+l} | x_k, y_{1:n}) \geq c_{k,l}^{-2}\, p(x_{k+l} | x'_k, y_{1:n}) \tag{4.30}$$

and the inequality holds irrespective of (x_k, x'_k) *provided both conditional probabilities are well defined. (iii) Furthermore, the Markov chain on* $\mathcal{X}$ *with transition kernel* $p(x_{k+1} | x_k, y_{1:n})$,

$k = 1, \ldots, n-1$, *forgets its initialisation in the following sense: for all* $n \geq m \geq k \geq 1$,

$$|\mathbb{E}\left[s(X_m)|\, x_k, y_{1:n}\right] - \mathbb{E}\left[s(X_m)|\, x'_k, y_{1:n}\right]| \leq \operatorname{osc}(s) \prod_{i=1}^{\lfloor \frac{m-k}{L} \rfloor} (1 - c^{-2}_{k+(i-1)L,L}) \tag{4.31}$$

irrespective of (x_k, x'_k) *provided both conditional expectations are well defined. If* $c_n \leq c < \infty$ *for all* n, *then (iv)* $c_{k,l} \leq c(l) < \infty$ *for* $l \geq L$, $k \geq 1$ *and the rate in (4.31) is geometric, and (v) letting* $\varphi_n(x_n) = \mathbb{E}\left[s(X_1)|\, x_n, y_{1:n-1}\right]$, *for all* $l \geq L$, *for all* n, *all* $A \subseteq \operatorname{supp}(\eta_n)$

$$\operatorname{osc}_A(\varphi_n) \leq \operatorname{osc}(s)(1 - c(l)^{-1})^{\lfloor n/l \rfloor}. \tag{4.32}$$

Proof. (Outline only) Property (i) is a consequence of some well known facts for finite state Markov chains. We use the fact that, under the stated assumptions, the Markov chain restricted to $\mathcal{X}_f$ has a stationary distribution, say $\nu(x)$, and we have $\nu(\mathcal{X}_f) = 1$ and $\nu > 0$ on $\mathcal{X}_f$. This ensures the ratio $p(x_{k+l}|x_k)/p(x_{k+l}|x'_k)$ is close to 1 uniformly in its arguments and k, provided l is large enough. The result now follows from the the fact that G_n is bounded from below and above. Property (ii) follows from (i) while the forgetting property in (4.31) is a simple consequence of (4.30), e.g. see Del Moral [2004]. Property (iv) is proved similarly to (i) using instead the uniform bound on G_n. To verify (v) use (iv) and (4.27), i.e. iterate the backward kernels starting with $B = \operatorname{supp}(\eta_{n-l})$ □

Finally, we will need the following lemma to prove Proposition 4.1

Lemma 4.3. *Given* $\{y_n\}_{n\geq 1}$, *assume there exists a finite constant* c *such that* $c^{-1} \leq G_n \leq c$ *for all* n *and that (4.19) holds then, for all* n, $1 < k \leq n$,

$$\sup_{(x_k, x'_k) \in \operatorname{supp}(\eta_k)} \frac{p(y_{k:n}|x_k, y_{1:k-1})}{p(y_{k:n}|x'_k, y_{1:k-1})} < \infty.$$

Proof. Using $|\log(b) - \log(a)| \leq \frac{|b-a|}{\min(a,b)}$,

$$\begin{aligned} \log \frac{p(y_{k:n}|x_k, y_{1:k-1})}{p(y_{k:n}|x'_k, y_{1:k-1})} &= \sum_{i=k}^{n} \log p(y_i|x_k, y_{1:i-1}) - \log p(y_i|x'_k, y_{1:i-1}) \\ &\leq \sum_{i=k}^{n} \frac{|p(y_i|x_k, y_{1:i-1}) - p(y_i|x'_k, y_{1:i-1})|}{\min(p(y_i|x_k, y_{1:i-1}), p(y_i|x'_k, y_{1:i-1}))}. \end{aligned}$$

Since $p(y_i|x_k, y_{1:i-1}) = \mathbb{E}\left[G_i(X_i)|\, x_k, y_{1:i-1}\right]$, each ratio can be bounded using (4.19) and constant c, which then results in a geometric sum and gives the desired uniform bound. □

We can now present the proof of Proposition 4.1.

Proof. ***(Proposition 4.1):*** The asymptotic variance is

$$\sum_{i=0}^{n} \eta_i \left([G_{i,n}\ D_{i,n}(S_n - \mathbb{Q}_n(S_n))]^2 \right). \tag{4.33}$$

Consider the infinite duration model. Consider the ith term: For any $A \subseteq \mathcal{X}$,

$$\begin{aligned}
\eta_i \left([G_{i,n}\ D_{i,n}(S_n - \mathbb{Q}_n(S_n))]^2 \right) &\leq \|G_{i,n}\|^2_{\mathrm{supp}(\eta_i)}\, \eta_i \left([D_{i,n}(S_n - \mathbb{Q}_n(S_n))]^2 \right) \\
&\leq \|G_{i,n}\|^3_{\mathrm{supp}(\eta_i)} \int \eta_i(dx_i)\eta_i(dx_i') \left([D_{i,n}(S_n)(x_i) - D_{i,n}(S_n)(x_i')]^2 \right) \\
&\leq \|G_{i,n}\|^3_{\mathrm{supp}(\eta_i)} \left(\left[\mathrm{osc}_{A \cap \mathrm{supp}(\eta_i)} D_{i,n}(S_n) \right]^2 + 2n^2 \eta_i(\overline{A}) \right)
\end{aligned} \tag{4.34}$$

Now let $A = \{1, \ldots, L\} \times \{1, \ldots, R\}$. It follows from (4.19) that for some integer L',

$$\sup_{x_i, x_i' \in \mathrm{supp}(\eta_i)} \left| \mathbb{E}\left[\sum_{k=i}^{n} s_k(X_k) \middle| x_i, y_{1:n} \right] - \mathbb{E}\left[\sum_{k=i}^{n} s_k(X_k) \middle| x_i', y_{1:n} \right] \right| \leq L' c(L')^2,$$

and from (4.29) that

$$\sup_{x_i, x_i' \in A \cap \mathrm{supp}(\eta_i)} \left| \mathbb{E}\left[\sum_{k=1}^{i-1} s_k(X_k) \middle| x_i, y_{1:n-1} \right] - \mathbb{E}\left[\sum_{k=1}^{i-1} s_k(X_k) \middle| x_i', y_{1:n-1} \right] \right| \leq (i-1) 2c(L)^2 C \gamma^L I_{[i \geq L]} + Lc(L).$$

Thus using Lemma 4.3 to uniformly bound $\|G_{i,n}\|_{\mathrm{supp}(\eta_i)}$ and the fact that the bounds in (4.28) are satisfied for all L large enough with $c(L) < c' < \infty$, (4.33) can be upper bounded by

$$\begin{aligned}
&C' \sum_{i=1}^{n} \left((i-1)^2 \gamma^{2L} I_{[i \geq L]} + L^2 + (L')^2 + n^2 \eta_i(\overline{A}) \right) \\
&\leq C' n^3 \gamma^{2L} + C' n L^2 + C' n (L')^2 + n^3 C \gamma^L
\end{aligned}$$

where C' is independent of L and n. Setting $L = k \log n$ for some fixed constant k we see that (4.33) is upper bounded by a term $\mathcal{O}(n \log^2 n)$.

The proof for the finite duration model follows the same lines where Lemma 4.2 is used instead of Corollary 4.1, hence it is omitted. □

4.A.3.1 Verification of Example 4.2 satisfying the conditions of Proposition 4.1

The first condition of Theorem 4.1 is satisfied since $g(y_k|z_k) > 0$ for all z_k, y_k. It follows from (4.23) that

$$c^{-1} \leq \frac{\int \prod_{i=1}^{n} f(z_i''|z_{i-1}'')g(y_i|z_i'')\, dz_{1:n}''}{\int \prod_{i=1}^{n} f(z_i'|z_{i-1}')g(y_i|z_i')\, dz_{1:n}'} \leq c$$

for all $n \geq 1$, $y_{1:n}$, z_0' ,z_0''. This, together with (4.20) implies

$$c^{-1} \leq \frac{p(y_{k:n}|x_k, y_{1:k-1})}{p(y_{k:n}|x_k', y_{1:k-1})} \leq c \tag{4.35}$$

for all $(x_k, x_k') \in \text{supp}(\eta_k)$, $k \leq n$. (4.35) now implies the term $\|G_{i,n}\|_{\text{supp}(\eta_i)}$ in (4.34) is also uniformly bounded by the constant c. (Note that the condition $c^{-1} \leq G_n \leq c$ for all n in Proposition 4.1 is used to verify the term $\|G_{i,n}\|_{\text{supp}(\eta_i)}$ in (4.34) is uniformly bounded in n and is now no longer needed for this example as we have direct verification via (4.35).)

Since

$$\begin{aligned} p(x_k|x_{k-1}, y_{1:n}) &\propto p(y_{k:n}|x_k, x_{k-1}, y_{1:k-1})p(x_k|x_{k-1}, y_{1:k-1}) \\ &= p(y_{k:n}|x_k, y_{1:k-1})p(x_k|x_{k-1}), \end{aligned}$$

we have that

$$p(x_k|x_{k-1}, y_{1:n}) \geq c^{-1}p(x_k|x_{k-1}) \geq c^{-1}\lambda\, \delta_1(x_k) \tag{4.36}$$

for all $k \leq n$, and obviously for $k > n$ too. To establish forward forgetting, it follows from the minorization condition in (4.36) that

$$\mathbb{E}\left[s_k(X_k)|\, x_1, y_{1:n}\right] - \mathbb{E}\left[s_k(X_k)|\, x_1', y_{1:n}\right] \leq \text{osc}(s_k)\left(1 - c^{-1}\lambda\right)^{k-1}.$$

Let $A = \{1, \ldots, L\}$. For $x_{k+L} \in A$, $x_k \in \text{supp}(\eta_k)$ and $x_k' \in \text{supp}(\eta_k)$,

$$\frac{p(x_{k+L}, y_{k:k+L-1}|x_k, y_{1:k-1})}{p(x_{k+L}, y_{k:k+L-1}|x_k', y_{1:k-1})} = \frac{p(y_{k:k+L-1}|x_{k+L}, x_k, y_{1:k-1})}{p(y_{k:k+L-1}|x_{k+L}, x_k', y_{1:k-1})} \frac{p(x_{k+L}|x_k)}{p(x_{k+L}|x_k')}.$$

By (4.35), the first ratio is bounded by c. The second ratio is 1. Thus $c_{k,L}(A) \leq c$. Using (4.36), $\sup_{i \geq L+1} \mathbb{E}\left[I_{\overline{A}}(X_i)|\, y_{1:i-1}\right] \leq \gamma^L$ where $\gamma = 1 - c^{-1}\lambda$. Hence the bounds in (4.22) apply with constants independent of L and n.

Chapter 5

Estimating the Static Parameters in Linear Gaussian Multiple Target Tracking Models

Summary: *In this we work we propose both offline and online maximum likelihood estimation (MLE) techniques for inferring the static parameters of a multiple target tracking (MTT) model with linear Gaussian dynamics. We present the batch and online versions of the expectation-maximisation (EM) algorithm for short and long data sets respectively, and we show how Monte Carlo approximations of these methods can be implemented. Performance is assessed in numerical examples using simulated data for various scenarios.*

The material in this chapter resembles my contribution in the extended work Yıldırım et al. [2012b]. Also, an early version of this work is published in Yıldırım et al. [2012c]. I was introduced to the problem studied in this chapter by Dr. Sumeetpal S. Singh and Dr. Thomas Dean.

5.1 Introduction

The multiple target tracking (MTT) problem concerns the analysis of data from multiple moving objects which are partially observed in noise to extract highly reliable motion trajectories. The MTT framework has been traditionally applied to solve surveillance problems but more recently there has been a surge of interest in Biological Signal Processing, e.g. see Yoon and Singh [2008].

The MTT framework is comprised of the following ingredients. A set of multiple independent targets moving in the surveillance region in a Markov fashion. The number of targets varies over time due to departure of existing targets (known as death) and the arrival of new targets (known as birth). The initial number of targets are unknown and the maximum number of targets present at any given time is unrestricted. At each time each target may generate an observation which is a noisy record of its *state*. Targets that do not generate observations are said to be undetected at that time. Additionally,

there maybe spurious observations generated which are unrelated to targets (known as clutter). The observation set at each time is the collection of all target generated and false measurements recorded at that time, but without any information on the origin or association of the measurements. False measurements, unknown origin of recorded measurements, undetected targets and a time varying number of targets renders the task of extracting the motion trajectory of the underlying targets from the observation record, which is known as *tracking* in the literature, a highly challenging problem.

There is a large body of work on the development of algorithms for tracking multiple moving targets. These algorithms can be categorised by how they handle the data association (or unknown origin of recorded measurements) problem. Among the main approaches are the Multiple Hypothesis Tracking (MHT) algorithm [Reid, 1979] and the probabilistic MHT (PMHT) variant [Streit and Luginbuhi, 1995], the joint probabilistic data association filter (JPDAF) [Bar-Shalom and Fortmann, 1988; Bar-Shalom and Li, 1995], and the probability hypothesis density (PHD) filter [Mahler, 2003; Singh et al., 2009]. With the advancement of Monte Carlo methodology, sequential Monte Carlo (SMC) (or particle filtering) and Markov chain Monte Carlo (MCMC) methods have been applied to the MTT problem, e.g. SMC and MCMC implementations of JPDAF [Hue et al., 2002; Vermaak et al., 2005], SMC implementations for MHT and PMHT [Ng et al., 2005; Oh et al., 2009], and SMC implementations of the PHD filter [Vo et al., 2003, 2005; Whiteley et al., 2010], to mention a few.

Compared to the huge amount of work on developing tracking algorithms, the problem of estimating the static parameters of the tracking model has been largely neglected, although it is rarely the case that these parameters are known. Some exceptions include the work of Storlie et al. [2009] where they extended the MHT algorithm to simultaneously estimate the parameters of the MTT model. A full Bayesian approach for estimating the model parameters using MCMC was presented in Yoon and Singh [2008]. Recently, Singh et al. [2011] presented an approximated maximum likelihood method derived by using a Poisson approximation for the posterior distribution of the hidden targets which is also central to the derivation of PHD filter in Mahler [2003]. Additionally, versions of PHD and Cardinalised PHD (CPHD) filters that can learn the clutter rate and detection profile while filtering were proposed in Mahler et al. [2011].

In this chapter, we present maximum likelihood estimation (MLE) algorithms to infer the static parameters of the MTT model when the individual targets move according to a linear Gaussian state-space model and when the target generated observations are linear functions of the target state corrupted with additive Gaussian noise; we will henceforth call this a linear Gaussian MTT model. We maximise the likelihood function using the expectation-maximisation (EM) algorithm and we present both online and batch EM algorithms. Because we assume a linear Gaussian MTT model, we are able to

present the exact recursions for updating static parameter estimate. We stress though that these recursions are not obvious primarily because the MTT model allows for false measurements, unknown origin of recorded measurements, undetected targets and a time varying number of targets with unknown birth and death times. To the best of our knowledge, this is a novel development in the target tracking field. To implement the proposed EM algorithms, an estimate of the posterior distribution of the hidden targets given the observations is required, and in the linear Gaussian setting, the continuous values of the target states can be marginalised out. But, because the number of possible association of observations to targets grows very quickly with time, we have to resort to approximation schemes that focus the computation in the expectation(E)-step of the EM algorithms on the most likely associations; that is, we approximate the E-step with a Monte Carlo method. For this we employ both SMC which give rise to the following different MLE algorithms:

- SMC-EM algorithm for offline estimation; and
- SMC online EM algorithm for online estimation.

We implement these two algorithms for simulated examples under various tracking scenarios and provided recommendations for practitioner on which one is to be preferred.

The EM algorithms we present in this chapter can be implemented with any Monte Carlo scheme for inferring the target states in MTT and reducing the errors in the approximation of the E-step can only be beneficial to the EM parameter estimates. We do not fully explore the use of the various Monte Carlo target tracking algorithms that have been proposed in the literature and instead focus on the following. When using SMC to approximate the E-step, we compute the L-best assignments [Murty, 1968] as the sequential proposal scheme of the particle filter. This L-best assignments approached has appeared previously in the literature in the context of tracking, e.g. see Cox and Miller [1995]; Danchick and Newnam [2006]; Ng et al. [2005]. An alternative approach, for example, could be to approximate the E-step by using the MCMC data association (MCMC-DA) algorithm proposed for target tracking in Oh et al. [2009]. Also a full Bayesian estimation approach has been proposed by Yoon and Singh [2008].

The remainder of the chapter is organised as follows. In Section 5.2, we describe the MTT model and formulate the static parameter estimation problem. In Section 5.3, we present the batch and online EM algorithms. Section 5.4 contains the numerical examples and we conclude the chapter with a discussion of our findings in Section 5.5. The Appendix contains further details on the derivation of the EM algorithms for MTT, and details of the SMC algorithm we use in this chapter. We also make an attempt to analyse the computational complexity of the EM algorithms in the Appendix.

5.1.1 Notation

We introduce random variables (also sets and mappings) with capital letters such as $X, Y, Z, \mathbf{X}, A$ and denote their realisations by corresponding small case letters $x, y, z, \mathbf{x}, a$. If a random variable X has a density $\nu(x)$, with all densities being defined w.r.t. the Lebesgue measure (denoted by dx), we write $X \sim \nu(\cdot)$ to make explicit the law of X. We use $\mathbb{E}_\theta[\cdot|\cdot]$ for the (conditional) expectation operator; for random variables X, Y and Z and a function $(x, y) \to f(x, y)$, $\mathbb{E}_\theta[f(X, Z)|Y = y]$ is the expectation of the random variable $f(X, Z)$ w.r.t. the joint distribution of X, Z conditioned on $Y = y$. $\mathbb{E}_\theta[f(X, z)|y]$ is the expectation of the function $x \to f(x, z)$ for a fixed z given $Y = y$.

5.2 Multiple target tracking model

Consider a single target tracking model where a moving object (or target) is observed when it traverses in a surveillance region. We define the target state and the noisy observation at time t to be the random variables $X_t \in \mathcal{X} \subset \mathbb{R}^{d_x}$ and $Y_t \in \mathcal{Y} \subset \mathbb{R}^{d_y}$ respectively. The statistical model most commonly used for the evolution of individual targets $\{X_t, Y_t\}_{t\geq 1}$ is the hidden Markov model (HMM). In a HMM, it is assumed that $\{X_t\}_{t\geq 1}$ is a hidden Markov process with initial and transition probability densities μ_ψ and f_ψ, respectively, and $\{Y_t\}_{t\geq 1}$ is the observation process with the conditional observation density g_ψ, i.e.

$$\begin{aligned} &X_1 \sim \mu_\psi(\cdot), \quad X_t|(X_{1:t-1} = x_{1:t-1}) \sim f_\psi(\cdot|x_{t-1}) \\ &Y_t|\left(\{X_i = x_i\}_{i\geq 1}, \{Y_i = y_i\}_{i\neq t}\right) \sim g_\psi(\cdot|x_t). \end{aligned} \tag{5.1}$$

Here the densities μ_ψ, f_ψ and g_ψ are parametrised by a real valued vector $\psi \in \Psi \subset \mathbb{R}^{d_\psi}$. In this work, we consider a specific type of HMM, the Gaussian linear state-space model (GLSSM), which can be specified as

$$\mu_\psi(x) = \mathcal{N}(x; \mu_b, \Sigma_b), \quad f_\psi(x'|x) = \mathcal{N}(x'; Fx, W), \quad g_\psi(y|x) = \mathcal{N}(y; Gx, V). \tag{5.2}$$

where $\mathcal{N}(x; \mu, \Sigma)$ denotes the probability density function for the multivariate normal distribution with mean μ and covariance Σ. In this case, ψ parametrizes $(\mu_b, \Sigma_b, F, G, W, V)$.

In a MTT model, the state and the observation at each time ($t \geq 1$) are random finite sets, $\mathbf{X}_t = \left\{X_{t,1}, X_{t,2}, \ldots, X_{t,K_t^x}\right\}$ and $\mathbf{Y}_t = \left\{Y_{t,1}, Y_{t,2}, \ldots, Y_{t,K_t^y}\right\}$. Here each element of $\mathbf{X}_t$ is the state of an individual target and elements of $\mathbf{Y}_t$ are the distinct measurements of these targets at time t. The number of targets K_t^x under surveillance changes over time due to targets entering and leaving the surveillance region $\mathcal{X}$. $\mathbf{X}_t$ evolves to $\mathbf{X}_{t+1}$ as follows: with probability p_s each target $\mathbf{X}_t$ 'survives' and is displaced according to the

state transition density f_ψ in (5.2), otherwise it dies. The random deletion and Markov motion happens independently for all the elements of $\mathbf{X}_t$. In addition to the surviving targets, new targets are created. The number of new targets created per time follows a Poisson distribution with mean λ_b and each of their states is initiated independently according to the initial density μ_ψ in (5.2). Now $\mathbf{X}_{t+1}$ is defined to be the superposition of the states of the surviving and evolved targets from time t and the newly born targets at time $t+1$. The points of $\mathbf{X}_t$ are observed through the following model: with probability p_d, each point of $\mathbf{X}_t$ generates a noisy observation in the observation space $\mathcal{Y}$ through the observation density g_ψ in (5.2). This happens independently for each point of $\mathbf{X}_t$. In addition to these target generated observations, false measurements are also generated. The number of false measurements collected at each time follows a Poisson distribution with mean λ_f and their values are uniform over $\mathcal{Y}$. $\mathbf{Y}_t$ is the superposition of observations originating from the detected targets and these false measurements.

A series of random variables, which are essential for the statistical analysis to follow are now defined. Let C_t^s be a $K_{t-1}^x \times 1$ vector of 1's and 0's where 1's indicate survivals and 0's indicate deaths of targets at time t. More clearly, for $i = 1, \ldots, K_{t-1}^x$,

$$C_t^s(i) = \begin{cases} 1 & i\text{'th target at time } t-1 \text{ survives to time } t \\ 0 & i\text{'th target at time } t-1 \text{ does not survive to time } t \end{cases}.$$

The number of surviving targets at time t is $K_t^s = \sum_{i=1}^{K_{t-1}^x} C_t^s(i)$. We also define the $K_t^s \times 1$ vector I_t^s containing the indices of surviving targets at time t,

$$I_t^s(i) = \min\left\{ k : \sum_{j=1}^{k} C_t^s(j) = i \right\}, \quad i = 1, \ldots, K_t^s.$$

Note that $I_t^s(i)$ denotes the ancestor of target i from time $t-1$, i.e. $X_{t-1,I_t^s(i)}$ evolves to $X_{t,i}$ for $i = 1, \ldots, K_t^s$. Denoting the number of 'births' at time n as K_t^b, we have $K_t^x = K_t^s + K_t^b$. Note that according to these definitions, the surviving targets from time $t-1$ are re-labeled as $X_{t,1}, \ldots, X_{t,K_t^s}$, and the newly born targets are denoted as $X_{t,K_t^s+1}, \ldots, X_{t,K_t^x}$. Next, given K_t^x targets we define C_t^d to be a $K_t^x \times 1$ vector of 1's and 0's where 1's indicate detections and 0's indicate non-detections. For $i = 1, \ldots, K_t^x$,

$$C_t^d(i) = \begin{cases} 1 & i\text{'th target at time } t \text{ is detected at time } t \\ 0 & i\text{'th target at time } t \text{ is not detected at time } t \end{cases}.$$

Therefore, the number of detected targets at time t is $K_t^d = \sum_{i=1}^{K_t^x} C_t^d(i)$. Similarly, we

also define the $K_t^d \times 1$ vector I_t^d showing the indices of the detected targets,

$$I_t^d(i) = \min\left\{k : \sum_{j=1}^{k} C_t^d(j) = i\right\}, \quad i = 1, \ldots, K_t^d.$$

$I_t^d(i)$ denotes the label of the i-th detected target at time t. So the detected targets at time t are $X_{t,I_t^d(1)}, \ldots, X_{t,I_t^d(K_t^d)}$. Finally, defining the number of false measurements at time t as K_t^f, we have $K_t^y = K_t^d + K_t^f$ and the association from the detected targets to the observations can be represented by a one-to-one mapping

$$A_t : \{1, \ldots, K_t^d\} \to \{1, \ldots, K_t^y\}$$

where at time t the i'th detected target is target $I_t^d(i)$ with state value $X_{t,I_t^d(i)}$ and generates $Y_{t,A_t(i)}$. We assume that A_t is uniform over the set of all $K_t^y!/K_t^f!$ possible one-to-one mappings. To summarise, we give the list of the random variables in the MTT model introduced in this section as well as a sample realisation of them in Figure 5.1.

The main difficulty in an MTT problem is that in general we do not know birth-death times of targets, whether they are detected or not, and which observation point in $\mathbf{Y}_t$ is associated to which detected target in $\mathbf{X}_t$. Let

$$Z_t = \left(C_t^s, C_t^d, K_t^b, K_t^f, A_t\right)$$

be the collection of the just mentioned unknown random variables at time t, and

$$\theta = (\psi, p_s, p_d, \lambda_b, \lambda_f) \in \Theta = \Psi \times [0,1]^2 \times [0,\infty)^2$$

be the vector of the MTT model parameters. We can write the joint likelihood of all the random variables of the MTT model up to time n given θ as

$$p_\theta(z_{1:n}, \mathbf{x}_{1:n}, \mathbf{y}_{1:n}) = p_\theta(z_{1:n}) p_\theta(\mathbf{x}_{1:n}|z_{1:n}) p_\theta(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}, z_{1:n})$$

where

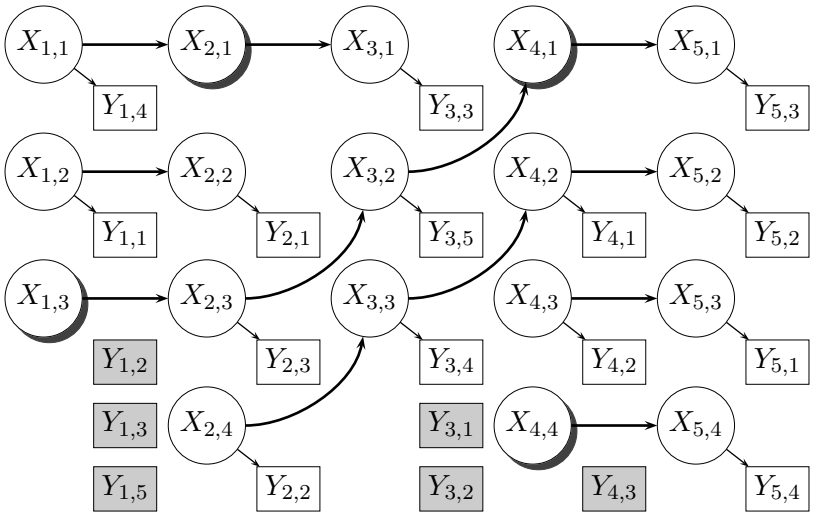
$$p_\theta(z_{1:n}) = \prod_{t=1}^{n} \left(p_s^{k_t^s}(1-p_s)^{k_{t-1}^x - k_t^s} \mathcal{PO}(k_t^b; \lambda_b) p_d^{k_t^d}(1-p_d)^{k_t^x - k_t^d} \mathcal{PO}(k_t^f; \lambda_f) \frac{k_t^f!}{k_t^y!} \right) \tag{5.3}$$

$$p_\theta(\mathbf{x}_{1:n}|z_{1:n}) = \prod_{t=1}^{n} \left(\prod_{j=1}^{k_t^s} f_\psi(x_{t,j}|x_{t-1,i_t^s(j)}) \prod_{j=k_t^s+1}^{k_t^x} \mu_\psi(x_{t,j}) \right) \tag{5.4}$$

$$p_\theta(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}, z_{1:n}) = \prod_{t=1}^{n} \left(|\mathcal{Y}|^{-k_t^f} \prod_{j=1}^{k_t^d} g_\psi(y_{t,a_t(j)}|x_{t,i_t^d(j)}) \right) \tag{5.5}$$

The list of the variables in the MTT model

$X_{t,k}$, $Y_{t,k}$: k'th target and k'th observation at time t.
$\mathbf{X}_t = \{X_1, \ldots, X_{K_t^x}\}$, $\mathbf{Y}_t = \{Y_{t,1}, \ldots, Y_{t,K_t^y}\}$: Sets of targets and observations at time t.
K_t^b, K_t^f: Numbers of newborn targets and false measurements at time t
K_t^s, K_t^d: Numbers of targets survived from time $t-1$ to time t and detected at time t.
K_t^x, K_t^y: Numbers of alive targets, observations at time t. $K_t^x = K_t^s + K_t^b$, $K_t^y = K_t^d + K_t^f$.
C_t^s: $K_{t-1}^x \times 1$ vector of 0's and 1's indicating survivals from time $t-1$ to time t.
C_t^d: $K_t^x \times 1$ vector of 0's and 1's indicating detections at time t.
I_t^s: $K_t^s \times 1$ vector of indices of surviving targets from time $t-1$ to time t.
I_t^d: $K_t^d \times 1$ vector of indices of detected targets at time t.
$A_t : \{1, \ldots, K_t^d\} \to \{1, \ldots, K_t^y\}$: Association from detected targets to observations at time t.
$Z_t = (C_t^s, C_t^d, K_t^b, K_t^f, A_t)$



$C_{1:5}^s = ([\,], [1,1,1], [1,0,1,1], [0,1,1], [1,1,1,1])$; $I_{1:5}^s = ([\,], [1,2,3], [1,3,4], [2,3], [1,2,3,4])$; $C_{1:5}^d = ([1,1,0], [0,1,1,1], [1,1,1], [0,1,1,0], [1,1,1,1])$; $I_{1:5}^d = ([1,2], [2,3,4], [1,2,3], [2,3], [1,2,3,4])$; $K_t^s = (0,3,3,2,4)$; $K_{1:5}^b = (3,1,0,2,0)$; $K_t^d = (2,3,3,2,4)$; $K_{1:5}^f = (3,0,2,1,0)$, $A_{1:5} = ([4,1], [1,3,2], [3,5,4], [1,2], [3,2,1,4])$.

Figure 5.1: Top: The list of the random variables in the MTT model. Bottom: A realisation for an MTT model: States of a targets are connected with arrows. Also, observations generated from targets are connected to those targets with arrows. Misdetected targets are highlighted with shadows, and observations from false measurements are coloured with grey.

Here $\mathcal{PO}(k;\lambda)$ denotes the probability mass function of the Poisson distribution with mean λ, $|\mathcal{Y}|$ is the volume (w.r.t. the Lebesgue measure) of $\mathcal{Y}$ and the term $k_t^f!/k_t^y!$ in (5.3) corresponds to the law of A_t. The marginal likelihood of the observation sequence $\mathbf{y}_{1:n}$ is

$$p_\theta(\mathbf{y}_{1:n}) = \mathbb{E}_\theta \left[p_\theta(\mathbf{y}_{1:n} | \mathbf{X}_{1:n}, Z_{1:n}) \right]. \tag{5.6}$$

The main aim of this work is, given $\mathbf{Y}_{1:n} = \mathbf{y}_{1:n}$, to estimate the static parameter θ^* where we assume the data is generated by some true but unknown $\theta^* \in \Theta$. Our main contribution is to present the EM algorithms, both batch and online versions, for computing the

MLE of θ^*:

$$\theta_{\mathrm{ML}} = \arg\max_{\theta\in\Theta} p_\theta(\mathbf{y}_{1:n}).$$

5.3 EM algorithms for MTT

In this section we present the batch and online EM algorithms for linear Gaussian MTT models. The notation is involved and we provide a list of some important variables used in the derivation of the EM algorithms in Table 5.1 at the end of the section.

5.3.1 Batch EM for MTT

Given $\mathbf{Y}_{1:n} = \mathbf{y}_{1:n}$, the EM algorithm for maximising $p_\theta(\mathbf{y}_{1:n})$ in (5.6) is given by the following iterative procedure: if θ_j is the estimate of the EM algorithm at the j'th iteration, then at iteration $j+1$ the estimate is updated by first calculating the following intermediate optimisation criterion, which is known as the expectation (E) step,

$$\begin{aligned} Q(\theta_j,\theta) &= \mathbb{E}_{\theta_j}\left[\log p_\theta(\mathbf{X}_{1:n}, Z_{1:n}, \mathbf{y}_{1:n})|\mathbf{y}_{1:n}\right] \\ &= \mathbb{E}_{\theta_j}\left[\log p_\theta(Z_{1:n}) + \log p_\theta(\mathbf{X}_{1:n}, \mathbf{y}_{1:n}|Z_{1:n})|\mathbf{y}_{1:n}\right] \\ &= \mathbb{E}_{\theta_j}\left[\log p_\theta(Z_{1:n}) + \mathbb{E}_{\theta_j}\left\{\log p_\theta(\mathbf{X}_{1:n}, \mathbf{y}_{1:n}|Z_{1:n})|\mathbf{y}_{1:n}, Z_{1:n}\right\}|\mathbf{y}_{1:n}\right] \end{aligned} \tag{5.7}$$

The updated estimate is then computed in the maximisation (M) step

$$\theta_{j+1} = \arg\max_{\theta\in\Theta} Q(\theta_j,\theta).$$

This procedure is repeated until θ_j converges (or in practice ceases to change significantly). From equations (5.2)-(5.5), it can be shown that the E-step at the j'th iteration reduces to calculating the expectations of fifteen sufficient statistics of $\mathbf{x}_{1:n}$, $z_{1:n}$ and $\mathbf{y}_{1:n}$ denoted by $S_{1,n},\ldots,S_{15,n}$. (From now on, any dependency on $\mathbf{y}_{1:n}$ in these sufficient statistics and further variables arising from them will be omitted from the notation for simplicity.) Sufficient statistics $S_{1,n}(\mathbf{x}_{1:n}, z_{1:n})$ to $S_{7,n}(\mathbf{x}_{1:n}, z_{1:n})$ are:

$$\begin{aligned} &\sum_{t=1}^{n}\sum_{k=1}^{k_t^d} x_{t,i_t^d(k)} x_{t,i_t^d(k)}^T, \quad \sum_{t=1}^{n}\sum_{k=1}^{k_t^d} x_{t,i_t^d(k)} y_{t,a_t(k)}^T, \quad \sum_{t=2}^{n}\sum_{k=1}^{k_t^s} x_{t-1,i_t^s(k)} x_{t-1,i_t^s(k)}^T, \quad \sum_{t=2}^{n}\sum_{k=1}^{k_t^s} x_{t,k} x_{t,k}^T, \\ &\sum_{t=2}^{n}\sum_{k=1}^{k_t^s} x_{t-1,i_t^s(k)} x_{t,k}^T, \quad \sum_{t=1}^{n}\sum_{k=k_t^s+1}^{k_t^x} x_{t,k}, \quad \sum_{t=1}^{n}\sum_{k=k_t^s+1}^{k_t^x} x_{t,k} x_{t,k}^T. \end{aligned} \tag{5.8}$$

These sufficient statistics are related to those used for estimating the static parameter of a linear Gaussian single target tracking model, and this relation will be made more explicit later. The rest of the sufficient statistics $S_{8,n}(z_{1:n})$ to $S_{15,n}(z_{1:n})$ do not depend

on $\mathbf{x}_{1:n}$.

$$[S_{8,n}, \ldots, S_{15,n}](z_{1:n}) = \sum_{t=1}^{n} \left[\sum_{k=1}^{k_t^d} y_{t,a_t(k)} y_{t,a_t(k)}^T, k_t^d, k_t^x, k_t^s k_{t-1}^x, k_t^b, k_t^f, 1 \right] \tag{5.9}$$

Let $S_{m,n}^{\theta}$ denote the expectation of the m'th sufficient statistic $S_{m,n}$ w.r.t. the law of the latent variables $\mathbf{X}_{1:n}$ and $Z_{1:n}$ of the MTT model given the observation $\mathbf{y}_{1:n}$ for a given θ, i.e.

$$S_{m,n}^{\theta} = \begin{cases} \mathbb{E}_{\theta}\left[S_{m,n}(\mathbf{X}_{1:n}, Z_{1:n}) | \, \mathbf{y}_{1:n} \right] & 1 \leq m \leq 7, \\ \mathbb{E}_{\theta}\left[S_{m,n}(Z_{1:n}) | \, \mathbf{y}_{1:n} \right] & 8 \leq m \leq 15. \end{cases} \tag{5.10}$$

Then the solution to the M-step is given by a known function $\Lambda : \left\{ \left(S_{1,n}^{\theta}, \ldots, S_{15,n}^{\theta} \right) \right\} \to \Theta$ such that at iteration j

$$\theta_{j+1} = \arg\max_{\theta} Q(\theta_j, \theta) = \Lambda\left(S_{1,n}^{\theta_j}, \ldots, S_{15,n}^{\theta_j} \right).$$

The explicit expression of Λ depends on the parametrisation of the MTT model, in particular on the parametrisation of the matrices $F, G, W, V, \mu_b, \Sigma_b$. An example is provided below.

Example 5.1. *(The constant velocity model:) Each target has a position and velocity in the xy-plane and the position of a target is restricted to the window $[-\kappa, \kappa]^2$, hence*

$$X_t = [X_t(1), X_t(2), X_t(3), X_t(4)]^T \in \mathcal{X} = \mathbb{R}^2 \times [0, \infty)^2,$$

where $X_t(1), X_t(2)$ are the x and y coordinates and $X_t(3), X_t(4)$ are the velocities in x and y directions. Only a noisy measurement of the position of the target is available

$$[Y_t(1), Y_t(2)] \in \mathcal{Y} = [-\kappa, \kappa]^2.$$

We assumed a bounded $\mathcal{Y}$ and regard observations that are not recorded due to being outside this interval as also missed detection. With reference to (5.2)*, the state-space of*

the model are:

$$\mu_b = [\mu_{bx}, \mu_{by}, 0, 0]^T, \quad \Sigma_b = \begin{pmatrix} \sigma_{bp}^2 I_{2\times 2} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \sigma_{bv}^2 I_{2\times 2} \end{pmatrix}$$
$$F = \begin{pmatrix} I_{2\times 2} & \Delta I_{2\times 2} \\ \mathbf{0}_{2\times 2} & I_{2\times 2} \end{pmatrix}, \quad G = \begin{pmatrix} I_{2\times 2} & \mathbf{0}_{2\times 2} \end{pmatrix}$$
$$W = \begin{pmatrix} \sigma_{xp}^2 I_{2\times 2} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \sigma_{xv}^2 I_{2\times 2} \end{pmatrix}, \quad V = \sigma_y^2 I_{2\times 2}$$

Therefore, the parameter vector of this MTT model is

$$\theta = \left(\lambda_b, \lambda_f, p_d, p_s, \mu_{bp}, \mu_{bv}, \sigma_{bp}^2, \sigma_{bv}^2, \sigma_{xp}^2, \sigma_{xv}^2, \sigma_y^2\right).$$

The update rule Λ *for* θ *at the M-step of the EM algorithm is*

$$\begin{aligned}
\mu_{bx} &= S_{6,n}^\theta(1)/S_{13,n}^\theta, \quad \mu_{by} = S_{6,n}^\theta(2)/S_{13,n}^\theta, \\
\sigma_{bp}^2 &= \frac{1}{2} S_{13,n}^\theta tr\left(\left(S_{7,n}^\theta - 2S_{6,n}^\theta \mu_b^T + S_{13,n}^\theta \mu_b \mu_b^T\right) M_p^T M_p\right) \\
\sigma_{bv}^2 &= \frac{1}{2} S_{13,n}^\theta tr\left(\left(S_{7,n}^\theta - 2S_{6,n}^\theta \mu_b^T + S_{13,n}^\theta \mu_b \mu_b^T\right) M_v^T M_v\right) \\
\sigma_{xp}^2 &= tr\left(S_{4,n}^\theta M_p M_p^T - 2S_{5,n}^\theta M_p F_p + S_{3,n}^\theta F_p^T F_p\right)/2S_{11,n}^\theta, \\
\sigma_{xv}^2 &= tr\left(S_{4,n}^\theta M_v M_v^T - 2S_{5,n}^\theta M_v F_v + S_{3,n}^\theta F_v^T F_v\right)/2S_{11,n}^\theta, \\
\sigma_y^2 &= tr\left(S_{8,n}^\theta - 2GS_{2,n}^\theta + GS_{1,n}^\theta G\right)/2S_{9,n}^\theta, \\
p_d &= S_{9,n}^\theta/S_{10,n}^\theta, \quad p_s = S_{11,n}^\theta/S_{12,n}^\theta, \\
\lambda_b &= S_{13,n}^\theta/S_{15,n}^\theta, \quad \lambda_f = S_{14,n}^\theta/S_{15,n}^\theta,
\end{aligned}$$

where $M_p = \begin{bmatrix} I_{2\times 2} & 0_{2\times 2} \end{bmatrix}$, $M_v = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \end{bmatrix}$, *and* F_p *and* F_v *are the upper and lower halves of* F, *that is* $F_p(i,j) = F(i,j)$ *and* $F_v(i,j) = F(2+i,j)$ *for* $i = 1, 2$ *and* $j = 1, \ldots, 4$.

5.3.1.1 Estimation of sufficient statistics

It is easy to calculate the expectation of the sufficient statistics in (5.9) that do not depend on $\mathbf{x}_{1:n}$. Noting that Z_t is discrete, we simply calculate $S_{m,n}(z_{1:n})$ for every $z_{1:n}$ with a positive mass w.r.t. to the density $p_\theta(z_{1:n}|\mathbf{y}_{1:n})$ and calculate the expectations as

$$S_{m,n}^\theta = \sum_{z_{1:n}} S_{m,n}(z_{1:n}) p_\theta(z_{1:n}|\mathbf{y}_{1:n}).$$

For those sufficient statistics in (5.8) that depend on $\mathbf{x}_{1:n}$, consider the last expression in (5.7) with the following factorisation of the posterior

$$p_\theta(\mathbf{x}_{1:n}, z_{1:n}|\mathbf{y}_{1:n}) = p_\theta(\mathbf{x}_{1:n}|z_{1:n}, \mathbf{y}_{1:n})p_\theta(z_{1:n}|\mathbf{y}_{1:n}).$$

This factorisation suggests that we can write the required expectations as

$$\begin{aligned} S^\theta_{m,n} &= \mathbb{E}_\theta \left[S_{m,n}(\mathbf{X}_{1:n}, Z_{1:n}) | \, \mathbf{y}_{1:n} \right] \\ &= \mathbb{E}_\theta \left[\mathbb{E}_\theta \left[S_{m,n}(\mathbf{X}_{1:n}, Z_{1:n}) | \, Z_{1:n}, \mathbf{y}_{1:n} \right] | \, \mathbf{y}_{1:n} \right]. \end{aligned} \tag{5.11}$$

Let us define the integrand of the outer expectation in (5.11) which is the conditional expectation

$$\widetilde{S}^\theta_{m,n}(z_{1:n}) = \mathbb{E}_\theta \left[S_{m,n}(\mathbf{X}_{1:n}, z_{1:n}) | \, z_{1:n}, \mathbf{y}_{1:n} \right].$$

as a matrix-valued function with domain $\mathcal{Z}^n$. Then, we can obtain $S^\theta_{m,n}$ by calculating $\widetilde{S}^\theta_{m,n}(z_{1:n})$ for every $z_{1:n}$ with a positive mass w.r.t. the density $p_\theta(z_{1:n}|\mathbf{y}_{1:n})$ and then calculate

$$S^\theta_{m,n} = \sum_{z_{1:n}} \widetilde{S}^\theta_{m,n}(z_{1:n}) p_\theta(z_{1:n}|\mathbf{y}_{1:n}).$$

The crucial point here is that it is possible to calculate $\widetilde{S}^\theta_{m,n}(z_{1:n})$ for any given $z_{1:n}$. In fact, the availability of this calculation is based on the following fact: *conditional on* $\{Z_t\}_{t\geq 1}$, $\{\mathbf{X}_t, \mathbf{Y}_t\}_{t\geq 1}$ *may be regarded as a collection of independent GLSSM's (with different starting and ending times, possible missing observations) and observations which are not relevant to any of these GLSSM's.* In the context of MTT, each GLSSM corresponds to a target and irrelevant observations correspond to false measurements. We defer details on how $\widetilde{S}^\theta_{m,n}(z_{1:n})$ is calculated to Section 5.3.2.

5.3.1.2 Stochastic versions of EM

For exact calculation of the E-step of the EM algorithm we need $p_\theta(z_{1:n}|\mathbf{y}_{1:n})$ which is infeasible to calculate due to the huge cardinality of $\mathcal{Z}^n$. We thus resort to Monte Carlo approximations of $p_\theta(z_{1:n}|\mathbf{y}_{1:n})$ which we then use in the E-step; in literature this approach is known as a stochastic version of the EM algorithm [Celeux and Diebolt, 1985; Delyon et al., 1999; Wei and Tanner, 1990]). We know from the previous sections that given $Z_{1:n} = z_{1:n}$ the posterior distribution $p_\theta(\mathbf{x}_{1:n}|\mathbf{y}_{1:n}, z_{1:n})$ is Gaussian and conditional expectations can be evaluated. Therefore, it is sufficient to have the Monte Carlo approximation for $p_\theta(z_{1:n}|\mathbf{y}_{1:n})$ only, which is expressed as

$$\widehat{p}_\theta(z_{1:n}|\mathbf{y}_{1:n}) = \sum_{i=1}^{N} w_n^{(i)} \delta_{z_{1:n}^{(i)}}(z_{1:n}), \quad \sum_{i=1}^{N} w_n^{(i)} = 1. \tag{5.12}$$

Then, the particle approximations for the expectations of the sufficient statistics are

$$\widehat{S}^{\theta}_{m,n} = \begin{cases} \sum_{i=1}^{N} w_n^{(i)} \widetilde{S}^{\theta}_{m,n}(z_{1:n}^{(i)}), & 1 \leq m \leq 7, \\ \sum_{i=1}^{N} w_n^{(i)} S_{m,n}(z_{1:n}^{(i)}), & 8 \leq m \leq 15. \end{cases}$$

When θ changes with each EM iteration, the appropriate update scheme at iteration j involves a stochastic approximation procedure, where in the E-step one calculates a weighted average of $\widehat{S}^{\theta_1}_{m,n}, \ldots, \widehat{S}^{\theta_j}_{m,n}$; the resulting algorithm is known as the stochastic approximation EM (SAEM) [Delyon et al., 1999]. Specifically, let $\gamma = \{\gamma_j\}_{j \geq 1}$, called the step-size sequence, be a positive decreasing sequence satisfying

$$\sum_j \gamma_j = \infty, \quad \sum_j \gamma_j^2 < \infty.$$

A common choice is $\gamma_j = j^{-\alpha}$ for $0.5 < \alpha \leq 1$. The SAEM algorithm is given in Algorithm 5.1.

Algorithm 5.1. ***The SAEM algorithm for the MTT model***
Start with θ_1 *and* $\widehat{S}^{(0)}_{\gamma,m,n} = 0$ *for* $m = 1, \ldots, 15$. *For* $j = 1, 2, \ldots$

- ***E-step:*** *Calculate* $\widehat{S}^{\theta_j}_{m,n}$ *for each* m, *and calculate the weighted averages*

$$\widehat{S}^{(j)}_{\gamma,m,n} = (1 - \gamma_j)\, \widehat{S}^{(j-1)}_{\gamma,m,n} + \gamma_j \widehat{S}^{\theta_j}_{m,n}. \tag{5.13}$$

- ***M-step*** *Update the parameter estimate using* $\Lambda(\cdot)$ *as before*

$$\theta_{j+1} = \Lambda\left(\widehat{S}^{(j)}_{\gamma,1,n}, \ldots, \widehat{S}^{(j)}_{\gamma,15,n}\right).$$

In general, the Monte Carlo approximation $\widehat{p}_{\theta_j}(z_{1:n}|\mathbf{y}_{1:n})$ in (5.13) is performed either sampling N samples from $p_{\theta_j}(z_{1:n}|\mathbf{y}_{1:n})$ using a SMC method with N particles or using a MCMC method (e.g. the MCMC-DA algorithm of Oh et al. [2009]), in which case weights $w_n^{(i)} = 1/N$, $i = 1, \ldots, N$. In this work, we use the SMC method and we will call the resulting SAEM algorithm SMC-EM. We use SMC to obtain the approximations $\{\widehat{p}_{\theta}(z_{1:t}|\mathbf{y}_{1:t})\}_{1 \leq t \leq n}$ sequentially as follows. Assume that we have the approximation at time $t-1$

$$\widehat{p}_{\theta}(z_{1:t-1}|\mathbf{y}_{1:t-1}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{z_{1:t-1}^{(i)}}(z_{1:t-1}).$$

To avoid weight degeneracy, at each time one can resample from $\widehat{p}_{\theta}(z_{1:t-1}|\mathbf{y}_{1:t-1})$ to obtain a new collection of N particles, each with weight $\bar{w}_{t-1}^{(i)} = 1/N$, and then proceed to the time t. Alternatively, this resampling operation can be done according to a criterion which measures the weight degeneracy (e.g. see Doucet et al. [2000b]). We define the

$N \times 1$ random mapping

$$\Pi_t : \{1, \ldots, N\} \to \{1, \ldots, N\}$$

containing the indices of the resampled particles, i.e. $\Pi_t(i) = j$ if the i'th resampled particle is $z_{1:t-1}^{(j)}$. (If no resampling is performed at the end of time $t-1$, then $\Pi_t(i) = i$, and $\bar{w}_{t-1}^{(i)} = w_{t-1}^{(i)}$ for all i.) Then, given $\mathbf{y}_t$ and $\Pi_t = \pi_t$, the particle $z_t^{(i)}$ at time t is sampled from a proposal distribution

$$q_\theta \left(z_t \left| z_{1:t-1}^{(\pi_t(i))}, \mathbf{y}_{1:t} \right. \right)$$

for $i = 1, \ldots, N$. Therefore, $z_t^{(i)}$ is connected to $z_{1:t-1}^{(\pi_t(i))}$ and the i'th path particle at time t is $z_{1:t}^{(i)} = (z_t^{(i)}, z_{1:t-1}^{(\pi_t(i))})$ and its new weight is

$$w_t^{(i)} \propto \bar{w}_{t-1}^{(\pi_t(i))} \times \frac{p_\theta(z_t^{(i)} | z_{t-1}^{(\pi_t(i))}) p_\theta(\mathbf{y}_t | \mathbf{y}_{1:t-1}, z_{1:t}^{(i)})}{q_\theta(z_t^{(i)} | z_{1:t-1}^{(\pi_t(i))}, \mathbf{y}_{1:t})}. \tag{5.14}$$

Note that we also need to implement SMC for the online EM algorithm in order to obtain a Monte Carlo approximation of the E-step. Our SMC algorithm calculates the L-best linear assignments [Murty, 1968] as the sequential proposal; see Appendix 5.A.2 for details.

5.3.2 Online EM for MTT

We showed in the previous section how to implement the batch EM algorithm for MTT using Monte Carlo approximations. However, the batch EM algorithm is computationally demanding when the data sequence $\mathbf{y}_{1:n}$ is long since one iteration of the EM requires a complete browse of the data. In these situations, the online version of the EM algorithm which updates the parameter estimates as a new data record is received at each time can be a cheaper alternative. In this section, we present a SMC online EM algorithm for linear Gaussian MTT models.

An important observation at this point is that the sufficient statistics of interest for the EM algorithm have a certain additive form such that the difference of $S_{m,n}(\mathbf{x}_{1:n}, z_{1:n})$ and $S_{m,n-1}(\mathbf{x}_{1:n-1}, z_{1:n-1})$ only depends on $(\mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{y}_n)$. This enables us to compute the required expectations in the E-step of the EM algorithm effectively in an online manner. We shall see in this section that, with a fixed amount of computation and memory per time, it is possible to update from $\widetilde{S}_{m,t-1}^\theta(z_{1:t-1})$ to $\widetilde{S}_{m,t}^\theta(z_{1:t})$ given $\mathbf{y}_t$ and z_t at time t. To show how to handle the sufficient statistics in (5.8) for the MTT model, we first start with a single GLSSM and then extend the idea to the MTT case by showing the relation between the sufficient statistics in a single GLSSM and in the MTT model.

5.3.2.1 Online smoothing in a single GLSSM

Consider the HMM $\{X_t, Y_t\}_{t\geq 1}$ defined in (5.1). It is possible to evaluate expectations of additive functionals of $X_{1:n}$ of the form

$$S_n(x_{1:n}) = s(x_1) + \sum_{t=2}^{n} s(x_{t-1}, x_t)$$

(with possible dependency on $y_{1:n}$ also allowed) w.r.t. the posterior density $p_\theta(x_{1:n}|y_{1:n})$ in an online manner using only the filtering densities $\{p_\theta(x_t|y_{1:t})\}_{1\leq t\leq n}$. The technique is based on the following recursion on the intermediate function [Cappé, 2011; Del Moral et al., 2009]

$$\begin{aligned} T_t^\theta(x_t) :=& \mathbb{E}_\theta \left[S_t(X_{1:t})|X_t = x_t, y_{1:t}\right] \\ =& \mathbb{E}_\theta \left[T_{t-1}^\theta(X_{t-1}) + s(X_{t-1}, x_t)\middle|\, y_{1:t-1}, x_t\right] \end{aligned}$$

with the initial condition $T_1^\theta(x_1) = s(x_1)$. Note that the expectation required for the recursion is w.r.t. the backward transition density $p_\theta(x_{t-1}|y_{1:t-1}, x_t)$. The required expectation $\mathbb{E}_\theta\left[S_n(X_{1:n})|y_{1:n}\right]$ can then be calculated as the expectation of the intermediate function $T_n^\theta(x_n)$ w.r.t. the filtering density $p_\theta(x_n|y_{1:n})$, that is,

$$\mathbb{E}_\theta \left[S_n(X_{1:n})\middle|\, y_{1:n}\right] = \mathbb{E}_\theta \left[T_n^\theta(X_n)\middle|\, y_{1:n}\right].$$

Consider now the GLSSM that is defined in (5.2), where, additionally, Y_t is possibly non-observable and C_t^d is the indicator of detection at time t. It is well known that, given $\{(Y_t, C_t^d) = (y_t, c_t^d)\}_{t\geq 1}$, the prediction and filtering densities $p_\theta(x_t|y_{1:t-1}, c_{1:t-1}^d)$ and $p_\theta(x_t|y_{1:t}, c_{1:t}^d)$ are Gaussians with means $\left(\mu_{t|t-1}, \mu_{t|t}\right)$ and covariances $\left(\Sigma_{t|t-1}, \Sigma_{t|t}\right)$ and are updated sequentially as follows:

$$(\mu_{t|t-1}, \Sigma_{t|t-1}) = F\mu_{t-1|t-1}, F\Sigma_{t-1|t-1}F^T + W, \tag{5.15}$$

$$(\mu_{t|t}, \Sigma_{t|t}) = \begin{cases} \left(\mu_{t|t-1} + \Sigma_{t|t-1}G^T\Gamma_t^{-1}\epsilon_t,\right. \\ \left.\Sigma_{t|t-1} - \Sigma_{t|t-1}G^T\Gamma_t^{-1}G\Sigma_{t|t-1}\right), & c_t^d = 1 \\ \left(\mu_{t|t-1}, \Sigma_{t|t-1}\right), & c_t^d = 0. \end{cases} \tag{5.16}$$

where $\Gamma_t = G\Sigma_{t|t-1}G^T + V$ and $\epsilon_t = y_t - G\mu_{t|t-1}$. Also, letting $B_t = \Sigma_{t|t}F^T(F\Sigma_{t|t}F^T + W)^{-1}$, $b_t = (I_{d_x\times d_x} - B_tF)\mu_{t|t}$, and $\Sigma_{t|t+1} = (I_{d_x\times d_x} - B_tF)\Sigma_{t|t}$ we can show that the backward transition density required for the forward smoothing recursion is Gaussian as well

$$p_\theta(x_{t-1}|y_{1:t-1}, c_{1:t-1}^d, x_t) = \mathcal{N}\left(x_{t-1}; B_{t-1}x_t + b_{t-1}, \Sigma_{t-1|t}\right).$$

We define the matrix valued functions

$$\bar{S}_{m,l} : \mathcal{X}^l \times \{0,1\}^l \times \mathcal{Y}^l \to \mathbb{R}^{d_x \times d_m},$$

such that $\bar{S}_{m,l}(x_{1:l}, c^d_{1:l}, y_{1:l})$ for $m = 1, \ldots, 7$ are in the following form:

$$\sum_{t=1}^{l} c^d_t x_t x_t^T, \quad \sum_{t=1}^{l} c^d_t x_t y_t^T, \quad \sum_{t=2}^{l} x_{t-1} x_{t-1}^T, \quad \sum_{t=2}^{l} x_t x_t^T, \quad \sum_{t=2}^{l} x_{t-1} x_t^T, \quad x_1, \quad x_1 x_1^T. \tag{5.17}$$

(so, $d_2 = d_y$ and $d_6 = 1$, else $d_m = d_x$). These functions are actually the sufficient statistics in the MTT model corresponding to a single target. Then it is possible to define the incremental functions

$$\bar{s}_m : \left(\mathcal{X} \cup \mathcal{X}^2\right) \times \{0,1\} \times \mathcal{Y} \to \mathbb{R}^{d_x \times d_m} \tag{5.18}$$

where $\bar{s}_m$'s are defined such that for $m = 1, \ldots, 7$

$$\bar{S}_{m,l}(x_{1:l}, c^d_{1:l}, y_{1:l}) = \bar{s}_m(x_1, c^d_1, y_1) + \sum_{t=2}^{l} \bar{s}_m(x_{t-1}, x_t, c^d_t, y_t).$$

For example, $\bar{s}_1(x_1, c^d_1, y_1) = c^d_1 x_1 x_1^T$, $\bar{s}_3(x_1, c^d_1, y_1) = 0_{d_x \times d_x}$, $\bar{s}_3(x_{t-1}, x_t, c^d_t, y_t) = x_{t-1} x_t^T$, $\bar{s}_6(x_1, c^d_1, y_1) = c^d_1 x_1$, $\bar{s}_7(x_{t-1}, x_t, c^d_t, y_t) = 0_{d_x \times d_x}$, etc. We observe that each sufficient statistic is a matrix valued quantity, hence its expectation can be calculated using forward smoothing by treating each element of the matrix separately. For example, for

$$\bar{S}_{1,n}(x_{1:n}, c^d_{1:n}, y_{1:n}) = \sum_{t=1}^{n} c^d_t x_t x_t^T,$$

we perform forward smoothing for each

$$\bar{S}_{1,n,ij}(x_{1:n}, c^d_{1:n}, y_{1:n}) = \sum_{t=1}^{n} c^d_t x_t(i) x_t(j), \quad i, j = 1, \ldots, d_x.$$

It was shown in Elliott and Krishnamurthy [1999] that, the intermediate function

$$\bar{T}^{\theta}_{1,t,ij}(x_t, c^d_{1:t}) := \mathbb{E}_\theta \left[\bar{S}_{1,t,ij}(X_{1:t}, c^d_{1:t}, y_{1:t}) \middle| c^d_{1:t}, x_t, y_{1:t} \right]$$

for the i, j'th element is a quadratic in x_t:

$$\bar{T}^{\theta}_{1,t,ij}(x_t, c^d_{1:t}) = x_t^T \bar{P}_{1,t,ij} x_t + \bar{q}^T_{1,t,ij} x_t + \bar{r}_{1,t,ij}, \tag{5.19}$$

where $\bar{P}_{1,t,ij}$ is a $d_x \times d_x$ matrix, $\bar{q}_{1,t,ij}$ is a $d_x \times 1$ vector, and $\bar{r}_{1,t,ij}$ is a scalar. Online smoothing is then performed via the following recursion over the variables $\bar{P}_{1,t,ij}, \bar{q}_{1,t,ij}, \bar{r}_{1,t,ij}$.

$$\begin{aligned}
\bar{P}_{1,t+1,ij} &= B_t^T \bar{P}_{1,t,ij} B_t + c_{t+1}^d e_i e_j^T, \\
\bar{q}_{1,t+1,ij} &= B_t^T \bar{q}_{1,t,ij} + B_t^T \left(\bar{P}_{1,t,ij} + \bar{P}_{1,t,ij}^T\right) b_t, \\
\bar{r}_{1,t+1,ij} &= \bar{r}_{1,t,ij} + \operatorname{tr}\left(\bar{P}_{1,t,ij} \Sigma_{t|t+1}\right) + \bar{q}_{1,t,ij}^T b_t + b_t^T \bar{P}_{1,t,ij} b_t,
\end{aligned}$$

where e_i is the i'th column of the identity matrix of the size d_x, and $\operatorname{tr}(A)$ is the trace of the matrix A. For the initial value of $\bar{T}_{1,1,ij}^{\theta}(x_1, c_1^d)$, $\bar{P}_{1,1,ij} = c_1^d e_i e_j^T, q_{1,1,ij} = 0_{d_x \times 1}, \bar{r}_{1,1,ij} = 0$. Therefore, the i,j'th element of the required expectation at time n can be calculated as

$$\mathbb{E}_\theta \left[\bar{T}_{1,n,ij}^{\theta}(X_n, c_{1:n}^d) \middle| y_{1:n}, c_{1:n}^d \right] = \operatorname{tr}\left(\bar{P}_{1,n,ij}\left(\Sigma_{n|n} + \mu_{n|n}\mu_{n|n}^T\right)\right) + \bar{q}_{1,n,ij}^T \mu_{n|n} + \bar{r}_{1,n,ij}.$$

We can similarly obtain the recursions for the other sufficient statistics in terms of variables $\bar{P}_{m,t,ij}, \bar{q}_{m,t,ij}, \bar{r}_{m,t,ij}$ for the m'th sufficient statistic (see Appendix 5.A.1) [Elliott and Krishnamurthy, 1999].

Remark 5.1. *Note that $\bar{P}_{1,t,ji} = (\bar{P}_{1,t,ij})^T$ (similarly for $\bar{q}_{1,t,ij}$) and therefore need only be calculated for $j \geq i$. Note that the variables $\mu_{t|t}, \Sigma_{t|t}, \Gamma_t, \epsilon_t, B_t, b_t, \Sigma_{t|t+1}, \bar{P}_{m,t,ij}, \bar{q}_{m,t,ij}, \bar{r}_{m,t,ij}$ obviously depend on $c_{1:t}^d$, $y_{1:t}$ and θ, but we made this dependency implicit in our notation for simplicity. We will carry on with this simplification in the rest of the chapter.*

5.3.2.2 Application to MTT

We showed above how to calculate expectations of the required sufficient for a single GLSSM. We can extend that idea to the scenario in the MTT case, where there may be multiple GLSSM's at a time, with different starting and ending times and possible missing observations. Recall that at time t the targets which are alive are the k_t^s surviving targets from $t-1$ and the k_t^b newly born targets at time t, so the number of targets is $k_t^x = k_t^s + k_t^b$. For each alive target, we can calculate the moments of the prediction density $p_\theta(x_{t,k}|\mathbf{y}_{1:t-1}, z_{1:t})$ for the state

$$(\mu_{t|t-1,k}, \Sigma_{t|t-1,k}) = \begin{cases} \left(F\mu_{t-1|t-1,i_t^s(k)}, F\Sigma_{t-1|t-1,i_t^s(k)}F^T + W\right), & k \leq k_t^s, \\ (\mu_b, \Sigma_b), & k_t^s < k \leq k_t^x \end{cases}.$$

Recall that $i_t^s(k)$ appears above due to the relabelling of surviving targets from time $t-1$. Also, given the detection vector c_t^d and the association vector a_t, we calculate the moments of the filtering density $p_\theta(x_{t,k}|\mathbf{y}_{1:t}, z_{1:t})$ for the targets using the prediction

moments

$$(\mu_{t|t,k}, \Sigma_{t|t,k}) = \begin{cases} \left(\mu_{t|t-1,k} + \Sigma_{t|t-1,k} G^T \Gamma_{t,k}^{-1} \epsilon_{t,k}, \Sigma_{t|t-1,k} - \Sigma_{t|t-1,k} G^T \Gamma_{t,k}^{-1} G \Sigma_{t|t-1,k}\right), & c_t^d(k) = 1 \\ \left(\mu_{t|t-1,k}, \Sigma_{t|t-1,k}\right), & c_t^d(k) = 0. \end{cases}$$

where $\Gamma_{t,k} = G\Sigma_{t|t-1,k}G^T + V$ and $\epsilon_{t,k} = y_{t,a_t(i'_t(k))} - G\mu_{t|t-1,k}$, where $i'_t(k) = \sum_{j=1}^{k} c_t^d(j)$. Note that if the k'th alive target at time t is detected, it will be the $i'_t(k)$'th detected target, which explains $i'_t(k)$ in $\epsilon_{t,k}$. In a similar manner, we calculate $B_{t,k}$, $b_{t,k}$, and $\Sigma_{t|t+1,k}$ using $\mu_{t|t,k}$ and $\Sigma_{t|t,k}$ for $k = 1, \ldots, k_t^x$ in analogy with B_t, b_t, and $\Sigma_{t|t+1}$.

In the following, we will present the rules for one-step update of the expectations

$$\widetilde{S}_{m,n}^{\theta}(z_{1:n}) = \mathbb{E}_\theta \left[S_{m,n}(\mathbf{X}_{1:n}, z_{1:n}) \middle| \mathbf{y}_{1:n}, z_{1:n} \right]$$

of the sufficient statistics $S_{m,n}(\mathbf{x}_{1:n}, z_{1:n})$ that are defined in (5.8). Observe that we can write for $1 \leq m \leq 7$,

$$S_{m,n}(\mathbf{x}_{1:n}, z_{1:n}) = s_m(\mathbf{x}_1, z_1) + \sum_{t=2}^{n} s_m(\mathbf{x}_{t-1}, \mathbf{x}_t, z_t), \tag{5.20}$$

where the functions s_m can be written in terms of $\bar{s}_m$'s (5.18) as follows:

$$s_m(\mathbf{x}_1, z_1) = \sum_{k=1}^{k_1^b} \bar{s}_m(x_{1,k}, c_1^d(k), y_{1,a_1(i'_1(k))}),$$

$$s_m(\mathbf{x}_{t-1}, \mathbf{x}_t, z_t) = \sum_{k=1}^{k_t^s} \bar{s}_m(x_{t-1,i_t^s(k)}, x_{t,k}, c_t^d(k), y_{t,a_t(i'_t(k))}) + \sum_{k=k_t^s+1}^{k_t^x} \bar{s}_m(x_{t,k}, c_t^d(k), y_{t,a_t(i'_t(k))}).$$

where, again, $i'_t(k) = \sum_{j=1}^{k} c_t^d(j)$. (Notice that if $c_t^d(k) = 0$ this $i'_t(k)$ can still be used as a convention; since the choice of the observation point in $\mathbf{y}_t$ is irrelevant as it will have no contribution being multiplied by $c_t^d(k)$.) Therefore, the forward smoothing recursion for those sufficient statistics in (5.8) at time t

$$T_{m,t}^{\theta}(\mathbf{x}_t, z_{1:t}) = \mathbb{E}_\theta \left[T_{m,t-1}^{\theta}(\mathbf{X}_{t-1}, z_{1:t-1}) + s_m\left(\mathbf{X}_{t-1}, \mathbf{x}_t, z_t\right) | \mathbf{x}_t, \mathbf{y}_{1:t-1}, z_{1:t-1} \right] \tag{5.21}$$

can be handled once we have the forward smoothing recursion rules for the sufficient statistics in (5.17). For $k = 1, \ldots, k_t^x$, let $T_{m,t,k}^{\theta}$ denote the forward smoothing recursion function for the m'th sufficient statistic for k'th alive target at time t. For the surviving targets, k'th target at time t is a continuation of the $i_t^s(k)$'the target at time $t-1$.

Therefore, we have the recursion update for $T^{\theta}_{m,t,k}$ for $1 \leq k \leq k^s_t$ as

$$T^{\theta}_{m,t,k}(x_{t,k}, z_{1:t}) = \mathbb{E}_{\theta}\Big[T^{\theta}_{m,t-1,i^s_t(k)}(X_{t-1,i^s_t(k)}, z_{1:t-1}) + \bar{s}_m(X_{t-1,i^s_t(k)}, x_{t,k}, c^d_t(k), y_{a_t(i'_t(k))})|x_{t,k}, \mathbf{y}_{1:t-1}, z_{1:t-1}\Big].$$

For the targets born at time t (for $k^s_t+1 \leq k \leq k^x_t$), the recursion function is initiated as $T^{\theta}_{m,t,k}(x_{t,k}, z_{1:t}) = s_m(x_{t,k}, c^d_t(k))$. Therefore, the (i,j)'th component of the recursion function can be written as

$$T^{\theta}_{m,t,k,ij}(x_{t,k}, z_{1:t}) = x^T_{t,k} P_{m,t,k,ij} x_{t,k} + q_{m,t,k,ij} x_{t,k} + r_{m,t,k,ij}$$

similarly to the single GLSSM case, where this time we have the additional subscript k. For surviving targets the recursion variables $P_{m,t,k,ij}, q_{m,t,k,ij}, r_{m,t,k,ij}$ for each m, i, j are updated from $P_{m,t-1,i^s_t(k),ij}, q_{m,t-1,i^s_t(k),ij}, r_{m,t-1,i^s_t(k),ij}$, by using $\mu_{t-1|t-1,i^s_t(k)}$, $\Sigma_{t-1|t-1,i^s_t(k)}$, $B_{t-1,i^s_t(k)}$, $b_{t-1,i^s_t(k)}$, $\Sigma_{t-1|t,i^s_t(k)}$, $c^d_t(k)$ and, $y_{t,a_t(i'_t(k))}$ with $i'_t(k) = \sum_{j=1}^{k} c^d_t(j)$. For the targets born at time t (for $k^s_t + 1 \leq k \leq k^x_t$), the variables are set to their initial values in the same way as in Section 5.3.2.1 using $c^d_t(k)$ and, if $c^d_t(k) = 1$, $y_{t,a_t(i'_t(k))}$. The conditional expectations of sufficient statistics

$$\widetilde{S}^{\theta}_{m,t}(z_{1:t}) = \mathbb{E}_{\theta}\left[T^{\theta}_{m,t}\left(\mathbf{X}_t, z_{1:t}\right)\middle|\, \mathbf{y}_{1:t}, z_{1:t}\right]$$

can then be calculated by using the forward recursion variables and the filtering moments. Let

$$\widetilde{S}^{\theta}_{m,t,k}(z_{1:t}) = \mathbb{E}_{\theta}\left[T^{\theta}_{m,t,k}(X_{t,k}, z_{1:t})\middle|\, \mathbf{y}_{1:t}, z_{1:t}\right]$$

denote the expectation of the m'th sufficient statistic for the k'th alive target at time t, where its (i,j)'th component is

$$\widetilde{S}^{\theta}_{m,t,k,ij}(z_{1:t}) = \mathrm{tr}\left(P_{m,t,k,ij}\left(\mu_{t|t,k}\mu^T_{t|t,k} + \Sigma_{t|t,k}\right)\right) + q^T_{m,t,k,ij}\mu_{t|t,k} + r_{m,t,k,ij}.$$

Then, the required conditional expectation for the m'th sufficient statistic can be written as the sum of two quantities

$$\widetilde{S}^{\theta}_{m,t}(z_{1:t}) = \widetilde{S}^{\theta}_{alive,m,t}(z_{1:t}) + \widetilde{S}^{\theta}_{dead,m,t}(z_{1:t}). \tag{5.22}$$

where the quantities are respectively the contributions of the alive targets at time t and

dead targets up to time t to the conditional expectation $\widetilde{S}^{\theta}_{m,t}(z_{1:t})$

$$\widetilde{S}^{\theta}_{alive,m,t}(z_{1:t}) = \sum_{k=1}^{k_t^x} \widetilde{S}^{\theta}_{m,t,k}(z_{1:t}),$$
$$\widetilde{S}^{\theta}_{dead,m,t}(z_{1:t}) = \sum_{j=1}^{t} \sum_{k:c_j^s(k)=0}^{k_{j-1}^x} \widetilde{S}^{\theta}_{m,j-1,k}(z_{1:j-1}) \tag{5.23}$$

As (5.22) shows, we also need to calculate $\widetilde{S}^{\theta}_{dead,m,t}(z_{1:t})$ at each time and by (5.23) this can easily be done by storing $\widetilde{S}^{\theta}_{dead,m,t-1}(z_{1:t-1})$ at time $t-1$ and using the recursion

$$\widetilde{S}^{\theta}_{dead,m,t}(z_{1:t}) = \widetilde{S}^{\theta}_{dead,m,t-1}(z_{1:t-1}) + \sum_{k:c_t^s(k)=0}^{k_{t-1}^x} \widetilde{S}^{\theta}_{m,t-1,k}(z_{1:t-1})$$

where the terms in the sum correspond to targets that terminate at time $t-1$.

Finally, the sufficient statistics $S_{8,n}(z_{1:n}), \ldots, S_{15,n}(z_{1:n})$ can be calculated online since we can write for each $m = 8, \ldots, 15$

$$S_{m,n}(z_{1:n}) = \sum_{t=1}^{n} s_m(z_t)$$

for some suitable functions s_m which can easily be constructed from (5.9). Hence they can be updated online as

$$S_{m,t}(z_{1:t}) = S_{m,t-1}(z_{1:t-1}) + s_m(z_t). \tag{5.24}$$

We now present Algorithm 5.2 to show how these one-step update rules for the sufficient statistics in the MTT model can be implemented. For simplicity of the presentation, we will use a short hand notation for representing the forward recursion variables in a batch way. Let $\mathcal{T}^{\theta}_{m,t}(z_{1:t}) = (\mathcal{T}^{\theta}_{m,t,k}(z_{1:t}), k = 1, \ldots, k_t^x)$ where

$$\mathcal{T}^{\theta}_{m,t,k}(z_{1:t}) = (P_{m,t,k,ij}, q_{m,t,k,ij}, r_{m,t,k,ij} : \text{all } i, j)$$

denote all the variables required for the forward smoothing recursion for the m'th sufficient statistic for the k'th alive target at time t. We can now present the algorithm using this notation.

Algorithm 5.2. ***One step update for sufficient statistics in the MTT model*** *We have* $\mathcal{T}^{\theta}_{m,t-1}(z_{1:t-1})$, $\widetilde{S}^{\theta}_{dead,m,t-1}(z_{1:t-1})$, $m = 1, \ldots, 7$, $S^{\theta}_{m',t-1}(z_{1:t-1})$, $m' = 8, \ldots, 15$ *at time* $t-1$. *Given* z_t *and* $\mathbf{y}_t$,
- *Set* $i_x = 0$, $i_d = 0$, $\widetilde{S}^{\theta}_{alive,m,t}(z_{1:t}) = 0$ *and* $\mathcal{S}^{\theta}_{dead,m,t}(z_{1:t}) = \mathcal{S}^{\theta}_{dead,m,t-1}(z_{1:t-1})$ *for* $m =$

$1, \ldots, 7$.
- *for* $i = 1, \ldots, k^x_{t-1} + k^b_t$

- *if* $i \leq k^x_{t-1}$ *and* $c^s_t(i) = 1$, *(the i'th target at time* $t-1$ *survives), or if* $i > k^x_{t-1}$, *(a new target is born), set* $i_x = i_x + 1$.

  - *In case of survival, use* $\mu_{t-1|t-1,i}$ *and* $\Sigma_{t-1|t-1,i}$ *to obtain the prediction moments* $\mu_{t|t-1,i_x}$ *and* $\Sigma_{t|t-1,i_x}$. *In case of birth, set the prediction distribution* $\mu_{t|t-1,i_x} = \mu_b$ *and* $\Sigma_{t|t-1,i} = \Sigma_b$.

    * *If* $c^d_t(i_x) = 1$, i_x*'th target is detected:* $i_d = i_d + 1$. *Use* $\mu_{t|t-1,i_x}$ *and* $\Sigma_{t|t-1,i_x}$ *and* $y_{t,a_t(i_d)}$ *to update the filtering moments* $\mu_{t|t,i_x}$ *and* $\Sigma_{t|t,i_x}$.
    * *If* $c^d_t(i_x) = 0$, i_x*'th target is not detected: Set* $\left(\mu_{t|t,i_x}, \Sigma_{t|t,i_x}\right) = \left(\mu_{t|t-1,i_x}, \Sigma_{t|t-1,i_x}\right)$.

  - *For* $m = 1, \ldots, 7$

    * *In case of survival, update the recursion variables* $\mathcal{T}^\theta_{m,t,i_x}(z_{1:t})$ *using* $\mathcal{T}^\theta_{m,t-1,i}(z_{1:t-1})$, $\mu_{t-1|t-1,i}$, $\Sigma_{t-1|t-1,i}$, $b_{t-1,i}$, $B_{t-1,i}$, $\Sigma_{t-1|t,i}$, $c^d_t(i_x)$ *and* $y_{t,a_t(i_d)}$ *if* $c^d_t(i_x) = 1$. *In case of birth, initiate* $\mathcal{T}^\theta_{m,t,i_x}(z_{1:t})$ *using* $c^d_t(i_x)$ *and* $y_{t,a_t(i_d)}$ *if* $c^d_t(i_x) = 1$.
    * **(optional)** *Calculate* $\widetilde{S}^\theta_{m,t,i_x}(z_{1:t})$ *using* $\mathcal{T}^\theta_{m,t,i_x}(z_{1:t})$, $\mu_{t|t,i_x}$ *and* $\Sigma_{t|t,i_x}$ *and update* $\widetilde{S}^\theta_{alive,m,t}(z_{1:t}) \leftarrow \widetilde{S}^\theta_{alive,m,t}(z_{1:t}) + \widetilde{S}^\theta_{m,t,i_x}(z_{1:t})$.

- *if* $i \leq k^x_{t-1}$ *and* $c^s_t(i) = 0$, *the i'th target at time* $t-1$ *is dead. For* $m = 1, \ldots, 7$,

  - *Calculate* $\widetilde{S}^\theta_{m,t-1,i}(z_{1:t-1})$ *from* $\mathcal{T}_{m,t-1,i}(z_{1:t-1})$, $\mu_{t-1|t-1,i}$ *and* $\Sigma_{t-1|t-1,i}$.
  - *Update* $\widetilde{S}^\theta_{dead,m,t}(z_{1:t}) \leftarrow \widetilde{S}^\theta_{dead,m,t}(z_{1:t}) + \widetilde{S}^\theta_{m,t-1,i}(z_{1:t-1})$.

- **(optional)** *Update* $\widetilde{S}^\theta_{m,t}(z_{1:t}) = \widetilde{S}^\theta_{alive,m,t}(z_{1:t}) + \widetilde{S}^\theta_{dead,m,t}(z_{1:t})$ *for* $m = 1, \ldots, 7$.
- *Update* $S_{m,t}(z_{1:t}) = S_{m,t-1}(z_{1:t-1}) + s_m(z_t)$ *for* $m = 8, \ldots, 15$.

Notice that the lines of the algorithm labeled as "optional" are not necessary for the recursion and need not to be performed at every time step. For example, we can use Algorithm 5.2 in a batch EM to save memory, in that case we perform these steps only at the last time step n to obtain the required expectations. Notice also that we included the update rule for the sufficient statistics in (5.9) for completeness.

5.3.2.3 Online EM implementation

In order to develop an online EM algorithm, we exploit the availability of calculating $\widetilde{S}^\theta_{1,t}, \ldots, \widetilde{S}^\theta_{7,t}$ and $S_{8,t}, \ldots, S_{15,t}$ in an online manner as shown in Section 5.3.2.2. In online EM, running averages of sufficient statistics are calculated and then used to update the estimate of θ^* at each time [Cappé, 2009, 2011; Elliott et al., 2002; Mongillo and Deneve, 2008]. Let θ_1 be the initial guess of θ^* before having made any observations and at time t, let $\theta_{1:t}$ be the sequence of parameter estimates of the online EM algorithm computed

sequentially based on $\mathbf{y}_{1:t-1}$. When $\mathbf{y}_t$ is received, we first update the posterior density to have $\widehat{p}_{\theta_{1:t}}(z_{1:t}|\mathbf{y}_{1:t})$, and compute for $1 \leq m \leq 7$

$$T_{\gamma,m,t}^{\theta_{1:t}}(\mathbf{x}_t, z_{1:t}) = \mathbb{E}_{\theta_{1:t}}\left[(1-\gamma_t)T_{\gamma,m,t-1}^{\theta_{1:t-1}}(\mathbf{X}_{t-1}, z_{1:t-1}) + \gamma_t s_m(\mathbf{X}_{t-1}, \mathbf{x}_t, z_t)\,\middle|\,\mathbf{x}_t, \mathbf{y}_{1:t-1}, z_{1:t-1}\right] \tag{5.25}$$

for the values $z_{1:t} = z_{1:t}^{(i)}$ for $i = 1, \ldots, N$, where we have the same constraints on the step-size sequence $\{\gamma_t\}_{t\geq 1}$ as in the SAEM algorithm. This modification reflects on the updates rules for the variables in $\mathcal{T}_{m,t}^{\theta}$. To illustrate the change in the recursions with an example, the recursion rules for the variables for $S_{1,t}(x_{1:t}, c_{1:t}^d)$ for the simple GLSSM case become (see Appendix 5.A.1)

$$\begin{aligned}
\bar{P}_{\gamma,1,t+1,ij} &= (1-\gamma_{t+1})B_t^T \bar{P}_{\gamma,1,t,ij} B_t + \gamma_{t+1} c_{t+1}^d e_i e_j^T \\
\bar{q}_{\gamma,1,t+1,ij} &= (1-\gamma_{t+1})\left(B_t^T \bar{q}_{\gamma,1,t,ij} + B_t^T\left(\bar{P}_{\gamma,1,t,ij} + \bar{P}_{\gamma,1,t,ij}^T\right) b_t\right) \\
\bar{r}_{\gamma,1,t+1,ij} &= (1-\gamma_{t+1})\left(\bar{r}_{\gamma,1,t,ij} + \operatorname{tr}\left(\bar{P}_{\gamma,1,t,ij}\Sigma_{t|t+1}\right) + \bar{q}_{\gamma,1,t,ij}^T b_t + b_t^T \bar{P}_{\gamma,1,t,ij} b_t\right)
\end{aligned}$$

So this time we have $\mathcal{T}_{\gamma,m,t}^{\theta_{1:t}}(z_{1:t}) = (\mathcal{T}_{\gamma,m,t,k}^{\theta_{1:t}}(z_{1:t}), k = 1, \ldots, k_t^x)$ where

$$\mathcal{T}_{\gamma,m,t,k}^{\theta_{1:t}}(z_{1:t}) = \left(P_{\gamma,m,t,k,ij}, q_{\gamma,m,t,k,ij}, r_{\gamma,m,t,k,ij} : \text{all } i,j\right).$$

and the conditional expectations

$$\widetilde{S}_{\gamma,m,t}^{\theta_{1:t}}(z_{1:t}) = \widetilde{S}_{\gamma,alive,m,t}^{\theta_{1:t}}(z_{1:t}) + \widetilde{S}_{\gamma,dead,m,t}^{\theta_{1:t}}(z_{1:t})$$

can be calculated by using $\mathcal{T}_{\gamma,m,t,k}^{\theta_{1:t}}(z_{1:t})$ as in Section 5.3.2.2. Finally, regarding those $S_{m,t}$ in (5.9), we calculate $8 \leq m \leq 15$.

$$S_{\gamma,m,t}(z_{1:t}) = (1-\gamma_t)S_{\gamma,m,t-1}(z_{1:t-1}) + \gamma_t s_m(z_t). \tag{5.26}$$

for the values $z_{1:t} = z_{1:t}^{(i)}$ for $i = 1, \ldots, N$. In the maximisation step, we update the parameter estimate by $\theta_{t+1} = \Lambda(\widehat{S}_{\gamma,1,t}^{\theta_{1:t}}, \ldots, \widehat{S}_{\gamma,15,t}^{\theta_{1:t}})$ where the expectations are obtained

$$\widehat{S}_{\gamma,m,t}^{\theta_{1:t}} = \begin{cases} \sum_{i=1}^N w_t^{(i)} \widetilde{S}_{\gamma,m,t}^{\theta_{1:t}}(z_{1:t}^{(i)}), & 1 \leq m \leq 7, \\ \sum_{i=1}^N w_t^{(i)} S_{\gamma,m,t}(z_{1:t}^{(i)}), & 8 \leq m \leq 15. \end{cases}$$

In practice, the maximisation step is not executed until a burn-in time t_b for added stability of the estimators (e.g. see Cappé [2009]).

Notice that the SMC online EM algorithm can be implemented with the help of Algorithm 5.2 the only changes are (5.25) and (5.26) instead of (5.21) and (5.24). Algorithm 5.3 describes the SMC online EM algorithm for the MTT model.

Algorithm 5.3. ***The SMC online EM algorithm for the MTT model***

- ***E-step:*** *If* $t = 1$, *start with* θ_1, *obtain* $\widehat{p}_{\theta_1}(z_1|\mathbf{y}_1) = \sum_{i=1}^{N} w_1^{(i)} \delta_{z_1^{(i)}}(z_1)$, *and for* $i = 1, \ldots, N$ *initialise*
  $\mathcal{T}_{\gamma,m,1}^{\theta_1}(z_1^{(i)})$, $\widetilde{S}_{\gamma,dead,m,1}^{\theta_1}(z_1^{(i)})$ *for* $m = 1, \ldots, 7$ *and* $S_{\gamma,m',1}(z_1^{(i)})$ *for* $m' = 8, \ldots, 15$,
  If $t \geq 2$,
  Obtain $\widehat{p}_{\theta_{1:t}}(z_{1:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{z_{1:t}^{(i)}}(z_{1:t})$ *from* $\widehat{p}_{\theta_{1:t-1}}(z_{1:t-1}|\mathbf{y}_{1:t-1})$ *along with* π_t.
  For $i = 1, \ldots, N$, *set* $j = \pi_t(i)$. *Use Algorithm 5.2 with the stochastic approximation to obtain*
  $\mathcal{T}_{\gamma,m,t}^{\theta_{1:t}}(z_{1:t}^{(i)})$, $\widetilde{S}_{\gamma,dead,m,t}^{\theta_{1:t}}(z_{1:t}^{(i)})$ *for* $m = 1, \ldots, 7$ *and* $S_{\gamma,m',t}(z_{1:t}^{(i)})$ *for* $m' = 8, \ldots, 15$ *from* $\mathcal{T}_{\gamma,m,t-1}^{\theta_{1:t-1}}(z_{1:t-1}^{(j)})$, $\widetilde{S}_{\gamma,dead,m,t-1}^{\theta_{1:t-1}}(z_{1:t-1}^{(j)})$ *for* $m = 1, \ldots, 7$ *and* $S_{\gamma,m',t-1}(z_{1:t-1}^{(j)})$ *for* $m' = 8, \ldots, 15$.

- ***M-step:*** *If* $t < t_b$, $\theta_{t+1} = \theta_t$. *Else, for* $i = 1, \ldots, N$, $m = 1, \ldots, 7$ *calculate* $\widetilde{S}_{\gamma,alive,m,t}^{\theta_{1:t}}(z_{1:t}^{(i)})$ *and* $\widetilde{S}_{\gamma,m,t}^{\theta_{1:t}}(z_{1:t}^{(i)}) = \widetilde{S}_{\gamma,alive,m,t}^{\theta_{1:t}}(z_{1:t}^{(i)}) + \widetilde{S}_{\gamma,dead,m,t}^{\theta_{1:t}}(z_{1:t}^{(i)})$ (*'**optional**' lines in Algorithm 5.2). Calculate the expectations*

$$\left[\widehat{S}_{\gamma,1,t}^{\theta_{1:t}}, \ldots, \widehat{S}_{\gamma,15,t}^{\theta_{1:t}}\right] = \sum_{i=1}^{N} w_n^{(i)} \left[\widetilde{S}_{\gamma,m,t}^{\theta}, \ldots, \widetilde{S}_{\gamma,7,t}^{\theta_{1:t}}, S_{\gamma,8,t}, \ldots, S_{\gamma,15,t}\right] \left(z_{1:t}^{(i)}\right).$$

  and update $\theta_{t+1} = \Lambda\left(\widehat{S}_{\gamma,1,t}^{\theta_{1:t}}, \ldots, \widehat{S}_{\gamma,15,t}^{\theta_{1:t}}\right)$.

Finally, before ending this section, we list in Table 5.1 some important variables used to describe the EM algorithms throughout the section.

5.4 Experiments and results

We observe the performances of the parameter estimation methods described in Section 5.3 using the constant velocity model in Example 5.1, where the parameter vector is

$$\theta = \left(\lambda_b, \lambda_f, p_d, p_s, \mu_{bp}, \mu_{bv}, \sigma_{bp}^2, \sigma_{bv}^2, \sigma_{xp}^2, \sigma_{xv}^2, \sigma_y^2\right).$$

Note that the constant velocity model assumes the position noise variance $\sigma_{xp}^2 = 0$. All other parameters are estimated.

5.4.1 Batch setting

We run two experiments using the model in the batch setting. In the first experiment, we generate an observation sequence of length $n = 100$ by using the parameter value

$$\theta^* = (0.2, 10, 0.90, 0.95, 0, 0, 25, 4, 0, 0.0625, 4)$$

Table 5.1: The list of the EM variables used in Section 5.3

Sections 5.3.1 and 5.3.1.1
$S_{m,n}$, $m = 1:15$, Sufficient statistics of the MTT model
$S^{\theta}_{m,n}$, $m = 1:15$, Expectation of $S_{m,n}$ conditional to $\mathbf{y}_{1:n}$
$\widetilde{S}^{\theta}_{m,n}$, $m = 1:7$, Expectation of $S_{m,n}$ conditional to $\mathbf{y}_{1:n}$ and $z_{1:n}$
Section 5.3.1.2
$\widehat{S}^{\theta}_{m,n}$, Monte Carlo estimation of $S^{\theta}_{m,n}$
$\widehat{S}^{(j)}_{\gamma,m,n}$, Weighted average of $\widehat{S}^{\theta_1}_{m,n}, \ldots, \widehat{S}^{\theta_j}_{m,n}$ for the SAEM algorithm
Section 5.3.2.1
$\bar{S}_{m,n}$, $m = 1:7$, Sufficient statistics of a single GLSSM
$\bar{s}_{m,t}$, $m = 1:7$, Incremental functions for $\bar{S}_{m,n}$
$\bar{S}_{m,n,ij}$, The (i,j)'th element of $\bar{S}_{m,n}$
$\bar{s}_{m,t,ij}$, The (i,j)'th element of $\bar{s}_{m,t}$
$\bar{T}_{m,t,ij}$, Forward smoothing recursion (FSR) function for $\bar{S}_{m,t,ij}$
$\bar{P}_{m,t,ij}, \bar{q}_{m,t,ij}, \bar{r}_{m,t,ij}$, Variables used to write $\bar{T}_{m,t,ij}$ in closed-form
Section 5.3.2.2
$s_{m,t}$, $m = 1:15$, Incremental functions for $S_{m,n}$
$T^{\theta}_{m,t}$, $m = 1:7$, FSR function for $S_{m,t}$
$T^{\theta}_{m,t,k}$, FSR function for m'th sufficient statistic of the k'th alive target at time t
$T^{\theta}_{m,t,k,ij}$, The (i,j)th element of $T^{\theta}_{m,t,k}$
$P_{m,t,k,ij}, q_{m,t,k,ij}, r_{m,t,k,ij}$, Variables to write $T_{m,t,k,ij}$
$\widetilde{S}^{\theta}_{m,t,k}$ Expectation of the m'th sufficient statistic of the k'th alive target at time t
$\widetilde{S}^{\theta}_{m,t,k,ij}$, The (i,j)'th element of $\widetilde{S}^{\theta}_{m,t,k}$
$\widetilde{S}^{\theta}_{alive,m,t}$, Contributions of the alive targets at time t to $\widetilde{S}^{\theta}_{m,t}$
$\widetilde{S}^{\theta}_{dead,m,t}$, Contributions of the dead targets up to time t to $\widetilde{S}^{\theta}_{m,t}$
Section 5.3.2.3
$T^{\theta_{1:t}}_{\gamma,m,t}$, Online estimation of $T^{\theta}_{m,t}$ using $\theta_{1:t}$
$P_{\gamma,m,t,k,ij}, q_{\gamma,m,t,k,ij}, r_{\gamma,m,t,k,ij}$: Variables to write $T_{\gamma,m,t,k,ij}$
$\widetilde{S}^{\theta_{1:t}}_{\gamma,alive,m,t}$, Online estimation of $\widetilde{S}^{\theta}_{alive,m,t}$ using $\theta_{1:t}$
$\widetilde{S}^{\theta_{1:t}}_{\gamma,dead,m,t}$, Online estimation of $\widetilde{S}^{\theta}_{dead,m,t}$ using $\theta_{1:t}$
$\widetilde{S}^{\theta_{1:t}}_{\gamma,m,t}$, Online estimation of $\widetilde{S}^{\theta}_{m,t}$ using $\theta_{1:t}$
$S_{\gamma,m,t}$, $m = 8:15$, Online calculation of $S_{m,n}$ using $\theta_{1:t}$
$\widehat{S}^{\theta_{1:t}}_{\gamma,m,t}$, Online estimation of $\widehat{S}^{\theta}_{m,t}$ using $\theta_{1:t}$

and window size $\kappa = 100$. This particular value of θ^* creates 1 target every 5 time steps on average, and the average life of a target is 20; therefore we expect to see around 4 targets per time.

Using the generated data set, we implemented SAEM in Algorithm 5.1) using SMC-EM for batch MLE estimation. We used $N = 200$ particles to implement the SMC method based on the L-best linear assignment to sample associations, where we set $L = 10$, the details of the SMC method are in Appendix 5.A.2. Regarding the step-size sequence in the SAEM algorithm, $\gamma_j = j^{-0.8}$ is used as the sequence of step-sizes for all parameters to be estimated, with the exception that $\gamma_j = j^{-0.55}$ is used for estimating σ_{xv}^2. That is to say, in the SAEM algorithm, $\widehat{S}_{\gamma,3,n}^{(j)}$, $\widehat{S}_{\gamma,4,n}^{(j)}$, and $\widehat{S}_{\gamma,5,n}^{(j)}$ are calculated using $\gamma_j = j^{-0.55}$, and $\widehat{S}_{\gamma,11,n}^{(j)}$ is calculated twice by using $\gamma_j = j^{-0.55}$ and $\gamma_j = j^{-0.80}$ separately (since it appears both in the estimation of σ_{xv}^2 and p_s), and for the rest of $\widehat{S}_{\gamma,m,n}^{(j)}$ $\gamma_j = j^{-0.80}$ is used.

Figure 5.2 shows the results obtained using SMC-EM after 2000. For comparison, we also execute the EM algorithm with the true data association and the resulting θ^* estimate will serve as the benchmark. Note that given the true association, the EM can be executed without the need for any Monte Carlo approximation, and it gave the estimate

$$\theta^{*,z} = (0.18, 9.94, 0.92, 0.97, -1.98, 0.91, 17.18, 5.92, 0, 0.027, 4.01).$$

The z in the superscript is to indicate that this value of θ maximises the joint probability density of $\mathbf{y}_{1:n}$ and $z_{1:n}$, i.e.

$$\theta^{*,z} = \arg\max_{\theta\in\Theta} \log p_\theta(\mathbf{y}_{1:n}, z_{1:n})$$

which is different than θ_{ML}. However, for a data size of 100, $\theta^{*,z}$ is expected to be closer to θ_{ML} than θ^* is, hence it is useful for evaluating the performances of the stochastic EM algorithms we present.

From Figure 5.2, we can see that almost all MLEs obtained using SMC-EM converge to certain values around $\theta^{*,z}$; except that σ_{xv}^2 has not converged within reasonable running time and it is worth investigating the reason behind this slow convergence. Our hypothesis is that the slow convergence of σ_{xv}^2 in SMC-EM is due to the fact that the algorithm spends most of its time to update the estimates of sufficient statistics by running the SMC algorithm for MTT going through all the observations. This may not be efficient in the sense that the algorithm uses too many samples (i.e. too much computation time) for estimating sufficient statistics for a fixed θ_j while θ_j is varying slowly over iterations.

Actually, we can speed up the estimation process by applying online SMC-EM on a

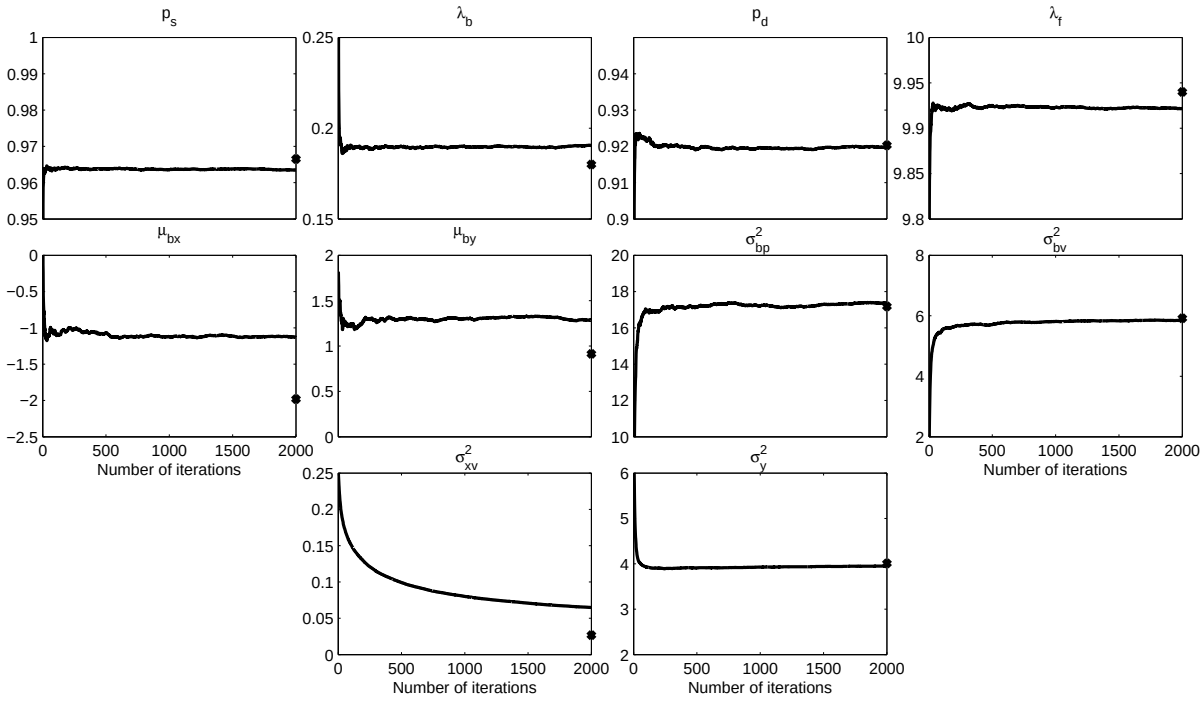


Figure 5.2: Batch estimates obtained using the SMC-EM algorithm for MLE. $\theta^{*,z}$ is shown as a cross.

sequence of repeated data. That is, we can concatenate the data as

$$[\mathbf{y}_{1:n}, \mathbf{y}_{1:n}, \ldots],$$

and run SMC online EM in Algorithm 5.3 for the concatenated data. Figure 5.3 shows both our previous SMC-EM estimates (vs number of iterations) in Figure 5.2 and the SMC online EM estimates (vs number of passes over the original data $\mathbf{y}_{1:n}$) on the concatenated data; and we note that that both algorithms are started with the same initial estimate of θ^*. Noting that the computational cost of one iteration of the SMC-EM algorithm and the computational cost of one pass of SMC online EM algorithm over the data are roughly the same, we observe that σ_{xv}^2 does indeed converge much quicker in this way. Actually; not only for σ_{xv}^2 but also for almost all parameters in θ SMC online EM on the concatenated data forgets its initial values and settle around the values to which it converges in a much quicker fashion. However, we cannot fully trust the results of SMC online EM algorithm on the repeated data, since the discontinuity introduced by making $\mathbf{y}_1$ follow $\mathbf{y}_n$ in the concatenated data will induce bias in the estimates, see Figure 5.3. As also suggested by the figure, we expect that this discontinuity will effect especially those parameters governing the birth-death and detection-clutter dynamics of the model, i.e. $p_s, \lambda_b, p_d, \lambda_f$, however it will have little effect on the parameters $\mu_{bx}, \mu_{by}, \sigma_{bp}^2, \sigma_{bv}^2, \sigma_{xv}^2, \sigma_y^2$ which govern the dynamics of the HMM associated to a target. In conclusion, a recommended way

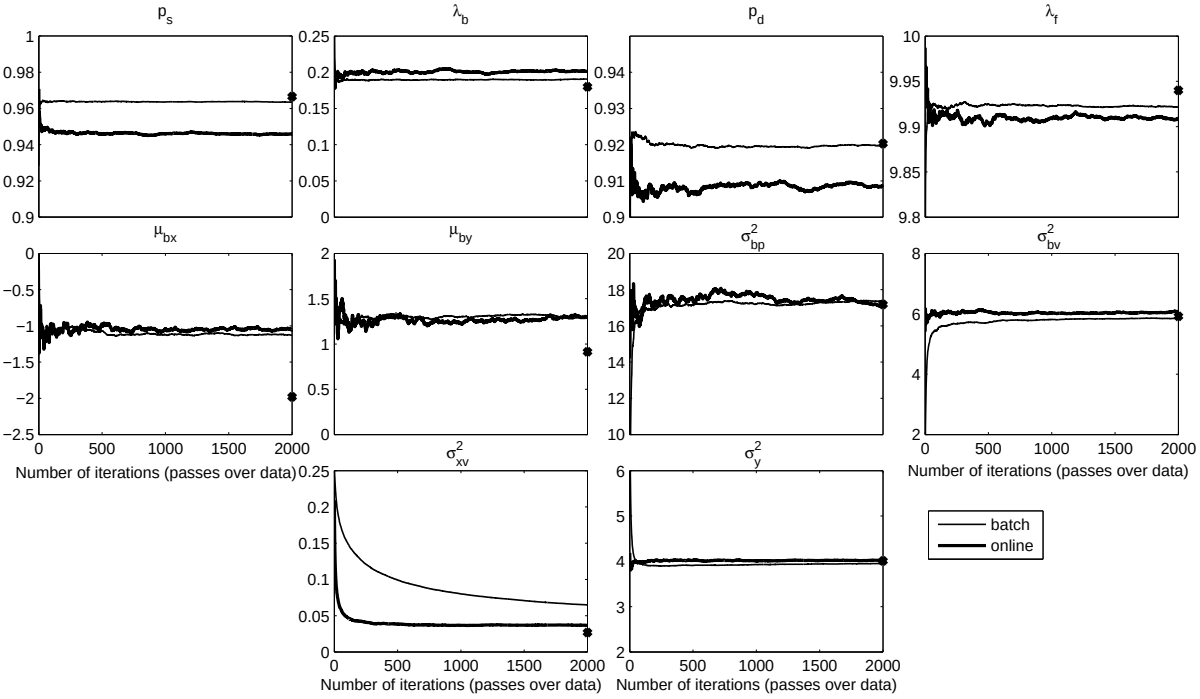


Figure 5.3: Comparison of online SMC-EM estimates applied to the concatenated data (thicker line) with batch SMC-EM.

to estimate θ^* in a batch setting is first running SMC online EM on $[\mathbf{y}_{1:n}, \mathbf{y}_{1:n}, \ldots]$ until convergence to get an estimator θ' of θ^*, and then running the batch SMC-EM initialised by θ'.

5.4.2 Online EM setting

We demonstrate the performance of the SMC online EM in Algorithm 5.3 in two settings.

5.4.2.1 Unknown fixed number of targets

In the first experiment for online estimation, we create a scenario where there are a constant but unknown number of targets that never die and travel in the surveillance region for a long time. That is, $K_0^x = K$ (which is unknown and to be estimated) and $\lambda_b = 0$ and $p_s = 1$. We also slightly modify our MTT model so that the target state is a stationary process. The modified model assumes that the state transition matrix F is

$$F = \begin{pmatrix} 0.99 I_{2\times 2} & \Delta I_{2\times 2} \\ \mathbf{0}_{2\times 2} & 0.99 I_{2\times 2} \end{pmatrix}, \tag{5.27}$$

and G, W and V are the same as the MTT model in Example 5.1. The change is to the diagonals of matrix F which should be $I_{2\times 2}$ for a constant velocity model. However,

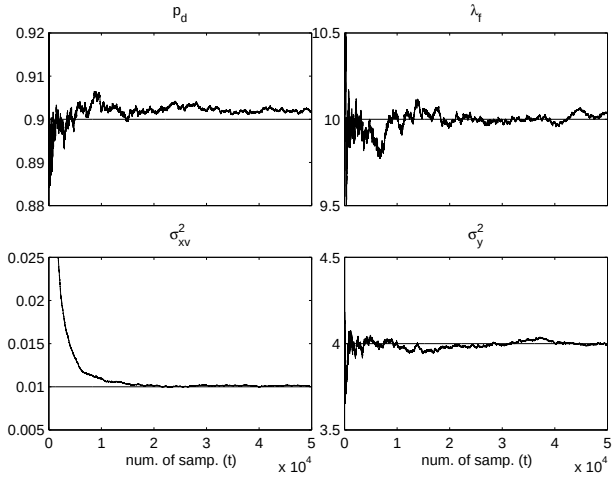


Figure 5.4: Online estimates of SMC-EM algorithm (Algorithm 5.3) for fixed number of targets. True values are indicated with a horizontal line. Initial estimates for $p_d, \lambda_f, \sigma_{xv}^2, \sigma_y^2$ are $0.6, 15, 0.25, 25$; they are not shown in order to zoom in around the converged values.

0.99 $I_{2\times 2}$ will lead to non-divergent targets, i.e. having a stationary distribution.

We create data of length $n = 50000$ with $K = 10$ targets which are initiated by using $\mu_{bx} = 0, \mu_{by} = 0, \sigma_{bx}^2 = 25, \sigma_{bv}^2 = 4$. The other parameters to create the data are $p_d = 0.9, \lambda_f = 10, \sigma_{xv}^2 = 0.01, \sigma_y^2 = 4$, and the window size $\kappa = 100$.

Figure 5.4 shows the estimates for parameters $p_d, \lambda_f, \sigma_{xv}^2, \sigma_y^2$ using the SMC online EM algorithm described in Algorithm 5.3, when $K_t^0 = K = 10$ is known. We used $L = 10$ and $N = 100$, and $\gamma_t = t^{-0.8}$ is taken for all of the parameters except σ_{xv}^2, where we used $\gamma_t = t^{-0.55}$. The burn-in time, until when the M-step is not executed, is $t_b = 10$. We can observe the estimates for the parameters quickly settle around the true values. Note that $\mu_x, \mu_y, \sigma_{bp}^2, \sigma_{bv}^2$ are not estimated here because they are the parameters of the initial distribution of targets which have no effect on the stationary distribution of a MTT model with fixed number of targets, and thus they are not identifiable by an online EM algorithm [Douc et al., 2004]. In practice, these parameters can be estimated by running a batch EM algorithm for the sequence of the first few observations, such as $\mathbf{y}_{1:50}$, fixing all other parameters to the values obtained by SMC online EM. This approximate MLE procedure is based on the fact that the parameters of the initial distribution will have will have negligible effect on the likelihood of observations $\mathbf{y}_t$ for large t.

The particle filter in Algorithm 5.3, which we used to produce the results in Figure 5.3, has all its particles having the same number of targets, which is the true K. However, K can be estimated by running several SMC online EM algorithms with different possible K's, and comparing the estimated likelihoods $p_{\theta_{1:t}}(\mathbf{y}_{1:t}|K)$ versus t. Figure 5.5 shows how the estimates of $p_{\theta_{1:t}}(\mathbf{y}_{1:t}|K)$ for values $K = 6, \ldots, 15$ compare with time. Both the left and right figures suggest that $p_{\theta_{1:t}}(\mathbf{y}_{1:t}|K)$ favours $K = 10$ starting from $t = 100$ and the

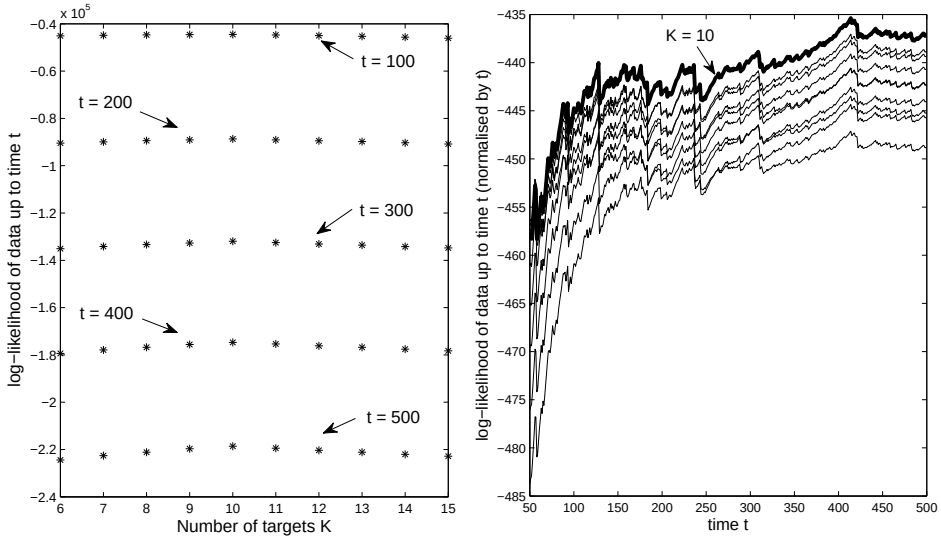


Figure 5.5: Left: estimates of $p_{\theta_{1:t}}(\mathbf{y}_{1:t}|K)$ (normalised by t) for values $t = 100 \ldots, t = 500$ and for $K = 6, \ldots, K = 15$. Right: Estimates of $p_{\theta_{1:t}}(\mathbf{y}_{1:t}|K)$ normalised by t for values $K = 6, \ldots, K = 15$, $K = 10$ is stressed with a bold plot.

decision on the number of targets can be safely made after about 200 time steps. We have also checked this comparison with different initial values for θ and found out that the comparison is robust to the initial estimate θ_0.

5.4.2.2 Unknown time varying number of targets

In the second experiment with online estimation, we consider the constant velocity model in Example 5.1 with a time-varying number of targets, i.e. $\lambda_b > 0$ and $p_s < 1$. We generated a set of data of length $n = 10^5$ using parameters

$$\theta^* = (0.2, 10, 0.90, 0.95, 0, 0, 25, 4, 0, 0.0625, 4)$$

and we estimated all of them (except $\sigma_{xp}^2 = 0$). Again, we used $L = 10$ and $N = 200$, and $\gamma_t = t^{-0.8}$ is taken for all of the parameters except σ_{xv}^2 for which we used $\gamma_t = t^{-0.55}$. The online estimates for those parameters are given in Figure 5.6 (solid lines). The initial values are taken to be $\theta_0 = (0.8, 0.5, 0.6, 13, -1, -1, 1, 1, 16, 0, 0.25, 25)$ which is not shown in the figure in order to zoom in around θ^*. We observe that the estimates have quickly left their initial values and settle around θ^*. Also, the parameter estimates for the initial distribution of newborn targets have the largest oscillations around their true values which is in agreement with the results in the batch setting.

Another important observation inferred from Figure 5.6 is the bias in the estimates of some of the parameters. This bias arises from the Monte Carlo approximation. To provide a clearer illustration of this Monte Carlo bias, we compared the SMC online EM estimates with the online EM estimates we would have if we were given the true data

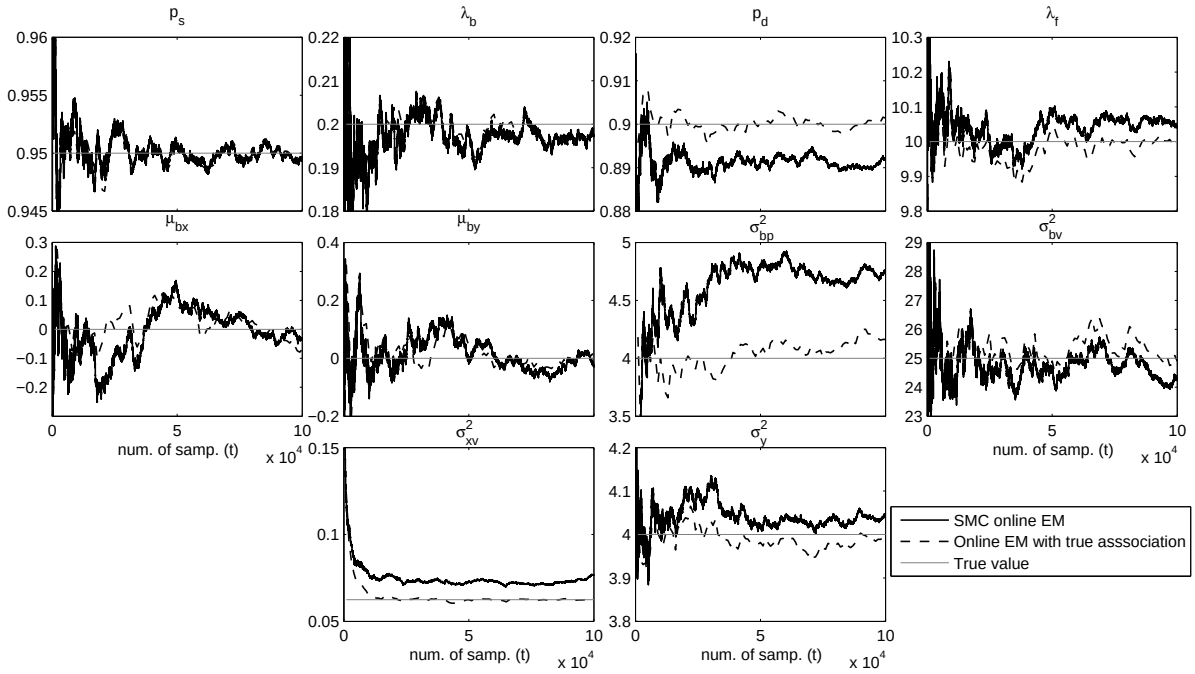


Figure 5.6: Estimates of online SMC-EM algorithm (Algorithm 5.3) for an MTT model with time varying number of targets, compared with online EM estimates when the true data association $\{Z_t\}_{t\geq 1}$ is known. For the estimates in case of known true association, $\theta_{1000,2000,\ldots,100000}$ are shown only. True values are indicated with a horizontal line.

association, i.e. $\{Z_t\}_{t\geq 1}$. The dashed lines in Figure 5.6 show the results obtained when the true association is known; for illustrative purposes we plot every 1000'th estimate only, hence the sequence $\theta_{1000,2000,\ldots,100000}$. It is interesting to observe from the plots that the trends of estimates over time are similar for most of the parameters; however for some of the parameters of (namely $p_d, \lambda_f, \sigma^2_{bv}, \sigma^2_{xv}, \sigma^2_y$) online SMC-EM have a bias.

In search of the source of the bias in our results, we ran the SMC online EM algorithm for the same data sequence, but this time by feeding the algorithm with the birth-death information, i.e. $\{K^b_t, C^s_t\}_{t\geq 0}$. Figure 5.7 shows that when $\{K^b_t, C^s_t\}_{t\geq 0}$ is provided to the algorithm, the bias will disappear. This indicates two things at the same time: (i) the bias is due to the poor tracking of birth time and death time of our SMC tracking algorithm for MTT; and (ii) the L-best approach for tracking the target-to-observation assignment, that is $\{C^d_t, K^f_t, A_t\}_{t\geq 1}$, is doing fine. Therefore, the bottle neck of the SMC tracking algorithm is birth-death tracking and, generally speaking, a better SMC scheme for the birth-death tracking may reduce the bias. At this point we would like to note that when the number of births per time is limited by a finite integer, *all* the variables of Z_t i.e. $\{K^b_t, K^f_t, C^s_t, C^d_t, A_t\}$ can be tracked within the L-best assignment framework, and we expect in this case the bias to be significantly smaller. However, since in our MTT model the number of births per time is unlimited (being a Poisson random variable), we cannot include birth-death tracking in the L-best assignment framework. For our discussion

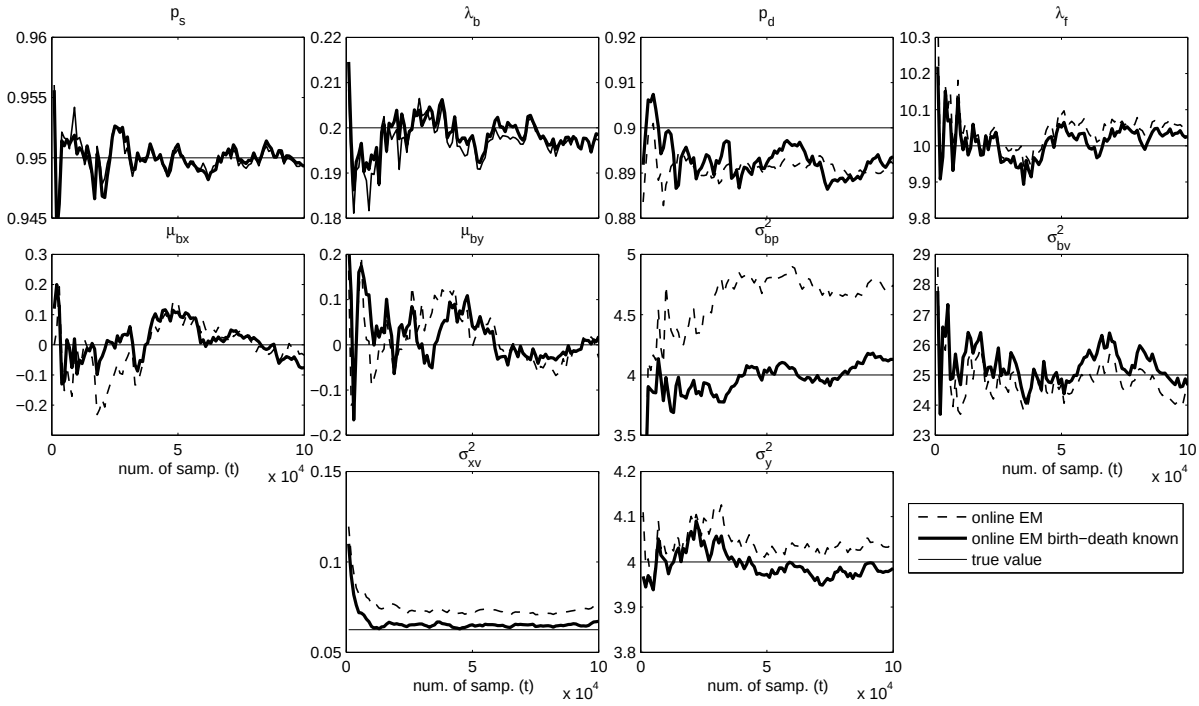


Figure 5.7: SMC online EM estimates when birth-death known (solid line) compared to the original results in Figure 5.6 (dashed lines). For illustrative purposes, every 1000th estimate is shown

on this issue to become clearer, we recommend the reader to see the SMC algorithm in Appendix 5.A.2.

5.5 Conclusion and Discussion

We have presented MLE algorithms for inferring the static parameters in the linear Gaussian MTT model. Based on our experiments on the offline and online EM implementations, our recommendations to the practitioner are: if batch estimation is permissible for the application then it should always be preferred. Moreover, online SMC-EM on concatenated data should be used to provide a good initial estimate for the batch SMC-EM. For very long data sets (in terms of time) and when there is a computational budget, online SMC-EM seems the most appropriate since the it is easier to control computational demands by restricting the number of particles. We have seen that there will be bias in some of the parameter estimates caused by the failure to track the birth-death dynamics accurately. We have not considered other tracking algorithms that work well in such scenarios such as those based on the PHD filter [Vo and Ma, 2006; Whiteley et al., 2010] which could be used provided track estimates can be extracted.

The linear Gaussian MTT model can be extended while still retaining EM based MLE. For example, split-merge scenarios for targets can be considered. Moreover, the number of

newborn targets per time need not be a Poisson random variable; for example the model may allow no births or at most one birth at a time determined by a Bernoulli random variable. Furthermore, false measurements need not be uniform, e.g. their distribution may be a Gaussian (or a Gaussian mixture) distribution. Also, we assumed that targets are born close to the centre of the surveillance region; however, different types of initiation for targets may be preferable in some applications.

For non-linear non-Gaussian MTT models, Monte Carlo type batch and online EM algorithms may still be applied by sampling from the hidden states $\mathbf{X}_t$'s provided that the sufficient statistics for the EM are available in the required additive form [Del Moral et al., 2009]. In those MTT models where sufficient statistics for EM are not available, other methods such as gradient based MLE methods can be useful (e.g. Poyiadjis et al. [2011]).

5.A Appendix

5.A.1 Recursive updates for sufficient statistics in a single GLSSM

Referring to the variables in Section 5.3.2.1, the intermediate functions for the sufficient statistics in (5.17) can be written as

$$T_{m,t,ij}(x_t, c^d_{1:t}) = x_t^T \bar{P}_{m,t,ij} x_t + \bar{q}^T_{m,t,ij} x_t + \bar{r}_{m,t,ij}$$

where $i, j = 1, \ldots, d_x$ for $m = 1, 3, 4, 5, 7$; $i = 1, \ldots, d_x, j = 1, \ldots, d_y$ for $m = 2$; and $i = 1, \ldots, d_x$, $j = 1$ for $m = 6$. All $\bar{P}_{m,t,ij}$'s, $\bar{q}_{m,t,ij}$'s and $\bar{r}_{m,t,ij}$'s are $d_x \times d_x$ matrices, $d_x \times 1$ vectors and scalars, respectively. Forward smoothing is then performed via recursions over these variables. Start at time 1 with the initial conditions $\bar{P}_{m,1,ij} = 0_{d_x \times d_x}$, $\bar{q}_{m,1,ij} = 0_{d_x \times 1}$, and $\bar{r}_{m,1,ij} = 0$ for all m except $\bar{P}_{1,1,ij} = c^d_1 e_i e_j^T$, $\bar{P}_{7,1,ij} = e_i e_j^T$, $\bar{q}_{2,1,ij} = c^d_1 y_1(j) e_i$, and $\bar{q}_{6,1,i1} = e_i$. At time $t+1$, update

$$\begin{aligned}
\bar{P}_{1,t+1,ij} &= B_t^T \bar{P}_{1,t,ij} B_t + c^d_{t+1} e_i e_j^T \\
\bar{q}_{1,t+1,ij} &= B_t^T \bar{q}_{1,t,ij} + B_t^T \left(\bar{P}_{1,t,ij} + \bar{P}^{\theta,T}_{1,t,ij} \right) b_t \\
\bar{r}_{1,t+1,ij} &= \bar{r}_{1,t,ij} + \operatorname{tr}\left(\bar{P}_{1,t,ij} \Sigma_{t|t+1} \right) + \bar{q}^T_{1,t,ij} b_t + b_t^T \bar{P}_{1,t,ij} b_t \\
\bar{P}_{2,t+1,ij} &= 0_{d_x \times d_x} \\
\bar{q}_{2,t+1,ij} &= B_t^T \bar{q}_{2,t,ij} + c^d_{t+1} y_{t+1}(j) e_i \\
\bar{r}_{2,t+1,ij} &= \bar{r}_{2,t,ij} + \bar{q}^T_{2,t+1,ij} b_t
\end{aligned}$$

$$
\begin{aligned}
\bar{P}_{3,t+1,ij} &= B_t^T \left(\bar{P}_{3,t,ij} + e_i e_j^T\right) B_t \\
\bar{q}_{3,t+1,ij} &= B_t^T \bar{q}_{3,t,ij} + B_t^T \left(\bar{P}_{3,t,ij} + \bar{P}_{3,t,ij}^T + e_i e_j^T + e_j e_i^T\right) b_t \\
\bar{r}_{3,t+1,ij} &= \bar{r}_{3,t,ij} + \operatorname{tr}\left(\left(\bar{P}_{3,t,ij} + e_i e_j^T\right) \Sigma_{t|t+1}\right) + \bar{q}_{3,t,ij}^T b_t \\
&\quad + b_t^T \left(\bar{P}_{3,t,ij} + e_i e_j^T\right) b_t \\
\bar{P}_{4,t+1,ij} &= B_t^T \bar{P}_{4,t,ij} B_t + e_i e_j^T \\
\bar{q}_{4,t+1,ij} &= B_t^T \bar{q}_{4,t,ij} + B_t^T \left(\bar{P}_{4,t,ij} + \bar{P}_{4,t,ij}^T\right) b_t \\
\bar{r}_{4,t+1,ij} &= \bar{r}_{4,t,ij} + \operatorname{tr}\left(\bar{P}_{4,t,ij} \Sigma_{t|t+1}\right) + \bar{q}_{4,t,ij}^T b_t + b_t^T \bar{P}_{4,t,ij} b_t \\
\bar{P}_{5,t+1,ij} &= B_t^T \bar{P}_{5,t,ij} B_t + e_i e_j^T B_t \\
\bar{q}_{5,t+1,ij} &= B_t^T \bar{q}_{5,t,ij} + B_t^T \left(\bar{P}_{5,t,ij} + \bar{P}_{5,t,ij}^T\right) b_t + e_j b_k^T e_i \\
\bar{r}_{5,t+1,ij} &= \bar{r}_{5,t,ij} + \operatorname{tr}\left(\bar{P}_{5,t,ij} \Sigma_{t|t+1}\right) + \bar{q}_{5,t,ij}^T b_t + b_t^T \bar{P}_{5,t,ij} b_t \\
\bar{P}_{6,t+1,i1} &= 0_{d_x \times d_x} \\
\bar{q}_{6,t+1,i1} &= B_t^T \bar{q}_{6,t,i1} \\
\bar{r}_{6,t+1,i1} &= \bar{r}_{6,t,i1} + \bar{q}_{6,t+1,i1}^T b_t \\
\bar{P}_{7,t+1,ij} &= B_t^T \left(\bar{P}_{7,t,ij}\right) B_t \\
\bar{q}_{7,t+1,ij} &= B_t^T \bar{q}_{7,t,ij} + B_t^T \left(\bar{P}_{7,t,ij} + \bar{P}_{7,t,ij}^T\right) b_t \\
\bar{r}_{7,t+1,ij} &= \bar{r}_{7,t,ij} + \operatorname{tr}\left(\bar{P}_{7,t,ij} \Sigma_{t|t+1}\right) + \bar{q}_{7,t,ij}^T b_t + b_t^T \bar{P}_{7,t,ij} b_t
\end{aligned}
$$

For the online EM algorithm, we simply modify the update rules by multiplying the terms on the right hand side containing e_t or $I_{d_x \times d_x}$ by γ_{t+1} and multiplying the rest of the terms by $(1 - \gamma_{t+1})$.

5.A.2 SMC algorithm for MTT

An SMC algorithm is mainly characterised by its proposal distribution. Hence, in this section we present the proposal distribution $q_\theta(z_t | z_{1:t-1}, \mathbf{y}_{1:t})$, where we exclude the superscripts for particle numbers from the notation for simplicity. Assume that $z_{1:t-1}$ is the ancestor of the particle of interest with weight w_{t-1}. We sample $z_t = \left(k_t^b, c_t^s, c_t^d, k_t^f, a_t\right)$ and calculate its weight by performing the following steps:

- *Birth-death move:* Sample $k_t^b \sim \mathcal{PO}(\cdot; \lambda_b)$ and $c_t^s(j) \sim \mathcal{BE}(\cdot; p_s)$ for $j = 1, \ldots, k_{t-1}^x$. Set $k_t^s = \sum_{j=1}^{k_{t-1}^x} c_t^s$ and construct the $k_t^s \times 1$ vector i_t^s from c_t^s. Set $k_t^x = k_t^s + k_t^b$ and calculate the prediction moments for the state. For $j = 1, \ldots, k_t^x$,

  - if $j \leq k_t^s$, set $\mu_{t|t-1,j} = F \mu_{t-1|t-1,i_t^s(j)}$ and $\Sigma_{t|t-1,j} = F \Sigma_{t-1|t-1,i_t^s(j)} F^T + W$.
  - if $j > k_t^s$, set $\mu_{t|t-1,j} = \mu_b$ and $\Sigma_{t|t-1,j} = \Sigma_b$.

  Also, calculate the moments of the conditional observation likelihood: For $j = 1, \ldots, k_t^x$, $\mu_{t,j}^y = G \mu_{t|t-1,j}$ and $\Sigma_{t,j}^y = G \Sigma_{t|t-1,j} G^T + V$.

- *Detection and association* Define the $k_t^x \times (k_t^y + k_t^x)$ matrix D_t as

$$D_t(i,j) = \begin{cases} \log(p_d \mathcal{N}(y_{t,i}; \mu_{t,j}^y, \Sigma_{t,j}^y)) & \text{if } j \leq k_t^y, \\ \log \frac{(1-p_d)\lambda_f}{|\mathcal{Y}|} & \text{if } i = j - k_t^y, \\ -\infty & \text{otherwise.} \end{cases}$$

and an assignment is a *one-to-one* mapping $\alpha_t : \{1, \ldots, k_t^x\} \to \{1, \ldots, k_t^y + k_t^x\}$. The cost of the assignment, up to an identical additive constant for each α_t is

$$d(D_t, \alpha_t) = \sum_{j=1}^{k_t^d} D_t(j, \alpha_t(j)).$$

Find the set $\mathcal{A}_L = \{\alpha_{t,1}, \ldots, \alpha_{t,L}\}$ of L assignments producing the highest assignment scores. The set $\mathcal{A}_L$ can be found using the Murty's assignment ranking algorithm [Murty, 1968] with a computational cost of $\mathcal{O}((k_t^x + k_t^y)^3 L)$ in the worst case. Finally, sample $\alpha_t = \alpha_{t,j}$ with probability

$$\kappa(\alpha_{t,j}) = \frac{\exp(d(D_t, \alpha_{t,j}))}{\sum_{j'=1}^{L} \exp(d(D_t, \alpha_{t,j'}))}, \quad j = 1, \ldots, L$$

Given α_t, one can infer c_t^d (hence i_t^d), k_t^d, k_t^f and the association a_t as follows:

$$c_t^d(k) = \begin{cases} 1 & \text{if } \alpha_t(k) \leq k_t^y, \\ 0 & \text{if } \alpha_t(k) > k_t^y. \end{cases}$$

Then $k_t^d = \sum_{j=1}^{k_t^x} c_t^d(k)$, $k_t^f = k_t^y - k_t^d$, i_t^d is constructed from c_t^d, and finally

$$a_t(k) = \alpha_t(i_t^d(k)), \quad k = 1, \ldots, k_t^d.$$

- *Reweighting:* After we sample $z_t = \left(k_t^b, c_t^s, c_t^d, k_t^f, a_t\right)$ from $q_\theta(z_t | z_{1:t-1}, \mathbf{y}_t)$, we calculate the weight of the particle as in (5.14), which becomes for this sampling scheme as

$$w_t \propto w_{t-1} \frac{e^{-\lambda_f}}{N!} \left(\frac{\lambda_f}{|\mathcal{Y}|}\right)^{k_t^y - k_t^x} \sum_{j=1}^{L} \exp(d(D_t, \alpha_{t,j})).$$

5.A.3 Computational complexity of EM algorithms

5.A.3.1 Computational complexity of SMC filtering

For simplicity, assume the true parameter value is θ. The computational cost of SMC filtering with θ and N particles, at time t, is

$$C_{\text{SMC}}(\theta, t, N) = \underbrace{c_1 N}_{\text{resampling}} + \sum_{i=1}^{N} \left[\underbrace{\left(c_2 K_{t-1}^{x(i)} + c_3 \right)}_{\text{birth-death sampling}} + \underbrace{d_x^3 \left(c_4 K_t^x + c_5 K_t^x K_t^y \right)}_{\text{moments and assignments}} + \underbrace{c_6 L \left(K_t^{x(i)} + K_t^y \right)^3}_{\text{Murty (worst case)}} \right]$$

where c_1 to c_6 are constants and c_3 is for sampling from the Poisson distribution. If we assume that SMC tracks number of births and deaths well in average (which it indeed does), then we can simplify the term above

$$C_{\text{SMC}}(\theta, t, N) \approx N \left[c_1 + c_3 + c_2 K_{t-1}^x + d_x^3 \left(c_4 K_t^x + c_5 K_t^x K_t^y \right) + c_6 L \left(K_t^x + K_t^y \right)^3 \right]$$

The process $\{K_t^x\}_{t \geq 1}$ is a Markov and its stationary distribution is $\mathcal{P}(\lambda_x)$ where $\lambda_x = \frac{\lambda_b}{1 - p_s}$. Also $K_t^y = K_t^d + K_t^f$ and for simplicity we write $K_t^d \approx p_d K_t^x$. Therefore the stationary distribution for $\{K_t^x + K_t^y\}_{t \geq 1} \approx \{(1 + p_d) K_t^x + K_t^f\}_{t \geq 1}$ is approximately $\mathcal{P}(\lambda_y)$ where $\lambda_y = \lambda_x (1 + p_d) + \lambda_f$. Therefore, assuming stationarity at time t and substituting $\mathbb{E}_{\mathcal{P}(\lambda)}(X^3) = \lambda^3 + 3\lambda^2 + \lambda$, the expected cost will be

$$\mathbb{E}_\theta \left[C_{\text{SMC}}(\theta, t, N) \right] \approx N \left[c_1 + c_3 + \left(c_2 + d_x^3 \left[c_4 + c_5 \left(p_d + \lambda_f \right) \right] \right) \lambda_x + c_5 p_d \lambda_x^2 + c_6 L \left(\lambda_y^3 + 3\lambda_y^2 + \lambda_y \right) \right]$$

It is worth emphasising that the computational cost of the SMC depends on θ unlike many time series models.

5.A.3.2 SMC-EM for the batch setting

The SMC-EM algorithm for the batch setting which is optimised with respect to computation time first runs a SMC filter by storing all its path trajectories i.e. $\{Z_{1:n}^{(i)}\}_{1 \leq i \leq N}$ and then calculates the estimates of required sufficient statistics for each $Z_{1:n}^{(i)}$ by using a forward filtering backward smoothing (FFBS) technique. Therefore, the overall expected cost of an optimised SMC-EM applied to a data of size n is

$$C_{\text{SMC-EM}} = C_{\text{FFBS}}(\theta, n, N) + \sum_{t=1}^{n} C_{\text{SMC}}(\theta, t, N) + c_7$$

where c_7 is the cost of the M-step, i.e. Λ. Let us denote the total number of targets up to time n is M and let $L_1, \ldots, L_M$ be their life lengths. The computational cost of FFBS to calculate the smoothed estimates of sufficient statistics for a target of life length L is

$\mathcal{O}(d_x^3 L)$. Therefore,

$$C_{\text{FFBS}}(\theta, n, N) = \sum_{i=1}^{N} \sum_{m=1}^{M^{(i)}} c_8 d_x^3 L_m^{(i)}$$

Assume the particle filter tracks well and $M^{(i)}$ and $L_m^{(i)}$, $m = 1, \ldots, M^{(i)}$ for particles $i = 1, \ldots, N$ are close enough to the true values M and L_m for $m = 1, \ldots, M$. Then, we have

$$C_{\text{FFBS}}(\theta, n, N) \approx \sum_{i=1}^{N} \sum_{m=1}^{M} c_8 d_x^3 L_m.$$

for some constant c_8. The expected values of L_m and M are $1/(1-p_s)$, $n\lambda_b$, respectively. Also, assume stationarity at all times so that the expectations of the terms $C_{\text{SMC}}(\theta, t, N)$ are the same and we have

$$\mathbb{E}_\theta \left[C_{\text{FFBS}}(\theta, n, N) \right] \approx c_8 N n d_x^3 \lambda_x.$$

As a result, given a data set of n time points, the overall expected cost of an optimised SMC-EM for the batch setting per iteration is

$$\mathbb{E}_\theta \left[C_{\text{SMC-EM}} \right] \approx \mathbb{E}_\theta \left[C_{\text{FFBS}}(\theta, n, N) \right] + n \mathbb{E}_\theta \left[C_{\text{SMC}}(\theta, t, N) \right] + c_7$$

5.A.3.3 SMC online EM

The overall cost of an SMC online EM for a data set of n time points is

$$C_{\text{SMC online EM}} \approx \sum_{t=1}^{n} \left[C_{\text{FSR}}(\theta, t, N) + C_{\text{SMC}}(\theta, t, N) + c_7 \right].$$

The forward smoothing recursion and maximisation used in the SMC online EM requires

$$C_{\text{FSR}}(\theta, t, N) = \sum_{i=1}^{N} c_9 K_t^{x(i)} d_x^5$$

calculations at time t for a constant c_9, whose expectation is $c_9 N \lambda_x d_x^5$ at stationarity. The overall expected cost of an SMC online EM for a data of n time steps, assuming stationarity, is

$$\mathbb{E}_\theta \left[C_{\text{SMC online EM}}(\theta, n, N) \right] \approx n \left(\mathbb{E}_\theta \left[C_{\text{FSR}}(\theta, t, N) \right] + \mathbb{E}_\theta \left[C_{\text{SMC}}(\theta, t, N) \right] + c_7 \right)$$

Chapter 6

Approximate Bayesian Computation for Maximum Likelihood Estimation in Hidden Markov Models

Summary: *In this chapter, we present methodology for implementing maximum likelihood estimation (MLE) in hidden Markov models (HMM) with intractable likelihoods in the context of approximate Bayesian computation (ABC). We show how both batch and online versions of gradient ascent MLE and expectation maximisation (EM) algorithms can be used for those HMMs using the ABC approach to confront the intractability. We demonstrate the performance of our methods first with examples on estimating the parameters of two intractable distributions, which are the α-stable and g-and-k distributions, and then with an example on estimating the parameters of the stochastic volatility model with α-stable returns.*

6.1 Introduction

6.1.1 Hidden Markov models

Hidden Markov models (HMM) are important statistical models in many fields including Bioinformatics (e.g. Durbin et al. [1998]), Econometrics (e.g. Kim et al. [1998]) and Population genetics (e.g. Felsenstein and Churchill [1996]); see also Cappé et al. [2005] for a recent overview. An HMM can be defined as a model comprised of the processes $\{R_k\}_{k\geq 1}$ and $\{Y_k\}_{k\geq 1}$. The latent process $\{R_k \in \mathcal{R} \subseteq \mathbb{R}^{d_r}\}_{k\geq 1}$ is a Markov chain with an initial density ν_θ and the transition density f_θ, i.e.,

$$R_1 \sim \nu_\theta(r_1)dr_1, \quad R_k|(R_{1:k-1} = r_{1:k-1}) \sim f_\theta(r_k|r_{k-1})dr_k, \quad k \geq 2. \tag{6.1}$$

It is assumed that $\nu_\theta(r)$ and $f_\theta(r|r')$ are densities on $\mathcal{R}$ with respect to (w.r.t.) a suitable dominating measure denoted generically as dr. Next, $\{Y_k \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}\}_{k\geq 1}$ is the observation process where Y_k is conditionally independent of all other random variables given

$R_k = r_k$ and it has the conditional observation density $g_\theta(\cdot|r_k)$ on $\mathcal{Y}$ w.r.t. dy, i.e.

$$Y_k|\left(\{R_i\}_{i\geq 1} = \{r_i\}_{i\geq 1}, \{Y_i\}_{i\geq 1, i\neq k} = \{y_i\}_{i\geq 1, i\neq k}\right) \sim g_\theta(y_k|r_k)dy_k, \quad k \geq 1. \tag{6.2}$$

Finally, the law of the HMM is parametrised by θ taking values in some compact subset Θ of the Euclidean space $\mathbb{R}^{d_\theta}$.

6.1.2 Parameter estimation

A problem that often arises when choosing which HMM to fit to a particular data set is that of parameter estimation. Typically, the problem is formulated as choosing a particular HMM among the range of HMMs with transitional laws in (6.1) and (6.2) which are parametrised by $\theta \in \Theta$. We will denote the observed random variables of the HMM, i.e. data, up to time n as $\hat{Y}_{1:n}$, which are independent copies of the random variables $Y_{1:n}$. Then, given a sequence of observations $\hat{Y}_{1:n}$ the objective is to find the parameter vector $\theta^* \in \Theta$ that corresponds to the particular HMM from which the data were generated.

A common approach to estimating θ^* is *maximum likelihood estimation* (MLE). In the MLE approach, the parameter estimate given the observations $\hat{Y}_{1:n}$, denoted θ_{ML}, is obtained by the following maximisation procedure:

$$\theta_{\mathrm{ML}} = \arg\max_{\theta\in\Theta} p_\theta(\hat{Y}_{1:n}),$$

where $p_\theta(\hat{Y}_{1:n})$ is the probability density, or the *likelihood*, of the observations $\hat{Y}_{1:n}$, defined by

$$p_\theta(y_{1:n}) = \int_{\mathcal{R}^n} \nu_\theta(r_1) g_\theta(y_1|r_1) \left[\prod_{k=2}^{n} f_\theta(r_k|r_{k-1}) g_\theta(y_k|r_k)\right] dr_{1:n}, \quad \forall y_{1:n} \in \mathcal{Y}^n. \tag{6.3}$$

Unless the model is simple, e.g. linear Gaussian or when $\mathcal{R}$ is a finite set, one can seldom evaluate the likelihood in (6.3) analytically. There are a variety of techniques, for example sequential Monte Carlo (SMC), for numerically estimating or maximising the likelihood using Monte Carlo; see Kantas et al. [2009] for a recent comprehensive and comparative review and discussion of SMC methods for parameter estimation in HMMs.

6.1.3 Approximate Bayesian computation for parameter estimation

In a wide range of applications the standard Monte Carlo methods cannot be used for parameter estimation, for example when the probability density $g_\theta(\cdot|r)$ of the observed

state of the HMM given the hidden state $R_k = r$ is *intractable* for any $r \in \mathcal{R}$. By intractability, we mean either that this density cannot be evaluated analytically and has no unbiased Monte Carlo estimator or that it is computationally prohibitive to calculate. Despite the intractability, one is often still able to generate samples from the observation process for any value of the parameter θ; e.g. see Jasra et al. [2012] for an example in the HMM context. Specifically, one can usually sample from $g_\theta(\cdot|r)$ by first sampling $U \in \mathcal{U}$ from a straightforward distribution with density $\mu_\theta(u|r)$ on $\mathcal{U}$ w.r.t. to the dominating measure du, and then applying a certain transformation $t_\theta : \mathcal{U} \times \mathcal{R} \to \mathcal{Y}$ such that

$$t_\theta(U, r) \sim g_\theta(\cdot|r).$$

If $g_\theta(\cdot|r)$ is an intractable density which cannot be evaluated analytically, then typically it is the case that the function t_θ is a highly non-linear mapping (see Section 6.4.1 for an example), as observed in a different context in Guyader et al. [2011]. If $g_\theta(\cdot|r)$ is available in analytical form but prohibitive to calculate, then U often consists of the latent variables of a hierarchical model which generates the observation and t_θ is in rather simple form.

The ability to sample from a distribution with an intractable probability density has led to the development of *approximate Bayesian computational* (ABC) methods, in which the basic idea is to circumvent the intractability of a distribution by generating samples from it. ABC has been a highly popular method for confronting intractability, one can see e.g. Pritchard et al. [1999], Beaumont et al. [2002], Marjoram et al. [2003] for its first examples and Marin et al. [2011] for a recent review.

As the name approximate Bayesian computation reveals, classical ABC methods treat the problem of estimating θ^* in a Bayesian framework where one assigns a prior distribution to θ. Numerous Monte Carlo schemes based on rejection sampling [Pritchard et al., 1999], Markov chain Monte Carlo (MCMC) [Marjoram et al., 2003], SMC samplers [Del Moral et al., 2012], etc. have been proposed in this context with success. However, when they deal with very large data sets, numerical Bayesian methods for static parameter estimation are known to suffer either from computational complexity (when MCMC is used) or from particle path degeneracy (when SMC is used); see Andrieu et al. [2005]; Olsson et al. [2008] for a discussion of this issue on a general basis.

As an alternative to Bayesian estimation, in Dean et al. [2011] the use of ABC was investigated in the MLE context, where θ^* is estimated by taking the value of θ which maximises some principled ABC approximation of the likelihood, which is itself estimated using Monte Carlo simulation. We will refer to the procedure of maximising this ABC approximation of the likelihood for the purpose MLE as *ABC MLE* from now on. However, the authors in Dean et al. [2011] do not propose a methodology for implementing the ABC MLE approaches presented in their work.

6.1.4 Outline of the chapter

In this chapter, we present both batch and online methods to implement ABC MLE for HMMs with intractable observation densities. In particular, we will demonstrate how *gradient ascent* and *expectation-maximisation* (EM) algorithms can be implemented in batch and online settings. We show how to apply the idea of *noisy ABC* [Dean et al., 2011; Fearnhead and Prangle, 2012; Wilkinson, 2008] within these methods in order to get rid of any asymptotic bias introduced by ABC approximations. The methods we provide are always equipped with a Monte Carlo technique which is SMC based in its most general form. The SMC scheme for ABC that we propose in this work is based on a slight but crucial modification of the SMC scheme for ABC that is proposed in Jasra et al. [2012] for SMC filtering. In fact, it is that modification which makes the MLE methods implementable.

The organisation of the rest of the chapter is as follows: First, we will review the ABC MLE approaches for HMMs with intractable likelihoods in Section 6.2. Then, in Section 6.3 we will present the methodology to implement the approaches covered in Section 6.2. We will demonstrate the performance of the developed methods with three different examples in Section 6.4. The first two examples are on estimating the parameters of α-stable and *g-and-k* distributions given a sequence of i.i.d. random variables, noting that the i.i.d. case corresponds to a special kind of HMM. The final example we will show is on online estimation of the static parameters of the stochastic volatility model with α-stable returns. The last section will contain a discussion on the methods developed and the results obtained.

6.2 ABC MLE approaches for HMM

6.2.1 Standard ABC MLE

Given data $\hat{Y}_{1:n}$ generated from a general statistical model parametrised by θ, one popular method for approximating the likelihood $p_\theta(\hat{Y}_{1:n})$ is ABC. In the standard ABC approach, one approximates the likelihood $p_\theta(\hat{Y}_{1:n})$ via the probability of the form

$$P_\theta\left(\rho_{d_s(n)}\left[s_n(Y_{1:n}); s_n(\hat{Y}_{1:n})\right] \leq \epsilon \middle| \hat{Y}_{1:n}\right). \tag{6.4}$$

In (6.4), $Y_{1:n}$ denotes the observed random variables of the statistical model, $s_n : \mathcal{Y}^n \to \mathbb{R}^{d_s(n)}$ is a statistic associated to n data points, $\rho_d(\cdot;\cdot)$ is some suitable metric on $\mathbb{R}^d$, and $\epsilon > 0$ is a constant which reflects the accuracy of the approximation. One expects that the approximation gets better as ϵ goes towards zero. In practice the probability in (6.4) is itself estimated using Monte Carlo techniques. The intuitive justification for the ABC

approximation is that, if the statistic s_n is sufficient for θ, for sufficiently small ϵ

$$\frac{P_\theta\left(\rho_{d_s(n)}\left[s_n(Y_{1:n}); s_n(\hat{Y}_{1:n})\right] \leq \epsilon \middle| \hat{Y}_{1:n}\right)}{V^{\epsilon,\kappa}_{\hat{Y}_{1:n}}} \approx p_\theta(\hat{Y}_{1:n})$$

where $V^{\epsilon,\kappa}_{\hat{Y}_{1:n}}$ denotes the volume of the $\rho_{d_s(n)}$-ball of radius ϵ around the point $s_n(\hat{Y}_{1:n})$. Thus the probabilities in (6.4) will provide a good approximation to the likelihood, up to the value of some normalising factor which is independent of θ and hence can be ignored.

In Dean et al. [2011], the authors considered performing ABC parameter estimation for HMMs using a specialisation, proposed in Jasra et al. [2012], of the standard ABC likelihood approximation (6.4) for when the observations are generated by a HMM. Specifically, given a sequence of observations $\hat{Y}_{1:n}$ from the HMM defined in (6.1) and (6.2), they approximate the corresponding likelihood function $p_\theta(\hat{Y}_{1:n})$ in (6.3) (up to a proportionality) with the probability

$$\begin{aligned} &P_\theta\left(Y_1 \in B^\epsilon_{\hat{Y}_1}, \ldots, Y_n \in B^\epsilon_{\hat{Y}_n} \middle| \hat{Y}_{1:n}\right) \\ &\quad = \int_{\mathcal{R}^n \times \mathcal{Y}^n} \nu_\theta(r_1) g_\theta(y_1|r_1) \mathbb{I}_{B^\epsilon_{\hat{Y}_1}}(y_1) \left[\prod_{k=2}^n f_\theta(r_k|r_{k-1}) g_\theta(y_k|r_k) \mathbb{I}_{B^\epsilon_{\hat{Y}_k}}(y_k)\right] dr_{1:n} dy_{1:n}, \quad (6.5) \end{aligned}$$

where B^ϵ_y denotes the ball of radius ϵ centred around the point y for all $y \in \mathbb{R}^{d_y}$. In the same work, it was observed that the ABC approximate likelihood obtained by normalising the probability in (6.5) is equal to the likelihood of the data $\hat{Y}_{1:n}$ under the law of the *perturbed HMM* $\{R_k, Y^\epsilon_k\}_{k\geq 1}$ defined such that observed states $\{Y^\epsilon_k\}_{k\geq 1}$ admit

$$Y^\epsilon_k = Y_k + \epsilon Z_k, \quad Z_k \sim^{\text{i.i.d.}} \text{Unif}_{B^1_0}, \quad k \geq 1. \quad (6.6)$$

This observation can be verified as follows. While the initial and transition densities for the hidden states of the perturbed HMM $\{R_k, Y^\epsilon_k\}_{k\geq 1}$ are still ν_θ and f_θ, the components of the observation process $\{Y^\epsilon_k\}_{k\geq 1}$ has the 'perturbed' observation density

$$g^\epsilon_\theta(y|r) = \frac{1}{|B^\epsilon_y|} \int_{\mathcal{Y}} g_\theta(y'|r) \mathbb{I}_{B^\epsilon_y}(y') dy'.$$

We will denote the likelihood of data $\hat{Y}_{1:n}$ under the law of the perturbed HMM as $p^\epsilon_\theta(\hat{Y}_{1:n})$, where p^ϵ_θ is defined as

$$p^\epsilon_\theta(y_{1:n}) = \int_{\mathcal{R}^n} \nu_\theta(r_1) g^\epsilon_\theta(y_1|r_1) \prod_{k=2}^n f_\theta(r_k|r_{k-1}) g^\epsilon_\theta(y_k|r_k) dr_{1:n}, \quad \forall y_{1:n} \in \mathcal{Y}^n. \quad (6.7)$$

It can be seen from (6.5) and (6.7) that

$$P_\theta\left(Y_1 \in B^\epsilon_{\hat{Y}_1}, \ldots, Y_n \in B^\epsilon_{\hat{Y}_n} \,\middle|\, \hat{Y}_{1:n}\right) = p^\epsilon_\theta(\hat{Y}_{1:n}) \prod_{k=1}^{n} |B^\epsilon_{\hat{Y}_k}|$$

Therefore, the proportionality between the density $p^\epsilon_\theta(\hat{Y}_{1:n})$ and the probability $P_\theta(Y_1 \in B^\epsilon_{\hat{Y}_1}, \ldots, Y_n \in B^\epsilon_{\hat{Y}_n} | \hat{Y}_{1:n})$ clearly does not depend on the value of θ. This observation leads to a very useful fact: In order to find the value of θ that maximises the principled approximation of the likelihood in (6.5), one could in theory implement MLE for the perturbed HMMs $\{R_k, Y^\epsilon_k\}_{k\geq 1}$ using the observations $\hat{Y}_{1:n}$ for $Y^\epsilon_{1:n}$. The benefit of this approach is that it retains the Markovian structure of the model which facilitates both the mathematical analysis and computational implementation of the method. Note that equation (6.5) is also a formalisation of the basic idea behind ABC MLE; all the other ABC MLE approaches reviewed in the subsequent sections can be regarded to be based on modifications of (6.5).

6.2.2 Noisy ABC MLE

It was shown in Dean et al. [2011] that the standard ABC MLE, if implemented exactly, leads to an asymptotically biased estimate of the parameter vector θ^* in the sense that as $n \to \infty$ the corresponding ABC MLE estimate will converge to some point $\theta^{*,\epsilon} \neq \theta^*$ in the parameter space Θ, although this bias can be made arbitrarily small by choosing a sufficiently small value of ϵ. This bias is due to the fact that in standard ABC MLE one approximates the likelihood of the data generated by the HMM $\{R_k, Y_k\}_{k\geq 0}$ with the likelihood of the same data under the law of the perturbed HMM $\{R_k, Y^\epsilon_k\}_{k\geq 1}$. Thus in effect standard ABC MLE is equivalent to performing MLE with a misspecified collection of models which in general leads to biased parameter estimates [White, 1982]. A related observation was also made in Wilkinson [2008], it was shown that the standard ABC would be 'calibrated' under the assumption of model error, which corresponds to the mismatch between the original HMM $\{R_k, Y_k\}_{k\geq 1}$ and the perturbed HMM $\{R_k, Y^\epsilon_k\}_{k\geq 1}$ here in this context.

The asymptotic bias in the standard ABC MLE approach can be removed by modifying the data $\hat{Y}_{1:n}$ so that the law corresponding to the process that generated the modified data is equal to the law of the perturbed HMM $\{R_k, Y^\epsilon_k\}_{k\geq 1}$ defined above. In practice this can be done by simply adding independent uniform noise to each of the observations $\hat{Y}_1, \ldots, \hat{Y}_n$ to get noisy observations

$$\hat{Y}^\epsilon_k = \hat{Y}_k + \epsilon Z_k, \quad Z_k \sim^{\text{i.i.d.}} \text{Unif}_{B^1_0}, \quad 1 \leq k \leq n. \tag{6.8}$$

One can then perform the same ABC MLE approach described in Section 6.2.1 using

the noisy observations in place of the original ones. The resulting method is known as the *noisy ABC MLE* [Dean et al., 2011] and it produces an unbiased estimator of the parameter vector θ^* as $n \to \infty$. The intuitive reason for the unbiasedness is easy to see; the probability density of the noisy data $\hat{Y}^{\epsilon}_{1:n}$ is precisely $p^{\epsilon}_{\theta}(\hat{Y}^{\epsilon}_{1:n})$, which is the likelihood function of the perturbed HMM $\{R_k, Y^{\epsilon}_k\}_{k\geq 1}$ given $Y^{\epsilon}_{1:n} = \hat{Y}^{\epsilon}_{1:n}$. The term "noisy ABC" was also used in Fearnhead and Prangle [2012] in a Bayesian framework with the same idea of adding noise to the (statistics of) data for the purpose of "calibrating the model" instead of "removing the bias".

6.2.3 Smoothed ABC MLE

When the standard and the noisy ABC MLE approaches described above are subject to a SMC implementation of an iterative MLE method, it may be necessary to have sufficiently accurate particle approximation of certain quantities, such as gradients of some densities, at least locally in a neighbourhood of the current parameter estimate. However it is well known that Monte Carlo estimates of gradients of densities can be poor, especially when the densities themselves contain discontinuities. This can present problems due to the presence of the kernel of the uniform density in the ABC approximation of the likelihood in (6.7). The fundamental problem is that the ABC approximation of the likelihoods essentially involves convolving the true observation density g_θ with the density of a uniform distribution. The sharp discontinuities of this distribution mean that Monte Carlo estimates of expectations w.r.t. it are very poor at capturing the variations of these expected values w.r.t. any underlying parameters.

The way to resolve this situation is to implement ABC with an approximation of the likelihoods that involves convolving the g_θ with the density of a smooth centred distribution denoted by κ. In particular, this can be done by using the *smoothed ABC MLE* (S-ABC MLE) approach described in Dean et al. [2011]. In this approach one approximates the true likelihood $p_\theta(\hat{Y}_{1:n})$ of the data $\hat{Y}_{1:n}$ with the likelihood of the data under the law of the perturbed HMM $\{R_k, Y^{\epsilon,\kappa}_k\}_{k\geq 0}$ which is this time defined such that the observed states $\{Y^{\epsilon,\kappa}_k\}_{k\geq 1}$ admit

$$Y^{\epsilon,\kappa}_k = Y_k + \epsilon Z_k, \quad Z_k \sim^{\text{i.i.d.}} \kappa, \quad k \geq 1,$$

Therefore, $p_\theta(\hat{Y}_{1:n})$ is approximated by $p^{\epsilon,\kappa}_\theta(\hat{Y}_{1:n})$ where $p^{\epsilon,\kappa}_\theta$ is defined as

$$p^{\epsilon,\kappa}_\theta(y_{1:n}) = \int_{\mathcal{R}^n} \nu_\theta(r_1) g^{\epsilon,\kappa}_\theta(y_1|r_1) \left[\prod_{k=2}^{n} f_\theta(r_k|r_{k-1}) g^{\epsilon,\kappa}_\theta(y_k|r_k)\right] dr_{1:n}, \quad \forall y_{1:n} \in \mathcal{Y} \tag{6.9}$$

where this time the perturbed observation density $g^{\epsilon,\kappa}_\theta$ is obtained by convolving g_θ with

κ i.e.

$$g_\theta^{\epsilon,\kappa}(y|r) = \int_{\mathcal{Y}} g_\theta(y'|r)\frac{1}{\epsilon}\kappa\left(\frac{y-y'}{\epsilon}\right)dy'.$$

In practice the random variables Z_k are often chosen to be standard normal random variables.

Finally, we note that the S-ABC MLE suffers from the same problems with asymptotic bias as does the standard ABC MLE. However, the notion of noisy ABC MLE has a natural extension to the smoothed case which again results in an unbiased estimate of the parameter values. That is, instead of using $\hat{Y}_{1:n}$, one can use new observations $\hat{Y}_{1:n}^{\epsilon,\kappa}$ obtained by

$$\hat{Y}_k^{\epsilon,\kappa} = \hat{Y}_k + \epsilon Z_k, \quad Z_k \sim^{\text{i.i.d.}} \kappa, \quad 1 \leq k \leq n. \tag{6.10}$$

The resulting approach will be called *smoothed noisy ABC MLE* (SN-ABC MLE) in this chapter.

6.2.4 Summary

In Table 6.1, we summarise the ABC MLE approaches we have covered in this section. In Section 6.3, in order to avoid unnecessary repeatings, we will present our methodology for implementing ABC MLE for HMMs mainly based on the SN-ABC MLE approach only. We have chosen SN-ABC MLE for its desirable properties, namely unbiasedness and its broader applicability; although we do dot have any loss of generality by having done so. The gradient ascent and EM algorithms explained in Sections 6.3.1 and 6.3.2 can be modified for the other ABC MLE approaches (if applicable) with obvious modifications such as removing the noise or using the uniform distribution rather than κ. Finally, note that these algorithms involve SMC approximations in practice, and indeed choosing an appropriate SMC scheme is of great essence in order to be able to implement the algorithms.

Table 6.1: A comparison of ABC MLE approaches.

MLE method	**output**	**bias**	**applicability**
ideal MLE	$\arg\max_{\theta\in\Theta} p_\theta(\hat{Y}_{1:n})$	unbiased	impossible
standard ABC MLE	$\arg\max_{\theta\in\Theta} p_\theta^\epsilon(\hat{Y}_{1:n})$	biased	restricted
noisy ABC MLE	$\arg\max_{\theta\in\Theta} p_\theta^\epsilon(\hat{Y}_{1:n}^\epsilon)$	unbiased	restricted
S-ABC MLE	$\arg\max_{\theta\in\Theta} p_\theta^{\epsilon,\kappa}(\hat{Y}_{1:n})$	biased	generally applicable
SN-ABC MLE	$\arg\max_{\theta\in\Theta} p_\theta^{\epsilon,\kappa}(\hat{Y}_{1:n}^{\epsilon,\kappa})$	unbiased	generally applicable

6.3 Implementing ABC MLE

Although it was shown in Jasra et al. [2012] that the ABC approximate likelihoods (6.5) can themselves be estimated using SMC methods, there was no investigation as to how these SMC based estimates can be used in practice to find the parameter value that maximises the ABC approximate likelihood. The purpose of this section is to show that due to the underlying mathematical structure of ABC one can efficiently and accurately implement the ABC MLE procedure via the standard MLE algorithms. We discuss in detail how this can be done in Sections 6.3.1 and 6.3.2. However, here we give the basic idea behind the algorithms described in those sections.

Consider the SN-ABC MLE approach where the true likelihood $p_\theta(\hat{Y}_{1:n})$ in (6.3) is approximated with $p_\theta^{\epsilon,\kappa}(\hat{Y}_{1:n}^{\epsilon,\kappa})$ where $p_\theta^{\epsilon,\kappa}(\cdot)$ is the S-ABC MLE likelihood defined in (6.9) and $\hat{Y}_{1:n}^{\epsilon,\kappa}$ are observations with added smooth noise defined in (6.10). Recall that $p_\theta^{\epsilon,\kappa}(\hat{Y}_{1:n}^{\epsilon,\kappa})$ is the likelihood of $\hat{Y}_{1:n}^{\epsilon,\kappa}$ under the law of the perturbed HMM $\{R_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$; so one would obtain the SN-ABC MLE estimate of θ^* if they could implement MLE for $\{R_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$ given $Y_{1:n}^{\epsilon,\kappa} = \hat{Y}_{1:n}^{\epsilon,\kappa}$. However, MLE for $\{R_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$ is as hard as MLE for the original HMM $\{R_k, Y_k\}_{k\geq 1}$, which we already know to be impossible due to the intractability of $g_\theta(\cdot|r)$. The reason is similar; this time the perturbed observation density $g_\theta^{\epsilon,\kappa}(\cdot|r)$ is intractable. Therefore, we cannot use the HMM $\{R_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$ directly to implement SN-ABC MLE.

The crucial point here is that one can construct an *equivalent* HMM to $\{R_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$ which *is* tractable in terms of its densities. Recall that one can usually sample from $g_\theta(\cdot|r)$ by first sampling $U \in \mathcal{U}$ from $\mu_\theta(\cdot|r)$ and applying the transformation $t_\theta(U, r)$. From this it follows that $p_\theta^{\epsilon,\kappa}(\cdot)$ is also the likelihood function corresponding to the *expanded HMMs*

$$\{(R_k, U_k), Y_k^{\epsilon,\kappa}\}_{k\geq 1}$$

where $\{R_k\}_{k\geq 1}$ is equal to the hidden state of the original HMM, $U_k \sim \mu_\theta(\cdot|R_k)$ for all k and

$$Y_k^{\epsilon,\kappa} = t_\theta(U_k, R_k) + \epsilon Z_k, \quad Z_k \sim^{\text{i.i.d.}} \kappa, \quad k \geq 1.$$

For this particular HMM, we have $\{X_k := (R_k, U_k)\}_{k\geq 1}$ to be the latent process taking values from $\mathcal{X} = \mathcal{R}\times\mathcal{U}$ and $\{Y_k^{\epsilon,\kappa}\}_{k\geq 1}$ is the observation process. The initial and transition densities $\pi_\theta(x)$ and $q_\theta(x|x')$ for $\{X_k\}_{k\geq 1}$ w.r.t. the dominating measure $dx = drdu$ and the observation density $h_\theta^{\epsilon,\kappa}(y|x)$ for $\{Y_k\}_{k\geq 1}$ w.r.t. dy are as follows:

$$\pi_\theta(x) = \nu(r)\mu_\theta(u|r), \quad q_\theta(x'|x) = f_\theta(r'|r)\mu_\theta(u'|r'), \quad h_\theta^{\epsilon,\kappa}(y|x) = \frac{1}{\epsilon}\kappa\left(\frac{y - t_\theta(x)}{\epsilon}\right). \tag{6.11}$$

where $x = (r, u)$ and $x' = (r', u')$. Depending on whether we choose to use the S-ABC MLE or SN-ABC MLE, we take $Y_{1:n}^{\epsilon,\kappa} = \hat{Y}_{1:n}$ or $Y_{1:n}^{\epsilon,\kappa} = \hat{Y}_{1:n}^{\epsilon,\kappa}$, respectively. Again, to avoid repeating ourselves, from now on we will carry on with the SN-ABC MLE approach and as a result we are given a particular realisation $Y_{1:n}^{\epsilon,\kappa} = \hat{Y}_{1:n}^{\epsilon,\kappa}$. SN-ABC MLE reduces to searching for

$$\theta_{\mathrm{ML}}^{\epsilon,\kappa} = \arg\max_{\theta\in\Theta} p_\theta^{\epsilon,\kappa}(\hat{Y}_{1:n}^{\epsilon,\kappa}),$$

where $p_\theta^{\epsilon,\kappa}$ is defined in (6.9) and it can be rewritten in terms of the densities in (6.11) as

$$p_\theta^{\epsilon,\kappa}(y_{1:n}) = \int_{\mathcal{X}^n} \pi_\theta(x_1) h_\theta^{\epsilon,\kappa}(y_1|x_1) \left[\prod_{k=2}^{n} q_\theta(x_k|x_{k-1}) h_\theta^{\epsilon,\kappa}(y_k|x_k)\right] dx_{1:n}, \quad \forall y_{1:n} \in \mathcal{Y}^n. \tag{6.12}$$

Thus, in practice one can find the SN-ABC MLE estimate by applying standard MLE algorithms to the expanded HMMs $\{X_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$ using the noisy observations $\hat{Y}_{1:n}^{\epsilon,\kappa}$.

Finally, we note that there is a published remark [Andrieu et al., 2012] which also mentions (independently) the idea of making use of the intermediate random variable U in the ABC context in a Bayesian framework; however the idea is not developed further and finalised with an implementable method.

6.3.0.1 SMC algorithm for the expanded HMM

In Sections 6.3.1 and 6.3.2, we describe two possible SMC based MLE methods for HMM in the ABC context by exploiting the availability of this expanded HMM $\{X_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$. Before going into the details of these methods, we present our SMC filtering scheme for $\{X_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$ in the algorithm below.

Algorithm 6.1. ***SMC filtering for the expanded HMM*** $\{X_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$
For $k = 1$; for $i = 1, \ldots, N$ sample $R_1^{(i)} \sim \nu_\theta$, $U_1^{(i)} \sim \mu_\theta(\cdot|R_1^{(i)})$, and set $X_1^{(i)} = (R_1^{(i)}, U_1^{(i)})$ and calculate

$$W_1^{(i)} \propto h_\theta^{\epsilon,\kappa}(Y_1^{\epsilon,\kappa}|X_1^{(i)}), \quad \sum_{i=1}^{N} W_1^{(i)} = 1.$$

For $k = 2, 3, \ldots$

- *Resample $\{X_{0:k-1}^{(i)}\}_{1\leq i\leq N}$ according to the weights $\{W_{k-1}^{(i)}\}_{1\leq i\leq N}$ to get resampled particles $\{\widetilde{X}_{0:k-1}^{(i)}\}_{1\leq i\leq N}$.*
- *For $i = 1, \ldots, N$, sample $R_k^{(i)} \sim f_\theta(\cdot|\widetilde{R}_{k-1}^{(i)})$ and $U_k^{(i)} \sim \mu_\theta(\cdot|R_k^{(i)})$; set $X_k^{(i)} = (R_k^{(i)}, U_k^{(i)})$ and $X_{1:k}^{(i)} = (\widetilde{X}_{1:k-1}^{(i)}, X_k^{(i)})$. Calculate*

$$W_k^{(i)} \propto h_\theta^{\epsilon,\kappa}(Y_k^{\epsilon,\kappa}|X_k^{(i)}), \quad \sum_{i=1}^{N} W_k^{(i)} = 1.$$

Algorithm 6.1 provides the SMC estimates for the posterior distributions such as $p_\theta^{\epsilon,\kappa}(x_{1:n}|Y_{1:n}^{\epsilon,\kappa})$, $p_\theta^{\epsilon,\kappa}(x_n|Y_{1:n}^{\epsilon,\kappa})$ and $p_\theta^{\epsilon,\kappa}(x_n|Y_{1:n-1}^{\epsilon,\kappa})$. In this work, we will consider the last one in particular, whose SMC approximation is provided by Algorithm 6.1 as

$$p_\theta^{\epsilon,\kappa,N}(dx_n|Y_{1:n-1}^{\epsilon,\kappa}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X_n^{(i)}}(dx_n) \tag{6.13}$$

The implementation in Algorithm 6.1 is based on the bootstrap particle implementation of Gordon et al. [1993]. Note that any SMC implementation may be used, e.g. the auxiliary SMC method of Pitt and Shephard [1999], bootstrap with optimal proposal Doucet et al. [2001]; see e.g. Del Moral [2004]; Doucet et al. [2001] for more examples.

In Jasra et al. [2012], the authors propose an SMC filtering algorithm using another expanded HMM, namely

$$\{(R_k, Y_k), Y_k^\epsilon\}_{k\geq 1}.$$

In this HMM, the hidden state at time k is (R_k, Y_k) i.e. the components of the original HMM and Y_k^ϵ is defined in (6.6). It is indeed possible to sample from the hidden states (R_k, Y_k) by sampling from their transition density

$$f_\theta(r_k|r_{k-1})g_\theta(y_k|r_k)$$

and the SMC filter only needs to calculate the density of a uniform distribution centred at the value of Y_k^ϵ on order to weight the sampled particles. For any given θ and $y_{1:n}$, this SMC filtering algorithm also provides an unbiased SMC estimate of the ABC probability in (6.5), hence of the ABC likelihood $p_\theta^\epsilon(y_{1:n})$ in (6.7) for any given θ and $y_{1:n}$ up to a proportionality. Moreover, the algorithm for $\{(R_k, Y_k), Y_k^\epsilon\}$ can be modified for $\{(R_k, Y_k), Y_k^{\epsilon,\kappa}\}$ with a straightforward manner, namely the uniform distribution is replaced with a smooth distribution centred at $Y_k^{\epsilon,\kappa}$. The resulting SMC filtering algorithm would be equivalent to Algorithm 6.1 in terms of filtering for the hidden process $\{R_k\}_{k\geq 1}$ of the original HMM; also we can have an unbiased SMC estimate of $p_\theta^{\epsilon,\kappa}(y_{1:n})$ for any given θ and $y_{1:n}$. The problem, however, is that it is in general required to be able to *compute* the transition density of hidden states of a HMM (or its gradient) in order to implement MLE methods for it. Obviously neither $f_\theta(r_k|r_{k-1})g_\theta(y_k|r_k)$ nor its gradient can be computed; therefore $\{(R_k, Y_k), Y_k^\epsilon\}_{k\geq 1}$ is not practical for implementing MLE. As a result, although both HMMs are equivalent in the SMC filtering context, $\{(R_k, U_k), Y_k^\epsilon\}_{k\geq 1}$ should be preferred over $\{(R_k, Y_k), Y_k^\epsilon\}_{k\geq 1}$ in the MLE context.

6.3.1 Gradient ascent ABC MLE

We show in this section that it is possible to devise batch and online gradient ascent algorithms for HMMs with intractable observation densities in order to implement the SN-ABC MLE approach. Specifically, we apply the gradient ascent MLE algorithm to $\{X_k, Y_k^{\epsilon,\kappa} = t_\theta(X_k) + \epsilon Z_k\}_{k\geq 1}$ where $\{Z_k\}_{k\geq 1}$ are taken as i.i.d. standard normal random variables i.e.

$$h_\theta^{\epsilon,\kappa}(y|x) = \mathcal{N}(y; t_\theta(x), \epsilon^2).$$

Also, for simplicity we fix $\hat{Y}_{1:n}^{\epsilon,\kappa} = y_{1:n}$.

6.3.1.1 Batch gradient ascent

The batch gradient ascent algorithm is an iterative procedure implemented as follows: We begin with $\theta^{(0)}$ and assume that we have the estimate $\theta^{(j-1)}$ at the end of the the $(j-1)$'th iteration. At the j'th iteration we update the parameter

$$\theta^{(j)} = \theta^{(j-1)} + \gamma_j \nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n})\big|_{\theta=\theta^{(j-1)}}.$$

Here $\{\gamma_j\}_{j\geq 1}$ is the sequence of step sizes satisfying $\sum_j \gamma_j = \infty$ and $\sum_j \gamma_j^2 < \infty$, ensuring convergence of the algorithm when it is used with the Monte Carlo approximations of the gradients $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n})$. It was shown in Poyiadjis et al. [2011] and Del Moral et al. [2011] that a stable SMC approximation of $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n})$, which we briefly outline in the following, is available for HMMs. First, the gradient term can be written as

$$\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n}) = \int_{\mathcal{X}^n} \left[S_{\theta,n}(x_{1:n}) + \nabla_\theta \log h_\theta^{\epsilon,\kappa}(y_n|x_n)\right] p_\theta^{\epsilon,\kappa}(x_{1:n}|y_{1:n}) dx_{1:n} \tag{6.14}$$

where the additive functional $S_{\theta,n} : \mathcal{X}^n \to \mathbb{R}^{d_\theta}$ is defined from additional functions as follows:

$$\begin{aligned} &S_{\theta,n}(x_{1:n}) = \sum_{k=1}^{n} s_{\theta,k}(x_{k-1}, x_k), \qquad (6.15)\\ &s_{\theta,k}(x_{k-1}, x_k) = \nabla_\theta \log h_\theta^{\epsilon,\kappa}(y_{k-1}|x_{k-1}) + \nabla_\theta \log q_\theta(x_k|x_{k-1}), \quad 2 \leq k \leq n,\\ &s_{\theta,1}(x_0, x_1) := s_{\theta,1}(x_1) = \nabla_\theta \log \pi_\theta(x_1). \end{aligned}$$

Notice that we have omitted the dependency on $y_{1:n-1}$ from the definition of $S_{\theta,n}$ for notational simplicity. The integral in (6.14) is simply the expectation of sum of the additive function $S_{\theta,n}$ and $\nabla_\theta \log h_\theta^{\epsilon,\kappa}(y_n|\cdot)$ under the posterior distribution of $X_{1:n}$ given $y_{1:n}$, and it can be evaluated in a forward manner as follows: Define the function T_n^θ :

$\mathcal{X} \to \mathbb{R}^{d_\theta}$

$$T_n^\theta(x_n) := \int_{\mathcal{X}^{n-1}} S_{\theta,n}(x_{1:n}) p_\theta^{\epsilon,\kappa}(x_{1:n-1}|y_{1:n-1}, x_n) dx_{1:n-1}$$
$$= \int_{\mathcal{X}} \left[T_{n-1}^\theta(x_{n-1}) + s_\theta(x_{n-1}, x_n)\right] p_\theta^{\epsilon,\kappa}(x_{n-1}|y_{1:n-1}, x_n) dx_{n-1}. \quad (6.16)$$

The recursion in (6.16) is a forward-only version of forward filtering backward smoothing and it is called the *forward smoothing recursion* in Del Moral et al. [2009]. Forward smoothing recursion is available for all additive functionals that have the form in (6.15); and it is particularly helpful in a sequential setting since one can perform the recursion using only the densities $\{p_\theta^{\epsilon,\kappa}(x_k|y_{1:k-1})\}_{k\geq 1}$, where we will call $p_\theta^{\epsilon,\kappa}(x_k|y_{1:k-1})$ the filter at time k. For simplicity, define

$$\eta_{\theta,k}(dx_k) := p_\theta^{\epsilon,\kappa}(x_k|y_{1:k-1})dx_k, \quad k \geq 1,$$

and let $\nu(\varphi) = \int \varphi(x)\nu(dx)$ for any measure ν on the σ-algebra generated by $\mathcal{X}$ and any bounded Borel measurable function φ defined on $\mathcal{X}$. Then, we can write

$$p_\theta^{\epsilon,\kappa}(x_{n-1}|y_{1:n-1}, x_n)dx_{n-1} = \frac{\eta_{\theta,n-1}(dx_{n-1}) h_\theta^{\epsilon,\kappa}(y_{n-1}|x_{n-1}) q_\theta(x_n|x_{n-1})}{\eta_{\theta,n-1}\left[h_\theta^{\epsilon,\kappa}(y_{n-1}|\cdot) q_\theta(x_n|\cdot)\right]}.$$

Once we have T_n^θ from T_{n-1}^θ using the forward smoothing recursion, it is possible to evaluate $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n})$ using again the filtering densities only:

$$\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n}) = \frac{\eta_{\theta,n}\left[T_n^\theta h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right] + \eta_{\theta,n}\left[\nabla_\theta h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right]}{\eta_{\theta,n}\left[h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right]}$$

Exact calculation of the filtering densities $\eta_{\theta,n}$ and hence $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n})$ is not possible, therefore Monte Carlo approximations are needed. We have already shown by equation (6.13) in Section 6.3.0.1 Using Algorithm 6.1 with N particles, it is possible to recursively compute particle approximations $\eta_{\theta,n}^N$ of $\eta_{\theta,n}$

$$\eta_\theta^N(dx_n) = \frac{1}{N}\sum_{i=1}^N \delta_{X_n^{(i)}}(dx_n),$$

where $X_n^{(i)}$, $i = 1, \ldots, N$, are called particles and δ_x is the Dirac measure concentrated at x. Also, a stable SMC approximation to $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n})$ is available by computing the recursion in (6.16) using the following $\mathcal{O}(N^2)$ particle approximation to the backward transition distribution $p_\theta^{\epsilon,\kappa}(dx_{n-1}|x_n, y_{1:n-1})$

$$p_\theta^{\epsilon,\kappa,N}(dx_{n-1}|x_n, y_{1:n-1}) = \frac{\eta_{\theta,n-1}^N(dx_{n-1}) h_\theta^{\epsilon,\kappa}(y_{n-1}|x_{n-1}) q_\theta(x_n|x_{n-1})}{\eta_{\theta,n-1}^N\left[h_\theta^{\epsilon,\kappa}(y_{n-1}|\cdot) q_\theta(x_n|\cdot)\right]}.$$

6.3.1.2 Online gradient ascent

The batch gradient ascent algorithm may be inefficient when n is large since each iteration requires a complete browse over the data sequence. An alternative to the batch algorithm is its online version, called the online gradient ascent MLE algorithm. An online gradient ascent algorithm can be implemented as follows [Del Moral et al., 2011; Poyiadjis et al., 2011]: Given $y_{1:n-1}$, assume we have the estimate θ_{n-1}. When y_n is received, we update the parameter

$$\theta_n = \theta_{n-1} + \gamma_n \nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_n|y_{1:n-1})\big|_{\theta=\theta_{n-1}}.$$

The gradient $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_n|y_{1:n-1})$ can be calculated making use of the filter derivative:

$$\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_n|y_{1:n-1}) = \frac{\eta_{\theta,n}\left[\nabla_\theta h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right] + \zeta_{\theta,n}\left[h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right]}{\eta_{\theta,n}\left[h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right]}$$

where $\zeta_{\theta,n}(dx_n)$ is the derivative of the filter $\eta_{\theta,n}$ and is defined as

$$\zeta_{\theta,n}(dx_n) = \eta_{\theta,n}(dx_n)\left[T_n^\theta(x_n) - \eta_{\theta,n}\left(T_n^\theta\right)\right]$$

Therefore, using an SMC algorithm, it is possible to recursively compute particle approximations $\zeta_{\theta,n}^N$ of $\zeta_{\theta,n}$ by using the same $\mathcal{O}(N^2)$ particle approximation to the forward smoothing recursion (i.e. T_n^θ's) as in the batch gradient ascent case to compute $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n})$. The resulting approximation of $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_n|y_{1:n-1})$, which is

$$\nabla_\theta^N \log p_\theta^{\epsilon,\kappa}(y_n|y_{1:n-1}) = \frac{\eta_{\theta,n}^N\left[\nabla_\theta h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right] + \zeta_{\theta,n}^N\left[h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right]}{\eta_{\theta,n}^N\left[h_\theta^{\epsilon,\kappa}(y_n|\cdot)\right]},$$

was numerically shown to be stable in Poyiadjis et al. [2011] and this was proved in Del Moral et al. [2011]. Without going into further details, we refer the reader to these works for the implementation details (e.g. see Algorithms 1 and 2 in Del Moral et al. [2011]) as well as proven stability results.

6.3.1.3 Controlling the stability

If the functions $s_{\theta,k}$ and hence the additive functionals $S_{\theta,n}$ have very high or infinite variances; we expect failure of the gradient ascent MLE algorithm. In particular, assuming $\kappa = \mathcal{N}(0,1)$, the gradient term

$$\nabla_\theta \log h_\theta^{\epsilon,\kappa}(Y_k^{\epsilon,\kappa}|X_k) = \frac{1}{\epsilon^2}\left[Y_k^{\epsilon,\kappa} - t_\theta(X_k)\right]\nabla_\theta t_\theta(X_k)$$

can be problematic in this sense. We may circumvent the instability problem by transforming each of the observations $\hat{Y}_1, \ldots, \hat{Y}_n$ to a subset $\mathcal{Y}_s \subseteq \mathcal{Y}$ by using a one-to-one func-

tion $\psi : \mathcal{Y} \to \mathcal{Y}_s$. Then, we can implement SN-ABC MLE for the HMM $\{X_k, Y_k^{\epsilon,\kappa,\psi}\}_{k\geq 1}$, where this time

$$Y_k^{\epsilon,\kappa,\psi} = \psi(Y_k) + \epsilon Z_k, \quad Z_k \sim^{\text{i.i.d.}} \mathcal{N}(0,1), \quad k \geq 1.$$

In this case, the observation density of the HMM $\{X_k, Y_k^{\epsilon,\kappa,\psi}\}_{k\geq 1}$ becomes

$$h_\theta^{\epsilon,\kappa,\psi}(y|x) = \mathcal{N}(y; \psi[t_\theta(x)], \epsilon^2).$$

Finally, the likelihood function of $\{X_k, Y_k^{\epsilon,\kappa,\psi}\}_{k\geq 1}$ is $p_\theta^{\epsilon,\kappa}$ in (6.12) with $h_\theta^{\epsilon,\kappa}$ is replaced by $h_\theta^{\epsilon,\kappa,\psi}$ i.e.

$$p_\theta^{\epsilon,\kappa,\psi}(y_{1:n}) = \int_{\mathcal{X}^n} \pi_\theta(x_1) h_\theta^{\epsilon,\kappa,\psi}(y_1|x_1) \left[\prod_{k=2}^{n} q_\theta(x_k|x_{k-1}) h_\theta^{\epsilon,\kappa,\psi}(y_k|x_k)\right] dx_{1:n}, \quad \forall y_{1:n} \in \mathcal{Y}^n.$$

We choose ψ such that the gradient of the logarithm of the new observation density

$$\nabla_\theta \log h_\theta^{\epsilon,\kappa,\psi}(Y_k^{\epsilon,\kappa,\psi}|X_k) = \frac{1}{\epsilon^2}\left(Y^{\epsilon,\kappa,\psi} - \psi[t_\theta(X_k)]\right) \nabla_\theta \psi[t_\theta(X_k)]$$

has smaller variance than it would have if no transformation were used. Note that in the case of a transformation function applied, we obtain the noisy data by first transforming the real data and then adding the noise, that is,

$$\hat{Y}_k^{\epsilon,\kappa,\psi} = \psi(\hat{Y}_k) + \epsilon Z_k, \quad Z_k \sim^{\text{i.i.d.}} \mathcal{N}(0,1), \quad 1 \leq k \leq n.$$

6.3.1.4 Special case: i.i.d. random variables with an intractable density

An i.i.d. process $\{Y_k\}_{k\geq 1}$ can be seen as a special type of HMM. Specifically, $\{Y_k\}_{k\geq 1}$ are i.i.d. w.r.t. a distribution with an intractable probability density g_θ. The objective is to perform MLE given a data sequence $\hat{Y}_{1:n}$ generated from the i.i.d. process. Again, we have the assumption that g_θ is intractable but we can sample from g_θ by generating $U \in \mathcal{U}$ from μ_θ, and by applying a certain transformation function $t_\theta : \mathcal{U} \to \mathcal{Y}$ so that $t_\theta(U) \sim g_\theta$.

Let us consider again the SN-ABC MLE approach where we want to maximise the likelihood of the noisy observations $\hat{Y}_{1:n}^{\epsilon,\kappa} = y_{1:n}$ under the law of the HMM $\{U_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$. The observation density for this HMM is modified as

$$h_\theta^{\epsilon,\kappa}(y|u) = \frac{1}{\epsilon}\kappa\left(\frac{y - t_\theta(u)}{\epsilon}\right).$$

Since we have $p_\theta^{\epsilon,\kappa}(y_n|y_{1:n-1}) = p_\theta^{\epsilon,\kappa}(y_n)$ and hence $\log p_\theta^{\epsilon,\kappa}(y_{1:n}) = \sum_{i=1}^n \log p_\theta^{\epsilon,\kappa}(y_n)$; the

batch and online gradient ascent MLE update rules algorithms reduce to

$$\theta^{(j)} = \theta^{(j-1)} + \gamma_j \sum_{k=1}^{n} \nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_k)\big|_{\theta=\theta^{(j-1)}}, \quad \theta_n = \theta_{n-1} + \gamma_n \nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_n)\big|_{\theta=\theta_{n-1}}.$$

Therefore, both batch and online gradient ascent algorithms involve independent Monte Carlo approximations to $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_n)$. Noting that

$$\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y) = \int_{\mathcal{Y}} \left[\nabla_\theta \log \mu_\theta(u) + \nabla_\theta \log h_\theta^{\epsilon,\kappa}(y|u)\right] p_\theta^{\epsilon,\kappa}(u|y) du, \tag{6.17}$$

the Monte Carlo approximation of $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y)$ involves a Monte Carlo approximation to the posterior distribution $p_\theta^{\epsilon,\kappa}(u|y)$ of U given y. We can use standard MCMC or importance sampling methods to obtain an approximation of $p_\theta^{\epsilon,\kappa}(u|y)$ with $N \geq 1$ samples as

$$p_\theta^{\epsilon,\kappa,N}(du|y) = \sum_{i=1}^{N} W^{(i)} \delta_{U^{(i)}}(du), \quad \sum_{i=1}^{N} W^{(i)} = 1.$$

If a MCMC is used to generate samples from $p_\theta^{\epsilon,\kappa,N}(u|y)$, we simply have $W^{(i)} = 1/N$. If self normalised importance sampling is used with a proposal density $\xi_\theta(u|y)$ then

$$W^{(i)} \propto \frac{\mu_\theta(U^{(i)}) h_\theta^{\epsilon,\kappa}(y|U^{(i)})}{\xi_\theta(U^{(i)}|y)}.$$

Therefore, the Monte Carlo approximation of (6.17) becomes

$$\nabla_\theta^N \log p_\theta^{\epsilon,\kappa}(y) = \sum_{i=1}^{N} W^{(i)} \left[\nabla_\theta \log \mu_\theta(U^{(i)}) + \nabla_\theta \log h_\theta^{\epsilon,\kappa}(y|U^{(i)})\right].$$

One important point to note about the i.i.d. case is that the original $\mathcal{O}(N^2)$ algorithm mentioned above reduces to an $\mathcal{O}(N)$ algorithm, so for a fixed computational source one can implement the gradient ascent algorithms with much more particles. Secondly, because of reduced computational complexity, we have more freedom to choose a sophisticated method for the Monte Carlo approximation, such as SMC samplers [Del Moral et al., 2006] (even though these methods are applicable also within the SMC algorithm for general HMMs with additional computational costs).

6.3.2 Expectation-maximisation

Although not as general as the gradient ascent MLE algorithm, the EM algorithm may be available in some models in the ABC context, at least for a part of the parameters in

θ. Consider the expanded HMM $\{X_k, Y_k^{\epsilon,\kappa}\}_{k\geq 1}$ and assume that that the quantity

$$Q(\theta', \theta) = \int_{\mathcal{X}^n} \log p_\theta^{\epsilon,\kappa}(x_{1:n}, y_{1:n}) p_{\theta'}^{\epsilon,\kappa}(x_{1:n}|y_{1:n}) dx_{1:n}$$

can be maximised w.r.t. θ. Then the EM algorithm at the j'th iteration calculates $Q(\theta^{(j-1)}, \theta)$ (E-step) sets $\theta^{(j)}$ to be the maximiser of $Q(\theta^{(j-1)}, \theta)$ (M-step) i.e.

$$\theta^{(j)} = \arg\max_{\theta\in\Theta} Q(\theta^{(j-1)}, \theta)$$

Moreover, if the joint density $p_\theta^{\epsilon,\kappa}(x_{1:n}, y_{1:n})$ of observations as well as latent variables belongs to the exponential family w.r.t. θ, then the E-step reduces to calculating the expectations of some additive sufficient statistics $S_n : \mathcal{X}^n \to \mathbb{R}^m$ (for some $m > 0$) defined similarly to (6.15) as

$$S_n(x_{1:n}) = s_1(x_1) + \sum_{k=2}^{n} s(x_{k-1}, x_k) \tag{6.18}$$

w.r.t. the posterior distribution $p_{\theta^{(j-1)}}(x_{1:n}|y_{1:n})$ of $X_{1:n}$ conditioned on $Y_{1:n}^{\epsilon,\kappa} = y_{1:n}$ at $\theta = \theta^{(j-1)}$. The M-step, then, can be characterised as a mapping $\Lambda : \mathbb{R}^m \to \Theta$ such that

$$\theta^{(j)} = \Lambda(S_n^{\theta^{(j-1)}}) = \arg\max_{\theta\in\Theta} Q(\theta^{(j-1)}; \theta).$$

where, for $\theta \in \Theta$, S_n^θ denotes the expectation of S_n w.r.t. $p_\theta^{\epsilon,\kappa}(x_{1:n}|y_{1:n})$ i.e.

$$S_n^\theta = \int_{\mathcal{X}^n} S_n(x_{1:n}) p_\theta^{\epsilon,\kappa}(x_{1:n}|y_{1:n}) dx_{1:n}.$$

Calculation of S_n^θ follows similar steps as calculating $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(y_{1:n})$ in the gradient ascent algorithm in the sense that we can use the forward smoothing recursion described in Section 6.3.1.1 to calculate S_n^θ since the sufficient statistics S_n are in the additive form [Del Moral et al., 2009]. Note that to emphasise the analogy between (6.15) and (6.18) we use the same letter S for those additive sufficient statistics.

Similar to the online gradient ascent algorithm, the availability of the recursive calculation of S_n^θ enables us to develop the online version of the EM algorithm [Cappé, 2009, 2011; Elliott et al., 2002; Mongillo and Deneve, 2008]. This can be done by modifying the forward smoothing recursion by borrowing ideas from stochastic approximation. Let $\theta_{0:n-1}$ denote the parameter estimates obtained sequentially by the online EM algorithm

given the data $y_{1:n-1}$. When y_n is received, we calculate

$$T_{\gamma,n}(x_n) = \int_{\mathcal{X}} \left[(1-\gamma_n)T_{\gamma,n-1}(x_n) + \gamma_n s_n(x_{n-1}, x_n)\right] p^{\epsilon,\kappa}_{\theta_{0:n-1}}(x_{n-1}|x_n, y_{1:n-1})dx_{n-1}$$

$$\mathcal{S}_{\gamma,n} = \int_{\mathcal{X}} T_{\gamma,n}(x_n) p^{\epsilon,\kappa}_{\theta_{0:n-1}}(x_n|y_{1:n})dx_n$$

$$= \frac{\eta_{\theta_{0:n-1},n}\left[T_{\gamma,n} h^{\epsilon,\kappa}_{\theta_{n-1}}(y_n|\cdot)\right]}{\eta_{\theta_{0:n-1},n}\left[h^{\epsilon,\kappa}_{\theta_{n-1}}(y_n|\cdot)\right]}$$

and update $\theta_n = \Lambda(\mathcal{S}_{\gamma,n})$. The subscript $\theta_{0:n-1}$ indicates that the estimations up to time n have contributions to the filtering densities (hence to $T_{\gamma,n}$ and $\mathcal{S}_{\gamma,n}$).

There are both $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ SMC methods available for approximation to S^θ_n and $\mathcal{S}_{\gamma,n}$ for the batch and online cases, respectively. Actually, the $\mathcal{O}(N^2)$ method is analogous to the $\mathcal{O}(N^2)$ method described in Section 6.3.1. Whereas, the $\mathcal{O}(N)$ method is directly based on the path space approximation of $p^{\epsilon,\kappa}_\theta(x_{1:n}|y_{1:n})$ obtained by the SMC filter in Algorithm 6.1. One can see Cappé [2009] for an $\mathcal{O}(N)$ implementation. Although the $\mathcal{O}(N)$ method is computationally less demanding, its estimates have larger Monte Carlo variance compared to those of the $\mathcal{O}(N^2)$ method. Finally, both methods produce stable estimates for θ^* when used in an EM algorithm unlike the gradient ascent algorithm which strictly requires the $\mathcal{O}(N^2)$ method for stability [Poyiadjis et al., 2011].

In the case of i.i.d. processes the EM algorithms simplify in a similar way as in the gradient ascent algorithms, and it will not be detailed here again. The important points are worth emphasising, though: we have $\{X_k = U_k\}$ the SMC implementation to calculate the expectations of sufficient statistics $S_n(U_{1:n}) = \sum_{k=1}^n s_k(U_k)$ breaks into independent Monte Carlo approximations of $\{p^{\epsilon,\kappa}_\theta(u_k|y_k)\}_{k\geq 1}$ and the $\mathcal{O}(N^2)$ algorithm can be implemented with $\mathcal{O}(N)$ calculations. Finally, if needed, the same one-to-one transformation approach explained in Section 6.3.1.3 could be used for the EM algorithm in order to stabilise the sufficient statistics (or their estimates) required for the algorithm.

6.4 Numerical examples

In this section we demonstrate the performance of the methods described in Section 6.3 with several numerical examples. The models we study are sequences of i.i.d. random variables from α-stable and *g-and-k* distributions and the stochastic volatility model with α-stable returns. The experiments focus on different aspects of the methods.

6.4.1 MLE for α-stable distribution

We first consider the problem of estimating the parameter values of a sequence of i.i.d. α-stable random variables. We denote $\mathcal{A}(\alpha, \beta, \mu, \sigma)$ to be the α-stable distribution. The parameters of the distribution,

$$\theta = (\alpha, \beta, \mu, \sigma) \in \Theta = (0, 2] \times [-1, 1] \times \mathbb{R} \times [0, \infty),$$

are the shape, skewness, location, and scale parameters, respectively. Several methods for estimating parameter values for stable distributions have been proposed, including a Bayesian approach based on ABC, see Peters et al. [2011]. In this example we consider estimating these parameters using SN-ABC MLE implemented with the online gradient ascent algorithm.

One can generate a random sample from $\mathcal{A}(\alpha, \beta, \mu, \sigma)$ by generating $U = (U_1, U_2)$, where $U_1 \sim \text{Unif}_{(-\pi/2, \pi/2)}$ and $U_2 \sim \text{Exp}(1)$ independently, and setting

$$Y := t_\theta(U) := \sigma t_{\alpha,\beta}(U) + \mu.$$

The transformation function $t_{\alpha,\beta}$ is defined as [Chambers et al., 1976]

$$t_{\alpha,\beta}(U) = \begin{cases} S_{\alpha,\beta} \frac{\sin\left[\alpha(U_1 + B_{\alpha,\beta})\right]}{[\cos(U_1)]^{1/\alpha}} \left(\frac{\cos\left[U_1 - \alpha(U_1 + B_{\alpha,\beta})\right]}{U_2} \right)^{(1-\alpha)/\alpha}, & \alpha \neq 1 \\ X = \frac{2}{\pi} \left[\left(\frac{\pi}{2} + \beta U_1 \right) \tan U_1 - \beta \log \left(\frac{U_2 \cos U_1}{\frac{\pi}{2} + \beta U_1} \right) \right], & \alpha = 1. \end{cases}$$

where

$$B_{\alpha,\beta} = \frac{\tan^{-1}\left(\beta \tan \frac{\pi\alpha}{2}\right)}{\alpha} \quad S_{\alpha,\beta} = \left(1 + \beta^2 \tan^2 \frac{\pi\alpha}{2}\right)^{1/2\alpha}$$

Since the only discontinuity in the transformation function is at $\alpha = 1$, we can safely use the gradient ascent method for estimating θ^* with the restriction $\alpha \in (0, 1)$ or $\alpha \in (1, 2]$.

As the variance of the α-stable distribution does not exist unless $\alpha = 2$, Monte Carlo estimates of the gradient $\nabla_\theta \log p_\theta^{\epsilon,\kappa}(Y_k^{\epsilon,\kappa})$ are expected to have very high or infinite variance. Instead, we propose using the HMM $\{U_k, Y_k^{\epsilon,\kappa,\psi}\}_{k \geq 1}$ with

$$Y_k^{\epsilon,\kappa,\psi} = \tan^{-1}(Y_k) + \epsilon Z_k, \quad Z_k \sim \mathcal{N}(0, 1), \quad k \geq 1,$$

to make the gradient ascent algorithm stable. For this HMM we have

$$h_\theta^{\epsilon,\kappa,\psi}(y|u) = \mathcal{N}\left(y; \tan^{-1}\left[t_\theta(u)\right], \epsilon^2\right),$$
$$\nabla_\theta \log h_\theta^{\epsilon,\kappa,\psi}(y|u) = \frac{1}{\epsilon^2}\left(y - \tan^{-1}\left[t_\theta(u)\right]\right) \frac{\nabla_\theta t_\theta(u)}{1 + t_\theta(u)^2}.$$

Since $\tan^{-1}(\cdot)$ squeezes the data to a finite interval the variance of $Y^{\epsilon,\kappa,\psi}$ is obviously bounded. The variance of $\nabla_\theta \log h_\theta^{\epsilon,\kappa,\psi}(Y^{\epsilon,\kappa,\psi}|U)$ is not straightforward to evaluate analytically due to the highly non-linear factors from t_θ involved in the expression. However, in order to check whether the transformation stabilises gradients, we can look at the empirical distribution of $\nabla_\theta^N \log p_\theta^{\epsilon,\kappa,\psi}(Y^{\epsilon,\kappa,\psi})$ when $\tan^{-1}(\cdot)$ is used. For this purpose, we generated 10^5 samples $\hat{Y}_i \sim \mathcal{A}(1.5, 0.5, 0, 0.5)$ and $Z_i \sim \mathcal{N}(0,1)$, $i = 1, \ldots, 10^5$, and for each sample we estimated $\nabla_\theta^N \log p_\theta^{\epsilon,\kappa,\psi}(\hat{Y}_i^{\epsilon,\kappa,\psi})$ using $N = 1000$ samples for when $\hat{Y}_i^{\epsilon,\kappa,\psi} = \tan^{-1}(\hat{Y}_i) + \epsilon Z_k$, with $\epsilon = 0.1$. Figure 6.1 shows the histograms of

$$\left\{\nabla_\theta^N \log p_\theta^{\epsilon,\kappa,\psi}(\hat{Y}_i^{\epsilon,\kappa,\psi})\right\}_{1\leq i\leq 10^5}$$

as a numerical approximation to the distribution of $\nabla_\theta^N \log p_\theta^{\epsilon,\kappa,\psi}(Y^{\epsilon,\kappa,\psi})$. From the figure, one can observe that transformation does stabilise the gradients, which is quite important for securing the well behaving of the gradient ascent algorithm.

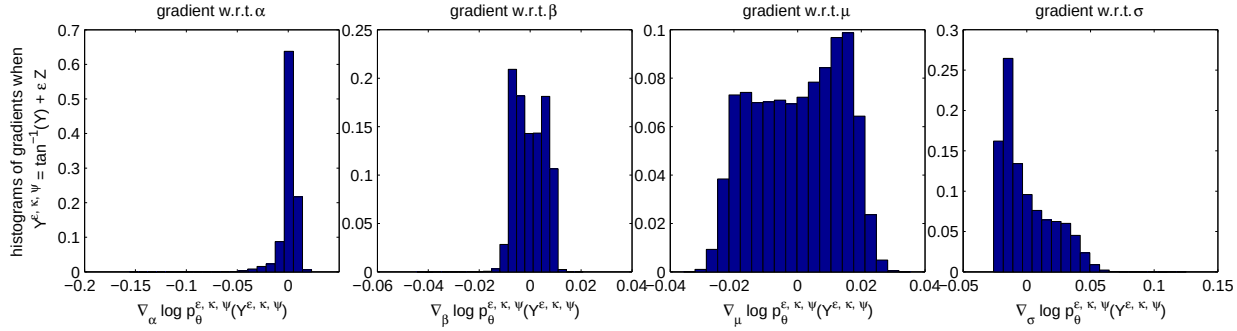


Figure 6.1: Histograms of Monte Carlo estimates of gradients of $\log p_\theta^{\epsilon,\kappa,\psi}(Y^{\epsilon,\kappa,\psi})$ w.r.t. the parameters of the α-stable distribution with $\tan^{-1}(\cdot)$ being used. 10^5 samples were used for generating the histograms.

We implemented the SN-ABC MLE approach with $\epsilon = 0.1$ using the online gradient ascent algorithm to avoid any asymptotic bias in the parameter estimates. Self normalised importance sampling is used in the Monte Carlo approximation part with the proposal density being μ to sample $N = 1000$ particles at each time step. Figure 6.2 shows the online estimation results for θ given a sequence of 10^5 i.i.d. α-stable random variables and stability results for the gradients that are estimated during the algorithm.

In the next experiment we aimed demonstrate how bias is removed from the gradient ascent algorithm by adding noise to data. For this aim we implemented the SN-ABC MLE and S-ABC MLE approaches with $\epsilon = 0.1$ on the same data set of 10^5 samples generated from $\mathcal{A}(1.5, 0.5, 0, 0.5)$ (and transformed with $\tan^{-1}(\cdot)$) . The results in Figure 6.3 are the online estimates averaged over 50 runs for both algorithms. For the SN-ABC MLE algorithm, in each of the 50 runs we added i.i.d. Gaussian noise to the true data set transformed with $\tan^{-1}(\cdot)$, independently from other runs. Figure 6.3 reveals that S-ABC

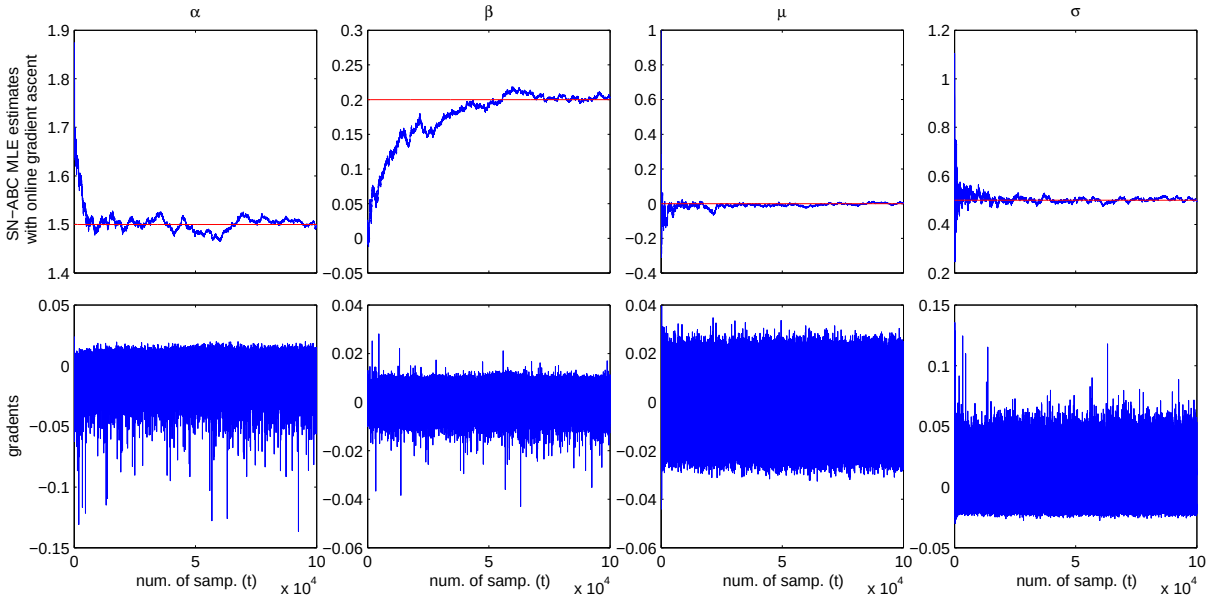


Figure 6.2: On the top: Online estimation of α-stable parameters from a sequence of i.i.d. random variables using online gradient ascent MLE. True parameters $(\alpha, \beta, \mu, \sigma) = (1.5, 0.2, 0, 0.5)$ are indicated with a horizontal line. At the bottom: Gradient of incremental likelihood for the α-stable parameters

MLE introduces biases mainly in the shape and skewness parameters α and β; whereas these biases are removed SN-ABC MLE. As for the scale and location parameters, both algorithms have almost identical mean estimates, which are unbiased.

6.4.2 MLE for *g-and-k* distribution

The *g-and-k* distribution is determined by variables (A, B, g, k, c) and is defined by its quantile function Q_θ, which is the inverse of the cumulative distribution function F_θ

$$Q_\theta(u) = F_\theta^{-1}(u) = A + B\left[1 + c\frac{1 - e^{-g\phi(u)}}{1 + e^{-g\phi(u)}}\right]\left(1 + \phi(u)^2\right)^k \phi(u), \quad u \in (0,1). \tag{6.19}$$

where $\phi(u)$ is the u'th standard normal quantile. The parameters of the distribution

$$\theta = (g, k, A, B) \in \Theta = \mathbb{R} \times (-0.5, \infty) \times \mathbb{R} \times [0, \infty)$$

are the skewness, kurtosis, location, and scale parameters, and c is usually fixed to 0.8. Therefore, one can generate from the *g-and-k* distribution by first sampling $U \sim \text{Unif}_{(0,1)}$ and then returning $t_\theta(u) = Q_\theta(u)$ given $U = u$ (see e.g. Rayner and MacGillivray [2002] for details).

Bayesian parameter estimation for the *g-and-k* distribution using ABC is recently performed in Fearnhead and Prangle [2012], we consider MLE for θ using SN-ABC MLE.

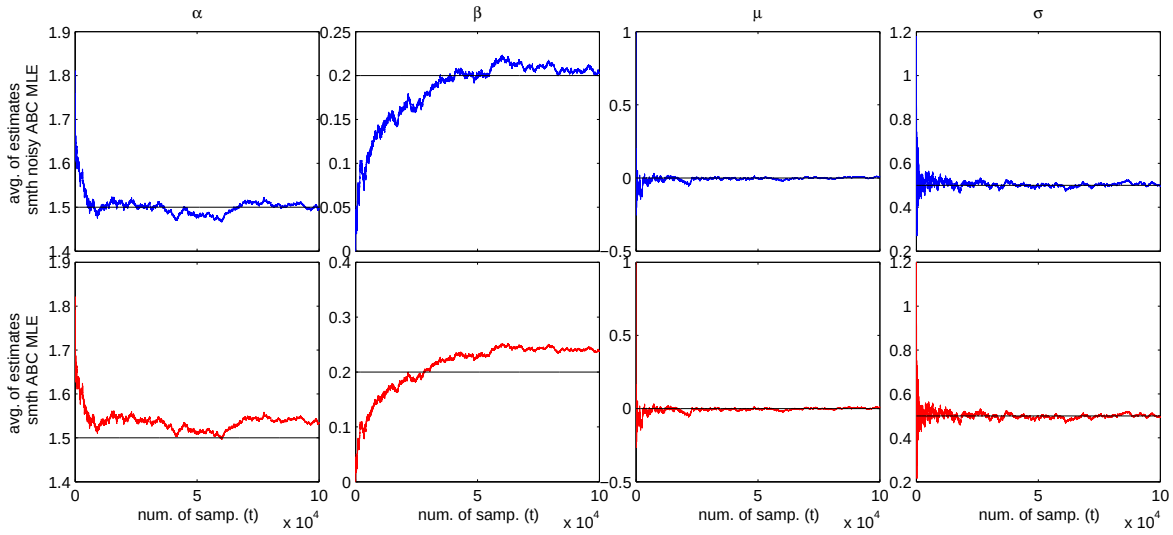


Figure 6.3: S-ABC MLE and SN-ABC MLE estimates of the parameters of the α-stable distribution (averaged over 50 runs) using the online gradient ascent algorithm for the same data set. For SN-ABC MLE, a different noisy data sequence obtained from the original data set is used in each run. True parameters $(\alpha, \beta, \mu, \sigma) = (1.5, 0.2, 0, 0.5)$ are indicated with a horizontal line.

Note that Q_θ in (6.19) is differentiable w.r.t. θ, so the gradient ascent algorithms are applicable. To avoid gradients with very high variances resulting from the factor $(1 + \phi(u)^2)^k$ in Q_θ, similar to the case of α-stable distribution, we use $\psi(\cdot) = \tan^{-1}(\cdot)$ to transform $\hat{Y}_k$ and added noise to $\tan^{-1}(\hat{Y}_k)$ with $\epsilon = 0.1$ to implement SN-ABC MLE with gradient ascent algorithm. Also, during our experiments, we observed that MLE performs better for those distributions whose location parameter A is closer to 0, which must be a result of the non-linear behaviour of the transformation function $\tan^{-1}(\cdot)$. Therefore, whenever possible, it is suggested to estimate the location parameter using a heuristic way (such as looking at the histogram or finding the mean of a first few samples) as a preprocessing step, subtract the heuristically estimated value $\hat{A}$ of A from the samples, perform MLE on the (approximately) centred data, and add back $\hat{A}$ to the estimated location obtained by the MLE algorithm. Figure 6.4 shows the mean and the (log-)variance of SN-ABC MLE estimates of $\theta = (2, 0.5, 10, 2)$ in time which are obtained from 50 runs on the *same* noisy data sequence. Therefore, the accuracy of the mean and the amount of variance correspond to the performance of the Monte Carlo approximation of the gradients $\nabla_\theta^N \log p_\theta^{\epsilon,\kappa,\psi}(y)$. Self normalised importance sampling is used with $N = 1000$ samples generated from μ. From the results shown in the figure, one can deduce that the bias introduced by the finite number of particles is negligible for $N = 1000$ and the variance of the algorithm reduces in time resulting in the convergence of the estimates to the true parameter values.

The next experiment shows how the gradient ascent algorithm can be used in a batch

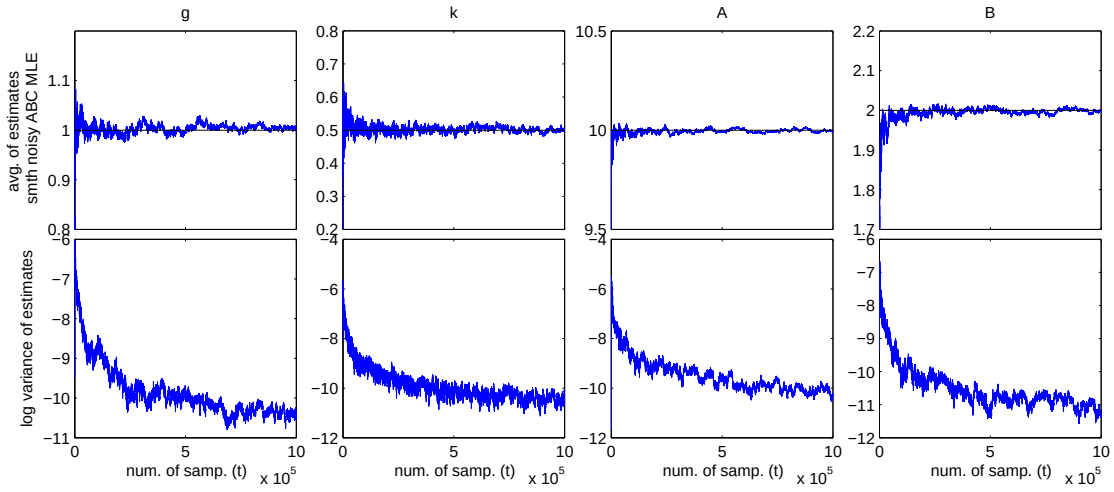


Figure 6.4: Mean and the variance (over 50 runs) of SN-ABC MLE estimates using the online gradient ascent algorithm. Same noisy data sequence is used in each run. True parameters $(g, k, A, B) = (2, 0.5, 10, 2)$ are indicated with a horizontal line.

setting when the data set is too small for the online algorithm to converge. We implemented the batch gradient ascent SN-ABC MLE algorithm on data sets of $n = 1000$ i.i.d. samples from the same *g-and-k* distribution, that is, for $\theta = (2, 0.5, 10, 2)$. A detailed study of the MLE for *g-and-k* distribution can be found in Rayner and MacGillivray [2002] where the MLE methods based on numerical approximation of the likelihood itself are investigated; here we present the results of an alternative numerical method to compute the MLE which is not included in their work. We generated 500 data sets of size $n = 1000$ and performed batch gradient ascent algorithm for SN-ABC MLE with $\epsilon = 0.1$ for each data set. Again, the same self normalised importance sampling procedure is used with $N = 1000$ samples. The upper half of Figure 6.5 shows the estimation results versus number of iterations on a single data set. It can be seen that 1000 iterations are sufficient for the convergence of the gradient ascent algorithm. Note that for short data sets such as those with size 1000, MLE may have a considerable variance as the estimates out of the single data set reveal. The lower half of Figure 6.5 shows the (approximate) distributions (histograms over 20 bins) of the MLE estimate for θ. The mean and variance of the MLE estimates for (g, k, A, B) are $(2.004, 0.503, 9.995, 1.996)$ and $(0.0151, 0.0021, 0.0052, 0.0213)$ respectively. These moments of the MLE for this particular θ and same data size n are also obtained in Rayner and MacGillivray [2002] (see Table 3); the results are comparable. Also, note that this is not the limit of our algorithm; the contribution of the Monte Carlo approximation to bias and variance can be reduced further by increasing the number of particles N, or the Monte Carlo bias can even be removed, such as by using MCMC instead of self normalised importance sampling.

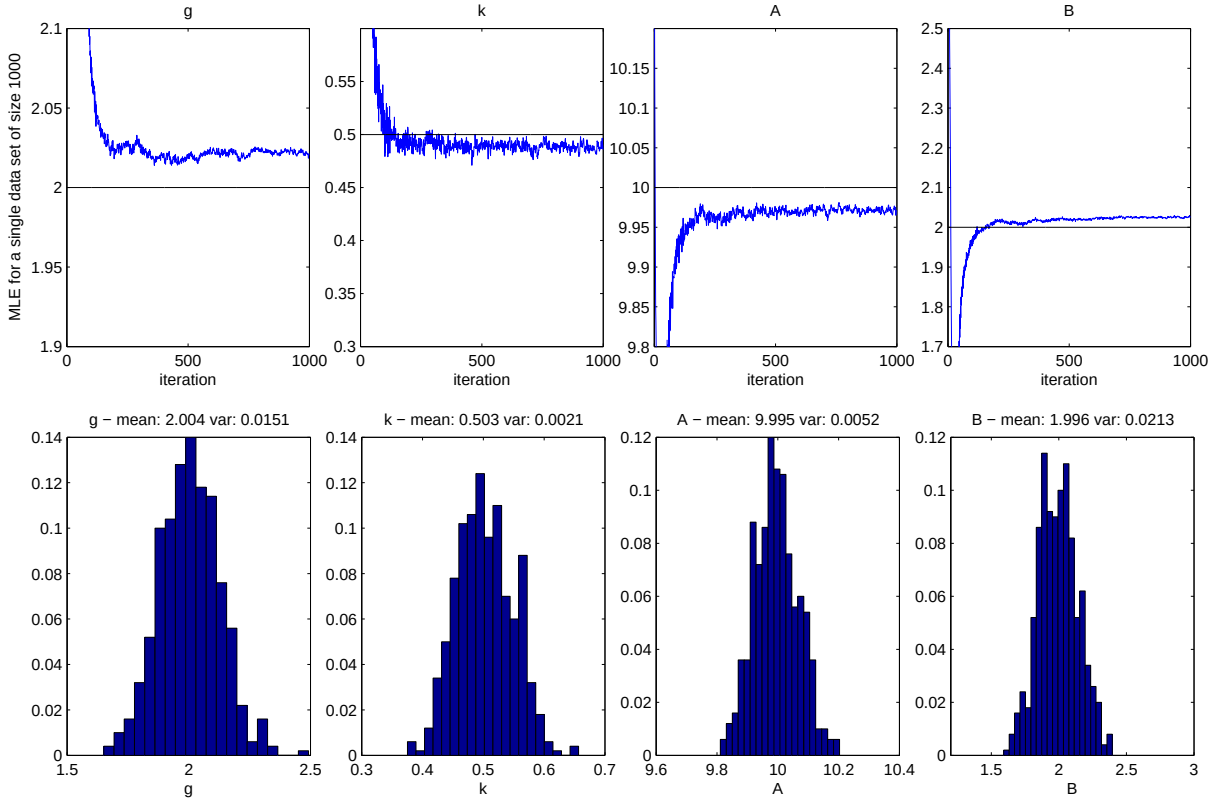


Figure 6.5: Top: SN-ABC MLE estimates of *g-and-k* parameters from a sequence of i.i.d. random variables using the batch gradient ascent algorithm. True parameters $(g, k, A, B) = (2, 0.5, 10, 2)$ are indicated with a horizontal line. Bottom: Approximate distributions (histograms over 20 bins) of the estimates

6.4.3 The stochastic volatility model with symmetric α-stable returns

The stochastic volatility model with α-stable returns (SVαSR) is a model used in analysing economical data. The hidden process $\{R_k \in \mathbb{R}\}_{k\geq 1}$ represents the log-volatility in time whereas the observation process $\{Y_k \in \mathbb{R}\}_{k\geq 1}$ shows the return values. The model for $\{R_k, Y_k\}_{k\geq 1}$ is:

$$\begin{aligned} &R_1 \sim \mathcal{N}\left(0, \sigma_x^2/(1-\phi^2)\right), \quad R_k = \phi R_{k-1} + \sigma_x V_k, \quad V_k, \sim \mathcal{N}(0,1), \quad k \geq 2, \\ &Y_k \sim e^{R_k/2}\mathcal{A}(\alpha, 0, 0, 1), \quad k \geq 1. \end{aligned} \tag{6.20}$$

The model is an alternative of the stochastic volatility model with Gaussian returns as observed series tend to be heavy-tailed and display discontinuities. For more discussion on the model as well as a review of methods for estimating the static parameters of such models, see Lombardi and Calzolari [2009] and the references therein. Those existing methods for parameter estimation in SVαSR, however, are batch and suitable for only short data sequences. We test our online algorithms implementing SN-ABC MLE for

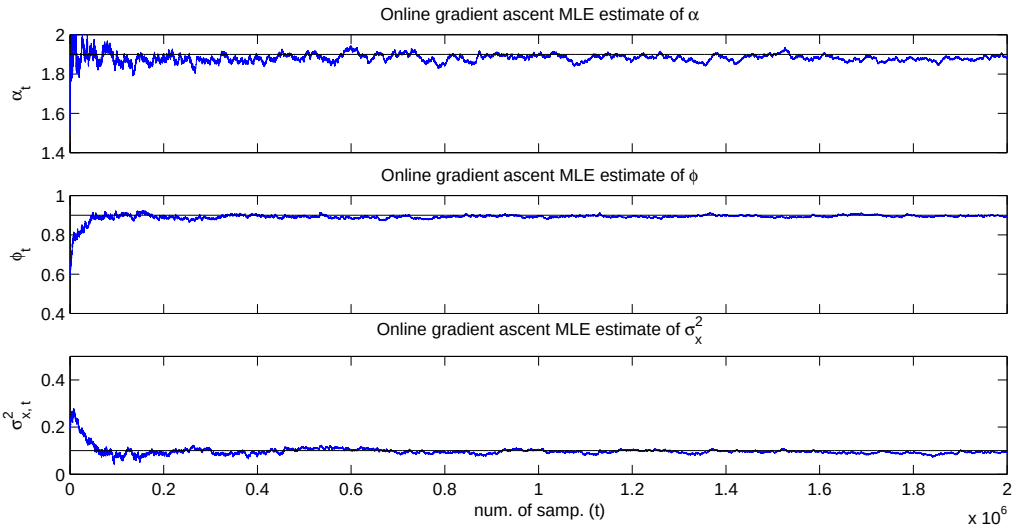


Figure 6.6: Online estimation of SVαR parameters using online gradient ascent algorithm to implement SN-ABC MLE. True parameter values $(\alpha, \phi, \sigma_x^2) = (1.9, 0.9, 0.1)$ are indicated with a horizontal line.

this model in a scenario where a very long data sequence is given (or is being received sequentially).

Since the likelihood involves the α-stable distribution, for stability of the gradient ascent algorithm we add noise to the $\tan^{-1}(\cdot)$ of $\hat{Y}_k$ to have $\hat{Y}_k^{\epsilon,\kappa,\psi} = \tan^{-1}(\hat{Y}_k) + \epsilon Z_k, \quad Z_k \sim \mathcal{N}(0,1)$. The densities π_θ, q_θ, and $h_\theta^{\epsilon,\kappa,\psi}$ corresponding to the HMM $\{X_k = (R_k, U_k), Y_k^{\epsilon,\kappa,\psi}\}_{k\geq 1}$ with $U_k = (U_{k,1}, U_{k,2})$ are as follows:

$$\begin{aligned}
\pi_\theta(x) &= \mathcal{N}\left(r; 0, \sigma_x^2/(1-\phi^2)\right) \frac{1}{\pi} \mathbb{I}_{[-\pi/2,\pi/2]}(u_1) \mathbb{I}_{[0,\infty)}(v) e^{-u_2}, \\
q_\theta(x'|x) &= \mathcal{N}(r'; \phi r, \sigma_x^2) \frac{1}{\pi} \mathbb{I}_{[-\pi/2,\pi/2]}(u_1') \mathbb{I}_{[0,\infty)}(u_2') e^{-u_2'}, \\
h_\theta^{\epsilon,\kappa,\psi}(y|x) &= \mathcal{N}\left(y; \tan^{-1}\left[e^{r/2} t_{\alpha,0}(u)\right], \epsilon^2\right),
\end{aligned}$$

where $x = (r, u)$ and $x' = (r', u')$ and $u = (u_1, u_2)$.

Estimates of $\theta = (\alpha, \phi, \sigma_x^2)$ obtained with the online gradient ascent implementation of the SN-ABC MLE described in Section 6.3.1 using $N = 500$ particles for a data sequence of 2×10^6 samples is shown in Figure 6.6. $\theta^* = (1.9, 0.9, 0.1)$ was used for generating the data. The estimates seem to converge after around 5×10^5 samples. We also implemented the online EM algorithm to perform noisy smoothed ABC MLE on the same data. Note that the maximisation step for α is not feasible, that is why the EM algorithm is restricted to estimate only the hidden state parameters, assuming α is known. The sufficient statistics needed to estimate σ_x^2 and ϕ are provided in Del Moral et al. [2009]. The online EM results for the model are shown in Figure 6.7.

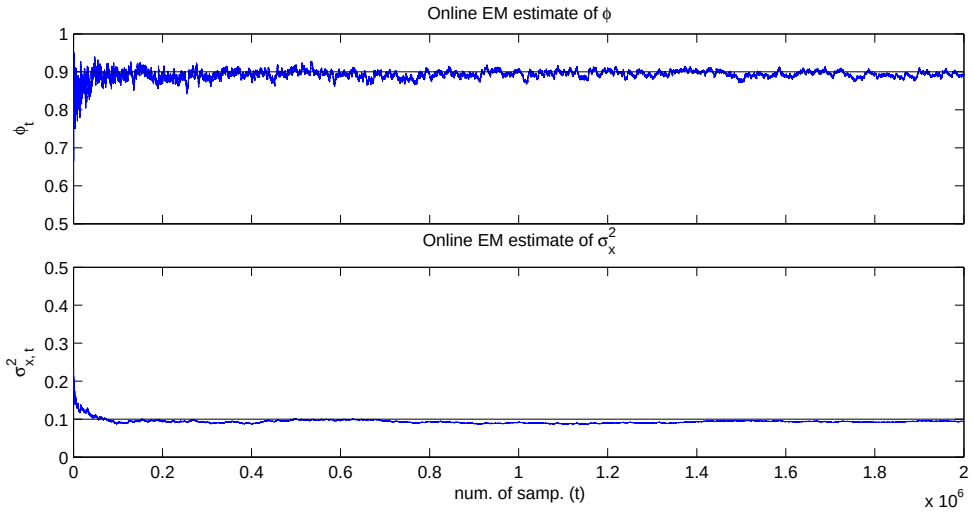


Figure 6.7: Online estimation of SVαR parameters ($\alpha = 1.9$ is known) using the online EM algorithm to implement SN-ABC MLE. True parameter values $(\phi, \sigma_x^2) = (0.9, 0.1)$ are indicated with a horizontal line.

6.5 Discussion

In this chapter, we presented a novel methodology for implementing MLE in HMMs with intractable likelihoods in the context of ABC. We showed how both batch and online versions of gradient ascent and EM algorithms can be used for those HMMs by using the ABC approach to confront the intractability. We also demonstrated how to implement noisy ABC ideas to get rid of an asymptotic (in size of data) ABC bias in our estimates. As also suggested by the examples that we cover, the gradient ascent algorithm is applicable to more cases than the EM. This is not surprising, though; since intractability mostly arises from the non-linear characteristics of t_θ, and in general we do not expect to be able to find sufficient statistics for the parameters involved in t_θ.

Note that gradient ascent and EM are not the only possible methods to implement ABC MLE, although we only covered them due to their similarity and popularity for the practitioner. Once, we can construct the expanded HMM with tractable transitional laws, we can use potentially any other MLE method that works for HMMs. One such example is the iterated filtering algorithm [Ionides et al., 2011] which can be useful for HMMs having non-linear state space dynamics.

Chapter 7

An Online Expectation-Maximisation Algorithm for Nonnegative Matrix Factorisation Models

Summary: *In this chapter we formulate the nonnegative matrix factorisation (NMF) problem as a maximum likelihood estimation problem for hidden Markov models and propose online expectation-maximisation (EM) algorithms to estimate the NMF and the other unknown static parameters. We also propose a sequential Monte Carlo approximation of our online EM algorithm. We show the performance of the proposed method with two numerical examples.*

The work done in this chapter is published in Yıldırım et al. [2012a]. This idea for this chapter was initiated during a discussion between Dr. Taylan Cemgil and myself.

7.1 Introduction

With the advancement of sensor and storage technologies, and with the cost of data acquisition dropping significantly, we are able to collect and record vast amounts of raw data. Arguably, the grand challenge facing computation in the 21st century is the effective handling of such large data sets to extract meaningful information for scientific, financial, political or technological purposes [Donoho, 2000]. Unfortunately, classical batch processing methods are unable to deal with very large data sets due to memory restrictions and slow computational time.

One key approach for the analysis of large datasets is based on the matrix and tensor factorisation paradigm. Given an observed dataset Y, where Y is a matrix of a certain dimension and each element of it corresponds to an observed data point, the matrix factorisation problem is the computation of matrix factors B and X such that Y is

approximated by the matrix product BX, i.e.,

$$Y \approx BX.$$

(Later we will make our notation and inferential goals more precise.) Indeed, many standard statistical methods such as clustering, independent components analysis, non-negative matrix factorisation (NMF), latent semantic indexing, collaborative filtering can be expressed and understood as matrix factorisation problems [Koren et al., 2009; Lee and Seung, 1999; Singh and Gordon, 2008].

Matrix factorisation models also have well understood probabilistic/statistical interpretations as probabilistic generative models and many standard algorithms mentioned above can also be derived as maximum likelihood or maximum a-posteriori parameter estimation procedures [Cemgil, 2009; Févotte and Cemgil, 2009; Salakhutdinov and Mnih, 2008]. The advantage of this interpretation is that it enables one to incorporate domain specific prior knowledge in a principled and consistent way. This can be achieved by building hierarchical statistical models to fit the specifics of the application at hand. Moreover, the probabilistic/statistical approach also provides a natural framework for sequential processing which is desirable for developing online algorithms that pass over each data point only once. While the development of effective online algorithms for matrix factorisation are of interest on their own, the algorithmic ideas can be generalised to more structured models such as tensor factorisations (e.g. see Kolda and Bader [2009]).

In this work our primary interest is estimation of B (rather than B and X), which often is the main objective in NMF problems. We formulate the NMF problem as a maximum likelihood estimation (MLE) problem for hidden Markov models (HMMs). The advantage of doing so is that the asymptotic properties of MLE for HMM's has been studied in the past by many authors and these results may be adapted to the NMF framework. We propose a sequential Monte Carlo (SMC) based online EM algorithm [Cappé, 2009; Del Moral et al., 2009] for the NMF problem. SMC introduces a layer of bias which decreases as the number of particles in the SMC approximation is increased.

In the literature, several online algorithms have been proposed for online computation of matrix factorisations. Mairal et al. [2010] propose an online optimisation algorithm, based on stochastic approximations, which scales up gracefully to large data sets with millions of training samples. A proof of convergence is presented for the Gaussian case. There are similar formulations applied to other matrix factorisation formulations, notably NMF [Lefevre et al., 2011] and Latent Dirichlet Allocation [Hoffman et al., 2010], as well as alternative views for NMF which are based on incremental subspace learning [Bucak and Gunsel, 2009]. Although the empirical results of these methods suggest good performance, their asymptotic properties have not been established.

7.1.1 Notation

Let A be a $M \times N$ matrix. The (m,n)'th element of A is $A(m,n)$. If M (or N) is 1, then $A(i) = A(1,i)$ (or $A(i,1)$). The m'th row of A is $A(m,\cdot)$. If A and B are both $M \times N$ matrices, $C = A \odot B$ denotes element-by-element multiplication, i.e., $C(m,n) = A(m,n)B(m,n)$; $\frac{A}{B}$ (or A/B) means element-by-element division, in a similar way. $\mathbf{1}_{M\times N}$ ($\mathbf{0}_{M\times N}$) is a $M \times N$ matrix of 1's (0's), where $\mathbf{1}_{M\times 1}$ is abbreviated to $\mathbf{1}_M$. $\mathbb{N} = \{0,1,2,\ldots\}$ and $\mathbb{R}_+ = [0,\infty)$ are the sets of nonnegative integers and real numbers. Random variables will be defined by using capital letters, such as X, Y, Z, etc., and their realisations will be corresponding small case letters (x, y, z, etc.). The indicator function $I_\alpha(x) = 1$ if $x = \alpha$, otherwise it is 0; also, for a set A, $I_A(x) = 1$ if $x \in A$, otherwise it is 0.

7.2 The Statistical Model for NMF

Consider the following HMM comprised of the latent processes $\{X_t, Z_t\}_{t\geq 1}$ and the observation process $\{Y_t\}_{t\geq 1}$. The process $\left\{X_t \in \mathbb{R}_+^K\right\}_{t\geq 1}$ is a Markov process of $K \times 1$ nonnegative vectors with an initial density μ_ψ and the transition density f_ψ for $t = 2, 3, \ldots$

$$X_1 \sim \mu_\psi(x),\ X_t|\left(X_{t-1} = x_{t-1}\right) \sim f_\psi(x_t|x_{t-1}), \tag{7.1}$$

where $\psi \in \Psi$ is a finite dimensional parameter which parametrizes the law of the Markov process. $Z_t \in \mathbb{N}^{M\times K}$ is a $M \times K$ matrix of nonnegative integers, and its elements are independent conditioned on X_t as follows:

$$Z_t|\left(X_t = x_t\right) \sim \prod_{m=1}^{M}\prod_{k=1}^{K} \mathcal{PO}(z_t(m,k); B(m,k)x_t(k))$$

where $B \in \mathbb{R}_+^{M\times K}$ is an $M \times K$ nonnegative matrix. Here $\mathcal{PO}(v;\lambda)$ denotes the Poisson distribution on $\mathbb{N}$ with intensity parameter $\lambda \geq 0$

$$\mathcal{PO}(v;\lambda) = \exp\left(v\log\lambda - \lambda - \log v!\right),$$

The $M \times 1$ observation vector Y_t is conditioned on Z_t in a deterministic way

$$Y_t(m) = \sum_{k=1}^{K} Z_t(m,k), \quad m = 1,\ldots,M.$$

This results in the conditional density of Y_t given $X_t = x_t$, denoted by g_B, being a product of Poisson densities

$$Y_t | \left(X_t = x_t\right) \sim g_B(y_t|x_t) = \prod_{m=1}^{M} \mathcal{PO}\left(y_t(m); B(m,\cdot)x_t\right). \tag{7.2}$$

Hence the likelihood of y_t given x_t can analytically be evaluated. Moreover, the conditional posterior distribution $\pi_B(z_t|y_t, x_t)$ of Z_t given y_t and x_t has a factorized closed form expression:

$$\begin{aligned} Z_t | \left(Y_t = y_t, X_t = x_t\right) &\sim \pi_B(z_t|y_t, x_t) \\ &= \prod_{m=1}^{M} \mathcal{M}\left(z_t(m,\cdot); y_t(m), \rho_{t,m}\right) \end{aligned} \tag{7.3}$$

where $\rho_{t,m}(k) = B(m,k)x_t(k)/B(m,\cdot)x_t$ and $\mathcal{M}$ denotes a multinomial distribution defined by

$$\mathcal{M}(v; \alpha, \rho) = I_\alpha \left(\sum_{k=1}^{K} v_k\right) \alpha! \prod_{k=1}^{K} \frac{\rho_k^{v_k}}{v_k!},$$

where $v = [v_1 \dots v_K]$ is a realisation of the vector valued random variable $V = [V_1 \dots V_K]$, $\rho = (\rho_1, \dots, \rho_K)$, and $\sum_{k=1}^{K} \rho_k = 1$. It is a standard result that the marginal mean of the k'th component is $\mathbb{E}_{\alpha,\rho}\left[V_k\right] = \alpha\rho_k$.

Let $\theta = (\psi, B) \in \Theta = \Psi \times \mathbb{R}_+^{M\times K}$ denote all the parameters of the HMM. We can write the joint density of $(X_{1:t}, Z_{1:t}, Y_{1:t})$ given θ as

$$p_\theta(x_{1:t}, z_{1:t}, y_{1:t}) = \mu_\psi(x_1) g_B(y_1|x_1) \pi_B(z_1|y_1, x_1) \prod_{i=2}^{t} f_\psi(x_i|x_{i-1}) g_B(y_i|x_i) \pi_B(z_i|x_i, y_i). \tag{7.4}$$

From (7.4), we observe that the joint density of $(X_{1:t}, Y_{1:t})$

$$p_\theta(x_{1:t}, y_{1:t}) = \mu_\psi(x_1) g_B(y_1|x_1) \prod_{i=2}^{t} f_\psi(x_i|x_{i-1}) g_B(y_i|x_i)$$

defines the law of another HMM $\{X_t, Y_t\}_{t\geq 1}$ comprised of the latent process $\{X_t\}_{t\geq 1}$, with initial and transitional densities μ_ψ and f_ψ, and the observation process $\{Y_t\}_{t\geq 1}$ with the observation density g_B. Finally, the likelihood of data is given by

$$p_\theta(y_{1:T}) = \mathbb{E}_\psi \left[\prod_{t=1}^{T} g_B(y_t|X_t)\right]. \tag{7.5}$$

In this work, we treat θ as unknown and seek for the MLE solution θ^* for it, which satisfies

$$\theta^* = \arg\max_{\theta\in\Theta} p_\theta(y_{1:T}). \tag{7.6}$$

7.2.1 Relation to the classical NMF

In the classical NMF formulation [Lee and Seung, 1999, 2000], given a $M \times T$ nonnegative matrix $Y = [y_1 \dots y_T]$, we want to factorize it to $M \times K$ and $K \times T$ nonnegative matrices B and $X = [X_1 \dots X_T]$ such that the difference between Y and BX is minimised according to a divergence

$$(B^*, X^*) = \arg\min_{B,X} D(Y||BX). \tag{7.7}$$

One particular choice for D is the generalised Kullback-Leibler (KL) divergence which is written as

$$D(Y||U) = \sum_{m=1}^{M}\sum_{t=1}^{T} Y(m,t)\log\frac{Y(m,t)}{U(m,t)} - Y(m,t) + U(m,t)$$

Noticing the similarity between the generalised KL divergence and the Poisson distribution, [Lee and Seung, 1999] showed that the minimisation problem can be formulated in a MLE sense. More explicitly, the solution to

$$\begin{aligned}(B^*, X^*) &= \arg\max_{B,X} \ell(y_1, \dots, y_T|B, X),\\ \ell(y_1, \dots, y_T|B, X) &= \textstyle\prod_{t=1}^{T} g_B\left(y_t|X_t\right)\end{aligned} \tag{7.8}$$

is the same as the solution to (7.7). In our formulation of the NMF problem, $X = [X_1 \dots X_T]$ is not a static parameter but it is a random matrix whose columns constitute a Markov process. Therefore, the formulation for MLE in our case changes to maximising the expected value of the likelihood in (7.8) over the parameter $\theta = (B, \psi)$ with respect to (w.r.t.) the law of X

$$(B^*, \psi^*) = \arg\max_{(B,\psi)\in\Theta} \mathbb{E}_\psi\left[\ell(y_1, \dots, y_T|B, X)\right]. \tag{7.9}$$

It is obvious that (7.6) and (7.9) are equivalent. We will see in Section 7.3 that the introduction of the additional process $\{Z_t\}_{t\geq 1}$ is necessary to perform MLE using the EM algorithm (see Lee and Seung [2000] for its first use for the problem stated in (7.7)).

7.3 EM algorithms for NMF

Our objective is to estimate the unknown θ given $Y_{1:T} = y_{1:T}$. The EM algorithm can be used to find the MLE for θ. We first introduce the batch EM algorithm and then explain how an online EM version can be obtained.

7.3.1 Batch EM

With the EM algorithm, given the observation sequence $y_{1:T}$ we increase the likelihood $p_\theta(y_{1:T})$ in (7.5) iteratively until we reach a maximal point on the surface of the likelihood. The algorithm is as follows:

Choose $\theta^{(0)}$ for initialisation. At iteration $j = 0, 1, \ldots$

- **E-step:** Calculate the intermediate function which is the expectation of the log joint distribution of $(X_{1:T}, Z_{1:T}, Y_{1:T})$ with respect to the law of $(X_{1:T}, Z_{1:T})$ given $Y_{1:T} = y_{1:T}$.

$$Q(\theta^{(j)}; \theta) = \mathbb{E}_{\theta^{(j)}} \left[\log p_\theta(X_{1:T}, Z_{1:T}, Y_{1:T}) \middle| Y_{1:T} = y_{1:T} \right]$$

- **M-step:** The new estimate is the maximiser of the intermediate function

$$\theta^{(j+1)} = \arg\max_\theta Q(\theta^{(j)}; \theta)$$

With a slight modification of the update rules found in Cemgil [2009, Section 2], one can show that for NMF models the update rule for B reduces to calculating the expectations

$$\widehat{S}_{1,T} = \mathbb{E}_{\theta^{(j)}} \left[\sum_{t=1}^{T} X_t \middle| Y_{1:T} = y_{1:T} \right], \quad \widehat{S}_{2,T} = \mathbb{E}_{\theta^{(j)}} \left[\sum_{t=1}^{T} Z_t \middle| Y_{1:T} = y_{1:T} \right]$$

and updating the parameter estimate for B as

$$B^{(j+1)} = \widehat{S}_{2,T} / \left(\mathbf{1}_M \left[\widehat{S}_{1,T} \right]^T \right).$$

Moreover, if the transition density f_ψ belongs to an exponential family, the update rule for ψ becomes calculating the expectation of a $J \times 1$ vector valued function

$$\widehat{S}_{3,T} = \mathbb{E}_{\theta^{(j)}} \left[\sum_{t=1}^{T} s_{3,t}(X_{t-1}, X_t) \middle| Y_{1:T} = y_{1:T} \right]$$

and updating the estimate for ψ using a maximisation rule

$$\Lambda : \mathbb{R}^J \to \Psi, \quad \psi^{(j+1)} = \Lambda\left(\widehat{S}_{3,T}\right).$$

Note that $s_{3,t}$ and Λ depend on the NMF model, particularly to the probability laws in (7.1) defining the Markov chain for $\{X_t\}_{t\geq 1}$. Therefore, we have to find the mean estimates of the following sufficient statistics at time t.

$$S_{1,t}(x_{1:t}) = \sum_{i=1}^{t} x_i, \quad S_{2,t}(z_{1:t}) = \sum_{i=1}^{t} z_i, \quad S_{3,t}(x_{1:t}) = \sum_{i=1}^{t} s_{3,t}(x_{t-1}, x_t). \tag{7.10}$$

Writing the sufficient statistics in additive forms as in (7.10) enables us to use a forward recursion to find the expectations of the sufficient statistics in an online manner. This leads to an online version of the EM algorithm as we shall see in the following section.

7.3.2 Online EM

To explain the methodology in a general sense, assume that we want to calculate the expectations $\widehat{S}_t = \mathbb{E}_\theta\left[S_t(X_{1:t}, Z_{1:t})\middle| Y_{1:t} = y_{1:t}\right]$ of sufficient statistics of the additive form

$$S_t(x_{1:t}, z_{1:t}) = \sum_{i=1}^{t} s_i(x_{i-1}, z_{i-1}, x_i, z_i) \tag{7.11}$$

w.r.t. the posterior density $p_\theta(x_{1:t}, z_{1:t}|y_{1:t})$ for a given parameter value B. Letting $u_t = (x_t, z_t)$ for simplicity, we define the intermediate function

$$T_t(u_t) = \int S_t(u_{1:t}) p_\theta(u_{1:t-1}|y_{1:t-1}, u_t) du_{1:t-1}.$$

One can show that we have the forward recursion [Cappé, 2011; Del Moral et al., 2009]

$$T_t(u_t) = \int \left[T_{t-1}(u_{t-1}) + s_t(u_{t-1}, u_t)\right] p_\theta(u_{t-1}|y_{1:t-1}, u_t) du_{t-1} \tag{7.12}$$

with the convention $T_0(u) = 0$. Hence, T_t can be computed online, so are the estimates

$$\widehat{S}_t = \int T_t(u_t) p_\theta(u_t|y_{1:t}) du_t.$$

We can decompose the backward transition density $p_\theta(u_{t-1}|y_{1:t-1}, u_t)$ and the filtering density $p_\theta(u_t|y_{1:t})$ as

$$p_\theta(x_{t-1}, z_{t-1}|y_{1:t-1}, x_t, z_t) = \pi_B(z_{t-1}|x_{t-1}, y_{t-1})p_\theta(x_{t-1}|x_t, y_{1:t-1}), \tag{7.13}$$

$$p_\theta(x_t, z_t|y_{1:t}) = \pi_B(z_t|x_t, y_t)p_\theta(x_t|y_{1:t}) \tag{7.14}$$

where π_B is defined in (7.3). From (7.10) we know that the required sufficient statistics are additive in the required form; therefore, the recursion in (7.12) is possible for the NMF model. The recursion for $S_{3,t}$ depends on the choice of the transition density f_ψ; however the recursions for $S_{1,t}$ and $S_{2,t}$ are the same for any model regardless of the choice of f_ψ. For this reason, we shall have a detailed look at (7.12) for the first two sufficient statistics $S_{1,t}$ and $S_{2,t}$.

For $S_{1,t}$, notice from (7.13) that, $p_\theta(x_{t-1}, z_{t-1}|y_{1:t-1}, x_t, z_t)$ does not depend on z_t. Moreover, the sufficient statistic $S_{1,t}$ is not a function of $z_{1:t}$. Therefore, z_{t-1} in (7.12) integrates out, and $T_{1,t}$ is a function of x_t only. Hence we will write it as $T_{1,t}(x_t)$. To sum up, we have the recursion

$$T_{1,t}(x_t) = x_t + \int T_{1,t-1}(x_{t-1})p_\theta(x_{t-1}|x_t, y_{1:t-1})dx_{t-1}.$$

For $S_{2,t}$, we claim that $T_{2,t}(x_t, z_t) = z_t + C_t(x_t)$ where $C_t(x_t)$ is a nonnegative $M \times K$ matrix valued function depending on x_t but not z_t, and the recursion for $C_t(x_t)$ is expressed as

$$C_t(x_t) = \int \left[C_{t-1}(x_{t-1}) + \frac{B \odot \left(y_{t-1}x_{t-1}^T\right)}{(Bx_{t-1})\,\mathbf{1}_K^T} \right] p_\theta(x_{t-1}|x_t, y_{1:t-1})dx_{t-1}$$

This claim can be verified by induction. Start with $t = 1$. Since $T_{2,0} = \mathbf{0}_{M\times K}$, we immediately see that $T_{2,t}(x_1, z_1) = z_1 = z_1 + C_1(x_1)$ where $C_1(x_1) = \mathbf{0}_{M\times K}$. For general $t > 1$, assume that $T_{2,t-1}(x_{t-1}, z_{t-1}) = z_{t-1} + C_{t-1}(x_{t-1})$. Using (7.13),

$$T_{2,t}(x_t, z_t) = z_t + \int \left[z_{t-1} + C_{t-1}(x_{t-1})\right] \pi_B(z_{t-1}|x_{t-1}, y_{t-1})p_\theta(x_{t-1}|x_t, y_{1:t-1})dx_{t-1}dz_{t-1}$$

Now, observe that the (m, k)'th element of the integral $\int z_{t-1}\pi_B(z_{t-1}|x_{t-1}, y_{t-1})dz_{t-1}$ is $\frac{B(m,k)y_{t-1}(m)x_{t-1}(k)}{B(m,\cdot)x_{t-1}}$. So, we can write the integral as

$$\int z_{t-1}\pi_B(z_{t-1}|x_{t-1}, y_{t-1})dz_{t-1} = \frac{B \odot \left(y_{t-1}x_{t-1}^T\right)}{(Bx_{t-1})\,\mathbf{1}_K^T}$$

So we are done. Using a similar derivation and substituting (7.14) into (7.13), we can

show that

$$\widehat{S}_{2,t} = \int \left(C_t(x_t) + \frac{B \odot \left(y_t x_t^T\right)}{(Bx_t)\,\mathbf{1}_K^T} \right) p_\theta(x_t|y_{1:t}) dx_t.$$

The online EM algorithm is a variation over the batch EM where the parameter is re-estimated each time a new observation is received. In this approach running averages of the sufficient statistics are computed [Cappé, 2009, 2011; Elliott et al., 2002; Mongillo and Deneve, 2008], [Kantas et al., 2009, Section 3.2.]. Specifically, let $\gamma = \{\gamma_t\}_{t\geq 1}$, called the step-size sequence, be a positive decreasing sequence satisfying $\sum_{t\geq 1} \gamma_t = \infty$ and $\sum_{t\geq 1} \gamma_t^2 < \infty$. A common choice is $\gamma_t = t^{-a}$ for $0.5 < a \leq 1$. Let θ_1 be the initial guess of θ^* before having made any observations and at time t, let $\theta_{1:t}$ be the sequence of parameter estimates of the online EM algorithm computed sequentially based on $y_{1:t-1}$. Letting $u_t = (x_t, z_t)$ again to show for the general case, when y_t is received, online EM computes

$$T_{\gamma,t}(u_t) = \int \left[(1-\gamma_t)\, T_{\gamma,t-1}(u_{t-1}) + \gamma_t s_t(u_{t-1}, u_t)\right] p_{\theta_{1:t}}(u_{t-1}|y_{1:t-1}, u_t) du_{t-1}, \tag{7.15}$$

$$\mathcal{S}_t = \int T_{\gamma,t}(u_t) p_{\theta_{1:t}}(u_t|y_{1:t}) du_t \tag{7.16}$$

and then applies the maximisation rule using the estimates $\mathcal{S}_t$. The subscript $\theta_{1:t}$ on the densities $p_{\theta_{1:t}}(u_{t-1}|y_{1:t-1}, u_t)$ and $p_{\theta_{1:t}}(u_t|y_{1:t})$ indicates that these laws are being computed sequentially using the parameter θ_k at time k, $k \leq t$. (See Algorithm 7.1 for details.) In practice, the maximisation step is not executed until a burn-in time t_b for added stability of the estimators as discussed in Cappé [2009].

The online EM algorithm can be implemented exactly for a linear Gaussian state-space model [Elliott et al., 2002] and for finite state-space HMM's. [Cappé, 2011; Mongillo and Deneve, 2008]. An exact implementation is not possible for NMF models in general, therefore we now investigate SMC implementations of the online EM algorithm.

7.3.3 SMC implementation of the online EM algorithm

Recall that $\{X_t, Y_t\}_{t\geq 1}$ is also a HMM with the initial and transition densities μ_ψ and f_ψ in (7.1), and the observation density g_B in (7.2). Since the conditional density $\pi_B(z_t|x_t, y_t)$ has a close form expression, it is sufficient to have a particle approximation to only $p_\theta(x_{1:t}|y_{1:t})$. This approximation can be performed in an online manner using a SMC approach. Suppose that we have the particle approximation to $p_\theta(x_{1:t}|y_{1:t})$ at time t with

N particles

$$p_\theta^N(dx_{1:t}|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{1:t}^{(i)}}(dx_{1:t}), \quad \sum_{i=1}^{N} w_t^{(i)} = 1, \tag{7.17}$$

where $x_{1:t}^{(i)} = (x_1^{(i)}, \ldots, x_t^{(i)})$ is the n'th path particle with weight $w_t^{(i)}$ and δ_x is the dirac measure concentrated at x. The particle approximation of the filter at time t can be obtained from $p_\theta^N(dx_{1:t}|y_{1:t})$ by marginalization

$$p_\theta^N(dx_t|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t^{(i)}}(dx_t).$$

At time $t+1$, for each n we draw $x_{t+1}^{(i)}$ from a proposal density $q_\theta(x_{t+1}|x_t^{(i)})$ with a possible implicit dependency on y_{t+1}. We then update the weights according to the recursive rule:

$$w_{t+1}^{(i)} \propto \frac{w_t^{(i)} f_\psi(x_{t+1}^{(i)}|x_t^{(i)}) g_B(y_{t+1}|x_{t+1}^{(i)})}{q_\theta(x_{t+1}^{(i)}|x_t^{(i)})}.$$

To avoid weight degeneracy, at each time one can resample from (7.17) to obtain a new collection of particles $x_t^{(i)}$ with weights $w_t^{(i)} = 1/N$, and then proceed to the time $t+1$. Alternatively, this resampling operation can be done according to a criterion which measures the weight degeneracy [Doucet et al., 2000b]. The SMC online EM algorithm for NMF models executing (7.15) and (7.16) based on the SMC approximation of $p_\theta(x_{1:t}|y_{1:t})$ in (7.17) is presented Algorithm 7.1.

Algorithm 7.1. ***SMC online EM algorithm for NMF models***

- ***E-step:*** *If $t = 1$, initialise θ_1; sample $\widetilde{x}_1^{(i)} \sim q_{\theta_1}(\cdot)$, and set $w_1^{(i)} = \frac{\mu_{\psi_1}(\widetilde{x}_1^{(i)}) g_{B_1}(y_1|\widetilde{x}_1^{(i)})}{q_{\theta_1}(\widetilde{x}_1^{(i)})}$, $\widetilde{T}_{1,1}^{(i)} = \widetilde{x}_1^{(i)}$, $\widetilde{C}_1^{(i)} = 0$, $\widetilde{T}_{3,1}^{(i)} = s_{3,1}(\widetilde{x}_1^{(i)})$, $i = 1, \ldots, N$. If $t > 1$,*

  - *For $i = 1, \ldots, N$, sample $\widetilde{x}_t^{(i)} \sim q_{\theta_t}(\cdot|x_{t-1}^{(i)})$ and compute*

$$\widetilde{T}_{1,t}^{(i)} = (1-\gamma_t) T_{1,t-1}^{(i)} + \gamma_t \widetilde{x}_t^{(i)},$$

$$\widetilde{T}_{3,t}^{(i)} = (1-\gamma_t) T_{3,t-1}^{(i)} + \gamma_t s_{3,t}(x_{t-1}^{(i)}, \widetilde{x}_t^{(i)})$$

$$\widetilde{C}_t^{(i)} = (1-\gamma_t) C_{t-1}^{(i)} + (1-\gamma_t)\gamma_{t-1} \frac{B_t \odot \left(y_{t-1} x_{t-1}^{(i)T}\right)}{\left(B_t x_{t-1}^{(i)}\right) \mathbf{1}_K^T},$$

$$\widetilde{w}_t^{(i)} \propto \frac{w_{t-1}^{(i)} f_{\psi_t}(\widetilde{x}_t^{(i)}|x_{t-1}^{(i)}) g_{B_t}(y_t|\widetilde{x}_t^{(i)})}{q_{\theta_t}(\widetilde{x}_t^{(i)}|x_{t-1}^{(i)})}.$$

– *Resample from particles* $\{(\widetilde{x}_t, \widetilde{T}_{1,t}, \widetilde{C}_t, \widetilde{T}_{3,t})^{(i)}\}$ *for* $i = 1, \ldots, N$ *according to the weights* $\{\widetilde{w}_t^{(i)}\}_{i=1,\ldots,N}$ *to get* $\{(x_t, T_{1,t}, C_t, T_{3,t})^{(i)}\}$ *for* $i = 1, \ldots, N$ *each with weight* $w_t^{(i)} = 1/N$.

- ***M-step:*** *If* $t < t_b$, *set* $B_{t+1} = B_t$. *Else, calculate using the particles before resampling*

$$\mathcal{S}_{1,t} = \sum_{i=1}^{N} \widetilde{T}_t^{1(i)} \widetilde{w}_t^{(i)},$$

$$\mathcal{S}_{2,t} = \sum_{i=1}^{N} \left(\widetilde{C}_t^{(i)} + \gamma_t \frac{B_t \odot \left(y_t \widetilde{x}_t^{(i)T} \right)}{\left(B_t \widetilde{x}_t^{(i)} \right) \mathbf{1}_K^T} \right) \widetilde{w}_t^{(i)}$$

$$\mathcal{S}_{3,t} = \sum_{i=1}^{N} \widetilde{T}_t^{3(i)} \widetilde{w}_t^{(i)},$$

update the parameter $\theta_{t+1} = (B_{t+1}, \psi_{t+1})$, $B_{t+1} = \frac{\mathcal{S}_{2,t}}{\mathbf{1}_M [\mathcal{S}_{1,t}]^T}$, $\psi_{t+1} = \Lambda(\mathcal{S}_{3,t})$.

Algorithm 7.1 is a special application of the SMC online EM algorithm proposed in Cappé [2009] for a general state-space HMM, and it only requires $\mathcal{O}(N)$ computations per time step. Alternatively, one can implement an $\mathcal{O}(N^2)$ SMC approximation to the online EM algorithm, see Del Moral et al. [2009] for its merits and demerits over the current $\mathcal{O}(N)$ implementation. The $\mathcal{O}(N^2)$ is made possible by plugging the following SMC approximation to $p_\theta(x_{t-1}|x_t, y_{1:t-1})$ into (7.12)

$$p_\theta^N(dx_{t-1}|x_t, y_{1:t-1}) \quad = \quad \frac{p_\theta^N(dx_{t-1}|y_{1:t-1}) f_\psi(x_t|x_{t-1})}{\int p_\theta^N(dx_{t-1}|y_{1:t-1}) f_\psi(x_t|x_{t-1})}.$$

7.4 Numerical examples

7.4.1 Multiple basis selection model

In this simple basis selection model, $X_t \in \{0, 1\}^K$ determines which columns of B are selected to contribute to the intensity of the Poisson distribution for observations. For $k = 1, \ldots, K$,

$$X_1(k) \sim \mu(\cdot), \quad \text{Prob}(X_t(k) = i | X_{t-1}(k) = j) = P(j, i),$$

where μ_0 is a distribution over $\mathcal{X}$ and P is such that $P(1,1) = p$ and $P(2,2) = q$. Estimation of $\psi = (p, q)$ can be done by calculating

$$\widehat{S}_{3,T} = \mathbb{E}_\theta \left[\sum_{i=1}^{T} s_{3,i}(X_{i-1}, X_i) \middle| Y_{1:T} = y_{1:T} \right], \quad s_{3,t}(x_t, x_{t-1}) = \sum_{k=1}^{K} \begin{bmatrix} I_{(0,0)}(x_{t-1}(k), x_t(k)) \\ I_0(x_t(k)) \\ I_{(1,1)}(x_{t-1}(k), x_t(k)) \\ I_1(x_t(k)) \end{bmatrix}$$

and applying the maximisation rule $(p^{(j+1)}, q^{(j+1)}) = \Lambda(\widehat{S}_{3,t}^{(j)})$ where $\Lambda(\cdot)$ for this model is defined as

$$\Lambda\left(\widehat{S}_{3,t}\right) = \left(\widehat{S}_{3,t}(1)/\widehat{S}_{3,t}(2), \widehat{S}_{3,t}(3)/\widehat{S}_{3,t}(4)\right).$$

Figure 7.4.1 shows the estimation results of the exact implementation of online EM (with $\gamma_t = t^{-0.8}$ and $t_b = 100$) for the 8×5 matrix B (assuming (p, q) known) given the 8×100000 matrix Y which is simulated $p = 0.8571, q = 0.6926$.

7.4.2 A relaxation of the multiple basis selection model

In this model, the process $\{X_t \in (0,1)\}_{t\geq 1}$ is not a discrete one, but it is a Markov process on the unit interval $(0,1)$. The law of the Markov chain for $\{X_t\}_{t\geq 1}$ is as follows: for $k = 1, \ldots, K$, $X_1(k) \sim \mathcal{U}(0,1)$, and

$$X_{t+1}(k)|(X_t(k) = x) \sim \rho(x)\mathcal{U}(0,x) + (1-\rho(x))\mathcal{U}(x,1),$$
$$\rho(x) = \begin{cases} \alpha, & \text{if } x \leq 0.5 \\ 1-\alpha, & \text{if } x > 0.5. \end{cases}$$

When α is close to 1, the process will spend most of its time around 0 and 1 with a strong correlation. (Figure 7.4.2 shows a realisation of $\{X_t(1)\}_{t\geq 1}$ for 500 time steps when $\alpha = 0.95$.) For estimation of α, one needs to calculate

$$\widehat{S}_{3,T} = \mathbb{E}_\theta \left[\sum_{i=1}^{T} s_{3,i}(X_{i-1}, X_i) \middle| Y_{1:T} = y_{1:T} \right],$$
$$s_{3,t}(x_{t-1}, x_t) = \begin{bmatrix} I_{A_{x_{t-1}(k)}}(x_{t-1}(k), x_t(k)) \\ I_{(0,1)\times(0,1)/A_{x_{t-1}(k)}}(x_{t-1}(k), x_t(k)) \end{bmatrix}$$

where, for $u \in (0,1)$, we define the set

$$A_u = ((0, 0.5] \times (0, u]) \cup ((0.5, 1) \times (u, 1)).$$

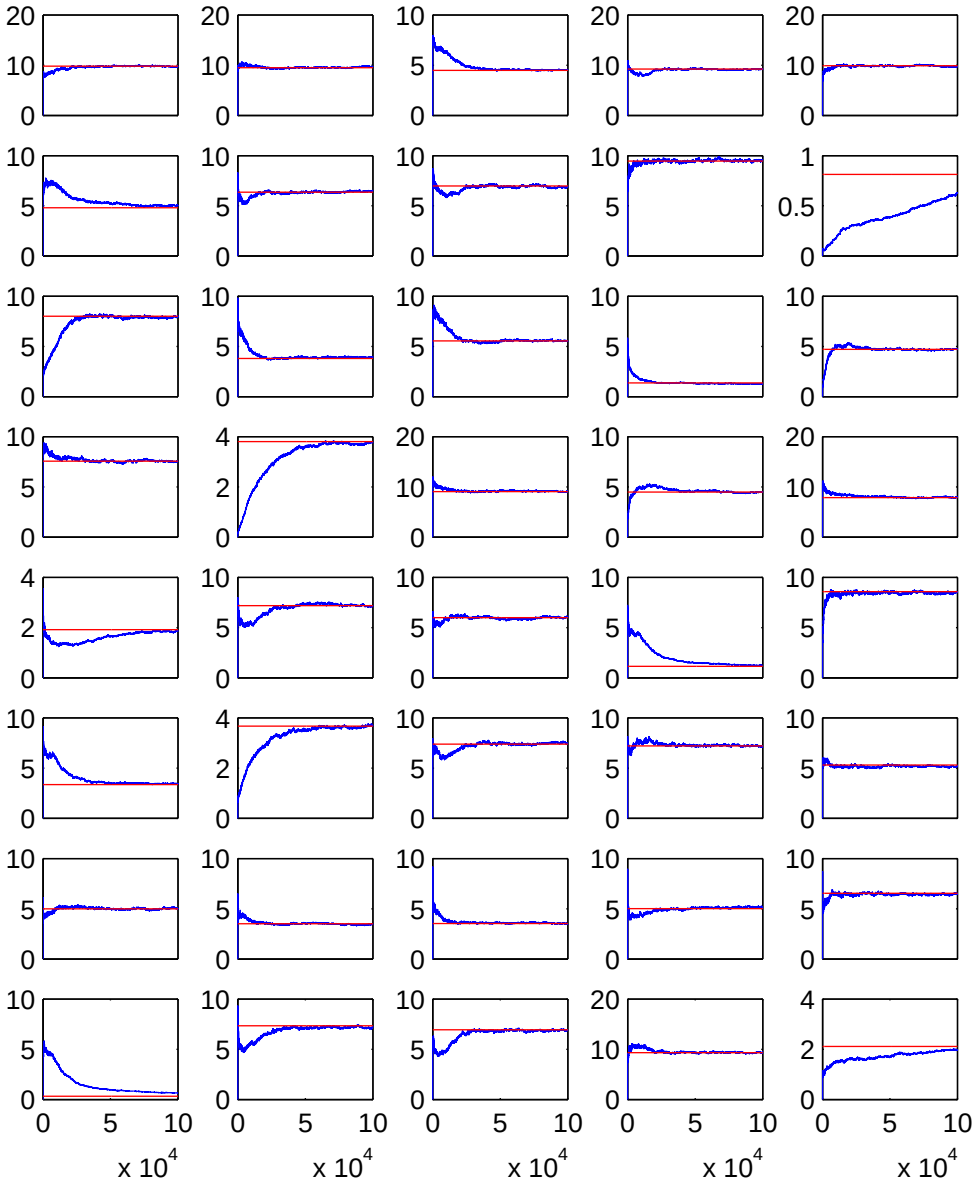


Figure 7.1: Online estimation of B in the NMF model in Section 7.4.1 using exact implementation of online EM for NMF. The (i, j)'th subfigure shows the estimation result for the $B(i, j)$ (horizontal lines).

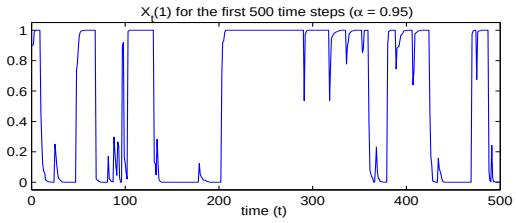


Figure 7.2: A realisation of $\{X_t(1)\}_{t\geq 1}$ for $\alpha = 0.95$.

The maximisation step for α is characterised as

$$\Lambda\left(\widehat{S}_{3,t}\right) = \widehat{S}_{3,t}(1)/\left(\widehat{S}_{3,t}(1) + \widehat{S}_{3,t}(2)\right).$$

We generated a 8×50000 observation matrix Y by using a 8×5 matrix B and $\alpha = 0.95$. We used the SMC EM algorithm described in Algorithm 7.1 to estimate B (assuming α known), with $N = 1000$ particles, $q_\theta(x_t|x_{t-1}) = f_\varphi(x_t|x_{t-1})$, $\gamma_t = t^{-0.8}$, and $t_b = 100$. Figure 7.4.2 shows the estimation results.

7.5 Discussion

In this chapter, we presented and online EM algorithm for NMF models with Poisson observations. We demonstrated an exact implementation and the SMC implementation of the online EM method on two separate NMF models. However, the method is applicable to any NMF model where the columns of the matrix X can be represented as a stationary Markov process, e.g. the log-Gaussian process.

The results in Section 7.4 do not reflect on the generality of the method, i.e., only B is estimated but the parameter φ is assumed to be known, although we formulated the estimation rules for all of the parameters in θ. Also, we perform experiments where the dimension of the B matrix may be too small for realistic scenarios. Note that in Algorithm 7.1 we used the bootstrap particle filter, which is the simplest SMC implementation. The SMC implementation may be improved devising sophisticated particle filters, (e.g. those involving better proposal densities that learn from the current observation, SMC samplers, etc.), and we believe that only with that improvement the method can handle more complete problems with higher dimensions.

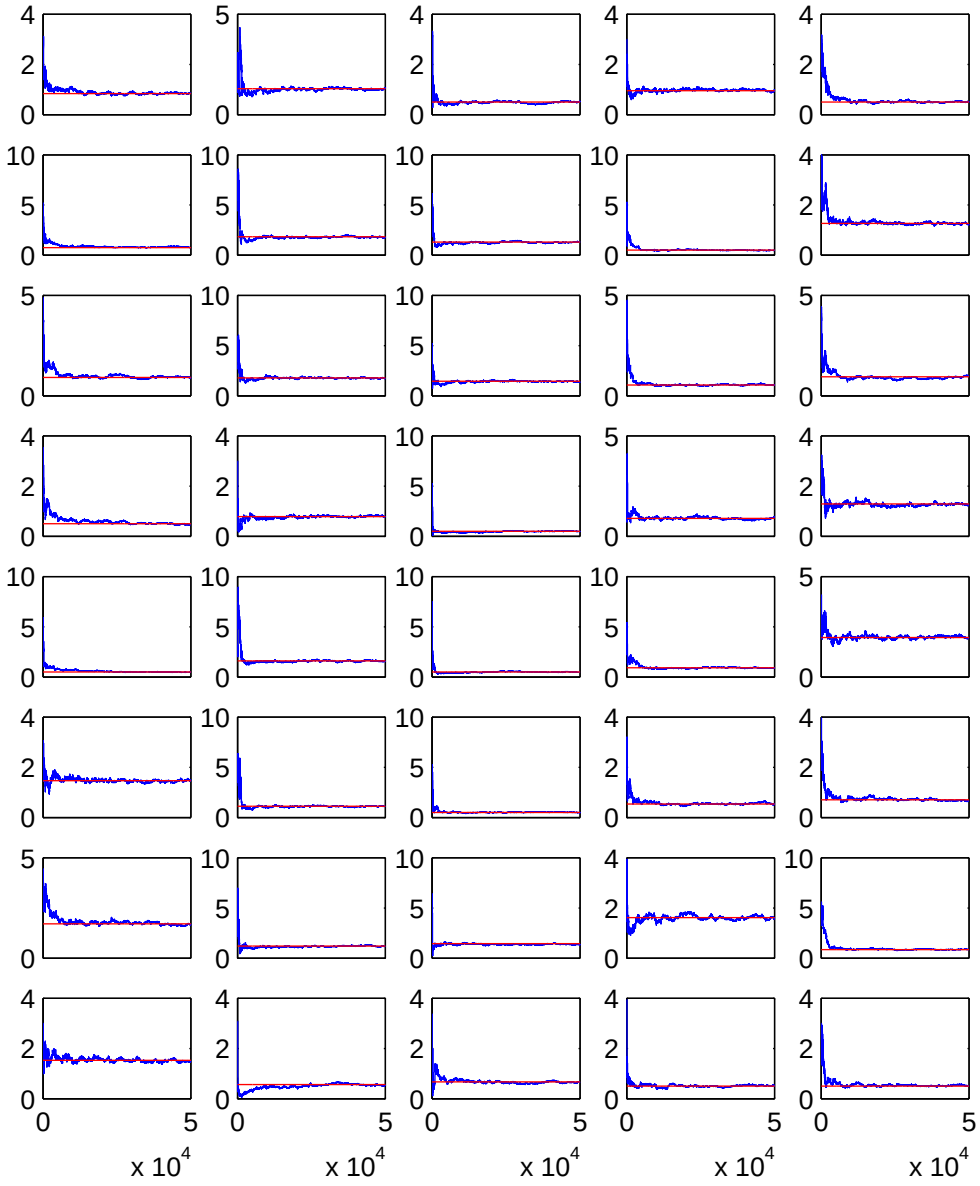


Figure 7.3: Online estimation of B in the NMF model in Section 7.4.2 using Algorithm 7.1. The (i, j)'th subfigure shows the estimation result for $B(i, j)$ (horizontal lines).

Chapter 8

Conclusions

Summary: *In this thesis, we developed batch and online SMC methods for maximum likelihood parameter estimation in several time series models. In the following, we summarise our contributions and suggest possible future directions of our work.*

8.1 Contributions

In Chapter 4, we presented a novel SMC online EM algorithm for the changepoint model and studied the stability of the associated SMC estimates. The computational cost of the developed algorithm is linear with the number of particles, unlike its counterpart online EM algorithms in the general state-space case.

In Chapter 5, we presented MLE algorithms for inferring the static parameters of the linear Gaussian MTT model, a problem which has largely been left untouched by researchers in the area. We analysed both the computational and statistical aspects of the algorithms via several numerical examples. Our developed algorithm is applicable (with slight modifications) to many extensions of the specific MTT model that we studied, as long as linear Gaussian dynamics of the MTT model are preserved by those extensions.

In Chapter 6, we presented a novel methodology for implementing MLE in HMMs with intractable observation densities. We demonstrated how both batch and online versions of gradient ascent and EM algorithms can be used for those HMMs by using the ABC approach to address the intractability. The idea of maximum likelihood parameter estimation in the context of ABC is out of the mainstream of the ABC literature. However, we think that its implementation is particularly useful for the case of long data sets where Bayesian approaches, when implemented with Monte Carlo, tend to fail because of particle degeneracy. Our algorithms are based on noisy ABC ideas and hence their estimators for the static parameter do not contain any asymptotic (in size of data) ABC bias.

In Chapter 7, we formulated the NMF problem as an MLE problem for HMMs and adapted the online SMC EM algorithm for general HMMs to estimate the matrix factors and the other unknown static parameters of the NMF model. We believe that formulating the NMF model as an HMM is a useful approach that enables the practitioner to solve

the NMF problem with more ease. Our statistical approach to NMF provides a natural framework for sequential processing which can be of significant importance in the areas of signal processing.

8.2 Future directions

In Chapter 4, one limitation of the proposed online EM algorithm for changepoint models is that it is applicable only when the constituent laws of the changepoint model given belong to the exponential family and the latent variables of each regime of the changepoint model can be integrated out analytically. In cases where this is not the case, gradient based MLE algorithms can be used, see e.g. Caron et al. [2011]. Another limitation was the assumption that the observations across segments are conditionally independent. However, there are changepoint models where observations across segments are conditionally dependent (e.g. see Barbu and Limnios [2008, Chapter 6]). Therefore, it would be a useful extension if our method could be generalised to the case where this dependency is allowed.

One obvious extension of our work for the linear Gaussian MTT model in Chapter 5 is to consider batch and online MLE algorithms in non-linear non-Gaussian MTT models. Note that for non-linear non-Gaussian models, Monte Carlo type batch and online EM algorithms may still be applied provided that the sufficient statistics for the EM are available in the required additive form [Del Moral et al., 2009]. When this condition on the sufficient statistics is not met, other methods such as gradient based MLE methods can be useful (e.g. Poyiadjis et al. [2011]).

Although we have made an initial step towards MLE in HMMs with intractable densities in Chapter 6, we have not solved all the issues regarding intractability. For example, there are cases when we cannot use gradient ascent MLE, such as when the non-linear transformation function used to generate observations has discontinuities with respect to the unknown parameter. Another case that is out of the reach of our algorithm is when the state transition law for the latent process has an intractable density. These challenging problems motivate the need to extend the algorithms developed in this thesis to cover these cases. Also note that more sophisticated SMC algorithms, at a cost of more computations, can be proposed to improve precision of the proposed algorithms. For example, at each step of the SMC filtering algorithm, an SMC sampler can be applied for targeting a sequence of ABC approximations with decreasing ABC error term ϵ [Dean et al., 2012].

The SMC online EM for NMF in Chapter 7 uses the simplest SMC implementation, namely the bootstrap particle filter. Real-life NMF problems exist in high dimensions where bootstrap would be most probably inefficient. The SMC implementation may be

improved by using more sophisticated particle filters, e.g. those involving better proposal densities that learn from the current observation, SMC samplers, etc. We believe that only with such improvements could the method handle the more practicle problems with higher dimensions. Moreover, the EM algorithm may not be applicable to more general statistical NMF models due to the lack of sufficient statistics, in those cases the online gradient MLE algorithm may be useful.

Our final comments contain a general concern: In parallel to increasing amount of research on online methods, the need for a comprehensive comparison of those methods in terms of their statistical and computational performances is increasing. Therefore, both numerical and theoretical analysis of state of the art online parameter estimation methods would be helpful in order to clarify the merits of different approaches proposed so far. Although such an attempt has been made recently for general SMC parameter estimation methods by Kantas et al. [2009]; an extensive work on online parameter estimation methods only would be an important contribution to the literature. The analysis and comparison of those methods would include investigation of their rate of convergence, accuracy, statistical efficiency, and computational complexity in both time and number of particles.

References

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Prentice-Hall, New York. 41

Andrieu, C., Davy, M., and Doucet, A. (2001). Improved auxiliary particle filtering: applications to time-varying spectral analysis. In *Statistical Signal Processing, 2001. Proceedings of the 11th IEEE Signal Processing Workshop on*, pages 309–312. 32, 39

Andrieu, C., De Freitas, J., and Doucet, A. (1999). Sequential MCMC for Bayesian model selection. In *Proceedings of the IEEE Workshop on Higher Order Statistics*, pages 130–134. 59, 60

Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64:827–836. 53

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72:269–342. 58, 59

Andrieu, C., Doucet, A., and Lee, A. (2012). Comments on Fearnhead & Prangle. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74:451–452. 146

Andrieu, C., Doucet, A., Singh, S., and Tadić, V. (2004). Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438. 61, 62, 66

Andrieu, C., Doucet, A., and Tadić, V. B. (2005). On-line parameter estimation in general state-space models. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 332–337. 4, 29, 60, 72, 139

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188. 46

Bar-Shalom, Y. and Fortmann, T. E. (1988). *Tracking and Data Association*. Academic Press, Boston. 102

Bar-Shalom, Y. and Li, X. (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishig. 102

Barbu, V. and Limnios, N. (2008). *Semi-Markov Chains and Hidden Semi-Markov Models toward Applications: Their Use in Reliability and DNA Analysis*. Springer. 72, 75, 77, 180

Beaumont, M., Cornuet, J., Marin, J., and Robert, C. (2009). Adaptive approximate Bayesian computation. *Biometrica*, 96(4):983–990. 38

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035. 139

Blackwell, D. (1947). Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, 18(1):105–110. 53

Braun, J. V. and Muller, H. G. (1998). Statistical methods for DNA sequence segmentation. *Statistical Sciences*, 13:142–162. 72

Briers, M., Doucet, A., and Maskell, S. (2010). Smoothing algorithms for state-space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89. 66

Bucak, S. S. and Gunsel, B. (2009). Incremental subspace learning via non-negative matrix factorization. *Pattern Recognition*, 42:788–797. 164

Campillo, F. and Rossi, V. (2009). Convolution particle filter for parameter estimation in general state-space models. *Aerospace and Electronic Systems, IEEE Transactions on*, 45(3):1063 –1072. 4, 38, 60

Cappé, O. (2009). Online sequential Monte Carlo EM algorithm. In *Proceedings of the IEEE Workshop on Statistical Signal Processing*. 4, 54, 67, 68, 78, 82, 120, 121, 153, 154, 164, 171, 173

Cappé, O. (2011). Online EM algorithm for hidden Markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749. 67, 73, 77, 78, 79, 114, 120, 153, 169, 171

Cappé, O., Godsill, S., and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924. 2, 46

Cappé, O., Guillin, A., Marin, J. M., and Robert, C. P. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13:907–930. 35

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer. 2, 18, 25, 40, 42, 79, 137

Caron, F., Doucet, A., and Gottardo, R. (2011). On-line changepoint detection and parameter estimation with application to genomic data. *Statistics and Computing*, pages 1–17. 10.1007/s11222-011-9248-x. 72, 75, 80, 92, 180

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). An improved particle filter for non-linear problems. *Radar Sonar & Navigation, IEE Proceedings*, 146:2–7. 30

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82. 66, 111

Celeux, G., Marin, J.-M., and Robert, C. P. (2006). Iterated importance sampling in missing data problems. *Computational Statistics and Data Analysis*, 50(12):3386–3404. 35

Cemgil, A. T. (2009). Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009:4:1–4:17. 164, 168

Cemgil, A. T., Kappen, H. J., and Barber, D. (2006). A generative model for music transcription. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(2):679–694. 72

Chambers, J. M., Mallows, C. L., and Stuck, B. W. (1976). Method for simulating stable random variables. *Journal of the American Statistical Association*, 71(354):340–344. 155

Chen, R. and Liu, J. (1996). Predictive updating methods with application to Bayesian classification. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58:397–415. 28

Chen, R. and Liu, J. S. (2000). Mixture kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(3):493–508. 43, 53

Chib, S. (1998). Estimation and comparison of multiple change-point models. *Journal of Econometrics*, 86:221241. 72

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrica*, 89(3):539–551. 32, 35, 58

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411. 53

Chopin, N. (2007). Dynamic detection of change points in long time series. *Annals of the Institute of Statistical Mathematics*, 59(2):349–366. 72, 79

Collings, I. and Ryden, T. (1998). A new maximum likelihood gradient algorithm for on-line hidden Markov model identification. In *in Proceedings of the IEEE International. Conference on Acoustic, Speech, and Signal Processing*, pages 2261–2264. 63

Coquelin, P., Deguest, R., and Munos, R. (2009). Sensitivity analysis in HMMs with application to likelihood maximization. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 387–395. 63

Cox, I. J. and Miller, M. L. (1995). On finding ranked assignments with application to multi-target tracking and motion correspondence. *Aerospace and Electronic Systems, IEEE Transactions on*, 32:48–9. 103

Crisan, D., Moral, P. D., and Lyons, T. (1999). Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields*, 5(3):293–318. 30

Danchick, R. and Newnam, G. E. (2006). Reformulating Reid's MHT method with generalised Murty K-best ranked linear assignment algorithm. *Radar, Sonar and Navigation, IEE Proceedings-*, 153(1):13–22. 103

Dean, T., Singh, S., Jasra, A., and Peters, G. (2011). Parameter estimation for hidden Markov models with intractable likelihoods. Technical Report 1103.5399, arXiv.org. 38, 139, 140, 141, 142, 143

Dean, T., Singh, S., and Yıldırım, S. (2012). Efficient and accurate approximate Bayesian computation for hidden Markov models. preprint. 180

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer-Verlag, New York. 12, 25, 42, 46, 48, 61, 79, 83, 97, 147

Del Moral, P. and Doucet, A. (2003). On a class of genealogical and interacting metropolis models. In Azéma, J., Émery, M., Ledoux, M., and Yor, M., editors, *Séminaire de Probabilités XXXVII*, volume 1832 of *Lecture Notes in Mathematics*, pages 415–446. Springer Berlin Heidelberg. 29, 54, 82

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68:411–436. 32, 33, 34, 152

Del Moral, P., Doucet, A., and Jasra, A. (2012). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22:1009–1020. 32, 38, 139

Del Moral, P., Doucet, A., and Singh, S. (2009). Forward smoothing using sequential Monte Carlo. Technical Report 638, Cambridge University, Engineering Department. 4, 54, 56, 57, 68, 73, 77, 82, 89, 92, 114, 131, 149, 153, 161, 164, 169, 173, 180

Del Moral, P., Doucet, A., and Singh, S. (2010). A backward particle interpretation of Feynman-Kac formulae. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44:947–975. 56, 83

Del Moral, P., Doucet, A., and Singh, S. (2011). Uniform stability of a particle approximation of the optimal filter derivative. Technical Report CUED/F-INFENG/TR 668, Cambridge University, Engineering Department. 4, 54, 62, 63, 64, 148, 150

Delyon, B., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):pp. 94–128. 66, 111, 112

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–38. 65

Dias, A. and Embrechts, P. (2004). Change-point analysis for dependence structures in finance and insurance. In *Risk Measures for the 21st Century*, chapter 16, pages 321–335. Wiley Finance Series. 72

Dong, M. and He, D. (2007). A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology. *Mechanical Systems and Signal Processing*, 21(5):2248–2266. 75

Donoho, D. L. (2000). High-dimensional data analysis: the curses and blessings of dimensionality. In *American Mathematical Society Conf. Math Challenges of the 21st Century*. 163

Douc, R., Éric Moulines, and Rydén, T. (2004). Asymptotic properties of the maximum likelihood estimator in autoregressive models with Markov regime. *Annals of Statistics*, 32(5):2254–2304. 127

Douc, R., Garivier, A., Moulines, E., and Olsson, J. (2011). Sequential Monte Carlo smoothing for general state space hidden Markov models. *Annals of Applied Probability*, 21(6):2109–2145. 56

Doucet, A. (1997). *Monte Carlo methods for Bayesian estimation of hidden Markov models. Application to radiation signals (in French)*. PhD thesis, University Paris-Sud Orsay, France. 28

Doucet, A., Briers, M., and Sénécal, S. (2006). Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711. 30

Doucet, A., De Freitas, J., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York. 12, 25, 39, 46, 79, 147

Doucet, A., de Freitas, N., Murphy, K., and Russell, S. (2000a). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 176–183, San Francisco, CA. Morgan Kaufmann. 43, 52, 53

Doucet, A., Godsill, S., and Andrieu, C. (2000b). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208. 2, 25, 26, 28, 46, 55, 92, 112, 172

Doucet, A. and Johansen, A. M. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. In Crisan, D. and Rozovsky, B., editors, *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press. 2, 46

Durbin, J. and Koopman, S. J. (2000). Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62:3–56. 2, 46

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. CUP: Cambridge. 137

Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science, Special Issue*, pages 131–137. 11, 13

Elliott, R. and Krishnamurthy, V. (1999). New finite-dimensional filters for parameter estimation of discrete-time linear Gaussian models. *Automatic Control, IEEE Transactions on*, 44(5):938–951. 57, 115, 116

Elliott, R. J., Ford, J. J., and Moore, J. B. (2002). On-line almost-sure parameter estimation for partially observed discrete-time linear systems with known noise characteristics. *International Journal of Adaptive Control and Signal Processing*, 16:435–453. 67, 73, 78, 79, 120, 153, 171

Fearnhead, P. (2002). MCMC, sufficient statistics and particle filters. *Journal of Computational and Graphical Statistics*, 11:848–862. 59

Fearnhead, P. (2006). Efficient and exact Bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16:203–213. 72

Fearnhead, P. (2008). Computational methods for complex stochastic systems: a review of some alternatives to MCMC. *Statistics and Computing*, 18(2):151–171. 2, 46

Fearnhead, P. and Clifford, P. (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65:887–889. 30, 43, 53

Fearnhead, P. and Liu, Z. (2007). On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605. 30, 42, 53, 72, 79, 80

Fearnhead, P. and Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474. 36, 38, 140, 143, 157

Fearnhead, P. and Vasileiou, D. (2009). Bayesian analysis of isochores. *Journal of the American Statistical Association*, 104(485):132–141. 72, 75, 77, 80, 89, 90, 91, 92

Felsenstein, J. and Churchill, G. (1996). A hidden Markov model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution*, 13:93–104. 137

Févotte, C. and Cemgil, A. T. (2009). Nonnegative matrix factorisations as probabilistic inference in composite models. In *Proc. 17th European Signal Processing Conference (EUSIPCO'09)*, Glasgow. 164

Gales, M. J. F. and Young, S. J. (1993). The theory of segmental hidden Markov models. Technical report, Cambridge University Engineering Department. 72, 75, 77

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409. 24

Gelman, A. and Meng, X. L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13:163–185. 32

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6(6):721–741. 24

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339. 16, 17, 27

Gilks, W. R. (1992). Derivative-free adaptive rejection sampling for Gibbs sampling. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*, pages 641–649. Oxford University Press, Oxford, UK. 14

Gilks, W. R. and Berzuini, C. (2001). Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146. 4, 30, 35, 58, 59

Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 44(4):455–472. 14

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC. 18

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(2):337–348. 14

Givens, G. H. and Raftery, A. E. (1996). Local adaptive importance sampling for multivariate densities with strong non-linear relationships. *J. Amer. Statist. Assoc.*, 91:132–141. 33

Godsill, S., Doucet, A., and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99:156–168. 55

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F*, 140(6):107–113. 2, 29, 30, 39, 48, 147

Guyader, A., Hengartner, N., and Matzner-Lober, E. (2011). Simulation and estimation of extreme quantiles and extreme probabilities. *Applied Mathematics & Optimization*, 64(2):171–196. 139

Handschin, J. E. (1970). Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6:555–563. 26

Handschin, J. E. and Mayne, D. (1969). Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9:547–559. 26

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 52(1):97–109. 18, 22, 23

Hernando, D., Valentino, C., and Cybenko, G. (2005). Efficient computation of the hidden Markov model entropy for a given observation sequence. *Information Theory, IEEE Transactions on*, 51:2681–2685. 57

Higuchi, T. (2001). Self-organizing time series model. In Doucet, A., De Freitas, J., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*, pages 429–444. Springer-Verlag. 4, 60

Hitchcock, D. B. (2003). A history of the Metropolis-Hastings algorithm. *The American Statistician*, 57:254–257. 24

Hoffman, M., Blei, D., and Bach, F. (2010). Online learning for latent Dirichlet allocation. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 856–864. 164

Hue, C., Le Cadre, J.-P., and Perez, P. (2002). Sequential Monte Carlo methods for multiple target tracking and data fusion. *Signal Processing, IEEE Transactions on*, 50(2):300–325. 102

Ionides, E. L., Bhadra, A., and King, A. (2011). Iterated filtering. *The Annals of Statistics*, 39(3):1776–1802. 69, 162

Jasra, A., Singh, S., Martin, J., and McCoy, E. (2012). Filtering via approximate Bayesian computation. *Statistics and Computing* (to appear), pages 1–15. 38, 139, 140, 141, 145, 147

Johansen, A. M., Moral, P. D., and Doucet, A. (2005). Sequential Monte Carlo samplers for rare events. Technical report, University of Cambridge, Department of Engineering. 32

Johnson, T. D., Elashoff, R. M., and Harkema, S. J. (2003). A Bayesian change-point analysis of electromyographic data: detecting muscle activation patterns and associated applications. *Biostatistics*, 4:143–164. 72

Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls 3*, pages 182–193. 46

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME; Series D: Journal of Basic Engineering*, 82:35–45. 45

Kantas, N., Doucet, A., Singh, S. S., and Maciejowski, J. M. (2009). An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In *Proceedings IFAC System Identification (SysId) Meeting.* 3, 4, 58, 67, 72, 78, 138, 171, 181

Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *The Review of Economic Studies*, 65:361–393. 137

Kitagawa, G. (1996). Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 1:1–25. 2, 30, 46

Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215. 4, 60

Kitagawa, G. and Sato, S. (2001). Monte Carlo smoothing and self-organising state-space model. In Doucet, A., De Freitas, J., and Gordon, N., editors, *Sequential Monte Carlo in Practice*, pages 178–195. New York: Springer. 55

Klaas, M., de Freitas, N., and Doucet, A. (2005). Toward practical N^2 Monte Carlo: the marginal particle filter. In *UAI*, pages 308–315. AUAI Press. 50, 51

Kolda, T. and Bader, B. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500. 164

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288. 17, 28, 48

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37. 164

Lavielle, M. and Lebarbier, E. (2001). An application of MCMC methods for the multiple change-points. *Signal Processing*, 81:39–53. 72

Le Gland, F. and Mevel, L. (1997). Recursive estimation in hidden Markov models. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 4, pages 3468–3473. 63, 64

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791. 164, 167

Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. 167

Lefevre, A., Bach, F., and Fevotte, C. (2011). Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence. In *(WASPAA) IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 313–316. 164

Liang, F. (2002). Dynamically weighted importance sampling in Monte Carlo computation. *Journal of the American Statistical Association*, 97:807–821. 35

Lin, M. T., Zhang, J. L., Cheng, Q., and Chen, R. (2005). Independent particle filters. *Journal of the American Statistical Association*, 100(472):1412–1421. 48, 51

Liu, J. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119. 17, 48

Liu, J. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer Verlag, New York, NY, USA. 29

Liu, J. and Chen, R. (1995). Blind deconvolution via sequential imputation. *Journal of the American Statistical Association*, 90:567–576. 28, 48

Liu, J. and Chen, R. (1998). Sequential Monte-Carlo methods for dynamic systems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 93:1032–1044. 2, 30, 46

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In Doucet, A., De Freitas, J., and Gordon, N. J., editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York. 60

Lombardi, M. J. and Calzolari, G. (2009). Indirect estimation of α-stable stochastic volatility models. *Computational Statistics & Data Analysis*, 53(6):2298–2308. 160

Lund, R. and Reeves, J. (2002). Detection of undocumented changepoints: A revision of the two-phase regression model. *Journal of Climate*, 15:2547–2554. 72

Mahler, R. (2003). Multitarget Bayes filtering via first-order multitarget moments. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(4):1152 – 1178. 102

Mahler, R., Vo, B., and Vo, B. (2011). CPHD filtering with unknown clutter rate and detection profile. *Signal Processing, IEEE Transactions on*, 59(8):3497–3513. 102

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60. 164

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2011). Approximate Bayesian computational methods. *Statistics and Computing*, pages 1–14. 36, 38, 139

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 100(26):15324–15328. 37, 139

Marsaglia, G. (1977). The squeeze method for generating gamma variates. *Computers and Mathematics with Applications*, 3(4):321–325. 14

Mayne, D. (1966). A solution of the smoothing problem for linear dynamic systems. *Automatica*, 4:73–92. 26

Mengersen, K. and Tweedie, R. L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *Annals of Statistics*, 24:101–121. 23

Metropolis, N. (1987). The beginning of the Monte Carlo method. *Los Alamos Science, Special Issue*, pages 125–130. 11

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092. 18, 22, 23

Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):pp. 335–341. 2, 11, 12

Meyn, S. and Tweedie, R. L. (2009). *Markov Chains and Stochastic Stability*. Cambridge University Press, New York, NY, USA, 2nd edition. 18

Mongillo, G. and Deneve, S. (2008). Online learning with hidden Markov models. *Neural Computation*, 20(7):1706–1716. 57, 67, 73, 78, 79, 120, 153, 171

Murphy, K. P. (2002). Hidden semi-Markov models (hsmms). Technical report, UBC. 75

Murty, K. G. (1968). An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16(3):682–687. 103, 113, 133

Neal, R. (2001). Annealed importance sampling. *Statistics and Computing*, 11:125–139. 32, 35

Newman, M. E. J. and Barkema, G. T. (1999). *Monte Carlo Methods in Statistical Physics*. Oxford University Press, USA. 12

Ng, W., Li, J., Godsill, S., and Vermaak, J. (2005). A hybrid approach for online joint detection and tracking for multiple targets. In *Aerospace Conference, 2005 IEEE*, pages 2126 –2141. 102, 103

Nielsen, S. F. (2000). The stochastic EM algorithm: Estimation and asymptotic results. *Bernoulli*, 6(3):457–489. 66

Ó Ruanaidh, J. and Fitzgerald, W. J. (1996). *Numerical Bayesion Methods Applied to Signal Processing*. Springer, New York. 72

Oh, S., Russell, S., and Sastry, S. (2009). Markov chain monte carlo data association for multi-target tracking. *Automatic Control, IEEE Transactions on*, 54(3):481 –497. 102, 103, 112

Oliver, J. L., Carpena, P., Hackenberg, M., and Bernaola-Galvan, P. (2004). Isofinder: Computational prediction of isochores in genome sequences. *Nucleic Acids Research*, 32:W287–W29. 90

Olsson, J., Cappé, O., Douc, R., and Moulines, E. (2008). Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14:155–179. 4, 29, 55, 66, 72, 82, 139

Peters, G., Sisson, S., and Fan, Y. (2011). Likelihood-free bayesian inference for alpha stable models. *Computational Statistics and Data Analysis*, 56(11):3743–3756. 155

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599. 31, 39, 48, 49, 147

Polson, N. G., Stroud, J. R., and Müller, P. (2008). Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):413–428. 4, 60

Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80. 4, 54, 55, 62, 63, 64, 69, 131, 148, 150, 154, 180

Press, W. H. (2007). *Numerical Recipes : The Art of Scientific Computing*. Cambridge University Press, 3rd edition. 11

Pritchard, J., Seielstad, M., Perez-Lezaun, A., and Feldman, M. (1999). Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16:1791–1798. 36, 139

Punskaya, E., Andrieu, C., Doucet, A., and Fitzgerald, W. J. (2002). Bayesian curve fitting using MCMC with applications to signal segmentation. *IEEE Transactions on Signal Processing*, 50:747–758. 72

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. 39, 41, 45

Rayner, G. D. and MacGillivray, H. L. (2002). Numerical maximum likelihood estimation for the g-and-k and generalized g-and-h distributions. *Statistics and Computing*, 12(1):57–75. 157, 159

Reid, D. B. (1979). An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24:843–854. 102

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. New York: Springer, 2 edition. 12, 13, 15, 18, 25

Roberts, G. and Smith, A. (1994). Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207–216. 23, 25

Roberts, G. and Tweedie, R. (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83:95–110. 23, 25

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 12(4):1151–1172. 36

Rubin, D. B. (1987). A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modest: the SIR algorithm (discussion of Tanner and Wong). *Journal of the American Statistical Association*, 82:543–546. 29

Salakhutdinov, R. and Mnih, A. (2008). Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20. 164

Särkkä, S., Vehtari, A., and Lampinen, J. (2004). Rao-Blackwellized Monte Carlo data association for multiple target tracking. In *In Proceedings of the Seventh International Conference on Information Fusion*, pages 583–590. 53

Shiryaev, A. N. (1995). *Probability*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition. 12, 13, 18

Singh, A. P. and Gordon, G. J. (2008). A unified view of matrix factorization models. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, ECML PKDD '08, pages 358–373, Berlin, Heidelberg. Springer-Verlag. 164

Singh, S., Whiteley, N., and Godsill, S. (2011). An approximate likelihood method for estimating the static parameters in multi-target tracking models. In Barber, D., Cemgil, T., and Chiappa, S., editors, *Bayesian Time Series Models*, chapter 11, pages 225–244. Cambridge University Press. 102

Singh, S. S., Vo, B.-N., Baddeley, A., and Zuyev, S. (2009). Filters for spatial point processes. *SIAM Journal on Control and Optimization*, 48(4):2275–2295. 102

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 104(6):1760–1765. 38

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2009). Correction for Sisson et al., Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 106(39):16889. 38

Sorenson, H. W. (1985). *Kalman Filtering: Theory and Application*. IEEE Press, reprint edition. 45

Stephens, D. A. (1994). Bayesian retrospective multiple-changepoint identification. *Applied Statistics*, 43:159–178. 72

Storlie, C., Lee, T., Hannig, J., and Nychka, D. (2009). Tracking of multiple merging and splitting targets: A statistical perspective. *Statistica Sinica*, 19:1–52. 102

Storvik, G. (2002). Particle filters in state space models with the presence of unknown static parameters. *Signal Processing, IEEE Transactions on*, 50(2):281–289. 59

Streit, R. and Luginbuhi, T. (1995). Probabilistic multi-hypothesis tracking. Technical Report 10,428, Naval Undersea Warfare Center Division, Newport, Rhode Island. 102

Tavaré, S., Balding, D., Griffith, R., and Donnelly, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518. 36

Tierney, L. (1994). Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1762. 18, 23, 25

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 46(2):257–267. 63

Toni, T., Welch, D. Strelkowa, N., Ipsen, A., and Stumpf, M. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202. 38

Vermaak, J., Godsill, S., and Perez, P. (2005). Monte carlo filtering for multi target tracking and data association. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(1):309 – 332. 102

Vo, B. and Ma, W. (2006). The Gaussian mixture probability hypothesis density filter. *Signal Processing, IEEE Transactions on*, 54(11):4091–4104. 130

Vo, B.-N., Singh, S., and Doucet, A. (2003). Random finite sets and sequential monte carlo methods in multi-target tracking. In *Radar Conference, 2003. Proceedings of the International*, pages 486–491. 102

Vo, B.-N., Singh, S., and Doucet, A. (2005). Sequential Monte Carlo methods for multitarget filtering with random finite sets. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(4):1224–1245. 102

von Neumann, J. (1951). Various techniques used in connection with random digits. *Journal of Research of the National Bureau of Standards*, 12:36–38. 13

Wei, G. C. G. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704. 66, 111

West, M. (1993). Approximating posterior distributions by mixture. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 55:409–422. 33

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, 50:1–25. 142

Whiteley, N., Doucet, A., and Andrieu, C. (2009). Particle MCMC for multiple change-point models. Technical report, University of Bristol, Department of Mathematics. 75, 92

Whiteley, N., Singh, S., and Godsill, S. (2010). Auxiliary particle implementation of probability hypothesis density filter. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(3):1437–1454. 102, 130

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85. 30

Wilkinson, R. (2008). Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. Technical report, 0811.3355, arXiv.org. 37, 140, 142

Wills, A., Schn, T. B., and Ninness, B. (2008). Parameter estimation for discrete-time nonlinear systems using EM. In *Proc. 17th IFAC World Congress*. 66

Yıldırım, S., Cemgil, A. T., and Singh, S. S. (2012a). An online expectation-maximisation algorithm for nonnegative matrix factorisation models. In *16th IFAC Symposium on System Identification*. SYSID 2012. 163

Yıldırım, S., Jiang, L., Singh, S. S., and Dean, T. (2012b). Estimating the static parameters in linear Gaussian multiple target tracking models. Technical Report CUED/F-INFENG/TR.681, University of Cambridge, Department of Engineering. 101

Yıldırım, S., Jiang, L., Singh, S. S., and Dean, T. (2012c). A Monte Carlo expectation-maximisation algorithm for parameter estimation in multiple target tracking. In *Information Fusion (FUSION), 2012 15th International Conference on*. Fusion 2012. 101

Yıldırım, S., Singh, S. S., and Doucet, A. (2012d). An online expectation-maximization algorithm for changepoint models. *Journal of Computational and Graphical Statistics, to appear*. 71

Yoon, J. and Singh, S. (2008). A Bayesian approach to tracking in single molecule fluorescence microscopy. Technical Report CUED/F-INFENG/TR-612, University of Cambridge. 101, 102, 103