Uriel Moreira Silva

A General Framework for Sequential Parameter Learning in Hidden Markov Models

Belo Horizonte - MG

Uriel Moreira Silva

A General Framework for Sequential Parameter Learning in Hidden Markov Models

Tese apresentada ao Departamento de Estatística da UFMG como requisito parcial para a obtenção do título de Doutor em Estatística.

Orientador: Luiz Henrique Duczmal
Co-orientadora: Denise Bulgarelli Duczmal

Belo Horizonte - MG
December 30, 2020

à minha família

AGRADECIMENTOS

Primeiramente agradeço aos meus orientadores Luiz e Denise, sem o apoio dos quais esse trabalho simplesmente não seria possível. Suas contribuições à minha formação e ao meu trabalho serão sempre parte integral da minha trajetória.

Agradeço profundamente à toda minha família, que é base tanto da minha vida pessoal quanto profissional. Em especial agradeço à minha mãe Bernadete, ao meu pai Evandro, à minha esposa Alice e à minha sogra Dalva pela força, paciência e pelo apoio incondicional. O amor de vocês é a luz do meu caminho.

Às professoras Guta e Waleska, obrigado pelo carinho e pelo acolhimento em um momento tão crucial da minha jornada. Sou profundamente grato à vocês e aos colegas do OSUBH, onde compartilhei tantas experiências fundamentais ao meu crescimento.

Por fim, agradeço à minha família extendida: todos aqueles com os quais não compartilho laços consanguíneos mas cujo carinho e consideração são como se assim o fosse. Os amigos de Itaúna, de Belo Horizonte, do Departamento de Estatística da UFMG, do Ibmec, enfim; todas essas pessoas maravilhosas com as quais eu tive o prazer de conviver.

Agradeço também à CAPES, FAPEMIG e CNPq por financiamento parcial durante o desenvolvimento desse trabalho.

I think that when we know that we actually do live in uncertainty, then we ought to admit it; it is of great value to realize that we do not know the answers to different questions. This attitude of mind - this attitude of uncertainty - is vital to the scientist, and it is this attitude of mind which the student must first acquire.

- Richard P. Feynman

RESUMO

Nessa tese é introduzido um novo paradigma de aprendizagem de parâmetros sequencial em modelos de Markov ocultos, capaz de acomodar vários outros algoritmos encontrados na literatura como casos particulares. Essa generalidade é possível principalmente devido à um formalismo alternativo para regularização nesses modelos. Para ilustrar a flexibilidade do novo paradigma, foram desenvolvidos três novos algoritmos, incluindo uma versão melhorada e completamente adaptada do clássico filtro de Liu e West. Considerando também esquemas de reamostragem mais eficientes, é ilustrado que em alguns casos o desempenho inadequado de alguns algoritmos de aprendizagem de parâmetros sequencial previamente observado na literatura pode em sua maioria ser atribuído à degeneração de caminhos inerente à esses métodos, degeneração essa que a metodologia proposta ativamente busca mitigar. Destaca-se também que é fornecida evidência de que os algoritmos para aprendizagem de parâmetros discutidos aqui podem fornecer estimativas compatíveis com algoritmos computacionalmente intensivos e que compõem o estado da arte dessa literatura, como Monte Carlo via cadeias de Markov baseados em métodos de partículas.

Palavras-chave: *Inferência Bayesiana, Métodos de Monte Carlo sequenciais, Modelos de Markov ocultos*

ABSTRACT

In this thesis we introduce a novel framework for sequential parameter learning in Hidden Markov models capable of accommodating several other algorithms found in the literature as special cases. This generality is achieved mainly by providing an alternative formalism to the role of regularization in this setting. In order to illustrate the flexibility allowed by this framework, we develop three novel algorithms, including an improved and fully-adapted version of the celebrated Liu and West filter. By also considering more efficient resampling schemes, we illustrate that in some cases the poor performance of sequential parameter learning algorithms previously observed in the literature can mostly be attributed to the inherent path degeneracy in these methods, which we actively aim to mitigate. Crucially, we also provide evidence that the parameter learning algorithms discussed here can provide estimates that are compatible with state-of-the-art computationally intensive algorithms, such as particle Markov Chain Monte Carlo.

Keywords: *Bayesian inference; Sequential Monte Carlo methods; Hidden Markov models*

LIST OF ABBREVIATIONS

APF = Auxiliary Particle Filter

AR = autoregressive

CLT = Central Limit Theorem

FA = fully-adapted

FALW = Fully-adapted Liu and West

FAPF = Fully-adapted Auxiliary Particle Filter

FF = fertility factor

FFBS = Forward-Filtering Backward-Sampling

FS = Flury and Shephard

HMM = Hidden Markov model

iid = independent and identically distributed

IS = Importance Sampling

KF = Kalman Filter

LW = Liu and West

MC = Monte Carlo

MCMC = Markov Chain Monte Carlo

MH = Metropolis-Hastings

NLSM = Nonlinear Seasonal Model

PF = particle filter

PIMH = Particle Independent Metropolis Hastings

PL = Particle Learning

pMCMC = particle Markov Chain Monte Carlo

PMMH = Particle Marginal Metropolis Hastings

RAM = Robust Adaptive Metropolis

RPL = Regularized Particle Learning

RWM = Random Walk Metropolis

SIR = Sampling Importance Resampling

SIS = Sequential Importance Sampling

SLLN = Strong Law of Large Numbers

SMC = Sequential Monte Carlo

SNR = signal-to-noise ratio

SV = Stochastic Volatility

WLLN = Weak Law of Large Numbers

Contents

Chapter 1

Introduction

In this thesis we deal with inference for Hidden Markov Models (HMMs), also known as state space models. These are discrete-time stochastic processes essentialy composed of a Markov Chain that can only be observed via another process (hence the name). Although the term "Hidden Markov model" is typically reserved for the cases in which the state space is finite and the term "state space model" to refer to those in which the state space is continuous and obey a certain type of Markov transition, here we will follow the tradition of Cappé et al. (2005) and use the term Hidden Markov model to denote both of these objects, as well as even more general ones as seen later on.

Although deceptively simple in their formulation, HMMs can exhibit a surprising range of behaviors, being suitable to problems ranging from genetics to economics. Perhaps the most classical application of this type of model is tracking a moving object subject to measurement error. An important and quite famous case of this instance is the navigation system developed for the Apollo project (Grewal and Andrews, 2010), which relied on the celebrated Kalman filter (Kalman, 1960).

Another surprising fact regarding HMMs is how difficult performing inference for them really is in the general case. Historically, substantial developments in inference for these models could only be made by imposing several restrictions on the process, making the ensuing results particular only to a specific subset of cases. Despite this, the corresponding procedures have found many fruitful and important areas of application, such as genetics for the Baum-Welch algorithm (starting with Baum and Petrie, 1966, which requires a finite state space) and tracking for the aforementioned Kalman filter (Kalman, 1960, which requires linear and Gaussian components).

With the advent of more powerful computers, however, the field of inference for Hidden Markov models naturally turned from analytical to simulation-based techniques, which can in principle be applied to any general HMM. Amongst these, the first widely successful method is the Bootstrap Filter of Gordon et al. (1993), which really set the tone for the subsequent developments in the field; another landmark algorithm is the Auxiliary Particle Filter of Pitt and Shephard (1999). Both techniques fall under the Sequential Monte Carlo (SMC) class of methods, also known as particle filters. That particular decade saw many theoretical and applied contributions to SMC, many of which are contained in the classical review volume by Doucet et al. (2001).

The main defining feature of SMC methods is, as the name implies, that they are sequential. They are therefore naturally suitable for performing inference in HMMs, exploiting the inherent sequential structure of these models in order to yield very efficient results. This is in stark contrast to non-sequential algorithms such as the numerically-

intensive Markov Chain Monte Carlo algorithms, which is mostly why SMC algorithms have become so popular in practice.

Most of the early SMC methods for HMMs focused entirely on performing inference for the underlying Markov chain, also known as the states of the model. However, HMMs are frequently indexed by a set of "static" parameters (this naming convention is used to distinguish these from the components of the hidden chain, which are often thought of as the "dynamic" parameters of the model), and inference for these have been the main focus of the literature for the last two decades.

Essentially, there are two main approaches for dealing with parameter inference in HMMs: offline (or batch) techniques, in which data come in "batches", i.e. in blocks at a time, and everything has to be recomputed everytime a new block comes in; and online techniques, in which data comes in sequentially and inference is performed on-the-fly. In a Bayesian inference paradigm, the latter are also given the name of sequential parameter learning techniques.

To give an example of how diverse and fruitful the literature on sequential parameter learning techniques is, we highlight the works of Kitagawa (1998), Andrieu et al. (1999), BøLviken et al. (2001), Liu and West (2001), Gilks and Berzuini (2001), Chopin (2002), Fearnhead (2002), Storvik (2002), Vercauteren et al. (2005), Polson et al. (2008), Flury and Shephard (2009), Carvalho et al. (2010), Chopin et al. (2013) and Fulop and Li (2013). Collectively, the variety of fields in which these techniques are used to deal with problems arising in empirical settings also illustrate their effectiveness. Examples range from tracking (Wang et al., 2009; Ghaeminia et al., 2010; Liang and Piché, 2010; Nemeth et al., 2013) to epidemiology (Rodeiro and Lawson, 2006; Dukic et al., 2012; Lin and Ludkovski, 2014; Liu et al., 2015), ecology (Peters et al., 2010), econometrics (Golightly and Wilkinson, 2006; Carvalho and Lopes, 2007; Fulop and Li, 2013), finance (Yümlü et al., 2015; Jacquier et al., 2016; Warty et al., 2018; Virbickaitė et al., 2019) and even psychometrics (Reichenberg, 2018).

Popular and efficient as they might be, however, sequential parameter learning methods suffer from the unavoidable problem of path degeneracy (Andrieu et al., 2005). Path degeneracy mostly stems from inefficient resampling (resampling is an integral part of SMC) and poor (some of them even having nonvanishing asymptotic biases) rules for moving the parameters around in their space. As noted as early as Andrieu et al. (1999) and by several subsequent works in the literature (e.g. Andrieu et al., 2005; Chopin et al., 2010; Lopes and Tsay, 2011; Prado and Lopes, 2013; Kantas et al., 2015), path degeneracy is identified as the main culprit for the poor performance of sequential parameter learning in most settings. Typically, however, no further effort is made to improve upon this behavior.

Our main goal in this thesis is to provide a general framework for parameter learning capable of accommodating several of the methods found in the literature as particular cases. Within this unified framework, we explore the performance of methods that already exist and some of which we also propose here. By actively attempting to reduce path degeneracy in the ensuing algorithms, we also provide evidence that they can then perform quite well in practice, providing compatible results with state-of-the-art numerically intensive methods.

This thesis is organized as follows: Chapters 1 and 2 contain a formalization of the main concepts and properties regarding HMMs and essential results on state inference needed for a proper understanding of the core material. Then, Chapter 3 on parameter inference contains our main contributions and original research. Chapter 4 contains

simulation-based experiments and numerical results that illustrate the effectiveness of our methods in practice while providing a further contribution in their own right, and Chapter 5 contains the concluding remarks.

1.1 Hidden Markov Models

Let $(\Omega, \mathcal{F}, \mathbb{P}_\theta)$ be a probability space, where $\mathbb{P}_\theta$ is in a parametric family $\mathcal{P} = \{\mathbb{P}_\theta : \theta \in \Theta\}$. We denote by p any generic probability density of $\mathbb{P}_\theta$ with respect to a suitable sigma-finite dominating measure and by $z_{1:k}$ the sequence $(z_1, \ldots, z_k)$ for positive integer k. As it is common practice in the literature, upper and lowercase respectively are used here to distinguish random variables from their realized values, e.g. $X(\omega) = x$ for a particular $\omega \in \Omega$.

Definition 1.1.1. *A* state space *or* hidden Markov *model (HMM) is a discrete-time stochastic process $(X_t, Y_t)_{t\geq 0}$ indexed by $\theta \in \Theta$ and taking values in $\mathcal{X} \times \mathcal{Y}$ such that $(X_t)_{t\geq 0}$ is an unobserved Markov process and, for each t, the probability distribution of $Y_t|X_t$ depends only on X_t.*

The definition of a Markov process (the term Markov *chain* is also frequently used) requires that, for $t \geq 1$, the law of $X_t|X_{1:t-1}$ depends only on X_{t-1}. That is,

$$p(x_t|x_{1:t-1}, \theta) = p(x_t|x_{t-1}, \theta) := f(x_t|x_{t-1}, \theta), \tag{1.1}$$

with $\nu(x_0|\theta)$ being the initial distribution[1] of the chain, i.e. corresponding to X_0. Informally, the term "unobserved" (or "hidden") is used here to point out that $(X_t)_{t\geq 0}$ is a latent process, i.e. not directly available for inference. Instead, inference about the model can only be made through the *measurements* (more often simply referred to as *observations*) $(Y_t)_{t\geq 0}$.

The rest of Definition 1.1.1 requires that, for each t, $Y_t|X_t$ depend only on X_t. In essence, this means that

$$p(y_t|x_t, x_{1:t-1}, y_{1:t-1}, \theta) = p(y_t|x_t, \theta) := g(y_t|x_t, \theta). \tag{1.2}$$

The density function g is sometimes known as the *conditional likelihood* of Y_t given X_t, since it can be interpreted as the likelihood of X_t assuming a certain value x_t given the observed value $Y_t = y_t$. From here on, we will sometimes recall equation (1.2) as the *conditional independence* property that the sequence $(Y_t)_{t\geq 0}$ possesses in HMMs; see item 1.5 in Proposition 1.1.1 below for more details.

An alternative representation of an HMM is in graph form, more specifically as a directed and acyclic graph; see Figure 1.1. This alternative representation allows us to see intuitively how the model evolves over time, and neatly summarizes the serial dependence across $(X_t)_{t\geq 0}$, the conditional independence of $(Y_t)_{t\geq 0}$ and the global dependence on θ.

Before moving on to further discuss state and parameter inference, we summarize in Proposition 1.1.1 some important properties concerning HMMs that routinely appear throughout the rest of the text.

[1] Although ν is actually the probability density function of $\mathbb{P}_\theta$ with respect to the sigma-finite dominating measure dx_0, it is a common practice in the literature to use the terms density and distribution interchangeably. We feel that this adoption, along with some abuses in notation such as using $X_0 \sim \nu$ to denote that X_0 has a distribution with density ν, enhances the flow of the text, and is mantained here except in cases where a clear distinction cannot be extracted from the context.

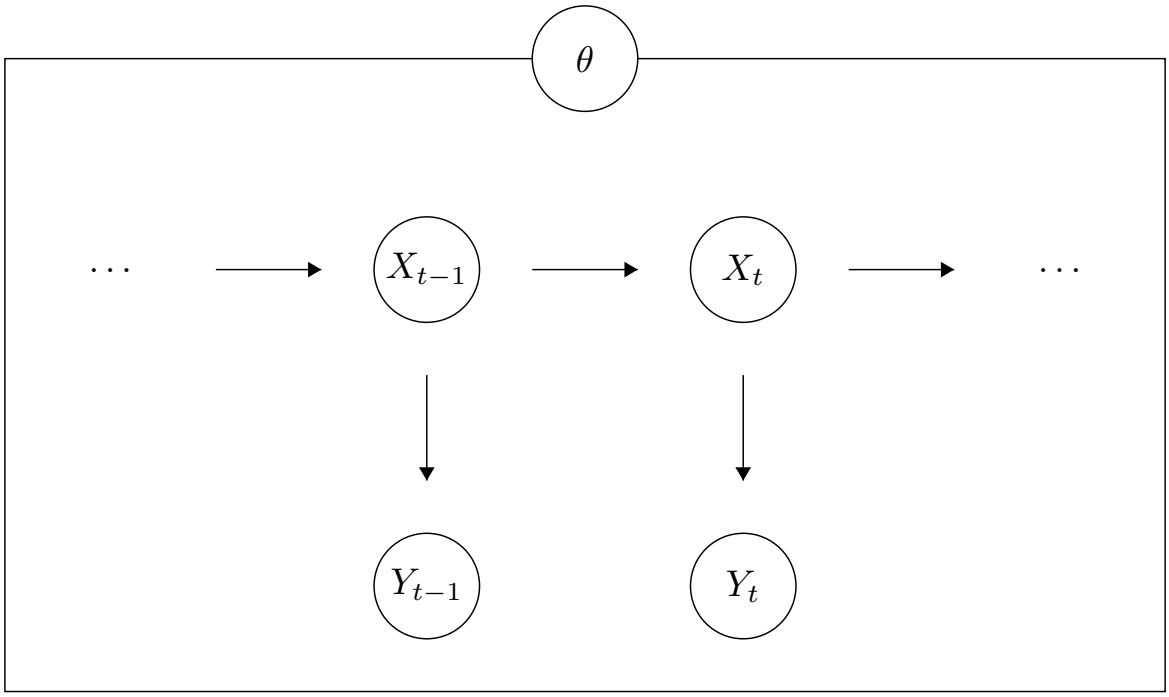

Figure 1.1: Graphical representation of a Hidden Markov model.

Proposition 1.1.1. *Let $(X_t, Y_t)_{t\geq 0}$ be a hidden Markov model according to Definition 1.1.1. Then*

(i) The sequence $(X_t)_{t\geq 0}$ admits the predictive decomposition[2]

$$p(x_{0:t}|\theta) = \nu(x_0|\theta)\prod_{k=1}^{t} f(x_k|x_{k-1},\theta). \tag{1.3}$$

(ii) For $t \geq 1$ and $0 \leq k \leq t-1$, $X_t|X_{k:t-1}, Y_{k:t-1} =^d X_t|X_{t-1}$ (here "$=^d$" denotes equality in distribution), i.e.

$$p(x_t|x_{k:t-1}, y_{k:t-1}, \theta) = f(x_t|x_{t-1},\theta). \tag{1.4}$$

(iii) Given $(X_t)_{t\geq 0}$, the sequence $(Y_t)_{t\geq 0}$ is conditionally independent, i.e.

$$p(y_{0:t}|x_{0:t},\theta) = \prod_{k=0}^{t} g(y_k|x_k,\theta). \tag{1.5}$$

[2]A predictive decomposition is just a useful way of writing the joint density of a sequence $Z_{0:k}$ as the product of its conditional densities. That is,

$$p(z_{0:k}) = p(z_0)\frac{p(z_{0:1})}{p(z_0)}\frac{p(z_{0:2})}{p(z_{0:1})}\cdots\frac{p(z_{0:k})}{p(z_{0:k-1})} = p(z_0)\prod_{j=1}^{k} p(z_0|z_{0:j-1}),$$

which follows by simple induction and the definition of a conditional density, i.e. $p(z_j|z_{0:j-1}) = p(z_j, z_{0:j-1})/p(z_{0:j-1}) = p(z_{0:j})/p(z_{0:j-1})$. Although we have assumed a specific ordering here, it is clear the the decomposition holds for any combination of disjoint subsets of $Z_{0:k}$.

(iv) The process $(X_t, Y_t)_{t\geq 0}$ is jointly Markovian, i.e. for $t \geq 1$,

$$p(x_t, y_t|x_{1:t-1}, y_{1:t-1}, \theta) = p(x_t, y_t|x_{t-1}, y_{t-1}, \theta). \tag{1.6}$$

Proof.

(i) The predictive decomposition of the law of $X_{0:t}$ is

$$p(x_{0:t}|\theta) = p(x_0|\theta) \prod_{k=1}^{t} p(x_k|x_{0:k-1}, \theta).$$

Now, $\nu(x_0|\theta) \equiv p(x_0|\theta)$ is the initial distribution of $(X_t)_{t\geq 0}$. Hence, by the Markov property (1.1) of this sequence we have that each $p(x_k|x_{0:k-1}, \theta) = p(x_k|x_{k-1}, \theta) \equiv f(x_k|x_{k-1}, \theta)$, yielding the desired result

$$p(x_{0:t}|\theta) = \nu(x_0|\theta) \prod_{k=1}^{t} f_\theta(x_k|x_{k-1}, \theta).$$

(ii) We can rewrite the probability density of $X_t|X_{k:t-1}, Y_{k:t-1}$ as

$$p(x_t|x_{k:t-1}, y_{k:t-1}, \theta) = \frac{p(y_{k:t-1}|x_t, x_{k:t-1}, \theta)p(x_t|x_{k:t-1}, \theta)p(x_{k:t-1}|\theta)}{p(y_{k:t-1}|x_{k:t-1}, \theta)p(x_{k:t-1}|\theta)}.$$

Since by (1.2) each Y_k depends only on X_k, we have that $p(y_{k:t-1}|x_t, x_{k:t-1}, \theta) = p(y_{k:t-1}|x_{k:t-1}, \theta)$. Further, from the Markov property (1.1), we have $p(x_t|x_{k:t-1}, \theta) = f(x_t|x_{t-1}, \theta)$ and, therefore, that

$$p(x_t|x_{k:t-1}, y_{k:t-1}, \theta) = \frac{p(y_{k:t-1}|x_{k:t-1}, \theta)f(x_t|x_{t-1}, \theta)p(x_{k:t-1}|\theta)}{p(y_{k:t-1}|x_{k:t-1}, \theta)p(x_{k:t-1}|\theta)} = f(x_t|x_{t-1}, \theta),$$

as required.

(iii) The joint density of $Y_{0:t}|X_{0:t}$ can be factored as

$$\begin{aligned} p(y_{0:t}|x_{0:t}, \theta) &= p(y_0|x_{0:t}, \theta)p(y_1|y_0, x_{0:t}, \theta) \cdots p(y_t|y_{0:t-1}, x_{0:t}, \theta) \\ &= g(y_0|x_0, \theta)g(y_1|x_1, \theta) \cdots g(y_t|x_t, \theta) = \prod_{k=0}^{t} g(y_k|x_k, \theta), \end{aligned}$$

which again follows from the fact that each Y_k depends only on X_k, established in (1.2).

(iv) Since the density of $(X_t, Y_t|X_{t-1}, Y_{t-1})$ admits the decomposition

$$p(x_t, y_t|x_{t-1}, y_{t-1}, \theta) = p(y_t|x_t, x_{t-1}, y_{t-1}, \theta)p(x_t|x_{t-1}, y_{t-1}, \theta),$$

which we know from (1.2) and item (ii) to be equal to $g(x_t|y_t, \theta) \cdot f(x_t|x_{t-1}, \theta)$, it suffices to show that $p(x_t, y_t|x_{0:t-1}, y_{0:t-1}) = g(x_t|y_t, \theta) \cdot f(x_t|x_{t-1}, \theta)$. First, notice that

$$p(x_t, y_t|x_{0:t-1}, y_{0:t-1}, \theta) = \frac{p(y_{0:t}|x_{0:t}, \theta)p(x_{0:t}|\theta)}{p(y_{0:t-1}|x_{0:t-1}, \theta)p(x_{0:t-1}|\theta)}.$$

Therefore, by items (i) and (iii),

$$p(x_t, y_t|x_{0:t-1}, y_{0:t-1}, \theta) = \frac{\prod_{k=0}^{t} g(y_k|x_k, \theta)}{\prod_{k=0}^{t-1} g(y_k|x_k, \theta)} \frac{\nu(x_0|\theta) \prod_{k=1}^{t} f(x_k|x_{k-1}, \theta)}{\nu(x_0|\theta) \prod_{k=1}^{t-1} f(x_k|x_{k-1}, \theta)}.$$

This ratio is equal to $g(x_t|y_t, \theta) \cdot f(x_t|x_{t-1}, \theta)$, completing the proof.

□

As a technical note, it must be pointed out that throughout the text we implicitly assume that all the conditional probability distributions we deal with always exist. Although in a more general setting this assumption can be dropped (Cappé et al., 2005), the tools required to provide a consistent definition of HMMs in this case are considerably more involved than the ones adopted here, and for ease of exposition we shall therefore avoid them.

In closing this section, note that since $\mathbb{P}_\theta$ is in a parametric family and since the law of each Y_t depends only on the corresponding state X_t, both θ and X_t can be thought of as model parameters in a general sense, with the Markov chain's *states* $(X_t)_{t\geq 0}$ typically being referred to as *dynamic* parameters and θ being referred to as *static* parameters. This naming convention reflects the fact that θ is a fixed (albeit usually unknown) quantity, whereas the states naturally vary over time. Hereafter we reserve the terms *state* inference and *parameter* inference to respectively distinguish between inference for the states and inference for the static parameters.

Chapter 2

State Inference

Upon observing a sample $Y_{1:n} = y_{1:n}$ from an HMM, we usually want to infer about the sequence of hidden states $X_{0:n}$ and static parameter values θ that are most consistent with this data, i.e. performing *state* and *parameter* inference, respectively. There are however instances where only state estimation is required, as in e.g. traditional applications in physics and biology, where θ represents a set of known physical and/or chemical constants. In this chapter, we will concern ourselves only with such chases: performing state inference conditional on complete knowledge of θ. Most of the material discussed here also serve as a building block for the subsequent parameter inference techniques we discuss in Chapter 3.

Now, state inference is usually classified into one of three main types, according to the level of information available at time t:

(i) *Prediction*: computing $X_t|Y_{1:t-1}$

(ii) *Filtering*: computing $X_t|Y_{1:t}$

(iii) *Smoothing*: computing $X_t|Y_{1:n}$

We will start our presentation of state inference techniques with the filtering problem, since prediction and smoothing can be derived directly from the filtering solution. Since θ is assumed to be fixed and known, throughout this chapter we will omit dependence on it to alleviate notation. Also, note that although no explicit statistical inference paradigm is assumed for performing state inference (since it can essentially be framed as a probabilistic problem), traditionally the terminology developed for it is almost entirely Bayesian in nature.

2.1 Filtering

State filtering in HMMs consists in computing the marginal posterior distribution $p(x_t|y_{1:t})$ for each t. Conceptually, this is a simple problem since if we have the joint posterior distribution of $X_{0:t}$ given[1] $Y_{1:t} = y_{1:t}$ we can simply integrate it over the image set of $X_{0:t-1}$, yielding

$$p(x_t|y_{1:t}) := \int_{\mathcal{X}^t} p(x_{0:t}|y_{1:t}) dx_{0:t-1}. \tag{2.1}$$

[1] In this work we always assume that only $Y_{1:t}$ is observed, so that inference for X_0 is based on its prior $\nu(x_0|\theta)$. This is equivalent to treating Y_0 as arbitrary or as the prior information about the model itself, so that e.g. $p(x_0|y_0, \theta) = \nu(x_0|\theta)$.

However, the problem with (2.1) is that in general neither $p(x_{0:t}|y_{1:t})$ nor its integral over $\mathcal{X}^t$ are available analytically. Moreover, since the dimension of the problem increases with t, techniques that directly approximate the integral in (2.1) (such as quadrature) usually perform very poorly as a result of the so-called *curse of dimensionality* phenomenom (see e.g. Asmussen and Glynn, 2007, p. 264, Liu, 2008, p. 2 and Robert and Casella, 2004, p. 136.).

An ingenious solution to the filtering problem, first made practical by Gordon et al. (1993) is to use *Monte Carlo* (MC) simulation techniques (see Appendix A for a brief review). More specifically, we rely on a subset of MC known generally as *Sequential Monte Carlo* (SMC), or *particle filters*, which are essentially *importance sampling* (IS) methods which fully exploit the sequential nature of HMMs in order to obtain substantial efficiency gains over competing alternatives.

As in the general IS case (see Section A.2), application of SMC simply requires us to be able to produce N random draws/*particles* $X^i_{0:t}|Y_{1:t}$, $i = 1, \ldots, N$ from a *proposal* distribution $q(x_{0:t}|y_{1:t})$ and evaluate, up to proportionality, the corresponding *unnormalized* and *normalized importance weights* $\pi^i_t \propto p(x^i_{0:t}|y_{1:t})/q(x^i_{0:t}|y_{1:t})$ and $w^i_t := \pi^i_t / \sum_{j=1}^N \pi^j_t$. Since clearly $w^i_t > 0$ for all i and $\sum_{i=1}^N w^i_t = 1$, the weighted sample $(X^i_{0:t}, w^i_t)_{i=1}^N$ forms a discrete distribution on the probability space generated by the HMM $(X_t, Y_t)_{t\geq 0}$ with probability measure denoted by $\hat{\mathbb{P}}$, i.e. such that $\hat{\mathbb{P}}(X_{0:t} = x^i_{0:t}) = w^i_t$ for each i and t.

The discrete measure $\hat{\mathbb{P}}$ is also absolutely continuous with respect to the same sigma-finite measure dominating $\mathbb{P}$ itself. Therefore, by the Radon-Nikodym theorem, it has a density with respect to this measure that we will denote by $\hat{p}$. It is easy to see that

$$\hat{p}(x_{0:t}|y_{1:t}) := \sum_{i=1}^N w^i_t \delta_{x^i_{0:t}}(dx_{0:t}), \tag{2.2}$$

since by definition the probability (under $\hat{\mathbb{P}}$) of $X_{0:t}$ being equal a single particle $x^j_{0:t}$ is given by w^j_t, which is equal to

$$\hat{\mathbb{P}}(X_{0:t} = x^j_{0:t}) = \int_{x^j_{0:t}} \sum_{i=1}^N w^i_t \delta_{x^i_{0:t}}(dx_{0:t}) dx_{0:t} = w^j_t.$$

Here, $\delta_a(dx)$ denotes a generic Dirac measure (or point mass) of an increment dx at the point a.

We commonly refer to $\hat{p}(x_{0:t}|y_{1:t})$ as the *Monte Carlo estimate* or *particle approximation* to the *target* density $p(x_{0:t}|y_{1:t})$. As in (2.1), the corresponding approximation to the filtering density is obtained by simply integrating over the joint approximation $\hat{p}(x_{0:t}|y_{1:t})$ over $X_{0:t-1}$, i.e.

$$\begin{aligned}
\hat{p}(x_t|y_{1:t}) &:= \int_{\mathcal{X}^t} \hat{p}(x_{0:t}|y_{1:t}) dx_{0:t-1} \\
&= \int_{\mathcal{X}^t} \sum_{i=1}^N w^i_t \delta_{x^i_{0:t}}(dx_{0:t}) dx_{0:t-1} \\
&= \int_{\mathcal{X}^t} \sum_{i=1}^N w^i_t \delta_{x^i_t}(dx_t) \delta_{x^i_{0:t-1}}(dx_{0:t-1}) dx_{0:t-1} \\
&= \sum_{i=1}^N w^i_t \delta_{x^i_t}(dx_t) \int_{\mathcal{X}^t} \delta_{x^i_{0:t-1}}(dx_{0:t-1}) dx_{0:t-1}
\end{aligned}$$

$$= \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(dx_t). \tag{2.3}$$

Note that from the second to the third equations above we have used the fact that any two point masses are always disjoint, i.e. $\delta_{(a,b)}(x,y) = \delta_a(x) \cdot \delta_b(y)$. The interchangeability of sums and integrals here is trivially justified by the fact that the sums are taken over a finite index $1 \leq i \leq N$.

2.1.1 Sequential Importance Sampling

We now turn to the problem of how to produce draws from $p(x_{0:t}|y_{1:t})$ and evaluate the corresponding importance weights $\pi_t := \pi(x_{0:t}, y_{1:t}) = p(x_{0:t}|y_{1:t})/q(x_{0:t}|y_{1:t})$. The first (and simplest) of the methods we explore here is aptly named *Sequential Importance Sampling*, or SIS for short.

As its name implies, SIS is an algorithm which is sequential in nature, enabling it to exploit the natural Markovian structure of HMMs in order to obtain efficiency gains both when sampling the particles and computing their weights. More precisely, in SIS we assume that the proposal distribution satisfies the following assumption.

Assumption 2.1.1 (SIS). *Under q, $Y_t|(X_{0:t-1}, Y_{1:t-1})$ is equal in distribution to $Y_t|Y_{1:t-1}$.*

A direct implication of the SIS Assumption 2.1.1 is that we can decompose the proposal distribution as

$$\begin{aligned}
q(x_{0:t}|y_{1:t}) &= \frac{q(x_{0:t}, y_{1:t})}{q(y_{1:t})} \\
&= \frac{q(x_t, x_{0:t-1}, y_t, y_{1:t-1})}{q(y_t, y_{1:t-1})} \\
&= \frac{q(x_t|x_{0:t-1}, y_t, y_{1:t-1}) q(y_t|x_{0:t-1}, y_{1:t-1}) q(x_{0:t-1}|y_{1:t-1}) q(y_{1:t-1})}{q(y_t|y_{1:t-1}) q(y_{1:t-1})} \\
&= q(x_{0:t-1}|y_{1:t-1}) q(x_t|x_{0:t-1}, y_{1:t}) \frac{q(y_t|x_{0:t-1}, y_{1:t-1})}{q(y_t|y_{1:t-1})} \\
&= q(x_{0:t-1}|y_{1:t-1}) q(x_t|x_{0:t-1}, y_{1:t}) \frac{q(y_t|y_{1:t-1})}{q(y_t|y_{1:t-1})} \\
&= q(x_{0:t-1}|y_{1:t-1}) q(x_t|x_{0:t-1}, y_{1:t}).
\end{aligned} \tag{2.4}$$

Essentially, Assumption 2.1.1 means that $X_{0:t}|Y_{1:t} =^d \{X_{0:t-1}|Y_{1:t-1}, X_t|(X_{0:t-1}, Y_{1:t})\}$, i.e. that the joint distribution of $X_{0:t}$ given $Y_{1:t}$ admits $X_{0:t-1}|Y_{1:t-1}$ and $X_t|(X_{0:t-1}, Y_{1:t})$ as its marginals. In practice, this means that we can sample the entire path $x_{0:t}^i$ by first sampling $x_t^i \sim q(x_t|x_{0:t-1}^i, y_{1:t})$ and then setting $x_{0:t}^i = (x_t^i, x_{0:t-1}^i)$.

Now, the efficiency gains of being able to marginally sample x_t^i conditional on its history $x_{0:t-1}^i$ should be very clear: at each step, SIS requires that we perform only $\mathcal{O}(N)$ operations instead of the $\mathcal{O}(tN)$ operations that we would usually require to sample the entire path $x_{0:t}^i$. If we have a sample of n observations $Y_{1:n} = y_{1:n}$, SIS therefore requires a total of $\mathcal{O}(nN)$ operations to sample the entire sequence of states $x_{0:n}^i$ rather than the $\mathcal{O}(n^2 N)$ that would be required if this was not done sequentially[2].

[2]The n^2 term arises due to $\mathcal{O}(N) + \mathcal{O}(2N) + \cdots + \mathcal{O}(nN) = \mathcal{O}(n^2 N)$, since the complexity when sampling the full $x_{0:t}^i$ at each time step is $\mathcal{O}(tN)$.

Another efficiency gain made possible by Assumption 2.1.1 is in computing the importance weights w_t. In SIS, we can evaluate importance weights recursively time in $\mathcal{O}(N)$ operations, again avoiding the $\mathcal{O}(tN)$ complexity required if this was not done sequentially. In order to achieve this, first note that we can decompose the target distribution $p(x_{0:t}|y_{1:t})$ as

$$\begin{aligned} p(x_{0:t}|y_{1:t}) &= \frac{p(x_{0:t}, y_{1:t})}{p(y_{1:t})} \\ &= \frac{p(x_t, x_{0:t-1}, y_t, y_{1:t-1})}{p(y_{1:t})} \\ &= \frac{p(y_t|x_t, x_{0:t-1}, y_{1:t-1})p(x_t|x_{0:t-1}, y_{1:t-1})p(x_{0:t-1}|y_{1:t-1})p(y_{1:t-1})}{p(y_t|y_{1:t-1})p(y_{1:t-1})}. \end{aligned}$$

Since by (1.2) Y_t depends only on X_t, we have $p(y_t|x_t, x_{0:t-1}, y_{1:t-1}) = g(y_t|x_t)$. Further, item (ii) of Proposition 1.1.1 applied with $k = 0$ yields $p(x_t|x_{0:t-1}, y_{1:t-1}) = f(x_t|x_{t-1})$ and, therefore, that

$$p(x_{0:t}|y_{1:t}) = p(x_{0:t-1}|y_{1:t-1})\frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|y_{1:t-1})}. \tag{2.5}$$

Combining equations (2.4) and (2.5), we therefore have a recursion for the unnormalized importance weights:

$$\begin{aligned} \pi_t &= \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})} \\ &= \frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1}|y_{1:t-1})}\frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|y_{1:t-1})q(x_t|x_{0:t-1}, y_{1:t})} \\ &= \pi_{t-1}\frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|y_{1:t-1})q(x_t|x_{1:t-1}, y_{1:t})}. \end{aligned} \tag{2.6}$$

Now, in general an analytical expression for $p(y_t|y_{1:t-1})$ is not available, thus making pointwise evaluation of π_t impossible. However, since this density does not depend on $x_{0:t}$, it can be essentially treated as a proportionality constant and "integrated out" by summing over the importance weights of all the sampled particles, as shown in Section A.2. More specifically, let $\pi_t^i := \pi(x_{0:t}^i|y_{1:t}) = p(x_{0:t}^i|y_{1:t})/q(x_{0:t}^i|y_{1:t})$. We then define $w_t^i := \pi_t^i / \sum_{j=1}^N \pi_t^j$ and, by (2.6), this is equivalent to

$$\begin{aligned} w_t^i &= \frac{\pi_t^i}{\sum_{j=1}^N \pi_t^j} \\ &= \frac{\pi_{t-1}^i \frac{f(x_t^i|x_{t-1}^i)g(y_t|x_t^i)}{p(y_t|y_{1:t-1})q(x_t^i|x_{0:t-1}^i, y_{1:t})}}{\sum_{j=1}^N \pi_{t-1}^j \frac{f(x_t^j|x_{t-1}^j)g(y_t|x_t^j)}{p(y_t|y_{1:t-1})q(x_t^j|x_{0:t-1}^j, y_{1:t})}} \\ &= \frac{\frac{1}{p(y_t|y_{1:t-1})}\pi_{t-1}^i \frac{f(x_t^i|x_{t-1}^i)g(y_t|x_t^i)}{q(x_t^i|x_{0:t-1}^i, y_{1:t})}}{\frac{1}{p(y_t|y_{1:t-1})}\sum_{j=1}^N \pi_{t-1}^j \frac{f(x_t^j|x_{t-1}^j)g(y_t|x_t^j)}{q(x_t^j|x_{0:t-1}^j, y_{1:t})}} \\ &= \frac{\pi_{t-1}^i \frac{f(x_t^i|x_{t-1}^i)g(y_t|x_t^i)}{q(x_t^i|x_{0:t-1}^i, y_{1:t})}}{\sum_{j=1}^N \pi_{t-1}^j \frac{f(x_t^j|x_{t-1}^j)g(y_t|x_t^j)}{q(x_t^j|x_{0:t-1}^j, y_{1:t})}}, \end{aligned}$$

as required. It is therefore common practice to simply define π_t up to a proportionality constant, since this constant is going to be factored out either way when we compute the normalized importance weights. In light of this fact, the weight recursion for π_t then takes its usual form

$$\pi_t \propto w_{t-1} \frac{f(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{0:t-1}, y_{1:t})}, \tag{2.7}$$

since w_{t-1}^i is itself proportional to π_{t-1}^i up to the constant $1/\sum_{j=1}^N \pi_{t-1}^j$.

Starting at $t = 0$, SIS is initialized by sampling $x_0^i \sim \nu(x_0)$ and setting $\pi_0^i \propto 1$ and therefore $w_0^i = 1/N$ for $i = 1, \ldots, N$. We then proceed sequentially by sampling $x_t^i \sim q(x_t|x_{0:t-1}^i, y_{1:t})$, computing $\pi_t^i \propto w_{t-1}^i f(x_t^i|x_{t-1}^i)g(y_t|x_t^i)/q(x_t^i|x_{0:t-1}^i, y_{1:t})$, normalizing $w_t^i = \pi_t^i/\sum_{j=1}^N \pi_t^j$ and setting $x_{0:t}^i = (x_t^i, x_{0:t-1}^i)$ for each $i = 1, \ldots, N$, until $t = n$. The entire procedure is summarized in Algorithm 2.1.

Algorithm 2.1: Sequential Importance Sampling (SIS)

Initialization
for $i = 1$ **to** N **do**
  draw $x_0^i \sim \nu(x_0)$
  set $\pi_0^i \propto 1$
end
for $i = 1$ **to** N **do**
  set $w_0^i = 1/N$
end

Main recursion
for $t = 1$ **to** n **do**
  for $i = 1$ **to** N **do**
    draw $x_t^i \sim q(x_t|x_{0:t-1}^i, y_{1:t})$
    compute $\pi_t^i \propto w_{t-1}^i f(x_t^i|x_{t-1}^i)g(y_t|x_t^i)/q(x_t^i|x_{0:t-1}^i, y_{1:t})$
  end
  for $i = 1$ **to** N **do**
    compute $w_t^i = \pi_t^i/\sum_{j=1}^N \pi_t^j$
  end
end

At the end of each step in Algorithm 2.1, we have weighted samples $(x_{0:t}^i, w_t^i)_{i=1}^N$ approximately distributed according to $X_{0:t}|Y_{1:t}$, resulting in a final sample $(x_{0:n}^i, w_n^i)_{i=1}^N$ approximating the law of $X_{0:n}|Y_{1:n}$. With these samples, we can compute the approximation to any of the features of the filtering density $p(x_t|y_{1:t})$. For example, if h is any $\mathbb{P}$-integrable and $\mathcal{B}$-measurable function, a particle approximation to $\mathbb{E}_{\mathbb{P}}[h(X_t)|Y_{1:t}] = \int_{\mathcal{X}} h(x_t)p(x_t|y_{1:t})dx_t$ is the weighted sum $\sum_{i=1}^N w_t^i h(x_t^i)$, obtained by simply replacing $p(x_t|y_{1:t})$ with our SIS estimate $\hat{p}(x_t|y_{1:t})$ given by (2.3) in the corresponding integral. More precisely,

$$\begin{aligned}\hat{\mathbb{E}}_{\mathbb{P}}[h(X_t)|Y_{1:t}] &\equiv \mathbb{E}_{\hat{\mathbb{P}}}[h(X_t)|Y_{1:t}] \\ &:= \int_{\mathcal{X}} h(x_t)\hat{p}(x_t|y_{1:t})dx_t\end{aligned}$$

$$
\begin{aligned}
&= \int_{\mathcal{X}} h(x_t) \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(dx_t) dx_t \\
&= \sum_{i=1}^{N} w_t^i \int_{\mathcal{X}} h(x_t) \delta_{x_t^i}(dx_t) dx_t \\
&= \sum_{i=1}^{N} w_t^i h(x_t^i).
\end{aligned}
$$

Now, although the moment approximation above was derived for a function of the filtered states $X_t|Y_{1:t}$, the same argument clearly holds for more general functions and functionals of the entire path $X_{0:t}$ or any of its subsets by making the appropriate substitutions (e.g. by replacing $\hat{p}(x_t|y_{1:t})$ with $\hat{p}(x_{0:t}|y_{1:t})$ as given in (2.2) for the entire path). Under the same conditions discussed in Section A.2, these moment estimators can also show to obey a law of large numbers and a central limit theorem. In particular, as $N \to +\infty$, the density estimators (2.3) and (2.2) approximate their targets $p(x_t|y_{1:t})$ and $p(x_{0:t}|y_{1:t})$ arbitrarily well.

Example 2.1.1 (Gaussian random walk). Consider the HMM defined by

$$
\begin{aligned}
X_t &= X_{t-1} + \tau U_t, & U_t &\sim N(0,1), & (2.8)\\
Y_t &= X_t + \sigma V_t, & V_t &\sim N(0,1),, & (2.9)
\end{aligned}
$$

where $\tau > 0$ and $\sigma > 0$ are scalars, $(U_t)_{t\geq 0}$ and $(V_t)_{t\geq 0}$ are mutually and serially independent (i.e. $U_t \perp\!\!\!\perp U_s$, $V_t \perp\!\!\!\perp V_s$ and $U_t \perp\!\!\!\perp V_s$ for all t and s) and the state prior is given by $X_0 \sim \mathcal{N}(0, \tau^2)$, where $\mathcal{N}(m, s^2)$ denotes a Normal distribution with mean m and variance s^2.

The model defined by (2.8-2.9) is known as a (Gaussian) *random walk plus noise*, since it is essentially a (Gaussian) random walk $(X_t)_{t\geq 0}$ that is only observed through $(Y_t)_{t\geq 0}$ with (Gaussian) noise σV_t. In the time series literature, this model is also known as the *Local Level Model* (Durbin and Koopman, 2012).

Since both the state transition and observation equations 2.8 and (2.9) are linear and Gaussian, the filtering distribution $X_t|Y_{1:t}$ can actually be computed exactly through the use of the celebrated *Kalman filter* (hereafter referred to as KF; see Appendix B for details). Although the existence of an analytical solution clearly eliminates the need for simulation-based techniques, this example is still useful as a benchmark to evaluate these methods against.

Figure 2.1 contains the observations (black points) and states (green solid line) of a simulated series of the Gaussian random walk model with $n = 200$ and $\theta = (\tau^2, \sigma^2) = (10, 1)$. For this particular series, the observations and states seem to be almost juxtaposed, which happens because the states are very informative relative to the observations. One way to assess this is by looking at the *signal-to-noise ratio* (SNR), which for this model is $\text{SNR} = \tau^2/\sigma^2 = 10/1 = 10$, implying that in this configuration the states are roughly 10 times more informative than the observations.

Intuitively, a large SNR also implies that the filtering task in a model is simpler, since the magnitude of the noise affecting observations of the underlying hidden states is then relatively small. Figure 2.2 shows that this is indeed the case for the Kalman filter (Algorithm B.1, initialized with $X_\nu = 0$ and $\Sigma_\nu = 10^7$), with the estimated states $X_{t|t}$ (solid blue lines) closely tracking the true states (black triangles). However, the

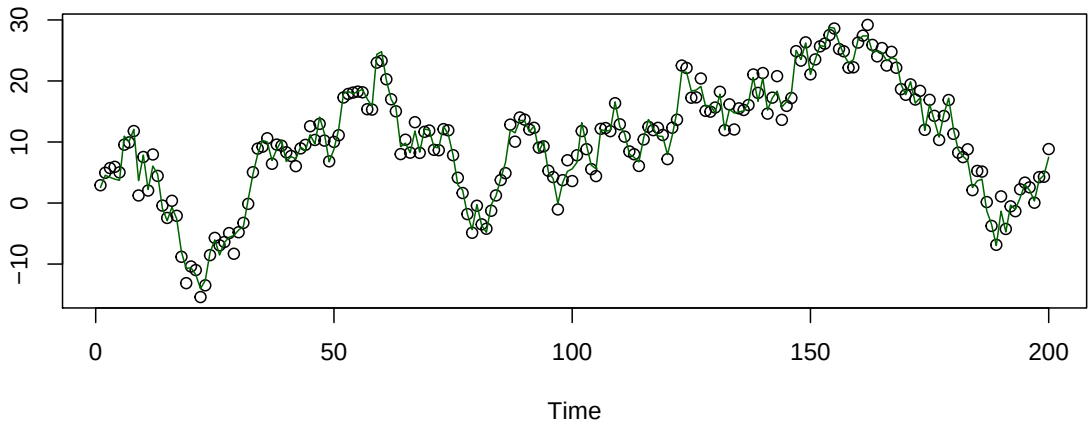

Figure 2.1: Simulated observations (black points) and states (green solid line) of the Gaussian random walk plus noise model (2.8-2.9) with $n = 200$, $\tau^2 = 10$, $\sigma^2 = 1$.

same does not hold for the SIS estimates $\hat{x}_t := \sum_{i=1}^N w_t^i x_t^i$ (solid red lines) produced by Algorithm 2.1 with $N = 10,000$ particles and implemented with proposal distribution $q(x_t|x_{0:t-1}, y_{1:t}) = f(x_t|x_{t-1}) = d\mathcal{N}(x_t|x_{t-1}, \tau^2)$, where $d\mathcal{N}(x|m, s^2)$ denotes the density of a Gaussian random variable with mean m and variance s^2 evaluated at the point x. Although for the first time indices there is a certain agreement between the SIS estimates and the true states, they quickly diverge from their target as time goes by. Note that in this problem the weight recursion (2.7) is

$$\pi_t \propto w_{t-1} \frac{f(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{1:t-1}, y_{1:t})} = w_{t-1} \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = w_{t-1} g(y_t|x_t),$$

where here $g(y_t|x_t) = d\mathcal{N}(y_t|x_t, \sigma^2)$.

Although here we could certainly improve the performance of SIS by considering "better" proposal distributions (in the sense of producing less variable importance weights – see Proposition 2.1.1 and the accompanying discussion) or even by increasing the number of simulated particles N, this does not address the root of the problem. We explain this more precisely in Section 2.1.2 below.

□

2.1.2 Sequential Importance Resampling

Although it might seem surprising at first, the poor performance of SIS shown in Example 2.1.1 is actually well-appreciated in the literature of sequential Monte Carlo methods, and is caused by what is known as the *weight degeneracy* or *particle degeneracy* (and sometimes simply *degeneracy*) phenomenom.

As its own name implies, degeneracy can be understood as a collapse of the weighted sample $(x_{0:t}^i, w_t^i)_{i=1}^N$ to a single particle $x_{0:t}^j$, with $w_t^j = 1$ and $w_t^i = 0$ for all $i \neq j$. Intuitively, this happens because as t increases, the weight recursion (2.7) assigns to the larger importance weights even larger importance weights, leading to the eventual collapse of the system.

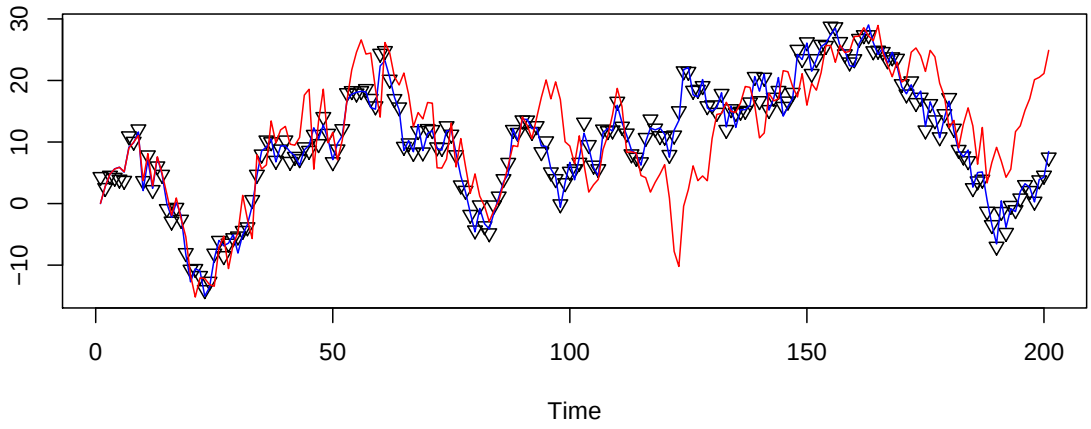

Figure 2.2: True states (black triangles) and corresponding Kalman filter (solid blue line) and SIS estimates (solid red line) for the Gaussian random walk plus noise model (2.8-2.9) with $n = 200$, $\tau^2 = 10$, $\sigma^2 = 1$ and $N = 10,000$.

Formally, the degeneracy of a particle system is defined as a property of the importance weight sequence $(\pi_t)_{t\geq 0}$, namely that its unconditional variance is nondecreasing with time, i.e. $\mathrm{var}(\pi_t) \geq \mathrm{var}(\pi_{t-1})$ for all $t \geq 1$. Although degeneracy has been noted earlier in the literature (see e.g. Gordon et al., 1993), this property was first established by Kong et al. (1994) in the form of the following theorem.

Theorem 2.1.1 (Kong, Liu and Wong, 1994). *The importance weights $(\pi_t)_{t\geq 0}$ form a martingale sequence in t. This implies that their variance is nondecreasing with time.*

Proof. The original proof in Kong et al. (1994) assumes a particular choice of proposal distribution (namely the *optimal proposal* discussed in Proposition 2.1.1), but the theorem is actually valid for any density q satisfying Assumption 2.1.1. Although this has been noted at least as early as Doucet et al. (2000), these authors do not provide an actual proof of this statement; we therefore provide one below.

First, consider the importance weights as an explicit function of the random variables (rather than their realized values) $X_{0:t}$ and $Y_{1:t}$, that is, $\pi_t \equiv \pi(X_{0:t}, Y_{1:t}) = p(X_{0:t}|Y_{1:t})/q(X_{0:t}|Y_{1:t})$. The definition of a martingale requires that $(\pi_t)_{t\geq 0}$ satisfy, for all t,

(i) $\mathbb{E}_{\mathbb{Q}}[|\pi(X_{0:t}, Y_{1:t})|] < +\infty$,

(ii) $\mathbb{E}_{\mathbb{Q}}[\pi(X_{0:t}, Y_{1:t})|\mathcal{F}_{t-1}] = \pi(X_{0:t-1}, Y_{1:t-1}) =: \pi_{t-1}$,

where $\mathcal{F}_{t-1} := \sigma(X_{0:t-1}, Y_{1:t-1})$ is the sigma-algebra generated by $(X_{0:t-1}, Y_{1:t-1})$ and $\mathbb{E}_{\mathbb{Q}}$ denotes expectation taken with respect to the measure $\mathbb{Q}$, defined such that q is the density of $\mathbb{Q}$.

For the first part, from $\pi_t \geq 0$ for all t (since π_t is the ratio of two probability densities, which are $\mathbb{Q}$-almost surely positive) clearly comes $|\pi_t| = \pi_t$. Without any loss of generality, we can by construction assume that the proposal q is equal to p whenever it is considered only as a function of the observations, i.e. $q(y_{1:t}) = p(y_{1:t})$, given that in

practice this does not affect the simulation of the particle system nor the computation of the importance weights. We then have that

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}}[\pi(X_{0:t}, Y_{1:t})] &= \int_{\mathcal{X}^{t+1}\times\mathcal{Y}^t} \pi(x_{0:t}, y_{1:t}) q(x_{0:t}, y_{1:t}) dx_{0:t} dy_{1:t} \\
&= \int_{\mathcal{X}^{t+1}\times\mathcal{Y}^t} \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})} q(x_{0:t}, y_{1:t}) dx_{0:t} dy_{1:t} \\
&= \int_{\mathcal{X}^{t+1}\times\mathcal{Y}^t} \frac{p(x_{0:t}, y_{1:t})}{p(y_{1:t})} \frac{q(y_{1:t})}{q(x_{0:t}, y_{1:t})} q(x_{0:t}, y_{1:t}) dx_{0:t} dy_{1:t} \\
&= \int_{\mathcal{X}^{t+1}\times\mathcal{Y}^t} \frac{p(x_{0:t}, y_{1:t})}{p(y_{1:t})} \frac{p(y_{1:t})}{q(x_{0:t}, y_{1:t})} q(x_{0:t}, y_{1:t}) dx_{0:t} dy_{1:t} \\
&= \int_{\mathcal{X}^{t+1}\times\mathcal{Y}^t} p(x_{0:t}, y_{1:t}) dx_{0:t} dy_{1:t} \\
&= 1 \\
&< +\infty.
\end{aligned}$$

For the second part, the weight recursion (2.6) implies that

$$\pi(X_{0:t}, Y_{1:t}) = \pi(X_{0:t-1}, Y_{1:t-1}) \frac{f(X_t|X_{t-1}) g(Y_t|X_t)}{p(Y_t|Y_{1:t-1}) q(X_t|X_{0:t-1}, Y_{1:t})},$$

where once again we use uppercase to emphasize the fact that the functions are taken with respect to the random variables $X_{0:t}$ and $Y_{1:t}$ themselves rather than their realized values. Taking the conditional expectation under $\mathbb{Q}$ of the above expression with respect to $\mathcal{F}_{t-1}$ then yields

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}}[\pi(X_{0:t}, Y_{1:t})|\mathcal{F}_{t-1}] &= \int_{\mathcal{X}\times\mathcal{Y}} \pi(x_t, X_{0:t-1}, y_t, Y_{1:t-1}) q(x_t, y_t|X_{0:t-1}, Y_{1:t-1}) dx_t dy_t \\
&= \pi(X_{0:t-1}, Y_{1:t-1}) \int_{\mathcal{X}\times\mathcal{Y}} \frac{f(x_t|X_{t-1}) g(y_t|x_t)}{p(y_t|Y_{1:t-1}) q(x_t|X_{0:t-1}, y_t, Y_{1:t-1})} q(x_t, y_t|X_{0:t-1}, Y_{1:t-1}) dx_t dy_t.
\end{aligned}$$

Now, we can decompose the integrating density as

$$\begin{aligned}
q(x_t, y_t|X_{0:t-1}, Y_{1:t-1}) &= q(x_t|X_{0:t-1}, y_t, Y_{1:t-1}) q(y_t|X_{0:t-1}, Y_{1:t-1}) \\
&= q(x_t|X_{0:t-1}, y_t, Y_{1:t-1}) q(y_t|Y_{1:t-1}),
\end{aligned}$$

where the last equality follows by the SIS Assumption 2.1.1. Since we assumed that $p = q$ when taken as a function of $Y_{1:t}$, we also have $q(y_t|Y_{1:t-1}) = p(y_t|Y_{1:t-1})$, which allows us to further write

$$q(x_t, y_t|X_{0:t-1}, Y_{1:t-1}) = q(x_t|X_{0:t-1}, y_t, Y_{1:t-1}) p(y_t|Y_{1:t-1}).$$

Finally, substituting the above equation in the expression for $\mathbb{E}_{\mathbb{Q}}[\pi(X_{0:t}, Y_{1:t})|\mathcal{F}_{t-1}]$ yields

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}}[\pi(X_{0:t}, Y_{1:t})|\mathcal{F}_{t-1}] &= \pi(X_{0:t-1}, Y_{1:t-1}) \cdot \\
&\cdot \int_{\mathcal{X}\times\mathcal{Y}} \frac{f(x_t|X_{t-1}) g(y_t|x_t)}{p(y_t|Y_{1:t-1}) q(x_t|X_{0:t-1}, y_t, Y_{1:t-1})} q(x_t|X_{0:t-1}, y_t, Y_{1:t-1}) p(y_t|Y_{1:t-1}) dx_t dy_t \\
&= \pi(X_{0:t-1}, Y_{1:t-1}) \int_{\mathcal{X}\times\mathcal{Y}} f(x_t|X_{t-1}) g(y_t|x_t) dx_t dy_t
\end{aligned}$$

$$
\begin{aligned}
&= \pi(X_{0:t-1}, Y_{1:t-1}) \int_{\mathcal{X}} f(x_t|X_{t-1}) \left(\int_{\mathcal{Y}} g(y_t|x_t) dy_t \right) dx_t \\
&= \pi(X_{0:t-1}, Y_{1:t-1}) \int_{\mathcal{X}} f(x_t|X_{t-1}) dx_t \\
&= \pi(X_{0:t-1}, Y_{1:t-1}) \\
&= \pi_{t-1},
\end{aligned}
$$

where the interchange between the order of integrability above is trivially justified by the fact that $f(x_t|X_{t-1})$ and $g(y_t|x_t)$ are both probability densities (since then both integrals are finite).

Having established that $(\pi_t)_{t\geq 0}$ is a martingale sequence, we can use the Law of Total Variance (Proposition C.1.2) to decompose

$$
\begin{aligned}
\text{var}_{\mathbb{Q}}(\pi_t) &= \text{var}_{\mathbb{Q}}[\mathbb{E}_{\mathbb{Q}}(\pi_t|\mathcal{F}_{t-1})] + \mathbb{E}_{\mathbb{Q}}[\text{var}_{\mathbb{Q}}(\pi_t|\mathcal{F}_{t-1})] \\
&= \text{var}_{\mathbb{Q}}(\pi_{t-1}) + \mathbb{E}_{\mathbb{Q}}[\text{var}_{\mathbb{Q}}(\pi_t|\mathcal{F}_{t-1})] \\
&\geq \text{var}_{\mathbb{Q}}(\pi_{t-1}),
\end{aligned}
$$

where the inequality follows from the fact that $\mathbb{E}_{\mathbb{Q}}[\text{var}_{\mathbb{Q}}(\pi_t|\mathcal{F}_{t-1})]$ is an almost surely-$\mathbb{Q}$ nonnegative variable (the expectation of a nonnegative random variable is always nonnegative, which by definition is true for $\text{var}_{\mathbb{Q}}(\pi_t|\mathcal{F}_{t-1})$ above).

□

Theorem 2.1.1 establishes that degeneracy is unavoidable in SIS, and Example 2.1.1 illustrates that even in simple settings the algorithm can exhibit poor performance as a result. In practice, degeneracy is the cost of the efficiency gains provided by the sequential nature of the algorithm.

Now, degeneracy is fundamentally caused by the weight updating mechanism assigning ever larger importance weights to an ever smaller number of particles. Heuristically, we could circumvent this by simply replicating the particles with the largest weights at each SIS step. This is indeed the most popular approach to dealing with degeneracy, and is known as *resampling*. Resampling essentially ensures (at least in probability) that the diversity[3] of the resulting particle set is then at least greater than or equal to what it would be without it. In essence, resampling is just sampling $(x^i_{0:t})_{i=1}^N$ with replacement from the discrete distribution $(x^i_{0:t}, w^i_t)_{i=1}^N$; hence the name.

More formally, resampling is a procedure in which we draw, for each $i = 1, \ldots, N$, a number of copies ξ^i_t of the particle $x^i_{0:t}$ with probability w^i_t and then let $(\tilde{x}^i_{0:t})_{i=1}^N$ be the set formed by the ξ^i_t copies of each $x^i_{0:t}$. In practice, this can be accomplished in several ways, of which the most common one is *multinomial resampling*. As its name implies, this method consists of drawing $(\xi^i_t)_{i=1}^N$ jointly from a Multinomial distribution with size N and probabilities $(w^i_t)_{i=1}^N$. Basic properties of Multinomial random variables allow us to easily establish that this scheme has both the desirable properties of *unbiasedness* and maintaining the size of the particle set intact, given that the expected number of copies

[3]Most of the research on resampling methods has its roots in *evolutionary optimization*, a field in which optimization techniques are inspired by the behavior of real-world biological systems (see e.g. Simon, 2013). The nomenclature around the various aspects of resampling therefore comes mostly from this field, particularly from the so-called genetical algorithms. Given that resampling is such an integral part of SMC methods, it is thus not uncommon to see terms such as "swarm", "fitness" and "diversity" in this context.

of each particle is exactly $N \cdot w_t^i$ and that in this case $(\xi_t^i)_{i=1}^N$ satisfy $\sum_{i=1}^N \xi_t^i = N$. We will discuss these properties in more detail in Section 3.2.2.

Resampling is clearly an instance of perfect Monte Carlo sampling (see Section A.1), since here both the proposal and the target are the same, i.e. the discrete distribution formed by $(w_t^i)_{i=1}^N$. An important (and often subtle) implication of this is that the resulting particle set $(\tilde{x}_{0:t}^i)_{1=1}^N$ is now an equally weighted sample from p, i.e. with uniform importance weights $(1/N)_{i=1}^N$. In practice this means that after resampling the importance weights w_t^i are "reset" to $w_t^i \leftarrow 1/N$. The SIS weight recursion (2.7) in this case becomes

$$\pi_t \propto \frac{1}{N} \frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{q(x_t|\tilde{x}_{0:t-1}, y_{1:t})} \propto 1 \cdot \frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{q(x_t|\tilde{x}_{0:t-1}, y_{1:t})} = \frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{q(x_t|\tilde{x}_{0:t-1}, y_{1:t})}. \tag{2.10}$$

The technique corresponding to SIS with resampling is aptly named *Sequential Importance Resampling* (SIR), and it is summarized in Algorithm 2.2. Note that the particles x_0^i drawn from the prior $\nu(x_0)$ are not resampled, i.e. no resampling takes place at $t = 0$.

Algorithm 2.2: Sequential Importance Resampling (SIR)

Initialization
for $i = 1$ **to** N **do**
  draw $x_0^i \sim \nu(x_0)$
  set $\pi_0^i \propto 1$
end
for $i = 1$ **to** N **do**
  set $w_0^i = 1/N$
end
for $i = 1$ **to** N **do**
  set $\tilde{x}_0^i = x_0^i$
end

Main recursion
for $t = 1$ **to** n **do**
  for $i = 1$ **to** N **do**
    draw $x_t^i \sim q(x_t|\tilde{x}_{0:t-1}^i, y_{1:t})$
    compute $\pi_t^i \propto f(x_t^i|\tilde{x}_{t-1}^i)g(y_t|x_t^i)/q(x_t^i|\tilde{x}_{0:t-1}^i, y_{1:t})$
  end
  for $i = 1$ **to** N **do**
    compute $w_t^i = \pi_t^i / \sum_{j=1}^N \pi_t^j$
  end
  for $i = 1$ **to** N **do**
    sample $\tilde{x}_{0:t}^i$ from $(x_{0:t}^i, w_t^i)_{i=1}^N$
  end
end

Now, although the resampled set $(\tilde{x}_{0:t}^i, w_t^i)_{i=1}^N$ produced by SIR is also an approximate sample from $p(x_{0:t}|y_{1:t})$, it is usually not desirable to use this sample to directly estimate features of the target density. Recalling that our SIS moment estimator of $\mathbb{E}_{\mathbb{P}}[h(X_t)|Y_{1:t}]$ is given by $\sum_{i=1}^N w_t^i h(x_t^i)$, Carpenter et al. (1999) proved that the variance (under $\mathbb{Q}$)

of the corresponding SIR estimator $\sum_{i=1}^{N} N^{-1} h(\tilde{x}_t^i)$ is always larger than the variance of SIS-based estimator, with the relative difference being as large as 100% in the case where $(x_{0:t}^i)_{i=1}^N$ is already equally weighted prior to resampling. This basically occurs due to the fact that resampling is in itself a stochastic procedure, and thus inherently introduces additional Monte Carlo variance in the system. In practice, when using SIR to build particle approximations of functionals of $X_t|Y_{1:t}$, these approximations should therefore be computed based on the weighted sample $(x_{0:t}^i, w_t^i)_{i=1}^N$, i.e. just prior to resampling.

Despite the fact that resampling is relatively simple to explain and implement, it complicates the underlying theory of SMC considerably, since although each x_t^i is still drawn independently from q, the particles $x_{0:t}^i$ are no longer independent. It can still be proven, however, (see e.g. Crisan and Doucet 2002, Chopin et al. 2004 and Del Moral 2004) that the usual asymptotic results for the general Importance Sampling algorithm discussed in Section A.2 (and which are also clearly valid for SIS) still hold under resampling.

At this point it is important to point out that, even in SIR, degeneracy is unavoidable, since in general resampling does not change the fact that the sequence of importance weights $(\pi_t)_{t\geq 0}$ still constitutes a martingale in t. However, as the next example illustrates, resampling is usually effective enough in mitigating degeneracy to make SMC-based inference feasible (and often quite successful) in practice.

Example 2.1.2 (Gaussian random walk, continued). Let us return to the Gaussian random walk plus noise model (2.8-2.9) of Example 2.1.1. Figure 2.3 contains the true states (black triangles) along with the corresponding Kalman Filter (Algorithm B.1) estimates $X_{t|t}$ (solid thicker blue lines) and SIR (Algorithm 2.2) estimates $\hat{x}_t = \sum_{i=1}^{N} w_t^i x_t^i$ (solid thinner red lines) for each $t = 0, \ldots, n$. The simulated series is the same as in Example 2.1.1, i.e. with $\sigma^2 = 1$ and $\tau^2 = 10$, and for SIR we once again use the state transition proposal $q(x_t|x_{1:t-1}, y_{1:t}) = f(x_t|x_{t-1})$ and $N = 10,000$ particles. The resampling step of SIR was implemented with the alias method of Vose (1991), designed to be a fast $\mathcal{O}(N)$ implementation of multinomial resampling (see Algorithm 3.5). Note that for this model the SIR weight recursion (2.7) is

$$\pi_t \propto \frac{f(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{1:t-1}, y_{1:t})} = \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t),$$

where $g(y_t|x_t) = d\mathcal{N}(y_t|x_t, \sigma^2)$.

By comparing Figures 2.2 and 2.3, we can see the clear difference in the performance between SIS and SIR in this example. The SIS estimates quickly diverge from the Kalman Filter estimates, whereas there seems to be almost a juxtaposition between the Kalman Filter estimates and the SIR estimates (the Pearson correlation between them diverges from 1 by $5 \cdot 10^{-7}$). The difference between SIS and SIR here is entirely attributable to degeneracy, which although not entirely eliminated from the problem is successfully mitigated by resampling.

□

Now, a natural question that arises here is how to objectively measure the degree of degeneracy to which a particular sample $(x_{0:t}^i, w_t^i)_{i=1}^N$ is subject to. Since degeneracy is associated with an increase in variance of the importance weights over time, heuristically a metric for it should involve an estimate of this variance. This is exactly the case of the so-called *effective sample size* (ESS) of Kong et al. (1994), defined by

$$\mathrm{ESS}_t \equiv \mathrm{ESS}(\pi_t) := \frac{1}{1 + \mathrm{var}_{\mathbb{Q}}(\pi_t)}. \tag{2.11}$$

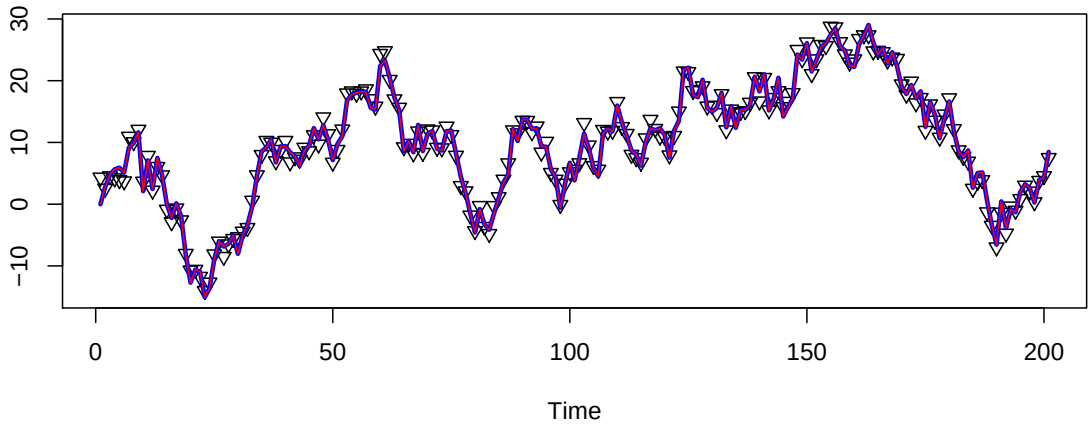

Figure 2.3: True states (black triangles) and corresponding Kalman filter (solid thick blue line) and SIR estimates (solid thin red line) for the Gaussian random walk plus noise model (2.8-2.9) with $n = 200$, $\tau^2 = 10$, $\sigma^2 = 1$ and $N = 10,000$.

Since $0 \leq \text{var}_{\mathbb{Q}}(\pi_t) < +\infty$, ESS_t takes values in the $(0, 1]$ interval. By analogy with the basic theory of Analysis of Variance and Experimental Design (see e.g. Montgomery, 2018), the effective sample size defined by (2.11) can be interpreted as the proportion of independent and identically distributed (iid) samples from the target p that would be required to convey the same information contained in the sample with importance weight π_t. Therefore, a small ESS_t indicates a strong impact of weight degeneracy in the particle system and typically a less efficient SMC procedure as a result.

In practice, we must estimate ESS_t, given that an analytic expression for $\text{var}_{\mathbb{Q}}(\pi_t)$ is in general not available. To accomplish this, first note that

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}}(\pi_t) &= \int_{\mathcal{X}^{t+1}\times\mathcal{Y}^t} \pi_t q(x_{0:t}, y_{1:t}) dx_{0:t} dy_{1:t} \\
&= \int_{\mathcal{X}^{t+1}\times\mathcal{Y}^t} \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})} q(x_{0:t}, y_{1:t}) dx_{0:t} dy_{1:t} \\
&= \int_{\mathcal{X}^{t+1}\times\mathcal{Y}^t} \frac{p(x_{0:t}|y_{1:t})}{\frac{q(x_{0:t},y_{1:t})}{q(y_{1:t})}} q(x_{0:t}, y_{1:t}) dx_{0:t} dy_{1:t} \\
&= \int_{\mathcal{Y}^t} \left(\int_{\mathcal{X}^{t+1}} p(x_{0:t}|y_{1:t}) dx_{0:t} \right) q(y_{1:t}) dy_{1:t} \\
&= \int_{\mathcal{Y}^t} 1 \cdot q(y_{1:t}) dy_{1:t} \\
&= 1,
\end{aligned}$$

which, since $\mathbb{E}_{\mathbb{Q}}(\pi_t^2) = [\mathbb{E}_{\mathbb{Q}}(\pi_t)]^2 + \text{var}_{\mathbb{Q}}(\pi_t)$, implies that $\mathbb{E}_{\mathbb{Q}}(\pi_t^2) = 1^2 + \text{var}_{\mathbb{Q}}(\pi_t) = 1 + \text{var}_{\mathbb{Q}}(\pi_t)$. By the Weak Law of Large Numbers (WLLN)[4], we then have $N^{-1} \sum_{i=1}^{N} (\pi_t^i)^2 \to^{\mathbb{P}}$

[4]Although the WLLN results used here would be usually stated for convergence in probability under $\mathbb{Q}$ (which is the measure that we are taking the expectations with respect to), we are implicitly using the fact that the convergence also holds for $\mathbb{P}$, given that in the Importance Sampling framework considered here (see Section A.2) the proposal measure $\mathbb{Q}$ is assumed to dominate $\mathbb{P}$.

$\mathbb{E}_{\mathbb{Q}}(\pi_t^2) = 1 + \text{var}_{\mathbb{Q}}(\pi_t)$ and, by continuous mapping, that $[N^{-1}\sum_{i=1}^N(\pi_t^i)^2]^{-1} \to^{\mathbb{P}} [1 + \text{var}_{\mathbb{Q}}(\pi_t)]^{-1} = \text{ESS}_t$, which establishes that $[N^{-1}\sum_{i=1}^N(\pi_t^i)^2]^{-1}$ is a consistent estimator of ESS_t (here $\to^{\mathbb{P}}$ denotes convergence in probability under $\mathbb{P}$, and in general an estimator $\hat{\theta}$ is consistent for θ if $\hat{\theta} \to^{\mathbb{P}} \theta$).

Now, given that we can usually only evaluate π_t^i up to proportionality, the estimator based on $\sum_{i=1}^N(\pi_t^i)^2$ also can't be used in practice. The solution then is to turn to an analogous estimator based on the normalized importance weights w_t^i, the ones that we can indeed routinely compute. Noting that

$$\begin{aligned} N \cdot \sum_{i=1}^N (w_t^i)^2 &= N \cdot \sum_{i=1}^N \left(\frac{\pi_t^i}{\sum_{j=1}^N \pi_t^j} \right)^2 \\ &= N \cdot \sum_{i=1}^N \left(\frac{\pi_t^i/N}{\sum_{j=1}^N \pi_t^j/N} \right)^2 \\ &= N \cdot \sum_{i=1}^N \frac{(\pi_t^i)^2/N^2}{(\sum_{j=1}^N \pi_t^j/N)^2} \\ &= \frac{\sum_{i=1}^N (\pi_t^i)^2/N}{(\sum_{j=1}^N \pi_t^j/N)^2}, \end{aligned}$$

the numerator $\sum_{i=1}^N(\pi_t^i)^2/N$ converges to $1 + \text{var}_{\mathbb{Q}}(\pi_t)$, as established before. For the denominator, the WLLN implies that $\sum_{j=1}^N \pi_t^j/N \to^{\mathbb{P}} \mathbb{E}_{\mathbb{Q}}(\pi_t) = 1$, which by continuous mapping then implies that $(\sum_{j=1}^N \pi_t^j/N)^2 \to^{\mathbb{P}} 1^2 = 1$. Finally, by applying *Slutsky's theorem* (Shao, 2003, p. 60) we have that the ratio between these quantities converges in probability under $\mathbb{P}$ to $[1 + \text{var}_{\mathbb{Q}}(\pi_t)]/1 = 1 + \text{var}_{\mathbb{Q}}(\pi_t)$, which by continuous mapping then ensures that $[N \cdot \sum_{i=1}^N (w_t^i)^2]^{-1} \to^{\mathbb{P}} 1 + \text{var}_{\mathbb{Q}}(\pi_t) = \text{ESS}_t$.

In summary, our consistent estimator of the Effective Sample Size (2.11) is defined by

$$\widehat{\text{ESS}}_t := \frac{1}{N \cdot \sum_{i=1}^N (w_t^i)^2}. \tag{2.12}$$

Note that, unlike the true ESS, $\widehat{\text{ESS}}_t$ actually takes its values in the $[1/N, 1]$ interval. However, as $N \to +\infty$, this interval does converge to the correct one, i.e. $(0, 1]$.

Example 2.1.3 (Gaussian random walk, continued). Returning to Example 2.1.2, Figure 2.4 contains the estimated $\widehat{\text{ESS}}_t$ given in (2.12) for both SIS (upper panel) and SIR (lower panel). We can see that the ESS for SIS drops exponentially fast to $1/N = 10^{-4}$ and then becomes stationary at this value, whereas the ESS for SIR fluctuates around its mean of about 0.28. This again illustrates that resampling in SMC is indeed effective in mitigating degeneracy, contributing to an increase in diversity in the particle set and thus helping to avoid its collapse.

□

Although resampling had already been around in the non-sequential inference literature (see e.g. Smith and Gelfand, 1992), its first appearance in the SMC context is made in the seminal paper of Gordon et al. (1993) with the introduction of the so-called *Bootstrap Filter*. The bootstrap filter is a special case of SIR in which the proposal is $q(x_t|x_{0:t-1}, y_{1:t}) = f(x_t|x_{t-1})$. In this case, the weight recursion (2.10) becomes

$$\pi_t \propto \frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{q(x_t|\tilde{x}_{0:t-1}, y_{1:t})} = \frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{f(x_t|\tilde{x}_{t-1})} = g(y_t|x_t). \tag{2.13}$$

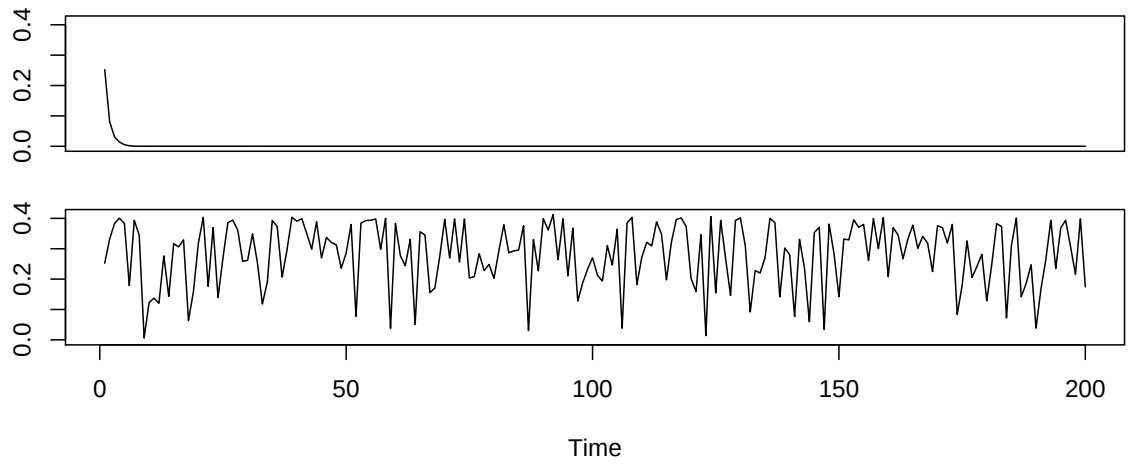

Figure 2.4: Estimated effective sample sizes for SIS (top row) and SIR (bottom row) for the Gaussian random walk plus noise model (2.8-2.9) with $n = 200$, $\tau^2 = 10$, $\sigma^2 = 1$ and $N = 10,000$.

The simple form of (2.13) allows the implementation of the bootstrap filter in situations where samples from $f(x_t|x_{t-1})$ can be produced but this density cannot be evaluated pointwise. An important example of this instance is the case of discretely-observed diffusions (Fearnhead et al., 2008).

Another important special case of SIR is the so-called *Optimal SIR* (Petetin and Desbouvries, 2013), in which we choose $q(x_t|x_{0:t-1}, y_{1:t}) = p(x_t|x_{t-1}, y_t)$ as proposal distribution. The term "optimal" here is understood in the sense of Proposition A.2.1, i.e. that the variance of the importance weights under this proposal is minimal. To formally establish that $p(x_t|x_{t-1}, y_t)$ is indeed optimal, we state and prove the following result (which is the analog of A.2.1 in SMC), due to Doucet et al. (2000).

Proposition 2.1.1. *The proposal distribution $q(x_t|x_{0:t-1}, y_{1:t}) = p(x_t|x_{t-1}, y_t)$ minimizes the conditional variance (under $\mathbb{Q}$) of the importance weights π_t given $X_{0:t-1}$ and $Y_{1:t-1}$.*

Proof. We will first prove the result for the SIS case and extend it to SIR afterwards. Let us decompose $p(x_t|x_{t-1}, y_t)$ as

$$p(x_t|x_{t-1}, y_t) = \frac{p(x_t, x_{t-1}, y_t)}{p(x_{t-1}, y_t)} = \frac{p(y_t|x_t, x_{t-1})p(x_t|x_{t-1})p(x_{t-1})}{p(y_t|x_{t-1})p(x_{t-1})} = \frac{g(y_t|x_t)f(x_t|x_{t-1})}{p(y_t|x_{t-1})},$$

where the last equality follows from the fact that $Y_t|(X_t, X_{t-1}) =^d Y_t|X_t$, as established in (1.2). This allows us to write the weight recursion (2.7) as

$$\begin{aligned}
\pi_t &\propto \pi_{t-1} \frac{f(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{0:t-1}, y_{1:t})} \\
&= \pi_{t-1} \frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(x_t|x_{t-1}, y_t)} \\
&= \pi_{t-1} \frac{f(x_t|x_{t-1})g(y_t|x_t)}{\frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|x_{t-1})}}
\end{aligned}$$

$$= \pi_{t-1} p(y_t|x_{t-1}). \tag{2.14}$$

Since (2.14) does not depend on x_t, the variance of π_t under the proposal $p(x_t|x_{t-1}, y_t)$ equals zero, which is by definition the minimum variance attainable for any random variable.

More explicitly, by taking $\pi_t := \pi(X_{0:t}, Y_{1:t})$ as a function of the random variables $X_{0:t}$ and $Y_{1:t}$, we have that

$$\begin{aligned}
\mathrm{var}_{\mathbb{Q}}[\pi(X_{0:t}, Y_{1:t})|X_{0:t-1}, Y_{1:t}] = \\
&= \mathbb{E}_{\mathbb{Q}}[\pi^2(X_{0:t}, Y_{1:t})|X_{0:t-1}, Y_{1:t}] - \{\mathbb{E}_{\mathbb{Q}}[\pi(X_{0:t}, Y_{1:t})|X_{0:t-1}, Y_{1:t}]\}^2 \\
&= \int_{\mathcal{X}} [\pi(x_t, X_{0:t-1}, Y_{1:t})]^2 q(x_t|X_{0:t-1}, Y_{1:t}) dx_t + \\
&\qquad - \left\{ \int_{\mathcal{X}} \pi(x_t, X_{0:t-1}, Y_{1:t}) q(x_t|X_{0:t-1}, Y_{1:t}) dx_t \right\}^2 \\
&= \int_{\mathcal{X}} [\pi_{t-1} p(Y_t|X_{t-1})]^2 p(x_t|X_{t-1}, Y_t) dx_t + \\
&\qquad - \left\{ \int_{\mathcal{X}} \pi_{t-1} p(Y_t|X_{t-1}) p(x_t|X_{t-1}, Y_t) dx_t \right\}^2 \\
&= [\pi_{t-1} p(Y_t|X_{t-1})]^2 \int_X p(x_t|X_{t-1}, Y_t) dx_t + \\
&\qquad - [\pi_{t-1} p(Y_t|X_{t-1})]^2 \left\{ \int_X p(x_t|X_{t-1}, Y_t) dx_t \right\}^2 \\
&= [\pi_{t-1} p(Y_t|X_{t-1})]^2 \cdot 1 - [\pi_{t-1} p(Y_t|X_{t-1})]^2 \cdot 1^2 \\
&= [\pi_{t-1} p(Y_t|X_{t-1})]^2 - [\pi_{t-1} p(Y_t|X_{t-1})]^2 \\
&= 0.
\end{aligned}$$

The proof in the SIR case is essentially the same, given that (2.10) and (2.7) fundamentally differ only by the factor π_{t-1}, which does not depend on x_t.

□

It is interesting to note that, even under the optimal proposal, the particle system is still subject to degeneracy. Remember from the proof of Theorem 2.1.1 that

$$\mathrm{var}_{\mathbb{Q}}(\pi_t) = \mathrm{var}_{\mathbb{Q}}(\pi_{t-1}) + \mathbb{E}_{\mathbb{Q}}[\mathrm{var}_{\mathbb{Q}}(\pi_t|\mathcal{F}_{t-1})],$$

where $\mathcal{F}_{t-1} := \sigma(X_{0:t-1}, Y_{1:t-1})$. Under the optimal proposal, by using the target and proposal recursions (2.5) and (2.4) we can use induction to write the importance weights as

$$\begin{aligned}
\pi_t &= \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})} \\
&= \frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1}|y_{1:t-1})} \frac{f(x_t|x_{t-1}) g(y_t|x_t)}{q(x_t|x_{1:t-1}, y_{1:t})} \frac{1}{p(y_t|y_{1:t-1})} \\
&= \frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1}|y_{1:t-1})} \frac{f(x_t|x_{t-1}) g(y_t|x_t)}{p(x_t|x_{t-1}, y_t)} \frac{1}{p(y_t|y_{1:t-1})} \\
&= \frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1}|y_{1:t-1})} \frac{f(x_t|x_{t-1}) g(y_t|x_t)}{\frac{f(x_t|x_{t-1}) g(y_t|x_t)}{p(y_t|x_{t-1})}} \frac{1}{p(y_t|y_{1:t-1})}
\end{aligned}$$

$$= \frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1}|y_{1:t-1})} \frac{p(y_t|x_{t-1})}{p(y_t|y_{1:t-1})}$$

$$\dots$$

$$= \frac{p(x_{0:1}|y_1)}{q(x_{0:1}|y_1)} \prod_{k=2}^{t} \frac{p(y_k|x_{k-1})}{p(y_k|y_{1:k-1})}$$

$$= \frac{p(y_1|x_1, x_0)p(x_1|x_0)p(x_0)}{p(y_1)} \frac{q(y_1)}{q(x_1|x_0, y_1)q(y_1|x_0)q(x_0)} \prod_{k=2}^{t} \frac{p(y_k|x_{k-1})}{p(y_k|y_{1:k-1})}.$$

Now, since by (1.2) comes $p(y_1|x_1, x_0) = g(y_1|x_1)$, by definition $p(x_0) = \nu(x_0)$, by (1.1) $p(x_1|x_0) = f(x_1|x_0)$, by Assumption 2.1.1 $q(y_1|x_0) = q(y_1)$ and by construction $q(x_1|x_0, y_1) = p(x_1|x_0, y_1) = f(x_1|x_0)g(y_1|x_1)/p(y_1|x_0)$, $q(y_1) = p(y_1)$ and $q(x_0) = \nu(x_0)$, this expression simplifies to

$$\pi_t = \frac{g(y_1|x_1)f(x_1|x_0)\nu(x_0)}{p(y_1)} \frac{p(y_1)}{\frac{f(x_1|x_0)g(y_1|x_1)}{p(y_1|x_1)} p(y_1)\nu(x_0)} \prod_{k=2}^{t} \frac{p(y_k|x_{k-1})}{p(y_k|y_{1:k-1})}$$

$$= \frac{p(y_1|x_0)}{p(y_1)} \prod_{k=2}^{t} \frac{p(y_k|x_{k-1})}{p(y_k|y_{1:k-1})},$$

implying that only in the case when the ratio above is not a function of y_t does the term $\mathrm{var}_{\mathbb{Q}}(\pi_t|\mathcal{F}_{t-1})$ vanishes, avoiding the increase in variance that defines degeneracy.

Finally, the weight recursion (2.10) for optimal SIR becomes

$$\pi_t \propto \frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{q(x_t|\tilde{x}_{0:t-1}, y_{1:t})} = \frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{p(x_t|\tilde{x}_{t-1}, y_t)} = \frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{\frac{f(x_t|\tilde{x}_{t-1})g(y_t|x_t)}{p(y_t|\tilde{x}_{t-1})}} = p(y_t|\tilde{x}_{t-1}). \quad (2.15)$$

Note that for implementation of optimal SIR to be feasible the ability of sampling from $p(x_t|x_{t-1}, y_t)$ and evaluating $p(y_t|x_{t-1})$ pointwise are both required. Although this severely limits the applicability of this optimal choice in practice, Proposition 2.1.1 suggests that it still should be made whenever possible.

2.1.3 Auxiliary Particle Filter

A convenient, popular and quite general alternative framework for including resampling into SMC methods is the *Auxiliary Particle Filter* (APF) of Pitt and Shephard (1999). They introduce an additional auxiliary variable k (hence the name) taking values into $\{1, \dots, N\}$ such that $(x_{l:t-1}, k) := x^k_{l:t-1}$ for integer $0 \leq l \leq t-1$ and, in particular, $(x_{0:t-1}, k) = x^k_{0:t-1}$ and $(x_{t-1}, k) = x^k_{t-1}$. Resampling in this setting then consists of simply sampling k with replacement from $\{1, \dots, N\}$ with corresponding probability λ^k_t (aptly named *intermediate weight*) and setting $\tilde{x}_{0:t-1} := (x_{0:t-1}, k) = x^k_{0:t-1}$. Note however that, unlike SIR, the APF resampling step thus takes place *before* the propagation/sampling of the states X_t.

The main reason for reversing the resampling order in APF is the so-called property of *adaptation*, defined by Pitt and Shephard (1999) as the ability of the filter to include current information about the observations Y_t *prior* to sampling the states X_t. Intuitively, this in most cases avoids problems with outlying observations and noninformative conditional likelihoods $g(y_t|x_t)$, since the system has a chance to "adapt" to the current

observation and thus (possibly) lead to a more representative sample $(x_{0:t}^i, w_t^i)_{i=1}^N$. Some authors refer to the APF as a *resample-propagate* framework and to SIR as a *propagate-resample* framework in order to make the resampling and propagation order explicit.

Let us now derive the filtering recursions for APF. Since here we have to perform inference for both $X_{0:t}$ and k, the proposal distribution is now a function of $(X_{0:t}, k)$, and assumed to satisfy

$$q(x_{0:t}, k|y_{1:t}) = q(x_{0:t-1}^k|y_{1:t})q(x_t|x_{0:t-1}^k, y_{1:t})\lambda_t^k, \tag{2.16}$$

where $\lambda_t^k := q(k|x_{0:t-1}, y_{1:t})$, the intermediate weight, is simply the marginal proposal for k conditional on $X_{0:t-1}$ and $Y_{1:t}$. Alternatively, analogous to SIR we can derive the recursion (2.16) explicitly. First, write

$$\begin{aligned} q(x_{0:t}, k|y_{1:t}) &= \frac{q(x_t|x_{0:t-1}, k, y_{1:t})q(y_t|x_{0:t-1}, k, y_{1:t-1})q(x_{0:t-1}, k|y_{1:t-1})q(y_{1:t-1})}{q(y_{1:t})} \\ &= \frac{q(x_t|x_{0:t-1}^k, y_{1:t})q(y_t|x_{0:t-1}, k, y_{1:t-1})q(x_{0:t-1}^k|y_{1:t-1})q(y_{1:t-1})}{q(y_t|y_{1:t-1})q(y_{1:t-1})} \\ &= q(x_{0:t-1}^k|y_{1:t-1})q(x_t|x_{0:t-1}^k, y_{1:t})\frac{q(y_t|x_{0:t-1}, k, y_{1:t-1})}{q(y_t|y_{1:t-1})} \end{aligned}$$

since $(x_{0:t-1}, k) = x_{0:t-1}^k$ (note that we chose to leave $(x_{0:t-1}, k)$ explicit in the numerator of the third term on the right side). Then, we decompose the last term in the above equation as

$$\begin{aligned} \frac{q(y_t|x_{0:t-1}, k, y_{1:t-1})}{q(y_t|y_{1:t-1})} &= \frac{q(x_{0:t-1}, k, y_{1:t})}{q(x_{0:t-1}, k, y_{1:t-1})q(y_t|y_{1:t-1})} \\ &= \frac{q(k|x_{0:t-1}, y_{1:t})q(y_t|x_{0:t-1}, y_{1:t-1})q(x_{0:t-1}, y_{1:t-1})}{q(k|x_{0:t-1}, y_{1:t-1})q(x_{0:t-1}, y_{1:t-1})q(y_t|y_{1:t-1})} \\ &= \frac{q(k|x_{0:t-1}, y_{1:t})q(y_t|x_{0:t-1}, y_{1:t-1})}{q(k|x_{0:t-1}, y_{1:t-1})q(y_t|y_{1:t-1})} \end{aligned}$$

and, since from Assumption 2.1.1 comes $q(y_t|x_{0:t-1}, y_{1:t-1}) = q(y_t|y_{1:t-1})$, this can be further rewritten as

$$\frac{q(y_t|x_{0:t-1}, k, y_{1:t-1})}{q(y_t|y_{1:t-1})} = \frac{q(k|x_{0:t-1}, y_{1:t})q(y_t|y_{1:t-1})}{q(k|x_{0:t-1}, y_{1:t-1})q(y_t|y_{1:t-1})} = \frac{q(k|x_{0:t-1}, y_{1:t})}{q(k|x_{0:t-1}, y_{1:t-1})}.$$

Now, although we cannot simplify this ratio further, note that it is proportional to the numerator $q(k|x_{0:t-1}, y_{1:t})$, since the denominator $q(k|x_{0:t-1}, y_{1:t-1})$ is not a function of y_t. We therefore have

$$\frac{q(y_t|x_{0:t-1}, k, y_{1:t-1})}{q(y_t|y_{1:t-1})} \propto q(k|x_{0:t-1}, y_{1:t}) = \lambda_t^k,$$

which then finally establishes[5] (2.16). As mentioned above, the process for sampling $(x_{0:t}^i, k_i)$ for each i in the APF then consists of sampling k_i (i.e. resampling), sampling

[5] As a technical note, it is worth pointing out that in this process we only specifiy $q(x_{0:t}, k|y_{1:t})$ up to proportionality. However, since we can sample exactly from this distribution, we can deal with the constant of proportionality in practice by normalizing $\lambda_t^i = \pi_{\lambda,t}^i / \sum_{j=1}^N \pi_{\lambda,t}^j$, where $\pi_{\lambda,t}^i$ are the *unnormalized intermediate weights*.

x_t^i conditional on $(x_{0:t-1}^i, k_i) = x_{0:t-1}^{k_i}$ and then setting $x_{0:t}^i = (x_t^i, x_{0:t-1}^{k_i})$ as the current particle.

As for the target, we proceed analogously to the derivation of (2.5) and decompose $p(x_{0:t}, k|y_{1:t})$ as

$$\begin{aligned} p(x_{0:t}, k|y_{1:t}) &= \frac{p(x_{0:t}, k, y_{1:t})}{p(y_{1:t})} \\ &= \frac{p(y_t|x_t, x_{0:t-1}, k, y_{1:t-1})p(x_t|x_{0:t-1}, k, y_{1:t-1})p(x_{0:t-1}, k|y_{1:t-1})p(y_{1:t-1})}{p(y_t|y_{1:t-1})p(y_{1:t-1})} \\ &= \frac{p(x_{0:t-1}, k|y_{1:t-1})p(x_t|x_{0:t-1}, k, y_{1:t-1})p(y_t|x_t, x_{0:t-1}, k, y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ &\propto p(x_{0:t-1}, k|y_{1:t-1})p(x_t|x_{0:t-1}, k, y_{1:t-1})p(y_t|x_t, x_{0:t-1}, k, y_{1:t-1}) \end{aligned}$$

and, since $(x_{0:t-1}, k) = x_{0:t-1}^k$, by (1.2) $p(y_t|x_t, x_{0:t-1}, k, y_{1:t-1}) = g(y_t|x_t)$ and by item (ii) of Proposition 1.1.1 $p(x_t|x_{0:t-1}, k, y_{1:t-1}) = f(x_t|x_{t-1}^k)$, we get

$$p(x_{0:t}, k|y_{1:t}) \propto p(x_{0:t-1}^k|y_{1:t-1}) f(x_t|x_{t-1}^k) g(y_t|x_t). \tag{2.17}$$

Finally, by properly redefining our importance weights as $\pi_t := p(x_{0:t}, k, y_{1:t}) = p(x_{0:t}, k|y_{1:t})/q(x_{0:t}, k|y_{1:t})$ and by applying (2.16) and (2.17), the APF weight recursion is

$$\begin{aligned} \pi_t &:= \frac{p(x_{0:t}, k|y_{1:t})}{q(x_{0:t}, k|y_{1:t})} \\ &\propto \frac{p(x_{0:t-1}^k|y_{1:t-1})}{q(x_{0:t-1}^k|y_{1:t-1})} \frac{f(x_t|x_{t-1}^k)g(y_t|x_t)}{q(x_t|x_{0:t-1}^k, y_{1:t})\lambda_t^k} \\ &= \frac{\pi_{t-1}^k}{\lambda_t^k} \frac{f(x_t|x_{t-1}^k)g(y_t|x_t)}{q(x_t|x_{0:t-1}^k, y_{1:t})} \qquad (2.18) \\ &\propto \frac{w_{t-1}^k}{\lambda_t^k} \frac{f(x_t|x_{t-1}^k)g(y_t|x_t)}{q(x_t|x_{0:t-1}^k, y_{1:t})}, \qquad (2.19) \end{aligned}$$

where once again we note that $w_{t-1}^{k_i} \propto \pi_{t-1}^{k_i}$ up to $1/\sum_{j=1}^N \pi_{t-1}^j$.

As mentioned before, it turns out that the APF framework is quite general, in that it includes most commonly found particle filters in the literature as special cases, including SIR. Although at first it might not seem clear that SIR fits within the APF framework (due to the different resampling order), by taking $\lambda_t^i = w_{t-1}^i$ the implementation is equivalent, since the set $(x_{0:t-1}^i)$ resampled at the end of a SIR step at $t-1$ is the same as the one resampled at the start of an APF step at t. Starting with $x_0^i \sim \nu(x_0)$ and $\pi_0^i \propto 1 \implies w_0^i = 1/N$ for $i = 1, \ldots, N$, the APF is summarized in Algorithm 2.3.

Now, there are two basic design choices that one must make within the APF framework: the choice of intermediate weights λ_t^k and of the proposal density $q(x_t|x_{0:t-1}^k, y_{1:t})$. Usually, these choices are made so as to keep the importance weights π_t as constant as possible, since in light of Proposition A.2.1 this ensures that the variance of π_t (conditional on both $X_{0:t-1}$ and $Y_{1:t}$) is minimal[6]. The optimal choice in this sense is to let

[6]Note that although π_t is proportional to a constant (as a function of $X_{0:t}$ and k), its *unconditional* variance $\mathrm{var}_{\mathbb{Q}}(\pi_t)$ is in general not zero, since it still involves the variance of the proportionality constant $p(y_t|y_{1:t-1})$.

Algorithm 2.3: Auxiliary Particle Filter

Initialization
for $i = 1$ **to** N **do**
  draw $x_0^i \sim \nu(x_0)$
  set $\pi_0^i \propto 1$
end
for $i = 1$ **to** N **do**
  set $w_0^i = 1/N$
end

Main recursion
for $t = 1$ **to** n **do**
  for $i = 1$ **to** N **do**
    sample k_i from $\{1, \ldots, N\}$ with probability λ_t^i
    draw $x_t^i \sim q(x_t | x_{0:t-1}^{k_i}, y_{1:t})$
    compute $\pi_t^i \propto \frac{w_{t-1}^{k_i}}{\lambda_t^{k_i}} \frac{f(x_t^i | x_{t-1}^{k_i}) g(y_t | x_t^i)}{q(x_t^i | x_{0:t-1}^{k_i}, y_{1:t})}$
  end
  for $i = 1$ **to** N **do**
    compute $w_t^i = \pi_t^i / \sum_{j=1}^N \pi_t^j$
  end
end

$\lambda_t^k \propto w_{t-1}^k p(y_t | x_{t-1}^k)$ and $q(x_t | x_{0:t-1}^k, y_{1:t}) = p(x_t | x_{t-1}^k, y_t)$, since in this case the weight recursion (2.19) becomes

$$\pi_t \propto \frac{w_{t-1}^k}{w_{t-1}^k p(y_t | x_{t-1}^k)} \frac{f(x_t | x_{t-1}^k) g(y_t | x_t)}{p(x_t | x_{t-1}^k, y_t)} = \frac{1}{p(y_t | x_{t-1}^k)} \frac{f(x_t | x_{t-1}^k) g(y_t | x_t)}{\frac{f(x_t | x_{t-1}^k) g(y_t | x_t)}{p(y_t | x_{t-1}^k)}} = 1. \tag{2.20}$$

Note that in the above derivation of (2.20) we have used the fact that $p(x_t | x_{t-1}^k, y_t) = f(x_t | x_{t-1}^k) g(y_t | x_t) / p(y_t | x_{t-1}^k)$, established earlier in the proof of Proposition 2.1.1. In the terminology of Pitt and Shephard (1999), a filter such that (2.20) holds is said to be *fully-adapted* (FA), and this specific incarnation is referred to as the *Fully-Adapted Auxiliary Particle Filter* (FAPF).

In practice, a fully-adapted procedure produces exact draws from the target distribution p (although we have to remember that we need infinitely many draws in order to arbitrarily approximate p with these samples, due to $(x_{0:t}^i, w_t^i)_{i=1}^N$ consisting of a discrete support). Note that, although still adopting the proposal $q(x_t | x_{0:t-1}^k, y_{1:t}) = p(x_t | x_{t-1}^k, y_t)$, Optimal SIR is not fully-adapted due to the choice of intermediate weights $\lambda_t^i = w_{t-1}^i$. Therefore, although abiding to Proposition 2.1.1 (ensuring minimal conditional variance of the importance weights), it does not satisfy the more general proportional relationship $q \propto p$ required by Proposition A.2.1. Also note that in general a SIR procedure can be adapted (by making the proposal q an explicit function of y_t), but the intermediate weights are not a function of y_t, possibly leading to a less efficient procedure. The converse of adapted procedures (such as the bootstrap filter, in which the proposal $f(x_t | x_{t-1})$ is not a function of y_t) are called *blind* procedures.

Now, the main problem associated with full adaptation is that it requires the ability to simulate from $p(x_t|x_{t-1}, y_t)$ and evaluate $p(y_t|x_{t-1})$ pointwise, both of which are usually unfeasible in practice. In this case Pitt and Shephard (1999) proposed approximating these densities by taking $\lambda_t^k \propto w_{t-1}^k g(y_t|\mu_t^k)$ and $q(x_t|x_{0:t-1}^k, y_{1:t}) = f(x_t|x_{t-1}^k)$, where $\mu_t := \mu(X_{0:t-1})$ is any prediction of $X_{0:t-1}$ (such as the one-step-ahead conditional expectation, median or mode of $X_t|X_{0:t-1}$). This so-called "lookahead" strategy is in principle readily appliable to any HMM, and if μ_t is close to X_t, the resulting intermediate weights will be close to the optimal ones. The weight recursion (2.19) for the lookahead strategy is

$$w_t \propto \frac{w_{t-1}^k}{w_{t-1}^k g(y_t|\mu_t^k)} \frac{f(x_t|x_{t-1}^k) g(y_t|x_t)}{f(x_t|x_{t-1}^k)} = \frac{g(y_t|x_t)}{g(y_t|\mu_t^k)}. \tag{2.21}$$

From (2.21), we can see that the closer $g(y_t|\mu_t)$ is to $g(y_t|x_t)$, the closer w_t is to being constant, which implies that the lookahead strategy is also most successful whenever the observations are very informative. Note here that although μ_t is denoted at time t, it is actually a function of $x_{0:t-1}$, and therefore is required to satisfy $(\mu_t, k) = \mu_t^k$.

2.2 Prediction

We now turn our attention to the problem of prediction, i.e. computing $p(x_t|y_{1:t-1})$ or, more generally, $p(x_{t+h}|y_{1:t})$ for positive integer h. Owing to their sequential structure, prediction is a very natural procedure within HMMs, and it can be done in a variety of different ways (see e.g. Doucet et al., 2000). Here, however, we shall limit ourselves to presenting only the simplest method for accomplishing this task, since it is popular and usually yields good results in practice.

First, consider the case of $h = 1$, corresponding to a *one-step-ahead prediction*. We have

$$\begin{aligned} p(x_{t+1}|y_{1:t}) &= \int_{\mathcal{X}^{t+1}} p(x_{t+1}, x_{0:t}|y_{1:t}) dx_{0:t} \\ &= \int_{\mathcal{X}^{t+1}} p(x_{t+1}|x_{0:t}, y_{1:t}) p(x_{0:t}|y_{1:t}) dx_{0:t} \\ &= \int_{\mathcal{X}^{t+1}} f(x_{t+1}|x_t) p(x_{0:t}|y_{1:t}) dx_{0:t}, \end{aligned} \tag{2.22}$$

where $p(x_{t+1}|x_{0:t}, y_{1:t}) = f(x_{t+1}|x_t)$ follows from item (ii) of Proposition 1.1.1 with $k = 0$. Since in general an analytical expression for (2.22) is not available, we can approximate it by replacing $p(x_{0:t}|y_{1:t})$ in (2.22) with its particle estimate $\hat{p}(x_{0:t}|y_{1:t})$ given in (2.2), yielding

$$\begin{aligned} \check{p}(x_{t+1}|y_{1:t}) &:= \int_{\mathcal{X}^{t+1}} f(x_{t+1}|x_t) \hat{p}(x_{0:t}|y_{1:t}) dx_{0:t} \\ &= \int_{\mathcal{X}^{t+1}} f(x_{t+1}|x_t) \sum_{i=1}^N w_t^i \delta_{x_{0:t}^i}(dx_{0:t}) dx_{0:t} \\ &= \sum_{i=1}^N w_t^i f(x_{t+1}|x_t^i). \end{aligned} \tag{2.23}$$

Now, $\check{p}(x_{t+1}|y_{1:t})$ is typically an efficient estimator of $p(x_{t+1}|y_{1:t})$, since it is derived using *Rao-Blackwellization* (see e.g. Doucet et al. 2000 and Section A.3). However, it is

usually not very useful in computing other features of $X_{t+1}|Y_{1:t}$, such as its moments or more general functionals, since the involved integrals might not always have analytical solutions.

An alternative approximation of $p(x_{t+1}|y_{1:t})$ can be derived directly from (2.23) by drawing, for each i, $x_{t+1}^i \sim f(x_{t+1}|x_t^i)$ and replacing $f(x_{t+1}|x_t^i)$ with its particle estimate $\hat{f}(x_{t+1}|x_t^i) := \delta_{x_{t+1}^i}(dx_{t+1})$ (since this is a perfect draw – see Section A.1 – the importance weights of this procedure are uniform and equal to 1, given that for each i we are only sampling a single x_{t+1}^i). The corresponding estimator $\hat{p}(x_{t+1}|y_{1:t})$ is then given by

$$\hat{p}(x_{t+1}|y_{1:t}) := \sum_{i=1}^{N} w_t^i \delta_{x_{t+1}^i}(dx_{t+1}). \tag{2.24}$$

Note that in analogy to the filtering problem in Section 2.1, the one-step-ahead prediction density (2.24) can be regarded as a marginal of the joint approximation $\hat{p}(x_{0:t+1}|y_{1:t})$ obtained by integrating it over the image set of $X_{0:t}$, where

$$\hat{p}(x_{0:t+1}|y_{1:t}) := \sum_{i=1}^{N} w_t^i \delta_{x_{0:t+1}^i}(dx_{0:t+1}). \tag{2.25}$$

For the general *h-step-ahead prediction* case, let

$$\begin{aligned}
p(x_{t+h}|y_{1:t}) &= \int_{\mathcal{X}^{t+h}} p(x_{t+h}, x_{0:t+h-1}|y_{1:t}) dx_{0:t+h-1} \\
&= \int_{\mathcal{X}^{t+h}} p(x_{t+h}|x_{0:t+h-1}, y_{1:t}) \cdot \\
&\qquad p(x_{t+h-1}|x_{0:t+h-2}, y_{1:t}) \cdots p(x_{t+1}|x_{0:t}, y_{1:t}) p(x_{0:t}|y_{1:t}) dx_{0:t+h-1} \\
&= \int_{\mathcal{X}^{t+h}} \left[\prod_{k=1}^{h} p(x_{t+k}|x_{0:t+k-1}, y_{1:t}) \right] p(x_{0:t}|y_{1:t}) dx_{0:t+h-1}.
\end{aligned}$$

Again from item (ii) of Proposition 1.1.1, we have that each term $p(x_{t+k}|x_{0:t+k-1}, y_{1:t})$ of the product inside the integral equals $f(x_{t+k}|x_{t+k-1})$, allowing us to further write the above expression as

$$p(x_{t+h}|y_{1:t}) = \int_{\mathcal{X}^{t+h}} \left[\prod_{k=1}^{h} f(x_{t+k}|x_{t+k-1}) \right] p(x_{0:t}|y_{1:t}) dx_{0:t+h-1}. \tag{2.26}$$

Like (2.22), $p(x_{t+h}|y_{1:t})$ can also be approximated by replacing $p(x_{0:t}|y_{1:t})$ with $\hat{p}(x_{0:t}|y_{1:t})$ in (2.26), i.e.

$$\begin{aligned}
\hat{p}(x_{t+h}|y_{1:t}) &:= \int_{\mathcal{X}^{t+h}} \left[\prod_{k=1}^{h} f(x_{t+k}|x_{t+k-1}) \right] \hat{p}(x_{0:t}|y_{1:t}) dx_{0:t+h-1} \\
&= \int_{\mathcal{X}^{t+h}} \left[\prod_{k=1}^{h} f(x_{t+k}|x_{t+k-1}) \right] \sum_{i=1}^{N} w_t^i \delta_{x_{0:t}^i}(dx_{0:t}) dx_{0:t+h-1} \\
&= \sum_{i=1}^{N} w_t^i \int_{\mathcal{X}^{h-1}} \left[\prod_{k=2}^{h} f(x_{t+k}|x_{t+k-1}) \right] f(x_{t+1}|x_t^i) dx_{t+1:t+h-1}.
\end{aligned} \tag{2.27}$$

Unlike in the one-step-ahead prediction case (2.23), no readily available estimator exists for $p(x_{t+h}|y_{1:t})$, since the integral in (2.27) might not have an analytical solution. Proceeding in the same way as before, however, we can sample $x_{t+1}^i \sim f(x_{t+1}|x_t^i)$, $x_{t+2}^i \sim f(x_{t+1}^i|x_t^i)$, ..., $x_{t+h}^i \sim f(x_{t+h}^i|x_{t+h-1}^i)$ for each i and, since these are all perfect samples, replace each $f(x_{t+k}|x_{t+k-1})$ with its IS counterpart $\hat{f}(x_{t+k}|x_{t+k-1}^i) := \delta_{x_{t+k}^i}(dx_{t+k})$, $k = 1, \ldots, h$. This further approximation yields

$$\begin{aligned}\hat{p}(x_{t+h}|y_{1:t}) &= \sum_{i=1}^N w_t^i \int_{\mathcal{X}^{h-1}} \left[\prod_{k=2}^h \hat{f}(x_{t+k}|x_{t+k-1}^i)\right] \hat{f}(x_{t+1}|x_t^i) dx_{t+1:t+h-1} \\ &= \sum_{i=1}^N w_t^i \int_{\mathcal{X}^{h-1}} \delta_{x_{t+1:t+h}^i}(dx_{t+1:t+h}) dx_{t+1:t+h-1} \\ &= \sum_{i=1}^N w_t^i \delta_{x_{t+h}^i}(dx_{t+h}). \end{aligned} \tag{2.28}$$

Note that if we do not replace $f(x_{t+h}|x_{t+h-1})$ with $\hat{f}(x_{t+h}|x_{t+h-1}^i)$, we end up with a Rao-Blackwellized estimator $\check{p}(x_{t+h}|y_{1:t}) := \sum_{i=1}^N w_t^i f(x_{t+h}|x_{t+h-1}^i)$ similar to (2.23). Again we note that $\hat{p}(x_{t+h}|y_{1:t})$ in (2.28) is just the marginal of the joint h-step-ahead prediction distribution $\hat{p}(x_{0:t+h}|y_{1:t})$ defined by

$$\hat{p}(x_{0:t+h}|y_{1:t}) := \sum_{i=1}^N w_t^i \delta_{x_{0:t+h}^i}(dx_{0:t+h}). \tag{2.29}$$

The entire prediction procedure described here is summarized in Algorithm 2.4 for a general positive integer h, from which the output is a weighted sample $(x_{0:t+h}^i, w_{0:t+h}^i)_{i=1}^N$. Note that entire process basically consists of keeping the importance weights constant at time t (since at each step we simply set $w_{0:t+h}^i = w_t^i$) and sequentially drawing new values from the Markovian transition density f.

Algorithm 2.4: h-step-ahead Prediction

```
for k = 1 to h do
    for i = 1 to N do
        draw x^i_{t+k} ~ f(x_{t+k}|x^i_{t+k-1})
        set w^i_{t+k} = w^i_t
    end
end
```

In closing, it is worth pointing out that resampling does not take place when performing prediction in an HMM, since there are no new observations coming in the model. Also, note that although we have derived the prediction density estimator $\hat{p}(x_{0:t+h}|y_{1:t})$ from an MC integration point-of-view, we could have done it from an MC sampling perspective instead. That is, by writing

$$w_{t+h} := \frac{p(x_{0:t+h}|y_{1:t})}{q(x_{0:t+h}|y_{1:t})} = \left[\prod_{k=1}^h \frac{f(x_{t+k}|x_{t+k-1})}{q(x_{t+k}|x_{0:t+k-1}, y_{1:t})}\right] \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})}$$

and taking $q(x_{t+h}|x_{0:t+h-1}, y_{1:t}) = f(x_{t+k}|x_{t+k-1})$ for each k, we also arrive at

$$w_{t+h} = \left[\prod_{k=1}^{h} \frac{f(x_{t+k}|x_{t+k-1})}{f(x_{t+k}|x_{t+k-1})}\right] \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})} = \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})} = w_t.$$

2.3 Smoothing

Our chapter on state inference closes with smoothing, which consists of computing $p(x_t|y_{1:n})$ for each $t = 0, \ldots, n$. More specifically, here we deal with *forward-backward smoothing* (also known as *fixed-interval smoothing*) which, as its name suggests, is a type of smoothing procedure that involves a forward filtering step and then a backward smoothing one, which is why this type of method is also referred to as *forward-filtering, backward-sampling* (FFBS) procedure (see also Section B.3). Forward-backward smoothing is a topic that has received considerable attention in the literature of HMMs, dating from classical pieces such as Kitagawa (1987), and although it is a very interesting and important subject in its own right, here we mainly restrict ourselves to the presentations of Doucet et al. (2000) and Godsill et al. (2004). Smoothing in general HMMs is intrinsycally connected to Kalman smoothing in linear and Gaussian HMMs (Section B.4).

At first sight, since from the output of a particle filter at $t = n$ we have a weighted sample $(x_{0:n}^i, w_n^i)_{i=1}^N$ approximately distributed according to $X_{0:n}|Y_{1:n}$, approximating $p(x_t|y_{1:n})$ should be as simple as integrating the corresponding $\hat{p}(x_{0:n}|y_{1:n})$ given in (2.2) over the image set of $(X_{0:t-1}, X_{t+1:n})$, i.e.

$$\begin{aligned}
\bar{p}(x_t|y_{1:n}) &:= \int_{\mathcal{X}^n} \hat{p}(x_{0:n}|y_{1:n}) dx_{0:t-1} dx_{t+1:n} \\
&= \int_{\mathcal{X}^n} \sum_{i=1}^{N} w_n^i \delta_{x_{0:n}^i}(dx_{0:n}) dx_{0:t-1} dx_{t+1:n} \\
&= \sum_{i=1}^{N} w_n^i \delta_{x_t^i}(dx_t). \qquad (2.30)
\end{aligned}$$

The problem with (2.30), however, is that although current particles x_n^i are recently rejuvenated, its paths $x_{0:t}^i$ have been successively resampled over time, and as the distance $|n - t|$ grows large, the estimate $\bar{p}(x_t|y_{1:n})$ is supported by an ever smaller number of unique particles. The result is similar to degeneracy, leading to an inefficient estimator due to it being supported only by a small set of effective (i.e. with nonnegligible weights) particles. In fact, this phenomenom is appropriately called *path degeneracy* (and *sample impoverishment* in the more general resampling framework from the literature on evolutionary optimization); see Section 3.2.2 for more details.

Since the problem with (2.30) in practice lies on the fact that the weights $(w_n^i)_{i=1}^N$ are not representative of the particle set $(x_{0:t}^i)_{i=1}^N$ of interest, we might then wish to change these weights accordingly in order to produce a better approximation to $p(x_t|y_{1:n})$. This is precisely the approach taken by Doucet et al. (2000) and Godsill et al. (2004), building on the seminal work by Kitagawa (1987).

We start by deriving a backward recursion for $p(x_t|y_{1:n})$. First, let

$$p(x_t, x_{t+1}|y_{1:n}) = p(x_t|x_{t+1}, y_{1:n}) p(x_{t+1}|y_{1:n}).$$

By decomposing the first term on the right side, we get

$$\begin{aligned}
p(x_t|x_{t+1}, y_{1:n}) &= \frac{p(x_t, x_{t+1}, y_{1:n})}{p(x_{t+1}, y_{1:n})} \\
&= \frac{p(y_{t+1}|x_t, x_{t+1}, y_{1:t}, y_{t+2:n})p(y_{t+2:n}|x_t, x_{t+1}, y_{1:t})p(x_t|x_{t+1}, y_{1:t})p(x_{t+1}, y_{1:t})}{p(y_{t+1}|x_{t+1}, y_{1:t}, y_{t+2:n})p(y_{t+2:n}|x_{t+1}, y_{1:t})p(x_{t+1}, y_{1:t})} \\
&= \frac{g(y_{t+1}|x_{t+1})p(y_{t+2:n}|x_t, x_{t+1}, y_{1:t})p(x_t|x_{t+1}, y_{1:t})}{g(y_{t+1}|x_{t+1})p(y_{t+2:n}|x_{t+1}, y_{1:t})} \\
&= p(x_t|x_{t+1}, y_{1:t})\frac{p(y_{t+2:n}|x_t, x_{t+1}, y_{1:t})}{p(y_{t+2:n}|x_{t+1}, y_{1:t})},
\end{aligned}$$

since by (1.2) Y_{t+1} depends only on X_{t+1}. We can simplify this further to

$$p(x_t|x_{t+1}, y_{1:n}) = p(x_t|x_{t+1}, y_{1:t}) \tag{2.31}$$

by showing that $p(y_{t+2:n}|x_t, x_{t+1}, y_{1:t}) = p(y_{t+2:n}|x_{t+1}, y_{1:t})$, which follows from

$$\begin{aligned}
p(y_{t+2:n}|x_{l:t+1}, y_{1:t}) &= \int_{X^{n-t-1}} p(y_{t+2:n}, x_{t+2:n}|x_{l:t+1}, y_{1:t})dx_{t+2:n} \\
&= \int_{X^{n-t-1}} p(y_{t+2:n}|x_{t+2:n}, x_{l:t+1}, y_{1:t})\cdot \\
&\qquad \cdot p(x_n|x_{t+2:n-1}, x_{l:t+1}, y_{1:t}) \cdots p(x_{t+2}|x_{l:t+1}, y_{1:t})dx_{t+2:n} \\
&= \int_{X^{n-t-1}} \left[\prod_{k=t+2}^{n} g(y_k|x_k)\right] \left[\prod_{j=t+2}^{n} f(x_j|x_{j-1})\right] dx_{t+2:n}
\end{aligned}$$

for integer $0 \leq l \leq t+1$ by applying (1.2) and item (iii) of Proposition 1.1.1 to each term of the first product and item (ii) of Proposition 1.1.1 to each term of the second. With (2.31) we can then write

$$\begin{aligned}
p(x_t, x_{t+1}|y_{1:n}) &= p(x_{t+1}|y_{1:n})p(x_t|x_{t+1}, y_{1:n}) \\
&= p(x_{t+1}|y_{1:n})p(x_t|x_{t+1}, y_{1:t}) \\
&= p(x_{t+1}|y_{1:n})\frac{p(x_t, x_{t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})} \\
&= p(x_{t+1}|y_{1:n})\frac{p(x_{t+1}|x_t, y_{1:t})p(x_t|y_{1:t})}{p(x_{t+1}|y_{1:t})} \\
&= p(x_{t+1}|y_{1:n})\frac{f(x_{t+1}|x_t)p(x_t|y_{1:t})}{p(x_{t+1}|y_{1:t})}
\end{aligned} \tag{2.32}$$

by again applying item (ii) of Proposition 1.1.1 to get $p(x_{t+1}|x_t, y_{1:t}) = f(x_{t+1}|x_t)$. Finally, by integrating $p(x_t|y_{1:n})$ with respect to X_{t+1} and using (2.32), we get

$$\begin{aligned}
p(x_t|y_{1:n}) &= \int_{\mathcal{X}} p(x_t, x_{t+1}|y_{1:n})dx_{t+1} \\
&= \int_{\mathcal{X}} p(x_{t+1}|y_{1:n})\frac{f(x_{t+1}|x_t)p(x_t|y_{1:t})}{p(x_{t+1}|y_{1:t})} \\
&= p(x_t|y_{1:t})\int_{\mathcal{X}} \frac{p(x_{t+1}|y_{1:n})f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})}dx_{t+1},
\end{aligned} \tag{2.33}$$

which is the desired backward recursion.

Now, assume that the approximation to $p(x_t|y_{1:n})$ is of the form

$$\hat{p}(x_t|y_{1:n}) := \sum_{i=1}^{N} w_{t|n}^i \delta_{x_t^i}(dx_t), \tag{2.34}$$

where $w_{t|n}^i$ are the corresponding normalized importance weights. By replacing $p(x_t|y_{1:t})$ with the particle estimate $\hat{p}(x_t|y_{1:t})$ given in (2.3) and $p(x_{t+1}|y_t)$ with the one-step-ahead predictive density Rao-Blackwellized estimator (2.23) in the backward smoothing recursion (2.33), we then have

$$\begin{aligned}
\hat{p}(x_t|y_{1:n}) &= \hat{p}(x_t|y_{1:t}) \int_{\mathcal{X}} \frac{\hat{p}(x_{t+1}|y_{1:n}) f(x_{t+1}|x_t)}{\check{p}(x_{t+1}|y_{1:t})} dx_{t+1} \\
&= \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(dx_t) \int_{\mathcal{X}} \frac{\hat{p}(x_{t+1}|y_{1:n}) f(x_{t+1}|x_t)}{\sum_{j=1}^{N} w_t^j f(x_{t+1}|x_t^j)} dx_{t+1}.
\end{aligned}$$

But (2.34) also implies that $\hat{p}(x_{t+1}|y_{1:n}) = \sum_{i=1}^{N} w_{t+1|n}^i \delta_{x_{t+1}^i}(dx_{t+1})$, which allows us to further write

$$\begin{aligned}
\hat{p}(x_t|y_{1:n}) &= \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(dx_t) \int_{\mathcal{X}} \frac{\hat{p}(x_{t+1}|y_{1:n}) f(x_{t+1}|x_t)}{\sum_{j=1}^{N} w_t^j f(x_{t+1}|x_t^j)} dx_{t+1} \\
&= \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(dx_t) \int_{\mathcal{X}} \frac{\sum_{l=1}^{N} w_{t+1|n}^l \delta_{x_{t+1}^l}(dx_{t+1}) f(x_{t+1}|x_t)}{\sum_{j=1}^{N} w_t^j f(x_{t+1}|x_t^j)} dx_{t+1} \\
&= \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(dx_t) \left\{ \frac{\sum_{l=1}^{N} w_{t+1|n}^l f(x_{t+1}^l|x_t)}{\sum_{j=1}^{N} w_t^j f(x_{t+1}^l|x_t^j)} \right\} \\
&= \sum_{i=1}^{N} w_t^i \sum_{l=1}^{N} \frac{w_{t+1|n}^l f(x_{t+1}^l|x_t)}{\sum_{j=1}^{N} w_t^j f(x_{t+1}^l|x_t^j)} \delta_{x_t^i}(dx_t).
\end{aligned}$$

Finally, by comparing the above expression with (2.34), we obtain a backward recursion for computing $w_{t|n}^i$ as a function of $w_{t+1|n}^i$, i.e.

$$w_{t|n}^i := w_t^i \sum_{l=1}^{N} \frac{w_{t+1|n}^l f(x_{t+1}^l|x_t)}{\sum_{j=1}^{N} w_t^j f(x_{t+1}^l|x_t^j)}, \tag{2.35}$$

with $w_{n|n}^i \equiv w_n^i$.

Starting with $w_{n|n}^i = w_n^i$ for each i, the forward-backward smoothing procedure described above is summarized in Algorithm 2.5. Note that since no simulation takes place here, the process is even simpler than that of prediction, and essentially amounts to refining the already existing importance weights' estimates. Similar to the filtering and prediction density approximations, $\hat{p}(x_t|y_{1:n})$ can also be seen as the marginal of a joint smoothing density $\hat{p}(x_{0:t}|y_{1:n})$, defined by

$$\hat{p}(x_{0:t}|y_{1:n}) := \sum_{i=1}^{N} w_{t|n}^i \delta_{x_{0:t}^i}(dx_{0:t}). \tag{2.36}$$

Algorithm 2.5: Forward-Backward Smoothing

Initialization
for $i = 1$ **to** N **do**
  set $w^i_{n|n} = w^i_n$
end

Backward recursion
for $t = n\text{-}1$ **to** 0 **do**
  for $i = 1$ **to** N **do**
    set $w^i_{t|n} = w^i_t \sum_{l=1}^{N} \frac{w^l_{t+1|n} f(x^l_{t+1}|x_t)}{\sum_{j=1}^{N} w^j_t f(x^l_{t+1}|x^j_t)}$
  end
end

Given that smoothing requires that we observe the entire sample $y_{1:n}$ before actually performing the required inference, it is generally referred to as an *offline* or *batch* procedure, as opposed to *online* procedures which can be performed as new observations arrive, such as filtering and prediction. Also, although sometimes this can be avoided (Elliott et al., 2008; Douc et al., 2011), typical FFBS procedures such as the one presented here have an $\mathcal{O}(nN^2)$ complexity, making their cost sometimes prohibitive when compared to the usual $\mathcal{O}(nN)$ operations required for filtering.

Chapter 3

Parameter Inference

In this chapter we turn our attention to the general situation in which θ is unknown and has to be estimated from the data (the act of inferring about static parameters is usually – specially within the Bayesian inference paradigm – referred to as *parameter learning*). Although our main concern in this work is in performing sequential inference for θ, i.e. in an online fashion that consists of obtaining estimates as new observations are incorporated into the model, we start with a brief presentation of the state-of-the-art on the literature of non-sequential inference methods, namely the *particle Markov Chain Monte Carlo* algorithm of Andrieu et al. (2010). This serves not only to introduce some ideas that will be necessary to our main contribution, but also as a contrast to (and the main bechmark for which to test against) our methods.

Note that throughout this chapter and the rest of this work we will limit ourselves to the Bayesian inference paradigm. This choice was essentialy made in order to give focus to our main contribution, but is by no means exhaustive. There is a vast amount of work on parameter inference for HMMs based on maximum likelihood and Expectation-Maximization-based techniques and within the classical inference paradigm more generally; see e.g. Kantas et al. (2015) and Schön et al. (2011) for two major reviews.

3.1 Particle Markov Chain Monte Carlo

The Particle Markov Chain Monte Carlo (pMCMC) algorithm of Andrieu et al. (2010) is a landmark method in parameter learning for HMMs, since it still targets the correct joint posterior density $p(x_{0:n}, \theta | y_{1:n})$ even when sampling $x_{0:n}$ from a SMC proposal and approximating the model likelihood with an unbiased particle estimate, allowing for impressive efficiency gains in practice as compared to usual MCMC methods.

First, assume that we can sample a static parameter θ' from the invariant Markov kernel (see Section A.4) $q(\theta'|\theta)$ and a state sequence $x'_{0:n}$ *exactly* (rather than only approximately, via SMC) from its target posterior $p(x_{0:n}|y_{1:n}, \theta')$ given the current pair $(x_{0:n}, \theta)$. This implies that the joint proposal for $(x'_{0:n}, \theta')$ given $(x_{0:n}, \theta)$ and $Y_{1:n} = y_{1:n}$, denoted by $q(x'_{0:n}, \theta'|x_{0:n}, \theta, y_{1:n})$, satisfies

$$q(x'_{0:n}, \theta | x_{0:n}, \theta, y_{1:n}) = q(\theta'|\theta) p(x'_{0:n}|y_{1:n}, \theta'). \tag{3.1}$$

This is equivalent to assuming that under q the law of $X'_{0:n}$ is *perfectly adapted* to the most recent value of θ (which is θ'), that θ' does not depend on $Y_{1:n}$ and that both $X'_{0:n}$ and θ' are independent of the current state path $X_{0:n}$, given that the identity (3.1) only

holds if $q(x'_{0:n}|x_{0:n}, \theta', \theta, y_{1:n}) = p(x'_{0:n}|y_{1:n}, \theta')$ and $q(\theta'|x_{0:n}, \theta, y_{1:n}) = q(\theta'|\theta)$, in light of

$$q(x'_{0:n}, \theta'|x_{0:n}, \theta, y_{1:n}) = q(x'_{0:n}|x_{0:n}, \theta', \theta, y_{1:n}) q(\theta'|x_{0:n}, \theta, y_{1:n}).$$

Since $p(x_{0:n}, \theta|y_{1:n}) = p(x_{0:n}|y_{1:n}, \theta)p(y_{1:n}|\theta)p(\theta)$, we can then write the probability (A.14) of accepting the new pair $(x'_{0:n}, \theta')$ within a Metropolis-Hastings (MH) sampling framework (Section A.4) as

$$\begin{aligned}
\alpha(x'_{0:n}, \theta'|x_{0:n}, \theta, y_{1:n}) &:= 1 \wedge \frac{p(x'_{0:n}, \theta'|y_{1:n})}{p(x_{0:n}, \theta|y_{1:n})} \frac{q(x_{0:n}, \theta|x'_{0:n}, \theta', y_{1:n})}{q(x'_{0:n}, \theta'|x_{0:n}, \theta, y_{1:n})} \\
&= 1 \wedge \frac{p(x'_{0:n}|y_{1:n}, \theta')p(y_{1:n}|\theta')p(\theta')}{p(x_{0:n}|y_{1:n}, \theta)p(y_{1:n}|\theta)p(\theta)} \frac{q(\theta|\theta')p(x_{0:n}|y_{1:n}, \theta)}{q(\theta'|\theta)p(x'_{0:n}|y_{1:n}, \theta')} \\
&= 1 \wedge \frac{p(y_{1:n}|\theta')p(\theta')}{p(y_{1:n}|\theta)p(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)},
\end{aligned} \tag{3.2}$$

where $x \wedge y := \min(x, y)$.

Now, in general we cannot sample exactly from $p(x_{0:n}|y_{1:n}, \theta)$ and neither evaluate the observation likelihood $p(y_{1:n}|\theta)$ necessary for computing (3.2), which is the primary reason for why we rely on SMC methods in the first place. It turns out, however, that within a *pseudo-marginal* MCMC framework (Andrieu et al., 2009) the sampler still leaves $p(x_{0:n}, \theta|y_{1:n})$ invariant even if we draw $x'_{0:n}$ from the particle approximation[1] $\hat{p}(x_{0:n}|y_{1:n}, \theta)$ defined in (2.2) and replace $p(y_{1:n}|\theta)$ with an unbiased estimate $\hat{p}(y_{1:n}|\theta)$. This perhaps surprising property, proven in Andrieu et al. (2010), is what makes pMCMC so powerful and appealing in practice. Under relatively weak conditions, the authors also show that the pMCMC sampler is ergodic.

Since our main objective here is to perform inference for θ, we will consider only the version of pMCMC designed for that purpose, namely the *Particle Marginal Metropolis Hastings* (PMMH) algorithm. In practice, no modification of the sampler presented so far is necessary other than simply ignoring the sampled state paths $x_{0:n}$, since we can deduce from (3.2) that the resulting algorithm still does indeed target the correct posterior $p(\theta|y_{1:n})$, given that $p(\theta|y_{1:n}) \propto p(y_{1:n}|\theta)p(\theta)$.

Now, in order to obtain estimates of $p(y_{1:n}|\theta)$ for a given value of θ we adopt the APF-based likelihood estimator of Pitt et al. (2012), defined by

$$\hat{p}(y_{1:n}|\theta) := \prod_{t=1}^{n} \left[\left\{ \sum_{i=1}^{N} \frac{\pi^i_{w,t}(\theta)}{N} \right\} \left\{ \sum_{i=1}^{N} \pi^i_{\lambda,t}(\theta) \right\} \right] \tag{3.3}$$

where $\pi^i_{w,t}(\theta)$ and $\pi^i_{\lambda,t}(\theta)$ are the *unnormalized importance* and *unnormalized intermediate* weights respectively (see Section D), i.e. satisfying $w^i_t = \pi^i_{w,t}(\theta) / \sum_{j=1}^{N} \pi^j_{w,t}(\theta)$ and $\lambda^i_t = \pi^i_{\lambda,t}(\theta) / \sum_{j=1}^{N} \pi^j_{\lambda,t}(\theta)$, taken here as explicit functions of θ. The PMMH procedure presented in this section is summarized in Algorithm 3.1. Note that this is actually a slight generalization introduced by Pitt et al. (2012) of the original PMMH proposed in Andrieu et al. (2010), which is limited to the SIR framework.

The converse of PMMH, i.e. in which the target is $p(x_{0:n}|y_{1:n}, \theta)$ for a fixed value of θ throughout the entire process, is called *Particle Independent Metropolis Hastings* (PIMH), in allusion to the fact that the newly sampled $x'_{0:n}$ is always independent of

[1] Note that since $\hat{p}(x_{0:n}|y_{1:n}, \theta) := \sum_{i=1}^{N} w^i_n \delta_{x^i_{0:n}}(dx_{0:n})$, in practice this is equivalent to resampling a single particle in SIR, i.e. draw $x^i_{0:n}$ with probability w^i_n.

Algorithm 3.1: Particle Marginal Metropolis Hastings

Initialization

draw $\theta^0 \sim p(\theta)$

run Algorithm 2.3 conditional on θ^0 and store $(\pi^i_{w,1:n}(\theta^0), \pi^i_{\lambda,1:n}(\theta^0))_{i=1}^N$

compute $\hat{p}(y_{1:n}|\theta^0) = \prod_{t=1}^n \left[\left\{\sum_{i=1}^N N^{-1}\pi^i_{w,t}(\theta^0)\right\}\left\{\sum_{i=1}^N \pi^i_{\lambda,t}(\theta^0)\right\}\right]$

set $\theta \leftarrow \theta^0$

set $\hat{p}(y_{1:n}|\theta) \leftarrow \hat{p}(y_{1:n}|\theta^0)$

Main recursion

for $i = 1$ **to** $B + M$ **do**

draw $\theta' \sim q(\theta'|\theta)$

run Algorithm 2.3 conditional on θ' and store $(\pi^i_{w,1:n}(\theta'), \pi^i_{\lambda,1:n}(\theta'))_{i=1}^N$

compute $\hat{p}(y_{1:n}|\theta') = \prod_{t=1}^n \left[\left\{\sum_{i=1}^N N^{-1}\pi^i_{w,t}(\theta')\right\}\left\{\sum_{i=1}^N \pi^i_{\lambda,t}(\theta')\right\}\right]$

draw $u \sim U[0,1]$

compute $\alpha(\theta'|\theta) = 1 \wedge \frac{\hat{p}(y_{1:n}|\theta')p(\theta')}{\hat{p}(y_{1:n}|\theta)p(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)}$

if $u <= \alpha(\theta'|\theta)$ **then**

set $\theta^i \leftarrow \theta'$

end

else

set $\theta^i \leftarrow \theta$

end

set $\theta \leftarrow \theta^i$

set $\hat{p}(y_{1:n}|\theta) \leftarrow \hat{p}(y_{1:n}|\theta^i)$

end

the previous $x_{0:n}$. PIMH can also be obtained as a direct marginalization of the general pMCMC with target $p(x_{0:n}, \theta|y_{1:n})$, and its acceptance probability is $\alpha(x'_{0:n}|x_{0:n}) = \hat{p}_{x'_{0:n}}(y_{1:n}|\theta)/\hat{p}_{x_{0:n}}(y_{1:n}|\theta)$, where here the additional subscript in $\hat{p}(y_{1:n}|\theta)$ indicates the state path from which the approximation was computed. However, since our interest in this work lies solely in performing parameter inference, we will not consider exploring PIMH further.

Finally, it should be noted that although pMCMC is very appealing in theory, the method needs a fair amount of tuning in order to perform well in practice. Important works in this regard include e.g. Pitt et al. (2012), Doucet et al. (2015), Sherlock et al. (2015).

3.2 Sequential Parameter Learning

Having briefly discussed our main reference for offline parameter learning methods in Section 3.1, we now turn our attention to our main problem, which as stated previously is to learn about $\theta \in \Theta$ *sequentially*, i.e. to compute $p(\theta|y_{1:t})$ for all t. In Sections 3.2.1 and 3.2.3 we introduce a novel class of algorithms for dealing with this problem, and in Section 3.2.4 we show how this framework accomodates many of the commonly

found methods for sequential parameter learning in the literature as special cases. These sections contain the main novel contribution of this thesis to the state-of-the-art of this literature.

3.2.1 A Novel Framework

Let θ_t denote the parameter associated with the posterior $p(\theta|y_{1:t})$ at time t. Although θ is still inherently static, keeping track of the inference for it across time allows us to implicitly define a sequence $(\theta_t)_{t\geq 0}$, where each $\theta_t := \theta|Y_{1:t}$ and with initial distribution given by the prior $\theta_0 \sim p(\theta)$. The joint posterior for $(X_{0:t}, k, \theta_{0:t})$ given[2] $Y_{1:t} = y_{1:t}$ then admits the recursion

$$p(x_{0:t}, k, \theta_{0:t}|y_{1:t}) = p(\theta_t|x_{0:t}, k, \theta_{0:t-1}, y_{1:t})p(y_t|x_{0:t}, k, \theta_{0:t-1}, y_{1:t-1})\cdot \\ \cdot p(x_t|x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t-1})\frac{p(x_{0:t-1}, k, \theta_{0:t-1}|y_{1:t-1})p(y_{1:t-1})}{p(y_t|y_{1:t-1})p(y_{1:t-1})}.$$

In the framework proposed here we implicitly assume that all marginal distributions of $(X_t, Y_t)_{t\geq 0}$ depend only on the most recent value of θ, i.e. that $(X_t, Y_t)_{t\geq 0}$ is *perfectly adapted* to the sequence $(\theta_t)_{t\geq 0}$ (in the sense defined in Section 3.1). We also adopt the same resampling formalism as the APF, implying that $(x_{l:t}, k) := (x_t, x^k_{l:t-1})$ and $(\theta_{l:t-1}, k) := \theta^k_{l:t-1}$ for integer $0 \leq l \leq t-1$. Therefore, by (1.2) and (1.1), we then have, respectively, that $p(y_t|x_{0:t}, k, \theta_{0:t-1}, y_{1:t-1}) = g(y_t|x_t, \theta^k_{t-1})$ and $p(x_t|x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t-1}) = f(x_t|x^k_{t-1}, \theta^k_{t-1})$, allowing us to further write the joint target distribution as

$$\begin{aligned} &p(x_{0:t}, k, \theta_{0:t}|y_{1:t}) = \\ &= p(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})g(y_t|x_t, \theta^k_{t-1})f(x_t|x^k_{t-1}, \theta^k_{t-1})\frac{p(x^k_{0:t-1}, \theta^k_{0:t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ &\propto p(x^k_{0:t-1}, \theta^k_{0:t-1}|y_{1:t-1})f(x_t|x^k_{t-1}, \theta^k_{t-1})g(y_t|x_t, \theta^k_{t-1})p(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t}). \end{aligned} \tag{3.4}$$

Assuming that the proposal for drawing $(X_{0:t}, k, \theta_{0:t})$ satifies

$$\begin{aligned} &q(x_{0:t}, k, \theta_{0:t}|y_{1:t}) = \\ &\quad = q(x^k_{0:t-1}, \theta^k_{0:t-1}|y_{1:t-1})q(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})q(x_t|x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})\lambda^k_t, \end{aligned} \tag{3.5}$$

where $\lambda^k_t := q(k|x_{0:t-1}, \theta_{0:t-1}, y_{1:t})$, we then have the weight recursion

$$\begin{aligned} \pi_t &:= \frac{p(x_{0:t}, k, \theta_{0:t}|y_{1:t})}{q(x_{0:t}, k, \theta_{0:t}|y_{1:t})} \\ &\propto \frac{p(x^k_{0:t-1}, \theta^k_{0:t-1}|y_{1:t-1})f(x_t|x^k_{t-1}, \theta^k_{t-1})g(y_t|x_t, \theta^k_{t-1})p(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})}{q(x^k_{0:t-1}, \theta^k_{0:t-1}|y_{1:t-1})q(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})q(x_t|x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})\lambda^k_t} \\ &= \frac{\pi^k_{t-1}}{\lambda^k_t}\frac{f(x_t|x^k_{t-1}, \theta^k_{t-1})g(y_t|x_t, \theta^k_{t-1})}{q(x_t|x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})}\frac{p(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})}{q(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})} \\ &\propto \frac{w^k_{t-1}}{\lambda^k_t}\frac{f(x_t|x^k_{t-1}, \theta^k_{t-1})g(y_t|x_t, \theta^k_{t-1})}{q(x_t|x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})}\frac{p(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})}{q(\theta_t|x_t, x^k_{0:t-1}, \theta^k_{0:t-1}, y_{1:t})}. \end{aligned} \tag{3.6}$$

[2]Once again we assume that only $Y_{1:t}$ is observed, treating Y_0 as arbitrary or as the model prior information, i.e. so that $p(x_0, \theta_0|y_0) = p(x_0|\theta_0, y_0)p(\theta_0|y_0) = \nu(x_0|\theta_0)p(\theta_0)$.

Note that pointwise evaluation of the weight recursion (3.6) requires the ability of not only evaluating the APF weights (2.19) but also the ratio $p(\theta_t|x_t, x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})/q(\theta_t|x_t, x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})$, at least up to a proportionality constant. This usually requires making additional assumptions about the specific (or approximate) form of the density $p(\theta_t|x_t, x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})$ and, as illustrated below in Section 3.2.4, is essentially what distinguishes one sequential parameter learning algorithm from the other.

The fundamental design choices in the framework proposed here are the intermediate weights λ_t^k and the state and static parameter proposals $q(x_t|x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})$ and $q(\theta_t|x_t, x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})$, respectively. Starting with $\theta_0^i \sim p(\theta)$, $x_0^i \sim \nu(x_0|\theta_0^i)$ and $\pi_0^i \propto 1 \implies w_0^i = 1/N$ for $i = 1, \ldots, N$, the algorithm for sequential parameter learning developed here is summarized in Algorithm 3.2. It should be clear that the class proposed here also contains the APF described in Algorithm 2.3 (i.e. without any parameter learning) by simply taking θ_t to be a fixed quantity θ^* for all t, or equivalently by assuming that $p(\theta) = \delta_{\theta^*}(d\theta)$.

Algorithm 3.2: Sequential Parameter Learning

Initialization
for $i = 1$ **to** N **do**
  draw $\theta_0^i \sim p(\theta)$
  draw $x_0^i \sim \nu(x_0|\theta_0^i)$
  set $\pi_0^i \propto 1$
end
for $i = 1$ **to** N **do**
  set $w_0^i = 1/N$
end

Main recursion
for $t = 1$ **to** n **do**
  for $i = 1$ **to** N **do**
    sample k_i from $\{1, \ldots, N\}$ with probability λ_t^i
    draw $x_t^i \sim q(x_t|x_{0:t-1}^{k_i}, \theta_{0:t-1}^{k_i}, y_{1:t})$
    draw $\theta_t^i \sim q(\theta_t|x_t, x_{0:t-1}^{k_i}, \theta_{0:t-1}^{k_i}, y_{1:t})$
    compute $\pi_t^i \propto \frac{w_{t-1}^{k_i}}{\lambda_t^{k_i}} \frac{f(x_t^i|x_{t-1}^{k_i}, \theta_{t-1}^{k_i}) g(y_t|x_t^i, \theta_{t-1}^{k_i})}{q(x_t^i|x_{0:t-1}^{k_i}, \theta_{0:t-1}^{k_i}, y_{1:t})} \frac{p(\theta_t^i|x_t^i, x_{0:t-1}^{k_i}, \theta_{0:t-1}^{k_i}, y_{1:t})}{q(\theta_t^i|x_t^i, x_{0:t-1}^{k_i}, \theta_{0:t-1}^{k_i}, y_{1:t})}$
  end
  for $i = 1$ **to** N **do**
    compute $w_t^i = \pi_t^i / \sum_{j=1}^N \pi_t^j$
  end
end

The main output from Algorithm 3.2 is the approximation

$$\hat{p}(x_{0:t}, \theta_{0:t}|y_{1:t}) := \sum_{i=1}^N w_t^i \delta_{(x_{0:t}^i, \theta_{0:t}^i)}(dx_{0:t} d\theta_{0:t}), \tag{3.7}$$

which is typically referred to as the *histogram-based* estimator of the joint posterior distribution of $(X_{0:t}, \theta_{0:t})$ given $Y_{1:t} = y_{1:t}$. To obtain an approximation to the target

$p(\theta|y_{1:t})$, we can simply integrate (3.7) over the support of $(X_{0:t}, \theta_{0:t-1})$, yielding

$$\hat{p}(\theta|y_{1:t}) := \int_{\mathcal{X}^{t+1}\times\Theta^t} \hat{p}(x_{0:t}, \theta_{0:t}|y_{1:t}) dx_{0:t} d\theta_{0:t-1} = \sum_{i=1}^{N} w_t^i \delta_{\theta_t^i}(d\theta). \tag{3.8}$$

Proceeding analogously, estimators of any marginal of $p(x_{0:t}, \theta_{0:t}|y_{1:t})$ can be obtained by integrating (3.7) accordingly. In particular, integrating over the entire path of the static parameters $\theta_{0:t}$ results in the state posterior (2.2) obtained in the "pure filtering" context of Section 2.1.

Besides the usual histogram-based estimator defined in (3.8), an alternative estimator of $p(\theta|y_{1:t})$ can be obtained via *Rao-Blackwellization* (Liu and Chen, 1998; Doucet et al., 2000). First, note that we can rewrite the target distribution as

$$\begin{aligned} p(\theta|y_{1:t}) &= \int_{\mathcal{X}^{t+1}\times\Theta^t} p(\theta, x_{0:t}, \theta_{0:t-1}|y_{1:t}) dx_{0:t} d\theta_{0:t-1} \\ &= \int_{\mathcal{X}^{t+1}\times\Theta^t} p(\theta|x_{0:t}, \theta_{0:t-1}, y_{1:t}) p(x_{0:t}, \theta_{0:t-1}|y_{1:t}) dx_{0:t} d\theta_{0:t-1} \\ &= \mathbb{E}_{\mathbb{P}}[p(\theta|X_{0:t}, \theta_{0:t-1}, Y_{1:t})|Y_{1:t}], \end{aligned} \tag{3.9}$$

i.e. as the conditional expectation (under $\mathbb{P}$) of $p(\theta|x_{0:t}, \theta_{0:t-1}, y_{1:t})$ given $Y_{1:t}$. Now, we can obtain a direct Monte Carlo approximation to (3.9) by simply replacing the integrating density $p(x_{0:t}, \theta_{0:t-1}|y_{1:t})$ with its particle approximation $\hat{p}(x_{0:t}, \theta_{0:t-1}|y_{1:t})$ in the corresponding integral. This gives

$$\begin{aligned} \check{p}(\theta|y_{1:t}) &:= \int_{\mathcal{X}^{t+1}\times\Theta^t} p(\theta|x_{0:t}, \theta_{0:t-1}, y_{1:t}) \hat{p}(x_{0:t}, \theta_{0:t-1}|y_{1:t}) dx_{0:t} d\theta_{0:t-1} \\ &= \int_{\mathcal{X}^{t+1}\times\Theta^t} p(\theta|x_{0:t}, \theta_{0:t-1}, y_{1:t}) \sum_{i=1}^{N} w_t^i \delta_{(x_{0:t}^i, \theta_{0:t-1}^i)}(dx_{0:t} d\theta_{0:t-1}) dx_{0:t} d\theta_{0:t-1} \\ &= \sum_{i=1}^{N} w_t^i p(\theta|x_{0:t}^i, \theta_{0:t-1}^i, y_{1:t}). \end{aligned} \tag{3.10}$$

The resulting expression for $\check{p}(\theta|y_{1:t})$ given in (3.10) is then known as the *Rao-Blackwellized* estimator of the posterior $p(\theta|y_{1:t})$.

The Rao-Blackwellized estimator (3.10) is typically (Liu and Chen, 1998) more efficient than the histogram-based estimator (3.8) whenever interest lies in approximating only the posterior $p(\theta|y_{1:t})$, i.e. the typical setting for sequential parameter learning. However, this comes at the cost of having to evaluate $p(\theta|x_{0:t}, \theta_{0:t-1}, y_{1:t})$ pointwise. Moreover, if interest lies in the moments and/or general functionals of $\theta|Y_{1:t}$, analytical solutions to the required integrals might be much more involved and sometimes unattainable when compared to the simpler histogram-based estimator (3.7).

A subtle point about the framework proposed here is that in Algorithm 3.2 we first sample the states X_t and only then sample the parameters θ_t. Although this might not appear relevant at first, explicitly adopting this order proves crucial for obtaining fully-adapted procedures, as discussed below. See Section 3.2.3 to see how we can accommodate methods that have the reverse sampling order, such as Liu and West (2001)'s and Storvik (2002)'s filters.

We obtain a fully-adapted sequential parameter learning procedure by taking intermediate weights $\lambda_t^k \propto w_{t-1}^k p(y_t|x_{t-1}^k, \theta_{t-1}^k)$, state proposal $q(x_t|x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t}) =$

$p(x_t|x_{t-1}^k, \theta_{t-1}^k, y_t)$ and static parameter proposal equal to $q(\theta_t|x_t, x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t}) = p(\theta_t|x_t, x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})$, since in this case the associated importance weights (3.6) become

$$\pi_t \propto \frac{w_{t-1}^k}{w_{t-1}^k p(y_t|x_{t-1}^k, \theta_{t-1}^k)} \frac{f(x_t|x_{t-1}^k, \theta_{t-1}^k) g(y_t|x_t, \theta_{t-1}^k)}{p(x_t|x_{t-1}^k, \theta_{t-1}^k, y_t)} \frac{p(\theta_t|x_t, x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})}{p(\theta_t|x_t, x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})} = 1.$$

On the other hand, if the sampling order is reversed (i.e. θ_t before X_t), we would end up with the weight recursion

$$\pi_t \propto \frac{w_{t-1}^k}{\lambda_t^k} \frac{f(x_t|x_{t-1}^k, \theta_t) g(y_t|x_t, \theta_t)}{q(x_t|x_{0:t-1}^k, \theta_t, \theta_{0:t-1}^k, y_{1:t})} \frac{p(\theta_t|x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})}{q(\theta_t|x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})}.$$

Here, even if we choose by analogy the intermediate weights $\lambda_t^k \propto w_{t-1}^k p(y_t|x_{t-1}^k, \theta_{t-1}^k)$ and proposals $q(x_t|x_{0:t-1}^k, \theta_t, \theta_{0:t-1}^k, y_{1:t}) = p(x_t|x_{t-1}^k, \theta_t, y_t)$ and $q(\theta_t|x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t}) = p(\theta_t|x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})$, the resulting weights would be

$$\pi_t \propto \frac{w_{t-1}^k}{w_{t-1}^k p(y_t|x_{t-1}^k, \theta_{t-1}^k)} \frac{f(x_t|x_{t-1}^k, \theta_t) g(y_t|x_t, \theta_t)}{p(x_t|x_{t-1}^k, \theta_t, y_t)} \frac{p(\theta_t|x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})}{p(\theta_t|x_{0:t-1}^k, \theta_{0:t-1}^k, y_{1:t})} = \frac{p(y_t|x_{t-1}^k, \theta_t)}{p(y_t|x_{t-1}^k, \theta_{t-1}^k)},$$

which in general are not proportional to 1. Note that in the above derivation we have used that $p(x_t|x_{t-1}^k, \theta, y_t) = f(x_t|x_{t-1}^k, \theta) g(y_t|x_t, \theta) / p(y_t|x_{t-1}^k, \theta)$ for either $\theta = \theta_{t-1}^k$ or $\theta = \theta_t$. This identity can easily be shown to hold by an argument analogous to the one used in the proof of Proposition 2.1.1.

In summary, the argument for sampling the states X_t before the parameters θ_t lies on the fact that at time t we cannot take the intermediate weights λ_t^k to be a function of the current parameter θ_t, since here resampling is done prior to sampling the parameters[3]. As far as sequential parameter learning is concerned, this also has the benefit that we always perform inference for the parameters based on the most recent information about the states.

3.2.2 Path Degeneracy and Resampling

Due to the unavoidable degeneracy inherent in sequential importance sampling methods, the resampling step is an integral part of SMC. Despite its benefits, however, resampling has an important drawback: *sample impoverishment*, which in this context takes the form of *path degeneracy* (Andrieu et al., 2005).

Path degeneracy manifests itself as the coalescence of particles' paths ocurring from successive resampling steps. As an example, consider a general functional[4] $Z_{l:t-1}$ of $(X_{l:t-1}, \theta_{l:t-1})$ defined for integer $0 \leq l \leq t-1$ and computed recursively along the filter's trajectory. At time t, $(z_{l:t-1}^{k_i})_{i=1}^N$ is the set resampled from $(z_{l:t-1}^i)_{i=1}^N$ and, due to some $z_{l:t-1}^i$'s typically having lower weights than others, the resampled set $(z_{l:t-1}^{k_i})_{i=1}^N$ will

[3] Although theoretically this could be dealt with in a *propagate-resample* framework (i.e. one in which we first sample the states/parameters and then perform the resampling step), this might be undesirable since then the resampled sequences $X_{0:t-1}^k$ and $\theta_{0:t-1}^k$ will not benefit from current information on y_t, i.e. the procedure will be *blind* as per the APF terminology. Works comparing propagate-resample and resample-propagate frameworks from a theoretical standpoint include e.g. Petetin and Desbouvries (2013) and from an empirical standpoint include e.g. Lopes and Tsay (2011).

[4] Examples of such functionals include sufficient statistics $\mathcal{S}_{0:t-1}$ for $(X_{0:t-1}, Y_{1:t-1})$ and even the state and static parameter paths $X_{0:t-1}$ and $\theta_{0:t-1}$ themselves; see the various examples at Section 3.2.4.

have fewer distinct values than $(z^i_{l:t-1})^N_{i=1}$. At time $t+1$, we now have $(z^i_{l:t})^N_{i=1}$, with each $z^i_{l:t} = (z^{k_i}_{l:t-1}, z^i_t)$. Therefore, when resampling takes place, the $z^{k_i}_{l:t-1}$'s are going to be resampled again, taking even fewer distinct values than before. Over time, this is compounded and the paths $z^i_{l:t}$ eventually degenerate (hence the name) to a single point.

Now, path degeneracy is progressively worse as l is closer to 0. Whenever $l = t-1$, i.e. when only Z_{t-1} is of interest, the sample impoverishment due to resampling is minimal since the transition from Z_{t-1} to Z_t essentially "replenishes" the number of distinct values the functional $Z_{l:t-1}$ can take from one step to the other. This is why path degeneracy can for the most part be ignored whenever interest lies only in state filtering, since the state transition from X_{t-1} to X_t will usually allow for a proper exploration of the state space even when the number of distinct values of X_{t-1} is small. Formally, this property is known in the literature as *exponential forgetting* (Del Moral, 2004) and as it name implies it refers to the ability of the functional to "forget" (i.e. eliminate its dependence on) past values exponentially fast, thus avoiding path degeneracy.

Whenever the state transition does not allow for a proper exploration of the state space (i.e. whenever the exponential forgetting property is not satisfied), however, path degeneracy can become problematic even if $l = t-1$. This is specially true for sequential parameter learning, since the static parameters for which we are trying to perform inference for usually have no "natural" dynamic. Here, even if we are only interested in the most recent value θ_t, if the parameters are static we implicitly have $\theta_t = \theta_{t-1}$ for all t and eventually $\theta_t = \theta_0$, meaning that at each time t we only resample from an ever-decreasing set of distinct values drawn from the prior $p(\theta)$.

Since path degeneracy is a direct consequence of sample impoverishment, we should therefore avoid sample impoverishment as much as possible. This can essentially be done in two ways: by choosing optimal proposal distributions and by considering more efficient resampling schemes. The first of these is very straightforward, since it simply consists of opting for fully-adapted procedures whenever possible.

On the other hand, considering more efficient resampling schemes might at first sight not be as straightforward, since there are several methodologies in the literature from which to choose from (see e.g. Randal et al., 2005; Li et al., 2015). It turns out, however, that there is a single resampling scheme that have been proven to have minimal variance amongst all *unbiased* (i.e. such that the expected number of offspring ξ^i_t of particle $z^i_{l:t-1}$ equals $N \cdot \lambda^i_t$) resampling methods. This method, introduced by Crisan and Lyons (2002), is usually known as the *tree-based branching algorithm*, or simply as *branching algorithm*.

Essentially, the branching algorithm relies on near-deterministic allocations in order to sample offspring, which in practice is why the method is so efficient. At time t, the ith particle is assigned a number of offspring ξ^i_t according to

$$\xi^i_t = \begin{cases} \lfloor N\lambda^i_t \rfloor & \text{with probability} \quad 1 - \{N\lambda^i_t\} \\ \lfloor N\lambda^i_t \rfloor + 1 & \text{with probability} \quad \{N\lambda^i_t\} \end{cases} \tag{3.11}$$

where $\lfloor \cdot \rfloor$ is the *floor* operator and $\{x\} := x - \lfloor x \rfloor$ is the non-integer part of x. There are several ways in which we can sample an entire offspring set $(\xi^i_t)^N_{i=1}$ satisfying $\sum^N_{i=1} \xi^i_t = N$ and (3.11). Without further details, we adhere to the procedure described in Bain and Crisan (2009), summarized in Algorithm 3.3.

It is worth pointing out that Algorithm 3.3 does not yield a set of indices $(k_i)^N_{i=1}$ from which we can perform resampling with but rather only the set of offspring $(\xi^i_t)^N_{i=1}$ produced by each particle. There is however a clear bijection relationship between them, in that we can produce one set directly from the other. For completeness, a procedure to

Algorithm 3.3: Tree-Based Branching Resampling

Alias Table
set $g = N$
set $h = N$

Sampling Indices
for *$i = 1$* **to** *N-1* **do**
  draw $u \sim U[0,1]$
  if $\{N\lambda_t^i\} + \{g - N\lambda_t^i\} < 1$ **then**
    if $u < 1 - (\{N\lambda_t^i\}/\{g\})$ **then**
      set $\xi_t^i = \lfloor N\lambda_t^i \rfloor$
    end
    else
      set $\xi_t^i = \lfloor N\lambda_t^i \rfloor + (h - \lfloor g \rfloor)$
    end
  end
  else
    if $u < 1 - (1 - \{N\lambda_t^i\})/(1 - \{g\})$ **then**
      set $\xi_t^i = \lfloor N\lambda_t^i \rfloor + 1$
    end
    else
      set $\xi_t^i = \lfloor N\lambda_t^i \rfloor + (h - \lfloor g \rfloor)$
    end
  end
  set $g \leftarrow g - N\lambda_t^i$
  set $h \leftarrow h - \xi_t^i$
end
set $\xi_t^N = h$

compute the index set $(k_i)_{i=1}^N$ from the set of offspring $(\xi_t^i)_{i=1}^N$ is summarized in Algorithm 3.4.

Now, in light of Algorithm 3.4 it is interesting to note that even outside the APF framework we inevitably have to use indices for resampling, although in e.g. SIR they are not explicitly included as a part of the particle system. In this case, the usual interpretation for resampling is as a procedure for drawing $(\tilde{z}_{l:t-1}^i)_{i=1}^N$ with replacement from the empirical distribution

$$\hat{p}(z_{l:t-1}|y_{1:t}) := \sum_{i=1}^{N} \lambda_t^i \delta_{z_{l:t-1}^i}(dz_{l:t-1}). \tag{3.12}$$

However, as discussed above, this is usually accomplished by drawing a set of offspring $(\xi_t^i)_{i=1}^N$ and computing their corresponding indices using a procedure similar to Algorithm 3.4.

Finally, for completeness we briefly describe in Algorithm 3.5 a fast $\mathcal{O}(N)$ procedure for directly producing index draws from a Multinomial distribution. This method was originally proposed by Vose (1991) and consists of building in $\mathcal{O}(N)$ operations an alias table from which draws can be produced in $\mathcal{O}(1)$ time. Note that in practice we strongly

Algorithm 3.4: Index Sampling

Initialization
set $i = 1$
set $c = 0$

Main recursion
while $c < N$ **do**
  if $\xi_t^i > 0$ **then**
    while $\xi_t^i > 0$ **do**
      set $c \leftarrow c + 1$
      set $k_c = i$
      set $\xi_t^i \leftarrow \xi_t^i - 1$
    end
  end
  set $i \leftarrow i + 1$
end

recommend that multinomial resampling be avoided at all costs, and used only for benchmarking different methods. In particular, it can be proven (Künsch, 2005) that by using the branching algorithm the additional Monte Carlo variance introduced in the particle system is reduced by a factor of at least two when compared with multinomial sampling.

3.2.3 Regularization

Other than adopting more efficient resampling techniques as mentioned in Section 3.2.2, still another avenue for mitigating path degeneracy comprises a set of techniques that can be collectively referred to as *regularization* algorithms (Musso et al., 2001). In essence, regularization is a modification of the resampling step to allow for resampled particles to assume values other than the ones specified by the current set of particles. In the above example, this means that with regularization the number of unique values in $(\tilde{z}_{l:t-1}^i)_{i=1}^N$ is typically larger than the number of unique values in $(z_{l:t-1}^i)_{i=1}^N$, increasing diversity. This also allows for additional exploration of the state space, which is specially important for static parameters since their support will no longer be limited to a subset of values initially drawn from the prior.

Recall from Section 3.2.2 that the resampling step can be interpreted as drawing a set $(\tilde{z}_{l:t-1}^i)_{i=1}^N$ with replacement from the empirical distribution $\hat{p}(z_{l:t-1}|y_{1:t})$ given in (3.12). The corresponding regularized distribution is then defined as the convolution of $\hat{p}(z_{l:t-1}|y_{1:t})$ with a *regularization kernel* (Silverman, 1986) $K(\cdot)$, i.e.

$$\begin{aligned}
\tilde{p}(z_{l:t-1}|y_{1:t}) &:= \int K(z_{l:t-1} - z_{l:t-1}^*)\hat{p}(z_{l:t-1}^*|y_{1:t})dz_{l:t-1}^* \\
&= \int K(z_{l:t-1} - z_{l:t-1}^*) \sum_{i=1}^N \lambda_t^i \delta_{z_{l:t-1}^i}(dz_{l:t-1}^*)dz_{l:t-1}^* \\
&= \sum_{i=1}^N \lambda_t^i K(z_{l:t-1} - z_{l:t-1}^i). \qquad (3.13)
\end{aligned}$$

Algorithm 3.5: Alias-based Multinomial Resampling

Initialization
for *i = 1* **to** N **do**
    set $p_i = \lambda_t^i$
end

Alias Table
set $s = 1$
set $l = 1$
for *i = 1* **to** N **do**
    if $\lambda_t^i > 1/N$ **then**
        set $\text{large}_l = i$
        set $l \leftarrow l + 1$
    end
    else
        set $\text{small}_s = i$
        set $s \leftarrow s + 1$
    end
end
while $s \neq 1$ *and* $s \neq 1$ **do**
    set $s \leftarrow s - 1$
    set $i = \text{small}_s$
    set $j = \text{large}_l$
    set $\text{prob}_i = N\lambda_t^i$
    set $\text{alias}_i = j$
    set $p_j \leftarrow p_j + (p_i - 1/N)$
    if $p_j > 1/N$ **then**
        set $\text{large}_l = j$
        set $l \leftarrow l + 1$
    end
    else
        set $\text{small}_s = j$
        set $s \leftarrow s + 1$
    end
end
while $s > 1$ **do**
    set $s \leftarrow s - 1$
    set $\text{prob}_{\text{small}_s} = 1$
end
while $l > 1$ **do**
    set $l \leftarrow l - 1$
    set $\text{prob}_{\text{large}_l} = 1$
end

Index Sampling
for *i = 1* **to** N **do**
    draw $u \sim U[0, N]$
    set $j = \lfloor u \rfloor$
    if $(u - j) \leq prob_j$ **then**
        set $k_i = j$
    end
    else
        set $k_i = \text{alias}_j$
    end
end

Traditionally, $K(\cdot)$ is assumed to be the probability density of a continuous random variable with zero mean and finite second moment taking values in $\mathbb{R}^{(t-1-l)\times d_z}$, where $d_z := \dim(Z_t)$. By replacing the empirical measure (3.12) with the regularized measure (3.13), it is then clear that the set of possible values assumed by $(\tilde{z}^i_{l:t-1})_{i=1}^N$ effectively goes from the finite set $(z^i_{l:t-1})_{i=1}^N$ to the uncountable image set of $K(\cdot)$. Note that $z^*_{l:t-1}$ used in the above derivation is only an integration variable; the density $\hat{p}$ in the first and second integral is still (3.12).

As mentioned before, regularization is specially effective in sequential parameter learning due to the fact that it allows for exploration of the parameter space by otherwise static parameters, thus giving them "artifical dynamics". The first widely succesful application of this idea is in the method proposed by Liu and West (2001), which relies on a Gaussian kernel with location and scale determined by past parameter values and an additional user-defined scale specified via discount factors (see also Section 3.2.4.2). In order to obtain a more general framework, however, here we will assume that $K(\cdot)$ is any probability distribution density, and develop an auxiliary variable formalism for regularization analogous to the one for resampling within the APF.

More specifically, let $(\tilde{z}^i_{l:t-1})_{i=1}^N$ be the set of resampled values drawn from (3.12). From the auxiliary variable interpretation presented so far we clearly have $\tilde{z}^i_{l:t-1} = z^{k_i}_{l:t-1}$ for each i, and by definition this is also equivalent to $z^{k_i}_{l:t-1} =: (z^i_{l:t-1}, k_i)$. Therefore, denoting by $\tilde{z}^i_{l:t-1}$ the regularized value instead of the resampled one, we can interpret the joint particle $(z^i_{l:t-1}, k_i)$ as a draw from the regularized measure (3.13) instead of the empirical measure (3.12), keeping the definition for k_i intact in the process. That is, we can reinterpret the regularization procedure as drawing the particle $\tilde{z}^i_{l:t-1} := (z^i_{l:t-1}, k_i)$ by first sampling k_i with probability λ^i_t and then sampling a value from $K(z_{l:t-1} - z^{k_i}_{l:t-1})$. This effectively generalizes the usual auxiliary interpretation for resampling in the APF, which here is clearly obtained by choosing $K(x) = \delta(x)$, i.e. the Dirac delta function, as the regularization kernel. For completeness, the weight recursion under regularization is

$$\pi_t \propto \frac{w^k_{t-1}}{\lambda^k_t} \frac{f(x_t|\tilde{x}_{t-1}, \tilde{\theta}_{t-1}) g(y_t|x_t, \tilde{\theta}_{t-1})}{q(x_t|\tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})} \frac{p(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})}{q(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})}. \tag{3.14}$$

Although this novel auxiliary variable interpretation of regularization might appear to be of little consequence, we will show in the examples of Section 3.2.4 below that this formalism is essentially what allows for our framework to include so many of the most common methods for sequential parameter learning found in the literature. In particular, in algorithms in which the sampling order is the opposite of the one adopted here (i.e. in which we first sample θ_t and then X_t) and exploration of the parameter space is only done through regularization, we can simply take $p(\theta_t|x_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) = \delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)$, so that $\theta^i_t = \tilde{\theta}^i_{t-1}$ for each t and i.

Algorithm 3.6 summarizes a generic method in the class of sequential parameter learning methods with regularization described so far. As stated before, if regularization is considered as an integral part of the resampling step, Algorithm 3.6 is simply Algorithm 3.2 with $z^{k_i}_{l:t-1}$ replaced by $\tilde{z}^i_{l:t-1}$ for both $z^{k_i}_{l:t-1} = x^{k_i}_{l:t-1}$ and $z^{k_i}_{l:t-1} = \theta^{k_i}_{l:t-1}$, with $l = 0$ or $l = t-1$.

3.2.4 Special cases

We now show how the framework proposed here can accommodate several of the algorithms for sequential parameter learning found in the literature as special cases. We

Algorithm 3.6: Sequential Parameter Learning with Regularization

Initialization
for $i = 1$ **to** N **do**
  draw $\theta_0^i \sim p(\theta)$
  draw $x_0^i \sim \nu(x_0|\theta_0^i)$
  set $\pi_0^i \propto 1$
end
for $i = 1$ **to** N **do**
  set $w_0^i = 1/N$
end

Main recursion
for $t = 1$ **to** n **do**
  for $i = 1$ **to** N **do**
    sample k_i from $\{1, \ldots, N\}$ with probability λ_t^i
    draw $(\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i) \sim K\big((x_{0:t-1}, \theta_{0:t-1}) - (x_{0:t-1}^{k_i}, \theta_{0:t-1}^{k_i})\big)$
    draw $x_t^i \sim q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$
    draw $\theta_t^i \sim q(\theta_t|x_t, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$
    compute $\pi_t^i \propto \frac{w_{t-1}^{k_i}}{\lambda_t^{k_i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})} \frac{p(\theta_t^i|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})}{q(\theta_t^i|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})}$
  end
  for $i = 1$ **to** N **do**
    compute $w_t^i = \pi_t^i / \sum_{j=1}^N \pi_t^j$
  end
end

have grouped existing methods according to our subjective perception of their defining features, and afterwards propose three novel algorithms as an illustration of the flexibility allowed by our framework.

3.2.4.1 Particle Jittering

Introduced in Gordon et al. (1993), *jittering* consists of adding small amounts of randomness to the particles during the resampling step so as to allow for further exploration of the state space in the case of slow-moving states (this was called a "roughening procedure" in the original paper). This idea was later expanded by Kitagawa (1998), which by taking static parameters to be a part of the ("augmented") state vector, allowed the method to be suitable for sequential parameter learning.

Essentially, jittering is simply an instance of regularization with an additive kernel with location $(x_{t-1}^{k_i}, \theta_{t-1}^{k_i})$ and user-defined variance matrix V_{t-1}, i.e.

$$K\big((x_{0:t-1}, \theta_{0:t-1}) - (x_{0:t-1}^{k_i} - \theta_{0:t-1}^{k_i})\big) = dG\big((x_{t-1}, \theta_{t-1})|(x_{t-1}^{k_i}, \theta_{t-1}^{k_i}), V_{t-1}\big) \cdot \delta_{(x_{0:t-2}^{k_i}, \theta_{0:t-2}^{k_i})}(dx_{0:t-2} d\theta_{0:t-2}), \quad (3.15)$$

where $dG(x|\mu, \Sigma)$ is the density of any continuous random variable (usually Gaussian) in $\mathbb{R}^{d_x \times d_\theta}$ with mean vector μ and variance matrix Σ, evaluated at point x. Note that here

only the most recent state and parameter particles $(x^i_{t-1}, \theta^i_{t-1})^N_{i=1}$, are regularized, since the ensuing Dirac measures imply that $(\tilde{x}^i_{0:t-2}, \tilde{\theta}^i_{0:t-2}) = (x^{k_i}_{0:t-2}, \theta^{k_i}_{0:t-2})$.

Since jittering was originally proposed within the SIR framework, the intermediate weights are given by $\lambda^i_t = w^i_{t-1}$ and the state proposal $q(x_t|\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$ is completely user-defined. The target parameter distribution is assumed to satisfy[5]

$$p(\theta_t|x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) = \delta_{\tilde{\theta}^i_{t-1}}(d\theta_t) \tag{3.16}$$

and, by taking $q(\theta_t|x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) = p(\theta_t|x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$, the corresponding weights (3.14) are

$$\pi^i_t \propto \frac{w^{k_i}_{t-1}}{w^{k_i}_{t-1}} \frac{f(x^i_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}) g(y_t|x^i_t, \tilde{\theta}^i_{t-1})}{q(x^i_t|\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})} \frac{\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)}{\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)} = \frac{f(x^i_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}) g(y_t|x^i_t, \tilde{\theta}^i_{t-1})}{q(x^i_t|\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})}. \tag{3.17}$$

3.2.4.2 Liu and West's Filter

Liu and West (2001)'s (LW) filter is widely recognized as the first successful sequential parameter learning method, and the main benchmark in the literature. As in particle jittering, its main reasoning is to allow for static parameters to properly explore their state space through regularization, but its adoption of the APF framework and adaptive kernel density estimation techniques make for a much more efficient method.

More specifically, the LW filter adopts the kernel

$$K\big((x_{0:t-1}, \theta_{0:t-1}) - (x^{k_i}_{0:t-1} - \theta^{k_i}_{0:t-1})\big) = d\mathcal{N}(\theta_{t-1}|m^{k_i}_{t-1}, h^2 V_{t-1}) \cdot \delta_{(x^{k_i}_{0:t-1}, \theta^{k_i}_{0:t-2})}(dx_{0:t-1} d\theta_{0:t-2}), \tag{3.18}$$

where

$$m^i_{t-1} := a\theta^i_{t-1} + (1-a)\bar{\theta}_{t-1}, \qquad \bar{\theta}_{t-1} := \sum_{i=1}^N \theta^i_{t-1}, \tag{3.19}$$

$$V_{t-1} := \sum_{i=1}^N w^i_{t-1} [\theta^i_{t-1} - \bar{\theta}_{t-1}][\theta^i_{t-1} - \bar{\theta}_{t-1}]^T, \qquad 0 \leq h \leq 1, \tag{3.20}$$

with $a = \sqrt{1-h^2}$. Note that in the LW filter only the most recent θ^i_{t-1}'s are regularized, whereas the states are simply resampled.

The choice of kernel (3.18) builds on the work of *shrinkage location* by West (1993a,b), yielding draws that are more concentrated around their mean $\bar{\theta}_{t-1}$ than simply using the usual jittering locations $\theta^{k_i}_{t-1}$. As a consequence, the overdispersion over time that usually affects kernel-based estimates (Liu and West, 2001) is completely avoided. By also requiring that $a^2 + h^2 = 1$, the method guarantees that the first two second moments of the regularized estimates $(\tilde{\theta}^i_{t-1})^N_{i=1}$ will be the same as those of $(\theta^i_{t-1})^N_{i=1}$, i.e. $\bar{\theta}_{t-1}$ and V_{t-1}. The kernel *bandwidth* h is usually selected according to a *discount factor* (West and Harrison, 1997) $\delta \in (0, 1]$ via $h = \sqrt{1 - [(3\delta - 1)/2\delta]^2}$, which according to the authors should typically be around 0.95 to 0.99.

As for the other choices, the LW filter uses a lookahead strategy by taking (μ^i_t, m^i_{t-1}) as the best guess for (x^i_t, θ^i_t), where here specifically $\mu^i_t = \mathbb{E}_{\mathbb{P}}(X_t|x^i_{0:t-1}, \theta^i_{0:t-1})$. This results in

[5] Note that drawing θ^i_t from $\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)$ simply amounts to setting $\theta^i_t = \tilde{\theta}^i_{t-1}$.

the intermediate weights $\lambda_t^i \propto w_{t-1}^i g(y_t|\mu_t^i, m_{t-1}^i)$ and state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{t-1}^i, y_{1:t})$ $= f(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)$. Similar to jittering, it is also assumed that

$$p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = \delta_{\tilde{\theta}_{t-1}^i}(d\theta_t), \tag{3.21}$$

so that movement along the parameter space is only made through regularization. Finally, we also have $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, yielding weights (3.14) equal to

$$\pi_t^i \propto \frac{w_{t-1}^{k_i}}{w_{t-1}^{k_i} g(y_t|\mu_t^{k_i}, m_{t-1}^{k_i})} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)} \frac{\delta_{\tilde{\theta}_{t-1}^i}(d\theta_t)}{\delta_{\tilde{\theta}_{t-1}^i}(d\theta_t)} = \frac{g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{g(y_t|\mu_t^{k_i}, m_{t-1}^{k_i})}. \tag{3.22}$$

Note that in its original version the LW filter first samples the static parameters θ_t^i from the kernel (3.18) and then samples the states x_t^i conditional on θ_t^i. However, as mentioned before, this is accommodated in our framework by setting $\theta_t^i = \tilde{\theta}_{t-1}^i$, yielding an equivalent implementation of the method.

3.2.4.3 Smooth Jittering

Introduced in Flury and Shephard (2009), the so-called *smoothly jittered particle filter* builds on previous work on smoothed bootstraps in the SMC context (Stravropoulos and Titterington, 2001) in order to provide asymptotically optimal choices and increase the overall efficiency of particle jittering.

The regularization kernel for the smooth jittering method is

$$K\big((x_{0:t-1}, \theta_{0:t-1}) - (x_{0:t-1}^{k_i} - \theta_{0:t-1}^{k_i})\big) = dG\big((x_{t-1}, \theta_{t-1})|\zeta_{t-1}^{k_i}, \hat{H}_{t-1}\big) \cdot \delta_{(x_{0:t-2}^{k_i}, \theta_{0:t-2}^{k_i})}(dx_{0:t-2} d\theta_{0:t-2}), \tag{3.23}$$

where $\zeta_{t-1}^i := (\zeta_{1,t-1}^i, \ldots, \zeta_{d_x+d_\theta,t-1}^i)$ and $\hat{H}_{t-1} := \text{diag}(\hat{h}_{1,t-1}, \ldots, \hat{h}_{d_x+d_\theta,t-1})$ are the kernel locations and bandwidths, respectively. Here, each location $\zeta_{j,t-1}^i$ and bandwidth $h_{j,t-1}$ are defined independently, according to

$$\zeta_{j,t-1}^i := \hat{\mu}_{j,t-1} + \sqrt{\frac{\hat{\sigma}_{j,t-1}^2 - \hat{h}_{j,t-1}^2}{\hat{\sigma}_{j,t-1}^2}} (z_{j,t-1}^i - \hat{\mu}_{j,t-1}) \tag{3.24}$$

and

$$h_{j,t-1} := 1.59[\hat{R}(\mathcal{X}, \Theta|y_{1:t-1})]^{1/3} \hat{\sigma}_{j,t-1} N^{-1/3}, \tag{3.25}$$

where $z_{j,t-1}^i$ is the jth element of the vector $(x_{t-1}^i, \theta_{t-1}^i)$, $j = 1, \ldots, d_x + d_\theta$, $\hat{\mu}_{j,t-1}$ and $\hat{\sigma}_{j,t-1}^2$ are estimates of the mean and variance of $(z_{j,t-1}^i)_{i=1}^N$ and $\hat{R}(\mathcal{X}, \Theta|y_{1:t-1})$ is an estimate of the functional defined by

$$R(\mathcal{X}, \Theta|y_{1:t-1}) := \int_{\mathcal{X}\times\Theta} \frac{g(y_{t-1}|x_{t-1}, \theta_{t-1})}{p(y_{t-1}|y_{1:t-2})} p(x_{t-1}, \theta_{t-1}|y_{1:t-1}) dx_{t-1} d\theta_{t-1}. \tag{3.26}$$

Note that although the location shrinkage equation (3.24) might at first appear distinct from the LW filter shrinkage (3.19), it can be easily seen that they are equivalent (for a unidimensional parameter) by taking $z_{j,t-1}^i = \theta_{t-1}^i$, $\hat{\mu}_{j,t-1}^i = \bar{\theta}_{t-1}$ and $\hat{h}_{j,t-1} = h\hat{\sigma}_{j,t-1}^2$.

Smooth jittering was proposed within a SIR framework, more specifically a bootstrap filter. Therefore, here the intermediate weights are $\lambda_t^i = w_{t-1}^i$, and the state proposal is $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{t-1}^i, y_{1:t}) = f(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)$. Here we again assume that

$$p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = \delta_{\tilde{\theta}_{t-1}^i}(d\theta_t) \tag{3.27}$$

and, with $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, we get importance weights (3.14) given by

$$\pi_t^i \propto \frac{w_{t-1}^{k_i}}{w_{t-1}^{k_i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)} \frac{\delta_{\tilde{\theta}_{t-1}^i}(d\theta_t)}{\delta_{\tilde{\theta}_{t-1}^i}(d\theta_t)} = g(y_t|x_t^i, \tilde{\theta}_{t-1}^i). \tag{3.28}$$

For a practical implementation of smooth jittering, Flury and Shephard (2009) prove that a consistent estimator of the functional (3.26) is given by

$$\hat{R}(\mathcal{X}, \Theta|y_{1:t-1}) := N \sum_{i=1}^{N} \left[\frac{g(y_{t-1}|x_{t-1}^i, \theta_{t-1}^i)}{\sum_{j=1}^{N} g(y_{t-1}|x_{t-1}^j, \theta_{t-1}^j)} \right]^2 = N \sum_{i=1}^{N} (w_{t-1}^i)^2 \tag{3.29}$$

since from (3.28) comes $w_{t-1}^i = g(y_{t-1}|x_{t-1}^i, \theta_{t-1}^i)/[\sum_{j=1}^{N} g(y_{t-1}|x_{t-1}^j, \theta_{t-1}^j)]$. Although by (3.24) the locations $\zeta_{j,t-1}^i$ can become undefined whenever the bandwidth $\hat{h}_{j,t-1}$ is greater than $\hat{\sigma}_{j,t-1}$ (which occurs if $\hat{R}(\mathcal{X}, \Theta|y_{1:t-1}) > N/4.02$), in practice this can be dealt with by increasing the number of particles N.

3.2.4.4 Resample-move

The *resample-move* algorithm proposed by Gilks and Berzuini (2001) introduces an idea that is in principle simple but very powerful: "rejuvenating" resampled paths $(x_{0:t-1}^{k_i}, \theta_{0:t-1}^{k_i})_{i=1}^N$ with *Markov Chain Monte Carlo* (MCMC) moves. This avoids (path) degeneracy entirely, since by drawing completely new paths from the MCMC kernel we guarantee diversity and therefore prevent sample impoverishment.

In our framework, the MCMC draws $(\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i)_{i=1}^N$ from the resample-move algorithm can be naturally formalized as draws from the regularization kernel

$$K\big((x_{0:t-1}, \theta_{0:t-1}) - (x_{0:t-1}^{k_i} - \theta_{0:t-1}^{k_i})\big) = M\big((x_{0:t-1}, \theta_{0:t-1}) - (x_{0:t-1}^{k_i} - \theta_{0:t-1}^{k_i})\big), \tag{3.30}$$

where $M(\cdot)$ is a Markov kernel having the target posterior $p(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1})$ as invariant density (see Section A.4). Since the algorithm is originally set within a SIR framework, the intermediate weights are $\lambda_t^i = w_{t-1}^i$, and the state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$ is user-defined. For the target parameter distribution, it is assumed again that

$$p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = \delta_{\tilde{\theta}_{t-1}^i}(d\theta_t) \tag{3.31}$$

and, with $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, the importance weight recursion (3.14) for this method is

$$\pi_t^i \propto \frac{w_{t-1}^{k_i}}{w_{t-1}^{k_i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})} \frac{\delta_{\tilde{\theta}_{t-1}^i}(d\theta_t)}{\delta_{\tilde{\theta}_{t-1}^i}(d\theta_t)} = \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})}. \tag{3.32}$$

Although in theory resample-move is very appealing, draws from the Markov kernel $M(\cdot)$ at each step are produced at $\mathcal{O}(t)$ complexity, making a complete run of the algorithm from times 1 through n an $\mathcal{O}(n^2 N)$ operation[6]. In order to circumvent this cost, the authors propose performing the rejuvenation step only at predefined times, according to user-defined probabilities $\mathcal{O}(t^{-\gamma})$ for $\gamma > 0$ (possibly different for each component) or even regularizing only a part of the resampled particles, $(x^i_{L:t-1}, \theta^i_{L:t-1})$ for integer $0 \leq L \leq t-1$. All of these modifications can be built directly into the definition of $M(\cdot)$.

An ingenious way to keep the complexity constant in the resample-move algorithm is to rely on a set of fixed-dimension sufficient statistics $\mathcal{S}_t := \mathcal{S}(X_{0:t}, Y_{0:t})$ which can be updated recursively by the map $\mathcal{S}_t \equiv \mathcal{S}(\mathcal{S}_{t-1}, X_t, Y_t)$. This idea was first proposed by Fearnhead (2002) within an APF framework and, when applicable (as e.g. in exponential families), it effectively transforms rejuvenation into a Gibbs sampling step, in which draws $(\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})$ then consist of sampling from $p(x_{t-1}, \theta_{t-1} | x^{k_i}_{0:t-1}, \theta^{k_i}_{0:t-1}) = p(x_{t-1}, \theta_{t-1} | \mathcal{S}^{k_i}_{t-1})$.

3.2.4.5 Storvik's Filter

Similarly inspired by the idea of using sets of fixed-dimensional sufficient statistics in order to perform Gibbs sampling moves, Storvik (2002) proposed a generalization of the SIR method capable of also performing inference for static parameters. This technique has the additional interpretation of a procedure that marginalizes the static parameters out of the joint target posterior $p(x_{0:t}, \theta_{0:t} | y_{1:t})$, providing additional theoretical justification for its effectiveness in mitigating degeneracy.

In Storvik's filter, only the current static parameters θ^i_{t-1} are regularized; current states x^i_{t-1} and past trajectories $(x^i_{0:t-2}, \theta^i_{0:t-2})$ are simply resampled. Since $\tilde{\theta}^i_{t-1}$ is sampled from the complete conditional $p(\theta_{t-1} | \mathcal{S}^{k_i}_{t-1})$, the regularization kernel here is then given by

$$K\big((x_{0:t-1}, \theta_{0:t-1}) - (x^{k_i}_{0:t-1} - \theta^{k_i}_{0:t-1})\big) = p(\theta_{t-1} | \mathcal{S}^{k_i}_{t-1}) \delta_{(x^{k_i}_{0:t-1}, \theta^{k_i}_{0:t-2})}(dx_{0:t-1} d\theta_{0:t-2}). \quad (3.33)$$

Note that the recursive update of the sufficient statistics via $\mathcal{S}^i_t = \mathcal{S}(\mathcal{S}^{k_i}_{t-1}, x^i_t, y_t)$ is performed deterministically at each time t after the newly propagated states x^i_t are available.

Since Storvik's method relies on regularization (and since in the original formulation θ_t is sampled before X_t, as in the LW filter), we again have that the target parameter distribution must satisfy

$$p(\theta_t | x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) = \delta_{\tilde{\theta}^i_{t-1}}(d\theta_t). \quad (3.34)$$

As for the rest of the design choices, Storvik (2002) requires that the intermediate weights satisfy $\lambda^i_t = w^i_{t-1}$ (since the method is set within the SIR framework), leaving the state proposal $q(x_t | \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$ to be specified by the user. By once more choosing the parameter proposal $q(\theta_t | x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) = p(\theta_t | x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$, we have importance weights (3.14) of the form

$$\pi^i_t \propto \frac{w^{k_i}_{t-1}}{w^{k_i}_{t-1}} \frac{f(x^i_t | \tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}) g(y_t | x^i_t, \tilde{\theta}^i_{t-1})}{q(x^i_t | \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})} \frac{\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)}{\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)} = \frac{f(x^i_t | \tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}) g(y_t | x^i_t, \tilde{\theta}^i_{t-1})}{q(x^i_t | \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})}. \quad (3.35)$$

In cases where $p(\theta_{t-1} | \mathcal{S}_{t-1})$ can be evaluated pointwise but is still difficult/impossible to sample from, Storvik (2002) allows the parameter proposal to differ from $\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)$.

[6] Note that $\mathcal{O}(1 \cdot N) + \mathcal{O}(2 \cdot N) + \ldots + \mathcal{O}(n \cdot N) = \mathcal{O}(n^2 \cdot N)$.

Although this situation rarely occurs in practice, in our framework it can be accommodated by performing an additional importance sampling step to sample from the kernel (3.33). This marginal IS step has importance weights given by $p(\tilde{\theta}^i_{t-1}|\mathcal{S}^{k_i}_{t-1})/K_q(\tilde{\theta}^i_{t-1}|x^{k_i}_{0:t-1}, \theta^{k_i}_{0:t-2}, y_{1:t})$, where $K_q(\cdot)$ is a kernel that we can easily sample from. As a result, the original weight recursion of the method (3.35) must also be multiplied by this marginal importance weight.

3.2.4.6 Particle Learning

Although also relying on sufficient statistics, the more recent *Particle Learning* (PL) technique of Carvalho et al. (2010) is set within a fully-adapted APF instead of the SIR framework adopted by previous methods. This is advantageous because fully-adapted procedures have minimal variance importance weights (see Proposition 2.1.1) and consequently less variable resampling weights, which in turn minimizes sample impoverishment and therefore path degeneracy. Another benefit pointed out by the authors (see also the footnote on Section 3.2.1) is that the resulting method is then a resample-propagate filter instead of a propagate-resample one, further contributing to the mitigation of degeneracy on static parameters' paths.

In PL, no regularization takes place; only resampling is performed. This implies taking

$$K\big((x_{0:t-1}, \theta_{0:t-1}) - (x^{k_i}_{0:t-1} - \theta^{k_i}_{0:t-1})\big) = \delta_{(x^{k_i}_{0:t-1}, \theta^{k_i}_{0:t-1})}(dx_{0:t-1} d\theta_{0:t-1}). \tag{3.36}$$

Since the procedure is set within the FAPF framework, the intermediate weights and the state proposal must satisfy $\lambda^i_t \propto w^i_{t-1} p(y_t|x^i_{t-1}, \theta^i_{t-1})$ and $q(x_t|\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) = p(x_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}, y_t)$. As for the target parameter distribution, the main underlying assumption here is that it satisfies

$$p(\theta_t|x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) = p(\theta_t|\mathcal{S}^i_t). \tag{3.37}$$

Additionally, it is also assumed that we can sample from $p(\theta|\mathcal{S}^i_t)$ and, therefore, that we can set $q(\theta_t|x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) = p(\theta_t|\mathcal{S}^i_t)$. The importance weights (3.14) for PL are thus given by

$$\pi^i_t \propto \frac{w^{k_i}_{t-1}}{w^{k_i}_{t-1} p(y_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1})} \frac{f(x^i_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}) g(y_t|x^i_t, \tilde{\theta}^i_{t-1})}{p(x^i_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}, y_t)} \frac{p(\theta^i_t|\mathcal{S}^i_t)}{p(\theta^i_t|\mathcal{S}^i_t)} = 1, \tag{3.38}$$

which in retrospect should be obvious given that the procedure is fully-adapted. Note that in the above derivation we have once again used that $p(x^i_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}, y_t) = f(x^i_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}) \cdot g(y_t|x^i_t, \tilde{\theta}^i_{t-1})/p(y_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1})$.

In the original formulation of PL, the sufficient statistics were included as part of an extended vector $Z_t := (\mathcal{S}_t, X_t, \theta_t)$ and inference was actually performed for Z_t instead of the pair (X_t, θ_t). Although unnecessary, this can be accommodated within our framework by rederiving target and proposal recursions for $Z_{0:t}$ instead of $(X_{0:t}, \theta_{0:t})$. The end result is the same as that of Carvalho et al. (2010): for each time t, we have a target (and an equal proposal, cancelling its terms in the corresponding importance weight recursion) for $\mathcal{S}_t$ given by $p(\mathcal{S}_t|\mathcal{S}^{k_i}_{t-1}, x^i_t, y_t)$ corresponding to a deterministic update made through the map $\mathcal{S}^i_t = \mathcal{S}(\mathcal{S}^{k_i}_{t-1}, x^i_t, y_t)$.

Finally, Carvalho et al. (2010) also consider the possibility of using state sufficient statistics $\mathcal{T}_t := \mathcal{T}(X_{0:t}, \theta_{0:t}, Y_{1:t})$ whenever available, which unlike $\mathcal{S}_t$ can also be made

functions of the static parameters. These statistics are also set to satisfy a recursive mapping $\mathcal{T}_t \equiv \mathcal{T}(T_{t-1}, X_t, \theta_t, Y_t)$ and their use is feasible if the conditional posterior $p(x_t|\mathcal{T}_{t-1}, y_t)$ is analytically available. In this case, the authors justify via *Rao-Blackwellization* (see Proposition A.3.1) that the corresponding intermediate weights $w_{t-1}^i p(y_t|\mathcal{T}_{t-1})$ will be less variable than $w_{t-1}^i p(y_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)$, yielding a more efficient procedure.

3.2.4.7 Hybrid LW-PL Filter

Introduced by Chen et al. (2010), this method is essentially a LW filter that also incorporates Gibbs sampling for the static parameters for which sufficient statistics are available, resulting in a hybrid version of Liu and West (2001) and Carvalho et al. (2010)'s algorithms.

Let $\theta = (\phi, \varphi)$, where ϕ is the subset of static parameters for which LW moves are made and φ is the subset for which draws conditional on sufficient statistics $\mathcal{S}_t$ are made. The regularization kernel is then

$$K\big((x_{0:t-1}, \theta_{0:t-1}) - (x_{0:t-1}^{k_i} - \theta_{0:t-1}^{k_i})\big) = d\mathcal{N}(\phi_{t-1}|m_{t-1}^{k_i}, h^2 V_{t-1})\delta_{\varphi_{t-1}^{k_i}}(d\varphi_{t-1})\delta_{(x_{0:t-1}^{k_i}, \theta_{0:t-2}^{k_i})}(dx_{0:t-1}d\theta_{0:t-2}), \tag{3.39}$$

where m_{t-1}^i and V_{t-1} are the same as in (3.19) but computed only for the subset $(\phi_{t-1}^i)_{i=1}^N$ instead of the entire $(\theta_{t-1}^i)_{i=1}^N$.

Under the same lookahead strategy within the APF framework adopted for the LW filter, here we also have intermediate weights $\lambda_t^i \propto w_{t-1}^i g(y_t|\mu_t^i, m_{t-1}^i, \varphi_{t-1}^i)$ and state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = f(x_t|\tilde{x}_{t-1}^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)$, where $\mu_t^i := \mathbb{E}_{\mathbb{P}}(X_t|x_{t-1}^i, \phi_{t-1}^i, \varphi_{t-1}^i)$. As for the target parameter distribution, we assume

$$p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = \delta_{\tilde{\phi}_{t-1}^i}(d\phi_t)p(\varphi_t|\mathcal{S}_t^i). \tag{3.40}$$

and, by letting $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, we have importance weights (3.14) given by

$$\begin{aligned}\pi_t^i &\propto \frac{w_{t-1}^{k_i}}{w_{t-1}^{k_i} g(y_t|\mu_t^{k_i}, m_{t-1}^i, \tilde{\varphi}_{t-1}^i)} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i) g(y_t|x_t^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)}{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)} \frac{\delta_{\tilde{\phi}_{t-1}^i}(d\phi_t)p(\varphi_t|\mathcal{S}_t^i)}{\delta_{\tilde{\phi}_{t-1}^i}(d\phi_t)p(\varphi_t|\mathcal{S}_t^i)} \\ &= \frac{g(y_t|x_t^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)}{g(y_t|\mu_t^{k_i}, m_{t-1}^i, \tilde{\varphi}_{t-1}^i)}.\end{aligned} \tag{3.41}$$

Here the sufficient statistics are updated in the same way as in the PL algorithm, i.e. by the map $\mathcal{S}_t^i = \mathcal{S}(\mathcal{S}_{t-1}^{k_i}, x_t^i, y_t)$.

Aside from the hybrid LW-PL filter of Chen et al. (2010), Rios and Lopes (2010) also proposed a hybrid LW method but based on Storvik (2002)'s formulation of the Gibbs sampling step associated with the propagated sufficient statistics for φ. The algorithm is very similar to the one presented in this section, and can be obtained by simply replacing $\delta_{\varphi_{t-1}^{k_i}}(d\varphi_{t-1})$ with $p(\varphi_{t-1}|\mathcal{S}_{t-1}^{k_i})$ in (3.39) and $p(\varphi_t|\mathcal{S}_t^i)$ with $\delta_{\tilde{\varphi}_{t-1}^i}(d\varphi_t)$ in (3.40). Note that for both Chen et al. (2010) and Rios and Lopes (2010) algorithms the same lookahead APF framework is adopted, since the LW filter is actually the "baseline" algorithm, making the Gibbs sampling moves actually optional. Naturally, both methods collapse back to the LW filter whenever $\theta = \phi$.

3.2.4.8 Fully-adapted Liu and West's Filter

We will now illustrate the flexibility allowed by the novel framework proposed in this thesis with the introduction of three novel sequential parameter learning algorithms. The first of these will be hereafter referred to as *Fully-adapted Liu and West's* (FALW) *filter*. As its name implies, this method consists of choosing a regularization kernel

$$K\big((x_{0:t-1},\theta_{0:t-1})-(x^{k_i}_{0:t-1}-\theta^{k_i}_{0:t-1})\big)= \\ d\mathcal{N}\big((x_{t-1},\theta_{t-1})|m^{k_i}_{t-1},h_{t-1}V_{t-1}h^T_{t-1}\big)\delta_{(x^{k_i}_{0:t-2},\theta^{k_i}_{0:t-2})}(dx_{0:t-2}d\theta_{0:t-2}) \tag{3.42}$$

similar to that of the LW filter, but under a FAPF framework instead of the original lookahead APF one. This is done by choosing intermediate weights $\lambda^i_{t-1}\propto w^i_{t-1}p(y_t|x^i_{t-1},\theta^i_{t-1})$ and state proposal $q(x^i_t|\tilde{x}^i_{0:t-1},\tilde{\theta}^i_{0:t-1},y_{1:t})=p(x_t|\tilde{x}^i_{t-1},\tilde{\theta}^i_{t-1},y_t)$. By also assuming that

$$p(\theta_t|x^i_t,\tilde{x}^i_{0:t-1},\tilde{\theta}^i_{0:t-1},y_{1:t})=\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t), \tag{3.43}$$

and that $q(\theta_t|x^i_t,\tilde{x}^i_{0:t-1},\tilde{\theta}^i_{0:t-1},y_{1:t})=p(\theta_t|x^i_t,\tilde{x}^i_{0:t-1},\tilde{\theta}^i_{0:t-1},y_{1:t})$, the corresponding importance weights (3.14) here become

$$\pi^i_t\propto\frac{w^{k_i}_{t-1}}{w^{k_i}_{t-1}p(y_t|\tilde{x}^i_{t-1},\tilde{\theta}^i_{t-1})}\frac{f(x^i_t|\tilde{x}^i_{t-1},\tilde{\theta}^i_{t-1})g(y_t|x^i_t,\tilde{\theta}^i_{t-1})}{p(x^i_t|\tilde{x}^i_{t-1},\tilde{\theta}^i_{t-1},y_t)}\frac{\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)}{\delta_{\tilde{\theta}^i_{t-1}}(d\theta_t)}=1. \tag{3.44}$$

Although at first sight the modifications to the original method by Liu and West (2001) that define FALW here might not warrant the definition of an entirely new algorithm, we highlight from the discussion at the end of Section 3.2.1 that full adaptation in this case is only possible due to our formalization of LW moves as draws from a regularization kernel, which essentially allows us to reverse the original sampling order from θ first and then X_t to X_t first and then θ. This is a situation which we have not encountered outside of our framework.

In light of Flury and Shephard (2009)'s work, we also make an additional improvement to FALW by choosing a diagonal variance matrix and the optimal bandwidth proposed by these authors in their smooth jittering method described in Section 3.2.4.3. More specifically, in (3.42) we take shrinkages $m^i_{t-1}:=(m^i_{1,t-1},\ldots,m^i_{d_x+d_\theta,t-1})$, bandwidths $h_{t-1}:=(h_{1,t-1},\ldots,h_{d_x+d_\theta,t-1})$ and variance matrix $V_{t-1}:=\mathrm{diag}((\sigma^i)^2_{1,t-1},\ldots,(\sigma^i)^2_{d_x+d_\theta,t-1})$ equal to, respectively,

$$m^i_{j,t-1}:=a_{j,t-1}z^i_{j,t-1}+(1-a_{j,t-1})\bar{z}^i_{j,t-1},\quad \bar{z}^i_{j,t-1}:=\sum_{i=1}^N w^i_{t-1}z^i_{j,t-1}, \tag{3.45}$$

$$h_{j,t-1}:=1.59[\hat{R}(\mathcal{X},\Theta|y_{1:t-1})]^{1/3}N^{-1/3},\quad \sigma^i_{j,t-1}:=\sqrt{\sum_{i=1}^N w^i_{t-1}(z^i_{j,t-1}-\bar{z}^i_{j,t-1})^2}, \tag{3.46}$$

where $a_{j,t-1}=\sqrt{1-h^2_{j,t-1}}$ and $z^i_{j,t-1}$ is the jth element of $(x^i_{t-1},\theta^i_{t-1})$, $j=1,\ldots,d_x+d_\theta$. Note that here the states x^i_{t-1} are also regularized along with the static parameters, analogous to the smooth jittering method.

Now, the estimate $\hat{R}(\mathcal{X},\Theta|y_{1:t-1})$ appearing in (3.46) should not be confused with (3.29), since although it is also a particle approximation to the same functional (3.26)

as in smooth jittering, the particle system here is different. For FALW, we can show (Appendix D) that

$$\hat{R}(\mathcal{X}, \Theta|y_{1:t-1}) = \sum_{i=1}^{N} \frac{g(y_{t-1}|x_{t-1}^i, \theta_{t-1}^i)}{\sum_{j=1}^{N} p(y_{t-1}|x_{t-2}^j, \theta_{t-2}^j)}. \tag{3.47}$$

It turns out that expressions (3.29) and (3.47) are actually particular instances of an estimator of a more general regularization functional than (3.26). This estimator is based on the unbiased APF particle approximation to the likelihood proposed by Pitt et al. (2012) and extended here for our sequential parameter learning framework, and its derivation can be found in Appendix D.

3.2.4.9 Regularized Particle Learning

The second method introduced in this work is a regularized version of the PL algorithm of Carvalho et al. (2010), hereafter referred to as *Regularized Particle Learning* (RPL). The theoretical reasoning for RPL is that, in addition to updating current parameters with sufficient statistics, regularizing resampled states and parameters would mitigate path degeneracy even further. In addition, for this method we do not restrict our attention to fully-adapted procedures, which makes it applicable for a much broader class of models.

The regularization kernel adopted in RPL is the same as that of the FALW method, i.e.

$$\begin{aligned} K\big((x_{0:t-1}, \theta_{0:t-1}) - (x_{0:t-1}^{k_i} - \theta_{0:t-1}^{k_i})\big) = \\ d\mathcal{N}\big((x_{t-1}, \theta_{t-1})|m_{t-1}^{k_i}, h_{t-1}V_{t-1}h_{t-1}^T\big)\delta_{(x_{0:t-2}^{k_i}, \theta_{0:t-2}^{k_i})}(dx_{0:t-2}d\theta_{0:t-2}), \end{aligned} \tag{3.48}$$

where the components of m_{t-1}^i, h_{t-1} and V_{t-1} are defined in (3.45) and (3.46). Accordingly, by adopting a general APF framework, we let the intermediate weights λ_t^i and state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$ to be defined by the user. Since this method is based on Particle Learning, the target parameter distribution is assumed to satisfy

$$p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|\mathcal{S}_t^i), \tag{3.49}$$

where $\mathcal{S}_t^i = \mathcal{S}(\mathcal{S}_{t-1}^{k_i}, x_t^i, y_t)$, as usual. Finally, by also assuming $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$ $= p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, the weight recursion (3.14) for RPL is

$$\begin{aligned} \pi_t^i &\propto \frac{w_{t-1}^{k_i}}{\lambda_t^{k_i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})} \frac{p(\theta_t^i|\mathcal{S}_t^i)}{p(\theta_t^i|\mathcal{S}_t^i)} \\ &= \frac{w_{t-1}^{k_i}}{\lambda_t^{k_i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})}. \end{aligned} \tag{3.50}$$

Note that the regularization of the past static parameters θ_{t-1}^i in RPL only affects the importance weights π_t^i and current sampled states x_t^i, since the current parameters θ_t^i are sampled independently from θ_{t-1}^i. Moreover, by taking the optimal importance weights $\lambda_t^i \propto w_{t-1}^i p(y_t|x_{t-1}^i, \theta_{t-1}^i)$ and optimal state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i, y_t)$, the procedure is once again fully-adapted, since in this case the weights (3.50) are equal to those in (3.38), i.e. proportional to 1.

3.2.4.10 Hybrid FALW-RPL Filter

The last novel method introduced here is a hybrid between the FALW and RPL algorithms. Similar to the hybrid LW-PL algorithm of Chen et al. (2010), this technique has the benefit of allowing for Gibbs updates whenever sufficient statistics are available for a subset φ_t of the static parameter vector θ_t, while also allowing for regularization-based inference for the rest of the parameters to be performed. In fact, just as in RPL, even the past values of this subset φ_{t-1} are also regularized, with the aim to mitigate path degeneracy even further. In order to keep the procedure as general as possible, we opt for the usual APF framework adopted in RPL rather than a strictly fully-adapted one as in FALW.

Let $\theta = (\phi, \varphi)$, where φ is the subset for which $p(\varphi|\mathcal{S}_t)$ is available. The regularization kernel adopted here is the same as in both FALW and RPL, i.e.

$$K\big((x_{0:t-1}, \theta_{0:t-1}) - (x_{0:t-1}^{k_i} - \theta_{0:t-1}^{k_i})\big) = \\ d\mathcal{N}\big((x_{t-1}, \theta_{t-1})|m_{t-1}^{k_i}, h_{t-1}V_{t-1}h_{t-1}^T\big)\delta_{(x_{0:t-2}^{k_i}, \theta_{0:t-2}^{k_i})}(dx_{0:t-2}d\theta_{0:t-2}), \quad (3.51)$$

with m_{t-1}^i, h_{t-1} and V_{t-1} defined in (3.45-3.46). Accordingly, we also allow intermediate weights λ_t^i to be user-defined, along with the state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\phi}_{0:t-1}^i, \tilde{\varphi}_{0:t-1}^i, y_{1:t})$. As for the target parameter distribution, it is the same as in Chen et al. (2010), i.e.

$$p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = \delta_{\tilde{\phi}_{t-1}^i}(d\phi_t)p(\varphi_t|\mathcal{S}_t^i). \quad (3.52)$$

Therefore, by taking $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, we have the following importance weights (3.14):

$$\begin{aligned}\pi_t^i &\propto \frac{w_{t-1}^{k_i}}{\lambda_t^{k_i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)g(y_t|x_t^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\phi}_{0:t-1}^i, \tilde{\varphi}_{t-1}^i, y_{1:t})} \frac{\delta_{\tilde{\phi}_{t-1}^i}(d\phi_t)p(\varphi_t|\mathcal{S}_t^i)}{\delta_{\tilde{\phi}_{t-1}^i}(d\phi_t)p(\varphi_t|\mathcal{S}_t^i)} \\ &= \frac{w_{t-1}^{k_i}}{\lambda_t^{k_i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)g(y_t|x_t^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\phi}_{0:t-1}^i, \tilde{\varphi}_{t-1}^i, y_{1:t})}. \end{aligned} \quad (3.53)$$

Chapter 4

Numerical Experiments

In this chapter we illustrate with numerical experiments that SMC-based algoritmhs for sequential parameter learning (both the novel ones introduced here and the ones already established in the literature) can perform poorly if proper care is not taken in mitigating path degeneracy, but can provide adequate inference once this issue is addressed. Through simulation-based experiments, we also argue that these methods can even provide estimates that are compatible with "exact" and computationally-intensive methods, such as quadrature and particle Markov Chain Monte Carlo.

4.1 iid Model

We start with a very simple experiment proposed by Chopin et al. (2010). Here, we have a scalar-valued state space model defined by

$$X_t|X_{t-1} =^d X_t \sim \mathcal{N}(0,1) \tag{4.1}$$
$$Y_t|X_t =^d Y_t \sim \mathcal{N}(0,1) \tag{4.2}$$

with initial state $X_0 \sim \mathcal{N}(0,1)$. This model describes a pathological situation in which the states are completely independent of their own past and of the observations, and there are no static parameters. From (4.1-4.2), we have $f(x_t^i|x_{t-1}^i) = d\mathcal{N}(x_t^i|0,1)$ and $g(y_t|x_t^i) = d\mathcal{N}(y_t|0,1)$.

Consider in this setting the problem of recursively estimating the sample mean $\bar{X}_t := (X_0 + X_1 + \ldots + X_t)/(t+1)$, starting at $\bar{X}_0 := X_0$. Since $\bar{X}_t = (t\bar{X}_{t-1} + X_t)/(t+1)$ for $t \geq 1$, we can construct a particle approximation for $p(\bar{x}_t|y_{1:t})$ sequentially in time with the output of Algorithm 3.6 as

$$\hat{p}(\bar{x}_t|y_{1:t}) := \sum_{i=1}^{N} w_t^i \delta_{\bar{x}_t^i}(d\bar{x}_t),$$

where each particle $\bar{x}_t^i$ also satisfies $\bar{x}_t^i = (t\bar{x}_{t-1}^{k_i} + x_t^i)/(t+1)$ for each time $t \leq 1$, with $\bar{x}_0^i = x_0^i$ for all $i = 1, \ldots, N$. For the practical implementation, both optimal importance weights and optimal state proposal are available for this experiment as

$$\lambda_t^i \propto w_{t-1}^i p(y_t|x_{t-1}^i) = w_{t-1}^i p(y_t) \propto w_{t-1}^i \quad \text{and} \quad p(x_t^i|\tilde{x}_{t-1}^i, y_t) = p(x_t^i) = d\mathcal{N}(x_t^i|0,1),$$

since from (4.1-4.2) we have $X_t \perp\!\!\!\perp X_r \perp\!\!\!\perp Y_s$ for all t, r, s (here $U \perp\!\!\!\perp V$ denotes independence between the random variables U and V). The corresponding weight recursion

(3.14) is therefore

$$\pi_t^i \propto \frac{w_{t-1}^i}{w_{t-1}^i} \frac{d\mathcal{N}(x_t^i|0,1)d\mathcal{N}(y_t|0,1)}{d\mathcal{N}(x_t^i|0,1)} = d\mathcal{N}(y_t|0,1) \propto 1,$$

given that $d\mathcal{N}(y_t|0,1)$ does not vary with i. After normalization, the importance weights for the experiment are then uniform, i.e. $w_t^i = 1/N$ for all i and t.

Figure 4.1 contains kernel density estimates (solid blue lines) of $(\bar{x}_t^i, 1/N)_{i=1}^N$ from $M = 50$ independent runs of Algorithm 3.6 using $N = 5,000$ particles, for $t = 500$, $t = 2,500$ and $t = 5,000$, along with the true density (dashed black lines) $p(\bar{x}_t|y_{1:t}) = p(\bar{x}_t) = d\mathcal{N}(\bar{x}_t|0,(t+1)^{-1})$. The results on the left panel are based on the standard multinomial resampling scheme and those on the right panel are based on the branching algorithm of Crisan and Lyons (2002); see Section 3.2.2 for details.

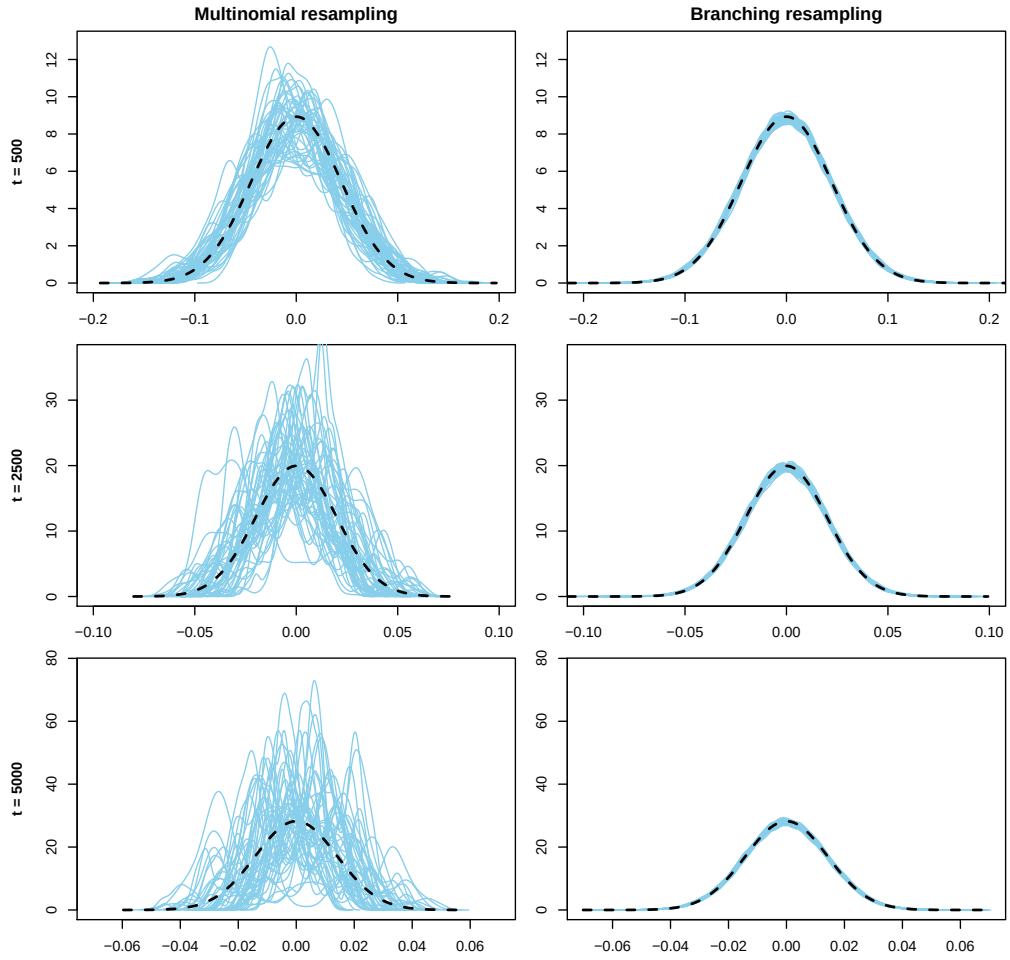

Figure 4.1: Independent SSM experiment (4.1-4.2): kernel density estimates (solid blue lines) of $(\bar{x}_t^i, 1/N)_{i=1}^N$ based on $M = 50$ independent runs of Algorithm 3.6 and the true density (dashed black lines) $d\mathcal{N}(\bar{x}_t|0,(t+1)^{-1})$ at $t = 500$, $t = 2,500$ and $t = 5,000$. The filters were run with $N = 5,000$ and multinomial (left column) and branching (right column) resampling.

Naturally, at first we might attribute the poor performance of the filter in approximating $p(\bar{x}_t|y_{1:t})$ entirely to weight degeneracy. However, note that from $\pi_t^i \propto d\mathcal{N}(y_t|0,1)$ we

have that $w_t^i = 1$, given that the constant of proportionality is $1/p(y_t|y_{1:t-1}) = 1/p(y_t) = 1/d\mathcal{N}(y_t|0,1)$. Since the weights are actually constant even before normalization, their variance (conditional or otherwise) is always zero, which by definition (Kong et al., 1994) implies that no weight degeneracy takes place in this experiment. In retrospect, this is not surprising given that the importance sampling procedure here is itself not sequential, given the serial and mutual independence between the states and observations.

Now, although the system defined here is not subject to weight degeneracy, it is definitely still subject to path degeneracy, since each particle $\bar{x}_{t-1}^i$ is resampled at the beginning of each step of the filter. Implicitly, this creates a dependence of $\bar{x}_t^i$ on $\bar{x}_{0:t-1}^i$ for each t, and the sample impoverishment inherent when successively resampling the latter ends up affecting the inference for the former.

A simple way to assess sample impoverishment is by looking at the *fertility factors* (FFs) (Baker, 1987) of the offspring produced by a resample algorithm over time. The FF is defined by

$$\begin{aligned} \mathrm{FF}_t \equiv \mathrm{FF}\big((\lambda_t^i)_{i=1}^N\big) &:= \frac{\#\{\text{distinct elements in } (k_t^i)_{i=1}^N\}}{\#\{(k_t^i)_{i=1}^N\}} \\ &= \frac{\#\{\text{distinct elements in } (k_t^i)_{i=1}^N\}}{N}, \end{aligned} \tag{4.3}$$

where $\#\{A\}$ denotes the cardinality of set A. Naturally, since the number of distinct elements is at least 1 and at most N, we have $1/N \leq \mathrm{FF}_t \leq N$.

For this experiment, the resampling weights are given by $\lambda_t^i = 1/N$ for all i, t. Under multinomial resampling, the probability that a fixed index j in $\{1, \ldots, N\}$ is not selected is then equal to

$$p(k_i \neq j, i = 1, \ldots, N) = (1 - \lambda_t^j)^N = \left(1 - \frac{1}{N}\right)^N,$$

from which we deduce that the probability of j indeed being selected is

$$p(k_i = j, i = 1, \ldots, N) = 1 - p(k_i \neq j, i = 1, \ldots, N) = 1 - \left(1 - \frac{1}{N}\right)^N \underset{N \to +\infty}{\longrightarrow} 1 - \frac{1}{e} \simeq 0.63.$$

Therefore, $\mathrm{FF}_t \approx 0.63$ for large N, meaning that multinomial resampling does not select approximately 37% of the values in the sample, leading over time to the poor performance observed in Figure 4.1.

On the other hand, recall from Section 3.2.2 – more specifically (3.11) – that the number of *offspring* ξ_t^j produced by particle j in the tree-based branching algorithm of Crisan and Lyons (2002) satisfies

$$\xi_t^j := \begin{cases} \lfloor N\lambda_t^j \rfloor & \text{with probability} \quad 1 - \{N\lambda_t^j\} \\ \lfloor N\lambda_t^j \rfloor + 1 & \text{with probability} \quad \{N\lambda_t^j\} \end{cases}$$

where $\lfloor \cdot \rfloor$ is the *floor* operator and $\{x\} := x - \lfloor x \rfloor$ is the non-integer part of x. Since here $N\lambda_t^j = N/N = 1$ implies that $\lfloor N\lambda_t^j \rfloor = 1$ and therefore $\{N\lambda_t^j\} = 0$, we have that ξ_t^j is degenerate at 1, i.e. each particle produces exactly one descendant in the branching algorithm. As a result, we have $(k_i)_{i=1}^N = (i)_{i=1}^N$, implying that $\mathrm{FF}_t = 1$ for all t. This perfect diversity entirely avoids any sample impoverishment and ensuing path degeneracy, leading to the sample $(\bar{x}_t^i, 1/N)_{i=1}^N$ being an actual exact draw from the true $p(\bar{x}_t^i)$, as seen in the right panel of Figure 4.1.

In closing, we point out that although the need for more efficient resampling schemes has long been recognized in the literature, systematic efforts to diagnose and mitigate the ensuing path degeneracy from sample impoverishment associated with low-efficiency resampling methods have not been made so far, at least to our knowledge. As pointed out in the discussion accompaying the original experiment in Chopin et al. (2010), this path degeneracy can significantly affect the inference based on any functional of the entire paths $(X_{0:t}, \theta_{0:t})$ that does not possess exponential forgetting properties (see Section 3.2.2), as illustrated in this example and in the ones that follow.

4.2 AR(1) + Noise Model

For the next experiment, consider the state space model defined by

$$X_t = \phi X_{t-1} + \sqrt{0.1} U_t, \qquad U_t \sim \mathcal{N}(0,1) \tag{4.4}$$

$$Y_t = X_t + \sigma V_t, \qquad V_t \sim \mathcal{N}(0,1) \tag{4.5}$$

with priors

$$X_0 \sim \mathcal{N}(0, 0.1), \qquad \phi \sim U[-1,1], \qquad \sigma^2 \sim IG(1/2, 1/2),$$

where $U[a,b]$ denotes a continuous uniform distribution on (a,b) and $IG(c,d)$ denotes an Inverse-Gamma distribution with shape c and scale d. We also assume that $X_0 \perp\!\!\!\perp U_t \perp\!\!\!\perp V_s$ for all t, s, and that $(U_t)_{t\geq 0}$ and $(V_t)_{t\geq 0}$ are serially independent.

The state space here is $\mathcal{X} = \mathbb{R} := (-\infty, +\infty)$ and the static parameter is $\theta = (\phi, \sigma^2)$, taking values in $\Theta = (-1,1) \times \mathbb{R}^+$, where $\mathbb{R}^+ := (0, +\infty)$. Since equations (4.4-4.5) describe a (Gaussian) autoregressive process $(X_t)_{t\geq 0}$ of order 1 observed via $(Y_t)_{t\geq 0}$ with (Gaussian) noise σV_t, the model is commonly referred to as the (Gaussian) *AR(1) + noise* model. Here $f(x_t^i | x_{t-1}^i, \theta_{t-1}^i) = d\mathcal{N}(x_t^i | \phi_{t-1}^i x_{t-1}^i, 0.1)$ and $g(y_t | x_t^i, \theta_{t-1}^i) = d\mathcal{N}(y_t | x_t^i, (\sigma^2)_{t-1}^i)$.

We reproduce here the specific configuration adopted by Kantas et al. (2015). By taking $\phi = 0.5$ and $\sigma^2 = 1$ as the true parameter values, we simulate a series of size $n = 5,000$ of model (4.4-4.5) and then perform $M = 50$ independent runs of the LW filter (Section 3.2.4.2) and the PL method (Section 3.2.4.6) using $N = 10,000$ particles. For the LW filter implementation, we have used $\delta = 0.99$. Since the model is linear and Gaussian, analytical expressions for both $p(y_t | x_{t-1})$ and $p(x_t | x_{t-1}, y_t)$ can be obtained by Proposition B.6.1 with $G_t = X_t$ and $F_t = \phi X_{t-1}$. Therefore, for the PL implementation, the optimal importance weights and state proposal distribution are

$$\lambda_t^i \propto w_{t-1}^i p(y_t | x_{t-1}^i, \theta_{t-1}^i) = w_{t-1}^i d\mathcal{N}(\phi_{t-1}^i x_{t-1}^i, (\sigma^2)_{t-1}^i + 0.1)$$

and

$$p(x_t | x_{t-1}^i, \theta_{t-1}^i, y_t) = d\mathcal{N}\left(x_t \middle| \frac{0.1 \cdot y_t + (\sigma^2)_{t-1}^i \phi_{t-1}^i x_{t-1}^i}{(\sigma^2)_{t-1}^i + 0.1}, \frac{(\sigma^2)_{t-1}^i \cdot 0.1}{(\sigma^2)_{t-1}^i + 0.1}\right),$$

and the posterior distributions for ϕ and σ^2 given the states and observations are

$$p(\phi | x_{0:t}, y_{1:t}) = d\mathcal{N}_{[-1,1]}(\phi | C^{-1} m_t, 0.1 \cdot C_t^{-1})$$

and

$$p(\sigma^2 | x_{0:t}, y_{1:t}) = dIG(\sigma^2 | a_t/2, b_t/2),$$

where $d\mathcal{N}_{[a,b]}(x|m,s^2)$ denotes the density (evaluated at x) of a Gaussian random variable truncated in the interval $[a,b]$ with mean m and variance s^2. The sufficient statistics for this example are $\mathcal{S}_t = (m_t, C_t, a_t, b_t)$, satisfying the following recursions for $t \geq 1$:

$$
\begin{aligned}
m_t &:= \sum_{j=2}^{t} X_j X_{j-1} = \sum_{j=2}^{t-1} X_j X_{j-1} + X_t X_{t-1} = m_{t-1} + X_t X_{t-1}, \\
C_t &:= \sum_{j=2}^{t} X_{j-1}^2 = \sum_{j=2}^{t-1} X_{j-1}^2 + X_{t-1}^2 = C_{t-1} + X_{t-1}^2, \\
a_t &:= 1 + t = 1 + (t-1) + 1 = a_{t-1} + 1, \\
b_t &:= 1 + \sum_{k=1}^{t} (Y_k - X_k)^2 = 1 + \sum_{k=1}^{t-1} (Y_k - X_k)^2 + (Y_t - X_t)^2 = b_{t-1} + (Y_t - X_t)^2,
\end{aligned}
$$

with $m_0 = C_0 = 0$ and $a_0 = b_0 = 1$. Note that since the parameter space Θ is restricted, we always work with transformed parameters $\check{\phi} = \operatorname{arctanh}(\phi)$ and $\check{\sigma}^2 = \log(\sigma^2)$ for the LW method; see Section E.1 for more details.

Figure 4.2 contains the corresponding estimated marginal posteriors $p(\phi|y_{1:n})$ and $p(\sigma^2|y_{1:n})$ for both LW and PL methods (based on multinomial resampling), along with their exact counterparts computed using quadrature based on the likelihood obtained from the Kalman filter (see section B.5) on a 100×100 equispaced grid. Although the results for σ^2 using PL (bottom right panel) are reasonable, the results for ϕ and LW's results in general are not. Overall, the variability across different runs is simply too large for the method's performance to be acceptable.

On the other hand, Figure 4.3 contains the same estimated marginal posteriors $p(\phi|y_{1:n})$ and $p(\sigma^2|y_{1:n})$, but now for the FALW and RPL methods. Here we also performed $M = 50$ independent runs for each filter with $N = 10,000$ particles, but resampled according to the tree-based branching algorithm. The results for FALW and RPL are much more consistent, with the estimated posteriors being not only much less variable across different runs, but also closer to the true quadrature-based posterior.

As argued previously, we can attribute most of the difference in performance from Figure 4.2 to Figure 4.3 to path degeneracy, since the only striking difference between the methods for both figures lie in the choice of resampling method (although the specific improvements that motivate us to propose FALW and RPL are also expected to mitigate path degeneracy even further). Although this might be expected in the LW filter (as a case of inefficient regularization, leading to the collapse of the static parameter paths), at first the PL method should not be susceptible to this phenomenom, given that the Gibbs sampling structure effectively marginalizes out past parameters and only relies on information from the current and at most one previous step via $\mathcal{S}(\mathcal{S}_{t-1}^{k_i}, x_t^i, y_t)$. As pointed out in e.g. Andrieu et al. (2005), Chopin et al. (2010) and Kantas et al. (2015), however (see also Section 3.2.2), the sufficient statistics are implicitly functions of the entire path $X_{0:t}$ and, without any exponential forgetting properties (since past statistics are only resampled and do not have an associated transition law), in time they also collapse.

4.3 Nonlinear Seasonal Model

We now consider an experiment proposed by Andrieu et al. (2010). The model, which we refer to simply as *Nonlinear Seasonal Model* (NLSM) was first introduced by Netto

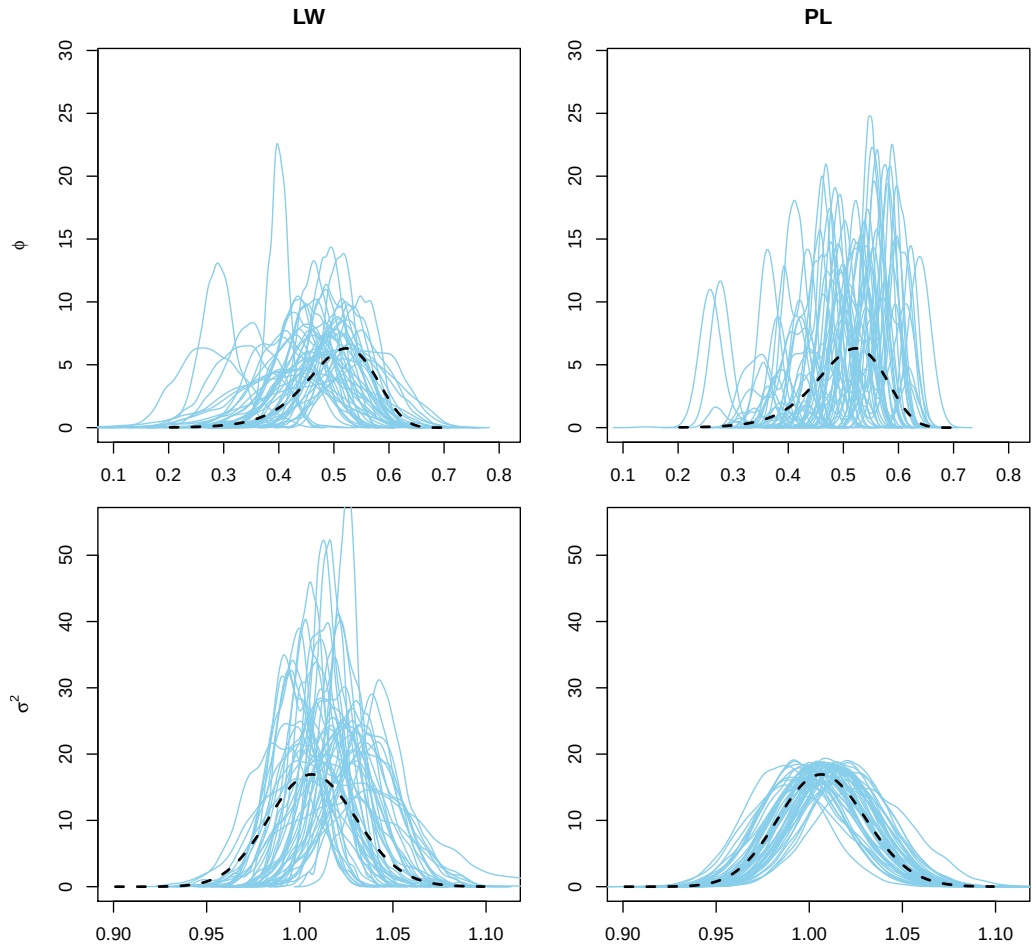

Figure 4.2: AR(1) + noise model (4.4-4.5): kernel density estimates (solid blue lines) of the posterior distributions for ϕ and σ^2 based on $M = 50$ independent runs of the LW filter (left column) and the PL method (right column). The filters were run with $N = 10,000$ and the true parameter values are $\phi = 0.5$ and $\sigma^2 = 1$. The black dashed lines are quadrature-based estimates using the likelihood from the Kalman filter.

et al. (1978) and is widely popular as a toy example in the particle filter literature (see e.g. Gordon et al., 1993; Kitagawa, 1987; Doucet et al., 2001; Cappé et al., 2005). It is defined by

$$X_t = \frac{X_{t-1}}{2} + 25\frac{X_{t-1}}{1+X_{t-1}^2} + 8\cos(1.2t) + \sigma_V V_t, \qquad V_t \sim \mathcal{N}(0,1) \tag{4.6}$$

$$Y_t = \frac{X_t^2}{20} + \sigma_W W_t, \qquad W_t \sim \mathcal{N}(0,1) \tag{4.7}$$

with priors

$$X_0 \sim \mathcal{N}(0,5), \qquad \sigma_V^2 \sim IG(1/2,1/2), \qquad \sigma^2 \sim IG(1/2,1/2),$$

where $(V_t)_{t\geq 0}$ and $(W_t)_{t\geq 0}$ are assumed to be mutually and serially independent, with $X_0 \perp\!\!\!\perp V_t$ and $X_0 \perp\!\!\!\perp W_t$ for all t.

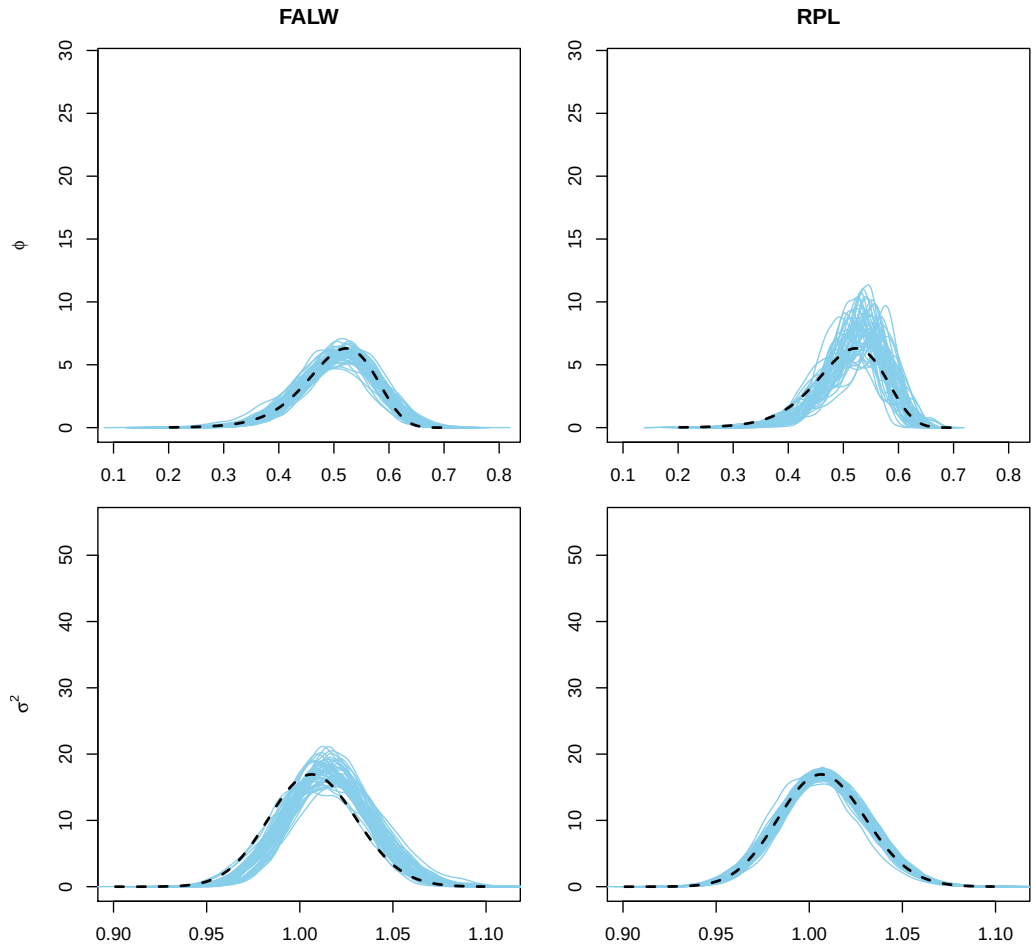

Figure 4.3: AR(1) + noise model (4.4-4.5): kernel density estimates (solid blue lines) of the posterior distributions for ϕ and σ^2 based on $M = 50$ independent runs of the FALW filter (left column) and the RPL method (right column). The filters were run with $N = 10,000$ and the true parameter values are $\phi = 0.5$ and $\sigma^2 = 1$. The black dashed lines are quadrature-based estimates using the likelihood from the Kalman filter.

The state space for the NLSM model is once again $\mathcal{X} = \mathbb{R}$, the static parameter vector is $\theta = (\sigma_V^2, \sigma_W^2)$ and the parameter space is $\Theta = \mathbb{R}^+ \times \mathbb{R}^+$. Here

$$f(x_t^i | x_{t-1}^i, \theta_{t-1}^i) = d\mathcal{N}\left(\frac{x_{t-1}^i}{2} + 25\frac{x_{t-1}^i}{1+(x_{t-1}^i)^2} + 8\cos(1.2t), (\sigma_V^2)_{t-1}^i\right)$$

and

$$g(y_t | x_t^i, \theta_{t-1}^i) = d\mathcal{N}\big(y_t | x_t^i, (\sigma_W^2)_{t-1}^i\big).$$

For this experiment, we take $\sigma_V^2 = 10$ and $\sigma_W^2 = 1$ as the true parameter values, simulate a series $y_{1:n}$ with $n = 500$ and then perform $M = 50$ independent runs of the smooth jittering method (Section 3.2.4.3, here denoted as FS) and the RPL method (Section 3.2.4.9) using $N = 50,000$ particles. For comparison, we also fit a pMCMC algorithm with $N = 5,000$ particles, branching resampling, Markov chain size $M = 50,000$, a burn-in of $B = 10,000$, initial values $(\sigma_V^2)_0 = (\sigma_W^2)_0 = 100$ and a random walk

proposal

$$q(\theta_j|\theta_{j-1}) = d\mathcal{N}\left(\begin{bmatrix}(\sigma_V^2)_j\\(\sigma_W^2)_j\end{bmatrix}\middle|\begin{bmatrix}(\sigma_V^2)_{j-1}\\(\sigma_W^2)_{j-1}\end{bmatrix}, \begin{bmatrix}0.15 & 0\\0 & 0.08\end{bmatrix}\right)$$

for $j = 1, \ldots, M + B$, as in Andrieu et al. (2010).

Since full adaptation in model (4.6-4.7) is not possible, the original PL algorithm is not feasible here. Moreover, given the the high signal-to-noise ratio $\sigma_V^2/\sigma_W^2 = 10$ and the fact that by construction the observations in this model are not very informative (since the state X_t is only observed via its square), lookahead APF strategies are not efficient, leading the usual LW filter to perform poorly. Therefore, design choices for RPL are the same as for the FS method: SIR intermediate weights $\lambda_t^i = w_{t-1}^i$ and blind state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = f(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)$.

The posterior distributions for σ_V^2 and σ_W^2 for the Gibbs sampling steps of the RPL method are given by

$$\sigma_V^2|(X_{0:t}, Y_{1:t}) \sim IG(a_t/2, b_t/2) \qquad \text{and} \qquad \sigma_W^2|(X_{0:t}, Y_{1:t}) \sim IG(b_t/2, c_t/2),$$

where the sufficient statistics $\mathcal{S}_t = (a_t, b_t, c_t, d_t)$ satisfy, for $t \geq 1$,

$$\begin{aligned}
a_t &:= 1 + t = 1 + (t-1) + 1 = a_{t-1} + 1,\\
b_t &:= 1 + \sum_{k=1}^{t}(X_k - F_k)^2 = 1 + \sum_{k=1}^{t-1}(Y_k - F_k)^2 + (Y_t - F_t)^2 = b_{t-1} + (Y_t - F_t)^2,\\
c_t &:= 1 + t = 1 + (t-1) + 1 = c_{t-1} + 1,\\
d_t &:= 1 + \sum_{k=1}^{t}\left(Y_k - \frac{X_k^2}{20}\right)^2\\
&\quad = 1 + \sum_{k=1}^{t-1}\left(Y_k - \frac{X_k^2}{20}\right)^2 + \left(Y_t - \frac{X_t^2}{20}\right)^2 = d_{t-1} + \left(Y_t - \frac{X_t^2}{20}\right)^2,
\end{aligned}$$

with $a_0 = b_0 = c_0 = d_0 = 1$ and $F_t := X_{t-1}/2 + 25X_{t-1}/(1 + X_{t-1}^2) + 8\cos(1.2t)$. Since the parameter space Θ in this example is also restricted, we have to work with log-variances $\check{\sigma}_V^2 = \log(\sigma_V^2)$ and $\check{\sigma}_W^2 = \log(\sigma_W^2)$ for the RPL and FS methods (see Section E.1), as well as for pMCMC. For the latter, the transformation implies that the usual acceptance probability of the pMCMC algorithm must be multiplied at each step j by the Jacobian

$$\left|\frac{(\sigma_V^2)_j \cdot (\sigma_W^2)_j}{(\sigma_V^2)_{j-1} \cdot (\sigma_W^2)_{j-1}}\right|;$$

see Section E.2 for further details. All the methods are implemented using branching resampling.

Figure 4.4 contains the estimated marginal posteriors for both σ_V^2 and σ_W^2 obtained via smooth jittering (solid blue lines, left column), RPL (solid blue lines, right column) and pMCMC (dashed lines, both columns). Taking the pMCMC estimates as the main reference here, we can see that overall both smooth jittering and RPL methods perform reasonably well, with the RPL estimates apparently having lower variance across runs than those from FS. It is worth pointing out that the difference in complexity between pMCMC and the SMC-based methods is striking: for an Intel® Core i7 CPU 860 running at 2.80 GHz, pMCMC took 57,251 seconds (about 15 hours and 54 minutes) to complete, while an average FS run took 20.15 seconds and an RPL one took 27.59 seconds.

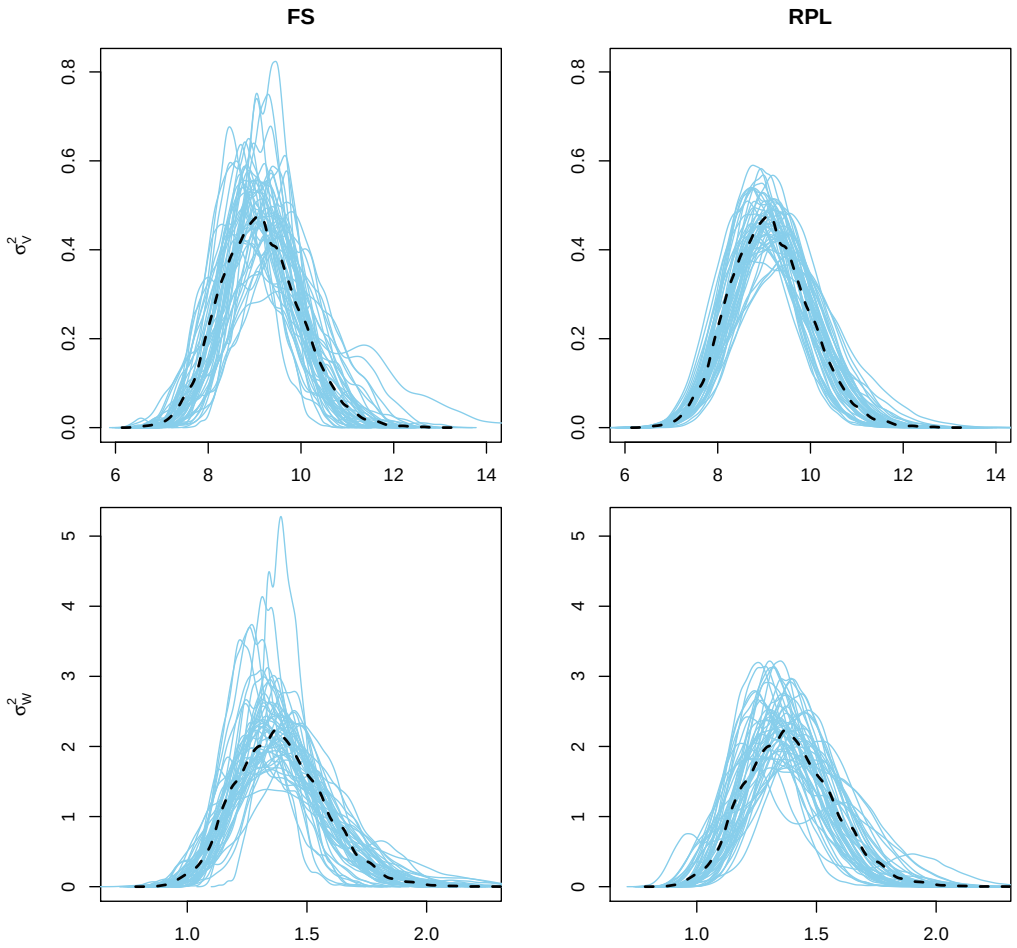

Figure 4.4: NLSM model (4.6-4.7): kernel density estimates (solid blue lines) of the posterior distributions for σ_V^2 and σ_W^2 based on $M = 50$ independent runs of the smooth jittering method (left column) and the RPL method (right column). The filters were run with $N = 50,000$ and the true parameter values are $\sigma_V^2 = 10$ and $\sigma_W^2 = 1$. The black dashed lines are pMCMC estimates based on a chain of size $M = 50,000$, burn-in of $B = 10,000$ and $N = 5,000$ particles.

4.4 Stochastic Volatility Model

For our last experiment, consider the discrete-time scalar-valued Stochastic Volatility (SV) model (Taylor, 1982), defined by

$$X_t = \phi X_{t-1} + \tau U_t, \qquad U_t \sim \mathcal{N}(0,1), \tag{4.8}$$

$$Y_t = \sigma \exp(X_t/2) V_t, \qquad V_t \sim \mathcal{N}(0,1), \tag{4.9}$$

with priors

$$X_0 \sim \mathcal{N}(0, \tau^2), \qquad \phi \sim \mathcal{N}_{[-1,1]}(0.95, 0.05^2), \qquad \tau^2 \sim G(2, 0.01), \qquad \sigma^2 \sim \mathcal{LN}(0,1),$$

where $\mathcal{N}_{[a,b]}(m, s^2)$ denotes a normal distribution with mean m and variance s^2 truncated in the interval $[a, b]$, $G(c, d)$ denotes a Gamma distribution with shape c and rate d, i.e.

with expectation $c \cdot d$ and $\mathcal{LN}(m_L, s_L^2)$ denotes a Log-Normal distribution with log-mean m_L and log-variance s_L^2. Here, once again $(U_t)_{t\geq 0}$ and $(V_t)_{t\geq 0}$ are assumed to be serially and mutually independent, and also independent of X_0.

The stochastic volatility model is a widely popular model in economics and finance, having many different incarnations (Shephard, 2005). A particularly interesting motivation for this model is as the discretely-observed version (Fearnhead et al., 2008) of the underlying diffusion

$$\begin{aligned} dX(t) &= \phi X(t)dt + \tau dB_X(t), \\ d\log P(t) &= \sigma \exp\{X(t)/2\} dB_Y(t), \end{aligned}$$

where $Y(t) = d\log P(t)$ is the increment of log-prices (referred to as *log-returns*) and $dB_X(t)$ and $dB_Y(t)$ are two independent standard Brownian motions (Doob, 1990).

The state space for the SV model is $\mathcal{X} = \mathbb{R}$, the static parameters are $\theta = (\phi, \tau^2\sigma^2)$ and the parameter space is $\Theta = (-1, 1) \times \mathbb{R}^+ \times \mathbb{R}^+$. Here we have $f(x_t^i | x_{t-1}^i, \theta_{t-1}^i) = d\mathcal{N}(x_t^i | \phi_{t-1}^i x_{t-1}^i, (\tau^2)_{t-1}^i)$ and $g(y_t | x_t^i, \theta_{t-1}^i) = d\mathcal{N}(y_t | 0, (\sigma^2)_{t-1}^i \exp(x_t^i/2))$.

For this experiment, we simulate a series of size $n = 1,000$ with true parameter values $\phi = 0.97$, $\tau^2 = 0.15^2$ and $\sigma^2 = e^{-0.23}$. These values are based on the estimates of Dahlin and Schön (2019) using a real time series of stock index prices. We perform $M = 50$ independent runs of the LW filter (Section 3.2.4.2) and the smooth jittering method (Section 3.2.4.3, referred to here as FS) using $N = 50,000$ particles and, for comparison, we fit a pMCMC algorithm similar to that of Dahlin and Schön (2019), but with $N = 500$ particles, a chain of size $M = 50,000$ and a burn-in of $B = 50,000$.

For the burn-in period of the pMCMC, we adopt the *Robust Adaptive Metropolis* algorithm of Vihola (2012) (see also Section A.4) with step size $\eta_j = j^{-2/3}$ (Vihola, 2012) and desired acceptance rate $\alpha_* = 0.234$ (Sherlock et al., 2015), starting with an independent covariance matrix with diagonals equal to 0.1, 0.01 and 0.05 (Dahlin and Schön, 2019). For the actual retained chain, we then use the resulting covariance matrix Σ_B of the burn-in period in a random walk metropolis of the form

$$\theta_j \sim \mathcal{N}(\theta_{j-1}, \Sigma_B),$$

for $j = B+1, \ldots, B+M$. All the methods were implemented using branching resampling, and due to parameter space restrictions we always work with $\check{\phi} = \arctan(\phi)$, $\check{\tau}^2 = \log(\tau^2)$ and $\check{\sigma}^2 = \log(\sigma^2)$; see Sections E.1 and E.2. The Jacobian of this transformation is

$$\left| \frac{(1-\phi_j^2) \cdot (\tau^2)_j \cdot (\sigma^2)_j}{(1-\phi_{j-1}^2) \cdot (\tau^2)_{j-1} \cdot (\sigma^2)_{j-1}} \right|.$$

Figure 4.5 contains the estimated marginal posteriors for ϕ, τ^2 and σ^2 obtained from the LW filter (solid blue lines, left column), FS (solid blue lines, right column) and pMCMC (dashed lines, both columns). Taking the pMCMC estimates again as the main reference, both the LW filter and the smooth jittering method perform reasonably well, with low variances across runs and similar posteriors to the one produced by pMCMC. Overall, the LW posteriors seem to have lower variance across runs but those from the FS method seem to agree more with the output from pMCMC. As far as complexity is concerned, for an Intel® Core i7 CPU 860 running at 2.80 GHz, pMCMC took 19,979 seconds (about 5 hours and 33 minutes) to complete, while in average LW took 45.64 seconds and RPL took 34.97 seconds.

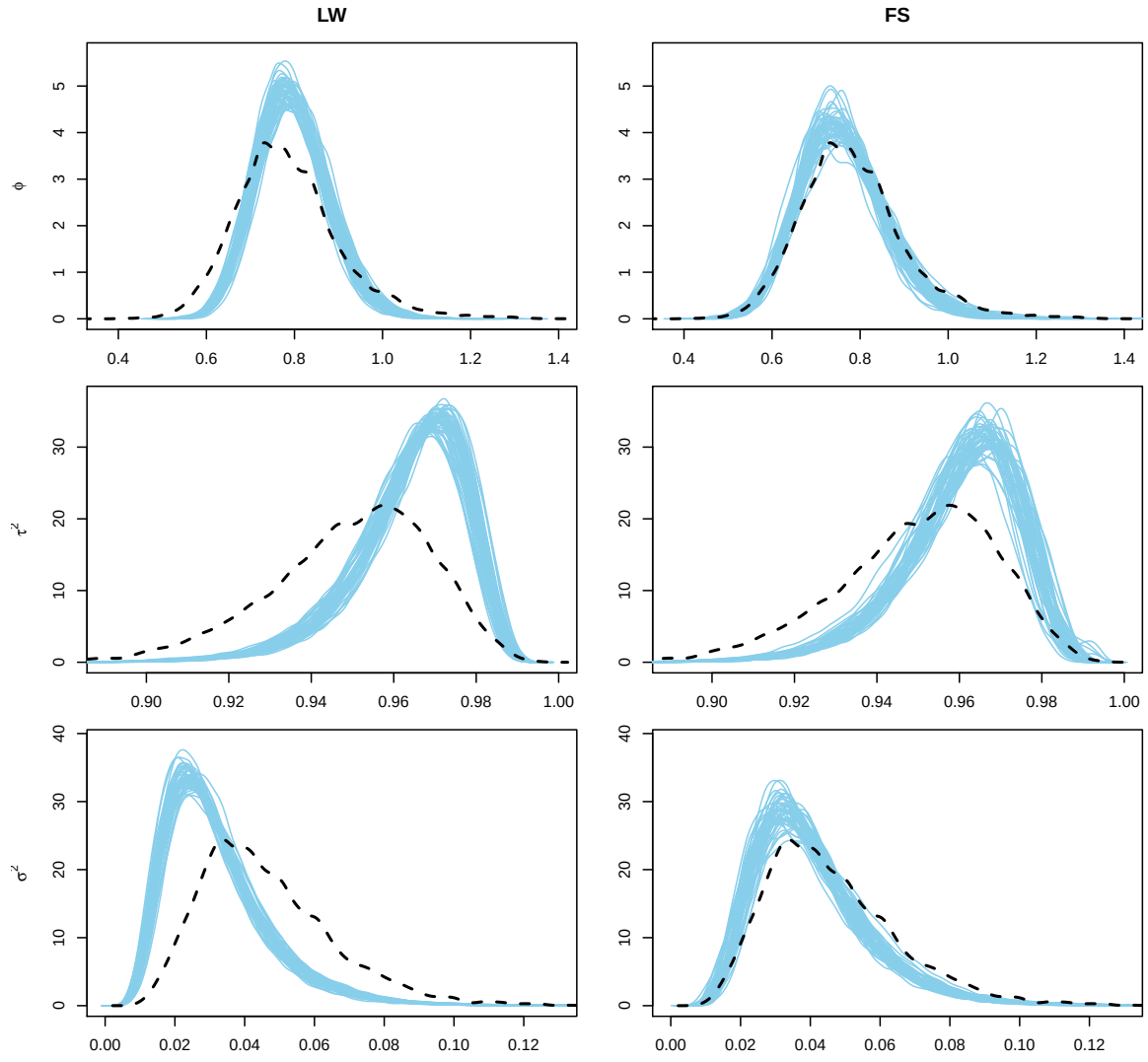

Figure 4.5: SV model (4.8-4.9): kernel density estimates (solid blue lines) of the posterior distributions for ϕ, τ^2 and σ^2 based on $M = 50$ independent runs of the smooth jittering method (left column) and the FS method (right column). The filters were run with $N = 50,000$ and the true parameter values are $\phi = 0.97$, $\tau^2 = 0.15^2 \simeq 0.02$ and $\sigma^2 = e^{-0.23} \simeq 0.80$. The black dashed lines are pMCMC estimates based on a chain of size $M = 50,000$, burn-in of $B = 50,000$ and $N = 500$ particles.

Chapter 5

Conclusions

In this thesis we introduced a novel framework for sequential parameter learning in Hidden Markov models, shown to be capable of accommodating several other algorithms found in the literature as special cases. A key feature of this framework is how we reinterpret regularization of past states and static parameters through the use of auxiliary variables, essentially generalizing the approach of Pitt and Shephard (1999). As an example of the flexibility allowed by our framework, we also developed three novel algorithms, including an improved and fully-adapted version of the celebrated Liu and West filter. This general framework is the main contribution of our work.

The other contribution we make to the literature is to further illustrate that the poor performance of sequential parameter learning algorithms previously observed in certain settings can mostly be attributed to the inherent path degeneracy in these methods, building on the work of da Silva (2016). By exploring procedures that actively aim to mitigate path degeneracy (such as more efficient resampling schemes and asymptotically unbiased regularization methods), we consider a series of simulation-based numerical experiments in which we attempt to show that once path degeneracy is properly assessed, even classical parameter learning algorithms such as the "vanilla" Liu and West filter can provide estimates that are compatible with state-of-the-art and more numerically intensive methods such as particle Markov Chain Monte Carlo.

As a main avenue for future research, hopefully the formalism developed for the framework proposed here will allow for a unified exploration of the theoretical properties of general sequential parameter learning algorithms, while also allowing for a better understanding of the common features defining these methods. Such a development would be crucial, since due to their apparent differences these techniques have only been traditionally studied separate from one another.

A further promising avenue for future work would be to use the sequential parameter learning techniques developed here in order to design more efficient proposals for particle Markov Chain Monte Carlo methods, as done by e.g. Wood et al. (2014) and Fearnhead and Meligkotsidou (2016). In principle this could yield powerful algorithms indeed, since sequential parameter learning methods require no tuning and as shown in this thesis they are good at exploring the global features of the posterior distributions of static parameters.

As far as improving the methods explored in this thesis themselves, the theory of Hamiltonian Monte Carlo (more specifically Langevin-type dynamics) could also be fruitfully applied within our framework in order to allow for even more efficient exploration of parameter spaces. This would parallel the work of e.g. Dahlin et al. (2015) and

Nemeth et al. (2016) on particle Markov Chain Monte Carlo methods but on a sequential inference context. Other possible improvements include e.g. the adaptive importance sampling method of Cornebise et al. (2008) and the hybrid particle filter algorithm of Petetin and Desbouvries (2013), which in principle could be readily adapted for performing parameter inference.

Since the literature on parameter learning methods (both online and offline) is quite extensive, we naturally omitted even popular methods from our presentation, such as the Practical Filtering technique of Polson et al. (2008), the SMC^2 framework of Chopin et al. (2013) and the Marginalized Resample-Move algorithm of Fulop and Li (2013). Although none of these methods fit within our framework in their full generality, all of them can certainly benefit from either the regularization or minimal variance resampling techniques (or even both) advocated in our work.

In closing, note that here we avoided certain heuristics designed for mitigating path degeneracy that although popular and insightful still do not have an established theoretical background. Examples of these include tempering of importance weights (Liu et al., 2001), resampling at random times according to arbitrary cutoffs (Doucet et al., 2000) and annealing of importance distributions according to arbitrary schedules (Peters et al., 2010). Exploring the theoretical properties and providing general practical guidelines for these heuristics could also be important goals for future research, since they have already been shown (see e.g. da Silva, 2016) to be quite effective in certain settings.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*. Wiley-Interscience.

Andrieu, C., De Freitas, N., and Doucet, A. (1999). Sequential mcmc for bayesian model selection. In *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics. SPW-HOS'99*, pages 130–134. IEEE.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C., Doucet, A., and Tadic, 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. IEEE.

Andrieu, C., Roberts, G. O., et al. (2009). The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725.

Asmussen, S. and Glynn, P. W. (2007). *Stochastic simulation: algorithms and analysis*, volume 57. Springer Science & Business Media.

Bain, A. and Crisan, D. (2009). *Fundamentals of Stochastic Filtering, 2009*. Springer.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, volume 206, pages 14–21.

Bartle, R. (1976). *The elements of real analysis*. Wiley, New York.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

BøLviken, E., Acklam, P. J., Christophersen, N., and StøRdal, J.-M. (2001). Monte carlo filters for non-linear state estimation. *Automatica*, 37(2):177–183.

Cappé, O., Moulines, E., and Ryden, T. (2005). *Inference in Hidden Markov ModelsR (Springer Series in Statistics)*. Springer.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7.

Carter, C. K. and Kohn, R. (1994). On gibbs sampling for state space models. *Biometrika*, 81(3):541–553.

Carvalho, C. M., Johannes, M. S., Lopes, H. F., Polson, N. G., et al. (2010). Particle learning and smoothing. *Statistical Science*, 25(1):88–106.

Carvalho, C. M. and Lopes, H. F. (2007). Simulation-based sequential analysis of markov switching stochastic volatility models. *Computational Statistics & Data Analysis*, 51(9):4526–4542.

Chen, H., Petralia, F., and Lopes, H. F. (2010). Sequential monte carlo estimation of dsge models. Technical report, The University of Chicago Booth School of Business.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539–552.

Chopin, N. et al. (2004). Central limit theorem for sequential monte carlo methods and its application to bayesian inference. *The Annals of Statistics*, 32(6):2385–2411.

Chopin, N., Iacobucci, A., Marin, J.-M., Mengersen, K., Robert, C. P., Ryder, R., and Schäfer, C. (2010). On particle learning. *arXiv preprint arXiv:1006.0554*.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). Smc^2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426.

Cornebise, J., Moulines, É., and Olsson, J. (2008). Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, 18(4):461–480.

Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on signal processing*, 50(3):736–746.

Crisan, D. and Lyons, T. (2002). Minimal entropy approximations and optimal algorithms. *Monte Carlo methods and applications*, 8(4):343–356.

da Silva, F. C. A. (2016). *Métodos Sequencias de Monte Carlo Bayesianos: Aspectos Computacionais, Inferenciais e Aplicações*. PhD thesis, Universidade Federal de Minas Gerais.

Dahlin, J., Lindsten, F., and Schön, T. B. (2015). Particle metropolis–hastings using gradient and hessian information. *Statistics and computing*, 25(1):81–92.

Dahlin, J. and Schön, T. B. (2019). Getting started with particle metropolis-hastings for inference in nonlinear dynamical models. *Journal of Statistical Software*, 88(CN2):1–41.

Del Moral, P. (2004). Feynman-kac formulae. In *Feynman-Kac Formulae*, pages 47–93. Springer.

Doob, J. L. (1990). *Stochastic Processes*. Wiley-Interscience.

Douc, R., Garivier, A., Moulines, E., Olsson, J., et al. (2011). Sequential monte carlo smoothing for general state space hidden markov models. *The Annals of Applied Probability*, 21(6):2109–2145.

Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.

Doucet, A., Pitt, M. K., Deligiannidis, G., and Kohn, R. (2015). Efficient implementation of markov chain monte carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313.

Dukic, V., Lopes, H. F., and Polson, N. G. (2012). Tracking epidemics with state-space seir and google flu trends. *Unpublished manuscript*.

Durbin, J. and Koopman, S. J. (2012). *Time series analysis by state space methods*, volume 38. OUP Oxford.

Elliott, R. J., Aggoun, L., and Moore, J. B. (2008). *Hidden Markov models: estimation and control*, volume 29. Springer Science & Business Media.

Fearnhead, P. (2002). Markov chain monte carlo, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics*, 11(4):848–862.

Fearnhead, P. and Meligkotsidou, L. (2016). Augmentation schemes for particle mcmc. *Statistics and Computing*, 26(6):1293–1306.

Fearnhead, P., Papaspiliopoulos, O., and Roberts, G. O. (2008). Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):755–777.

Flury, T. and Shephard, N. (2009). Learning and filtering via simulation: smoothly jittered particle filters. Economics Series Working Papers 469, University of Oxford, Department of Economics.

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of time series analysis*, 15(2):183–202.

Fulop, A. and Li, J. (2013). Efficient learning via simulation: A marginalized resample-move approach. *Journal of Econometrics*, 176(2):146–161.

Gamerman, D. and Lopes, H. F. (2006). *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339.

Ghaeminia, M. H., Shabani, A. H., and Shokouhi, S. B. (2010). Adaptive motion model for human tracking using particle filter. In *2010 20th International Conference on Pattern Recognition*, pages 2073–2076. IEEE.

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.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168.

Golightly, A. and Wilkinson, D. J. (2006). Bayesian sequential inference for nonlinear multivariate diffusions. *Statistics and Computing*, 16(4):323–338.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113.

Grewal, M. S. and Andrews, A. P. (2010). Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control systems*, 30(3):69–78.

Jacquier, E., Polson, N., and Sokolov, V. (2016). Sequential bayesian learning for merton's jump model with stochastic volatility. *arXiv preprint arXiv:1610.09750*.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351.

Kitagawa, G. (1987). Non-gaussian state—space modeling of nonstationary time series. *Journal of the American statistical association*, 82(400):1032–1041.

Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association*, pages 1203–1215.

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.

Künsch, H. R. (2005). Recursive monte carlo filters: algorithms and theoretical analysis. *The Annals of Statistics*, 33(5):1983–2021.

Li, T., Bolic, M., and Djuric, P. M. (2015). Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86.

Liang, C. and Piché, R. (2010). Mobile tracking and parameter learning in unknown non-line-of-sight conditions. In *2010 13th International Conference on Information Fusion*, pages 1–6. IEEE.

Lin, J. and Ludkovski, M. (2014). Sequential bayesian inference in hidden markov stochastic kinetic models with application to detection and response to seasonal epidemics. *Statistics and Computing*, 24(6):1047–1062.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer.

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer Science & Business Media.

Liu, J. S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044.

Liu, J. S., Chen, R., and Logvinenko, T. (2001). A theoretical framework for sequential importance sampling with resampling. In *Sequential Monte Carlo methods in practice*, pages 225–246. Springer.

Liu, Y.-Y., Li, S., Li, F., Song, L., and Rehg, J. M. (2015). Efficient learning of continuous-time hidden markov models for disease progression. In *Advances in neural information processing systems*, pages 3600–3608.

Lopes, H. F. and Tsay, R. S. (2011). Particle filters and bayesian inference in financial econometrics. *Journal of Forecasting*, 30(1):168–209.

Lu, T.-T. and Shiou, S.-H. (2002). Inverses of 2× 2 block matrices. *Computers & Mathematics with Applications*, 43(1-2):119–129.

Montgomery, D. (2018). *Design and analysis of experiments*. Wiley, Hoboken, NJ.

Musso, C., Oudjane, N., and Le Gland, F. (2001). Improving regularised particle filters. In *Sequential Monte Carlo methods in practice*, pages 247–271. Springer.

Nemeth, C., Fearnhead, P., and Mihaylova, L. (2013). Sequential monte carlo methods for state and parameter estimation in abruptly changing environments. *IEEE Transactions on Signal Processing*, 62(5):1245–1255.

Nemeth, C., Fearnhead, P., and Mihaylova, L. (2016). Particle approximations of the score and observed information matrix for parameter estimation in state–space models with linear computational cost. *Journal of Computational and Graphical Statistics*, 25(4):1138–1157.

Netto, M., Gimeno, L., and Mendes, M. (1978). On the optimal and suboptimal nonlinear filtering problem for discrete-time systems. *IEEE Transactions on Automatic Control*, 23(6):1062–1067.

Peters, G. W., Hosack, G. R., and Hayes, K. R. (2010). Ecological non-linear state space model selection via adaptive particle markov chain monte carlo (adpmcmc). *arXiv preprint arXiv:1005.2238*.

Petetin, Y. and Desbouvries, F. (2013). Optimal sir algorithm vs. fully adapted auxiliary particle filter: a non asymptotic analysis. *Statistics and computing*, 23(6):759–775.

Petris, G. (2009). *Dynamic linear models with R*. Springer-Verlag.

Pitt, M. K., dos Santos Silva, R., Giordani, P., and Kohn, R. (2012). On some properties of markov chain monte carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599.

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.

Prado, R. and Lopes, H. F. (2013). Sequential parameter learning and filtering in structured autoregressive state-space models. *Statistics and Computing*, 23(1):43–57.

Randal, D., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on image and signal processing and analysis*, pages 64–69.

Reichenberg, R. (2018). Dynamic bayesian networks in educational measurement: Reviewing and advancing the state of the field. *Applied Measurement in Education*, 31(4):335–350.

Rios, M. P. and Lopes, H. F. (2010). Evaluation and analysis of sequential parameter learning methods in markov switching stochastic volatility models. Technical report, The University of Chicago Booth School of Business.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer New York.

Rodeiro, C. L. V. and Lawson, A. B. (2006). Online updating of space-time disease surveillance models via particle filters. *Statistical methods in medical research*, 15(5):423–444.

Rudin, W. (1976). *Principles of mathematical analysis*. International series in pure and applied mathematics. McGraw-Hill, 3d ed edition.

Schön, T. B., Wills, A., and Ninness, B. (2011). System identification of nonlinear state-space models. *Automatica*, 47(1):39–49.

Shao, J. (2003). *Mathematical statistics*. Springer, New York.

Shephard, N. (1994). Partial non-gaussian state space. *Biometrika*, 81(1):115–131.

Shephard, N. (2005). *Stochastic volatility: selected readings*. Oxford University Press on Demand.

Sherlock, C., Fearnhead, P., Roberts, G. O., et al. (2010). The random walk metropolis: linking theory and practice through a case study. *Statistical Science*, 25(2):172–190.

Sherlock, C., Thiery, A. H., Roberts, G. O., Rosenthal, J. S., et al. (2015). On the efficiency of pseudo-marginal random walk metropolis algorithms. *The Annals of Statistics*, 43(1):238–275.

Shiryaev, A. N. (1995). *Probability (Graduate Texts in Mathematics) (v. 95)*. Springer.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. CRC press.

Simon, D. (2013). *Evolutionary optimization algorithms*. Wiley-Blackwell, Chichester.

Smith, A. F. and Gelfand, A. E. (1992). Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, 46(2):84–88.

Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289.

Stravropoulos, P. and Titterington, M. (2001). Improved particle filters and smothing. *Sequential Monte Carlo Methods in Practice*, pages 465–477.

Taylor, S. J. (1982). Financial returns modelled by the product of two stochastic processes-a study of the daily sugar prices 1961-75. *Time series analysis: theory and practice*, 1:203–226.

Vercauteren, T., Toledo, A. L., and Wang, X. (2005). Online bayesian estimation of hidden markov models with unknown transition matrix and applications to ieee 802.11 networks. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 4, pages iv–13. IEEE.

Vihola, M. (2012). Robust adaptive metropolis algorithm with coerced acceptance rate. *Statistics and Computing*, 22(5):997–1008.

Virbickaitė, A., Lopes, H. F., Concepción Ausín, M., and Galeano, P. (2019). Particle learning for bayesian semi-parametric stochastic volatility model. *Econometric Reviews*, 38(9):1007–1023.

Vose, M. D. (1991). A linear algorithm for generating random numbers with a given distribution. *IEEE Transactions on software engineering*, 17(9):972–975.

Wang, W.-p., Liao, S., and Xing, T.-w. (2009). Particle filter for state and parameter estimation in passive ranging. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 3, pages 257–261. IEEE.

Warty, S. P., Lopes, H. F., and Polson, N. G. (2018). Sequential bayesian learning for stochastic volatility with variance-gamma jumps in returns. *Applied Stochastic Models in Business and Industry*, 34(4):460–479.

West, M. (1993a). Approximating posterior distributions by mixtures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(2):409–422.

West, M. (1993b). Mixture models, monte carlo, bayesian updating, and dynamic models. *Computing Science and Statistics*, pages 325–325.

West, M. and Harrison, J. (1997). *Bayesian forecasting and dynamic models*. Springer series in statistics. Springer, New York, 2nd ed edition.

Wood, F., Meent, J. W., and Mansinghka, V. (2014). A new approach to probabilistic programming inference. In *Artificial Intelligence and Statistics*, pages 1024–1032.

Yümlü, M. S., Gürgen, F. S., Cemgil, A. T., and Okay, N. (2015). Bayesian changepoint and time-varying parameter learning in regime switching volatility models. *Digital Signal Processing*, 40:198–212.

Appendix A

Monte Carlo Methods

In this appendix we briefly introduce some notions about Monte Carlo methods which are necessary for a complete understanding of the core material in this thesis. There are several comprehensive references available on the general theory of Monte Carlo methods, such as Asmussen and Glynn (2007), Liu (2008) and Robert and Casella (2004). In particular, the material here is mostly inspired by Liu (2008).

A.1 Perfect Sampling

Let X be a random variable mapping the probability space $(\Omega, \mathcal{F}, \mathbb{P}_\Omega)$ onto $(\mathcal{X}, \mathcal{B}, \mathbb{P})$. If we can generate N independent random copies of X, each denoted by X^i, a natural estimator $\hat{\mathbb{P}}$ for the probability that $X \in A$ for $A \in \mathcal{F}$ is the proportion of X^i's in A, that is,

$$\hat{\mathbb{P}}(X \in A) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^i}(A), \tag{A.1}$$

where $x^i \in \mathcal{X}$ is the realized value of each X^i, i.e. $X^i(\omega) = x^i$ for a particular $\omega \in \Omega$ and $\delta_a(A)$ denotes point mass/Dirac measure at the point a, i.e. $\delta_a(A) = 1$ if $a \in A$ and $\delta_a(A) = 0$ if $a \notin A$. In particular, for the event $[X \leq x] := \{\omega : X(\omega) \leq x\}$, $x \in \mathbb{R}^{d_x}$, we have

$$\hat{F}(x) := \hat{\mathbb{P}}(X \leq x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^i}(X \leq x), \tag{A.2}$$

which is sometimes known as the *empirical (cumulative) distribution function* of the sample $(x^i)_{i=1}^N$. Here, d_x denotes the dimension of $\mathcal{X}$, i.e. $d_x := \dim(\mathcal{X})$.

Assume now that $\mathbb{P}$ is dominated by a suitable σ-finite measure dx and let $p := d\mathbb{P}/dx$ be its respective probability density. Then, by (A.2) we have

$$\begin{aligned}
\hat{p}(x) &:= \frac{d\hat{\mathbb{P}}(X \leq x)}{dx} \\
&= \frac{d}{dx}\left[\frac{1}{N} \sum_{i=1}^{N} \delta_{x^i}(X \leq x)\right] \\
&= \frac{1}{N} \sum_{i=1}^{N} \frac{d}{dx} \delta_{x^i}(X \leq x)
\end{aligned}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \delta_{x^i}(dx), \tag{A.3}$$

where $\delta_{x^i}(dx) := d\delta_{x^i}(X \leq x)/dx$, i.e. the point mass of an increment in dx at the point x^i. We call $\hat{p}$ the *Monte Carlo (MC) estimator* of p.

Since $\hat{p}$ is essentially the mean of a random sample, it should have all the good properties associated with this class of estimator. For example, $\hat{p}$ is unbiased. To see this, given that by definition the X^i's are independent and identically distributed (iid) to X, we have

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}}[\hat{p}(X)] &= \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{\mathbb{P}}[\delta_{X^i}(dx)] \\
&= \frac{1}{N} \cdot N \cdot \mathbb{E}_{\mathbb{P}}[\delta_X(dx)] \\
&= \int_{\mathcal{X}} \delta_X(dx) p(x) dx \\
&= p(X),
\end{aligned} \tag{A.4}$$

where $\hat{p}(X)$, $\delta_X(dx)$ and $\delta_{X^i}(dx)$ are understood as functions of the random variables X and X^i themselves, rather than their realized values.

Another important property of $\hat{p}(x)$ is that it is the estimator with smallest variance amongst the set of unbiased estimators for $p(x)$. This is clearly seen if we note that each $\delta_{X^i}(dx)$ is a Bernoulli distributed random variable with probability of success $p(X)$; we then have that $(\delta_{X^i}(dx))_{i=1}^N$ forms a random sample from this distribution, which has as its sufficient and complete statistic $N \cdot \hat{p}(X)$. The *Lehmann-Scheffé theorem* (Shao, 2003, p. 162) then establishes the desired result.

The estimator $\hat{p}$ also posesses good asymptotic properties. From (A.4) and by the Strong Law of Large Numbers (SLLN), we have that $\hat{p}$ converges almost surely to p. Also, since $\text{var}(\hat{p}) = p(1-p)/N$ is always finite, by the Central Limit Theorem (CLT) $\sqrt{N}(\hat{p} - p)$ converges in distribution to $\mathcal{N}(0, p(1-p))$, where $\mathcal{N}(\mu, \sigma^2)$ denotes a Normal distribution with mean μ and variance σ^2.

The CLT guarantees that convergence of $\hat{p}$ to p is $\mathcal{O}(N^{-1/2})$, which might be a descouraging result whenever more precise approximations are readily available, such as those based on numerical integration methods. However, most of these alternatives have $\mathcal{O}(N^{d_x})$ complexity, a fact widely known in the literature as *curse of dimensionality* (see e.g. Asmussen and Glynn, 2007, p. 264, Liu, 2008, p. 2 and Robert and Casella, 2004, p. 136.). Monte Carlo methods, on the other hand, do not suffer from this drawback, and typically require only $\mathcal{O}(N)$ operations to be performed; they are therefore invaluable in settings where the dimension of $\mathcal{X}$ is large.

Using Monte Carlo methods, we can also estimate the moments $\mathbb{E}_{\mathbb{P}}[h(X)]$ of any $\mathcal{B}$-measurable and $\mathbb{P}$-integrable function h of X. This is actually the most common setting for presenting MC methods; the "density-based" estimation approach taken so far is simply a convenient formalization. Since $\mathbb{E}_{\mathbb{P}}[h(X)] = \int_{\mathcal{X}} h(x)p(x)dx$, the MC approximation to this integral is obtained by simply replacing p with $\hat{p}$, i.e. by computing the expectation under $\hat{\mathbb{P}}$. More precisely,

$$\begin{aligned}
\hat{\mathbb{E}}_{\mathbb{P}}[h(X)] &:= \mathbb{E}_{\hat{\mathbb{P}}}[h(X)] \\
&= \int_{\mathcal{X}} h(x)\hat{p}(x)dx
\end{aligned}$$

$$= \int_{\mathcal{X}} h(x) \frac{1}{N} \sum_{i=1}^{N} \delta_{x^i}(dx)dx$$
$$= \frac{1}{N} \sum_{i=1}^{N} \int_{\mathcal{X}} h(x)\delta_{x^i}(dx)dx$$
$$= \frac{1}{N} \sum_{i=1}^{N} h(x^i), \tag{A.5}$$

where here we have used that $\int_{\mathcal{X}} h(x)\delta_a(dx)dx = h(a)$, a property of Dirac measures directly deduced from $\delta_a(\mathcal{X}) = 1$. Clearly, $\hat{\mathbb{E}}_{\mathbb{P}}[h(X)]$ thus defined is an unbiased estimator of $\mathbb{E}_{\mathbb{P}}[h(X)]$. Further, if the variance of $h(X)$ under $\mathbb{P}$, $\sigma_h^2 := \mathrm{var}_{\mathbb{P}}[h(X)]$, is bounded, the Strong Law of Large Numbers and the Central Limit Theorem also apply to $\hat{\mathbb{E}}_{\mathbb{P}}[h(X)]$, with almost sure convergence to $\mathbb{E}_{\mathbb{P}}[h(X)]$ and $\sqrt{N}\{\hat{\mathbb{E}}_{\mathbb{P}}[h(X)] - \mathbb{E}_{\mathbb{P}}[h(X)]\}$ converging in distribution to $\mathcal{N}(\mathbb{E}_{\mathbb{P}}[h(X)], \sigma_h^2)$.

A.2 Importance Sampling

So far, we have assumed that we can produce N iid copies or *particles* X^i from p[1]. This is a situation which is typically known as *perfect sampling*, as per the title of the last section. Assume now instead that we can no longer sample from p directly, but that we can sample from another distribution $\mathbb{Q}$ such that $\mathbb{Q}$ dominates $\mathbb{P}$ (we write $\mathbb{P} \ll \mathbb{Q}$) and which also has a probability density q with respect to dx. A simple change of measure allows us to express $\mathbb{P}(X \in A)$ for any $A \in \mathcal{B}$ in terms of $\mathbb{Q}$ as

$$\mathbb{P}(X \in A) = \int_A p(x)dx = \int_A \frac{q(x)}{q(x)} p(x)dx = \int_A \frac{p(x)}{q(x)} q(x)dx = \int_A \pi(x)p(x)dx, \tag{A.6}$$

where $\pi := p/q$ is called the *importance weight*. Note that, since $p/q = (d\mathbb{P}/dx)/(d\mathbb{Q}/dx) = d\mathbb{P}/d\mathbb{Q}$, by the Radon-Nikodym theorem this ratio is always well defined (π is the Radon-Nikodym derivative of $\mathbb{P}$ with respect to $\mathbb{Q}$, which always exists since both measures are finite and $\mathbb{P}$ is dominated by $\mathbb{Q}$).

Now, let $\pi^i := \pi(x^i) = p(x^i)/q(x^i)$ and define $w^i := \pi^i / \sum_{j=1}^N \pi^j$. Analogous to (A.3), our corresponding *importance sampling* (IS) estimator of p is defined by

$$\hat{p}(x) := \sum_{i=1}^{N} w^i \delta_{x^i}(dx). \tag{A.7}$$

The term "importance weight" comes from the fact that π indeed represents the "importance", i.e. relative probability, of each particle x^i. Since we draw from q in order to make inference about p, the usual terminology for these functions are the *proposal* and *target* densities, respectively. To distinguish between π and w, we often use the terms *unnormalized* and *normalized* importance weights. Note that by definition we have $w^i \geq 0$

[1]Technically, samples are produced according to the *law* of X, which is $\mathbb{P}$. However, since p is the unique probability density associated to $\mathbb{P}$, using them interchangeably whenever no confusion can be made is common practice and simplifies the exposition.

for all i and[2] $\sum_{i=1}^N w^i = \sum_{i=1}^N \pi^i / \sum_{j=1}^N \pi^j = 1$, i.e. the sampled particles and associated importance weights $(x^i, w^i)_{i=1}^N$ define a discrete probability distribution $\hat{\mathbb{P}}$ on $(\mathcal{X}, \mathcal{B})$.

A subtle point we have omitted so far is that although in IS we no longer require that direct sampling be made from p, we still assume that we can evaluate it pointwise. An advantage of working with the estimator (A.7), however, is that we only need to compute p up to a proportionality constant, since when computing w this constant is eliminated. To elaborate on this point further, assume that $\pi^i \propto p(x^i)/q(x^i)$ so that $\pi^i = K\check{p}(x^i)/q(x^i)$, where $\check{p}$ represents the part of p that we can evaluate analytically. Then

$$w^i = \frac{\pi^i}{\sum_{j=1}^N \pi^j} = \frac{K\check{p}(x^i)/q(x^i)}{\sum_{j=1}^N K\check{p}(x^j)/q(x^j)} = \frac{K}{K}\frac{\check{p}(x^i)/q(x^i)}{\sum_{j=1}^N \check{p}(x^j)/q(x^j)} = \frac{\check{p}(x^i)/q(x^i)}{\sum_{j=1}^N \check{p}(x^j)/q(x^j)},$$

i.e. we can still compute w^i using only the analytically available part of p (as long the rest of it is not a function of i).

Intuitively, the accuracy to which $\hat{\mathbb{P}}$ approximates $\mathbb{P}$ should be increasingly better as $N \to +\infty$, i.e. $\hat{p}$ should be a consistent estimator of p. To establish this, first note that the expected value of π (under $\mathbb{Q}$) is

$$\mathbb{E}_{\mathbb{Q}}[\pi(X)] = \int_{\mathcal{X}} \pi(x)q(x)dx = \int_{\mathcal{X}} \frac{p(x)}{q(x)}q(x)dx = \int_X p(x)dx = 1. \tag{A.8}$$

By the Weak Law of Large Numbers (SLLN), it follows that $\bar{\pi} := N^{-1}\sum_{i=1}^N \pi^i$ converges in probability (under $\mathbb{P}$)[3] to 1. Similarly, noting that $w^i = \pi^i / \sum_{j=1}^N \pi^j = N^{-1}\pi^i/\bar{\pi}$, by (A.7) we also have

$$\hat{p}(x) = \sum_{i=1}^N w^i \delta_{x^i}(dx) = \sum_{i=1}^N \frac{N^{-1}\pi^i}{\bar{\pi}}\delta_{x^i}(dx) \implies \bar{\pi}\cdot\hat{p}(x) = N^{-1}\sum_{i=1}^N \pi^i\delta_{x^i}(dx),$$

and, since under $\mathbb{Q}$ each $\pi^i\delta_{x^i}$ has expected value equal to (remember that the X^i's are iid)

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}}[\pi^i\delta_{X^i}(dx)] &= \int_{\mathcal{X}} \pi^i\delta_{X^i}(dx)q(x)dx \\
&= \int_{\mathcal{X}} \frac{p(x)}{q(x)}\delta_X(dx)q(x)dx \\
&= \int_{\mathcal{X}} p(x)\delta_{X^i}(dx)dx \\
&= \int_{\mathcal{X}} p(x)\delta_X(dx)dx \\
&= p(X),
\end{aligned}$$

[2] In order to ensure that the sum $\sum_{j=1}^N \pi^j$ is finite, it is enough that π^i be uniformly bounded for each i. This is satisfied by simply adopting the convention $0 \cdot \infty = 0$, since by definition $\mathbb{P} \gg \mathbb{Q}$ means that $q(x) = 0$ implies $p(x) = 0$ for any $x \in X$, which in turn implies that $w(x) = 0 \cdot 1/0 = 0 \cdot \infty = 0$.

[3] Remember that, since $\mathbb{P} \ll \mathbb{Q}$, convergence in probability under $\mathbb{Q}$ implies convergence in probability under $\mathbb{P}$.

by the WLLN we have that $\bar{\pi} \cdot \hat{p}$ converges in probability under $\mathbb{P}$ to p[4]. Finally, by Slutsky's theorem (Shao, 2003, p. 60) and continuous mapping of the function $x \mapsto x^{-1}$, $\hat{p} = \bar{\pi}/\bar{\pi} \cdot \hat{p}(x) = \bar{\pi}^{-1} \cdot (\bar{\pi} \cdot \hat{p})$ converges in probability to $1^{-1} \cdot p = p$, as required.

Note that in deriving the consistency result we cannot directly apply the WLLN to $\hat{p}$, since it involves the ratio of random variables. This is actually an example of a more general class of *ratio estimators*; see e.g. (Shao, 2003, p. 204). In particular, $\hat{p}$ is also biased for finite N.

Another desirable property of the IS estimator is that it has an associated Central Limit Theorem. Perhaps not surprisingly, the asymptotic distribution of $\hat{p}$ is the same as the MC estimator (A.3) under perfect sampling, i.e. as $N \to +\infty$ we have $\sqrt{N}(\hat{p} - p) \to^d \mathcal{N}(0, p(1-p))$. The proof in this case, however, is much more involved; see e.g. Geweke (1989) for further details. Crucially, the $\mathcal{O}(N^{-1/2})$ rate of convergence of IS is the same as in perfect sampling, and so is the $\mathcal{O}(N)$ complexity, with neither depending on $\dim(\mathcal{X})$.

So far we have merely stated that under any $\mathbb{Q} \gg \mathbb{P}$ the asymptotic behavior of $\hat{p}$ is the same. However, the finite sample performance of the estimator depends a great deal on the choice of proposal density. Heuristically, we want to choose q as "close" to p as possible, so that the variability of the weights are as small as possible. The following proposition establishes what the *optimal* choice of proposal distribution is, i.e. which choice of q leads to the smallest variance of π.

Proposition A.2.1. *The proposal distribution q that minimizes the variance of the importance weight π is $q \propto p$.*

Proof. Since $q \propto p$ is equivalent to $q = K \cdot p$, we have that

$$\pi(X) = \frac{p(X)}{Kp(X)} = \frac{1}{K},$$

where once again $\pi(X)$ is understood as a function of the random variable X. The variance of π under $\mathbb{Q}$ is therefore given by

$$\begin{aligned}
\mathrm{var}_{\mathbb{Q}}[\pi(X)] &= \mathbb{E}_{\mathbb{Q}}\{[\pi(X)]^2\} - \mathbb{E}^2_{\mathbb{Q}}[\pi(X)] \\
&= \int_{\mathcal{X}} [\pi(x)]^2 q(x) dx - \left[\int_{\mathcal{X}} \pi(x) q(x) dx\right]^2 \\
&= \int_{\mathcal{X}} \left[\frac{1}{K}\right]^2 q(x) dx - \left[\int_{\mathcal{X}} \frac{1}{K} q(x) dx\right]^2 \\
&= \left[\frac{1}{K}\right]^2 - \left[\frac{1}{K}\right]^2 \\
&= 0,
\end{aligned}$$

which is the minimum attainable variance for any random variable.

□

[4]Note that from the third to the fourth equations we have used the fact that under $\mathbb{P}$ the X^i's are equal in distribution to X, which implies that

$$\mathbb{E}_{\mathbb{P}}[\delta_{X^i}(dx)] = \mathbb{E}_{\mathbb{P}}[\delta_X(dx)] \iff \int_{\mathcal{X}} p(x)\delta_{X^i}(dx)dx = \int_{\mathcal{X}} p(x)\delta_X(dx)dx.$$

Although in most realistical situations we cannot sample from $q \propto p$ directly, Proposition A.2.1 provides theoretical justification for the heuristic presented above, i.e. that q should be "as close" to p as possible. In light of this result, this heuristic can be reinterpreted formally as choosing q so as to maintain the importance weights as constant/uniform as possible with as high a probability as possible, thus ensuring that the variance of π is relatively small.

In closing this section, note that IS clearly generalizes the case of perfect sampling by simply taking $q = p$, which implies that $\pi^i = 1$ and therefore $w^i = 1/N$ for each i. Not surprisingly, it follows from Proposition A.2.1 that this is also the optimal choice whenever possible.

A.3 Rao-Blackwellization

Rao-Blackwellization (Robert and Casella, 2004; Liu, 2008) is essentially based on the reasoning that even when performing inference based on Monte Carlo simulation methods, it is always beneficial to do as much analytical computation as possible. Although at first this might seem like simple intuition, it can be formally proved to be true based on the so-called *Rao-Blackwell inequality* (Corollary C.1.1). We will deal with the general Importance Sampling case here (see Section A.2), and defer to a treatment using the specialization of these ideas in the SMC context to Doucet et al. (2000).

First, assume perfect sampling conditions as in Section A.1. Suppose that we can decompose our random variable of interest $X \sim \mathbb{P}$ into $X = (X_1, X_2)$, that our interest lies in approximating $\mathbb{E}_{\mathbb{P}}[h(X)]$ for a $\mathcal{B}$-measurable and $\mathbb{P}$-integrable function h and that $\mathbb{E}_{\mathbb{P}}[h(X)|X_2]$ is known analytically.

Recall from Section A.1 that our estimator of $\mathbb{E}_{\mathbb{P}}[h(X)]$ (sometimes referred to as a *histogram-based estimator*) is given by

$$\hat{\mathbb{E}}_{\mathbb{P}}[h(X)] := \frac{1}{N}\sum_{i=1}^{N} h(x^i) \tag{A.9}$$

and define the corresponding *Rao-Blackwellized* estimator $\check{\mathbb{E}}_{\mathbb{P}}[h(X)]$ by

$$\check{\mathbb{E}}_{\mathbb{P}}[h(X)] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[h(x^i)|X_2]. \tag{A.10}$$

It is easy to see that (A.9) is unbiased for $\mathbb{E}_{\mathbb{P}}[h(X)]$, since it is an average of iid copies, each with expectation $\mathbb{E}_{\mathbb{P}}[h(X)]$. For (A.10), we can apply the Law of Total Expectation (Proposition C.1.1) to show that each term has expectation $\mathbb{E}_{\mathbb{P}}\{\mathbb{E}_{\mathbb{P}}[h(X)|X_2]\} = \mathbb{E}_{\mathbb{P}}[h(X)]$, also yielding an unbiased estimator.

The difference between (A.9) and (A.10) thus lies in their efficiency; by applying the Rao-Blackwell Inequality (Corollary C.1.1), we can conclude that, for each term in both estimators,

$$\mathrm{var}_{\mathbb{P}}[h(X^i)] \geq \mathrm{var}_{\mathbb{P}}\left\{\mathbb{E}_{\mathbb{P}}[h(X^i)|X_2]\right\}$$

and, since the copies are iid, this is clearly true for the estimators themselves as well. Note that although the argument was derived here using moment-based estimators, the same result holds for density-based estimation in general.

For the more general case of Importance Sampling, it is perhaps surprising that Rao-Blackwellization does not always yield more efficient estimators (this has to do with the

behavior of the normalized importance weights; see Liu 2008 for more details). It does, however, always leads to less variable importance weights, as shown in the following result.

Proposition A.3.1. *Let* $X := (X_1, X_2) \sim \mathbb{P}$ *on the probability space* $(\mathcal{X}, \mathcal{B}, \mathbb{P})$ *and* $\pi(x) := p(x)/q(x)$*, where* $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ *such that* $X_1 \in \mathcal{X}_1$*,* $X_2 \in \mathcal{X}_2$*,* $q(x) := d\mathbb{Q}(x)/dx$ *and* $\mathbb{Q} \gg \mathbb{P}$*. Then*

$$\text{var}_{\mathbb{Q}}[\pi(X)] \geq \text{var}_{\mathbb{Q}}[\pi_1(X_1)], \tag{A.11}$$

where $\pi_1(x_1) := p_1(x_1)/q_1(x_1)$*, with* $p_1(x_1) := \int_{\mathcal{X}_2} p(x)dx_2$ *and* $q_1(x_1) := \int_{\mathcal{X}_2} q(x)dx_2$*.*

Proof. Let $x = (x_1, x_2)$ and $q_{2|1}(x_2|x_1) := q(x_1, x_2)/q_1(x_1) = q(x)/q_1(x_1)$ be the conditional proposal density of X_2 given X_1. We can then write the *marginal importance weights* π_1 as

$$\begin{aligned}
\pi_1(X_1) &= \frac{p_1(x_1)}{q_1(x_1)} \\
&= \frac{\int_{\mathcal{X}_2} p(x_1, x_2)dx_2}{q_1(x_1)} \\
&= \int_{\mathcal{X}_2} \frac{p(x_1, x_2)}{q_1(x_1)} \frac{q_{2|1}(x_2|x_1)}{q_{2|1}(x_2|x_1)} dx_2 \\
&= \int_{\mathcal{X}_2} \frac{p(x_1, x_2)}{q_1(x_1) q_{2|1}(x_2|x_1)} q_{2|1}(x_2|x_1)dx_2 \\
&= \int_{\mathcal{X}_2} \frac{p(x_1, x_2)}{q_1(x_1) \frac{q(x_1,x_2)}{q_1(x_1)}} q_{2|1}(x_2|x_1)dx_2 \\
&= \int_{\mathcal{X}_2} \frac{p(x_1, x_2)}{q(x_1, x_2)} q_{2|1}(x_2|x_1)dx_2 \\
&= \int_{\mathcal{X}_2} \frac{p(x)}{q(x)} q_{2|1}(x_2|x_1)dx_2 \\
&= \mathbb{E}_{\mathbb{Q}}\left[\frac{p(X)}{q(X)}\middle| X_1\right] \\
&= \mathbb{E}_{\mathbb{Q}}[\pi(X)|X_1].
\end{aligned}$$

Finally, by applying the Rao-Blackwell inequality (Corollary C.1.1), we have

$$\text{var}_{\mathbb{Q}}[\pi(X)] \geq \text{var}_{\mathbb{Q}}\left\{\mathbb{E}_{\mathbb{Q}}[\pi(X)|X_1]\right\},$$

as required. □

In the form of Proposition A.3.1, we can see that the Rao-Blackwellization technique for obtaining more efficient weights in Importance Sampling is essentially a marginalization procedure, where we can obtain a reduction in variance by integrating out unnecessary components in the importance weights. As explored by e.g. Liu and Chen (1998), Doucet et al. (2000) and Carvalho et al. (2010) in the context of SMC methods, this type of marginalization should always be implemented whenever possible, and in practice can lead to substantial efficiency gains.

A.4 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) can be considered a major cornerstone method in Monte Carlo simulation, and is probably the most popular algorithm for estimating untractable posterior distributions in the context of Bayesian inference. In this section we will only touch upon what we believe are the most relevant aspects of MCMC to our work. For general references regarding both theoretical and applied aspects of MCMC algorithms, see e.g. Gamerman and Lopes (2006) and Liu (2008).

Essentially, MCMC algorithms rely on the construction of a Markov Chain with transition kernel $Q(x'|x)$ designed to have as stationary density the target density p. Although there are certain conditions (Doob, 1990) the Markov Chain needs to satisfy in order to possess (and reach) this stationary density, perhaps the most important of them is *invariance*, defined as

$$\int_{\mathcal{X}} p(x)Q(x'|x)dx = p(x'). \tag{A.12}$$

Regarding the choice of kernel, the most popular ones are the *Gibbs* and *Metropolis* (or *Metropolis-Hastings*, MH) kernels. The Metropolis kernel is defined by a transition function $q(x'|x)$ and *acceptance probability* as

$$Q(x'|x) := q(x'|x)\alpha(x'|x), \tag{A.13}$$

where

$$\alpha(x'|x) := 1 \wedge \frac{p(x')q(x|x')}{p(x)q(x'|x)} \tag{A.14}$$

and $x \wedge y := \min(x, y)$.

Now, although we can directly check that the MH kernel satisfies the invariance property (A.12), it is usually easier to check the stronger *detailed balance condition*, defined by

$$p(x)Q(x'|x) = p(x')Q(x|x'). \tag{A.15}$$

In general, a Markov Chain satifies detailed balance if and only if it is *reversible*, and reversibility in turn implies invariance. For the MH kernel, we have

$$\begin{aligned}
p(x)Q(x'|x) &= p(x)q(x'|x)\alpha(x'|x) \\
&= p(x)q(x'|x) \cdot \left\{ 1 \wedge \frac{p(x')q(x|x')}{p(x)q(x'|x)} \right\} \\
&= p(x)q(x'|x) \wedge p(x)q(x'|x)\frac{p(x')q(x|x')}{p(x)q(x'|x)} \\
&= p(x)q(x'|x) \wedge p(x')q(x|x') \\
&= p(x)q(x'|x)\frac{p(x')q(x|x')}{p(x')q(x|x')} \wedge p(x')q(x|x') \\
&= \frac{p(x)q(x'|x)}{p(x')q(x|x')}p(x')q(x|x') \wedge p(x')q(x|x') \\
&= \left\{ \frac{p(x)q(x'|x)}{p(x')q(x|x')} \wedge 1 \right\} \cdot p(x')q(x|x') \\
&= p(x')q(x|x') \cdot \left\{ 1 \wedge \frac{p(x)q(x'|x)}{p(x')q(x|x')} \right\} \\
&= p(x')q(x|x')\alpha(x|x')
\end{aligned}$$

$$= p(x')Q(x|x'),$$

where in the above derivation we have used the symmetry of the minimum operator, i.e. that $x \wedge y = y \wedge x$. This establishes that drawing from the MH kernel (A.13) always keeps p invariant.

The so-called *Metropolis-Hastings algorithm* is simply a MCMC procedure based on a Metropolis kernel. Starting with a draw x_0 from a prior distribution ν, the Markov Chain is updated until certain converge metrics are met (Robert and Casella, 2004). Since the procedure does not start with draws from the posterior p (which we assume to be impossible to sample from – this is the reason for using MCMC in the first place), assume that we reach the invariant density p only after B iterations. B is therefore referred to as the *burn-in* of the method, and since it is usually hard to determine analytically, in general it is simply set to a suitable value for which the system appears to be stationary (but see e.g. Gelman et al., 2013, for some caveats and pitfalls related to this approach). Unlike perfect sampling (Section A.1) or even Importance Sampling (Section A.2), we must therefore discard the samples obtained prior to reaching the burn-in period, i.e. if the chain is run for $B + M$ iterations, we only keep the last $(x^i)_{i=1}^M$ sampled values. The resulting set $(x^i)_{i=1}^M$ is a uniformly/equally weighted sample from p, albeit usually a dependent one due to the sequential construction of the method. This entire process is summarized in Algorithm A.1.

Algorithm A.1: Metropolis-Hastings Algorithm

Initialization
draw $x^0 \sim \nu(x)$
set $x \leftarrow x^0$

Main recursion
for $i = 1$ **to** $B + M$ **do**
  draw $x' \sim q(x'|x)$
  draw $u \sim U[0, 1]$
  compute $\alpha(x'|x) = 1 \wedge \frac{p(x')q(x|x')}{p(x)q(x'|x)}$
  if $u <= \alpha(x'|x)$ **then**
    set $x^i \leftarrow x'$
  end
  else
    set $x^i \leftarrow x$
  end
  set $x \leftarrow x^i$
end

Within the class of MH kernels, an important subset are the so-called *Random Walk Metropolis* (RWM) proposals. As their name implies, these consist of drawing new states x' from a distribution centered at x with arbitrary variance matrix Σ, obeying a random walk-type dynamic $x' = x + \Sigma^{1/2}z$, where $z \sim \text{iid}\,(0_{d_x}, I_{d_x})$, 0_{d_x} is a $d_x \times 1$ vector of zeros and I_{d_x} is the identity matrix of order d_x. As a popular example, when $z \sim \mathcal{N}(0_{d_x}, I_{d_x})$ the proposal is a *Gaussian Random Walk Metropolis* proposal, i.e.

$$q(x'|x) = d\mathcal{N}(x'|x, \Sigma). \tag{A.16}$$

Although RWM proposals are very popular and usually provide good results (Sherlock et al., 2010), in practice the scaling behavior of the algorithm (governed by Σ) can be hard to tune. Motivated by this fact, there have been several modifications of the "vanilla" RWM algorithm in order to allow for a scaling that is not only robust to outlying fluctuations but also adapts to the movement of the chain. Amongst these, we highlight the work of Vihola (2012), which introduced the so-called *Robust Adaptive Metropolis* (RAM) Algorithm.

Essentially, RAM targets a certain desired acceptance rate $\alpha_* \in [0,1]$ and uses a robust estimate of the variance matrix which is computed recursively. At iteration i, we sample $z^i \sim \mathcal{N}(0_{d_x}, I_{d_x})$ and set

$$x^i = x^{i-1} + S^{i-1} z^i,$$

where S^{i-1} satisfies

$$\Sigma^i := S^i (S^i)^T = S^{i-1} \left\{ I_{d_x} + \eta^i [\alpha(x^i | x^{i-1}) - \alpha_*] \frac{z^i (z^i)^T}{\|z^i\|_2^2} \right\} (S^{i-1})^T, \tag{A.17}$$

$\|z^i\|_2^2 := (z_1^i)^2 + \ldots + (z_{d_x}^i)^2 = (z^i)^T z^i$ and $(\eta^i)_{i \geq 1} \supset (0,1]$ is a sequence of decreasing *step sizes*.

It is important to point out that although the variance matrix Σ^i is computed recursively, it is implicitly a function of the entire past of the chain. This inevitably implies that even by adopting the MH rule (A.13) the corresponding kernel will not be invariant nor reversible (and not even Markovian), which is certainly not desirable. However, if we only update the variance matrix during the burn-in period, the resulting chain will still have the correct stationary distribution p. If the burn-in period is long enough for the variance matrix estimates to stabilize, we can therefore use Σ^B in subsequent draws, which typically ensure an acceptance rate of α_* and an improved mixing of the chain even without further adaptation (Vihola, 2012).

For a practical implementation of RAM, we usually do not evaluate Σ^i, since only S^i is necessary for sampling from the corresponding proposal. Since we can compute S^i via a rank one Cholesky update or downdate (depending on the sign of $[\alpha(x^i|x^{i-1}) - \alpha_*]$; see Vihola 2012), the method is computationally efficient in practice. Starting from a positive definite lower-diagonal matrix S_ν, the Gaussian version of RAM discussed here is summarized in Algorithm A.2. Note that since the Gaussian kernel is symmetric, i.e. $q(x^i|x^{i-1}) = q(x^{i-1}|x^i)$, the acceptance probability $\alpha(x^i|x^{i-1})$ in the algorithm is not a function of q.

Another widely popular class of kernels used for MCMC algorithms are the Gibbs kernels, the choice of which results in a subclass of MCMC methods known as *Gibbs sampling algorithms*. Essentially, Gibbs sampling is used whenever we know the *complete conditionals* $p(x_k|x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_{d_x})$ for each $k = 1, \ldots, d_x$, which is frequently the case in which we know the behavior of p only up to a global proportionality constant for which evaluation is difficult or unfeasible.

An attractive property of Gibbs sampling is that it always leaves p invariant, regardless of the order or even the schedule (i.e. deterministic or stochastic) chosen to update the components. In order to establish this, suppose that the chain is at $\mathbf{x}^i \sim p$, where $\mathbf{x}^i := (x_1^i, \ldots, x_{d_x}^i)$ and let $\mathbf{x}_{[-k]}^i := (x_1^i, \ldots, x_{k-1}^i, x_{k+1}^i, \ldots, x_{d_x}^i)$ for $k = 1, \ldots, d_x$. Then, by drawing

$$x_k^{i+1} \sim p(x_k | \mathbf{x}_{[-k]}^i),$$

Algorithm A.2: Gaussian Robust Adaptive Metropolis

Initialization
draw $x^0 \sim \nu(x)$
set $S^0 = S_\nu$
Main recursion
for $i = 1$ **to** $B + M$ **do**
  draw $z^i \sim d\mathcal{N}(0_{d_x}, I_{d_x})$
  compute $y^i = x^{i-1} + S^{i-1} z^i$
  draw $u \sim U[0,1]$
  compute $\alpha(y^i | x^{i-1}) = 1 \wedge \frac{p(y^i)}{p(x^{i-1})}$
  if $u <= \alpha(y^i | x^{i-1})$ **then**
    set $x^i = y^i$
  end
  else
    set $x^i = x^{i-1}$
  end
  compute S^i from $S^i (S^i)^T = S^{i-1} \left\{ I_{d_x} + \eta^i [\alpha(x^i | x^{i-1}) - \alpha_*] \frac{z^i (z^i)^T}{\|z^i\|_2^2} \right\} (S^{i-1})^T$
end

we have $(x_k^{i+1} | \mathbf{x}_{[-k]}^i) \perp\!\!\!\perp \mathbf{x}_{[-k]}^i$, implying that their joint probability density under $\mathbb{Q}$ is given by

$$\frac{d\mathbb{Q}(X_k \leq x_k^{i+1}, \mathbf{X}_{[-k]} \leq \mathbf{x}_{[-k]}^i)}{dx_k^{i+1} d\mathbf{x}_{[-k]}^i} = \frac{d\mathbb{Q}(X_k \leq x_k^{i+1})}{dx_k^{i+1}} \frac{d\mathbb{Q}(\mathbf{X}_{[-k]} \leq \mathbf{x}_{[-k]}^i)}{d\mathbf{x}_{[-k]}^i},$$

where $\mathbf{X}_{[-k]} := (X_1, \ldots, X_{k-1}, X_{k+1}, \ldots, X_{d_x})$. Therefore, since the chain is at p and we have sampled x_k^{i+1} from its complete conditional, we have

$$\begin{aligned} \frac{d\mathbb{Q}(X_k \leq x_k^{i+1})}{dx_k^{i+1}} \frac{d\mathbb{Q}(\mathbf{X}_{[-k]} \leq \mathbf{x}_{[-k]}^i)}{d\mathbf{x}_{[-k]}^i} &= p(x_k^{i+1} | \mathbf{x}_{[-k]}^i) p(\mathbf{x}_{[-k]}^i) \\ &= \frac{p(x_k^{i+1}, \mathbf{x}_{[-k]}^i)}{p(\mathbf{x}_{[-k]}^i)} p(\mathbf{x}_{[-k]}^i) \\ &= p(x_k^{i+1}, \mathbf{x}_{[-k]}^i), \end{aligned}$$

as required.

There are several variations of the "vanilla" Gibbs sampling discussed here in which the structure of the problem at hand is usually exploited in order to obtain efficiency gains; see e.g. Liu (2008) for various examples. Within the SMC context, there are also methods which make use of Gibbs sampling steps in order to sample states and even static parameters, such as e.g. Storvik's filter (Section 3.2.4.5), Particle Learning (Section 3.2.4.6), Hybrid Liu and West filter with Particle Learning (Section 3.2.4.7), Regularized Particle Learning (Section 3.2.4.9) and Hybrid Fully-Adapted Liu and West filter with Regularized Particle Learning (Section 3.2.4.10). All these methods thus possess the attractive property of leaving the posterior for the static parameters $p(\theta | y_{1:t})$ invariant at each step. For completeness, the vanilla Gibbs sampling method discussed here is

summarized in Algorithm A.3. Regarding computational efficiency, another attractive feature of Gibbs sampling in practice is the absence of a rejection sampling step as in MH.

Algorithm A.3: Vanilla Gibbs Sampler

Initialization
draw $x^0 \sim \nu(x)$
set $x \leftarrow x^0$

Main recursion
for $i = 1$ **to** $B + M$ **do**
  for $k = 1$ **to** d_x **do**
    draw $x'_k \sim p(x_k | x'_1, \ldots, x'_{k-1}, x_{k+1}, \ldots, x_{d_x})$
  end
  set $x^i \leftarrow x'$
  set $x \leftarrow x^i$
end

Appendix B

Linear and Gaussian Hidden Markov Models

Let $(X_t, Y_t)_{t\geq 0}$ be an HMM defined by

$$X_t = AX_{t-1} + RU_t, \qquad U_t \sim N(0, I_{d_x}), \tag{B.1}$$
$$Y_t = BX_t + SV_t, \qquad V_t \sim N(0, I_{d_y}), \tag{B.2}$$

where $(U_t)_{t\geq 0}$ and $(V_t)_{t\geq 0}$ are serially and mutually independent sequences which are also independent of $X_0 \sim N(X_\nu, \Sigma_\nu)$. Here, $d_x := \dim(X_t)$, $d_y := \dim(Y_t)$, $\mathcal{X} = \mathbb{R}^{d_x}$ and $\mathcal{Y} = \mathbb{R}^{d_y}$. The matrices A and R are $(d_x \times d_x)$, B is $(d_y \times d_x)$ and S is $(d_y \times d_y)$.

The model (B.1-B.2) is an important and recurrent type of HMM throughout the literature due to the fact that its filtering, prediction and smoothing distributions can all be computed exactly. These so-called *linear and Gaussian Hidden Markov Models* therefore provide us with a natural benchmark for which to test our methods against.

In this appendix we derive the corresponding analytical expressions for the distributions of $X_t|Y_{1:t}$, $X_t|Y_{1:t-1}$ and $X_t|Y_{1:n}$, as well as expressions for the likelihood $p(y_{1:n})$ and even discuss approximation of the posterior distribution $p(\theta|y_{1:n})$ via numerical integration. Although the notation here draws heavily on Cappé et al. (2005), our exposition is considerably simpler, and is inspired mostly by Petris (2009). Since throughout this entire chapter we only take expectations, variances and covariances with respect to $\mathbb{P}$, we write simply $\mathbb{E}$, cov and var to denote $\mathbb{E}_{\mathbb{P}}$, $\text{cov}_{\mathbb{P}}$ and $\text{var}_{\mathbb{P}}$, respectively.

B.1 The Regression Lemma

Before we move on to compute the filtering, prediction and smoothing state distributions of the linear and Gaussian HMM, we need to first prove an auxiliary result. Due to its importance in the theory of linear regression and more generally in multivariate statistics (Anderson, 2003), this result is sometimes known in the time series literature simply as the *regression lemma* (Durbin and Koopman, 2012).

Lemma B.1.1 (Regression Lemma). *Let $X \sim \mathcal{N}(\mu_x, \Sigma_{xx})$ and $Y \sim \mathcal{N}(\mu_y, \Sigma_{yy})$ be jointly distributed as*

$$\begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_{yy} \end{bmatrix} \right), \tag{B.3}$$

where $\Sigma_{xy} := \text{cov}(X, Y)$ *and* Σ_{yy} *is nonsingular. The conditional distribution of* X *given* Y *is therefore given by*

$$X|Y \sim \mathcal{N}(\mu_{x|y}, \Sigma_{x|y}), \tag{B.4}$$

where

$$\mu_{x|y} := \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y), \qquad \Sigma_{x|y} := \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T.$$

Proof. Let $F = (X, Y)$ and denote by μ_F and Σ_F its corresponding mean vector and covariance matrix given in (B.3). We can then write

$$\begin{aligned}
p(x|y) &= \frac{p(x,y)}{p(y)} \\
&= \frac{(2\pi)^{-(d_x+d_y)/2}\det(\Sigma_F)^{-1/2}\exp\left\{-\frac{1}{2}(F-\mu_F)^T\Sigma_F^{-1}(F-\mu_F)\right\}}{(2\pi)^{-d_y/2}\det(\Sigma_{yy})^{-1/2}\exp\left\{-\frac{1}{2}(y-\mu_y)^T\Sigma_{yy}^{-1}(y-\mu_y)\right\}} \\
&= (2\pi)^{-d_x/2}\left[\frac{\det(\Sigma_F)}{\det(\Sigma_{yy})}\right]^{-1/2} \cdot \\
&\qquad \cdot \exp\left\{-\frac{1}{2}\left[(F-\mu_F)^T\Sigma_F^{-1}(F-\mu_F) - (y-\mu_y)^T\Sigma_{yy}^{-1}(y-\mu_y)\right]\right\}.
\end{aligned}$$

Now, for a generic (2×2) block matrix E such that

$$E = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

it can be proved (Lu and Shiou, 2002) that, if D is nonsingular, the determinant of E is given by

$$\det(E) = \det\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = \det(D)\cdot\det(A - BD^{-1}C). \tag{B.5}$$

Further, if both D and $(A - BD^{-1}C)$ are nonsingular, the inverse of E is

$$\begin{aligned}
E^{-1} &= \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \\
&= \begin{bmatrix} (A-BD^{-1}C)^{-1} & -(A-BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A-BD^{-1}C)^{-1} & D + D^{-1}C(A-BD^{-1}C)^{-1}BD^{-1}. \end{bmatrix}
\end{aligned} \tag{B.6}$$

Therefore, we can use (B.5) to rewrite the determinant of Σ_F as

$$\det(\Sigma_F) = \det\left(\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_{yy} \end{bmatrix}\right) = \det\left(\Sigma_{yy}\right)\cdot\det\left(\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T\right),$$

which implies that the ratio $\det(\Sigma_F)/\det(\Sigma_{yy})$ appearing in the expression for $p(x|y)$ above is equal to

$$\frac{\det(\Sigma_F)}{\det(\Sigma_{yy})} = \frac{\det(\Sigma_{yy})\cdot\det(\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T)}{\det(\Sigma_{yy})} = \det(\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T) = \det(\Sigma_{x|y}),$$

where here we define $\Sigma_{x|y} := \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T$.

On the other hand, (B.6) implies that we can rewrite the inverse of Σ_F as

$$\begin{aligned}
\Sigma_F^{-1} &= \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_{yy} \end{bmatrix}^{-1} \\
&= \begin{bmatrix} (\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T)^{-1} & -(\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T)^{-1}\Sigma_{xy}\Sigma_{yy}^{-1} \\ -\Sigma_{yy}^{-1}\Sigma_{xy}^T(\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T)^{-1} & \Sigma_{yy} + \Sigma_{yy}^{-1}\Sigma_{xy}^T(\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T)^{-1}\Sigma_{xy}\Sigma_{xy}^{-1} \end{bmatrix} \\
&= \begin{bmatrix} \Sigma_{x|y}^{-1} & -\Sigma_{x|y}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1} \\ -\Sigma_{yy}^{-1}\Sigma_{xy}^T\Sigma_{x|y}^{-1} & \Sigma_{yy} + \Sigma_{yy}^{-1}\Sigma_{xy}^T\Sigma_{x|y}^{-1}\Sigma_{xy}\Sigma_{xy}^{-1} \end{bmatrix}.
\end{aligned}$$

Using this block inverse expression for Σ_F^{-1}, we can expand the quadratic form $(F - \mu_F)\Sigma_F^{-1}(F - \mu_F)$ as

$$\begin{aligned}
(F - \mu_F)\Sigma_F^{-1}(F - \mu_F) &= \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix}^T \begin{bmatrix} \Sigma_{x|y}^{-1} & -\Sigma_{x|y}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1} \\ -\Sigma_{yy}^{-1}\Sigma_{xy}^T\Sigma_{x|y}^{-1} & \Sigma_{yy} + \Sigma_{yy}^{-1}\Sigma_{xy}^T\Sigma_{x|y}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1} \end{bmatrix} \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix} \\
&= (x - \mu_x)^T\Sigma_{x|y}^{-1}(x - \mu_x) + \\
&\quad - (y - \mu_y)^T\Sigma_{yy}^{-1}\Sigma_{xy}^T\Sigma_{x|y}^{-1}(x - \mu_x) + \\
&\quad - (x - \mu_x)^T\Sigma_{x|y}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y) + \\
&\quad + (y - \mu_y)^T\big(\Sigma_{yy} + \Sigma_{yy}^{-1}\Sigma_{xy}^T\Sigma_{x|y}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1}\big)(y - \mu_y) \\
&= (x - \mu_x)^T\Sigma_{x|y}^{-1}(x - \mu_x) + \\
&\quad - \big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big]^T\Sigma_{x|y}^{-1}(x - \mu_x) + \\
&\quad - (x - \mu_x)^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + \\
&\quad + \big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big]^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] \\
&\quad + (y - \mu_y)^T\Sigma_{yy}(y - \mu_y) \\
&= (x - \mu_x)^T\Sigma_{x|y}^{-1}(x - \mu_x) + \\
&\quad - 2(x - \mu_x)^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + \\
&\quad + \big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big]^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + \\
&\quad + (y - \mu_y)^T\Sigma_{yy}(y - \mu_y) \\
&= x^T\Sigma_{x|y}^{-1}x - 2x_x^T\Sigma_{x|y}^{-1}\mu_x + \mu_x^T\Sigma_{x|y}^{-1}\mu_x + \\
&\quad - 2x^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + 2\mu^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + \\
&\quad + \big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big]^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + \\
&\quad + (y - \mu_y)^T\Sigma_{yy}(y - \mu_y) \\
&= x^T\Sigma_{x|y}^{-1}x + \\
&\quad - 2x_x^T\Sigma_{x|y}^{-1}\big[\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + \\
&\quad + \mu^T\Sigma_{x|y}^{-1}\mu + \\
&\quad + 2\mu^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + \\
&\quad + \big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big]^T\Sigma_{x|y}^{-1}\big[\Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)\big] + \\
&\quad + (y - \mu_y)^T\Sigma_{yy}(y - \mu_y) \\
&= x^T\Sigma_{x|y}^{-1}x +
\end{aligned}$$

$$
\begin{aligned}
&\quad - 2x_x^T \Sigma_{x|y}^{-1}\big[\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y-\mu_y)\big] + \\
&\quad + \big[\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y-\mu_y)\big]^T \Sigma_{x|y}^{-1}\big[\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y-\mu_y)\big] + \\
&\quad + (y-\mu_y)^T \Sigma_{yy}(y-\mu_y) \\
&= (x-\mu_{x|y})^T \Sigma_{x|y}^{-1}(x-\mu_{x|y}) + (y-\mu_y)^T \Sigma_{yy}(y-\mu_y),
\end{aligned}
$$

where $\mu_{x|y} := \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y-\mu_y)$. This in turn implies that the difference $(F-\mu_F)\Sigma_F^{-1}(F-\mu_F) - (y-\mu_y)^T\Sigma_{yy}(y-\mu_y)$ is then simply

$$
(x-\mu_{x|y})^T \Sigma_{x|y}^{-1}(x-\mu_{x|y}) + (y-\mu_y)^T \Sigma_{yy}(y-\mu_y) - (y-\mu_y)^T \Sigma_{yy}(y-\mu_y),
$$

which is clearly equal to $(x-\mu_{x|y})^T \Sigma_{x|y}^{-1}(x-\mu_{x|y})$.

Combinining all these results, we can finally rewrite the corresponding expression for $p(x|y)$ as

$$
\begin{aligned}
p(x|y) &= (2\pi)^{-d_x/2}\left[\frac{\det(\Sigma_F)}{\det(\Sigma_{yy})}\right]^{-1/2} \cdot \\
&\qquad \cdot \exp\left\{ -\frac{1}{2}\Big[(F-\mu_F)^T\Sigma_F^{-1}(F-\mu_F) - (y-\mu_y)^T\Sigma_{yy}^{-1}(y-\mu_y)\Big]\right\} \\
&= (2\pi)^{-d_x/2}\det(\Sigma_{x|y})^{-1/2}\exp\left\{ -\frac{1}{2}(x-\mu_{x|y})^T\Sigma_{x|y}^{-1}(x-\mu_{x|y})\right\},
\end{aligned}
$$

which is the density of a normally distributed variable with mean vector $\mu_{x|y}$ and variance matrix $\Sigma_{x|y}$, as required.

□

Note that in the above proof we have extensively used that the transpose of a product is given by the product of the transposes in the reverse order, i.e. $(AB)^T = B^T A^T$. Another property we have used is that a quadratic form can be equivalently written either as $z^T A w$ or $w^T A^T z$, since the end result is a scalar (which is, by definition, symmetric). This is also equal to $w^T A z$ for a symmetric matrix A, which is clearly true for $\Sigma_{x|y}^{-1}$ and its inverse, given that $\Sigma_{x|y}^T = (\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T)^T = \Sigma_{xx}^T - (\Sigma_{xy}^T)^T(\Sigma_{yy}^{-1})^T\Sigma_{xy}^T = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T = \Sigma_{x|y}$ (both Σ_{xx} and Σ_{yy} are symmetric, and if a matrix is symmetric, the same holds true for its inverse). Another result deduced from $z^T A w = w^T A z$ and that we have also extensively used is the association formula for symmetric quadratic forms $(z-w)^T A (z-w) = z^T A z - 2 z^T A w + w^T A w$.

B.2 Kalman Filtering and Prediction

Let $(X_t, Y_t)_{t\geq 0}$ be given by (B.1-B.2). In this section we consider the problem of computing the filtered law $X_t|Y_{1:t}$, resulting into a celebrated algorithm widely known as the *Kalman filter* (Kalman, 1960). As a direct byproduct of this procedure, the analytical expression for the prediction distribution $X_t|Y_{1:t-1}$ can also be obtained.

The Kalman filter is essentially an efficient sequential procedure to compute the mean and variance of $X_t|Y_{1:t}$, which then completely characterizes its distribution (Gaussian laws are almost-surely determined by their first and second moments). Hereafter, we let $X_{k|j} := \mathbb{E}(X_k|Y_{1:j})$ and $\Sigma_{k|j} := \mathrm{var}(X_k|Y_{1:j})$ for integers k and j to simplify notation.

Assume that at time $t-1$ we have already computed $X_{t-1|t-1}$ and $\Sigma_{t-1|t-1}$. From $X_t = AX_{t-1} + RU_t$, it follows that the one-step-ahead prediction state mean $X_{t|t-1}$ is then given by

$$\begin{aligned} X_{t|t-1} :&= \mathbb{E}(X_t|Y_{1:t-1}) \\ &= \mathbb{E}(AX_{t-1} + RU_t|Y_{1:t-1}) \\ &= A\mathbb{E}(X_{t-1}|Y_{1:t-1}) + R\mathbb{E}(U_t|Y_{1:t-1}) \\ &= AX_{t-1|t-1}, \end{aligned}$$

since $U_t \perp\!\!\!\perp Y_{1:t-1}$ implies $\mathbb{E}(U_t|Y_{1:t-1}) = \mathbb{E}(U_t) = 0$. Likewise, the one-step-ahead prediction state variance $\Sigma_{t|t-1}$ is

$$\begin{aligned} \Sigma_{t|t-1} &:= \text{var}(X_t|Y_{1:t-1}) \\ &= \text{var}(AX_{t-1} + RU_t|Y_{1:t-1}) \\ &= \text{var}(AX_{t-1}|Y_{1:t-1}) + \text{var}(RU_t|Y_{1:t-1}) + 2\text{cov}(AX_{t-1}, RU_t|Y_{1:t-1}) \\ &= A\text{var}(X_{t-1}|Y_{1:t-1})A^T + R\text{var}(U_t|Y_{1:t-1})R^T + 0_{(d_x \times d_x)} \\ &= A\Sigma_{t-1|t-1}A^T + RR^T, \end{aligned}$$

since $U_t \perp\!\!\!\perp Y_{1:t-1}$ implies $\text{var}(U_t|Y_{1:t-1}) = \text{var}(U_t) = I_{d_x}$ and $X_{t-1}|Y_{1:t-1} \perp\!\!\!\perp U_t|Y_{1:t-1}$ implies that their covariance is a $(d_x \times d_x)$ zero matrix, i.e. $\text{cov}(X_{t-1}, U_t|Y_{1:t-1}) = 0_{(d_x \times d_x)}$.

Now, consider the one-step-ahead prediction mean and variance of the *observation* Y_t given $Y_{1:t-1}$. From $Y_t = BX_t + SV_t$, we have

$$\mathbb{E}(Y_t|Y_{1:t-1}) = \mathbb{E}(BX_t + SV_t|Y_{1:t-1}) = B\mathbb{E}(X_t|Y_{1:t-1}) + S\mathbb{E}(V_t|Y_{1:t-1}) = BX_{t|t-1},$$

since $V_t \perp Y_{1:t-1}$ implies $\mathbb{E}(V_t|Y_{1:t-1}) = \mathbb{E}(V_t) = 0$. The corresponding variance matrix is

$$\begin{aligned} \text{var}(Y_t|Y_{1:t-1}) &= \text{var}(BX_t + SV_t|Y_{1:t-1}) \\ &= \text{var}(BX_t|Y_{1:t-1}) + \text{var}(SV_t|Y_{1:t-1}) + 2\text{cov}(BX_t, SV_t|Y_{1:t-1}) \\ &= B\text{var}(X_t|Y_{1:t-1})B^T + S\text{var}(V_t|Y_{1:t-1})S^T + 0_{(d_y \times d_y)} \\ &= B\Sigma_{t-1|t-1}B^T + SS^T, \end{aligned}$$

since $V_t \perp\!\!\!\perp Y_{1:t-1}$ implies $\text{var}(V_t|Y_{1:t-1}) = \text{var}(V_t) = I_{d_y}$ and $X_t|Y_{1:t-1} \perp\!\!\!\perp V_t|Y_{1:t-1}$ implies that their covariance is a $(d_y \times d_y)$ matrix of zeroes, i.e. $\text{cov}(X_{t-1}, V_t|Y_{1:t-1}) = 0_{(d_y \times d_y)}$.

Finally, consider the covariance between these one-step-ahead state and observation predictions, i.e. between $X_t|Y_{1:t-1}$ and $Y_t|Y_{1:t-1}$. It is given by

$$\begin{aligned} \text{cov}(X_t, Y_t|Y_{1:t-1}) &= \text{cov}(X_t, BX_t + SV_t, Y_{1:t-1}) \\ &= \text{cov}(X_t, BX_t|Y_{1:t-1}) + \text{cov}(X_t, SV_t|Y_{1:t-1}) \\ &= \Sigma_{t|t-1}B^T + 0_{(d_x \times d_y)} \\ &= \Sigma_{t|t-1}B^T, \end{aligned}$$

since we again have that $X_t|Y_{1:t-1} \perp\!\!\!\perp V_t|Y_{1:t-1}$ implies that their covariance is the zero $(d_x \times d_y)$ matrix $0_{(d_x \times d_y)}$. Note that the covariance between $Y_t|Y_{1:t-1}$ and $X_t|Y_{1:t-1}$ is simply the transpose of $\text{cov}(X_t, Y_t|X_{1:t-1})$, which is given by $B\Sigma_{t|t-1}$.

Given that the sigma-algebra generated by $Y_{1:t}$ clearly contains that of $Y_{1:t-1}$, i.e. $\sigma(Y_{1:t}) \supset \sigma(Y_{1:t-1})$, we have that the conditional distribution of $X_t|Y_{1:t-1}$ given $Y_t|Y_{1:t-1}$ is

the filtering distribution, i.e. $(X_t|Y_{1:t-1})|(Y_t|Y_{1:t-1}) =^d X_t|Y_{1:t}$. Since the joint distribution (see Remark B.2.1 below) of $X_t|Y_{1:t-1}$ and $Y_t|Y_{1:t-1}$ is given by

$$\begin{bmatrix} X_t|Y_{1:t-1} \\ Y_t|Y_{1:t-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} X_{t|t-1} \\ BX_{t|t-1} \end{bmatrix}, \begin{bmatrix} \Sigma_{t|t-1} & \Sigma_{t|t-1}B^T \\ B\Sigma_{t|t-1} & B\Sigma_{t|t-1}B^T + SS^T \end{bmatrix} \right),$$

by Lemma B.1.1 we have that the filtering distribution is Gaussian with mean

$$X_{t|t} := \mathbb{E}(X_t|Y_{1:t}) = X_{t|t-1} + \Sigma_{t|t-1}B^T\left(B\Sigma_{t|t-1}B^T + SS^T\right)^{-1}(Y_t - BX_{t|t-1})$$

and variance

$$\Sigma_{t|t} := \mathrm{var}(X_t|Y_{1:t}) = \Sigma_{t|t-1} - \Sigma_{t|t-1}B^T\left(B\Sigma_{t|t-1}B^T + SS^T\right)^{-1}B\Sigma_{t|t-1}.$$

Remark B.2.1. To use Lemma B.1.1, we additionally need to prove that $X_t|Y_{1:t-1}$ and $Y_t|Y_{1:t-1}$ are *jointly* Gaussian, which is in general a sufficient but not a necessary condition for them to be *marginally* Gaussian.

By directly rewriting the joint density of $X_t|Y_{1:t-1}$ and $Y_t|Y_{1:t-1}$ as

$$p(x_t, y_t|y_{1:t-1}) = p(y_t|x_t, y_{1:t-1})p(x_t|y_{1:t-1}) = g(y_t|x_t)p(x_t|y_{1:t-1}),$$

we then simply have to show that $X_t|Y_{1:t-1}$ is Gaussian, since $g(y_t|x_t)$ is Gaussian by definition, and the product of two Gaussian densities is also Gaussian (note that the last equality above follows from the fact that Y_t is almost surely determined by X_t).

We have

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int_{\mathcal{X}} p(x_t, x_{t-1}|y_{1:t-1})dx_{t-1} \\ &= \int_{\mathcal{X}} p(x_t|x_{t-1}, y_{1:t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \\ &= \int_{\mathcal{X}} f(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}, \end{aligned}$$

which establishes that $X_t|Y_{1:t-1}$ is a convolution of the Gaussian densities $f(x_t|x_{t-1})$ and $p(x_{t-1}|y_{1:t-1})$ (the latter follows from induction), which is once again Gaussian, as required.

It is usual (see e.g. Durbin and Koopman, 2012; Cappé et al., 2005; Petris, 2009) to restate the above result in terms of the one-step-ahead observation prediction *error* $\epsilon_t := Y_t - \mathbb{E}(Y_t|Y_{1:t-1}) = Y_t - BX_{t|t-1}$. Given $Y_{1:t-1}$, the mean of ϵ_t is clearly zero and its variance is the same as that of $Y_t|Y_{1:t-1}$. Defining this variance by Γ_t, and also defining the so-called *Kalman gain* by $K_t := \Sigma_{t|t-1}B^T\Gamma_t^{-1}$, we can collectively restate the entire sequential procedure as

$$X_{t|t-1} = AX_{t-1|t-1}, \tag{B.7}$$
$$\Sigma_{t|t-1} = A\Sigma_{t-1|t-1}A^T + RR^T, \tag{B.8}$$
$$\epsilon_t = Y_t - BX_{t|t-1}, \tag{B.9}$$
$$\Gamma_t = B\Sigma_{t|t-1}B^T + SS^T, \tag{B.10}$$
$$K_t = \Sigma_{t|t-1}B^T\Gamma_t^{-1}, \tag{B.11}$$
$$X_{t|t} = X_{t|t-1} + K_t\epsilon_t, \tag{B.12}$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - K_t B \Sigma_{t|t-1}. \tag{B.13}$$

Equations (B.7-B.8) are sometimes known as the *Kalman prediction* equations, whereas equations (B.9-B.13) are the *Kalman filtering* equations. Note that we can completely restate the filtering set of equations without the prediction part by simply replacing the appropriate expressions for $X_{t|t-1}$ and $\Sigma_{t|t-1}$ given in (B.7-B.8) in (B.9-B.13).

We initialize the recursion with $X_{0|0} := X_\nu$ and $\Sigma_{0|0} := \Sigma_\nu$, which are typically set to the mean and variance of X_0. The entire Kalman filtering procedure is summarized in Algorithm B.1.

Algorithm B.1: Kalman Filter

Initialization

set $X_{0|0} = X_\nu$
set $\Sigma_{0|0} = \Sigma_\nu$
compute $X_{1|0} = AX_\nu$
compute $\Sigma_{1|0} = A\Sigma_\nu A^T + RR^T$
compute $\epsilon_1 = Y_1 - BX_{1|0}$
compute $\Gamma_1 = B\Sigma_{1|0}B^T + SS^T$
compute $K_1 = \Sigma_{1|0}B^T\Gamma_1^{-1}$
compute $X_{1|1} = X_{1|0} + K_1\epsilon_1$
compute $\Sigma_{1|1} = \Sigma_{1|0} - K_1 B\Sigma_{1|0}$

Main recursion

for $t = 2$ **to** n **do**
    compute $X_{t|t-1} = AX_{t-1|t-1}$
    compute $\Sigma_{t|t-1} = A\Sigma_{t-1|t-1}A^T + RR^T$
    compute $\epsilon_t = Y_t - BX_{t|t-1}$
    compute $\Gamma_t = B\Sigma_{t|t-1}B^T + SS^T$
    compute $K_t = \Sigma_{t|t-1}B^T\Gamma_t^{-1}$
    compute $X_{t|t} = X_{t|t-1} + K_t\epsilon_t$
    compute $\Sigma_{t|t} = \Sigma_{t|t-1} - K_t B\Sigma_{t|t-1}$
end

B.3 Forward-Filtering, Backward-Sampling

Before computing the smoothing distributions $X_t|Y_{1:n}$ of linear and Gaussian models, we first derive a recursive method for sampling from these distributions, known as the *Forward-Filtering, Backward-sampling* (FFBS) algorithm. The FFBS is an invaluable ingredient when performing Bayesian inference for model (B.1-B.2), and was proposed independently by Carter and Kohn (1994), Frühwirth-Schnatter (1994) and Shephard (1994). Algorithms which have as their main goal to produce draws from the joint smoothing distribution $X_{0:n}|Y_{1:n}$ (rather than computing it analytically), are usually referred to as *simulation smoothers*.

The FFBS algorithm relies on the backward smoothing recursion

$$p(x_{0:n}|y_{1:n}) = p(x_0|x_{1:n}, y_{1:n}) \cdots p(x_n|y_{1:n})$$

$$= \prod_{k=0}^{n} p(x_k|x_{k+1:n}, y_{1:n}), \tag{B.14}$$

which suggests filtering up to $X_n|Y_{1:n}$ and then sampling backwards $X_{n-1}|(X_n, Y_{1:n})$, ..., $X_0|(X_{1:n}, Y_{1:n})$; hence the name. Now, each term $p(x_k|x_{k+1:n}, y_{1:n})$ is equivalent to

$$\begin{aligned} p(x_k|x_{k+1:n}, y_{1:n}) &= \frac{p(x_k, x_{k+1:n}, y_{1:n})}{p(x_{k+1:n}, y_{1:n})} \\ &= \frac{p(y_{k+1:n}|x_k, x_{k+1:n}, y_{1:k})p(x_{k+2:n}|x_k, x_{k+1}, y_{1:k})p(x_k|x_{k+1}, y_{1:k})p(x_{k+1}, y_{1:k})}{p(y_{k+1:n}|, x_{k+1:n}, y_{1:k})p(x_{k+2:n}|x_{k+1}, y_{1:k})p(x_{k+1}, y_{1:k})} \\ &= p(x_k|x_{k+1}, y_{1:k}), \end{aligned} \tag{B.15}$$

since from (1.2) and item (iii) of Proposition 1.1.1 comes

$$p(y_{k+1:n}|x_k, x_{k+1:n}, y_{1:k}) = p(y_{k+1:n}|x_{k+1:n}, y_{1:k}) = \prod_{j=k+1}^{n} g(y_j|x_j)$$

and from item (ii) of Proposition 1.1.1 comes

$$\begin{aligned} p(x_{k+2:n}|x_k, x_{k+1}, y_{1:k}) &= p(x_n|x_{k+2:n-1}, x_k, x_{k+1}, y_{1:k}) \cdots p(x_{k+2}|x_k, x_{k+1}, y_{1:k}) \\ &= \prod_{j=k+2}^{n} f(x_j|x_{j-1}) \\ &= p(x_{k+2:n}|x_{k+1}, y_{1:k}). \end{aligned}$$

Having established that $X_t|(X_{t+1}, Y_{1:n}) =^d X_t|(X_{t+1}, Y_{1:t})$, recall from Section B.2 that

$$X_t|Y_{1:t} \sim \mathcal{N}(X_{t|t}, \Sigma_{t|t}), \qquad X_{t+1}|Y_{1:t} \sim \mathcal{N}(AX_{t|t}, A\Sigma_{t|t}A^T + RR^T).$$

On the other hand, from (B.1) we can deduce that

$$\begin{aligned} \operatorname{cov}(X_t, X_{t+1}|Y_{1:t}) &= \operatorname{cov}(X_t, AX_t + RU_t|Y_{1:t}) \\ &= \operatorname{cov}(X_t, AX_t|Y_{1:t}) + \operatorname{cov}(X_t, RU_t|Y_{1:t}) \\ &= \operatorname{cov}(X_t, X_t|Y_{1:t})A^T + \operatorname{cov}(X_t, U_t|Y_{1:t})R^T \\ &= \operatorname{var}(X_t|Y_{1:t})A^T + 0_{d_x \times d_x}R^T \\ &= \Sigma_{t|t}A^T, \end{aligned}$$

since by assumption $\operatorname{cov}(X_t, U_t|Y_{1:t}) = 0$. Therefore, the joint distribution of $X_t|Y_{1:t}$ and $X_{t+1}|Y_{1:t}$ is given by

$$\begin{bmatrix} X_t|Y_{1:t} \\ X_{t+1}|Y_{1:t} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} X_{t|t} \\ AX_{t|t} \end{bmatrix}, \begin{bmatrix} \Sigma_{t|t} & \Sigma_{t|t}A^T \\ A\Sigma_{t|t} & A\Sigma_{t|t}A^T + RR^T \end{bmatrix} \right).$$

which, by Lemma B.1.1 and (B.15) implies that $X_t|(X_{t+1:n}, Y_{1:n})$ is Gaussian with mean

$$X_{t|n}^{\text{FFBS}} := X_{t|t} + \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}(X_{t+1} - AX_{t|t}) \tag{B.16}$$

and variance

$$\Sigma_{t|n}^{\text{FFBS}} := \Sigma_{t|t} - \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}A\Sigma_{t|t}. \tag{B.17}$$

Starting with $X_n|Y_{1:n}$, the complete FFBS procedure for producing a draw from $X_{0:n}|Y_{1:n}$ is summarized in Algorithm B.2.

Algorithm B.2: Forward-Filtering, Backward-Sampling

Initialization
draw $X_n|Y_{1:n} \sim \mathcal{N}(X_{n|n}, \Sigma_{n|n})$

Backward recursion
for $t = n\text{-}1$ **to** 0 **do**
compute $X_{t|n}^{\text{FFBS}} = X_{t|t} + \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}(X_{t+1} - AX_{t|t})$
compute $\Sigma_{t|n}^{\text{FFBS}} = \Sigma_{t|t} - \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}A\Sigma_{t|t}$
draw $X_t|(X_{t+1:n}, Y_{1:n}) \sim \mathcal{N}(X_{t|n}^{\text{FFBS}}, \Sigma_{t|n}^{\text{FFBS}})$
end

B.4 Kalman Smoothing

We finish our discussion of Kalman-filter based techniques with the so-called *Kalman smoothing* algorithm, which is the forward-backward smoother for linear and Gaussian models.

Recall from Section B.3 that $X_t|(X_{t+1}, Y_{1:n}) \sim \mathcal{N}(X_{t|n}^{\text{FFBS}}, \Sigma_{t|n}^{\text{FFBS}})$, where $X_{t|n}^{\text{FFBS}}$ and $\Sigma_{t|n}^{\text{FFBS}}$ are given in (B.16) and (B.17), respectively. Since $X_t|(X_{t+1}, Y_{1:n})$ is a function of X_{t+1}, it turns out that deriving the distribution of $X_t|Y_{1:n}$ from $X_t|(X_{t+1}, Y_{1:n})$ is as simple as using the Law of Total Expectation (Proposition C.1.1) and the Law of Total Variance (Proposition C.1.2) in order to obtain the corresponding mean and variances, since Gaussian distributions are completely determined by their first two moments.

For the mean $X_{t|n} := \mathbb{E}(X_t|Y_{1:n})$, by applying Proposition C.1.1 we obtain

$$\begin{aligned}
X_{t|n} &= \mathbb{E}\big[\mathbb{E}(X_t|X_{t+1}, Y_{1:n})|Y_{1:n}\big] \\
&= \mathbb{E}\big[X_{t|t} + \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}(X_{t+1} - AX_{t|t})|Y_{1:n}\big] \\
&= X_{t|t} + \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}(\mathbb{E}(X_{t+1}|Y_{1:n}) - AX_{t|t}) \\
&= X_{t|t} + \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}(X_{t+1|n} - AX_{t|t}). \qquad \text{(B.18)}
\end{aligned}$$

As for the variance $\Sigma_{t|n} := \text{var}(X_t|Y_{1:n})$, let $M_t := (A\Sigma_{t|t}A^T + RR^T)^{-1}$. Then, by Proposition C.1.2,

$$\begin{aligned}
\Sigma_{t|n} &= \text{var}\big[\mathbb{E}(X_t|X_{t+1}, Y_{1:n})|Y_{1:n}\big] + \mathbb{E}\big[\text{var}(X_t|X_{t+1}, Y_{1:n})|Y_{1:n}\big] \\
&= \text{var}\big[X_{t|t} + \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}(X_{t+1} - AX_{t|t})|Y_{1:n}\big] + \\
&\quad + \mathbb{E}\big[\Sigma_{t|t} - \Sigma_{t|t}A^T(A\Sigma_{t|t}A^T + RR^T)^{-1}A\Sigma_{t|t}|Y_{1:n}\big] \\
&= \text{var}\big[X_{t|t} + \Sigma_{t|t}A^T M_t(X_{t+1} - AX_{t|t})|Y_{1:n}\big] + \\
&\quad + \Sigma_{t|t} - \Sigma_{t|t}A^T M_t A\Sigma_{t|t} \\
&= \text{var}\big[(I_{d_x} - \Sigma_{t|t}A^T M_t A)X_{t|t} + \Sigma_{t|t}A^T M_t X_{t+1}|Y_{1:n}\big] \\
&= \text{var}\big[(I_{d_x} - \Sigma_{t|t}A^T M_t A)X_{t|t}|Y_{1:n}\big] + \\
&\quad + \text{var}\big[\Sigma_{t|t}A^T M_t X_{t+1}|Y_{1:n}\big] + \\
&\quad + 2\text{cov}\big[I_{d_x} - \Sigma_{t|t}A^T M_t A)X_{t|t}, \Sigma_{t|t}A^T M_t X_{t+1}|Y_{1:n}\big] + \\
&\quad + \Sigma_{t|t} - \Sigma_{t|t}A^T M_t A\Sigma_{t|t} \\
&= 0_{d_x \times d_x} + \Sigma_{t|t}A^T M_t \text{var}(X_{t+1}|Y_{1:n})M_t^T\Sigma_{t|t}A + 0_{d_x \times d_x} +
\end{aligned}$$

$$
\begin{aligned}
&+ \Sigma_{t|t} - \Sigma_{t|t} A^T M_t A \Sigma_{t|t} \\
&= \Sigma_{t|t} - \Sigma_{t|t} A^T M_t A \Sigma_{t|t} + \Sigma_{t|t} A^T M_t \Sigma_{t+1|n} M_t \Sigma_{t|t} A,
\end{aligned} \tag{B.19}
$$

where in the above derivation we have used that $\mathrm{var}(X_{t|t}|Y_{1:n}) = 0_{d_x \times d_x}$ (since $X_{t|t}$ is a constant with respect to $Y_{1:n}$) and by the same reason $\mathrm{cov}(X_{t|t}, X_{t+1}) = 0_{d_x \times d_x}$. We have also used that $M_t^T = M_t$ (i.e. that M_t is a symmetric matrix), which can be readily verified from its definition. The forward-backward smoothing procedure derived here is summarized in Algorithm B.3.

Algorithm B.3: Kalman Smoothing

Backward recursion
for *t = n-1* **to** *0* **do**
  compute $X_{t|n} = X_{t|t} + \Sigma_{t|t} A^T M_t (X_{t+1|n} - A X_{t|t})$
  compute $\Sigma_{t|n} = \Sigma_{t|t} - \Sigma_{t|t} A^T M_t A \Sigma_{t|t} + \Sigma_{t|t} A^T M_t \Sigma_{t+1|n} M_t \Sigma_{t|t} A$
end

B.5 Quadrature-based Estimates of Posterior Densities

Whenever we are dealing with linear and Gaussian HMMs with static parameters of a relatively small dimension d_θ, we might wish to approximate the posterior distribution of θ given $Y_{1:n} = y_{1:n}$ directly via numerical integration (also known as *quadrature*-based) methods (Asmussen and Glynn, 2007). This is done e.g. for the AR(1) + noise model in Section 4.2 to provide a "true" distribution for benchmarking our sequential parameter learning algorithms.

In order to compute a quadrature-based estimate of the posterior for a linear and Gaussian HMM (B.1-B.2), we first decompose

$$
p(\theta|y_{1:n}) = \frac{p(\theta, y_{1:n})}{p(y_{1:n})} = \frac{p(y_{1:n}|\theta)p(\theta)}{p(y_{1:n})} = \frac{p(y_{1:n}|\theta)p(\theta)}{\int_\Theta p(y_{1:n}|\theta)p(\theta)d\theta}, \tag{B.20}
$$

where the last equality follows from Bayes' theorem. Although in general we cannot compute the integral $\int_\Theta p(y_{1:n}|\theta)p(\theta)d\theta$ even under the linearity and normality assumptions, we can indeed compute the prior $p(\theta)$ (by assumption) and the likelihood $p(y_{1:n}|\theta)$.

To show how the likelihood $p(y_{1:n}|\theta)$ can be routinely obtained as a byproduct of the Kalman Filter (Algorithm B.1), recall from Section B.2 that the one-step-ahead observation prediction satifies

$$
\mathbb{E}(Y_t|Y_{1:t-1}) = B X_{t|t-1} \qquad \text{and} \qquad \Gamma_t := \mathrm{var}(Y_t|Y_{1:t-1}) = B \Sigma_{t|t-1} B^T + S S^T.
$$

Now, since Y_t is Gaussian for each t, we have that $Y_{1:t-1}$ and $Y_t|Y_{1:t-1}$ are also Gaussian (see Remark B.2.1). Therefore, the one-step-ahead observation predictive density is

$$
p(y_t|y_{1:t-1}, \theta) = d\mathcal{N}(y_t|B X_{t|t-1}, \Gamma_t). \tag{B.21}
$$

Finally, by (B.21) applying a predictive decomposition to $p(y_{1:n}|\theta)$ then yields

$$
p(y_{1:n}|\theta) = p(y_1|\theta) \prod_{t=2}^{n} p(y_t|y_{1:t-1})
$$

$$= d\mathcal{N}(y_1|BX_{1|0}, \Gamma_1) \prod_{t=2}^{n} d\mathcal{N}(y_t|BX_{t|t-1}, \Gamma_t)$$

$$= \prod_{t=1}^{n} d\mathcal{N}(y_t|BX_{t|t-1}, \Gamma_t) \tag{B.22}$$

Naturaly, the static parameters θ are typically included in model (B.1-B.2) as a part of the matrices/vectors A, B, R, S, X_ν and Σ_ν.

In possession of a pointwise estimate of the likelihood for any value of θ given the observations $Y_{1:n} = y_{1:n}$, we can also compute the integrand in $\int_\Theta p(y_{1:n}|\theta)p(\theta)d\theta$ for any value of θ. This means that we can approximate this integral with

$$\hat{p}(y_{1:n}) := \sum_{j_1=1}^{J_1} \cdots \sum_{j_{d_\theta}=1}^{J_{d_\theta}} p(y_{1:n}|\theta_{1,j_1}, \ldots, \theta_{d_\theta,j_{d_\theta}}) \cdot \\ \cdot p(\theta_{1,j_1}, \ldots, \theta_{d_\theta,j_{d_\theta}}) \Delta\theta_{1,j_1} \cdots \Delta\theta_{d_\theta,j_{d_\theta}}, \tag{B.23}$$

where in the expression above we sum over all values j_k of the kth component of θ (denoted θ_{k,j_k}), $j_k = 1, \ldots, J_k$ and $k = 1, \ldots, d_\theta$. Although there are several quadrature rules we can choose from to specify the values θ_{k,j_k} and increments $\Delta\theta_{k,j_k}$ (Asmussen and Glynn, 2007), the simplest and most popular method is to choose an *equispaced grid* and the *midpoint rule*, i.e. to let $\theta_{k,1} = a_k$, $\theta_{k,J_k} = b_k$ for $a_k < b_k$ and take $\theta_{k,j_l} = a_k + (b_k - a_k) \cdot l/J_k$ and $\Delta\theta_{k,j_l} = \theta_{k,j_l} - \theta_{k,j_{l-1}}$, $l = 2, \ldots, J_{k-1}$. This corresponds to a certain type of *Riemannian sum*, which under general conditions can be shown to converge to $p(y_{1:n})$ as each $a_k \to -\infty$, $b_k \to \infty$ and $J_k \to +\infty$; see e.g. Bartle (1976).

In closing, our quadrature-based pointwise estimate of the posterior $p(\theta|y_{1:n})$ is thus given by

$$\hat{p}(\theta|y_{1:n}) := \frac{p(y_{1:n}|\theta)p(\theta)}{\hat{p}(y_{1:n})} \tag{B.24}$$

for each value of θ, with $\hat{p}(y_{1:n})$ defined in (B.23). Note that for numerical stability we usually work with the estimate of the log-posterior, $\log \hat{p}(\theta|y_{1:n})$, although in this case we additionally have to deal with a log-sum of exponentials (see Section E.3). Typically, we evaluate the posterior (B.24) over the same interval used for computing (B.23).

B.6 Optimal Proposal Distributions

We now discuss how to compute the optimal importance weights $p(y_t|x_{t-1})$ and optimal state proposal distribution $p(x_t|x_{t-1}, y_t)$ in a scalar-valued linear Gaussian HMM.

Proposition B.6.1. *Let $(X_t, Y_t)_{t\geq 0}$ be an HMM defined by*

$$G_t = F_t + \tau U_t, \qquad \eta_t \sim N(0, \tau^2),$$
$$Y_t = G_t + \sigma V_t, \qquad \epsilon_t \sim N(0, \sigma^2),$$

where $G_t := G(X_t)$ and $F_t := F(X_{t-1})$ are (possibly nonlinear) functions of X_t (G is also assumed to be invertible), $X_0 \perp\!\!\!\perp (\epsilon_t)_{t\geq 0} \perp\!\!\!\perp (\eta_t)_{t\geq 0}$, $d_x = d_y = 1$ and $G(\mathcal{X}) = \mathbb{R}$. Then

$$p(y_t|x_{t-1}) = d\mathcal{N}(y_t|F_t, \sigma^2 + \tau^2). \tag{B.25}$$

and

$$p(x_t|x_{t-1}, y_t) = d\mathcal{N}\left(G^{-1}(x_t)\middle|\frac{\sigma^2 F_t + \tau^2 y_t}{\sigma^2+\tau^2}, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right) \cdot \left|\frac{dG(x)}{dx}\right|^{-1}_{x=G^{-1}(x_t)}, \tag{B.26}$$

where G^{-1} denotes the inverse of G and $|dG(x)/dx|^{-1}_{x=G^{-1}(x_t)}$ is the Jacobian of the transformation of $X_t \mapsto G(X_t)$ evaluated at the point $G^{-1}(x_t)$.

Proof. We start with the optimal weights. Since $G(\mathcal{X}) = \mathbb{R}$, they are given by

$$\begin{aligned}
p(y_t|x_{t-1}) &= \int_{G(\mathcal{X})} p(y_t, G_t|x_{t-1})dG_t \\
&= \int_{-\infty}^{+\infty} p(y_t|G_t, x_{t-1})p(G_t|x_{t-1})dG_t \\
&= \int_{-\infty}^{+\infty} g(y_t|G_t)f(G_t|x_{t-1})dG_t \\
&= \int_{-\infty}^{+\infty} f(G_t|x_{t-1})g(y_t|G_t)dG_t \\
&= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\tau^2}}\exp\left\{-\frac{1}{2\tau^2}(G_t-F_t)^2\right\}\cdot\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(y_t-G_t)^2\right\}dG_t \\
&= \frac{1}{2\pi\sqrt{\sigma^2\tau^2}}\int_{-\infty}^{+\infty}\exp\left\{-\frac{1}{2}\left[\frac{G_t^2-2G_tF_t+F_t^2}{\tau^2}+\frac{y_t^2-2G_ty_t+G_t^2}{\sigma^2}\right]\right\}dG_t \\
&= \frac{1}{2\pi\sqrt{\sigma^2\tau^2}}\exp\left\{-\frac{1}{2}\left[\frac{F_t^2}{\tau^2}+\frac{y_t^2}{\sigma^2}\right]\right\}\cdot \\
&\qquad\cdot\int_{-\infty}^{+\infty}\exp\left\{-\frac{1}{2}\left[G_t^2\left(\frac{1}{\tau^2}+\frac{1}{\sigma^2}\right)-2G_t\left(\frac{F_t}{\tau^2}+\frac{y_t}{\sigma^2}\right)\right]\right\}dG_t \\
&= \frac{1}{2\pi\sqrt{\sigma^2\tau^2}}\exp\left\{-\frac{1}{2}\frac{1}{\sigma^2\tau^2}(\sigma^2F_t^2+\tau^2y_t^2)\right\}\cdot \\
&\qquad\cdot\int_{-\infty}^{+\infty}\exp\left\{-\frac{1}{2}\frac{1}{\sigma^2\tau^2}\left[G_t^2(\sigma^2+\tau^2)-2G_t(\sigma^2F_t+\tau^2y_t)\right]\right\}dG_t \\
&= \frac{1}{2\pi\sqrt{\sigma^2\tau^2}}\exp\left\{-\frac{1}{2}\frac{\sigma^2+\tau^2}{\sigma^2\tau^2}\left(\frac{\sigma^2F_t^2+\tau^2y_t^2}{\sigma^2+\tau^2}\right)\right\}\cdot \\
&\qquad\cdot\int_{-\infty}^{+\infty}\exp\left\{-\frac{1}{2}\frac{\sigma^2+\tau^2}{\sigma^2\tau^2}\left[G_t^2-2G_t\left(\frac{\sigma^2F_t+\tau^2y_t}{\sigma^2+\tau^2}\right)+\right.\right. \\
&\qquad\qquad\left.\left.+\left(\frac{\sigma^2F_t+\tau^2y_t}{\sigma^2+\tau^2}\right)^2-\left(\frac{\sigma^2F_t+\tau^2y_t}{\sigma^2+\tau^2}\right)^2\right]\right\}dG_t \\
&= \frac{1}{2\pi\sqrt{\sigma^2\tau^2}}\exp\left\{-\frac{1}{2}\frac{\sigma^2+\tau^2}{\sigma^2\tau^2}\left[\left(\frac{\sigma^2F_t^2+\tau^2y_t^2}{\sigma^2+\tau^2}\right)-\left(\frac{\sigma^2F_t+\tau^2y_t}{\sigma^2+\tau^2}\right)^2\right]\right\}\cdot \\
&\qquad\cdot\int_{-\infty}^{+\infty}\exp\left\{-\frac{1}{2}\frac{\sigma^2+\tau^2}{\sigma^2\tau^2}\left[G_t-\left(\frac{\sigma^2F_t+\tau^2y_t}{\sigma^2+\tau^2}\right)\right]^2\right\}dG_t,
\end{aligned}$$

where from the second to the third equations we have applied (1.2) and (1.1). In the last equation, the integral is equal to $\sqrt{2\pi\sigma^2\tau^2/(\sigma^2+\tau^2)}$, which is the proportionality constant for a normally distributed random variable with mean $(\sigma^2F_t+\tau^2y_t)/(\sigma^2+\tau^2)$ and

variance $(\sigma^2\tau^2)/(\sigma^2+\tau^2)$. On the other hand, the difference inside the first exponential is equal to

$$\begin{aligned}
&\frac{(\sigma^2F_t^2+\tau^2y_t^2)(\sigma^2+\tau^2)-(\sigma^2F_t+\tau^2y_t)}{(\sigma^2+\tau^2)^2} = \\
&\qquad = \frac{\sigma^4F_t^2+\sigma^2\tau^2F_t^2+\tau^4y_t^2+\sigma^2\tau^2y_t^2-\sigma^4F_t^2-2\sigma^2\tau^2F_ty_t-\tau^4y_t^2}{(\sigma^2+\tau^2)^2} \\
&\qquad = \frac{\sigma^2\tau^2F_t^2-2\sigma^2\tau^2F_ty_t+\sigma^2\tau^2y_t^2}{(\sigma^2+\tau^2)^2} \\
&\qquad = (\sigma^2\tau^2)\frac{(y_t-F_t)^2}{(\sigma^2+\tau^2)^2}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
p(y_t|x_{t-1}) &= \frac{1}{2\pi\sqrt{\sigma^2\tau^2}}\exp\left\{-\frac{1}{2}\frac{\sigma^2+\tau^2}{\sigma^2\tau^2}(\sigma^2\tau^2)\frac{(y_t-F_t)^2}{(\sigma^2+\tau^2)^2}\right\}\cdot\sqrt{\frac{2\pi\sigma^2\tau^2}{\sigma^2+\tau^2}} \\
&= \frac{1}{\sqrt{2\pi(\sigma^2+\tau^2)}}\exp\left\{-\frac{1}{2}\frac{1}{\sigma^2+\tau^2}(y_t-F_t)^2\right\},
\end{aligned}$$

and $Y_t|X_{t-1}$ is Gaussian with mean F_t and variance $\sigma^2+\tau^2$, i.e.

$$p(y_t|x_{t-1}) = d\mathcal{N}(y_t|F_t,\sigma^2+\tau^2). \tag{B.27}$$

We now move on to compute the optimal state proposal distribution. In Proposition 2.1.1 we have established that

$$p(G_t|x_{t-1},y_t) = \frac{f(G_t|x_{t-1})g(y_t|x_t)}{p(y_t|x_{t-1})} \propto f(G_t|x_{t-1})g(y_t|x_t).$$

Now, in the derivation of (B.27) we have shown that the product $f(G_t|x_{t-1})\cdot g(y_t|G_t)$ is proportional to

$$\exp\left\{-\frac{1}{2}\frac{\sigma^2+\tau^2}{\sigma^2\tau^2}\left[G_t-\left(\frac{\sigma^2F_t+\tau^2y_t}{\sigma^2+\tau^2}\right)\right]^2\right\}.$$

Therefore, $G_t|(X_{t-1},Y_t)$ is Gaussian with mean $(\sigma^2F_t+\tau^2y_t)/(\sigma^2+\tau^2)$ and variance $(\sigma^2\tau^2)/(\sigma^2+\tau^2)$, i.e.

$$p(G_t|x_{t-1},y_t) = d\mathcal{N}\left(G_t\middle|\frac{\sigma^2F_t+\tau^2y_t}{\sigma^2+\tau^2},\frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right). \tag{B.28}$$

As for the distribution of $X_t|(X_{t-1},Y_t)$, since $X_t = G(G^{-1}(X_t))$, we simply have to transform $p(G_t|x_{t-1},y_t)$ in (B.28) accordingly, i.e.

$$\begin{aligned}
p(x_t|x_{t-1},y_t) &= p(G^{-1}(x_t)|x_{t-1},y_t)\cdot\left|\frac{dG(x)}{dx}\right|^{-1}_{x=G^{-1}(x_t)} \\
&= d\mathcal{N}\left(G^{-1}(x_t)\middle|\frac{\sigma^2F_t+\tau^2y_t}{\sigma^2+\tau^2},\frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right)\cdot\left|\frac{dG(x)}{dx}\right|^{-1}_{x=G^{-1}(x_t)}.
\end{aligned} \tag{B.29}$$

□

Appendix C

Useful Properties of Conditional Expectations

In this appendix we present some properties of conditional expectations that are useful in the main presentation of the text. As stated at the end of Chapter 1, we will avoid relying on the general characterization of conditional expectations (Shiryaev, 1995; Shao, 2003) in order to facilitate the exposition. In practice, this means that we always assume that in a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$ the *conditional probability* density (with respect to suitable measures dx and dy dominating $\mathbb{P}$) of a random variable $X \in \mathcal{X}$ given another random variable $Y \in \mathcal{Y}$ is always well-defined, and is given by

$$p(x|y) := \frac{p(x,y)}{p(y)}.$$

Similarly, the *conditional expectation* of X given Y is also always assumed to be well-defined (as long as $\mathbb{E}_{\mathbb{P}}(|X|) < +\infty$, where $\mathbb{E}_{\mathbb{P}}$ denotes expectation with respect to $\mathbb{P}$), and is given by

$$\mathbb{E}_{\mathbb{P}}(X|Y) := \int_{\mathcal{X}} x \cdot p(x|y)dx.$$

A property of conditional expectations that we often use here is that $\mathbb{E}_{\mathbb{P}}(X|X) = X$. In terms of conditional probabilities, the analogous property is

$$p(x|x) = \frac{p(x,x)}{p(x)} = \frac{p(x)}{p(x)} = \delta_X(dx). \tag{C.1}$$

Instances of where we implicitly use this property are in e.g. the proofs of Theorem 2.1.1 and Proposition 2.1.1.

As an example, consider computing $\mathbb{E}_{\mathbb{P}}[h(X,Y)|Y]$ for a $\mathbb{P}$-integrable and $\mathcal{B}$-measurable function h. It is intuitive that the density with respect to which we have to integrate is $p(x|y)$, but this can be formally shown using (C.1), i.e.

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}}[h(X,Y)|Y] &= \int_{\mathcal{X}\times\mathcal{Y}} h(x,y)p(x,y|y)dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} h(x,y)p(x|y)p(y|y)dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} h(x,y)p(x|y)\delta_Y(dy)dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} h(x,Y)p(x|Y)dx.
\end{aligned}$$

C.1 Total Expectation and Variance

In this section we prove two results frequently known as *Law of Total Expectation* (sometimes *Law of Iterated Expectations*) and *Law of Total Variance*. Again, see e.g. Shiryaev (1995) and Shao (2003) for their statements and proofs in the general case.

Proposition C.1.1 (Law of Total Expectation). *Let X and Y be random variables defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that $\mathbb{E}_{\mathbb{P}}(|X|) < +\infty$. Then*

$$\mathbb{E}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)] = \mathbb{E}_{\mathbb{P}}(X). \tag{C.2}$$

Proof. The definition of $\mathbb{E}_{\mathbb{P}}[h(Y)]$ for a $\mathbb{P}$-integrable and $\mathcal{B}$-measurable function h requires that

$$\mathbb{E}_{\mathbb{P}}[h(Y)] := \int_{\mathcal{Y}} h(y)p(y)dy.$$

Therefore, by taking $h = \mathbb{E}_{\mathbb{P}}(X|Y)$ (which by assumption is a $\mathcal{B}$-measurable and $\mathbb{P}$-integrable function of Y), we have

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)] &= \int_{\mathcal{Y}} \left[\int_{\mathcal{X}} x \cdot p(x|y)dx \right] p(y)dy \\
&= \int_{\mathcal{X}} \int_{\mathcal{Y}} x \frac{p(x,y)}{p(y)} p(y)dydx \\
&= \int_{\mathcal{X}} \int_{\mathcal{Y}} x \cdot p(x,y)dydx \\
&= \int_{\mathcal{X}} x \cdot p(x)dx \\
&= \mathbb{E}_{\mathbb{P}}(X).
\end{aligned}$$

The change in integration order made in the derivation above is justified by Fubini's theorem (integrals of probability densities are always finite, and by assumption $\mathbb{E}_{\mathbb{P}}(|X|) < +\infty$). Note that we implicitly have used that the integral over $\mathcal{Y}$ of the joint density $p(x, y)$ is equal to the marginal $p(x)$, i.e. $\int_Y p(x, y)dy = p(x)$.

□

Proposition C.1.2 (Law of Total Variance). *Let X and Y be random variables defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that $\mathbb{E}_{\mathbb{P}}[|h(Y)h(Y)^T|] < +\infty$ and $\mathbb{E}_{\mathbb{P}}[|XX^T|] < +\infty$, where A^T denotes the transpose of matrix A.. Then*

$$\mathrm{var}_{\mathbb{P}}(X) = \mathrm{var}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)] + \mathbb{E}_{\mathbb{P}}[\mathrm{var}_{\mathbb{P}}(X|Y)]. \tag{C.3}$$

Proof. For a $\mathbb{P}$-integrable and $\mathcal{B}$-measurable function h, the variance of $h(Y)$ under $\mathbb{P}$ is defined by

$$\mathrm{var}_{\mathbb{P}}[h(Y)] := \mathbb{E}_{\mathbb{P}} \left\{ \left[h(Y) - \mathbb{E}_{\mathbb{P}}[h(Y)] \right] \left[h(Y) - \mathbb{E}_{\mathbb{P}}[h(Y)] \right]^T \right\},$$

which we can decompose as

$$\begin{aligned}
\mathrm{var}_{\mathbb{P}}[h(Y)] &= \mathbb{E}_{\mathbb{P}} \left\{ \left[h(Y) - \mathbb{E}_{\mathbb{P}}[h(Y)] \right] \left[h(Y) - \mathbb{E}_{\mathbb{P}}[h(Y)] \right]^T \right\} \\
&= \mathbb{E}_{\mathbb{P}} \left\{ h(Y)h(Y)^T - h(Y)\mathbb{E}_{\mathbb{P}}[h(Y)]^T - \mathbb{E}_{\mathbb{P}}[h(Y)]h(Y)^T + \mathbb{E}_{\mathbb{P}}[h(Y)]\mathbb{E}_{\mathbb{P}}[h(Y)]^T \right\}
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_{\mathbb{P}}\left\{h(Y)h(Y)^T\right\} - \mathbb{E}_{\mathbb{P}}[h(Y)]\mathbb{E}_{\mathbb{P}}[h(Y)]^T + \\
&\qquad - \mathbb{E}_{\mathbb{P}}[h(Y)]\mathbb{E}_{\mathbb{P}}[h(Y)]^T + \mathbb{E}_{\mathbb{P}}[h(Y)]\mathbb{E}_{\mathbb{P}}[h(Y)]^T \\
&= \mathbb{E}_{\mathbb{P}}\left\{h(Y)h(Y)^T\right\} - \mathbb{E}_{\mathbb{P}}[h(Y)]\mathbb{E}_{\mathbb{P}}[h(Y)]^T.
\end{aligned}
$$

This means that by taking $h(Y) = \mathbb{E}_{\mathbb{P}}(X|Y)$ we then have

$$
\mathrm{var}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)] = \mathbb{E}_{\mathbb{P}}\left\{\mathbb{E}_{\mathbb{P}}(X|Y)\mathbb{E}_{\mathbb{P}}(X|Y)^T\right\} - \{\mathbb{E}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)]\}\{\mathbb{E}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)]^T\}
$$

and, since by Proposition C.1.1 follows that $\mathbb{E}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)] = \mathbb{E}_{\mathbb{P}}(X)$, this is equivalent to

$$
\mathrm{var}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)] = \mathbb{E}_{\mathbb{P}}\left\{\mathbb{E}_{\mathbb{P}}(X|Y)\mathbb{E}_{\mathbb{P}}(X|Y)^T\right\} - \mathbb{E}_{\mathbb{P}}(X)\mathbb{E}_{\mathbb{P}}(X)^T.
$$

On the other hand, again applying Proposition C.1.1 we can write the expectation $\mathbb{E}_{\mathbb{P}}[\mathrm{var}_{\mathbb{P}}(X|Y)]$ as

$$
\begin{aligned}
\mathbb{E}_{\mathbb{P}}[\mathrm{var}_{\mathbb{P}}(X|Y)] &= \mathbb{E}_{\mathbb{P}}\left\{\mathbb{E}_{\mathbb{P}}[XX^T|Y] - \mathbb{E}_{\mathbb{P}}(X|Y)\mathbb{E}_{\mathbb{P}}(X|Y)^T\right\} \\
&= \mathbb{E}_{\mathbb{P}}(XX^T) - \mathbb{E}_{\mathbb{P}}\left\{\mathbb{E}_{\mathbb{P}}(X|Y)\mathbb{E}_{\mathbb{P}}(X|Y)^T\right\}.
\end{aligned}
$$

Finally, using these results we can write the variance of X under $\mathbb{P}$ as

$$
\begin{aligned}
\mathrm{var}_{\mathbb{P}}(X) &= \mathbb{E}_{\mathbb{P}}(XX^T) - \mathbb{E}_{\mathbb{P}}(X)\mathbb{E}_{\mathbb{P}}(X)^T \\
&= \mathbb{E}_{\mathbb{P}}(XX^T) - \mathbb{E}_{\mathbb{P}}\left\{\mathbb{E}_{\mathbb{P}}(X|Y)\mathbb{E}_{\mathbb{P}}(X|Y)^T\right\} + \\
&\qquad + \mathbb{E}_{\mathbb{P}}\left\{\mathbb{E}_{\mathbb{P}}(X|Y)\mathbb{E}_{\mathbb{P}}(X|Y)^T\right\} - \mathbb{E}_{\mathbb{P}}(X)\mathbb{E}_{\mathbb{P}}(X)^T \\
&= \mathrm{var}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)] + \mathbb{E}_{\mathbb{P}}[\mathrm{var}_{\mathbb{P}}(X|Y)],
\end{aligned}
$$

as required.

□

Corollary C.1.1 (Rao-Blackwell Inequality). *Assume the conditions of Proposition C.1.2. Then*

$$
\mathrm{var}_{\mathbb{P}}(X) \geq \mathrm{var}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)]. \tag{C.4}
$$

Proof. By Proposition C.1.2, we have

$$
\mathrm{var}_{\mathbb{P}}(X) = \mathrm{var}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}(X|Y)] + \mathbb{E}_{\mathbb{P}}[\mathrm{var}_{\mathbb{P}}(X|Y)]
$$

and, since $\mathbb{E}_{\mathbb{P}}[\mathrm{var}_{\mathbb{P}}(X|Y)]$ is an almost surely-$\mathbb{P}$ nonnegative variable, the inequality follows directly.

□

Appendix D

Likelihood and Regularization Functional Estimators

In this appendix we derive an estimator for a general regularization functional and show that expressions (3.29) and (3.47) are particular instances of this estimator. The functional itself might be seen as an extension of (3.26) for the case in which interest lies in regularizing the entire past trajectories $(X_{0:t-1}, \theta_{0:t-1})$ instead of only (X_{t-1}, θ_{t-1}), and is defined by

$$\begin{aligned} R(\mathcal{X}^t, \Theta^t | y_{1:t-1}) &:= \int_{\mathcal{X}^t \times \Theta^t} \frac{p(y_{t-1}|x_{0:t-1}, \theta_{0:t-1}, y_{1:t-2})}{p(y_{t-1}|y_{1:t-2})} \cdot \\ &\qquad \cdot p(x_{0:t-1}, \theta_{0:t-1}|y_{t-1}, y_{1:t-2}) dx_{0:t-1} d\theta_{0:t-1} \\ &= \int_{\mathcal{X}^t \times \Theta^t} \frac{g(y_{t-1}|x_{t-1}, \theta_{t-1})}{p(y_{t-1}|y_{1:t-2})} p(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1}) dx_{0:t-1} d\theta_{0:t-1}, \end{aligned} \tag{D.1}$$

where from the first to the second line we have used that $p(y_{t-1}|x_{0:t-1}, \theta_{0:t-1}, y_{1:t-2}) = g(y_{t-1}|x_{t-1}, \theta_{t-1})$, implied from (1.2).

Naturally, the most common way to estimate (D.1) is by replacing $p(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1})$ with its particle approximation $\hat{p}(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1})$, yielding

$$\begin{aligned} \hat{R}(\mathcal{X}^t \times \Theta^t | y_{1:t-1}) &:= \int_{\mathcal{X}^t \times \Theta^t} \frac{g(y_{t-1}|x_{t-1}, \theta_{t-1})}{p(y_{t-1}|y_{1:t-2})} \hat{p}(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1}) dx_{0:t-1} d\theta_{0:t-1} \\ &= \int_{\mathcal{X}^t \times \Theta^t} \frac{g(y_{t-1}|x_{t-1}, \theta_{t-1})}{p(y_{t-1}|y_{1:t-2})} \sum_{i=1}^{N} w_{t-1}^i \delta_{x_{0:t-1}^i, \theta_{0:t-1}^i}(dx_{0:t-1} d\theta_{0:t-1}) dx_{0:t-1} d\theta_{0:t-1} \\ &= \frac{1}{p(y_{t-1}|y_{1:t-2})} \sum_{i=1}^{N} w_{t-1}^i g(y_{t-1}|x_{t-1}^i, \theta_{t-1}^i). \end{aligned} \tag{D.2}$$

However, since $p(y_{t-1}|y_{1:t-2})$ is in general not available in closed form, we need to make an additional approximation by replacing $p(y_{t-1}|y_{1:t-2})$ with an estimator $\hat{p}(y_{t-1}|y_{1:t-2})$ in (D.2). The estimator we adopt here is an extension of the likelihood estimator[1] proposed by Pitt et al. (2012) in the context of our sequential learning framework.

[1] Here we refer to the estimator of $p(y_{t-1}|y_{1:t-2})$ as a "likelihood estimator" due to the fact that the likelihood $p(y_{1:t})$ admits the decomposition $p(y_{1:t}) = p(y_1) \prod_{k=2}^{t} p(y_k|y_{1:k-1})$, which can therefore be estimated by simply replacing each $p(y_k|y_{1:k-1})$ with its corresponding estimate $\hat{p}(y_k|y_{1:k-1})$.

More specifically, we have[2]

$$p(y_t|y_{1:t-1}) = \int_{\mathcal{X}^{t+1}\times\{1,\dots,N\}\times\Theta^{t+1}} p(y_t, x_{0:t}, k, \theta_{0:t}|y_{1:t-1})dx_{0:t}dkd\theta_{0:t}$$
$$= \int_{\mathcal{X}^{t+1}\times\{1,\dots,N\}\times\Theta^{t+1}} p(\theta_t|x_t, x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t})p(y_t|x_{0:t}, k, \theta_{0:t-1}, y_{1:t-1})\cdot$$
$$\cdot\, p(x_t|x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t-1})p(x_{0:t-1}, k, \theta_{0:t-1}|y_{1:t-1})dx_{0:t}dkd\theta_{0:t}, \quad \text{(D.3)}$$

As before, we can obtain an estimator $\hat{p}(y_t|y_{1:t-1})$ of $p(y_t|y_{1:t-1})$ by simply replacing $p(x_{0:t-1}, k, \theta_{0:t-1}|y_{1:t-1})$ in (D.3) with its particle approximation $\hat{p}(x_{0:t-1}, k, \theta_{0:t-1}|y_{1:t-1})$, given by

$$\hat{p}(x_{0:t-1}, k, \theta_{0:t-1}|y_{1:t-1}) := \sum_{i=1}^{N} w_{t-1}^{k_i}\delta_{(x_{0:t-1}^i, k_i, \theta_{0:t-1}^i)}(dx_{0:t-1}dkd\theta_{0:t-1}).$$

Since from (1.2) and item (ii) of Proposition 1.1.1 we also have $p(y_t|x_{0:t}, k, \theta_{0:t-1}, y_{1:t-1})$ $= g(y_t|x_t, \tilde{\theta}_{t-1})$ and $p(x_t|x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t-1}) = f(x_t|\tilde{x}_{t-1}, \tilde{\theta}_{t-1})$, this yields

$$\hat{p}(y_t|y_{1:t-1}) := \int_{\mathcal{X}^{t+1}\times\{1,\dots,N\}\times\Theta^{t+1}} p(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})g(y_t|x_t, \tilde{\theta}_{t-1})\cdot$$
$$\cdot\, f(x_t|\tilde{x}_{t-1}, \tilde{\theta}_{t-1})\hat{p}(x_{0:t-1}, k, \theta_{0:t-1}|y_{1:t-1})dx_{0:t}dkd\theta_{0:t}$$
$$= \int_{\mathcal{X}^{t+1}\times\{1,\dots,N\}\times\Theta^{t+1}} f(x_t|\tilde{x}_{t-1}, \tilde{\theta}_{t-1})g(y_t|x_t, \tilde{\theta}_{t-1})p(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})\cdot$$
$$\cdot \sum_{i=1}^{N} w_{t-1}^{k_i}\delta_{(x_{0:t-1}^i, k_i, \theta_{0:t-1}^i)}(dx_{0:t-1}dkd\theta_{0:t-1})dx_{0:t}dkd\theta_{0:t}. \quad \text{(D.4)}$$

Now, let $\pi_{w,t}^i \equiv \pi_w(x_{0:t}^i, k_i, \theta_{0:t}^i, y_{1:t})$ and $\pi_{\lambda,t}^{k_i} \equiv \pi_\lambda(x_{0:t-1}^i, k_i, \theta_{0:t-1}^i, y_{1:t})$ respectively denote the *unnormalized importance weights* and *unnormalized intermediate weights* at time t. As their own names imply, these quantities are the terms that we sum in order to obtain the (thus *normalized*) corresponding importance weights w_t^i and intermediate weights λ_t^i that then sum to one across $i = 1, \dots, N$. That is, they satisfy

$$w_t^i = \frac{\pi_{w,t}^i}{\sum_{j=1}^N \pi_{w,t}^j} \qquad \text{and} \qquad \lambda_t^i = \frac{\pi_{\lambda,t}^i}{\sum_{j=1}^N \pi_{\lambda,t}^j}, \quad \text{(D.5)}$$

and from relation (D.5) and the recursion (3.14), an explicit expression for the unnormalized weights $\pi_{w,t}^i$ is given by

$$\pi_{w,t}^i = \frac{w_{t-1}^{k_i}}{\pi_{\lambda,t}^{k_i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})} \frac{p(\theta_t^i|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})}{q(\theta_t^i|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})}. \quad \text{(D.6)}$$

For clarity, it is useful to express $w_{t-1}^{k_i} = w(x_{0:t-1}^i, k_i, \theta_{0:t-1}^i, y_{1:t-1})$, which due to the point masses $\delta_{(x_{0:t-1}^i, k_i, \theta_{0:t-1}^i)}(dx_{0:t-1}dkd\theta_{0:t-1})$ in (D.4) is equivalent to $w(x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t-1})$ prior to integration. Along with (D.6), this allows us to write (D.4) as

$$\hat{p}(y_t|y_{1:t-1}) = \int_{\mathcal{X}^{t+1}\times\{1,\dots,N\}\times\Theta^{t+1}} f(x_t|\tilde{x}_{t-1}, \tilde{\theta}_{t-1})g(y_t|x_t, \tilde{\theta}_{t-1})p(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})\cdot$$

[2]Here, dk is understood as a measure (e.g. the counting measure) dominating the marginal probability measure associated with the auxiliary variable $k \in \{1, \dots, N\}$.

$$\cdot \sum_{i=1}^{N} w(x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t-1}) \delta_{(x^i_{0:t-1}, k_i, \theta^i_{0:t-1})}(dx_{0:t-1} dk d\theta_{0:t-1}) dx_{0:t} dk d\theta_{0:t}$$

$$= \int_{\mathcal{X}^{t+1} \times \{1,\ldots,N\} \times \Theta^{t+1}} w(x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t-1}) f(x_t | \tilde{x}_{t-1}, \tilde{\theta}_{t-1}) g(y_t | x_t, \tilde{\theta}_{t-1}) \cdot$$

$$\cdot p(\theta_t | x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t}) \cdot$$

$$\cdot \frac{\pi_\lambda(x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t})}{\pi_\lambda(x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t})} \frac{q(x_t | \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})}{q(x_t | \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})} \frac{q(\theta_t | x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})}{q(\theta_t | x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})} \cdot$$

$$\cdot \sum_{i=1}^{N} \delta_{(x^i_{0:t-1}, k_i, \theta^i_{0:t-1})}(dx_{0:t-1} dk d\theta_{0:t-1}) dx_{0:t} dk d\theta_{0:t}$$

and, by again using that $\pi_\lambda(x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t}) = \pi_\lambda(x^i_{0:t-1}, k_i, \theta^i_{0:t-1}, y_{1:t}) = \pi^{k_i}_{\lambda,t}$ inside the integral due to the point masses, we further have

$$\hat{p}(y_t | y_{1:t-1}) = \int_{\mathcal{X}^{t+1} \times \{1,\ldots,N\} \times \Theta^{t+1}} \frac{w(x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t-1})}{\pi_\lambda(x_{0:t-1}, k, \theta_{0:t-1}, y_{1:t})} \cdot$$

$$\cdot \frac{f(x_t | \tilde{x}_{t-1}, \tilde{\theta}_{t-1}) g(y_t | x_t, \tilde{\theta}_{t-1})}{q(x_t | \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})} \frac{p(\theta_t | x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})}{q(\theta_t | x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})} \cdot$$

$$\cdot q(x_t | \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t}) q(\theta_t | x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t}) \cdot$$

$$\cdot \sum_{i=1}^{N} \pi^{k_i}_{\lambda,t} \delta_{(x^i_{0:t-1}, k_i, \theta^i_{0:t-1})}(dx_{0:t-1} dk d\theta_{0:t-1}) dx_{0:t} dk d\theta_{0:t}. \tag{D.7}$$

Note at this point that the last term in (D.7) is proportional to the density from which we resample $(x^i_{0:t-1}, \theta^i_{0:t-1})$. Recall from Section 3.2.2 – more specifically equation (3.12) – that in resampling each $(x^{k_i}_{0:t-1}, \theta^{k_i}_{0:t-1})$ and k_i itself is selected with probability $\lambda^{k_i}_t := q(k_i | x^i_{0:t-1}, \theta^i_{0:t-1}, y_{1:t})$, satisfying $\lambda^{k_i}_t = \pi^{k_i}_{\lambda,t} / \sum_{j=1}^N \pi^j_{\lambda,t}$. That is, here we have

$$\hat{p}^*(x_{0:t-1}, k, \theta_{0:t-1} | y_{1:t}) := \sum_{i=1}^{N} \lambda^{k_i}_t \delta_{(x^i_{0:t-1}, k_i, \theta^i_{0:t-1})}(dx_{0:t-1} dk d\theta_{0:t-1}). \tag{D.8}$$

Since perfect draws can be produced from (D.8) through resampling, this means that we can make a further approximation by replacing (D.8) with

$$\sum_{i=1}^{N} \frac{1}{N} \delta_{(x^i_{0:t-1}, k_i, \theta^i_{0:t-1})}(dx_{0:t-1} dk d\theta_{0:t-1})$$

and, since the summation in the last line of (D.7) is given by

$$\sum_{i=1}^{N} \pi^{k_i}_{\lambda,t} \delta_{(x^i_{0:t-1}, k_i, \theta^i_{0:t-1})}(dx_{0:t-1} dk d\theta_{0:t-1}) =$$

$$= \sum_{i=1}^{N} \pi^{k_i}_{\lambda,t} \frac{\sum_{j=1}^N \pi^j_{\lambda,t}}{\sum_{j=1}^N \pi^j_{\lambda,t}} \delta_{(x^i_{0:t-1}, k_i, \theta^i_{0:t-1})}(dx_{0:t-1} dk d\theta_{0:t-1})$$

$$= \left\{ \sum_{j=1}^{N} \pi^j_{\lambda,t} \right\} \sum_{i=1}^{N} \lambda^{k_i}_t \delta_{(x^i_{0:t-1}, k_i, \theta^i_{0:t-1})}(dx_{0:t-1} dk d\theta_{0:t-1}),$$

this entire term can be approximated by

$$\left\{\sum_{j=1}^{N}\pi_{\lambda,t}^{j}\right\}\sum_{i=1}^{N}\frac{1}{N}\delta_{(x_{0:t-1}^{i},k_i,\theta_{0:t-1}^{i})}(dx_{0:t-1}dkd\theta_{0:t-1}).$$

Substituting this into (D.7) then gives

$$\begin{aligned}
\hat{p}(y_t|y_{1:t-1}) &= \int_{\mathcal{X}^{t+1}\times\{1,\ldots,N\}\times\Theta^{t+1}}\frac{w(x_{0:t-1},k,\theta_{0:t-1},y_{1:t-1})}{\pi_\lambda(x_{0:t-1},k,\theta_{0:t-1},y_{1:t})}\cdot\\
&\cdot\frac{f(x_t|\tilde{x}_{t-1},\tilde{\theta}_{t-1})g(y_t|x_t,\tilde{\theta}_{t-1})}{q(x_t|\tilde{x}_{0:t-1},\tilde{\theta}_{0:t-1},y_{1:t})}\frac{p(\theta_t|x_t,\tilde{x}_{0:t-1},\tilde{\theta}_{0:t-1},y_{1:t})}{q(\theta_t|x_t,\tilde{x}_{0:t-1},\tilde{\theta}_{0:t-1},y_{1:t})}\cdot\\
&\cdot q(x_t|\tilde{x}_{0:t-1},\tilde{\theta}_{0:t-1},y_{1:t})q(\theta_t|x_t,\tilde{x}_{0:t-1},\tilde{\theta}_{0:t-1},y_{1:t})\cdot\\
&\cdot\left\{\sum_{j=1}^{N}\pi_{\lambda,t}^{j}\right\}\sum_{i=1}^{N}\frac{1}{N}\delta_{(x_{0:t-1}^{i},k_i,\theta_{0:t-1}^{i})}(dx_{0:t-1}dkd\theta_{0:t-1})dx_{0:t}dkd\theta_{0:t}\\
&= \left\{\sum_{j=1}^{N}\pi_{\lambda,t}^{j}\right\}\sum_{i=1}^{N}\int_{\mathcal{X}^t\times\Theta^t}\frac{w(x_{0:t-1}^{i},k_i,\theta_{0:t-1}^{i},y_{1:t-1})}{\pi_\lambda(x_{0:t-1}^{i},k_i,\theta_{0:t-1}^{i},y_{1:t})}\cdot\\
&\cdot\frac{f(x_t|\tilde{x}_{t-1}^{i},\tilde{\theta}_{t-1}^{i})g(y_t|x_t,\tilde{\theta}_{t-1}^{i})}{q(x_t|\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t})}\frac{p(\theta_t|x_t,\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t})}{q(\theta_t|x_t,\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t})}\cdot\\
&\cdot q(x_t|\tilde{x}_{0:t-1},\tilde{\theta}_{0:t-1},y_{1:t})q(\theta_t|x_t,\tilde{x}_{0:t-1},\tilde{\theta}_{0:t-1},y_{1:t})dx_td\theta_t\\
&= \left\{\sum_{j=1}^{N}\pi_{\lambda,t}^{j}\right\}\cdot\\
&\cdot\sum_{i=1}^{N}\int_{\mathcal{X}^t\times\Theta^t}\pi_w(x_t,x_{0:t-1}^{i},k_i,\theta_t,\theta_{0:t-1}^{i},y_{1:t})q(x_t,\theta_t|\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t})dx_td\theta_t, \qquad \text{(D.9)}
\end{aligned}$$

where in the last line of (D.9) we note that the product of the marginal proposals $q(x_t|\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t})$ and $q(\theta_t|\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t})$ can be expressed as the joint proposal $q(x_t,\theta_t|\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t})$.

Now, in general the integral in (D.9) still has no analytic solution. We therefore make a final approximation by sampling the pair (x_t^i,θ_t^i) directly from the joint proposal $q(x_t,\theta_t|\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t})$ for each i and replacing this density in (D.9) with its (exact) Monte Carlo estimate

$$\hat{q}(x_t,\theta_t|\tilde{x}_{0:t-1}^{i},\tilde{\theta}_{0:t-1}^{i},y_{1:t}) := \frac{1}{N}\delta_{(x_t^i,\theta_t^i)}(dx_td\theta_t),$$

yielding, at last,

$$\begin{aligned}
\hat{p}(y_t|y_{1:t-1}) &= \left\{\sum_{j=1}^{N}\pi_{\lambda,t}^{j}\right\}\cdot\\
&\cdot\sum_{i=1}^{N}\int_{\mathcal{X}^t\times\Theta^t}\pi_w(x_t,x_{0:t-1}^{i},k_i,\theta_t,\theta_{0:t-1}^{i},y_{1:t})\frac{1}{N}\delta_{(x_t^i,\theta_t^i)}(dx_td\theta_t)dx_td\theta_t\\
&= \left\{\sum_{j=1}^{N}\pi_{\lambda,t}^{j}\right\}\sum_{i=1}^{N}\frac{\pi_w(x_t^i,x_{0:t-1}^{i},k_i,\theta_t^i,\theta_{0:t-1}^{i},y_{1:t})}{N}
\end{aligned}$$

$$= \left\{ \sum_{i=1}^{N} \frac{\pi_{w,t}^{i}}{N} \right\} \left\{ \sum_{i=1}^{N} \pi_{\lambda,t}^{i} \right\}. \tag{D.10}$$

As for the regularization functional estimator, replacing $p(y_{t-1}|y_{1:t-2})$ with the corresponding $\hat{p}(y_{t-1}|y_{1:t-2})$ obtained from (D.10) at time $t-1$ in (D.2) results in

$$\begin{aligned} \hat{R}(\mathcal{X}^t \times \Theta^t | y_{1:t-1}) &= \frac{\sum_{i=1}^{N} w_{t-1}^{i} g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})}{\hat{p}(y_{t-1}|y_{1:t-2})} \\ &= \frac{\sum_{i=1}^{N} w_{t-1}^{i} g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})}{\left\{ \sum_{i=1}^{N} \frac{\pi_{w,t-1}^{i}}{N} \right\} \left\{ \sum_{j=1}^{N} \pi_{\lambda,t-1}^{i} \right\}}. \end{aligned} \tag{D.11}$$

In closing, we can see how (D.11) generalizes the functional estimates of smooth jittering (3.29) and FALW (3.47) by making the appropriate substitutions. For smooth jittering, we have $\pi_{w,t-1}^{i} = g(y_{t-1}|x_{t-1}^{i}, \tilde{\theta}_{t-2}^{i})$, which is also equal to $g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})$ given that in this method $\theta_{t-1}^{i} = \tilde{\theta}_{t-2}^{i}$. This implies that $w_{t-1}^{i} = g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i}) / \sum_{j=1}^{N} g(y_{t-1}|x_{t-1}^{j}, \theta_{t-1}^{j})$ and $\pi_{\lambda,t-1}^{i} = w_{t-2}^{i}$, which in turn implies that $\sum_{i=1}^{N} \pi_{\lambda,t-1}^{i} = \sum_{i=1}^{N} w_{t-2}^{i} = 1$, from which (D.11) then becomes

$$\begin{aligned} \hat{R}(\mathcal{X}^t \times \Theta^t | y_{1:t-1}) &= \frac{\sum_{i=1}^{N} \frac{g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})}{\sum_{j=1}^{N} g(y_{t-1}|x_{t-1}^{j}, \theta_{t-1}^{j})} g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})}{\left\{ \sum_{i=1}^{N} \frac{g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})}{N} \right\} \left\{ \sum_{i=1}^{N} w_{t-2}^{i} \right\}} \\ &= N \sum_{i=1}^{N} \left[\frac{g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})}{\sum_{j=1}^{N} g(y_{t-1}|x_{t-1}^{j}, \theta_{t-1}^{j})} \right]^2. \end{aligned}$$

For FALW, we have $\pi_{w,t-1}^{i} = 1$, implying that $w_{t-1}^{i} = 1/N$ and that $\pi_{\lambda,t-1}^{i} = w_{t-2}^{i}\, p(y_{t-1}|x_{t-2}^{i}, \theta_{t-2}^{i})$. In this case, (D.11) then becomes

$$\hat{R}(\mathcal{X}^t \times \Theta^t | y_{1:t-1}) = \frac{\sum_{i=1}^{N} \frac{1}{N} g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})}{\left\{ \sum_{i=1}^{N} \frac{1}{N} \right\} \left\{ \sum_{i=1}^{N} \frac{1}{N} p(y_{t-1}|x_{t-2}^{i}, \theta_{t-2}^{i}) \right\}} = \sum_{i=1}^{N} \frac{g(y_{t-1}|x_{t-1}^{i}, \theta_{t-1}^{i})}{\sum_{j=1}^{N} p(y_{t-1}|x_{t-2}^{j}, \theta_{t-2}^{j})}.$$

Appendix E

Practical Implementation Notes

In this appendix we briefly describe some important details about the practical implementation of the algorithms described in this work. Often overlooked, these aspects are not only of theoretical interest in their own right, but also sometimes vital for a proper application of inference techniques for HMMs in practice.

E.1 Regularization of Constrained Parameters

Let $\theta \in \Theta$ be a parameter we want to regularize (see Section 3.2.3) such that $d_\theta := \dim(\theta)$. Since the regularization kernel K usually maps from $\mathbb{R}^{d_\theta}$ to $\mathbb{R}^{d_\theta}$, whenever Θ is a proper subset of $\mathbb{R}^{d_\theta}$ it might occur that the regularized parameter $\tilde{\theta} \sim K(\theta)$ might not be limited to Θ. Therefore, in order to avoid a rejection-type procedure (which might be inefficient, such as when θ occurs close to the border of Θ), we actually propose mapping

$$\eta := \psi(\theta), \qquad \psi(\Theta) = \mathbb{R}^{d_\theta}, \qquad \exists\, \psi^{-1} : \theta = \psi^{-1}(\eta), \psi^{-1}(\mathbb{R}^{d_\theta}) = \Theta.$$

Therefore, through the use of the invertible and $\mathcal{B}$-measurable function ψ, we can regularize $\tilde{\eta} \sim K(\eta) \in \mathbb{R}^{d_\theta}$ and then take $\tilde{\theta} = \psi^{-1}(\tilde{\eta})$, which is guaranteed to be constrained within Θ.

As an example, take ϕ and σ^2 in the AR(1) + noise example of Section 4.2. Here, $-1 < \phi < 1$ and $\sigma^2 > 0$, so that $\Theta = (-1, 1) \times \mathbb{R}^+$. By taking $\eta := (\eta_1, \eta_2) = \psi(\theta) = (\tanh^{-1}(\phi), \log(\sigma^2))$, we have $\eta \in \mathbb{R}^2$ and $\theta = \psi^{-1}(\eta) = (\tanh(\eta_1), \exp(\eta_2))$. A LW filter (see Section 3.2.4.2) regularization step here at time t therefore consists of drawing

$$\tilde{\eta}^i_{t-1} \sim \mathcal{N}(m^{k_i}_{t-1}, h^2 V_{t-1}),$$

and setting $\tilde{\theta}^i_{t-1} = \psi^{-1}(\tilde{\eta}^i_{t-1})$ for each i, where m^i_{t-1} and V_{t-1} defined in (3.19-3.20) are here taken as functions of $(\eta^i_{t-1})_{i=1}^N$ rather than $(\theta^i_{t-1})_{i=1}^N$.

E.2 pMCMC on Constrained Parameter Spaces

Now, consider the same setting of Section E.1 but suppose that instead of regularization we now want to perform inference for $\theta \in \Theta$ based on the particle MCMC of Section 3.1. Analogous to regularization, whenever we use a proposal distribution which is not constrained to Θ (such as e.g. Random Walk Metropolis proposals; see Section A.4), in order to avoid rejection sampling steps we also perform the MCMC moves on the

transformed parameters η and then obtain the original parameters through $\theta = \psi^{-1}(\eta)$. The difference here, however, lies in the fact that for pMCMC the acceptance probability is affected by the choice of parameter transformation ψ.

Recall from Section 3.1 that the acceptance probability for the proposed θ' given the current θ is

$$\alpha(\theta'|\theta) = 1 \wedge \frac{\hat{p}(y_{1:t}|\theta')p(\theta')}{\hat{p}(y_{1:t}|\theta)p(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)}. \tag{E.1}$$

Now, although we can easily define a proposal $q(\eta|\eta')$ acting on the space $H := \psi(\Theta)$, we usually only have priors for θ. We therefore need to take into account the transformation $\theta = \psi^{-1}(\eta)$, yielding the prior for η as

$$p(\eta) = p\big(\theta = \psi^{-1}(\eta)\big) \left| \frac{\partial \psi^{-1}(x)}{\partial x} \right|_{x=\psi^{-1}(\eta)},$$

where $|\partial \psi^{-1}(x)/\partial x|_{x=\psi^{-1}(\eta)}$ is the Jacobian of the transformation $\psi : \Theta \to \mathbb{R}^{d_\theta}$. However, since it is usually more convenient to evaluate the priors and likelihoods as a function of θ, we can use the Inverse Function Theorem (Rudin, 1976, pg. 221) to express the Jacobian as $|\partial \psi(x)/\partial x|^{-1}_{x=\psi(\theta)}$ and thus rewrite the acceptance probability (E.1) as

$$\alpha(\theta'|\theta) = 1 \wedge \frac{\hat{p}(y_{1:t}|\theta')p(\theta')}{\hat{p}(y_{1:t}|\theta)p(\theta)} \frac{\left[\left|\partial \psi(x)/\partial x\right|^{-1}_{x=\psi(\theta)}\right]}{\left[\left|\partial \psi(x)/\partial x\right|^{-1}_{x=\psi(\theta)}\right]} \frac{q(\eta|\eta')}{q(\eta'|\eta)}. \tag{E.2}$$

As an example, consider the Stochastic Volatility model of Section 4.4. Since here $\theta = (\phi, \tau^2, \sigma^2) \in (-1, 1) \times \mathbb{R}^+ \times \mathbb{R}^+$, we take $\psi(\theta) = (\tanh^{-1}(\phi), \log(\tau^2), \log(\sigma^2))$, with associated Jacobian

$$\left| \frac{\partial \psi(x)}{\partial x} \right|^{-1}_{x=\psi(\theta)} = \left| \begin{bmatrix} \frac{1}{1-\phi^2} & 0 & 0 \\ 0 & \frac{1}{\tau^2} & 0 \\ 0 & 0 & \frac{1}{\sigma^2} \end{bmatrix} \right|^{-1} = |(1-\phi^2) \cdot \tau^2 \cdot \sigma^2|.$$

Given that here the proposal to move from η to η' is

$$q(\eta'|\eta) = d\mathcal{N}(\eta, \Sigma)$$

for some covariance matrix Σ and that q is a symmetric function in (η', η), i.e. $q(\eta'|\eta) = q(\eta|\eta')$, the acceptance probability (E.2) in this case becomes

$$\begin{aligned} \alpha(\theta'|\theta) &= 1 \wedge \frac{\hat{p}(y_{1:t}|\theta')p(\theta')}{\hat{p}(y_{1:t}|\theta)p(\theta)} \frac{|[1-(\phi')^2] \cdot (\tau')^2 \cdot (\sigma')^2|}{|(1-\phi^2) \cdot \tau^2 \cdot \sigma^2|} \frac{q(\eta'|\eta)}{q(\eta'|\eta)} \\ &= 1 \wedge \frac{\hat{p}(y_{1:t}|\theta')p(\theta')}{\hat{p}(y_{1:t}|\theta)p(\theta)} \left| \frac{[1-(\phi')^2] \cdot (\tau')^2 \cdot (\sigma')^2}{(1-\phi^2) \cdot \tau^2 \cdot \sigma^2} \right|. \end{aligned}$$

E.3 Computing Log-sums of Exponentials

Now, consider the problem of computing

$$L(x) := \log \left(\sum_{j=1}^{N} \exp(x_j) \right), \tag{E.3}$$

where $x := (x_1, \ldots, x_N)$. This type of functional appears e.g. when computing importance weights in SMC methods or when computing quadrature-based estimates of log-posterior distributions (see Section B.5), and the main problem associated with it is that whenever some values x_j are large (in magnitude), there is overflow (if they are positive) or underflow (if they are negative). In order to improve upon this erratic numeric behavior, we will show in this section how we can compute $L(x)$ without having to evaluate such large terms.

First, let $m := \max(x_1, \ldots, x_N)$. Using standard properties of logarithms and exponentials, we can then rewrite (E.3) as

$$\begin{aligned} L(x) &= \log\left(\sum_{j=1}^{N} \exp(x_j)\right) \\ &= \log\left(\sum_{j=1}^{N} \frac{\exp(m)}{\exp(m)} \exp(x_j)\right) \\ &= \log\left(\exp(m) \sum_{j=1}^{N} \frac{\exp(x_j)}{\exp(m)}\right) \\ &= \log\big(\exp(m)\big) + \log\left(\sum_{j=1}^{N} \exp(x_j - m)\right) \\ &= m + \log\left(\sum_{j=1}^{N} \exp(x_j - m)\right). \end{aligned} \tag{E.4}$$

In (E.4), none of the evaluated terms in the sum is greater than 1, since by definition $\max_{1 \leq j \leq N} \big\{ \exp(x_j - m) \big\} = \exp(m - m) = \exp(0) = 1$, avoiding the numerical instability associated with computing $L(x)$ via (E.3).