Practical Filtering with Sequential Parameter Learning

Nicholas G. Polson †

Graduate School of Business, University of Chicago, Chicago, IL 60637, U.S.A.

Jonathan R. Stroud

Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, U.S.A.

Peter Müller

Department of Biostatistics, M. D. Anderson Cancer Center, Houston, TX 77030, U.S.A.

Summary. This paper develops a simulation-based approach to sequential parameter learning and filtering in general state-space models. Our approach is based on approximating the target posterior by a mixture of fixed-lag smoothing distributions. Parameter inference exploits a sufficient statistic structure and the methodology can be easily implemented by modifying state space smoothing algorithms. We avoid reweighting particles and hence sample degeneracy problems that plague particle filters that use sequential importance sampling. The method is illustrated using two examples: a benchmark autoregressive model with observation error and a high-dimensional dynamic spatio-temporal model. We show that the method provides accurate inference in the presence of outliers, model misspecification and high dimensionality.

Key Words: Filtering, Markov Chain Monte Carlo, Particle Filtering, Sequential Parameter Learning, Spatio-Temporal Models, State-Space Models.

1. Introduction

Modern-day applications of filtering in general state-space models often require sequential parameter learning. Typical examples include portfolio problems in finance and on-line disease monitoring in epidemiology. However, combined state and parameter learning presents many computational challenges. Standard particle filtering methods such as sampling-importance resampling (SIR, Gordon, Salmond and Smith, 1993; Liu and Chen, 1995; Kitagawa, 1996) and auxiliary particle filters (APF, Pitt and Shephard, 1999) can lead to unbalanced particle weights and degeneracy in the presence of outliers, model misspecification or high dimensionality. The degeneracy problem is exacerbated when sequential parameter learning is involved as this increases the effective dimension of the filtered state space (see Andrieu, Doucet and Tadić, 2005).

Many sequential parameter learning methods have been proposed within the particle filter framework. Gordon, Salmond and Smith (1993) augment the state vector to include static parameters; Berzuini, Best, Gilks and Larizza (1997) propose Markov chain Monte Carlo (MCMC) moves within the particle filter; Liu and West (2001) use kernel density estimation to approximate the parameter distribution; whereas Hürzeler and Künsch (2001) and Pitt (2002) provide off-line likelihood methods. Andrieu and Doucet (2003) and Andrieu, Doucet and Tadić (2005) consider recursive and batch maximum likelihood methods based on stochastic gradients and expectation-maximisation (EM) approaches. Del Moral, Doucet and Jasra (2006) propose a related approach using MCMC within a sequential Monte Carlo framework. Fearnhead (2002) and Storvik (2002) consider models with sufficient statistics for the parameters and apply particle filters to an augmented vector of states and sufficient statistics.

In this paper we propose a novel approach to Bayesian filtering and sequential parameter learning in general state-space models. Our approach relies on a rolling-window MCMC algorithm that approximates the target posterior distribution by a mixture of lag-k smoothing distributions. Using this approximation, we recast the filtering problem as a sequence of small smoothing problems which can be solved using

†*Address for correspondence:* Nicholas G. Polson, Graduate School of Business, University of Chicago, 5807 South Woodlawn Avenue, Chicago, IL 60637, U.S.A. E-mail: **ngp@chicagogsb.edu**

standard MCMC approaches (e.g., Carlin, Polson and Stoffer, 1992; Carter and Kohn, 1994). The method is particularly well suited for conditionally Gaussian models where fast MCMC smoothing methods have been well developed. To implement sequential parameter learning, we exploit a sufficient statistic structure as in Fearnhead (2002) and Storvik (2002) to achieve an algorithm with a linear computational cost. The features of our approach are the inclusion of static parameters, use of sufficient statistics for a fast parameter update, and a fixed-lag updating scheme.

Similar fixed-lag schemes have been successfully applied within the particle filter. Clapp and Godsill (1999) and Kitagawa and Sato (2001) use fixed-lag smoothers to estimate lagged state variables. Pitt and Shephard (2001) propose a fixed-lag auxiliary particle filter to deal with outliers. However, these approaches differ from ours in a number of ways: they are all based on importance sampling methods, and either they consider the smoothing problem, or they assume known parameters. In contrast, our method is based on a rolling-window MCMC approach, it addresses the filtering problem, and incorporates sequential parameter learning.

Our practical filtering approach requires three inputs for implementation. First, we must specify the number of independent state trajectories, N, used to approximate the filtering density. Second, we require the number of MCMC iterations, G, needed to obtain a draw from the fixed-lag smoothing distribution. Efficient sampling strategies for smoothing problems can be exploited to reduce the number of MCMC iterations required. Finally, we need the lag length, k, for the rolling-window state update. Section 2.6 proposes diagnostics to choose these inputs and Section 3 demonstrates their use through examples.

We illustrate our method with two applications. In a low-dimensional setting, we consider a benchmark autoregressive plus noise model with sequential parameter learning. For simulated data we find that there is little difference between our approach and Storvik's SIR and APF methods. However, when an unmodelled change point is included in the state process, we find that Storvik's algorithm experiences particle degeneracy leading to biased estimates and underestimation of posterior uncertainty, whereas our approach does not suffer from these problems. In a higher-dimensional setting, we consider a dynamic spatio-temporal model and show that our results closely match the posteriors from a full MCMC analysis.

The rest of the paper is organized as follows. Section 2 describes the practical filtering algorithm for combined state and parameter estimation. We also describe simulation-based algorithms for MCMC state and parameter generation, and provide diagnostics for choosing algorithmic inputs. Section 3 applies our methodology to the autoregressive and spatio-temporal models. Finally, Section 4 provides a final discussion of the work.

2. Sequential Parameter Learning and Filtering

The combined state filtering and sequential parameter learning problem can be described as follows. Consider a general state-space model with an observation vector $\mathbf{y}_t$, an unobserved state vector $\mathbf{x}_t$, and a parameter vector $\boldsymbol{\theta}$. The model is specified in terms of the densities

$$\begin{aligned} (Observation) \quad \mathbf{y}_t &\sim p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}) \\ (Evolution) \quad \mathbf{x}_t &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta}) \\ (Prior) \quad \boldsymbol{\theta} &\sim p(\boldsymbol{\theta}). \end{aligned}$$

The initial state distribution $p(\mathbf{x}_0|\boldsymbol{\theta})$ is assumed to be known. Bayesian state filtering and parameter learning requires calculation of the joint posterior distribution $p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{1:t})$ at each time $t = 1, \ldots, T$, where $\mathbf{y}_{s:t} = \{\mathbf{y}_s, \ldots, \mathbf{y}_t\}$ denotes the block of observations, and we define $\mathbf{x}_{s:t}$ similarly. Given the joint distribution, inference for the states and parameters is obtained through the marginal distributions $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ and $p(\boldsymbol{\theta}|\mathbf{y}_{1:t})$. In almost all cases, the joint filtering and parameter distribution is unavailable in closed form, and must be approximated using Monte Carlo methods (see Doucet, de Freitas and Gordon, 2001, for a comprehensive review of the subject).

2.1. Filtering with Fixed Parameters

Before considering combined state and parameter estimation, we first describe the pure filtering problem where the parameters $\boldsymbol{\theta}$ are assumed to be known. Here, the goal is to generate samples $\{\mathbf{x}_t\}$ from the filtering density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ for each time t. Our approach is to sample from the joint distribution $p(\mathbf{x}_{t-k+1:t}|\mathbf{y}_{1:t})$ and obtain draws from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ as the marginal. At the heart of the proposed algorithm is the representation of the joint distribution as a mixture of lag-k smoothing distributions:

$$\begin{aligned} p(\mathbf{x}_{t-k+1:t}|\mathbf{y}_{1:t}) &= \int p(\mathbf{x}_{t-k+1:t} \,|\, \mathbf{x}_{t-k}, \mathbf{y}_{1:t})\, dp(\mathbf{x}_{t-k}|\mathbf{y}_{1:t}) \\ &= \int p(\mathbf{x}_{t-k+1:t} \,|\, \mathbf{x}_{t-k}, \mathbf{y}_{t-k+1:t})\, dp(\mathbf{x}_{t-k}|\mathbf{y}_{1:t}), \end{aligned} \tag{1}$$

where the second identity exploits the Markovian property of the state-space model. To resolve the integral in (1) as a Monte Carlo average, we require samples $\{\mathbf{x}_{t-k}^{(i)}\}$ from $p(\mathbf{x}_{t-k}|\mathbf{y}_{1:t})$. Our approach makes the assumption that the lag k is sufficiently large so that samples from $p(\mathbf{x}_{t-k}|\mathbf{y}_{1:t-1})$ can be treated as samples from $p(\mathbf{x}_{t-k}|\mathbf{y}_{1:t})$. This approximation allows us to re-use draws of $\mathbf{x}_{t-k}$ from the previous step in the filtering algorithm without reweighting the samples.

By Bayes' theorem, we can show that this fixed-lag approximation is equivalent to assuming that $\mathbf{x}_{t-k}$ and $\mathbf{y}_t$ are conditionally independent given $\mathbf{y}_{t-k+1:t}$. Due to the theoretical properties of general hidden Markov models, this approximation is likely to hold for a reasonable value of k in many instances. This is because the sensitivity of the predictive distribution to the initial state typically decays quickly at an exponential rate (see Le Gland and Mevel, 1997; Künsch, 2001; Del Moral, Doucet and Jasra, 2006).

The algorithm for pure state filtering proceeds as follows. During an initial warm-up period ($t = 1, \ldots, k$), we sample from the full smoothing distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ using MCMC methods. Draws of $\mathbf{x}_t$ are used to represent the filtering distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. During the subsequent period ($t = k+1, \ldots, T$), we sample from the fixed-lag smoothing distribution $p(\mathbf{x}_{t-k+1:t}|\mathbf{x}_{t-k}, \mathbf{y}_{t-k+1:t})$ using MCMC methods, where the $\mathbf{x}_{t-k}$ are draws available from the previous step in the algorithm. After G iterations of the sampler, we store the last draw of $\mathbf{x}_{t-k+1}$ for the next stage of the algorithm and use the last draw of $\mathbf{x}_t$ to represent the filtering distribution. The algorithm is summarised below.

Algorithm 1: Filtering with Fixed Parameters

For each time period $t = k+1, \ldots, T$:

For each sample path $i = 1, \ldots, N$:

1. Run an MCMC with stationary distribution $p(\mathbf{x}_{t-k+1:t}|\mathbf{x}_{t-k}^{(i)}, \mathbf{y}_{t-k+1:t})$.
2. Define $\mathbf{x}_{t-k+1:t}^{(i)}$ as the last value of $\mathbf{x}_{t-k+1:t}$ in the chain.
3. Store $\mathbf{x}_{t-k+1}^{(i)}$ as a draw from $p(\mathbf{x}_{t-k+1}|\mathbf{y}_{1:t})$.
4. Report $\mathbf{x}_t^{(i)}$ as a draw from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

The algorithm depends on three inputs: N, the number of independent state trajectories stored and used to approximate the filtering density; G, the number of MCMC iterations to obtain one draw from the fixed-lag smoothing distribution in Step 1; and k, the rolling window width used for state updating. At each time step, for each of the N histories, we require G MCMC iterations to update k states, so the computational cost of the algorithm is $O(NGk)$ per time period.

2.2. Filtering with Unknown Parameters using Full Histories

We now turn to the problem of combined state and sequential parameter learning. Here, the goal is to simulate from the joint filtering distribution $p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{1:t})$ for each time t. Following our approach for pure filtering, we design an algorithm to simulate from the distribution $p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta}|\mathbf{y}_{1:t})$, which provides draws

from the filtering distribution $p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{1:t})$ as the marginal. We consider the mixture representation

$$\begin{aligned} p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta}|\mathbf{y}_{1:t}) &= \int p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{x}_{0:t-k}, \mathbf{y}_{1:t})\; dp(\mathbf{x}_{0:t-k}|\mathbf{y}_{1:t}) \\ &\approx \int p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{x}_{0:t-k}, \mathbf{y}_{1:t})\; dp(\mathbf{x}_{0:t-k}|\mathbf{y}_{1:t-1}). \end{aligned} \tag{2}$$

Note that the integration is now taken with respect to the joint distribution of the history $\mathbf{x}_{0:t-k}$ as opposed to the marginal draws of $\mathbf{x}_{t-k}$ as in (1). The approximation in (2) relies on the assumption that the lag k is large enough so that samples from $p(\mathbf{x}_{0:t-k}|\mathbf{y}_{1:t-1})$ can be used as samples from $p(\mathbf{x}_{0:t-k}|\mathbf{y}_{1:t})$. In Section 2.6, we discuss this approximation in more detail and propose diagnostic measures for choosing k.

Our approach requires simulating from the fixed-lag smoothing distribution $p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{x}^{(i)}_{0:t-k}, \mathbf{y}_{t-k+1:t})$, conditional on stored histories $\mathbf{x}^{(i)}_{0:t-k}$ at each time t. This can be done using MCMC methods by iterating between the conditional distributions

$$p(\mathbf{x}_{t-k+1:t} \,|\, \boldsymbol{\theta}, \mathbf{x}^{(i)}_{t-k}, \mathbf{y}_{t-k+1:t}) \quad \text{and} \quad p(\boldsymbol{\theta} \,|\, \mathbf{x}_{t-k+1:t}, \mathbf{x}^{(i)}_{0:t-k}, \mathbf{y}_{1:t}). \tag{3}$$

The states are generated efficiently using simulation smoothing algorithms as described in Section 2.4. The Markov property of the model ensures a fixed computational cost for the state update. For the parameters, however, the conditional distribution depends on the entire history $(\mathbf{x}_{0:t}, \mathbf{y}_{1:t})$, which leads to a computational cost which grows over time. This implies that the general state and learning algorithm is infeasible in real-time settings when fast updating is required. However, in the next subsection, we show how the parameter update can be simplified by exploiting sufficient statistics, which leads to an algorithm with a fixed computational cost.

The general algorithm for combined filtering and sequential parameter learning proceeds as follows. During a warm-up period ($t = 1, \ldots, k$), we simulate from the full smoothing distribution $p(\mathbf{x}_{0:t}, \boldsymbol{\theta}|\mathbf{y}_{1:t})$ using MCMC methods. The marginal draws of $(\mathbf{x}_t, \boldsymbol{\theta})$ are used as samples from the filtering distribution. For subsequent times ($t = k+1, \ldots, T$), we sample from the lag-k smoothing distribution $p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{x}_{0:t-k}, \mathbf{y}_{1:t})$ where $\mathbf{x}_{0:t-k}$ are draws available from the previous step in the algorithm. The values of $\mathbf{x}_{t-k+1}$ are added to the stored history for use at the next time period, and draws of $(\mathbf{x}_t, \boldsymbol{\theta})$ are used to represent the filtering distribution. The algorithm is summarised below.

Algorithm 2: Filtering with Sequential Parameter Learning

For each time period $t = k+1, \ldots, T$:

For each sample path $i = 1, \ldots, N$:

1. Run an MCMC with stationary distribution $p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta}|\mathbf{x}^{(i)}_{0:t-k}, \mathbf{y}_{1:t})$.
2. Define $(\mathbf{x}^{(i)}_{t-k+1:t}, \boldsymbol{\theta}^{(i)})$ as the last value of $(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta})$ in the chain.
3. Store $\mathbf{x}^{(i)}_{0:t-k+1}$ as a draw from $p(\mathbf{x}_{0:t-k+1}|\mathbf{y}_{1:t})$.
4. Report $(\mathbf{x}^{(i)}_t, \boldsymbol{\theta}^{(i)})$ as a draw from $p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{1:t})$.

Here N is the number of independent histories $\{\mathbf{x}_{0:t-k}\}$ which are stored and used to estimate the filtering distribution. G is the number of MCMC iterations required to obtain a sample from the fixed-lag smoothing distribution. The value of G will depend on the posterior dependence between the states and parameters, and block updating should be used wherever possible to reduce the number of required MCMC iterations. The number of MCMC iterations can be reduced by initializing the chain at the last values from the previous time period, since these values are typically quite close to the target distribution. The value of G can be chosen using standard MCMC diagnostic measures, either sequentially or off-line. The window width k is chosen so that the approximation in (2) is valid, as discussed in Section 2.6.

2.3. Sequential Parameter Learning with Sufficient Statistics

In many models, the parameter distribution depends on a low-dimensional set of sufficient statistics $\mathbf{S}_t = \mathcal{S}(\mathbf{x}_{0:t}, \mathbf{y}_{1:t})$, such that $p(\boldsymbol{\theta} \,|\, \mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = p(\boldsymbol{\theta} \,|\, \mathbf{S}_t)$. Fearnhead (2002) and Storvik (2002) exploit this sufficient statistic structure to implement sequential parameter learning within the particle filter. Here, we extend use this idea to obtain fast parameter updating within our practical filtering framework. Define $\mathbf{T}_t = \{\mathbf{S}_t, \mathbf{x}_t\}$ as the extended sufficient statistics for the parameters and states. We also assume a recursive updating rule of the form $\mathbf{T}_t = f(\mathbf{T}_{t-1}, \mathbf{x}_t, \mathbf{y}_t)$ is available. The joint distribution of the states and parameters given the history at time $t-k$ can then be written as

$$p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{x}_{0:t-k}, \mathbf{y}_{1:t-k}) = p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{T}_{t-k}).$$

The target distribution can be expressed as a mixture of lag-k smoothing distributions:

$$\begin{aligned} p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{y}_{1:t}) &= \int p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{T}_{t-k}, \mathbf{y}_{t-k+1:t})\, dp(\mathbf{T}_{t-k} \,|\, \mathbf{y}_{1:t}) \\ &\approx \int p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{T}_{t-k}, \mathbf{y}_{t-k+1:t})\, dp(\mathbf{T}_{t-k} \,|\, \mathbf{y}_{1:t-1}), \end{aligned} \tag{4}$$

where the integral is taken with respect to draws of the sufficient statistics $\{\mathbf{T}_{t-k}^{(i)}\}$ rather than the full histories $\{\mathbf{x}_{0:t-k}^{(i)}\}$ as in (2). The approximation in (4) is based on the assumption that k is large enough so that samples from $p(\mathbf{T}_{t-k}|\mathbf{y}_{1:t-1})$ can be used as samples from $p(\mathbf{T}_{t-k}|\mathbf{y}_{1:t})$.

The algorithm requires simulating from the fixed-lag smoothing distribution at each time t:

$$p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{T}_{t-k}^{(i)}, \mathbf{y}_{t-k+1:t}).$$

We consider a two-block MCMC sampler which iterates between the conditional distributions for the states and parameters as in (3). The state update remains unchanged from (3) and can be achieved at a fixed cost. The parameter distribution now simplifies due to the sufficient statistic structure to

$$p(\boldsymbol{\theta} \,|\, \mathbf{T}_{t-k}^{(i)}, \mathbf{x}_{t-k+1:t}, \mathbf{y}_{t-k+1:t}) \;\propto\; p(\boldsymbol{\theta} \,|\, \mathbf{T}_{t-k}^{(i)})\, p(\mathbf{x}_{t-k+1:t}, \mathbf{y}_{t-k+1:t} \,|\, \boldsymbol{\theta}, \mathbf{T}_{t-k}^{(i)}). \tag{5}$$

This leads to a parameter update with a fixed computational cost, as the posterior distribution in (5) only requires combining the stored sufficient statistics $\mathbf{T}_{t-k}^{(i)}$ with the last k observations and imputed states. This is the key to our fast parameter updating algorithm which is described below.

The filtering algorithm now updates and stores the sufficient statistics rather than the full state histories. During the initial warm-up period ($t = 1, \ldots, k$), we draw from the full smoothing distribution $p(\mathbf{x}_{0:t}, \boldsymbol{\theta} \,|\, \mathbf{y}_{1:t})$ using MCMC methods, and summarise the filtering distribution using draws of $(\mathbf{x}_t, \boldsymbol{\theta})$. For subsequent times ($t = k+1, \ldots, T$), we sample from the lag-k smoothing distribution $p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{T}_{t-k}, \mathbf{y}_{t-k+1:t})$, where $\mathbf{T}_{t-k}$ are the sufficient statistics from the previous step in the algorithm. Finally, the sufficient statistics $\mathbf{T}_{t-k+1}$ are computed and stored for the next step of the algorithm and the draws $(\mathbf{x}_t, \boldsymbol{\theta})$ are used to summarise the filtering density. The algorithm is given below.

Algorithm 3: Filtering with Sufficient Statistics

For each time period $t = k+1, \ldots, T$:

For each sample path $i = 1, \ldots, N$:

1. Run an MCMC with stationary distribution $p(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta} \,|\, \mathbf{T}_{t-k}^{(i)}, \mathbf{y}_{t-k+1:t})$.
2. Define $(\mathbf{x}_{t-k+1:t}^{(i)}, \boldsymbol{\theta}^{(i)})$ as the last value of $(\mathbf{x}_{t-k+1:t}, \boldsymbol{\theta})$ in the chain.
3. Set $\mathbf{T}_{t-k+1}^{(i)} = f(\mathbf{T}_{t-k}^{(i)}, \mathbf{x}_{t-k+1}^{(i)}, \mathbf{y}_{t-k+1})$.
4. Store $\mathbf{T}_{t-k+1}^{(i)}$ as a draw from $p(\mathbf{T}_{t-k+1} \,|\, \mathbf{y}_{1:t})$.
5. Report $(\mathbf{x}_t^{(i)}, \boldsymbol{\theta}^{(i)})$ as a draw from $p(\mathbf{x}_t, \boldsymbol{\theta} \,|\, \mathbf{y}_{1:t})$.

This provides a fast algorithm for sequential parameter learning, as it only tracks the state and sufficient statistics rather the full state trajectories. While algorithms such as Storvik (2002) and Fearnhead (2002) also exploit sufficient statistics, our approach uses them within a fixed-lag MCMC scheme, avoids importance sampling (and degeneracy) and provides N independent samples from the filtering distribution. This typically allows us to use smaller sample sizes than importance sampling approaches.

2.4. Methods for State Generation

The proposed filtering algorithms (Algorithms 1-3) require efficient methods for generating the states given the parameters. We rely on a number of existing Monte Carlo smoothing algorithms to do this. For linear Gaussian models, the states can be efficiently sampled as a block using the forward-filtering backward-sampling (FFBS) algorithm of Carter and Kohn (1994) and Frühwirth-Schnatter (1994). FFBS exploits the factorisation of the state distribution as

$$p(\mathbf{x}_{t-k+1:t}|\mathbf{x}_{t-k}, \mathbf{y}_{t-k+1:t}) \;=\; p(\mathbf{x}_t|\mathbf{x}_{t-k}, \mathbf{y}_{t-k+1:t}) \prod_{j=t-k+1}^{t-1} p(\mathbf{x}_j|\mathbf{x}_{j+1}, \mathbf{x}_{t-k}, \mathbf{y}_{t-k+1:j}).$$

This leads to a recursive approach for drawing $\mathbf{x}_{t-k+1:t}$. We first run a filter forward from time $t-k+1$ to t to obtain the filtered moments. We then draw $\mathbf{x}_t$ from $p(\mathbf{x}_t|\mathbf{x}_{t-k}, \mathbf{y}_{t-k+1:t})$ and sample backwards from the distributions $p(\mathbf{x}_j|\mathbf{x}_{j+1}, \mathbf{x}_{t-k}, \mathbf{y}_{t-k+1:j})$ for times $j = t-1, \ldots, t-k+1$. This provides direct draws of the states conditional on the parameters.

FFBS can also be used in conditionally Gaussian models, including scale mixtures of normals, discrete mixtures of normals, and multivariate models with multiplicative structure. For the latter, a fast algorithm can be obtained by partitioning the state vector into sub-blocks $\{\mathbf{x}_{t,b}\}$, such that each block is conditionally linear given the others. The states can then be drawn efficiently using a Gibbs sampler, where each block $\mathbf{x}_{t-k+1:t,b}$ is generated conditional on the others using FFBS. Weinberg, Brown and Stroud (2007) provide an example of this approach.

MCMC sampling efficiency can often be improved by introducing an extra block of latent state variables. While this increases the dimensionality of the state vector it poses little extra computational burden on our approach assuming that state sampling can be done efficiently and the latent variables and the states are nearly independent. Examples of latent variables in state-space models include Carlin, Polson and Stoffer (1992) for non-Gaussian errors, Kim, Shephard and Chib (1998) for stochastic volatility models, and Stroud, Müller and Polson (2003) for nonlinear models with state-dependent variances.

Simulation smoothing algorithms have been developed for other types of state-space models. Carlin, Polson and Stoffer (1992) provide single-state MCMC smoothing schemes for nonlinear state-space models. Shephard and Pitt (1997) and Gamerman (1998) propose block smoothing algorithms for dynamic exponential family models and dynamic generalised linear models, respectively. Scott (2001) provides MCMC methods for hidden Markov models, while Künsch (2001) considers general discrete time series models.

2.5. Methods for Parameter Generation

Section 2.3 showed how sufficient statistics can be exploited to achieve a fast parameter update. In this section, we consider an important class of Gaussian linear regression models for the parameters and show how the lag-k updating recursions for the sufficient statistics can be implemented. Conditional on the states, the models can be written as

$$\mathbf{y}_t \;=\; \mathbf{H}_t'\boldsymbol{\alpha} + \mathbf{e}_t, \qquad \mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I}), \tag{6}$$

$$\mathbf{x}_t \;=\; \mathbf{F}_t'\boldsymbol{\beta} + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{7}$$

where $\mathbf{H}_t' = H(\mathbf{x}_t)$ and $\mathbf{F}_t' = F(\mathbf{x}_{t-1})$ are matrices whose elements are possibly nonlinear functions of the states, and $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \tau^2, \sigma^2)$ is the vector of unknown parameters. In our Bayesian setting, the linear regression models (6)–(7) provide a conjugate structure for fast parameter updating.

We assume independent normal-inverse gamma prior distributions for the parameters $(\boldsymbol{\alpha}, \tau^2)$ and $(\boldsymbol{\beta}, \sigma^2)$. This leads to closed-form posterior distributions depending on a low-dimensional set of sufficient statistics which can be updated recursively in time. For the evolution parameters, the posteriors are

$$p(\boldsymbol{\beta}|\sigma^2, \mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{B}_t^{-1}\mathbf{b}_t, \sigma^2\mathbf{B}_t^{-1}) \quad (8)$$

$$p(\sigma^2|\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = \mathcal{IG}\left(\frac{\nu_t}{2}, \frac{d_t}{2}\right). \quad (9)$$

The sufficient statistics are $\mathbf{S}_t = \{\mathbf{B}_t, \mathbf{b}_t, \nu_t, d_t\}$. Using standard results from linear model theory, the lag-k updating recursions for the sufficient statistics are given by

$$\mathbf{B}_t = \mathbf{B}_{t-k} + \mathbf{F}'\mathbf{F} \quad (10)$$

$$\mathbf{b}_t = \mathbf{b}_{t-k} + \mathbf{F}'\mathbf{x} \quad (11)$$

$$\nu_t = \nu_{t-k} + kp \quad (12)$$

$$d_t = d_{t-k} + \mathbf{b}'_{t-k}\mathbf{B}_{t-k}\mathbf{b}_{t-k} + \mathbf{x}'\mathbf{x} - \mathbf{b}'_t\mathbf{B}_t\mathbf{b}_t, \quad (13)$$

where $\mathbf{F} = [\mathbf{F}_{t-k+1}, \ldots, \mathbf{F}_t]'$, $\mathbf{x} = (\mathbf{x}'_{t-k+1}, \ldots, \mathbf{x}'_t)'$, and p is the dimension of $\mathbf{x}_t$.

Similar results are available for the observation parameters. For these parameters, the posterior distributions at time t are given by $p(\boldsymbol{\alpha}|\tau^2, \mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{A}_t^{-1}\mathbf{a}_t, \tau^2\mathbf{A}_t^{-1})$ and $p(\tau^2|\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = \mathcal{IG}(n_t/2, s_t/2)$, and the sufficient statistics are $\mathbf{S}'_t = \{\mathbf{A}_t, \mathbf{a}_t, n_t, s_t\}$. The lag-$k$ updating recursions for $\mathbf{S}'_t$ have the same form as $\mathbf{S}_t$ in (10)-(13) where $\mathbf{x}$ is replaced by $\mathbf{y}$, $\mathbf{F}$ is replaced by $\mathbf{H}$, and p by the dimension of $\mathbf{y}_t$. The parameters $(\boldsymbol{\alpha}, \tau^2)$ and $(\boldsymbol{\beta}, \sigma^2)$ are independent a posteriori, so the entire parameter vector $\boldsymbol{\theta}$ can be generated as a block using the normal inverse-gamma posteriors. An example of this is given in Section 3.1.

Many other models lead to a sufficient statistic structure for the parameters. In particular, discrete state-space models often provide a structure amenable to fast lag-k updating. Examples include hidden Markov models with an unknown transition probability matrix (Cappé, Moulines and Rydèn, 2005), multiple change-point models (Chib, 1998) and dynamic categorical time series models (Carlin and Polson, 1992). Efficient state and parameter inference in these models can be achieved through simple Gibbs sampling algorithms.

2.6. Choosing the Lag Length k

The general state and parameter learning algorithm (Section 2.2) is based on the assumption that the lag k is sufficiently large so that samples from $p(\mathbf{x}_{0:t-k}|\mathbf{y}_{1:t-1})$ can be used as samples from $p(\mathbf{x}_{0:t-k}|\mathbf{y}_{1:t})$. By Bayes' rule, these two distributions are related by

$$p(\mathbf{x}_{0:t-k}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t-k}, \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}\, p(\mathbf{x}_{0:t-k}|\mathbf{y}_{1:t-1}). \quad (14)$$

Thus the target approximation is equivalent to assuming that $\mathbf{y}_t$ and $\mathbf{x}_{0:t-k}$ are conditionally independent given $\mathbf{y}_{1:t-1}$. When a sufficient statistic structure is available (Section 2.3), $\mathbf{x}_{0:t-k}$ is replaced with $\mathbf{T}_{t-k}$ in (14), and our assumption becomes that $\mathbf{y}_t$ and $\mathbf{T}_{t-k}$ are conditionally independent given $\mathbf{y}_{1:t-1}$.

From a practical perspective, a simple approach for choosing k is to run the algorithm for a range of values (e.g. $k = 10, 15, 25$) and monitor a set of state and parameter moments to see how they converge. The sensitivity of the predictive distribution $p(\mathbf{y}_t|\mathbf{T}_{t-k}, \mathbf{y}_{t-k+1:t-1})$ to $\mathbf{T}_{t-k}$ as a function of k in many situations will decay exponentially. A more formal approach would be to calculate a diagnostic measure D_k given by the distance between the two distributions $p(\mathbf{y}_t|\mathbf{T}_{t-k}, \mathbf{y}_{1:t-1})$ and $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$. For example, Kitagawa (1996) uses a Kullback-Leibler divergence metric.

In our experience a surprisingly low value works well for most models and parameters of interest. The approximation holds remarkably well for regression parameters in the observation and evolution equations as illustrated in the examples in Section 3. Clearly, in non-stationary environments with a fixed choice of k the method can break down. Moreover, when $\boldsymbol{\theta}$ enters the model only in the evolution equation then the choice of appropriate lag k could be highly dependent on the posterior range of $\boldsymbol{\theta}$. Trying to find a value of k such that $\mathbf{x}_{0:t-k}$ and $\mathbf{y}_t$ are approximately conditionally independent given $\mathbf{y}_{1:t-1}$ could critically depend

on the distribution of $\boldsymbol{\theta}$. Possible remedies in these situations include using a stochastic choice of k, maybe even as a function of historical MCMC draws.

From a theoretical perspective, a number of results are known on the sensitivity of the predictive distribution to the initial state in a pure filtering context and how this decays at an exponential rate (see Le Gland and Mevel, 1997; Künsch, 2001; Del Moral, Doucet and Jasra, 2006). Asymptotic results on when our target approximation is likely to work well in hidden Markov models can be found in Bickel, Ritov and Rydèn (1998) and Cappé, Moulines and Rydèn (2005), see Sections 4.3, 4.3.4 and 4.3.6 of the latter.

3. Applications

In this section, we analyse the empirical performance of our algorithm and compare it to the computationally intensive full MCMC as well as Storvik's particle filtering algorithm. First, we consider a benchmark autoregressive plus noise model with parameter learning. We show that Storvik's particle filter and our practical filter handle both state and sequential parameter learning efficiently under a correctly specified model. However, when an unmodelled change point is included in the state evolution process, the particle filtering methods provide poor inference, while our approach is quite accurate. Second, we consider a 10-dimensional spatio-temporal model and show that our filtering method compares favorably to standard particle filters in a higher-dimensional setting.

3.1. AR(1) plus Noise Model

The benchmark model for studying filtering methods and sequential parameter learning is an autoregressive plus noise model (see, for example, Pitt and Shephard, 1999; Storvik, 2002):

$$\begin{aligned} y_t &= x_t + e_t, & e_t &\sim \mathcal{N}(0, \tau^2), \\ x_t &= \beta_0 + \beta_1 x_{t-1} + w_t, & w_t &\sim \mathcal{N}(0, \sigma^2). \end{aligned}$$

Data are simulated using the parameter values $\beta_0 = 0$, $\beta_1 = 0.9$, $\sigma^2 = .04$ and $\tau^2 = .1$. The conditionally Gaussian structure of the model leads to an efficient two-block Gibbs sampler for state and parameter generation. Given the parameters, the states have a linear state-space form and can be efficiently generated using FFBS as in Section 2.4. Conditional on the states, the model takes a linear regression form for the parameters as in (6)-(7). We assume normal-inverse gamma priors with hyperparameters $\mathbf{B}_0 = \mathbf{I}$, $\mathbf{b}_0 = (0, 0.9)'$, $n_0 = 4$ and $s_0 = .4$, and the parameters β and τ^2 are generated from the posterior distributions given in Section 2.5. The sufficient statistic structure of the model can be exploited (Algorithm 3) to obtain an algorithm which is linear in time.

For the first experiment, 500 observations are generated and the filtering algorithm is run to estimate $(x_t, \beta_0, \beta_1, \tau^2)$. To choose the number of Gibbs iterations G for our algorithm, we run an MCMC smoother over the first 100 observations. The MCMC autocorrelations (not shown) decay rapidly to zero for all parameters and hence we choose a value of $G = 5$. The fixed-lag length is selected by running the filtering algorithm for different values of k. Figure 1 shows the filtered mean and 95% credible intervals for the parameters and the relative error for the filtered mean for different values of k. The errors decrease rapidly from lag $k = 2$ to $k = 3$, and then show a slow decrease up to lag $k = 25$. Clearly, a fairly small lag suffices for this model and we choose $k = 15$. The results below are based on $(N, G, k) = (1000, 5, 15)$.

Figure 2 compares our filtered mean and 95% credible intervals for the states and parameters with full MCMC. The practical filter closely matches full MCMC for state filtering and the sequential parameter learning of $\boldsymbol{\beta}$. The only minor difference comes in learning the observation variance τ^2. While practical filtering does a good job at tracking the marginal posterior for the first 200 time points by then the posterior standard deviation is slightly too small relative to full MCMC. This error is induced by the mixture of lag-k smoothing distributions approximation.

For the second experiment, we simulate a dataset of length 100 but include a large jump (change point) in the state evolution at time $t = 50$. Again we estimate the states and parameters $(x_t, \beta_0, \beta_1, \tau^2)$ using the priors above. For comparison, we also show the results for Storvik's SIR and APF algorithms with stratified sampling using $N = 10000$ particles. (The number of particles was chosen to make the run times for SIR

and APF roughly equal to ours.) The top row of Figure 3 shows the filtered means and 95% credible bands for β_1 while the bottom row shows the corresponding posterior densities at time $t = 50$. These plots show that the practical filter closely matches full MCMC while both particle filters (SIR and APF) produce biased estimates and underestimate the posterior variance. This poor performance is due to particle degeneracies in the filtering distribution.

Figure 4 shows results for the three filtering algorithms when a large outlier of $y = 6$ is included at time 50. Here, we show boxplots of the ratio of filter bias (estimated mean minus true mean) to the standard deviation for the states and parameters at selected time points. The results are based on 200 simulated datasets. The plot illustrates that the SIR algorithm performs poorly after the outlier with the bias in parameter estimation (β_1) persisting through time $t = 60$. The APF algorithm performs better than SIR with the bias disappearing by time $t = 53$. Finally, the practical filter recovers immediately after the outlier for both the states and the parameters. In general, the bias for the practical filter shows less variability than the other methods, indicating that our method is more reliable from dataset to dataset.

From a computational perspective, the first example (500 observations) requires roughly 45 seconds of CPU time using C code on a Pentium 1.8MHz processor, while the full MCMC requires 20 minutes. However, full MCMC grows quadratically in the length of the time series while our approach increases at a linear rate.

3.2. Dynamic Spatio-Temporal Model

To demonstrate the effectiveness of our algorithm in a higher-dimensional setting, we consider a dynamic spatio-temporal model proposed by Xu and Wikle (2007). The model is a vector autoregression with observation error of the form

$$\begin{aligned} \mathbf{y}_t &= \mathbf{x}_t + \mathbf{e}_t, & \mathbf{e}_t &\sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I}), \\ \mathbf{x}_t &= \mathbf{M}\mathbf{x}_{t-1} + \mathbf{w}_t, & \mathbf{w}_t &\sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}). \end{aligned}$$

The state and observation vectors $\mathbf{x}_t = (x_{t1}, \ldots, x_{tn})'$ and $\mathbf{y}_t = (y_{t1}, \ldots, y_{tn})'$ correspond to n equally-spaced locations along a spatial transect. The space-time structure in the model is induced by a tridiagonal transition matrix $\mathbf{M}$ defined in (15), which depends on three parameters $(\beta_1, \beta_2, \beta_3)$. For the analysis, we treat $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)'$ as an unknown parameter and estimate it sequentially within our filtering algorithm. The conditionally Gaussian structure of the model leads to a two-block Gibbs sampler for updating. The states have linear Gaussian form and can be sampled using FFBS. The parameters $\boldsymbol{\beta}$ have the linear regression form (7), where $\mathbf{F}_{t+1}$ is defined below:

$$\mathbf{M} = \begin{pmatrix} \beta_1 & \beta_2 & & 0 \\ \beta_3 & \beta_1 & \ddots & \\ & \ddots & \ddots & \beta_2 \\ 0 & & \beta_3 & \beta_1 \end{pmatrix}, \quad \mathbf{F}_{t+1} = \begin{pmatrix} x_{t1} & x_{t2} & 0 \\ x_{t2} & x_{t3} & x_{t1} \\ \vdots & \vdots & \vdots \\ x_{tn} & 0 & x_{t,n-1} \end{pmatrix}. \tag{15}$$

This leads to a fast parameter update which exploits a sufficient statistic structure (Algorithm 3). We assume a normal prior distribution for $\boldsymbol{\beta}$ with hyperparameters $\mathbf{B}_0 = 100\mathbf{I}$ and $\mathbf{b}_0 = (30, 30, 30)'$.

Data are generated with $T = 100$ and $n = 10$ using parameters $(\beta_1, \beta_2, \beta_3) = (.3, .6, .1)$, $\sigma^2 = 5$ and $\tau^2 = 1$. For our approach, we choose a Monte Carlo sample size of $N = 100$ to give a reasonable tradeoff between accuracy and run time. The number of Gibbs iterations $G = 3$ is selected based on the ACF plots from an MCMC smoother which showed rapid mixing of the Markov chain. The lag value $k = 10$ is chosen by comparing filtering runs for different values of k as in Section 3.1. The results are based on $(N, G, k) = (100, 3, 10)$. For comparison, a full MCMC is run with 1000 iterations. Figure 5 shows the filtered means and 95% credible intervals for the state variable x_{t5} and parameters $(\beta_1, \beta_2, \beta_3)$. The practical filter matches all of the sequential full MCMC posteriors very closely. The parameter posteriors exhibit quick parameter learning at the beginning of the sample before stabilising after about $t = 30$ observations.

Figure 6 provides a comparison with Storvik's SIR and APF algorithms with a large number of particles ($N = 10000$). The priors and algorithm parameters for our approach are the same as above. The figure

shows the filtered means and densities for β_2 for a series of length 50 which includes an outlier at time $t = 25$. The SIR algorithm performs poorly even before the outlier as particles are propagated from the transition density, and in high dimensions this can lead quickly to unbalanced weights and degeneracy. The APF performs far better showing that look-ahead schemes and adaptation to the new observation can help in higher dimensions. However at the outlier and thereafter our approach provides more accurate inference for β_2 and the other parameters (not shown) when compared to the full MCMC results.

4. Conclusions

This paper provides a Bayesian filtering and sequential parameter learning algorithm for general state space models. The approach relies on a lag-k mixture approximation for the filtering distribution and a sufficient statistic structure for the parameters. The approach is MCMC-based and applies readily to conditionally Gaussian models where the states can be generated efficiently. The filter can be easily implemented by converting existing smoothing code for state-space algorithms. Unlike sequential importance sampling approaches such as particle filters, it provides independent samples from the target distribution, does not suffer from particle degeneracies, and handles outliers and high dimensional problems well.

This filtering method has achieved recent success in financial applications, including sequential portfolio allocation and stochastic volatility models with jumps (Johannes, Polson and Stroud, 2002, 2006). However, sequential parameter learning still poses a number of computational challenges. Evolution variance parameters are notoriously hard to estimate sequentially (Stroud, Polson and Müller, 2004) and more research is required to understand this problem. Another avenue for future research is filtering in continuous-time models with discretely sampled data, where filling-in-the-missing-data estimators have been proposed in an MCMC smoothing framework (see Eraker, 2001; Elerian, Shephard and Chib, 2001). These estimators can be extended to the filtering context using the methodology proposed in this paper.

5. Acknowledgements

We thank the editor, associate editor and three referees for very helpful comments and suggestions which greatly improved the manuscript. We also thank Lurdes Inoue and Michael Pitt for helpful comments.

References

Andrieu, C. and Doucet, A. (2003) On-line expectation-maximization type algorithms for parameter estimation in general state space models. *Proc. IEEE ICASSP*.

Andrieu, C., Doucet, A. and Tadić, V. (2005) Online simulation-based methods for parameter estimation in nonlinear non Gaussian state-space models. *Proc. IEEE CDC*.

Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1997) Dynamic conditional independence models and Markov chain Monte Carlo methods. *Journal of the American Statistical Association*, **92**, 1403–1412.

Bickel, P. J., Ritov, Y. and Rydèn, T. (1998) Asymptotic normality of the maximum likelihood estimator for general hidden Markov models. *Annals of Statistics*, **26**, 1614–1635.

Cappé, O., Moulines, E. and Rydèn, T. (2005) *Inference in Hidden Markov Models*. New York: Springer.

Carlin, B. P. and Polson, N. G. (1992) Monte Carlo Bayesian methods for discrete regression models and categorical time series. In *Bayesian Statistics 4* (eds. J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), 577–586. Oxford: Oxford University Press.

Carlin, B. P., Polson, N. G. and Stoffer, D. S. (1992) A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *Journal of the American Statistical Association*, **87**, 493–500.

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

Chib, S. (1998) Estimation and comparison of multiple change point models. *Journal of Econometrics*, **86**, 221–241.

Clapp, T. C. and Godsill, S. (1999) Fixed-lag smoothing using sequential importance sampling. In *Bayesian Statistics 6* (eds. J. Bernardo, J. Berger, A. Dawid and A. Smith), 743–752. Oxford: Oxford University Press.

Del Moral, P., Doucet, A. and Jasra, A. (2006) Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, **68**, 411–436.

Doucet, A., de Freitas, J. F. G. and Gordon, N. (eds.) (2001) *Sequential Monte Carlo Methods in Practice*. Kluwer.

Elerian, O., Shephard, N. and Chib, S. (2001) Likelihood inference for discretely observed non-linear diffusions. *Econometrica*, **69**, 959–993.

Eraker, B. (2001) MCMC analysis of diffusion models with application to finance. *Journal of Business and Economic Statistics*, **19**, 177–191.

Fearnhead, P. (2002) MCMC, sufficient statistics and particle filter. *Journal of Computational and Graphical Statistics*, **11**, 848–862.

Frühwirth-Schnatter, S. (1994) Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, **15**, 183–202.

Gamerman, D. (1998) Monte Carlo Markov chains for dynamic generalised linear models. *Biometrika*, **85**, 215–227.

Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings*, vol. F-140, 107–113. IEE.

Hürzeler, M. and Künsch, H. (2001) Approximating and maximizing the likelihood for general SSM. In *Sequential Monte Carlo Methods in Practice* (eds. A. Doucet, J. de Freitas and N. Gordon). Springer.

Johannes, M. S., Polson, N. G. and Stroud, J. R. (2002) Sequential optimal portfolio performance: Market and volatility timing. *Tech. rep.*, Graduate School of Business, University of Chicago.

— (2006) Sequential parameter estimation in stochastic volatility models with jumps. *Tech. rep.*, Graduate School of Business, University of Chicago.

Kim, S., Shephard, N. and Chib, S. (1998) Stochastic volatility: Likelihood inference and comparison with ARCH models. *Review of Economic Studies*, **65**, 361–393.

Kitagawa, G. (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, **5**, 1–25.

Kitagawa, G. and Sato, S. (2001) Monte Carlo smoothing and self-organizing state-space model. In *Sequential Monte Carlo Methods in Practice* (eds. A. Doucet, J. de Freitas and N. Gordon). Springer.

Künsch, H. R. (2001) State space and hidden Markov models. In *Complex Stochastic Systems* (eds. D. C. O.E. Barndorff-Nielsen and C. Klüppelberg). Boca Raton: Chapman and Hall.

Le Gland, F. and Mevel, L. (1997) Exponential forgetting and geometric ergodicity in hidden Markov models. In *36th IEEE Conference on Decision and Control (CDC)*, 537–542.

Liu, J. and West, M. (2001) Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo Methods in Practice* (eds. A. Doucet, J. de Freitas and N. Gordon). Springer.

Liu, J. S. and Chen, R. (1995) Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, **93**, 1032–1044.

Pitt, M. and Shephard, N. (2001) Auxiliary variable based particle filters. In *Sequential Monte Carlo Methods in Practice* (eds. A. Doucet, J. de Freitas and N. Gordon), 273–293. Springer.

Pitt, M. K. (2002) Smooth particle filters for likelihood evaluation and maximisation. *Tech. rep.*, Department of Economics, University of Warwick.

Pitt, M. K. and Shephard, N. (1999) Filtering via simulation: Auxiliary particle filter. *Journal of the American Statistical Association*, **94**, 590–599.

Scott, S. (2001) Bayesian methods for hidden Markov models: Recursive computing in the 21st Century. *Journal of the American Statistical Association*, **97**, 337–351.

Shephard, N. and Pitt, M. K. (1997) Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, **84**, 653–667.

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

Stroud, J. R., Müller, P. and Polson, N. G. (2003) Nonlinear state-space models with state-dependent variances. *Journal of the American Statistical Association*, **98**, 377–386.

Stroud, J. R., Polson, N. G. and Müller, P. (2004) Practical filtering for stochastic volatility models. In *State Space and Unobserved Component Models* (eds. A. Harvey, S. Koopmans and N. Shephard), 236–247. Oxford University Press.

Weinberg, J., Brown, L. D. and Stroud, J. R. (2007) Bayesian forecasting of an inhomogeneous Poisson process with applications to call center data. *To appear,* Journal of the American Statistical Association.

Xu, K. and Wikle, C. K. (2007) Estimation of parameterized spatio-temporal dynamic models. *Journal of Statistical Planning and Inference*, **137**, 567–588.

Fig. 1. AR(1) plus noise model. Filtered means and 95% credible intervals for the parameters $(\beta_0, \beta_1, \tau^2)$ for the practical filter with $(N, G) = (1000, 5)$ for different values of k. The bottom row represents the time-averaged relative error for the posterior mean defined as the absolute error divided by the filtered standard deviation.

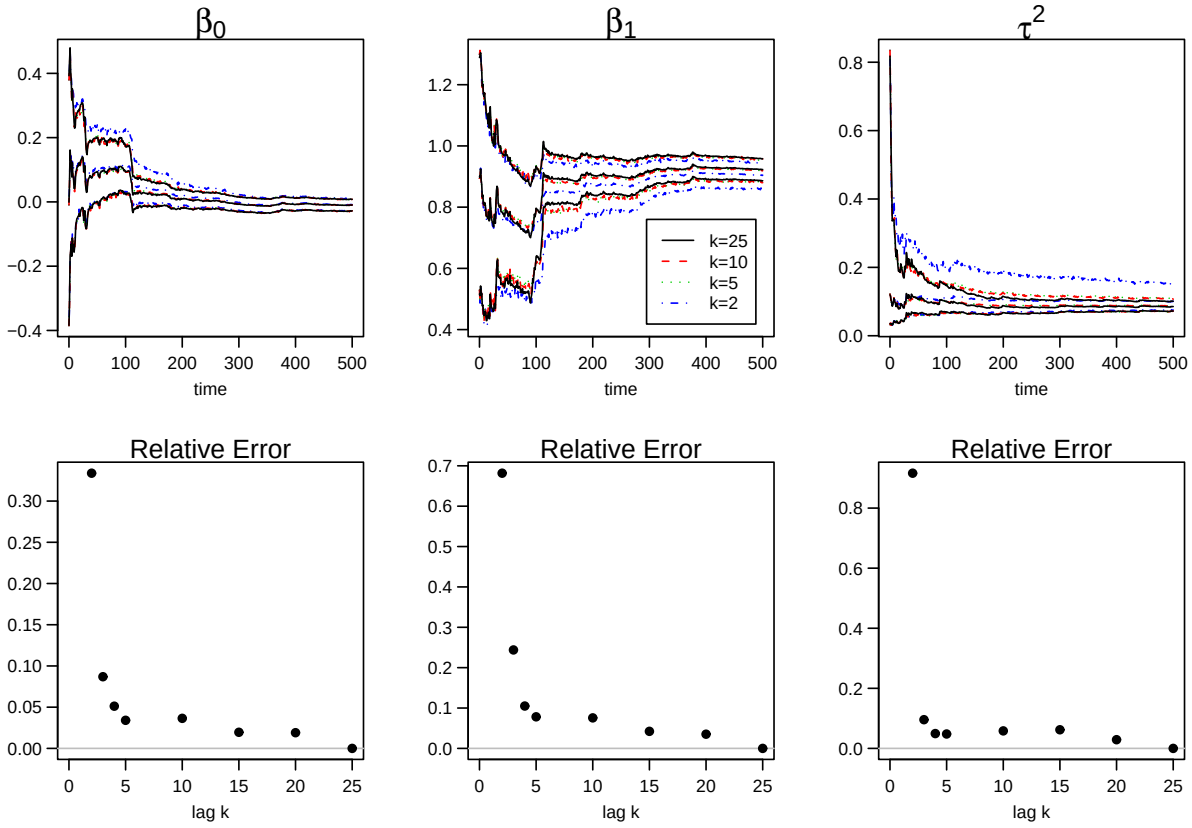

Fig. 2. AR(1) plus noise model. Filtered means and 95% credible intervals for the state variable $\mathbf{x}_t$, the evolution parameters (β_0, β_1) and the observation variance τ^2 The dashed lines represent the practical filter with $(N, G, k) = (1000, 5, 15)$, while the solid lines represent a full MCMC with 10000 iterations.

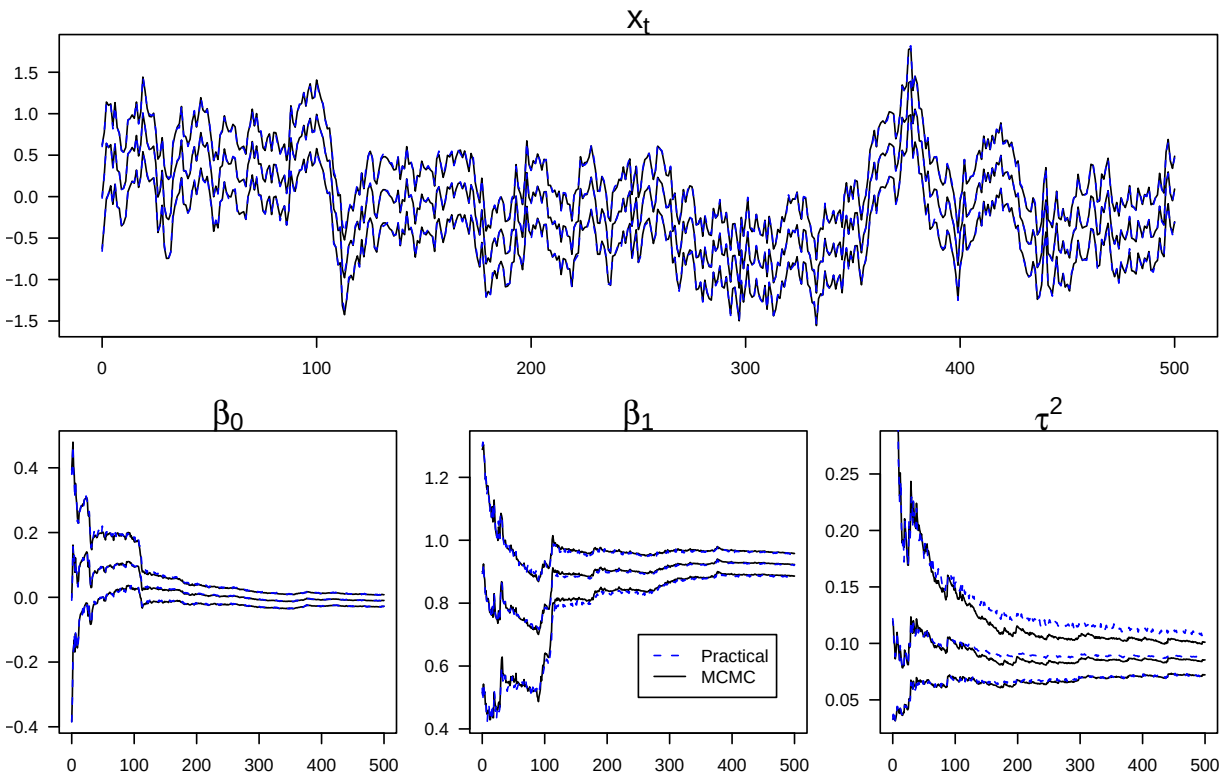

Fig. 3. AR(1) plus noise model with unmodelled change point at time 50. Filtered means and 95% credible intervals (top) and posterior distributions at time 50 (bottom) for the AR coefficient β_1. Left: Storvik's SIR particle filter algorithm with $N = 10000$. Center: Storvik's APF algorithm with $N = 10000$. Right: practical filter with $(N, G, k) = (1000, 5, 15)$. Solid lines in all panels are from a full MCMC with 10000 iterations.

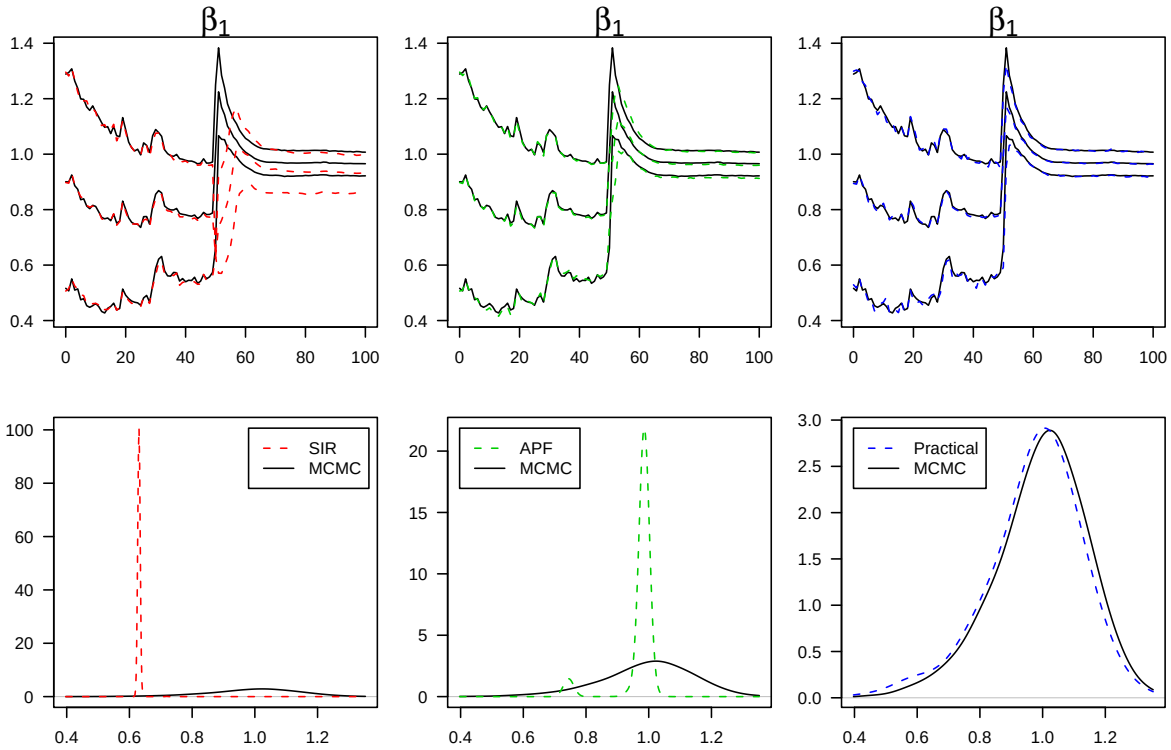

Fig. 4. AR(1) plus noise model with an outlier of $y_t = 6$ at time 50. Ratio of filter bias to posterior standard deviation, shown at times $t = 40, 50, 51, \ldots, 55, 60$. Each boxplot is based on 200 simulated datasets. Left: Storvik's SIR particle filter algorithm with $N = 10000$. Center: Storvik's APF algorithm with $N = 10000$. Right: practical filter with $(N, G, k) = (1000, 5, 15)$. Note that the vertical scales are different for each plot. To allow comparison of the algorithms, we include a vertical bar on the right of each plot which corresponds to the same interval in all plots on a given row.

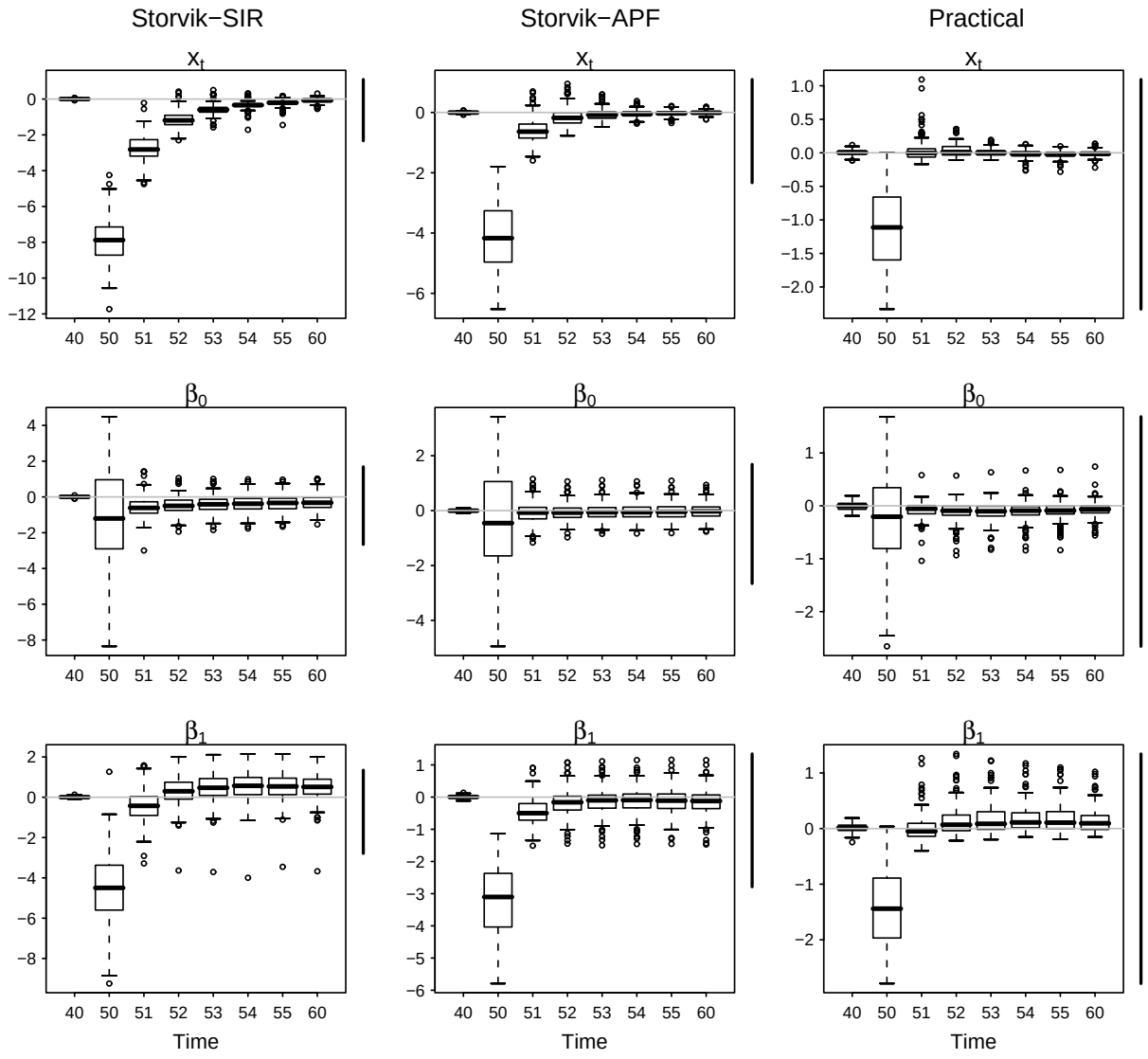

Fig. 5. Dynamic spatio-temporal model. Filtered means and 95% credible intervals for the state variable $\mathbf{x}_{t5}$ and the evolution parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$. The dashed lines represent the practical filter with $(N, G, k) = (100, 3, 10)$, while the solid lines represent a full MCMC with 1000 iterations.

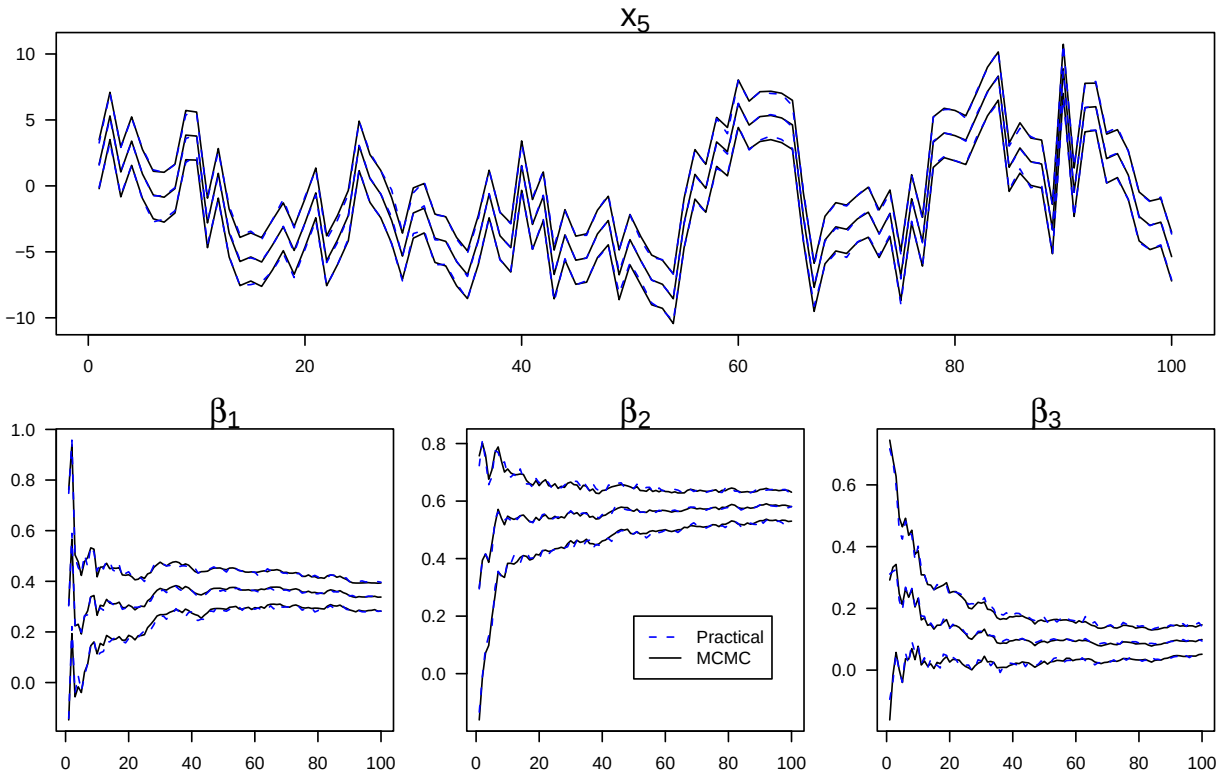

Fig. 6. Dynamic spatio-temporal model with an outlier at time 25. Filtered means and 95% credible intervals (top) and posterior distributions at time 25 (bottom) for the transition parameter β_2. Left: Storvik's SIR particle filter algorithm with $N = 10000$. Center: Storvik's APF algorithm with $N = 10000$. Right: practical filter with $(N, G, k) = (100, 3, 10)$. Solid lines in all panels are from a full MCMC with 1000 iterations.