Maximum Likelihood Parameter Estimation in General State-Space Models using Particle Methods

[1]George Poyiadjis, [2]Arnaud Doucet and [1]Sumeetpal S. Singh
[1] Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK.
[2] Departments of Computer Science and Statistics, University of British Columbia, Vancouver, BC, Canada.
Email: gp243,sss40@cam.ac.uk - arnaud@stat.ubc.ca

KEY WORDS: particle filter ; parameter estimation ; general state-space model; filter derivative ; hahn-jordan decomposition.

Abstract

A large number of time series can be described by non-linear, non-Gaussian state-space models. While state estimation for these models is now routinely performed using particle filters, maximum likelihood estimation of the model parameters is much more challenging. In this paper, we present new numerical methods to approximate the derivative of the optimal filter. We use this to perform batch and recursive maximum likelihood parameter estimation and tracking by maximizing the likelihood through a gradient ascent method. We generalize the method to include the second derivative of the optimal filter. This provides estimates of the Hessian of the likelihood and can be used to accelerate the gradient ascent method.

1. Introduction

Many time series problems arising in statistics, engineering and applied sciences are concerned with the estimation of the state of a dynamic model when only inaccurate observations are available. Most real-world problems are nonlinear and non-Gaussian, therefore optimal state estimation in such problems does not admit a closed form solution. Recently, there has been a surge of interest in Sequential Monte Carlo (SMC) methods, also known as particle filtering methods, to perform sequential state estimation in non-linear non-Gaussian models [7], [8], [9], [12], [15]. SMC methods are a set of simulation-based techniques that recursively generate and update a set of weighted samples, which provide approximations to the posterior probability distributions of interest. Under the assumption that the model parameters are known, numerous SMC algorithms have been proposed over the last decade; see [8] for a review. In real-world applications however, the model parameters denoted θ, are often unknown and also need to be estimated from the data. Maximum likelihood (ML) parameter estimation using SMC methods still remains an open problem, despite various earlier attempts in the literature.

The majority of the proposed SMC-based parameter estimation methods rely on augmenting the hidden state to include the unknown parameter and casting the problem as a filtering one [10], [16], [20]. Static parameter estimation with SMC is then implemented by either introducing artificial dynamics for the parameters or MCMC rejuvenation steps. The latter method is more elegant, since the model of interest is not artificially altered. However, the MCMC steps rely on sufficient statistics that are based on an approximation of the path posterior density $p_\theta(x_{0:n}|Y_{0:n})$ of the hidden process up to time n given the observations up to time n [1]. This density cannot be properly approximated using SMC methods, for a fixed number of particles, and the sufficient statistics degrade over time due to error accumulation [1].

In this paper we present an original maximum likelihood method that is based on a direct particle approximation of the derivative of the optimal filter. Previous attempts to approximate the filter derivative using particle methods - e.g. [5] [11] and [6] - were based implicitly on the sequence of path densities $\{p_\theta(x_{0:n}|Y_{0:n})\}$. As in the case of filtering-based parameter estimation, the approximation errors they produce increase with the data length. The methods we proposed here to approximate the filter derivative are based on the sequence of marginal distributions $\{p_\theta(x_n|Y_{0:n})\}$ and hence do not suffer from the aforementioned problem. We use the filter derivative approximation to compute the log-likelihood gradient and we combine it with a gradient ascent algorithm to generate maximum likelihood estimates of the model parameters. The approach is generalized to compute a particle approximation to the second derivative of the filter. This leads to an estimate of the Hessian of the likelihood that can be used to scale the gradient components and accelerate the convergence of the gradient algorithm. Additionally it may allow us to compute confidence regions for the estimated parameters.

The rest of the paper is organized as follows: In Section 2 the statistical model of interest is presented and the optimal filter and its first and second derivatives are described. In Section 3 we review the particle filter algorithm and derive particle methods for the derivatives of the filter. In

[1]For the rest of this paper we will adopt the following notation: for any sequence $\{z_k\}$ and random process $\{Z_k\}$, we define $z_{i:j} = (z_i, z_{i+1}, \ldots, z_j)$ and $Z_{i:j} = (Z_i, Z_{i+1}, \ldots, Z_j)$, respectively.

Section 4 we describe how the first and second filter derivative approximations can be used to perform ML parameter estimation in a recursive and a batch manner. Section 5 presents simulation results showing the performance of the proposed algorithm. Finally in Section 6 we discuss the results and provide some concluding remarks.

2. Optimal Filter and its Derivatives

2.1 State-Space Models

Let $\{X_n\}_{n\geq 0}$ and $\{Y_n\}_{n\geq 0}$ be $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_y}$-valued stochastic processes defined on a measurable space $(\Omega, \mathcal{F})$. These stochastic processes depend on a parameter $\theta \in \Theta$, where Θ is an open subset of $\mathbb{R}^{n_\theta}$. The process $\{X_n\}_{n\geq 0}$ is an unobserved (hidden) Markov process of initial density μ; i.e. $X_0 \sim \mu$, and a Markov transition density $f_\theta(x'|x)$; i.e.

$$X_{n+1}|X_n = x \sim f_\theta(\cdot|x). \tag{1}$$

Although $\{X_n\}_{n\geq 0}$ is unknown, it is partially observed through the observation process $\{Y_n\}_{n\geq 0}$. It is assumed that the observations conditioned upon $\{X_n\}_{n\geq 0}$ are independent with marginal density $g_\theta(y|x)$; i.e.

$$Y_n|X_n = x \sim g_\theta(\cdot|x). \tag{2}$$

This class of models includes many nonlinear and non-Gaussian time series models such as

$$X_{n+1} = \varphi_\theta(X_n, V_{n+1}),\ Y_n = \psi_\theta(X_n, W_n)$$

where $\{V_n\}_{n\geq 1}$ and $\{W_n\}_{n\geq 0}$ are mutually independent sequences of independent random variables and φ_θ, ψ_θ determine the evolution of the state and observation processes.

2.2 Optimal Filter Derivatives

Assume for the time being that θ is known. In such a case, sequential inference about the hidden process X_n is typically based on the sequence of joint posterior distributions $\{p_\theta(x_{0:n}|Y_{0:n})\}$. This summarizes all the relevant information available about $X_{0:n}$, up to time n. Using an importance sampling approach with an arbitrary important density $q_\theta(x_n|Y_n, x_{n-1})$, whose support includes the support of $g_\theta(Y_n|x_n)\, f_\theta(x_n|x_{n-1})$, it can be easily shown that the joint posterior density satisfies the recursion

$$p_\theta(x_{0:n}|Y_{0:n}) = \frac{\alpha_\theta(x_{n-1:n}, Y_n)}{p_\theta(Y_n|Y_{0:n:-1})} q_\theta(x_n|Y_n, x_{n-1})\, p_\theta(x_{0:n-1}|Y_{0:n-1}), \tag{3}$$

where the importance weights are given by

$$\alpha_\theta(x_{n-1:n}, Y_n) = \frac{g_\theta(Y_n|x_n)\, f_\theta(x_n|x_{n-1})}{q_\theta(x_n|Y_n, x_{n-1})}. \tag{4}$$

In most problems, we are interested in the marginal $p_\theta(x_n|Y_{0:n})$, which is known as the filtering density. This can be expressed as

$$p_\theta(x_n|Y_{0:n}) \propto \int \alpha_\theta(x_{n-1:n}, Y_n)\, q_\theta(x_n|Y_n, x_{n-1})\, p_\theta(x_{n-1}|Y_{0:n-1})\, dx_{n-1} \tag{5}$$

In applications such as parameter estimation and stochastic control, we are often interested in optimizing different performance criteria that require an approximation of the filter derivatives. In the context of parameter estimation, we will consider the first two derivatives of the optimal filter with respect to θ, namely $\nabla p_\theta(x_n|Y_{0:n})$ and $\nabla^2 p_\theta(x_n|Y_{0:n})$. To simplify the notation, let

$$p_\theta(x_n|Y_{0:n}) \triangleq \frac{\xi(x_n, Y_{0:n})}{\int \xi(x_n, Y_{0:n})\, dx_n} \tag{6}$$

where

$$\xi_\theta(x_n, Y_{0:n}) = g_\theta(Y_n|x_n) \int f_\theta(x_n|x_{n-1})\, p_\theta(x_{n-1}|Y_{0:n-1})\, dx_{n-1} \tag{7}$$
$$= g_\theta(Y_n|x_n)\, p_\theta(x_n|Y_{0:n-1}).$$

Under regularity assumptions, the first and second derivative of (6) leads to the following recursions[2],

$$\nabla p_\theta(x_n|Y_{0:n}) = \frac{\nabla \xi_\theta(x_n, Y_{0:n})}{\int \xi_\theta(x_n, Y_{0:n})\, dx_n} - p_\theta(x_n|Y_{0:n}) \frac{\int \nabla \xi_\theta(x_n, Y_{0:n})\, dx_n}{\int \xi_\theta(x_n, Y_{0:n})\, dx_n} \tag{8}$$

and

$$\nabla^2 p_\theta(x_n|Y_{0:n}) = \frac{\nabla^2 \xi_\theta(x_n, Y_{0:n})}{\int \xi_\theta(x_n, Y_{0:n})\, dx_n} - 2\nabla p_\theta(x_n|Y_{0:n}) \times \frac{\int \nabla \xi_\theta(x_n, Y_{0:n})\, dx_n}{\int \xi_\theta(x_n, Y_{0:n})\, dx_n} - p_\theta(x_n|Y_{0:n}) \frac{\int \nabla^2 \xi_\theta(x_n, Y_{0:n})\, dx_n}{\int \xi_\theta(x_n, Y_{0:n})\, dx_n}, \tag{9}$$

where

$$\nabla \xi_\theta(x_n, Y_{0:n}) = g_\theta(Y_n|x_n) \int f_\theta(x_n|x_{n-1}) p_\theta(x_{n-1}|Y_{0:n-1}) \times [\nabla \log g_\theta(Y_n|x_n) + \nabla \log f_\theta(x_n|x_{n-1})]\, dx_{n-1} + g_\theta(Y_n|x_n) \int f_\theta(x_n|x_{n-1})\, \nabla p_\theta(x_{n-1}|Y_{0:n-1})\, dx_{n-1} \tag{10}$$

[2]The 1st derivative $\nabla p_\theta(x_n|Y_{0:n})$ is an $n_\theta \times 1$ vector, where the i^{th} entry is given by $\frac{\partial p_\theta(x_n|Y_{0:n})}{\partial \theta_i}$. The 2nd derivative $\nabla^2 p_\theta(x_n|Y_{0:n})$ is an $n_\theta \times n_\theta$ matrix, where the $(i,j)^{th}$ entry is given by $\frac{\partial^2 p_\theta(x_n|Y_{0:n})}{\partial \theta_i \partial \theta_j}$.

and

$$\begin{aligned}\nabla^2\xi_\theta(x_n, Y_{0:n}) &= g_\theta(Y_n|x_n)\int f_\theta(x_n|x_{n-1})\\ &\times\Big\{[\nabla\log g_\theta(Y_n|x_n)+\nabla\log\ f_\theta(x_n|x_{n-1})]^2\\ &+\nabla^2\log g_\theta(Y_n|x_n)+\nabla^2\log\ f_\theta(x_n|x_{n-1})\Big\}\\ &\times p_\theta(x_{n-1}|Y_{0:n-1})dx_{n-1}\\ &+2g_\theta(Y_n|x_n)\int f_\theta(x_n|x_{n-1})\,[\nabla\log g_\theta(Y_n|x_n)\\ &+\nabla\log\ f_\theta(x_n|x_{n-1})]\,\nabla p_\theta(x_{n-1}|Y_{0:n-1})\ dx_{n-1}\\ &+g_\theta(Y_n|x_n)\int f_\theta(x_n|x_{n-1})\ \nabla^2 p_\theta(x_{n-1}|Y_{0:n-1})\ dx_{n-1}.\end{aligned} \tag{11}$$

Except in some simple cases, no closed-form expression can be obtained for either of the above recursions and one typically resorts to numerical approximations. Our main objective in this paper is to derive particle methods to approximate $\nabla p_\theta\left(x_n|\,Y_{0:n}\right)$ and $\nabla^2 p_\theta(x_n|Y_{0:n})$.

3. Particle Methods for the Filter Derivatives

3.1 Particle Filters

Particle methods are widely used to numerically approximate the filtering recursion in (3) by means of a weighted empirical distribution of a set of $N \gg 1$ samples, termed as particles. This empirical distribution is propagated sequentially as follows. Assume that at time $n-1$ a set of particles $X_{0:n-1}^{(1:N)} \triangleq \left[X_{0:n-1}^{(1)},\ldots,X_{0:n-1}^{(N)}\right]$ with corresponding weights $\widetilde{a}_{n-1}^{(1:N)} \triangleq \left[\widetilde{a}_{n-1}^{(1)},\ldots,\widetilde{a}_{n-1}^{(N)}\right]$ are available, with $\sum_{j=1}^N \widetilde{a}_{n-1}^{(j)} = 1$. We further assume that this weighted particle set is distributed approximately according to the joint density $p_\theta\left(x_{0:n-1}|\,Y_{0:n-1}\right)$. A standard way to approximate the joint density at the next time step is to extend the path using

$$X_n^{(i)} \sim q_\theta\left(x_n|\,Y_{0:n}\right) = \sum_{i=1}^N \widetilde{a}_{n-1}^{(i)} q_\theta\left(\cdot|\ Y_n, X_{n-1}^{(i)}\right). \tag{12}$$

Sampling from (12) is achieved by first sampling the discrete index i using a standard resampling algorithm, such as stratified or multinomial resampling. Then, the new particle $X_n^{(i)}$ is generated according to

$$X_n^{(i)} \sim q_\theta\left(\cdot|\ Y_n, X_{n-1}^{\varphi(i)}\right), \tag{13}$$

where $\varphi(i)$ is the discrete index obtained from the resampling mechanism. Note that the new set of equally weighted particles $X_{0:n}^{(1:N)} = \left[X_{0:n}^{(1)},\ldots,X_{0:n}^{(N)}\right]$, with $X_{0:n}^{(i)} = \left(X_{0:n-1}^{\varphi(i)}, X_n^{(i)}\right)$ will be approximately distributed according to the joint density $q_\theta\left(x_n|\,Y_n, x_{n-1}\right)p_\theta\left(x_{0:n-1}|\,Y_{0:n-1}\right)$. Substitution of this approximation into (3) leads to the updated empirical distribution

$$\widehat{p}_\theta\left(x_{0:n}|\,Y_{0:n}\right) = \sum_{i=1}^N \widetilde{a}_n^{(i)}\delta\left(x_{0:n} - X_{0:n}^{(i)}\right), \tag{14}$$

where

$$a_n^{(i)} = \alpha_\theta\left(X_{n-1}^{\varphi(i)}, X_n^{(i)}, Y_n\right) \text{ and } \widetilde{a}_n^{(i)} = \frac{a_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}}. \tag{15}$$

In practice, a particle approximation of the filtering density $p_\theta\left(x_n|\,Y_{0:n}\right)$ in (5) is obtained by marginalization of (14).

3.2 Filter derivative approximations

The filter derivatives $\nabla p_\theta\left(x_n|\,Y_{0:n}\right)$ and $\nabla^2 p_\theta\left(x_n|\,Y_{0:n}\right)$ are signed measures; i.e. they can take positive and negative values and integrate to zero. Empirical approximations of these measures using particle methods are still possible, provided that one uses the same set of particles as in the filter approximation, but with different weights. This idea that was first introduced in [5], computes the particle approximations $\widehat{\nabla p}_\theta\left(x_n|\,Y_{0:n}\right)$ and $\widehat{\nabla^2 p}_\theta\left(x_n|\,Y_{0:n}\right)$ by propagating the weighted particles on the path space and marginalizing the expressions

$$\widehat{\nabla p}_\theta\left(x_{0:n}|\,Y_{0:n}\right) = \sum_{i=1}^N \widetilde{a}_n^{(i)}\beta_n^{(i)}\delta\left(x_{0:n} - X_{0:n}^{(i)}\right), \tag{16}$$

$$\widehat{\nabla^2 p}_\theta\left(x_{0:n}|\,Y_{0:n}\right) = \sum_{i=1}^N \widetilde{a}_n^{(i)}\lambda_n^{(i)}\delta\left(x_{0:n} - X_{0:n}^{(i)}\right), \tag{17}$$

where $\beta_n^{(i)}, \lambda_n^{(i)}$ can be positive or negative.

As already mentioned, the drawback of this approach stems from the fact that it relies on the path, which is a space of growing dimensions. Consequently, as the length n of the path increases, the variance of $\widehat{\nabla p}_\theta\left(x_{0:n}|\,Y_{0:n}\right)$, $\widehat{\nabla^2 p}_\theta\left(x_n|\,Y_{0:n}\right)$ will increase and the approximations of their marginals will degrade severely. Another effect that deteriorates the performance of a path-based particle algorithm results from the arrangement of the particle mass on the state space. The derivative of a probability measure is a signed measure ν that can be expressed as $\nu = c\,(\pi_1 - \pi_2)$, where π_1, π_2 are two probability measures and c is a non-negative constant. This approach, known as weak derivative decomposition, allows an arbitrary number of possible decompositions for a given signed measure. The path-based particle method discussed in the previous section decomposes the derivative signed measures by approximating the two probability measures π_1 and π_2 by a set of positively and negatively weighted particles on overlapping regions

of the state space. To see this, consider a point x' and a neighborhood of it, $B_{x'}$, for which the sign measure satisfies $\nu > 0$. If we use the particle representation of (16) to approximate ν, an estimate of $\int I_{B_{x'}}(x)\,\nu(x)\,dx$ becomes equal to $\sum_{i=1}^{N} \widetilde{a}^{(i)}\beta^{(i)} I_{B_{x'}}(X^{(i)})$. While this is a valid approximation, we may have that for two particles k and l belonging to $B_{x'}$, the weights are not of the same sign, i.e., $\widetilde{a}_n^{(k)}\beta^{(k)} < 0$ while $\widetilde{a}_n^{(l)}\beta^{(l)} > 0$. In such a case we say that the particles *mix*, as illustrated in the top plot of Figure 1 for the case where $\mathbb{R}^{n_x} = \mathbb{R}^{n_\theta} = \mathbb{R}$. This implies that many particles with opposite signs can end up approximating regions of the state space that have low total mass (see for example low mass region around the value $x = 1$ in the top plot of Figure 1). This effect builds up due to the sequential nature of the algorithm and the implementation becomes less accurate and inefficient as the data length n increases.

We propose here an original method to approximate the optimal filter derivatives that is based on a direct point-wise approximation of (8) and (9) and hence does not suffer from the limitations discussed in the previous paragraphs. This method essentially integrates analytically a discrete approximation of the latent variable and will therefore have a lower variance. From a weak derivatives point of view, this is equivalent to a a particle implementation of a *Hahn-Jordan* decomposition, which ensures that the probability measures of the decomposition are concentrated on disjoint regions of the state. As a result, the algorithm does not suffer from mixing of the positively and negatively weighted particles, as illustrated in the bottom plot of Figure (1).

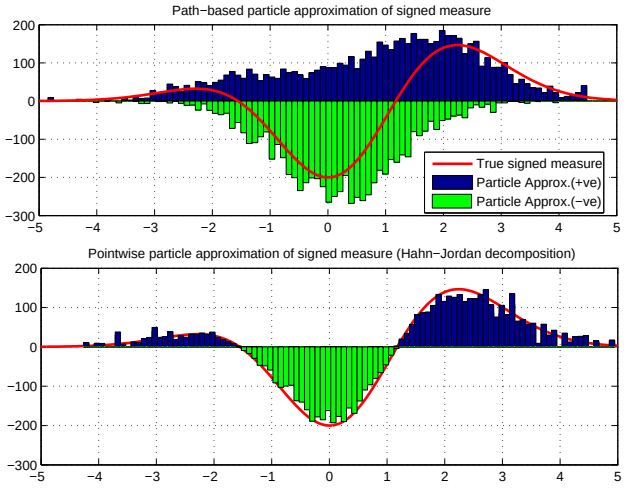

Figure 1: Top plot: Histogram representation of a path-based particle approximation of $\widehat{\nabla p}_\theta(x_n|Y_{0:n})$ w.r.t. a one-dimensional parameter θ. Bottom plot: Point-wise particle approximation of the same signed measure that maintains the positive and negative weights on separate regions of the state support (Hahn-Jordan decomposition).

3.3 Particle algorithm

In this section we describe the proposed sequential method to approximate the first two derivatives of the optimal filter. Assume that at time $n-1$, we have particle approximations of $p_\theta(x_{n-1}|Y_{0:n-1})$, $\nabla p_\theta(x_{n-1}|Y_{0:n-1})$ and $\nabla^2 p_\theta(x_{n-1}|Y_{0:n-1})$ of the form

$$\widehat{p}_\theta(x_{n-1}|Y_{0:n-1}) = \sum_{i=1}^{N} \widetilde{a}_{n-1}^{(i)} \delta\left(x_{n-1} - X_{n-1}^{(i)}\right), \quad (18)$$

$$\widehat{\nabla p}_\theta(x_{n-1}|Y_{0:n-1}) = \sum_{i=1}^{N} \widetilde{a}_{n-1}^{(i)} \beta_n^{(i)} \delta\left(x_{n-1} - X_{n-1}^{(i)}\right), \quad (19)$$

$$\widehat{\nabla^2 p}_\theta(x_{n-1}|Y_{0:n-1}) = \sum_{i=1}^{N} \widetilde{a}_{n-1}^{(i)} \lambda_{n-1}^{(i)} \delta\left(x_{n-1} - X_{n-1}^{(i)}\right). \quad (20)$$

Substitution of these into (7), (10) and (11) leads to the following point-wise approximations

$$\widetilde{\xi}_\theta(x_n, Y_{0:n}) = \sum_{k=1}^{N} \widetilde{a}_{n-1}^{(k)}\; g_\theta(Y_n|x_n) f_\theta\left(x_n|X_{n-1}^{(k)}\right), \quad (21)$$

$$\widetilde{\nabla\xi}_\theta(x_n, Y_{0:n}) = \sum_{k=1}^{N} \widetilde{a}_{n-1}^{(k)} g_\theta(Y_n|x_n)\; f_\theta\left(x_n|X_{n-1}^{(k)}\right) \times \left[\nabla \log g_\theta(Y_n|x_n) + \nabla \log\; f_\theta\left(x_n|X_{n-1}^{(k)}\right) + \beta_{n-1}^{(k)}\right], \quad (22)$$

and

$$\widetilde{\nabla^2\xi}_\theta(x_n, Y_{0:n}) = \sum_{k=1}^{N} \widetilde{a}_{n-1}^{(k)} g_\theta(Y_n|x_n)\; f_\theta\left(x_n|X_{n-1}^{(k)}\right) \times \left\{\left[\nabla \log g_\theta(Y_n|x_n) + \nabla \log\; f_\theta\left(x_n|X_{n-1}^{(k)}\right)\right]^2 + \nabla^2 \log g_\theta(Y_n|x_n) + \nabla^2 \log\; f_\theta\left(x_n|X_{n-1}^{(k)}\right) + 2\beta_{n-1}^{(k)}\left[\nabla \log g_\theta(Y_n|x_n) + \nabla \log f_\theta\left(x_n|X_{n-1}^{(k)}\right)\right] + \lambda_{n-1}^{(k)}\right\} \quad (23)$$

As in the standard particle filter, we generate a set of particles $X_n^{(i)}$, for $i = 1, ..., N$, using (12). Evaluating the point-wise approximations in (21), (22) and (23) at points $X_n^{(i)}$ yields the following particle approximations

$$\widehat{\xi}_\theta(x_n, Y_{0:n}) = \frac{1}{N}\sum_{i=1}^{N} a_n^{(i)} \delta_{X_n^{(i)}}(x_n), \quad (24)$$

$$\widehat{\nabla\xi}_\theta(x_n, Y_{0:n}) = \frac{1}{N}\sum_{i=1}^{N} \rho_n^{(i)} \delta_{X_n^{(i)}}(x_n) \quad \text{and} \quad (25)$$

$$\widehat{\nabla^2}\xi_\theta(x_n, Y_{0:n}) = \frac{1}{N}\sum_{i=1}^{N} \pi_n^{(i)} \delta_{X_n^{(i)}}(x_n), \tag{26}$$

where $a_n^{(i)} = \frac{\widetilde{\xi}_\theta(X_n^{(i)}, Y_{0:n})}{q_\theta \;\; X_n^{(i)}|Y_{0:n}}$, $\rho_n^{(i)} = \frac{\widetilde{\nabla\xi}_\theta(X_n^{(i)}, Y_{0:n})}{q_\theta \;\; X_n^{(i)}|Y_{0:n}}$ and $\pi_n^{(i)} = \frac{\widetilde{\nabla^2\xi}_\theta(X_n^{(i)}, Y_{0:n})}{q_\theta \;\; X_n^{(i)}|Y_{0:n}}$. Substitution of the last three approximations into (6), (8) and (9) gives

$$\widehat{p}_\theta(x_n|Y_{0:n}) = \sum_{i=1}^{N} \widetilde{a}_n^{(i)} \delta_{X_n^{(i)}}(x_n), \tag{27}$$

$$\widehat{\nabla p}_\theta(x_n|Y_{0:n}) = \sum_{i=1}^{N} \widetilde{a}_n^{(i)} \beta_n^{(i)} \delta_{X_n^{(i)}}(x_n) \quad \text{and} \tag{28}$$

$$\widehat{\nabla^2 p}_\theta(x_n|Y_{0:n}) = \sum_{i=1}^{N} \widetilde{a}_n^{(i)} \;\; \lambda_n^{(i)} \delta_{X_n^{(i)}}(x_n), \tag{29}$$

where

$$\widetilde{a}_n^{(i)} = \frac{a_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}}, \quad \widetilde{a}_n^{(i)}\beta_n^{(i)} = \frac{\rho_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}} - \widetilde{a}_n^{(i)} \frac{\sum_{j=1}^N \rho_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}},$$

$$\widetilde{a}_n^{(i)}\lambda_n^{(i)} = \frac{\pi_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}} - 2\widetilde{a}_n^{(i)}\beta_n^{(i)} \frac{\sum_{j=1}^N \rho_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}} - \widetilde{a}_n^{(i)} \frac{\sum_{j=1}^N \pi_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}}.$$

Note that (27) gives a filtering byproduct of the algorithm. Compared to standard path-based particle filters, the point-wise particle filter $\widehat{p}_\theta(x_n|Y_{0:n})$ requires $O\left(N^2\right)$ operations instead of $O\left(N\right)$. However, for a fixed number of particles N, it will outperform path-based methods due to the analytical integration involved.

4. ML Parameter Estimation

Let us now consider the general state space model of section 2.1, where the model parameter θ is unknown. We will assume that the model that generates the observation sequence $\{Y_n\}_{n\geq 0}$ evolves according to a true but unknown static parameter θ^*, i.e.

$$X_n|X_{n-1} = x_{n-1} \sim f_{\theta^*}(\,.\,|x_{n-1}) \tag{30}$$

$$Y_n|X_n = x_n \sim g_{\theta^*}(\,.\,|x_n). \tag{31}$$

Our objective is to identify θ^* based on $\{Y_n\}_{n\geq 0}$. We propose here two gradient algorithms to perform maximum likelihood estimation. These are based on a gradient ascent method that utilizes the estimates of the derivatives of the filter that were presented in the previous section. The first method is a recursive algorithm that updates the parameter estimate as soon as a new observation is received. This is based on the maximization of an average log-likelihood criterion and requires a large number of observations to be available. A batch version of the algorithm is also presented. This directly maximizes the log-likelihood of some available set of observations $Y_{0:n}$.

4.1 Recursive ML

A standard approach to Recursive ML (RML) estimation considers a series of log-likelihood functions $\{\log p_\theta(Y_{0:k})\}_{k\geq 0}$, where $\log p_\theta(Y_{0:k}) = \sum_{n=0}^{k} \log p_\theta(Y_n\,|Y_{0:n-1}\,)$ [17]. The expression $p_\theta(Y_n\,|Y_{0:n-1}\,)$ is known as the predictive likelihood and can be written as

$$p_\theta(Y_n\,|Y_{0:n-1}\,) = \int\int g_\theta(Y_n|x_n) f_\theta(x_n|x_{n-1})\; p_\theta(x_{n-1}|Y_{0:n-1})\; dx_{n-1:n} \tag{32}$$

Under suitable regularity conditions described in [21] it can be shown that the average log-likelihood converges to the following limit

$$\lim_{k\to\infty} \frac{1}{k+1}\sum_{n=0}^{k} \log p_\theta(Y_n\,|Y_{0:n-1}\,) = l\left(\theta\right), \tag{33}$$

where $l\left(\theta\right)$ is given by

$$l\left(\theta\right) = \int\int_{\mathbb{R}^{n_y}\times\mathcal{P}(\mathbb{R}^{n_x})} \log\left(\int g_\theta\left(\,y|\,x\right)\mu\left(x\right)dx\right)\lambda_{\theta,\theta^*}\left(dy, d\mu\right).$$

Here $\mathcal{P}\left(\mathbb{R}^{n_x}\right)$ is the space of probability distributions on $\mathbb{R}^{n_x}$ and $\lambda_{\theta,\theta^*}\left(dy, d\mu\right)$ is the joint invariant distribution of the couple $\left(Y_n, p_\theta\left(x_n|Y_{0:n-1}\right)\right)$. Note that $\lambda_{\theta,\theta^*}\left(\cdot\right)$ is a function of both θ^* and θ, since the observation component evolves according to the true parameter θ^*, while the prediction filter component evolves according to θ.

Following the approach used in [14] for finite state space models, it can be shown that $l\left(\theta\right)$ admits θ^* as a global maximum. The function $l\left(\theta\right)$ does not have an analytical expression and we do not have access to it. Nevertheless, identification of θ^* can still be achieved based on the ergodicity property in (33), which provides us with a set of accessible functions $\log p_\theta(Y_n\,|Y_{0:n-1}\,)$ that converge to $l\left(\theta\right)$. One way to exploit this in order to maximize $l\left(\theta\right)$, is to use a Stochastic Approximation (SA) algorithm to update the parameter estimate at time n using the recursion

$$\theta_n = \theta_{n-1} + \gamma_n \nabla \log p_{\theta_{0:n-1}}(Y_n|Y_{0:n-1}), \tag{34}$$

where θ_{n-1} is the parameter estimate at time $n-1$ and $\nabla \log p_\theta(Y_n|Y_{0:n-1})$ denotes the gradient of $\log p_\theta(Y_n|Y_{0:n-1})$[3]. Provided that the step size θ_n is a positive non-increasing sequence, such that $\sum\gamma_n = \infty$ and $\sum\gamma_n^2 < \infty$, it can be shown that θ_n will converge to the set of (global or local) maxima of $l\left(\theta\right)$.

[3]The SA requires an estimate of $\nabla \log p_\theta(Y_n|Y_{0:n-1})$ with θ held fixed. In our problem, θ cannot be fixed, since we are estimating it recursively. However, since θ changes slowly, a standard approach [18] is to reuse the previous particle calculations that were based on $\theta_{0:n-2}$ and use the parameter estimate θ_{n-1} at time $n-1$.

The remaining step in the development of the algorithm is to obtain a numerical approximation to $\nabla \log p_\theta(Y_n|Y_{0:n-1})$. This follows directly from the the expressions for $p_\theta(x_n|Y_{0:n})$ and $\nabla p_\theta(x_n|Y_{0:n})$ in Section 3.3, since comparison of (7) and (32) gives $p_\theta(Y_n\,|Y_{0:n-1}) = \int \xi_\theta(x_n, Y_{0:n})dx_n$. Using the particle approximations of $\xi_\theta(x_n, Y_{0:n})$ and $\nabla\xi_\theta(x_n, Y_{0:n})$ in (24) and (25) we obtain

$$\nabla \log \widehat{p}_\theta\,(Y_n|Y_{0:n-1}) = \frac{\widehat{\nabla p}_\theta\,(Y_n|Y_{0:n-1})}{\widehat{p}_\theta\,(Y_n|Y_{0:n-1})} = \frac{\int \widehat{\nabla\xi}_\theta(x_n, Y_{0:n})dx_n}{\int \widehat{\xi}_\theta(x_n, Y_{0:n})dx_n} = \frac{\sum_{j=1}^N \rho_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}}.$$

4.1.1 ***Adaptive SA***

Adaptive steps The SA of (34) can be thought of as a stochastic generalization of the steepest descent method. Faster convergence can be achieved if one employs a Newtonian method that is based on an estimate of the Hessian of the objective function and leads to an asymptotically optimal search direction [3]. In general, estimation of the Hessian is non-trivial and finite difference approximations are typically used to approximate it [19].

In our framework, the Hessian of the log-likelihood can be straightforwardly estimated using the particle approximations of the optimal filter and its first and second derivatives in (27), (28) and (29). More specifically, the $n_\theta \times n_\theta$ Hessian matrix estimate at time n will be given by

$$\nabla^2 \log \widehat{p}_\theta\,(Y_n|Y_{0:n-1}) = \frac{\widehat{\nabla^2 p_\theta}(Y_n|Y_{0:n-1})}{\widehat{p}_\theta(Y_n|Y_{0:n-1})} - \left(\frac{\widehat{\nabla p_\theta}(Y_n|Y_{0:n-1})}{\widehat{p}_\theta(Y_n|Y_{0:n-1})}\right)^2 = \frac{\sum_{j=1}^N \pi_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}} - \left(\frac{\sum_{j=1}^N \rho_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}}\right)^2.$$

This allows one to compute the asymptotic value of the Hessian using, for example, a recursion of the form

$$\overline{H}_n = \overline{H}_{n-1} + \frac{1}{n+1}\left(\widehat{H}_n - \overline{H}_{n-1}\right), \qquad (35)$$

where $\widehat{H}_n = \nabla^2 \log \widehat{p}_\theta\,(Y_n|Y_{0:n-1})$. This is simply a recursive calculation of the sample mean $\overline{H}_n$ up to time n. By construction, the true Hessian is a negative definite, symmetric matrix whose inverse can provide an adaptive step in (34). Direct inversion of the estimated value $\overline{H}_n$ will be possible only if this matrix is negative definite. In practice this is usually ensured by projecting $\overline{H}_n$ onto the set of negative definite matrices - see [4] and [19] for details.

The Newton-type SA recursion that can replace (34) will take the form

$$\theta_n = \theta_{n-1} + \gamma_n \overline{H}_n^{-1} \nabla \log \widehat{p}_{\theta_{0:n-1}}(Y_n|Y_{0:n-1}). \qquad (36)$$

This adaptive SA is particularly attractive, in terms of convergence acceleration, in the terminal phase of the algorithm, where the steepest descent-type method slows down.

Confidence Regions From a practical point of view, it is often desirable to assess the accuracy of the parameter estimate by means of confidence intervals or more generally confidence regions. In principle, central limit theorems that have been established for a number of standard SA algorithms allow the computation of confidence regions for the estimates - see [3] for detailed results. One of the difficulties is that the covariance matrix of the limiting multivariate normal distribution depends on the inverse of the Hessian of the objective function. The proposed method provides estimates for these quantity through (35).

4.1.2 RML Parameter Estimation Algorithm

The Recursive ML estimation is summarized as follows:

---

1. *Sampling Step*

For $i = 1, \ldots, N$, sample[4]

$X_n^{(i)} \sim q_{\theta_{0:n-1}}(\cdot\,|Y_{0:n}) = \sum_{i=1}^N \widetilde{a}_{n-1}^{(i)} q_{\theta_{n-1}}\left(\cdot \mid Y_n, X_{n-1}^{(i)}\right)$

2. *Weight Calculation*

- Compute $a_n^{(i)} = \frac{\widetilde{\xi}_{\theta_{0:n-1}}(X_n^{(i)}, Y_{0:n})}{q_{\theta_{0:n-1}}\ X_n^{(i)}|Y_{0:n}}$,

  $\rho_n^{(i)} = \frac{\widetilde{\nabla\xi}_{\theta_{0:n-1}}(X_n^{(i)}, Y_{0:n})}{q_{\theta_{0:n-1}}\ X_n^{(i)}|Y_{0:n}}$ and

  $\pi_n^{(i)} = \frac{\widetilde{\nabla^2\xi}_{\theta_{0:n-1}}(X_n^{(i)}, Y_{0:n})}{q_{\theta_{0:n-1}}\ X_n^{(i)}|Y_{0:n}}$ using (21), (22) and (23).
- Compute the weights $\widetilde{a}_n^{(i)} = \frac{a_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}}$,

  $\widetilde{a}_n^{(i)}\beta_n^{(i)} = \frac{\rho_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}} - \widetilde{a}_n^{(i)}\frac{\sum_{j=1}^N \rho_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}}$ and

  $\widetilde{a}_n^{(i)}\lambda_n^{(i)} = \frac{\pi_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}} - 2\widetilde{a}_n^{(i)}\beta_n^{(i)}\frac{\sum_{j=1}^N \rho_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}} - \widetilde{a}_n^{(i)}\frac{\sum_{j=1}^N \pi_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}}$

3. *Parameter Update Step*

- $\widehat{H}_n = \frac{\sum_{j=1}^N \pi_n^{(i)}}{\sum_{j=1}^N a_n^{(j)}} - \left(\frac{\sum_{j=1}^N \rho_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}}\right)^2$
- $\overline{H}_n = \overline{H}_{n-1} + \frac{1}{n+1}\left(\widehat{H}_n - \overline{H}_{n-1}\right)$, $\widetilde{H}_n = \Psi\left(\overline{H}_n\right)$
- $\theta_n = \theta_{n-1} + \gamma_n \widetilde{H}_n^{-1} \frac{\sum_{j=1}^N \rho_n^{(j)}}{\sum_{j=1}^N a_n^{(j)}}$

Remark 1: The function $\Psi\left(\overline{H}_n\right)$ is a mapping to the set of negative definite matrices, based on diagonal modifications to the Hessian.

Remark 2: This algorithm guarantees that $\sum_{i=1}^N \alpha_n^{(i)}\beta_n^{(i)} = \sum_{i=1}^N \alpha_n^{(i)}\lambda_n^{(i)} = 0$.

Remark 3: Even if $\theta_{n-1} \in \Theta$, it is possible that for the updated value we have $\theta_n \notin \Theta$. A standard approach to

[4] Note that in this approach the resampling step is included when we sample from $q_\theta\,(x_n\,|Y_{0:n})$.

prevent such divergence is to reproject the parameter value inside $\Theta = \prod_{\mu=1}^{n_\theta} \left[\theta_\mu^{\min}, \theta_\mu^{\max}\right]$.

4.2 Batch ML

In cases where a set of observations $Y_{0:n}$ is available, we describe here a batch (off-line) version (BML) of the previous algorithm. This algorithm maximizes the log-likelihood $\log p_\theta(Y_{0:n})$ using a SA recursion at iteration m given by

$$\theta_m = \theta_{m-1} + \gamma_m \nabla \log \widehat{p}_{\theta_{m-1}}(Y_{0:n}), \tag{37}$$

where $\nabla \log \widehat{p}_{\theta_{m-1}}(Y_{0:n})$ is an estimate of the derivative of the log-likelihood evaluated at point θ_{m-1}. This estimate can be obtained using a modified version of the RML method as follows: At iteration $m-1$, we run the RML algorithm from time 0 to time n by omitting the parameter update step and keeping the parameter value fixed at the current estimated value θ_{m-1}. At the end of the run, a Monte Carlo estimate of the derivative of the log-likelihood can be computed as

$$\nabla \log \widehat{p}_{\theta_{m-1}}(Y_{0:n}) = \sum_{k=0}^{n} \frac{\widehat{\nabla p}_{\theta_{m-1}}(Y_k|Y_{0:k-1})}{\widehat{p}_{\theta_{m-1}}(Y_k|Y_{0:k-1})} = \sum_{k=0}^{n} \frac{\sum_{j=1}^{N} \rho_k^{(j)}}{\sum_{j=1}^{N} a_k^{(j)}}$$

where $Y_{0:-1} = \varnothing$. This is used to update the parameter to θ_m, as given by (37).

5. Numerical Study

The RML and BML algorithms were tested, based on artificial and real observations.

5.1 Linear Gaussian State Space Model

We first consider the following scalar linear Gaussian state space model

$$X_{n+1} = \phi X_n + \sigma_V V_{n+1}, \; X_0 \sim \mathcal{N}\left(0, \frac{\sigma_V^2}{1-\phi^2}\right)$$
$$Y_n = X_n + \sigma_W W_n$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. We are interested in estimating the parameter $\theta \triangleq (\phi, \sigma_V, \sigma_W)$. In such a model, the optimal filter is given by the Kalman filter and exact expressions for the first and second derivative of the filter can be obtained. This allows us to compare our numerical methods with the ground truth. The RML algorithm was implemented using the optimal importance density $q_\theta(x_n|Y_n, x_{n-1}) \propto g_\theta(Y_n|x_n) f_\theta(x_n|x_{n-1})$ and $N = 1000$ particles. Figure 2 displays the analytical posterior density and its derivatives with respect to ϕ and compares them with the particle approximations we obtained. The analytical and numerical values

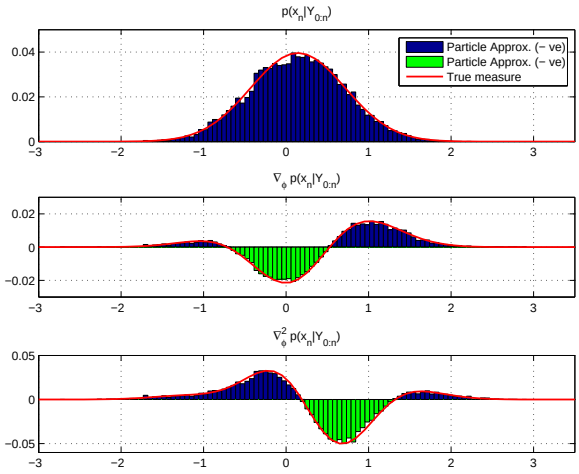

Figure 2: Linear Gaussian state space example: Analytical optimal filter and its first and second derivative w.r.t. ϕ and the particle approximations obtained using the proposed method.

of the score vector $\nabla \log p_\theta(Y_n|Y_{0:n-1})$ and the Hessian matrix $\nabla^2 \log p_\theta(Y_n|Y_{0:n-1})$ were compared up to $n = 10000$. These were almost indistinguishable. An example of the comparison results obtained for the component $\frac{\partial^2 \log p_\theta(Y_n|Y_{0:n-1})}{\partial \phi^2}$ is shown in Figure 3.

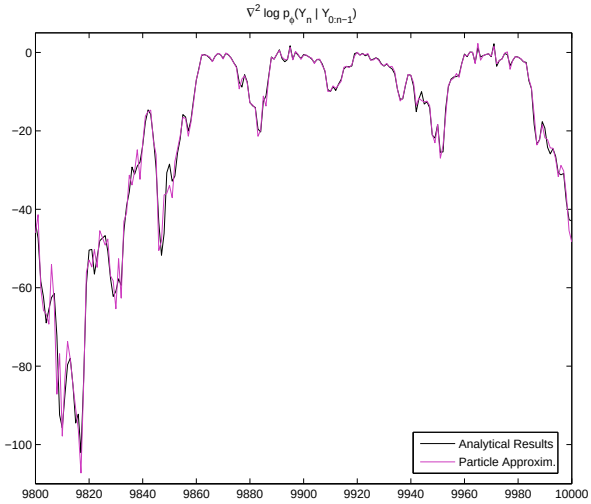

Figure 3: Analytical and numerical results for $\partial^2 \log p_\theta(Y_n|Y_{0:n-1}) / \partial \phi^2$ for the linear Gaussian state space model using $N = 1000$.

5.2 Stochastic Volatility Model

The RML algorithm was implemented using the following Stochastic Volatility model

$$X_{n+1} = \phi X_n + \sigma V_{n+1},\ X_0 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right)$$
$$Y_n = \beta \exp\left(X_n/2\right) W_n$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. We are interested in estimating the true parameter $\theta^* \triangleq (\sigma^*, \phi^*, \beta^*) = (0.35, 0.85, 0.65)$ from simulated data, where $\Theta = (0,\Xi) \times (-1,1) \times (0,\Xi)$ with $\Xi = 100$. We use $q_\theta(x_n|Y_n, x_{n-1}) = f_\theta(x_n|x_{n-1})$ and $N = 1000$ particles. As it can be seen for the results in Figure 4, the estimate converged to a value $\widehat{\theta}$ in the neighborhood of the true parameter.

We then applied our BML method to the pound/dollar daily exchange rates; see [9]. This time series consists of 945 data points. The parameter estimates for $M = 1000$ iterations using $N = 1000$ particles are shown in Figure 5. Our results are consistent with results obtained in [9].

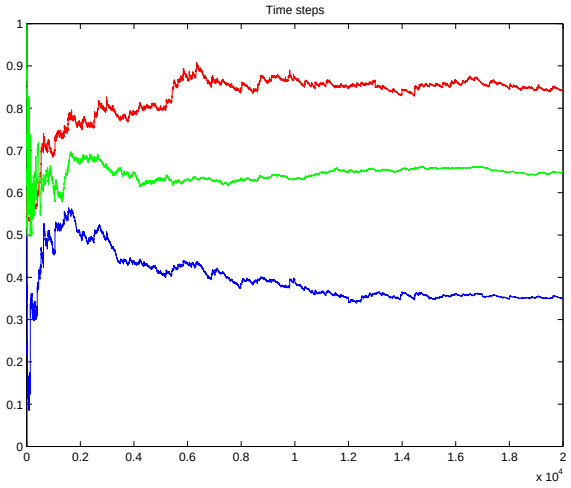

Figure 4: Sequence of RML parameter estimates for $\theta_n = (\sigma_n, \phi_n, \beta_n)$ and $N = 1000$. From top to bottom: ϕ_n, β_n and σ_n. The true values were $\theta^* = (0.35, 0.85, 0.65)$.

5.3 Parameter tracking

A unique advantage of the RML algorithm of section 4.1, is its ability to track variations in θ. A standard approach to track a time-varying parameter is to set the step-size to a small positive number γ, instead of a decreasing sequence γ_n [17]. The choice of value for γ will be a trade-off between tracking capability (large γ) and low estimation noise around the parameter (small γ). An example of the tracking performance of the RML algorithm based on the linear Gaussian state space model, having a time-varying drift parameter ϕ, is shown in Figure 6.

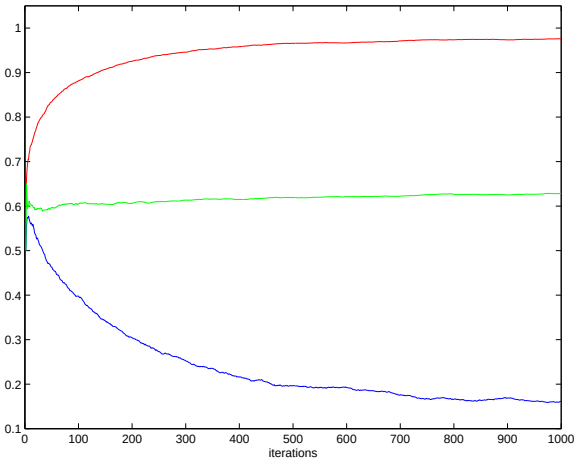

Figure 5: Sequence of BML parameter estimates for $\theta_m = (\sigma_m, \phi_m, \beta_m)$ and $N = 1000$. From top to bottom: ϕ_m, β_m and σ_m.

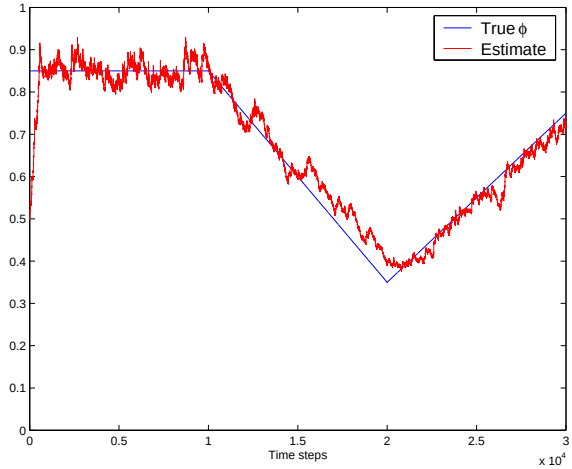

Figure 6: RML algorithm tracking performance for time-varying ϕ^*using $N = 1000$ particles.

6. Discussion

This paper has presented original particle methods to estimate the first and second derivative of the optimal filter in general state-space models. The methods use non-standard particle methods to approximate the Hahn-Jordan decomposition of the resultant signed measures. This allows the calculation of accurate approximations to the score vector and the Hessian matrix of the log-likelihood with respect to the model parameters. Based on this, we propose a recursive and a batch algorithm to perform ML parameter estimation using a gradient ascent method. The Hessian estimate can be used as an adaptive step-size in the gradient ascent recursion to provide faster convergence of the algorithm.

The computational cost of the proposed particle methods for the filter derivatives is quadratic in the number of particles. Fast computation methods can however be employed to address this issue [13]. The proposed methods can also be extended to the case where it is possible to integrate analytically a subset of the state variables, such as the class of partially observed linear Gaussian state-space models and conditionally linear Gaussian state-space models [2], [8]. Such extensions can provide efficient particle methods that reduce the variance of the Monte Carlo estimates.

References

[1] Andrieu C., Doucet A. and Tadic V.B. (2005) Online parameter estimation in general state space models, *Proc. IEEE CDC/ECC*

[2] Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed gaussian state space models. *J. Royal Statist. Soc.* B, **64**, 827-836.

[3] Benveniste, A., Métivier, M. and Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximation.* New York: Springer-Verlag.

[4] Bertsekas D. (1999), *Nonlinear Programming*, 2nd Edition, Athena Scientific.

[5] Cérou F., LeGland F. and Newton N.J. (2001) Stochastic particle methods for linear tangent equations. in *Optimal Control and PDE's - Innovations and Applications* (eds. J. Menaldi, E. Rofman & A. Sulem), pp. 231-240, IOS Press, Amsterdam.

[6] Doucet, A. and Tadić, V.B. (2003). Parameter estimation in general state-space models using particle methods. *Ann. Inst. Stat. Math.*, **55**, 409-422.

[7] Doucet A., Godsill S.J. and Andrieu C. (2000) On sequential Monte Carlo sampling methods for Bayesian filtering, *Statist. Comput.*, vol. 10, pp.197-208 .

[8] Doucet, A., de Freitas, J.F.G. and Gordon N.J. (eds.) (2001). *Sequential Monte Carlo Methods in Practice.* New York: Springer-Verlag.

[9] Durbin, J. and Koopman, S. J. (2000). Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives (with discussion). *J. R. Statist. Soc.* B, **62**, 3-56.

[10] Fearnhead P. (2002) MCMC, sufficient statistics and particle filter, *J. Comp. Graph. Stat.*, vol. 11, pp. 848-862.

[11] Guyader, A., LeGland, F. and Oudjane, N. (2003) A particle implementation of the recursive MLE for partially observed diffusions. *Proceedings of the 13th IFAC Symposium on System Identification*, 1305-1310.

[12] Kitagawa G. (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, **5**, 1-25.

[13] Klaas, M., Lang, D., Hamze, F. and de Freitas, N. (2004) Fast probability propagation: Beyond belief(s). Technical report, CS Department, University of British Columbia.

[14] LeGland F. and Mevel L. (1997) Recursive identification in hidden Markov models, *Proc. 36th IEEE Conf. Decision and Control*, pp. 3468-3473.

[15] Liu J.S. and Chen R. (1998) Sequential Monte Carlo methods for dynamic systems, *J. Am. Statist. Ass.*, vol. 93, pp. 1032-1044.

[16] Liu J. and West M. (2001) Combined parameter and state estimation in simulation-based filtering, In *Sequential Monte Carlo Methods in Practice* (eds Doucet A., de Freitas J.F.G. and Gordon N.J. New York: Springer-Verlag,.

[17] Ljung L. and Söderström T. (1983), *Theory and Practice of Recursive Identification*, MIT Press, Cambridge.

[18] Pflug G.C. (1996), *Optimization of Stochastic Models.* Kluwer.

[19] Spall J. C. (2000), Adaptive stochastic approximation by the simultaneous perturbation method," *IEEE Trans. Autom. Contr.,* vol. 45, pp. 1839–1853.

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

[21] Tadić V.B. and Doucet A. (2005) Exponential forgetting and geometric ergodicity for optimal filtering in general state-space models. *Stochastic Processes and Their Applications*, vol. 115, pp. 1408-1436.