A Sequential Smoothing Algorithm with Linear Computational Cost

Paul Fearnhead David Wyncoll Jonathan Tawn

May 9, 2008

Abstract

In this paper we propose a new particle smoother that has a computational complexity of $O(N)$, where N is the number of particles. This compares favourably with the $O(N^2)$ computational cost of most smoothers and will result in faster rates of convergence for fixed computational cost. The new method also overcomes some of the degeneracy problems we identify in many existing algorithms.

Through simulation studies we show that substantial gains in efficiency are obtained for practical amounts of computational cost. It is shown both through these simulation studies, and on the analysis of an athletics data set, that our new method also substantially outperforms the simple Filter-Smoother (the only other smoother with computational cost that is linear in the number of particles).

1 Introduction

State space models provide a flexible framework to handle non-linear time series. These models assume a time-series with observations Y_t that are conditionally independent given a hidden Markov process X_t. Formally the model is given by a state equation and an observation equation, which can be represented in terms of conditional distributions

$$X_{t+1}|\{X_{1:t} = x_{1:t}, Y_{1:t} = y_{1:t}\} \sim f(\cdot|x_t),$$
$$Y_t|\{X_{1:t} = x_{1:t}, Y_{1:t-1} = y_{1:t-1}\} \sim g(\cdot|x_t),$$

where we use the notation that $x_{1:t} = (x_1, \dots, x_t)$, and similarly for $y_{1:t}$. The model is completed through specifying an initial distribution for X_0.

When the observations are arriving sequentially we are often interested in the current value of the state X_t given all the available data. For this filtering problem, interest lies in estimating the posterior distribution $p(x_t|y_{1:t})$. Sequential Monte Carlo algorithms, known generically as particle filters, have recently emerged as a solution to this problem. These filters approximate $p(x_t|y_{1:t})$ by a set of N weighted particles; and are based on steps for sequentially producing a set of weighted particles that approximate $p(x_t|y_{1:t})$ given a set that approximates $p(x_{t-1}|y_{1:t-1})$. In their simplest form these algorithms produce equally weighted particles, and the particles can be viewed as an approximate sample from $p(x_t|y_{1:t})$ (see Gordon et al. (1993) or Kitagawa (1996)). The computational complexity of particle filters is usually linear in the number of particles,

N, and as such large numbers of them can be chosen to best approximate the target posterior.

Our interest in this paper lies in the related smoothing problem whose aim is to obtain estimates of previous states given a block of observations $y_1, \ldots, y_T$. While this problem can be theoretically solved with a slight modification of the particle filter (see Kitagawa (1996)), it is easy to show that this produces a poor approximation of the smoothing density $p(x_t|y_{1:T})$ for $t \ll T$. With this in mind, alternative algorithms to sequentially approximate $p(x_t|y_{1:T})$ after a particle filter have been developed (see for example Kitagawa (1996), Hürzeler and Künsch (1998), Doucet et al. (2000), Godsill et al. (2004) and Briers et al. (2004)). All these methods involve a step to re-weight particles that approximate a filter distribution so that the re-weighted particles approximate $p(x_t|y_{1:T})$. While they perform comparably well for a fixed number of particles, N, these algorithms have a complexity of $O(N^2)$ and therefore their use is restricted to smaller N than is used for particle filtering. Also, for some state-space models, particularly those for which a component of X_t is uniquely determined by the previous state, x_{t-1}, these alternative smoothing algorithms can degenerate; either they become equivalent to the $O(N)$ smoother of Kitagawa (1996) or cannot be applied at all.

We present a new smoothing algorithm. The basic idea is to allow the smoother to simulate new particles which will be used to approximate $p(x_t|y_{1:T})$, rather than just re-weighting existing particles. This approach avoids some of the degeneracy issues of existing particle smoothers. However, the most important feature of our new smoothing algorithm is that there is a set of models for which the computational complexity is linear in the number of particles. This set includes all models with a linear-Gaussian state equation, and all models for which the likelihood, $g(y_t|x_t)$ is integrable in x_t. This covers a wide-variety of models such as the bearings-only tracking model (Gordon et al., 1993), factor stochastic volatility models (Liu and West, 2001), time varying auto-regressive coefficient models (Kitagawa and Gersch, 1985) and ARCH models (Fearnhead, 2005) amongst many others.

Our work itself was motivated by problems of non-stationarity when modelling the extremes of a time-series. For example, in Section 5, we consider data on the fastest times for the women's 3000m (see Figure 4). Whilst up to 1982 there is evidence of a year-on-year improvement in times for this event, since 1982 times have plateaued – if anything times worsened in the latter half of the 1980s, perhaps due to increased regulation and testing for the use of performance enhancing drugs. It is natural to model the data from each year using a distribution which is motivated by asymptotic extreme value theory; and we incorporate non-stationarity into this distribution through allowing its location parameter to vary in a non-parametric way. We can thus obtain a state-space model for the data, where the state is the location parameter. Standard non-parametric models, such as random walks and integrated random walks, can be formulated as linear-Gaussian models for the location parameter. See Smith and Miller (1986) for an example of state space models being used to analyse athletics records.

The article is organised as follows. We begin Section 2 by describing particle filtering and go on to review the current methods for sequential smoothing while demonstating their flaws. In Section 3 we derive our new algorithm which attempts to overcome these. Section 4 contains a simulation study with a multivariate Normal model, which shows the substantial improvements our new smoother gives for models with a linear-Gaussian state equation. Section 5 compares relative efficiencies of the methods at analysing the athletics dataset, and addresses the question as to how extreme the 1993 world record of Wang Junxia was. To analyse these data, we develop an efficient EM algorithm that utilises our new smoother, to estimate fixed parameters in the extreme value distribution. Our analysis suggests that a new world record as or more extreme as that of Wang Junxia's would happen less than once every 5000 years from the evidence about the population of the other top athletes in this event.

2 Current methods for particle smoothing

2.1 Particle filtering

The aim of Bayesian filtering is to calculate sequentially the filter distributions $p(x_t|y_{1:t})$ upon receipt of observations y_t. The analytical solution to this problem is given by

$$p(x_t|y_{1:t}) \propto g(y_t|x_t) \int f(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1})\, \mathrm{d}x_{t-1}, \tag{1}$$

which relates $p(x_t|y_{1:t})$ to $p(x_{t-1}|y_{1:t-1})$ and y_t. This recursion is intractable in general, although an important exception to this is when both the state f and the likelihood g are linear-Gaussian densities and the prior is also Gaussian. In this case the solution is given by the Kalman filter recursions of Kalman (1960).

Particle filters aim to overcome the intractability of (1) by using potential draws of the state to approximate the unknown filter distributions. In general, we approximate the distribution $p(x_t|y_{1:t})$ by a discrete distribution with support $\{x_t^{(i)}\}_{i=1}^N$ and probability masses $\{w_t^{(i)}\}_{i=1}^N$. Applying this to $p(x_{t-1}|y_{1:t-1})$ in (1) gives the approximation

$$p(x_t|y_{1:t}) \simeq c g(y_t|x_t) \sum_{i=1}^{N} f(x_t|x_{t-1}^{(i)}) w_{t-1}^{(i)}, \tag{2}$$

where c is a normalising constant. A particle filter algorithm gives steps for producing weighted particles to approximate this. For reviews of various filtering methods see for example Liu and Chen (1998), Doucet et al. (2000) and Fearnhead (2008).

We focus our attention on the auxiliary particle filter of Pitt and Shephard (1999). This is a general method through which many simpler particle filters can be defined as special cases. In this approach we aim to approximate

$$c g(y_t|x_t) f(x_t|x_{t-1}^{(i)}) w_{t-1}^{(i)} \tag{3}$$

by

$$q(x_t|x_{t-1}^{(i)}, y_t)\beta_t^{(i)},$$

where $q(\cdot|x_{t-1}^{(i)}, y_t)$ is a distribution we can sample from and $\{\beta_t^{(i)}\}_{i=1}^N$ are normalised weights which sum to 1. We then use a combination of re-sampling and importance sampling to generate a weighted sample approximating (2).

Algorithm 1 gives the general algorithm for sequentially sampling weighted particles $\{(x_t^{(i)}, w_t^{(i)})\}$ approximating $p(x_t|y_{1:t})$. While the simplest way to initialise the algorithm is to sample from the prior $p(x_0)$ and propagate from $t = 1$, it is usually possible and more accurate to sample from $p(x_1|y_1)$ directly using standard importance sampling.

Algorithm 1 Auxiliary particle filter of Pitt and Shephard (1999)

```
1. Initialisation: Sample $\{x_0^{(i)}\}$ from the prior $p(x_0)$ and set $w_0^{(i)} = 1/N$ for all $i$.
2. For $t = 1, 2, \ldots$
    (a) Re-sample: Use the $\{\beta_t^{(i)}\}$ as probabilities to sample $N$ indices $j_1, ..., j_N$ from $\{1, ..., N\}$.
    (b) Propagate: Sample the new particles $x_t^{(i)}$ independently from $q(\cdot|x_{t-1}^{(j_i)}, y_t)$.
    (c) Re-weight: Assign each particle $x_t^{(i)}$ the corresponding importance weight
        $w_t^{(i)} \propto \frac{g(y_t|x_t^{(i)}) f(x_t^{(i)}|x_{t-1}^{(j_i)}) w_{t-1}^{(j_i)}}{q(x_t^{(i)}|x_{t-1}^{(j_i)}, y_t) \beta_t^{(j_i)}}$
        and normalise them to sum to 1.
```

The efficiency of the particle filter rests primarily on the choice of proposal density q and re-sampling weights $\beta_t^{(i)}$. In the simplest case we could have $q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$ and $\beta_t^{(i)} = w_{t-1}^{(i)}$ which is essentially the algorithm of Gordon et al. (1993). Such a choice would place mass unevenly on the particles thus wasting those with small weights. To rectify this the auxiliary approach produces more even weights if q and $\beta_t^{(i)}$ are chosen so that (3) is well approximated. In particular if $q(x_t|x_{t-1}^{(i)}, y_t) = p(x_t|x_{t-1}^{(i)}, y_t)$ and $\beta_t^{(i)} \propto p(y_t|x_{t-1}^{(i)}) w_{t-1}^{(i)}$ then the final weights $w_t^{(i)}$ will all equal $1/N$ and we say the filter is *adapted*. In most cases this optimal choice of q and $\beta_t^{(i)}$ is not possible, but we can still obtain and use good approximations to them.

Many authors have suggested further enhancements to the standard particle filter. The re-sampling step is optional and can be omitted by setting $j_i = i$ thus propagating each particle $x_{t-1}^{(i)}$ once. This eliminates any extra noise from re-sampling but gives uneven weights. Liu and Chen (1995) propose a measure of the *effective sample size* and resample only when it falls below a fixed threshold. Carpenter et al. (1999) show that the re-sampling noise is minimised by producing a stratified sample of the indices j_i and give an $O(N)$ algorithm to achieve this.

A further enhancement which takes advantage of the linear-Gaussian state is Rao-Blackwellisation; see Casella and Robert (2001) for an introduction to the topic and Doucet et al. (2000) for an application to particle filtering. The idea is that for some models it is possible to integrate out part of the state analytically. This enables the integrable part of the state to be represented by a distribution rather than a specific value. The advantage of Rao-Blackwellisation is that less Monte Carlo error is accrued in each update and so the variance of estimates is reduced. An example of the application of Rao-Blackwellisation is given in Section 5.

2.2 Smoothing while filtering

In its simplest form, smoothing can be achieved from a simple extension to the particle filter as shown by Kitagawa (1996), and we call the resulting algorithm the *Filter-Smoother*. As with the filter distribution $p(x_t|y_{1:t})$ in (1), we have a recursive solution for the joint smoothing distribution:

$$p(x_{1:t}|y_{1:t}) \propto g(y_t|x_t)f(x_t|x_{t-1})p(x_{1:t-1}|y_{1:t-1}). \tag{4}$$

By comparing (1) and (4) it is easy to show that the particle filter steps can be used to update weighted paths $\{(x_{1:t}^{(i)}, w_t^{(i)})\}_{i=1}^N$ approximating $p(x_{1:t}|y_{1:t})$. Doing so simply requires keeping track of the inheritance of the newly sampled particle $x_t^{(i)}$ by setting $x_{1:t}^{(i)} = (x_{1:t-1}^{(j_i)}, x_t^{(i)})$. This means that any filtering algorithm can be used and the method inherits the $O(N)$ computational complexity of the filter making large numbers of particles feasible.

While this Filter-Smoother approach can produce an accurate approximation of the filtering distribution $p(x_t|y_{1:t})$ it gives a poor representation of previous states. To see this we note that whenever we resample the paths $\{x_{1:t-1}^{(i)}\}$ by sampling the auxiliary variables $\{j_i\}$ we end up with multiple copies of some paths but lose others altogether. Therefore the number of distinct particles at any given time decreases monotonically the more times we resample. Also, with multiple copies of some particles, their weights are effectively added together on a single point so that marginally the weights become more uneven as we look back in time.

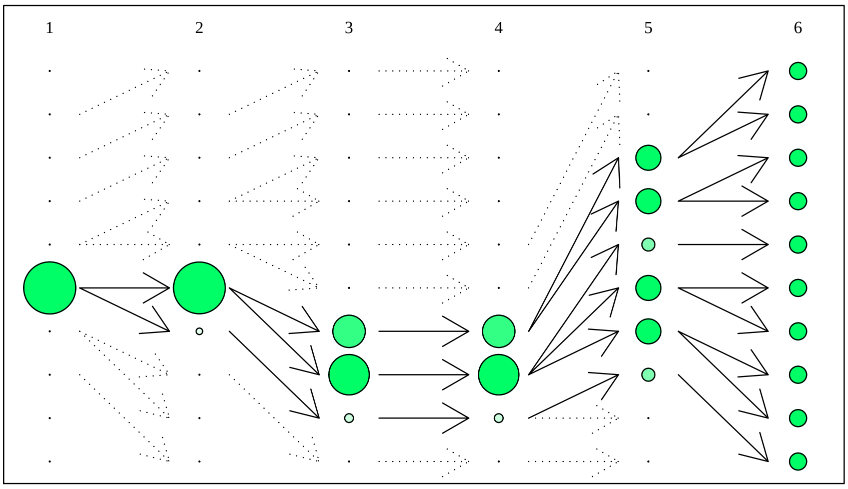


Figure 1: Plot showing how the simple smoother re-weights filter particles. The arrows represent the dependencies between the particles at time t and $t-1$ due to re-sampling. The size of the particle represents its total weight as a draw from the smoothed distribution.

This can be seen in Figure 1 which represents 10 smoothed paths $x_{1:6}^{(i)}$ showing how they re-weight filter particles. As you can see, particles which are lost due to re-sampling receive no weight and particles with many offspring have large weights. While the filter approximation at time 6 is good, the weights become more uneven as the number of weighted particles decreases going back in time. This is not surprising since the particles at times $t < 6$ are drawn to approximate $p(x_t|y_{1:t})$ so must be unevenly weighted if they are to represent a different distribution.

As a final point we note that re-sampling more infrequently can improve this method of smoothing although there is a limit to how much this can help. Even with no re-sampling, the approximation to $p(x_t|y_{1:T})$ will deteriorate as $T - t$ gets large: with the particle approximation tending to give non-negligible weight to all but a small subset of particles, and eventually only one particle having a non-negligible weight.

2.3 Other smoothing algorithms

Several algorithms have been proposed to improve on the simple Filter-Smoother. A common requirement is that a particle filter is run first to give weighted particles $\{(x_t^{(i)}, w_t^{(i)})\}_{i=1}^N$ approximating $p(x_t|y_{1:t})$ for $t = 1, \dots, T$.

2.3.1 Forward-Backward smoothing

The *Forward-Backward Smoother* of Doucet et al. (2000), as well as the related algorithms of Tanizaki and Mariano (1994) and Hürzeler and Künsch (1998), is based around the backwards recursion

$$p(x_t|y_{1:T}) = p(x_t|y_{1:t}) \int \frac{f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T})\, \mathrm{d}x_{t+1}, \quad \text{for } t < T.$$

The unknown densities can be approximated using filter particles from the current time t and smoother particles from $t + 1$ to obtain

$$p(x_t|y_{1:T}) \simeq \sum_{i=1}^{N} \delta(x_t - x_t^{(i)}) w_{t|T}^{(i)},$$

where

$$w_{t|T}^{(i)} := \sum_{j=1}^{N} \frac{f(x_{t+1|T}^{(j)}|x_t^{(i)}) w_t^{(i)}}{\sum_{k=1}^{N} f(x_{t+1|T}^{(j)}|x_t^{(k)}) w_t^{(k)}} w_{t+1|T}^{(j)} \tag{5}$$

and $\delta(\cdot)$ is the Dirac delta function. This approximation can be used to sequentially re-weight the filter particles backwards in time so that they represent the marginal smoothing densities.

2.3.2 Two-Filter smoothing

The *Two-Filter Smoother* of Briers et al. (2004) combines samples from a particle filter with those from a *backwards information filter* to produce estimates of $p(x_t|y_{1:T})$.

The backwards information filter produces sequential approximations of the likelihood $p(y_{t:T}|x_t)$ backwards through time and is based on the following recursion:

$$p(y_{t:T}|x_t) = g(y_t|x_t) \int f(x_{t+1}|x_t) p(y_{t+1:T}|x_{t+1}) \,\mathrm{d}x_{t+1}, \quad \text{for } t < T. \tag{6}$$

Since $p(y_{t:T}|x_t)$ is not a probability density function in x_t it may not have a finite integral over x_t in which case a particle representation will not work. The smoothing algorithm in Kitagawa (1996) assumes implicitly that this is not the case but Briers et al. (2004) propose the following construction which will always give a finite measure. They introduce an artificial prior distribution $\gamma_0(x_0)$ which, when substituted for $p(x_0)$, yields a backwards filter density

$$\tilde{p}(x_t|y_{t:T}) \propto \gamma_t(x_t) p(y_{t:T}|x_t), \tag{7}$$

where $\gamma_t(x_t) = \int f(x_t|x_{t-1}) \gamma_{t-1}(x_{t-1}) \,\mathrm{d}x_{t-1}$ is derived recursively from $\gamma_0(x_0)$.

An artificial prior is introduced so that $\gamma_t(x_t)$ is available in closed form which is only possible when the state is linear-Gaussian. If the prior $p(x_0)$ is also Gaussian then this can be used instead of $\gamma_0(x_0)$. If however the state is not linear-Gaussian but the likelihood $g(y_t|x_t)$ is integrable, we can instead propagate a particle representation of $p(y_{t:T}|x_t)$ by assuming $\gamma_t(x_t) \equiv 1$ throughout the following derivation.

Following on from (6) the backwards filter is derived from

$$\begin{aligned}\tilde{p}(x_t|y_{t:T}) &\propto \gamma_t(x_t) g(y_t|x_t) \int f(x_{t+1}|x_t) \frac{\tilde{p}(x_{t+1}|y_{t+1:T})}{\gamma_{t+1}(x_{t+1})} \,\mathrm{d}x_{t+1} \\ &\simeq \gamma_t(x_t) g(y_t|x_t) \sum_{k=1}^{N} \frac{f(\tilde{x}_{t+1}^{(k)}|x_t)}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})} \tilde{w}_{t+1}^{(k)},\end{aligned}$$

where the weighted particles $\{(\tilde{x}_{t+1}^{(k)}, \tilde{w}_{t+1}^{(k)})\}$ approximate $\tilde{p}(x_{t+1}|y_{t+1:T})$. This is very similar to the derivation of the forwards filter and as such many filtering algorithms and enhancements can be modified for this purpose.

For example, an auxiliary backwards filter in the style of Pitt and Shephard (1999) can be made by finding a distribution $\tilde{q}(\cdot|y_t, \tilde{x}_{t+1}^{(k)})$ we can sample from such that

$$\tilde{q}(x_t|y_t, \tilde{x}_{t+1}^{(k)}) \tilde{\beta}_t^{(k)} \simeq \gamma_t(x_t) g(y_t|x_t) f(\tilde{x}_{t+1}^{(k)}|x_t) \frac{\tilde{w}_{t+1}^{(k)}}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})}.$$

We then proceed analogously to Algorithm 1 for $t = T, ..., 1$ after initialising the algorithm with particles drawn from $\gamma_{T+1}(x_{T+1})$. An adapted backwards filter giving even weights $\tilde{w}_t^{(k)} = 1/N$ is achieved with $\tilde{q}(x_t|y_t, \tilde{x}_{t+1}^{(k)}) = p(x_t|y_t, \tilde{x}_{t+1}^{(k)})$ and $\tilde{\beta}_t^{(k)} \propto \tilde{p}(y_t|\tilde{x}_{t+1}^{(k)}) \tilde{w}_{t+1}^{(k)}$ where we again use $\tilde{p}$ to denote a distribution which uses $\gamma_0(x_0)$ as the prior instead of $p(x_0)$.

Having run a forwards particle filter and a backwards information filter, it is possible to combine the two to estimate $p(x_t|y_{1:T})$. The Two-Filter Smoother is based upon writing the target density as

$$\begin{aligned}p(x_t|y_{1:T}) &\propto p(x_t|y_{1:t-1}) \cdot p(y_{t:T}|x_t) \\ &\propto \int f(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1}) \,\mathrm{d}x_{t-1} \cdot \frac{\tilde{p}(x_t|y_{t:T})}{\gamma_t(x_t)}.\end{aligned}$$

Therefore filter particles $\{(x_{t-1}^{(j)}, w_{t-1}^{(j)})\}$ approximating $p(x_{t-1}|y_{1:t-1})$ and backwards filter particles $\{(\tilde{x}_t^{(k)}, \tilde{w}_t^{(k)})\}$ approximating $\tilde{p}(x_t|y_{t:T})$ are used to obtain

$$p(x_t|y_{1:T}) \simeq \sum_{k=1}^{N} \delta(x_t - \tilde{x}_t^{(k)})\tilde{w}_{t|T}^{(k)},$$

where

$$\tilde{w}_{t|T}^{(k)} :\propto \frac{\tilde{w}_t^{(k)}}{\gamma_t(\tilde{x}_t^{(k)})} \sum_{j=1}^{N} f(\tilde{x}_t^{(k)}|x_{t-1}^{(j)})w_{t-1}^{(j)}. \tag{8}$$

Thus particles from a forwards filter are used to re-weight those from a backwards filter so that they represent the target distribution.

2.4 Comparison of current particle smoothers

Both the Forward-Backward and Two-Filter smoothers aim to improve on the simple Filter-Smoother by removing its dependence on the inheritance paths of the particle filter. Forward-Backward smoothing does this by reweighting the filter particles while Two-Filter smoothing re-weights particles sampled from a backwards filter. However, both algorithms are $O(N^2)$ as the calculation of each particle's weight is an $O(N)$ operation[1]. Thus, while variants of these particle smoothers produce better estimates for a fixed particle number N, far fewer particles can be used for these algorithms than can for the Filter-Smoother in a fixed amount of time.

Another advantage of the Filter-Smoother is that it gives draws of the joint smoothing distribution $p(x_{1:T}|y_{1:T})$ rather than only the marginal distributions. It is possible to adapt the Forward-Backward Smoother to also draw samples from the joint smoothing distribution as shown in Hürzeler and Künsch (1998) and Godsill et al. (2004). Their derivation is similar to that of the Forward-Backward Smoother above and as such share its complexity and are determined by the re-sampling of the filter. They therefore achieve better samples of the joint distribution than the Filter-Smoother for a fixed N but give a slightly worse representation of the marginal distributions than the Forward-Backward Smoother.

Since the Forward-Backward Smoother and the Filter-Smoother rely on the support of filter particles we may expect them to approximate $p(x_t|y_{1:T})$ best for t close to T where the target is most similar to $p(x_t|y_{1:t})$. Likewise the Two-Filter Smoother may do best for small t when the backwards filter distribution $\tilde{p}(x_t|y_{t:T})$ is likely to be closest to our target. However when there is a large discrepancy between these distributions the particles will be weighted very unevenly as they will not be located in the right position to represent the smoothed distribution. Ideally we would like an algorithm which samples particles in the correct position for the smoothed distribution.

[1] The overall cost of calculating the weight (5) in the Forward-Backward Smoother is $O(N)$ as each of the terms in the denominator need to be calculated only once and can then be stored

2.5 Degeneracy of the Forward-Backward and Two-Filter smoothers

As a final point we note that the Forward-Backward and Two-Filter smoothers' reliance on the form of the state density causes degeneracy problems with certain models and filters. Specifically, this happens whenever $f(x_t|x_{t-1})$ is zero or approximately so for most combinations of possible x_t and x_{t-1}. As an example, consider the simple AR(2) process

$$z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + \epsilon_t$$

with $\epsilon_t \sim \mathcal{N}(0, \nu^2)$. The model can be written as a two-dimensional Markov process by defining the state as $x_t = (x_{t,1}, x_{t,2})'$ where $x_{t,1} = z_t$ and $x_{t,2} = z_{t-1}$.

This gives the state transition density

$$f(x_t|\tilde{x}_{t-1}) = \mathcal{N}(x_{t,1}|\phi_1\tilde{x}_{t-1,1} + \phi_2\tilde{x}_{t-1,2}, \nu^2)\,\delta(x_{t,2} - \tilde{x}_{t-1,1}),$$

where we write $\mathcal{N}(z|\mu, \nu^2)$ for the density of $\mathcal{N}(\mu, \nu^2)$ evaluated at z. This density is zero whenever the second component of x_t does not equal the first component of $\tilde{x}_{t-1}$. This means that for two sets of particles $\{\tilde{x}_{t-1}^{(j)}\}$ and $\{x_t^{(i)}\}$, $f(x_t^{(i)}|\tilde{x}_{t-1}^{(j)})$ is likely to be zero unless $x_t^{(i)}$ was generated from $\tilde{x}_{t-1}^{(j)}$.

Since the Forward-Backward Smoother relies on comparing particles sampled from the filter at time t with those at time $t+1$, it can be shown that the weight (5) reduces to the effective weight given to each particle by the Filter-Smoother. However the situation is worse for Two-Filter smoothing which fails completely as the forwards and backwards filter particles were sampled independently. With probability 1, no pairs of forwards and backwards filter particles match and so all the weights (8) will be zero.

3 New smoothing algorithm

We now describe our new smoothing algorithm which attempts to overcome the weaknesses of the current methods. Our primary aim is to draw new particles from the marginal smoothing densities directly rather than re-weight those drawn from another distribution. We describe the basic idea first, and then look at how the smoother can be implemented so that its computational cost is linear in the number of particles.

We start with a similar derivation to the Two-Filter Smoother which gives

$$\begin{aligned} p(x_t|y_{1:T}) &\propto p(x_t|y_{1:t-1}) \cdot g(y_t|x_t) \cdot p(y_{t+1:T}|x_t) \\ &\propto \int f(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1})\,\mathrm{d}x_{t-1} \cdot g(y_t|x_t) \cdot \\ &\qquad \int f(x_{t+1}|x_t) \frac{\tilde{p}(x_{t+1}|y_{t+1:T})}{\gamma_{t+1}(x_{t+1})}\,\mathrm{d}x_{t+1}, \end{aligned}$$

where we use the artificial prior and backwards filter in (7) above. These integrals can be approximated using weighted particles from a particle filter at time $t-1$ and from a backwards information filter at time $t+1$ to obtain

$$p(x_t|y_{1:T}) \simeq c \sum_{j=1}^{N} \sum_{k=1}^{N} f(x_t|x_{t-1}^{(j)}) w_{t-1}^{(j)} \cdot g(y_t|x_t) \cdot \frac{f(\tilde{x}_{t+1}^{(k)}|x_t)}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})} \tilde{w}_{t+1}^{(k)}, \qquad (9)$$

where c is a normalising constant. Though this formula can be written as the product of two sums, we write it as a double sum to emphasise that there are N^2 (j, k) pairs. We also note that any filtering algorithm can be used to generate $\{x_{t-1}^{(j)}\}$ and $\{\tilde{x}_{t+1}^{(k)}\}$ as long as the artificial prior $\gamma_{t+1}(x_{t+1})$ here is the same one used to sample $\{(\tilde{x}_{t+1}^{(k)}, \tilde{w}_{t+1}^{(k)})\}$ in the backwards information filter. As before we assume $\gamma_{t+1}(x_{t+1}) \equiv 1$ throughout if the backwards filter approximates $p(y_{t+1:T}|x_{t+1})$ instead of $\tilde{p}(x_{t+1}|y_{t+1:T})$.

To sample from this approximation we start by mirroring the auxiliary particle filter of Pitt and Shephard (1999) by finding a sampling distribution $\bar{q}$ and weights $\bar{\beta}_t^{(j,k)}$ such that

$$\bar{q}(x_t|x_{t-1}^{(j)}, y_t, \tilde{x}_{t+1}^{(k)})\bar{\beta}_t^{(j,k)} \simeq f(x_t|x_{t-1}^{(j)})g(y_t|x_t)f(\tilde{x}_{t+1}^{(k)}|x_t)\frac{w_{t-1}^{(j)}\tilde{w}_{t+1}^{(k)}}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})}.$$

Algorithm 2 gives the algorithm that results from using the $\bar{\beta}_t^{(j,k)}$s to sample (j, k) pairs before using $\bar{q}$ to sample new particles $\bar{x}_t^{(i)}$.

Algorithm 2 New $O(N^2)$ smoothing algorithm

1. **Filter forwards:** Run a particle filter to generate $\{(x_t^{(j)}, w_t^{(j)})\}$ approximating $p(x_t|y_{1:t})$ for $t = 0, \ldots, T$.

2. **Filter backwards:** Run a backwards information filter to generate $\{(\tilde{x}_t^{(k)}, \tilde{w}_t^{(k)})\}$ approximating $\tilde{p}(x_t|y_{t:T}) \propto \gamma_t(x_t)p(y_{t:T}|x_t)$ for $t = T, \ldots, 2$.

3. **Smooth:** For $t = 1, \ldots, T-1$

    (a) **Re-sample:** Calculate the $\bar{\beta}_t^{(j,k)}$s and use them as probabilities to sample N pairs $\{(j_i, k_i)\}_{i=1}^N$.

    (b) **Propagate:** Sample the new particles $\bar{x}_t^{(i)}$ independently from $\bar{q}(\cdot|x_{t-1}^{(j_i)}, y_t, \tilde{x}_{t+1}^{(k_i)})$.

    (c) **Re-weight:** Assign each particle $\bar{x}_t^{(i)}$ the weight

    $$\bar{w}_t^{(i)} \propto \frac{f(\bar{x}_t^{(i)}|x_{t-1}^{(j_i)})\, g(y_t|\bar{x}_t^{(i)})\, f(\tilde{x}_{t+1}^{(k_i)}|\bar{x}_t^{(i)})\, w_{t-1}^{(j_i)}\, \tilde{w}_{t+1}^{(k_i)}}{\bar{q}(\bar{x}_t^{(i)}|x_{t-1}^{(j_i)}, y_t, \tilde{x}_{t+1}^{(k_i)})\, \bar{\beta}_t^{(j_i,k_i)}\, \gamma_{t+1}(\tilde{x}_{t+1}^{(k_i)})}$$

    and normalise them to sum to 1.

Note that the output of Algorithm 2 is a set of triples, $(x_{t-1}^{(j_i)}, \bar{x}_t^{(i)}, \tilde{x}_{t+1}^{(k_i)})$, with associated weights, $\bar{w}_t^{(i)}$. These can be viewed as a particle approximation to $p(x_{t-1:t+1}|y_{1:T})$. If our interest solely lies in the marginal $p(x_t|y_{1:T})$ we just keep the particles, $\bar{x}_t^{(i)}$, and their associated weights, $\bar{w}_t^{(i)}$. We note further that the optimal choice of propagation density is $\bar{q}(x_t|x_{t-1}^{(j)}, y_t, \tilde{x}_{t+1}^{(k)}) = p(x_t|x_{t-1}^{(j)}, y_t, \tilde{x}_{t+1}^{(k)})$ while the optimal re-sampling probabilities are given by

$$\bar{\beta}_t^{(j,k)} \propto \int f(x_t|x_{t-1}^{(j)})g(y_t|x_t)f(\tilde{x}_{t+1}^{(k)}|x_t)\,\mathrm{d}x_t\,\frac{w_{t-1}^{(j)}\tilde{w}_{t+1}^{(k)}}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})}. \qquad (10)$$

We do not require our algorithm to generate samples for time T since these are available from the filter. Similarly, particles for time 1 are available from the backwards filter if we use $\gamma_0(x_0) = p(x_0)$ for the artificial prior.

Algorithm 2 overcomes the degeneracy problem of the Forward-Backward and Two-Filter smoothers when there is a deterministic relationship between the states at successive time-points, as demonstrated in Section 2.5 with the AR(2) model. Algorithm 2 will still have degeneracy problems where there is a deterministic relationship between components of states separated by two or more time-points. However it is simple, at least in theory, to extend our method so that we jointly sample a block $(x_t, \dots, x_{t+n})$ given filter particles $\{x_{t-1}^{(j)}\}$ and backwards filter particles $\{\tilde{x}_{t+n+1}^{(k)}\}$ (see Doucet et al. (2006) for an example of block sampling in particle filters). By choosing n sufficiently large such that there is not deterministic relationship between components of x_t and x_{t+n}, our approach to smoothing can then be applied in these cases.

Like the Two-Filter Smoother in Section 2, our smoothing step is not sequential and can be performed independently for each time t. Also, the computational complexity of each step is $O(N^2)$ which is comparable with all but the simplest Filter-Smoother. However, as it stands we have N^2 $\bar{\beta}_t^{(j,k)}$s to calculate making it $O(N^2)$ in memory also which could mean that it is impractical for even modest sample sizes.

3.1 Making Algorithm 2 $O(N)$

The above smoothing algorithm has a computational cost that is $O(N^2)$, that is quadratic in the number of particles, due to the need to calculate N^2 probabilities, $\bar{\beta}_t^{(j,k)}$. A simple approach to reduce the computational cost of the smoothing algorithm is to choose these probabilities so that they correspond to choosing particles at time $t-1$ and backward-filter particles at time $t+1$ independently of each other. Our algorithm will then be $O(N)$ in computational complexity as well as memory and as such will be much faster for large N.

Now the optimal distribution from which to choose the particles at time $t-1$ will be the corresponding marginal distribution of the optimal probabilities for $\bar{\beta}_t^{(j,k)}$, given in (10). Marginalising we get:

$$\begin{aligned}
\sum_{k=1}^{N} \bar{\beta}_t^{(j,k)} &\propto \sum_{k=1}^{N} \int f(x_t|x_{t-1}^{(j)}) g(y_t|x_t) f(\tilde{x}_{t+1}^{(k)}|x_t)\,\mathrm{d}x_t \frac{w_{t-1}^{(j)} \tilde{w}_{t+1}^{(k)}}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})} \\
&\xrightarrow{N\to\infty} \iint f(x_t|x_{t-1}^{(j)}) g(y_t|x_t) f(x_{t+1}|x_t)\,\mathrm{d}x_t \frac{w_{t-1}^{(j)} \tilde{p}(x_{t+1}|y_{t+1:T})}{\gamma_{t+1}(x_{t+1})}\,\mathrm{d}x_{t+1} \\
&\propto p(y_{t:T}|x_{t-1}^{(j)}) w_{t-1}^{(j)}.
\end{aligned}$$

Calculating this analytically will be impossible, but it suggests two simple approximations. The first is to sample particles at time $t-1$ according to their filtering weights $w_{t-1}^{(j)}$. However a better approach will be to sample according to an approximation of $p(y_t|x_{t-1}^{(j)}) w_{t-1}^{(j)}$, as it includes the information in the observation at time t. Now, in performing the particle filter we used the auxiliary filter which sampled particle $x_{t-1}^{(j)}$ with a probability $\beta_t^{(j)}$ which is chosen to be an approximation to $p(y_t|x_{t-1}^{(j)}) w_{t-1}^{(j)}$. Thus we suggest using exactly the same probabilities to sample the particles within one iteration of our sampling algorithm.

By similar calculations, it can be shown that we should optimally choose the backward-filter particles at time $t+1$ with probability proportional to $\tilde{p}(y_{1:t}|\tilde{x}_{t+1}^{(k)})\tilde{w}_{t+1}^{(k)}$. Again, we cannot calculate these exactly, but a simple idea is to use probabilities that approximate $\tilde{p}(y_t|\tilde{x}_{t+1}^{(k)})\tilde{w}_{t+1}^{(k)}$. Thus we can simply use the probabilities $\tilde{\beta}_t^{(k)}$ that were used in the backward filter, as these were chosen as to be an approximation to $\tilde{p}(y_t|\tilde{x}_{t+1}^{(k)})\tilde{w}_{t+1}^{(k)}$.

We thus obtain a similar algorithm to before, but with particles at time $t-1$ and $t+1$ sampled independently, and with $\bar{\beta}_t^{(j,k)}$ replaced by $\beta_t^{(j)}\tilde{\beta}_t^{(k)}$ in the calculation of the weight. Thus we have an $O(N)$ version of our smoothing algorithm shown in Algorithm 3. We note that we can speed up the algorithm further as the probabilities $\beta_t^{(j)}$ and $\tilde{\beta}_t^{(k)}$ (or even the auxiliary variables $\{j_i\}$ and $\{k_i\}$) can be saved from the filters to reduce the number of calculations in the smoothing step.

Algorithm 3 New $O(N)$ smoothing algorithm

Proceed as Algorithm 2 but substitute steps 3(a) and 3(c) for

3. (a) **Re-sample:** Use $\{\beta_t^{(j)}\}$ from the filter to sample $j_1, ..., j_N$ and $\{\tilde{\beta}_t^{(k)}\}$ from the backwards filter to sample $k_1, ..., k_N$ from $\{1, ..., N\}$

   (c) **Re-weight:** Assign each particle $\bar{x}_t^{(i)}$ the weight

$$\bar{w}_t^{(i)} \propto \frac{f(\bar{x}_t^{(i)}|x_{t-1}^{(j_i)})\, g(y_t|\bar{x}_t^{(i)})\, f(\tilde{x}_{t+1}^{(k_i)}|\bar{x}_t^{(i)})\, w_{t-1}^{(j_i)}\, \tilde{w}_{t+1}^{(k_i)}}{\bar{q}(\bar{x}_t^{(i)}|x_{t-1}^{(j_i)}, y_t, \tilde{x}_{t+1}^{(k_i)})\, \beta_t^{(j_i)}\, \tilde{\beta}_t^{(k_i)}\, \gamma_{t+1}(\tilde{x}_{t+1}^{(k_i)})}$$

   and normalise them to sum to 1.

4 Simulation study

We now compare the efficiency of our new algorithm against the currently available methods. Our simulations are based on a model with linear-Gaussian state and observation models. The specific state model we used is chosen to be the same as for our athletics application in Section 5. We have a chosen a linear-Gaussian observation model so that we can compare results of different particle smoothers with the true smoothing distributions obtained from the Kalman filter and smoother (see Kalman (1960) and Anderson and Moore (1979)).

Formally, we consider the model:

$$\begin{aligned} X_{t+1}|\{X_{1:t} = x_{1:t}, Y_{1:t} = y_{1:t}\} &\sim \mathcal{N}(Fx_t, Q), \\ Y_t|\{X_{1:t} = x_{1:t}, Y_{1:t-1} = y_{1:t-1}\} &\sim \mathcal{N}(Gx_t, R), \\ X_0 &\sim \mathcal{N}(\mu_0, \Sigma_0), \end{aligned}$$

where

$$F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad Q = \nu^2 \begin{pmatrix} \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix},$$
$$G = (1, 0), \qquad R = \tau^2.$$

The state is derived from the pair of stochastic differential equations (SDEs) $\mathrm{d}X_{t,1} = X_{t,2}\mathrm{d}t$ and $\mathrm{d}X_{t,2} = \nu \mathrm{d}B_t$ and so the first component $X_{t,1}$ is the integrated path of the random walk $X_{t,2}$. A noisy observation of the first component is made at each time step. The parameter ν^2 determines the smoothness of the state over time. With a large value of ν^2 the state can move freely and thus follows the observations. When ν^2 is small however the model makes a linear fit to the observations.

We compare the two versions of our new algorithm with the simple Filter-Smoother of Section 2.2, the Forward-Backward Smoother of Section 2.3.1 and the Two-Filter Smoother of Section 2.3.2. We also look at how the relative performance of the algorithms is affected by the ratio of the state noise ν^2 to observation noise τ^2. The details of our particle filter, backwards filter and smoothing algorithms for this model are given in Appendix A.1.

To compare the accuracy of our smoothing algorithms' estimates of $X_{t,d}$ we estimate the *effective sample size* $N_{\text{eff}}(X_{t,d})$. Motivated by the fact that

$$\mathrm{E}\left(\frac{(\bar{X} - \mu)^2}{\sigma^2}\right) = \frac{1}{N},$$

when $X^{(1)}, \ldots, X^{(N)}$ iid $\mathcal{N}(\mu, \sigma^2)$ and $\bar{X}$ is their sample mean, we take

$$N_{\text{eff}}(X_{t,d}) = \mathrm{E}\left(\frac{(\hat{x}_{t,d} - \mu_{t,d})^2}{\sigma^2_{t,d}}\right)^{-1}, \tag{11}$$

where $\mu_{t,d}$ and $\sigma^2_{t,d}$ are the true mean and variance of $X_{t,d}|y_{1:T}$ obtained from the Kalman smoother and $\hat{x}_{t,d}$ is the random estimate from a particle smoother. We can therefore crudely say that the weighted sample produced by our smoother is as accurate at estimating $X_{t,d}$ as an independent sample of size $N_{\text{eff}}(X_{t,d})$. To estimate the expectation in (11) we use the mean value from 100 repetitions of each algorithm.

We first compare the smoothing algorithms using model parameters of $\nu^2 = \tau^2 = 1$ with $\mu_0 = (0, 0)'$ and $\Sigma_0 = I_2$ for the prior. We generated 20 datasets, each of length 200, and averaged the effective sample sizes to remove effects caused by a single dataset.

We chose different numbers of particles for each algorithm to try to reflect the varying complexities of each method. We started by choosing $10,000$ particles for the Filter-Smoother and $3,000$ for the $O(N)$ version of our new algorithm since they then took approximately the same amount of time to run. We would have liked to scale the $O(N^2)$ algorithms to take the same time to run but their speeds varied greatly. Part of this may be due to how the algorithms are implemented in R. We therefore fixed the number of particles for these three algorithms at 300. This made the $O(N^2)$ version of our new algorithm faster but the other two methods slower than the Filter-Smoother. The average time taken by each algorithm per run is shown in Table 1.

Algorithm	Filter	Forward- Backward	Two-Filter	New $O(N^2)$	New $O(N)$
N	10,000	300	300	300	3,000
Run time (s)	224	688	358	40	255

Table 1: Number of particles used and average run time of each algorithm.

Figure 2 shows how the average effective number of particles for estimating $X_{t,1}$ varies through time for the five algorithms considered. The results for $X_{t,2}$ (not shown) are very similar.

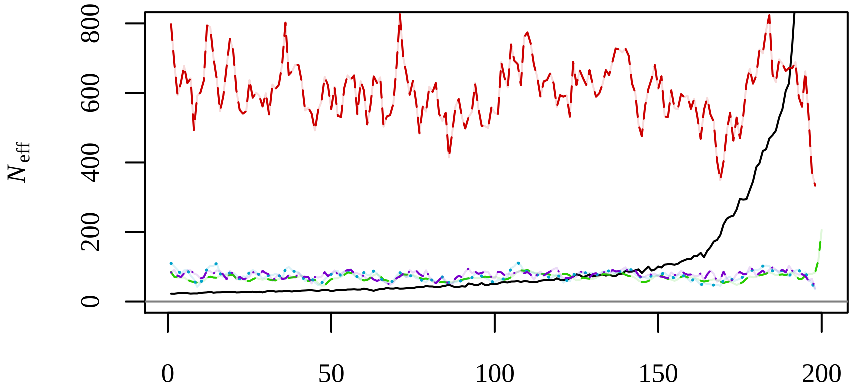


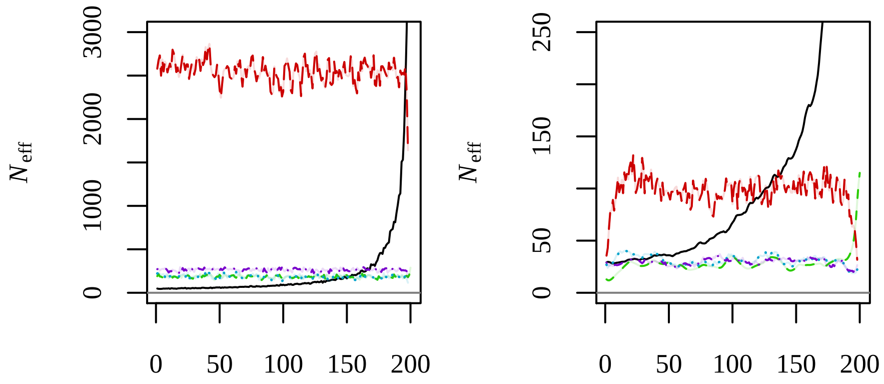
Figure 2: Average effective sample size for each of the 200 time steps using the filter (—), Forward-Backward (---) and Two-Filter smoothers (. . .) as well as the $O(N^2)$ (.–) and $O(N)$ version (— —) of our new algorithm.

We can see that the Filter-Smoother does very well for times close to $T = 200$ as this filter has by far the most particles and the filter and smoothing distributions are similar at this stage of the process. As predicted however this algorithm gets progressively worse as it goes backwards through time. This is not necessarily the case with the other algorithms whose efficiencies remain roughly constant over time when averaged over the 20 datasets. Of the two $O(N)$ algorithms we see that our new method vastly outperforms the Filter-Smoother for all but the final few time steps, despite taking a similar amount of time to run.

From Figure 2 we can also see that the three $O(N^2)$ algorithms have near identical efficiencies for this particular model. This may be because they are all derived in some way from the same formula, $p(x_t|y_{1:T}) \propto p(x_t|y_{1:t-1})p(y_{t:T}|x_t)$, and all combine filter particles with an $O(N^2)$ approximation of $p(y_{t:T}|x_t)$. We recall that these were run with the same number of particles N though in our implementation our new algorithm was faster than the other two here. However, even with this taken into account, the $O(N)$ version is many times more efficient for even these modest sample sizes.

To see how these results are affected by the ratio of the state noise ν^2 to the observation noise τ^2, we repeat the experiment first with $\nu^2 = 100$ while keeping $\tau^2 = 1$. This gives the state freedom to follow the observations which helps the algorithms to perform well. The results are shown in Figure 3a below.

Those for $\nu^2 = 1$ and $\tau^2 = 1/100$ gave very similar results.



(a) $\nu^2/\tau^2 = 100$

(b) $\nu^2/\tau^2 = 1/100$

Figure 3: Average effective sample sizes as in Figure 2 with different ratios of the state noise ν^2 to the observation noise τ^2.

We see that the accuracy of the Filter-Smoother still diminishes as it progresses backwards through time but all the other methods are close to their optimal efficiency of an effective sample size equal to N. This is particularly the case with our new $O(N^2)$ algorithm which outperforms the other $O(N^2)$ methods at every time step. Our new $O(N)$ algorithm however is by far the fastest allowing it to have 10 times as many particles as the slower methods. Its efficiency also suggests that our choice of re-sampling weights is reasonable.

We finally repeat the experiment with $\nu^2/\tau^2 = 1/100$ which makes the state highly dependent through time and causes all the particle methods to struggle. This can be seen from the low effective sample sizes in Figure 3b. Even though the Filter-Smoother diminishes at a faster rate than before it does better than the other algorithms for a large number of time steps. This is possibly due to the total accumulation of error in the filter, backwards filter and smoother, each of which performs badly in this case, which hinder the other methods. The Filter-Smoother eventually drops below the accuracy of our $O(N)$ method showing that our $O(N)$ algorithm can give stronger estimates of the earliest smoothing densities in even the toughest situations.

5 Athletic records

We use our smoothing algorithm to analyse data from the women's 3000m running event. Robinson and Tawn (1995) first studied the fastest times from 1972 to 1992 to assess whether Wang Junxia's record in 1993 was consistent with the previous data. They used an extreme value likelihood with a parametric trend to conclude that cutting 16.51s off the record, though unusual, was not exceptionally so. Smith (1997) outlined the benefits of a Bayesian analysis for calculating the probability of beating Wang Junxia's record given that a new record is set and Gaetan and Grigoletto (2004) extended this by using particle methods to model a dynamic trend.

While Gaetan and Grigoletto (2004) presented an attractive model for the data, it is our belief that the particle methods they used for their inference are highly inefficient. They used the smoothing algorithm of Tanizaki and Mariano (1994) with an AR(2) state which causes the smoother to degrade to the Filter-Smoother as shown in Section 2.5. They also introduced two random walks with extremely small variances to the state which causes all particle methods to struggle as was shown by the small effective sample sizes in Figure 3b in Section 4. We therefore believe that the conclusions they drew from using only $N = 1000$ particles are unreliable and we aim to produce a more robust analysis.

Whereas Gaetan and Grigoletto (2004) used the annual minimum running times, we use the r fastest annual times following the initial analysis of Robinson and Tawn (1995). Large amounts of data are now available on-line (for example from Larsson (2008)) from which the five fastest times of different athletes per year is shown in Figure 4. The natural choice for modelling the fastest annual running times is the generalised extreme value distribution for minima (GEVm) as motivated by asymptotic extreme value theory (see Coles (2001) for details as well as an introduction to extreme value techniques). This distribution has location and scale parameters $\mu \in \Re$ and $\sigma > 0$ as well as a shape parameter $\xi \in \Re$ which allows for many types of tail behaviour. Its cdf is given by

$$G(y|\mu, \sigma, \xi) = 1 - \exp\left\{ - \left[1 - \xi \left(\frac{y - \mu}{\sigma} \right) \right]_{+}^{-\frac{1}{\xi}} \right\},$$

where we define $[y]_+ := \max(y, 0)$. This distribution naturally extends to the r-smallest order statistic model which we use for the ith fastest records y_i per year. This model has a likelihood given by

$$g(y_{1:r}|\mu, \sigma, \xi) \propto \bar{G}(y_r|\mu, \sigma, \xi) \prod_{i=1}^{r} \frac{g(y_i|\mu, \sigma, \xi)}{\bar{G}(y_i|\mu, \sigma, \xi)},$$

where we write $\bar{G}(y|\mu, \sigma, \xi) := 1 - G(y|\mu, \sigma, \xi)$ for the GEVm survivor function and $g(y|\mu, \sigma, \xi)$ for the derivative of $G(y|\mu, \sigma, \xi)$ with respect to y.

For the state, Gaetan and Grigoletto (2004) used independent random walks for each of the three likelihood parameters. They used a second order random walk for μ to model the clear trend seen in the location of the data and for σ and ξ they used first order random walks with extremely small state variances to make them roughly constant in time. Since such state variances cause poor performance with the filters and smoothers, we assume for simplicity these parameters are fixed and known. We later estimate them using an EM algorithm based on our smoothing algorithm although we would have preferred to add them to the state to fully account for their uncertainty.

Since a second order AR(2) model for μ causes degeneracy problems with some smoothers, we instead adopt the smooth second order random walk given in the earlier simulation study of Section 4. We therefore augment the state with $\dot{\mu}$, the velocity of μ, giving us the two-dimensional state $x_t = (\mu_t, \dot{\mu}_t)^{'}$. Finally, for the prior we follow Gaetan and Grigoletto (2004) and use an uninformative normal distribution. Since the likelihood only depends on μ_t and the prior is Gaussian we used Rao-Blackwellisation to marginalise $\dot{\mu}_t$ thus improving the accuracy of the methods. Details of this step and the particle algorithms we used to achieve this are given in Appendix A.2.

For a fixed value of r and ν^2 we can estimate the likelihood parameters σ and ξ using an EM algorithm constructed using our new smoother (see Appendix B

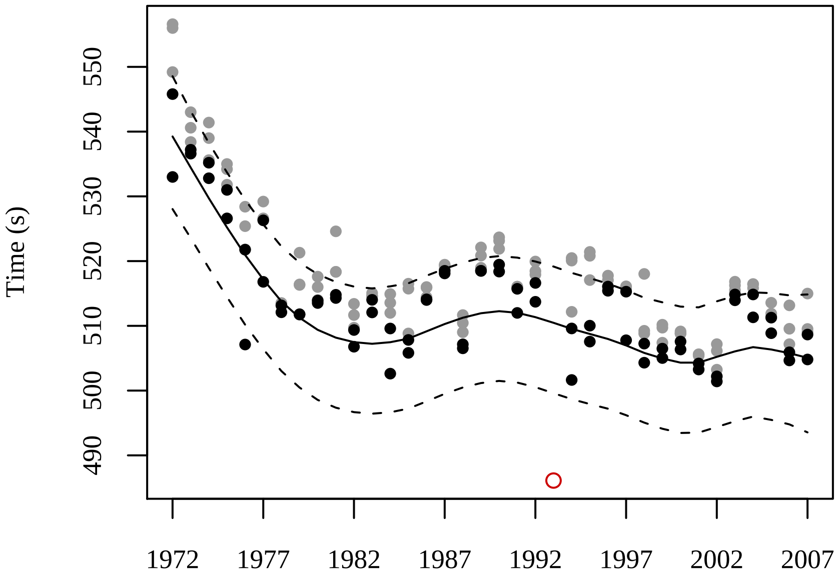


Figure 4: Five fastest times for the women's 3000m race between 1972 and 2007 with Wang Junxia's time in 1993. The two fastest annual times used for our fit are coloured black. Also shown is the mean and central 95% probability interval of the fitted predictive distribution for the fastest time per year.

for details). Simultaneously estimating ν^2 requires particles approximating the joint distribution $p(x_{t-1}, x_t|y_{1:T})$ which is possible using our approach (as our algorithm gives approximations to $p(x_{t-1:t+1}|y_{1:T})$, see Section 3). It is simpler however to select among a few possible ν^2 by maximising the model likelihood $p(y_{1972:2007}|\nu^2)$, which we estimate using the following formula of Kitagawa (1996):

$$p(y_{1972:2007}|\nu^2) \simeq \prod_{t=1972}^{2007} \sum_{i=1}^{N} g(y_{t,1:r}|\mu_{t|t-1}^{(i)}, \sigma, \xi) w_{t-1}^{(i)},$$

where $\mu_{t|t-1}^{(i)}$ is the first component of a predictive particle sampled from the state $f(\cdot|x_{t-1}^{(i)}, \nu^2)$ and $\{(x_t^{(i)}, w_t^{(i)})\}$ are sampled from a particle filter given ν^2. Table 2 shows a selection of ν^2 values with the corresponding EM estimates of σ and ξ and the model likelihood when we take $r = 2$.

ν^2	0.5	0.75	1	1.25	1.5	1.75	2
σ	4.36	4.25	4.22	4.15	4.12	4.04	4.01
ξ	-0.15	-0.13	-0.13	-0.11	-0.11	-0.09	-0.09
Likelihood ($\times 10^{-83}$)	0.41	2.72	3.68	3.50	2.41	1.52	1.02

Table 2: Model likelihood with σ and ξ estimates for different values of the smoothing parameter ν^2 and $r = 2$.

To select the number of observations r to include per year we constructed probability-probability and quantile-quantile plots to assess the model fit. Looking at $r = 1, \ldots, 5$ we concluded that the best fit was obtained from only two observations per year. As we see from Table 2, this leads us to select $\nu^2 = 1$ and estimate σ and ξ to be 4.22 and -0.13 respectively.

To estimate the probability of a new record in 1993 beating Wang Junxia's we use the $r = 2$ fastest times from 1972 to 2007 excluding 1993, denoted $y_{1972:2007}$, to estimate the predictive distribution of the fastest time in 1993. Given the parameters μ_{1993}, σ and ξ, the probability of Y_{1993}, a new record in 1993, beating Wang Junxia's time of 486.11s is given by

$$\mathrm{P}\{Y_{1993} \leq 486.11 | Y_{1993} \leq 502.62, \mu_{1993}, \sigma, \xi\} = \frac{G(486.11|\mu_{1993}, \sigma, \xi)}{G(502.62|\mu_{1993}, \sigma, \xi)},$$

where 502.62s was the world record prior to 1993. Unconditioning on μ_{1993}, we estimate the overall probability p_{rec} with

$$\begin{aligned} p_{\mathrm{rec}} &= \int \frac{G(486.11|\mu_{1993}, \sigma, \xi)}{G(502.62|\mu_{1993}, \sigma, \xi)} p(\mu_{1993}|y_{1972:2007})\, \mathrm{d}\mu_{1993} \\ &\simeq \sum_{i=1}^{N} \frac{G(486.11|\mu_{1993}^{(i)}, \sigma, \xi)}{G(502.62|\mu_{1993}^{(i)}, \sigma, \xi)} w_{1993}^{(i)}, \end{aligned}$$

where we use weighted particles to approximate $p(\mu_{1993}|y_{1972:2007})$.

To compare algorithms' efficiencies at approximating $p(\mu_{1993}|y_{1972:2007})$ we run a simulation to estimate the effective sample size $N_{\mathrm{eff}}(X_{t,1})$ as in Section 4 using 300 repetitions of each algorithm. However, since the true mean and variance of the target density are now unknown, we first estimate the true distribution using the Filter-Smoother with $750,000$ particles. Since our primary interest is to estimate the probability of a new record beating Wang Junxia's, we also calculate the sample variance of our estimate of this over the 300 repetitions used to estimate N_{eff}.

For this simulation we chose to compare only the $O(N)$ algorithms as both the Forward-Backward and the Two-Filter smoothers suffer problems of degeneracy when applied to the Rao-Blackwellised filter described in Appendix A.2. These smoothers could be applied to a non-Rao-Blackwellised filter, but given the simulation results in Section 4, it appears that these smoothers would not be competitive with our new smoother. Since we only require the marginal smoothing distribution for 1993, our new algorithm only requires the particle filter up to 1992 and the backwards filter back to 1994. We therefore chose the same number of particles, $10,000$, for both our algorithm and the Filter-Smoother and observed that they took roughly the same amount of time to run.

The results of the simulation are shown in Table 3. We can see that our new algorithm has an effective sample size over 8 times as large as that of the Filter-Smoother giving similarly less variable estimates. Of course, to calculate the marginals for every time step within the same amount of time our method could only use a third of the particles, but it would still outperform the Filter-Smoother for the majority of estimates.

Algorithm	N_{eff}	$\text{Var}(\hat{p}_{\text{rec}})$ $(\times 10^{-10})$
Filter-Smoother	130	3.616
New $O(N)$	1049	0.401

Table 3: Comparison of the efficiencies of the Filter-Smoother and our new algorithm for approximating $p(\mu_{1993}|y_{1972:2007})$ and the probability of a new record beating Wang Junxia's time. In both cases the average probability estimate was 1.92×10^{-4}.

Our analysis estimates the probability of a new record in 1993 beating Wang's to be 1.92×10^{-4}. This conflicts with the analysis of Gaetan and Grigoletto (2004) who showed Wang's record well within the reach of their boxplots of the conditional distribution. Apart from our doubts in the accuracy of their results, the main difference in the two analyses is that Gaetan and Grigoletto (2004) only used data on the fastest race for years up to 1993. Thus it may be the information in the extra data we use that leads us to a different conclusion about how extreme the world record of Wang is. We also admit that our analysis fails to account for the uncertainty in σ and ξ which could cause our estimate to be under-estimated. However, while Gaetan and Grigoletto (2004) attempted to account for this by augmenting the state with σ and ξ, this leads to poor performance of the particle methods so a new approach is required.

Appendix A Implementation of particle filters and smoothers

A.1 Multivariate Normal model

To implement the various smoothing algorithms we need to choose propagation densities for a particle filter, backwards information filter and the smoother itself. Using auxiliary algorithms throughout, the linear-Gaussian model assumption allows us to calculate the optimal densities and re-sampling probabilities. Using these we have adapted algorithms giving even weights of $1/N$ whenever we resample.

Writing $\mathcal{N}(x|\mu, \sigma^2)$ for the density of $\mathcal{N}(\mu, \sigma^2)$ evaluated at x, it is easy to show that the optimal filter is given by

$$\begin{aligned} q(x_t|x_{t-1}^{(j)}, y_t)\beta_t^{(j)} &= f(x_t|x_{t-1}^{(j)})g(y_t|x_t)w_{t-1}^{(j)} \\ &= \mathcal{N}(x_t|\mu_{t|t-1}^{(j)}, \Sigma_{t|t-1})\,\mathcal{N}(y_t|GFx_{t-1}^{(j)}, R + GQG^{'})w_{t-1}^{(j)}, \end{aligned}$$

where $\Sigma_{t|t-1} = (Q^{-1} + G^{'}R^{-1}G)^{-1}$ and $\mu_{t|t-1}^{(j)} = \Sigma_{t|t-1}(Q^{-1}Fx_{t-1}^{(j)} + G^{'}R^{-1}y_t)$. This is used for each algorithm but we only need to keep track of our trajectories for the simple Filter-Smoother.

For the backwards information filter we can use the actual prior $\gamma_t(x_t) = p(x_t) = \mathcal{N}(x_t|\mu_t, \Sigma_t)$, whose mean and covariance can be calculated sequentially using the prediction step of the Kalman filter. This gives

$$p(x_t|x_{t+1}) = \mathcal{N}(x_t|\tilde{F}x_{t+1} + \tilde{Q}\Sigma_t^{-1}\mu_t, \tilde{Q}),$$

where we define $\tilde{F} := \Sigma_t F^{'} \Sigma_{t+1}^{-1}$ and $\tilde{Q} := \Sigma_t F^{'} \Sigma_{t+1}^{-1} Q F^{'-1}$. We then obtain

$$\begin{aligned}\tilde{q}(x_t|y_t, \tilde{x}_{t+1}^{(k)})\tilde{\beta}_t^{(k)} &= p(x_t) g(y_t|x_t) f(\tilde{x}_{t+1}^{(k)}|x_t) \frac{\tilde{w}_{t+1}^{(k)}}{p(\tilde{x}_{t+1}^{(k)})} \\ &\propto \mathcal{N}(x_t|\mu_{t|t+1}^{(k)}, \Sigma_{t|t+1}) \times \\ &\qquad \mathcal{N}(y_t|G(\tilde{F}x_{t+1}^{(k)} + \tilde{Q}\Sigma_t^{-1}\mu_t), R + G\tilde{Q}G^{'})\tilde{w}_{t+1}^{(k)},\end{aligned}$$

where $\Sigma_{t|t+1} = (\Sigma_t^{-1} + G^{'}R^{-1}G + F^{'}Q^{-1}F)^{-1}$ and $\mu_{t|t+1}^{(k)} = \Sigma_{t|t+1}(\Sigma_t^{-1}\mu_t + G^{'}R^{-1}y_t + F^{'}Q^{-1}\tilde{x}_{t+1}^{(k)})$.

Finally, for our new smoothing algorithm we have

$$\begin{aligned}\bar{q}(x_t|x_{t-1}^{(j)}, y_t, \tilde{x}_{t+1}^{(k)}) &\propto f(x_t|x_{t-1}^{(j)}) g(y_t|x_t) f(\tilde{x}_{t+1}^{(k)}, x_t) \\ &\propto \mathcal{N}(x_t|\mu_{t|T}^{(j,k)}, \Sigma_{t|T}),\end{aligned}$$

where $\Sigma_{t|T} = (Q^{-1} + G^{'}R^{-1}G + F^{'}Q^{-1}F)^{-1}$ and $\mu_{t|T}^{(j,k)} = \Sigma_{t|T}(Q^{-1}Fx_{t-1}^{(j)} + G^{'}R^{-1}y_t + F^{'}Q^{-1}\tilde{x}_{t+1}^{(k)})$. The optimal re-sampling weights can be shown to be

$$\begin{aligned}\bar{\beta}_t^{(j,k)} &\propto p(\tilde{x}_{t+1}^{(k)}, y_t|x_{t-1}^{(j)}) \frac{w_{t-1}^{(j)}\tilde{w}_{t+1}^{(k)}}{p(\tilde{x}_{t+1}^{(k)})} \\ &= \mathcal{N}\left(\begin{pmatrix}\tilde{x}_{t+1}^{(k)} \\ y_t\end{pmatrix}\middle| \begin{pmatrix}F^2 \\ GF\end{pmatrix} x_{t-1}^{(j)}, \begin{pmatrix}Q + FQF^{'} & FQG^{'} \\ GQF^{'} & R + GQG^{'}\end{pmatrix}\right) \frac{w_{t-1}^{(j)}\tilde{w}_{t+1}^{(k)}}{p(\tilde{x}_{t+1}^{(k)})},\end{aligned}$$

which we can see does not factorise. Therefore, for the $O(N)$ version of our algorithm we use $\beta_t^{(j)}$ and $\tilde{\beta}_t^{(k)}$ from the filters as suggested in Section 3.1 as this should be a good approximation of the optimal weights.

A.2 Athletics records

Adapted auxiliary algorithms for this model will not be possible as the likelihood in μ is very complex. We therefore approximate the log likelihood $l(\mu_t)$ by a second-order Taylor approximation about an estimated mode $\hat{\mu}_t$ which leads to a normal approximation of the form

$$g(y_{t,1:r}|\mu_t) \simeq \mathcal{N}\left(\mu_t \,\middle|\, \hat{\mu}_t - \frac{l^{'}(\hat{\mu}_t)}{l^{''}(\hat{\mu}_t)}, -\frac{1}{l^{''}(\hat{\mu}_t)}\right)\Bigg|_{A_t}, \qquad (12)$$

where the distribution is restricted to the likelihood's support of

$$A_t := \{\mu_t|\sigma - \xi(y_{t,i} - \mu_t) > 0, \forall i\}.$$

In practise, we used the `optimize` function in R to estimate the mode at each time step.

To make the algorithms as efficient as possible we use Rao-Blackwellisation to reduce the variance of our estimates. For this we can marginalise the second component of the state $\dot{\mu}_t$ as the likelihood only depends on μ_t so the distribution of $\dot{\mu}_t|\mu_t$ can be updated by using only its mean and variance. This improves the overall approximation by allowing the second component of each particle to act as a normal distribution rather than a point mass. We therefore have particles of the form $x_t^{(i)} = (\mu_t^{(i)}, \dot{m}_t^{(i)}, \tau_t^{2(i)})'$, where $\dot{\mu}_t | \{\mu_t = \mu_t^{(i)}\} \sim \mathcal{N}(\dot{m}_t^{(i)}, \tau_t^{2(i)})$.

To create a marginalised particle filter it helps to think each particle $x_{t-1}^{(i)}$ as a kernel approximation to $p(\mu_{t-1}, \dot{\mu}_{t-1}|y_{1:t-1})$ of the form

$$\phi^{(i)}(\mu_{t-1}, \dot{\mu}_{t-1}) := \mathcal{N}(\mu_{t-1}, \dot{\mu}_{t-1}|\eta_{t-1}^{(i)}, K_{t-1}^{(i)}),$$

with

$$\eta_{t-1}^{(i)} := \begin{pmatrix} \mu_{t-1}^{(i)} \\ \dot{m}_{t-1}^{(i)} \end{pmatrix}, \qquad K_{t-1}^{(i)} := \begin{pmatrix} 0 & 0 \\ 0 & \tau_{t-1}^{2(i)} \end{pmatrix}.$$

This leads to the approximation of $p(\mu_t, \dot{\mu}_t|y_{1:t-1})$ by

$$\begin{aligned} \pi^{(i)}(\mu_t, \dot{\mu}_t) &:= \int f(\mu_t, \dot{\mu}_t|\mu_{t-1}, \dot{\mu}_{t-1})\, \phi^{(i)}(\mu_{t-1}, \dot{\mu}_{t-1})\, \mathrm{d}\mu_{t-1}\, \mathrm{d}\dot{\mu}_{t-1} \\ &= \mathcal{N}(\mu_t, \dot{\mu}_t|F\eta_{t-1}^{(i)}, Q + FK_{t-1}^{(i)}F'). \end{aligned}$$

To create the new particle $x_t^{(i)}$ we therefore use standard auxiliary particle filtering with target density

$$q^{\text{opt}}(\mu_t|x_{t-1}^{(i)}, y_t)\beta_t^{(i)} = \pi^{(i)}(\mu_t)\, g(y_t|\mu_t)\, w_{t-1}^{(i)}$$

to sample $\mu_t^{(i)}$ and then update the mean and variance of $\dot{\mu}_t|\{\mu = \mu_t^{(i)}\}$ with that of $\pi^{(i)}(\dot{\mu}_t|\mu_t^{(i)})$. For this we replace the likelihood by the approximation (12) to give us a constrained normal sampling density for $\mu_t^{(i)}$ and approximate the optimal re-sampling weights with

$$\beta_t^{(i)} \simeq \frac{\pi^{(i)}(\hat{\mu}_t)\, g(y_t|\hat{\mu}_t)\, w_{t-1}^{(i)}}{q(\hat{\mu}_t|x_{t-1}^{(i)}, y_t)},$$

where $\hat{\mu}_t$ is the mean of the sampling density $q(\mu_t|x_{t-1}^{(i)}, y_t)$.

For the backwards filter we again start by defining $\tilde{F} := \Sigma_t F' \Sigma_{t+1}^{-1}$ and $\tilde{Q} := \Sigma_t F' \Sigma_{t+1}^{-1} Q F^{'-1}$, where Σ_t is the variance of the normal prior at time t. It can then be shown that $p(\mu_t, \dot{\mu}_t|\mu_{t+1}, \dot{\mu}_{t+1})$ is equal to

$$\mathcal{N}\left(\begin{pmatrix} \mu_t \\ \dot{\mu}_t \end{pmatrix}\middle|\tilde{F}\begin{pmatrix} \mu_{t+1} \\ \dot{\mu}_{t+1} \end{pmatrix} + \tilde{Q}\Sigma_t^{-1}\begin{pmatrix} \hat{\mu}_t \\ \hat{\dot{\mu}}_t \end{pmatrix}, \tilde{Q}\right),$$

where $(\hat{\mu}_t, \hat{\dot{\mu}}_t)'$ is the mean of the prior at time t. We then combine this with a kernel $\phi^{(i)}(\mu_{t+1}, \dot{\mu}_{t+1})$ created from $x_{t+1}^{(i)}$ to give the density

$$\tilde{\pi}^{(i)}(\mu_t, \dot{\mu}_t) := \mathcal{N}\left(\begin{pmatrix} \mu_t \\ \dot{\mu}_t \end{pmatrix}\middle|\tilde{F}\eta_{t+1}^{(i)} + \tilde{Q}\Sigma_t^{-1}\begin{pmatrix} \hat{\mu}_t \\ \hat{\dot{\mu}}_t \end{pmatrix}, \tilde{Q} + \tilde{F}K_{t+1}^{(i)}\tilde{F}'\right).$$

We now proceed in exactly the same way as with the forwards filter using $\tilde{\pi}$ instead of π to sample $x_t^{(i)}$.

Finally, for our new smoothing algorithm, it can be shown that our target for $\mu_t^{(i)}$ in this marginalised setting is

$$\bar{q}^{\text{opt}}(\mu_t|x_{t-1}^{(j)}, y_t, x_{t+1}^{(k)})\, \bar{\beta}_t^{(j,k)} = \pi^{(j)}(\mu_t)\, w_{t-1}^{(j)} \cdot \frac{g(y_y|\mu_t)}{p(\mu_t)} \cdot \tilde{\pi}^{(k)}(\mu_t)\, w_{t+1}^{(k)}.$$

This leads us to sample $\mu_t^{(i)}$ as before using the density proportional to the product of $\pi^{(j)}(\mu_t)$, $\tilde{\pi}^{(k)}(\mu_t)$ and $p(\mu_t)^{-1}$ in place of $\pi^{(j)}(\mu_t)$. We can then calculate the mean and variance of $\dot{\mu}_t|\mu_t^{(i)}$ from the distribution proportional to

$$\frac{\pi^{(j)}(\dot{\mu}_t|\mu_t^{(i)})\,\tilde{\pi}^{(k)}(\dot{\mu}_t|\mu_t^{(i)})}{p(\dot{\mu}_t|\mu_t^{(i)})}.$$

The filter and backwards filter re-sampling weights were used again for the suboptimal $O(N)$ version of our algorithm.

For both the filter and the backwards filter the initial step was sampled using standard importance sampling as the target density is available in closed form and using it rather than propagating the prior greatly improves the algorithm. We also used the stratified sampling algorithm of Carpenter et al. (1999) in both the filters and our new algorithm to reduce the Monte Carlo error of re-sampling.

Since we chose not to include the data from 1993, for this time step in each of the above algorithms we proceed without the likelihood term $g(y_t|\mu_t)$.

Appendix B EM algorithm for parameter estimates

For our athletics model we require estimates of the fixed likelihood parameters $\theta = (\sigma, \xi)$ which we intend to obtain from the EM algorithm of Dempster et al. (1977). To do this we aim to maximise the likelihood $p(y_{1:T}|\theta)$ by iteratively maximising

$$Q(\theta|\theta^{(n)}) := \mathrm{E}\Big(\log(p(X_{0:T}, y_{1:T}|\theta)) \,\Big|\, y_{1:T}, \theta^{(n)}\Big)$$

to give $\theta^{(n+1)}$.

Since the parameters θ do not appear in the state density, the joint log likelihood can be written as

$$\log(p(x_{0:T}, y_{1:T}|\theta)) = \log(p(x_0)) + \sum_{t=1}^{T} \log(f(x_t|x_{t-1})) + \sum_{t=1}^{T} \log(g(y_t|x_t, \theta)).$$

We therefore have

$$\begin{aligned} Q(\theta|\theta^{(n)}) &= \text{const} + \sum_{t=1}^{T} \mathrm{E}\Big(\log(g(y_t|X_t, \theta)) \,\Big|\, y_{1:T}, \theta^{(n)}\Big) \\ &\simeq \text{const} + \sum_{t=1}^{T} \sum_{i=1}^{N} \log(g(y_t|x_t^{(i)}, \theta)) w_t^{(i)}, \end{aligned}$$

where $(x_t, w_t)^{(i)}$ are weighted particles approximating $p(x_t|y_{1:T}, \theta^{(n)})$. Thus we only require particles from the marginal smoothing densities to estimate the expectation so our new algorithm can be used directly. To estimate parameters from the state density with the EM algorithm, pairs of particles approximating $p(x_{t-1}, x_t|y_{1:T}, \theta^{(n)})$ are required which we note are available from our algorithm as either $(\bar{x}_{t-1}^{(i)}, \tilde{x}_t^{(k_i)})$ at time $t-1$ or as $(x_{t-1}^{(j_i)}, \bar{x}_t^{(i)})$ at time t.

The EM algorithm therefore proceeds as follows. We start with an initial estimate of our parameters, $\theta^{(0)}$. Then, given our current estimate $\theta^{(n)}$, we use our algorithm to generate particles from each marginal smoothing density $p(x_t|y_{1:T}, \theta^{(n)})$. Then we use numerical optimisation (such as the `optim` function in R) to maximise $Q(\theta|\theta^{(n)})$ to give us a new estimate $\theta^{(n+1)}$. For our athletics example we fitted a Kalman filter with the same state density but normal observations y_t to estimate the state $x_{1:T}$ and took maximum likelihood estimates of σ and ξ using the shifted data $y_t - \hat{x}_t$ to initialise the EM algorithm.

References

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal filtering.* Prentice-Hall Englewood Cliffs, New Jersey.

Briers, M., Doucet, A., and Maskell, S. (2004). Smoothing algorithms for state-space models. *Submitted to IEEE Transactions on Signal Processing.*

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

Casella, G. and Robert, C. P. (2001). Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.

Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values.* Springer, London.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Doucet, A., Briers, M., and Sénécal, S. (2006). Efficient Block Sampling Strategies for Sequential Monte Carlo Methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

Fearnhead, P. (2005). Using random Quasi-Monte Carlo within particle filters, with application to financial time series. *Journal of Computational and Graphical Statistics*, 14:751–769.

Fearnhead, P. (2008). Computational Methods for Complex Stochastic Systems: A Review of Some Alternatives to MCMC. *To appear in Statistics and Computing.*

Gaetan, C. and Grigoletto, M. (2004). Smoothing Sample Extremes with Dynamic Models. *Extremes*, 7(3):221–236.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo Smoothing for Nonlinear Time Series. *Journal of the American Statistical Association*, 99(465):156–169.

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

Hürzeler, M. and Künsch, H. R. (1998). Monte Carlo Approximations for General State-Space Models. *Journal of Computational and Graphical Statistics*, 7(2):175–193.

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

Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.

Kitagawa, G. and Gersch, W. (1985). A Smoothness Priors Time-Varying AR Coefficient Modeling of Nonstationary Covariance Time Series. *IEEE Transactions on Automatic Control*, 30(1):48–56.

Larsson, P. (2008). Track & Field all-time performances. `http://www.alltime-athletics.com/`. [last accessed 22 April 2008].

Liu, J. S. and Chen, R. (1995). Blind Deconvolution Via Sequential Imputations. *Journal of the American Statistical Association*, 90(430):567–576.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93(443):1032–1044.

Liu, J. S. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*, pages 197–224. Springer.

Pitt, M. K. and Shephard, N. (1999). Filtering Via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590–599.

Robinson, M. E. and Tawn, J. A. (1995). Statistics for Exceptional Athletics Records. *Applied Statistics*, 44(4):499–511.

Smith, R. L. (1997). Comment on "Statistics for Exceptional Athletics Records", by Robinson, M. E. and Tawn, J. A. *Applied Statistics*, 46(1):123–128.

Smith, R. L. and Miller, J. E. (1986). A Non-Gaussian State Space Model and Application to Prediction of Records. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(1):79–88.

Tanizaki, H. and Mariano, R. S. (1994). Prediction, Filtering and Smoothing in Non-Linear and Non-Normal Cases Using Monte Carlo Integration. *Journal of Applied Econometrics*, 9(2):163–179.