AISM (2007) 59: 349–366
DOI 10.1007/s10463-006-0053-9

Nicolas Chopin

Dynamic detection of change points in long time series

Received: 23 March 2005 / Revised: 8 September 2005 / Published online: 17 June 2006

Abstract We consider the problem of detecting change points (structural changes) in long sequences of data, whether in a sequential fashion or not, and without assuming prior knowledge of the number of these change points. We reformulate this problem as the Bayesian filtering and smoothing of a non standard state space model. Towards this goal, we build a hybrid algorithm that relies on particle filtering and Markov chain Monte Carlo ideas. The approach is illustrated by a GARCH change point model.

Keywords Change point models · GARCH models · Markov chain Monte Carlo · Particle filter · Sequential Monte Carlo · State state models

1 Introduction

The assumption that an observed time series follows the same fixed stationary model over a very long period is rarely realistic. In economic applications for instance, common sense suggests that the behaviour of economic agents may change abruptly under the effect of economic policy, political events, etc. For example, Mikosch and Stărică (2003, 2004) point out that GARCH models fit very poorly too long sequences of financial data, say 20 years of daily log-returns of some speculative asset. Despite this, these models remain highly popular, thanks to their forecast ability (at least on short to medium-sized time series) and their elegant simplicity (which facilitates economic interpretation). Against the common trend of building more and more sophisticated stationary models that may spuriously provide a better fit for such long sequences, the aforementioned authors argue that GARCH models remain a good 'local' approximation of the behaviour of financial data,

N. Chopin
School of Mathematics, University of Bristol, University Walk, Bristol BS8 1TW, UK
E-mail: nicolas.chopin@bris.ac.uk

provided that their parameters are allowed to change 'from time to time', that is at some unknown dates denominated 'change points'.

This paper addresses the general problem of detecting change points in time series data, whether in an on-line fashion or not, within a Bayesian framework, and without prior knowledge of the exact number of change points. We show that, in full generality, a change point model can be formulated as a non standard state space model, and we develop a particle filter and smoother that is specifically adapted to the particular structure of this state space model. In particular, a difficulty with this reformulated state space model is that its hidden process is not mixing well, which makes naive particle filtering inefficient. We address this issue by introducing local MCMC (Markov chain Monte Carlo) moves that are conditional on the date of latest change; appropriate references for particle filtering and MCMC are given in next section.

The Bayesian approach is particularly suited to the problem of change point detection as it does not resort to asymptotic justifications, which would be haphazard in a situation where each considered parametric model is restricted to a finite, possibly small interval of time. There is already an important Bayesian literature on multiple change point analysis; see in particular McCulloch and Tsay (1993), Barry and Hartigan (1993), Stephens (1994), Chib (1998), Gerlach et al. (2000), and also first example in Green (1995); most of them rely on MCMC methodology. The first advantage of our algorithm is that, in contrast to MCMC approaches, and thanks to its sequential nature, it allows for performing sequential analysis in a computationally efficient way. This is useful in an number of applied contexts. In the aforementioned example above, finance analysts are interested in having a 'quick' update (say 1 s of cpu time) of inference as soon as the day's observation is available, rather than re-analysing the whole data set through some off-line algorithm, which may take, say, several minutes, for long data sets.

A second advantage of our algorithm is that it can be turned into a particle smoother through a very simple modification. Thus it can also be applied to off-line analysis, when a complete data set is made available at once. We argue that, even in such off-line scenarios, our algorithm compares favourably to previous methods when the sample size T gets large. In particular, we will see that its computational cost increases linearly in T, and is comparable to that of a MCMC algorithm corresponding to the same model without change point, e.g. a single GARCH model in the above example. In contrast, MCMC samplers for change point models typically have a $O(T^2)$ computational cost, while their convergence properties tend to deterioriate for larger values of T. A notable exception is the $O(T)$ algorithm of Gerlach et al. (2000), in the specific case of Gaussian linear state space models. We shall elaborate on these points in the paper.

The paper is organised as follows. In Sect. 2, we formulate the problem of sequentially detecting change points as the filtering of a non-standard Bayesian state space model. In Sect. 3 we develop a particle filter algorithm specifically adapted to this non standard model. Section 4 explains how to extend our approach to off-line scenarios, when data need to be processed as a whole rather than sequentially. Section 5 presents some numerical experiments. Section 6 gives concluding remarks.

2 State space representation of change point models

We consider a generic discrete time series model indexed by a changing parameter $\theta_t, t \geq 1$,

$$y_t \sim p\left(y_t | y_{1:t-1}, \theta_t\right), \tag{1}$$

where $y_{1:t-1}$ denotes the subsequence $y_1, \ldots, y_{t-1}$. The changing parameter is assumed to follow a piece-wise constant process

$$\theta_t = \xi_k, \quad \text{provided that } \sum_{i=1}^{k-1} \delta_i < t \leq \sum_{i=1}^{k} \delta_i,$$

that is, for the δ_1 first observations, the parameter value is ξ_1, then for the δ_2 following observations, it is ξ_2, etc. The behaviour of the observed sequence within one of these periods of time will be informally referred to as a 'regime'. The δ_i's and the ξ_k's are unknown, and assigned some prior densities $\pi_\delta(\cdot)$ and $\pi_\xi(\cdot)$, the former with support over the set of positive integers. For simplicity these quantities are assumed to be prior independent and identically distributed, but we will see later that this assumption can be relaxed.

We propose to reformulate this generic model into a non-standard state space model, that is a model of an observed process (y_t) whose behaviour is expressed conditional upon a hidden Markov process (x_t). Let d_t the duration at time t since latest change, that is $d_t = t - \delta_1 - \cdots - \delta_{k-1}$ if in regime k, let $x_t = (\theta_t, d_t)$, then, conditional upon $x_{t-1} = (\theta_{t-1}, d_{t-1})$,

$$x_t = (\theta_t, d_t) = \begin{cases} (\theta_{t-1}, d_{t-1} + 1) & \text{with probability } \pi_\delta(\delta \geq d_{t-1} + 1 | \delta \geq d_{t-1}), \\ (\xi^*, 1) & \text{with probability } \pi_\delta(\delta = d_{t-1} | \delta \geq d_{t-1}), \end{cases} \tag{2}$$

where ξ^* is drawn independently from the prior $\pi_\xi(\cdot)$, and $\pi_\delta(\cdot | \delta \geq d_{t-1})$ refer to probabilities conditional on $\delta \geq d_{t-1}$, where δ is random (d_{t-1} is fixed) and follows the prior distribution π_δ.

In this particular context, state filtering, that is the sequential derivation of $p(x_t | y_{1:t})$, amounts to dynamically estimating the date of latest change, and the parameter value since then. In contrast, state smoothing, that is the derivation of $p(x_{1:T} | y_{1:T})$ allows for jointly estimating all the change points and the corresponding regime parameters, for a complete data set $y_{1:T}$. Since filtering and smoothing cannot be carried out analytically outside of some specific cases (including the normal linear state-space model, Kalman and Bucy, 1961), we will develop in next section a particle filter algorithm for these purposes. Note one could also simulate the smoothing distribution through Gibbs sampling (McCulloch and Tsay, 1993): denoting u_t the indicator function of a change, $u_t = 1_{d_t=1}$, such a sampler would involve simulations of u_t conditional on the $u_{t'}$, $t' \neq t$, and the other parameters. Unfortunately, this leads to a $O(T^2)$ algorithm, as the cost of simulating one single u_t is $O(T)$. Morever, there is a general agreement in the literature on the poor performance of this approach for general state space models (Carter and Kohn, 1994; Früwirth-Schnatter, 1994; de Jong and Shephard, 1995): the presence of numerous, strongly correlated components in the augmented joint density (of the

u_t's and parameters) tend to hinder the congervence of a Gibbs sampler. Note that, although it does rely on such a Gibbs structure, the reservible jump algorithm that Green (1995) specifically derived for change point models is also $O(T^2)$, as new change points are proposed by drawing from a $[1, T]$ uniform distribution, and are accepted or rejected according to a acceptance probability involving T terms; thus, the simulation of 'good' change points requires $O(T^2)$ operations on average. Finally, and as mentioned in the introduction, Gerlach et al. (2000), in the specific case of linear Gaussian models, obtain a $O(T)$ sampler, with better convergence properties, by marginalising out the parameters in the above conditionals. (We believe their algorithm is, in this specific case, more efficient than the one developed in this paper.)

3 Particle filtering

3.1 A first algorithm

Consider the problem of filtering a state space model with observed process (y_t) and hidden Markov process (x_t). Particle filtering consists of generating and updating a stream of weighted simulations $x_t^{(j)}$, $j = 1, \ldots, H$, commonly denominated 'particles', through iterative steps described below.

Step 1. Simulate independently for $j = 1, \ldots, H$,

$$x_t^{(j)} \sim p(x_t | x_{t-1}^{(j)}),$$

where $p(x_t|x_{t-1})$ stands for the conditional density of hidden Markov chain (x_t).
Step 2. Weight particles, for $j = 1, \ldots, H$,

$$w_t^{(j)} = p(y_t | y_{1:t-1}, x_t^{(j)}),$$

where $p(y_t|y_{1:t-1}, x_t)$ stands for the conditional likelihood of observed process (y_t).
Step 3. 'Resample' the particles, that is replace the current set of particles by a set containing $n_t^{(j)}$ replicates of $x_t^{(j)}$, $j = 1, \ldots, H$, where $n_t^{(j)}$ is random and fulfills $\mathbb{E}[n_t^{(j)}] = H w_t^{(j)} / \sum_{j=1}^H w_t^{(j)}$ and $\sum_{j=1}^H n_t^{(j)} = H$.
Step 4. $t \leftarrow t + 1$. Go to Step 1.

Step 1 of the first iteration ($t = 1$) generates independently the $x_1^{(j)}$'s from the prior distribution on x_1. The iteration of Step 1 and Step 2 is equivalent to a sequence of importance sampling operations: Step 1 transforms the target density from $p(x_{t-1}|y_{1:t-1})$ to $p(x_t|y_{1:t-1})$; then Step 2 computes the importance weights corresponding to the passage from $p(x_t|y_{1:t-1})$ to $p(x_t|y_{1:t})$. For our particular model the Markov transition $p(x_t|x_{t-1})$ in Step 1 is given by (2) and the conditional likelihood $p(y_t|y_{1:t-1}, x_t)$ in Step 2 is simply $p(y_t|y_{1:t-1}, \theta_t)$, as defined by (1), since $x_t = (\theta_t, d_t)$. The third step is a 'Darwinian' procedure that reproduces the most representative particles (those with large weights) and eliminates the others. A simple way of resampling is to draw independently H times from the

multinomial distribution which produces x_t^j with a probability proportional to $w_t^{(j)}$ Gordon et al. (1993), but more efficient alternatives exist, such as deterministic resampling (Kitagawa, 1996) or residual resampling (Liu and Chen, 1998). For a more general presentation of particle filters and their numerous applications, the reader is referred to Künsch (2001), Doucet et al. (2001) and references therein.

The weighted particle sample produced by the second step approximates the true filtering density $p(x_t|y_{1:t})$ in the sense that

$$\frac{\sum_{j=1}^{H} w_t^{(j)} \varphi\left(x_t^{(j)}\right)}{\sum_{j=1}^{H} w_t^{(j)}} \to \mathbb{E}\left[\varphi(x_t)|y_{1:t}\right]$$

almost surely as $H \to +\infty$, for any φ such that the expectation above exists. Under appropriate assumptions, asymptotic normality also holds (Chopin, 2004)

$$H^{1/2}\left(\frac{\sum_{j=1}^{H} w_t^{(j)} \varphi\left(x_t^{(j)}\right)}{\sum_{j=1}^{H} w_t^{(j)}} - \mathbb{E}\left[\varphi(x_t)|y_{1:t}\right]\right) \xrightarrow{\mathcal{D}} \mathcal{N}\{0, V_t(\varphi)\}$$

for a given sequence of asymptotic variances $V_t(\varphi)$.

While many variants exist and may be more efficient (see, for instance, Pitt and Shephard, 1999), the algorithm above, initially proposed by Gordon et al. (1993), typically works well for filtering a state space model whose hidden Markov process mixes well. The depletion in simulated values due to resampling is counterbalanced by the rejuvenation due to simulating from the Markov transition of the model. Under appropriate conditions (Del Moral and Miclo, 2000; Chopin, 2004), the sequence of $V_t(\varphi)$'s remains below some constant bound. Additionally the computational load remains constant along iterations.

Unfortunately, the Markov transition of our change point model does not mix properly as its first component remains constant with positive probability, see (2). We develop in the following sections various strategies for improving this initial algorithm.

3.2 Rao-Blackwellisation of the discrete component

Consider the simulation of $x_t^{(j)}$ conditional upon $x_{t-1}^{(j)}$ in Step 1. Given the particular structure of $p(x_t|x_{t-1})$, see (2), this would involve the simulation of a binary component, that is whether a change point occurs at time t or not. Since the probability of this event can be computed exactly, this binary component can be 'Rao-Blackwellised', that is to say, marginalised out in order to achieve variance reduction (Casella and Robert, 1996). The application of Rao-Blackwellisation to particle filters has been formalised by Doucet et al. (2000), see also Chen and Liu (2000), and proved to always lead to smaller asymptotic variances by Chopin (2004).

Assume $x_{t-1}^{(j)} = (\xi, d)$, and create two particles, each corresponding to one of the two possibilities, with weights:

$$\begin{aligned} x_t^{(j,1)} &= (\xi, d+1), & w_t^{(j,1)} &= \pi_\delta(\delta \geq d+1|\delta \geq d)\, p(y_t|y_{1:t-1}, \theta_t = \xi), \\ x_t^{(j,2)} &= (\xi^*, 1), & w_t^{(j,2)} &= \pi_\delta(\delta = d|\delta \geq d)\, p(y_t|y_{1:t-1}, \theta_t = \xi^*), \end{aligned}$$

where ξ^* is drawn independently from π_ξ. In this way we obtain a set of $2H$ particles, which can be resampled with respect to the weights $w_t^{(j,1)}$, $w_t^{(j,2)}$, so as to obtain H resampled particles. Note the probabilities $\pi_\delta(\delta \geq d+1|\delta \geq d)$, $\pi_\delta(\delta = d|\delta \geq d)$ do not need to be available in closed form. Since

$$\pi_\delta(\delta = d|\delta \geq d) = 1 - \pi_\delta(\delta \geq d+1|\delta \geq d) = \frac{\pi_\delta(d)}{1 - \sum_{k=1}^{d-1} \pi_\delta(k)},$$

one can store the partial sums $\sum_{k=1}^{d-1} \pi_\delta(k)$ when they are computed for the first time and re-use them as often as necessary.

This algorithm has an interesting connection with the optimal proposal strategy of Doucet et al. (2000). To see this, let's point out first that, in the initial algorithm, the next states can also be simulated from an arbitrary distribution $q_t(x_t|x_{t-1})$; the weight function have then the more general expression $w_t(x_{t-1}, x_t) = p(y_t|y_{1:t-1}, x_t)p(x_t|x_{t-1})/q_t(x_t|x_{t-1})$. The above authors have shown the proposal distribution that minimises the variance of the weights is $p(x_t|x_{t-1}, y_{1:t})$. In our case simulating from this optimal distribution would amount exactly to choose between $x_t^{(j,1)}$ and $x_t^{(j,2)}$, with respective probabilities proportional to $w_t^{(j,1)}$ and $w_t^{(j,2)}$. Thus this optimal strategy would have exactly the same computational cost as our Rao-Blackwellised particle filter, but the latter leads to even further variance reduction as the randomness inherent to the simulation of the binary component is removed.

3.3 Fractional move

Our Rao-Blackwellised particle filter remains highly inefficient due to the lack of mixing properties of the hidden process (x_t), as explained in (3). Considering the problem posed by constant parameters included in the hidden Markov process, Gilks and Berzuini (2001) proposed to create an artificial rejuvenation effect by 'moving' the particles through a MCMC kernel which is invariant by the current target density. Thanks to the invariance property, the asymptotic results given in §3 still hold. The reader is referred to Robert and Casella (2004) for a general presentation of MCMC methods.

Since the degeneracy of our Rao-Blackwellised algorithm is mainly due to the presence of constant parameters in the state variable, we propose to 'move' only the θ_t-component of each particle $x_t^{(j)} = (\theta_t^{(j)}, d_t^{(j)})$ through a MCMC kernel with invariant distribution

$$\eta_t^{(j)}(\xi) = p\left(\theta_t = \xi | d_t = d_t^{(j)}, y_{1:t}\right) \propto \pi_\xi(\xi) \prod_{k=t-d_t^{(j)}+1}^{t} p\left(y_k|y_{1:k-1}, \theta_k = \xi\right). \quad (3)$$

This comes down to implement a MCMC move with respect to the model corresponding to the current period, since latest change. This is conceptually simpler and computationally cheaper than considering the complete model corresponding to the observations up to t; in particular the computational cost should be $O(d_t^{(j)})$ rather than $O(t)$.

Since in most applications one wishes to have a constant computational cost for each iteration, we propose to move only a subset S of the particle system, under

the constraint $\sum_{j\in S} d_t^{(j)} \approx C$, for some constant C. This subset is obtained by drawing randomly without replacement among the resampled particles, until the sum of the $d_t^{(j)}$'s is larger than C. Then the last selected particle is discarded, and the remaining particles are moved.

Another rationale for this fractional move strategy is that the degeneracy effect due to constant parameters tend to decrease as the number of observations in the considered regime accumulates. As the corresponding conditional distribution concentrates on a smaller and smaller region, exploring locally becomes less and less necessary; see Chopin (2002) for a more formal argument on this phenomenon. We will see in our simulation experiments that this strategy indeed stabilises the Monte Carlo error over iterations, even if the numbers of moved particles becomes eventually extremely small.

In summary, our Rao-Blackwellised fractional move particle filter can be described as follows.

Step 1. Simulate independently for $j = 1, \ldots, H$, $\xi^{(j)} \sim \pi_\xi(\cdot)$, and conditional on $x_{t-1}^{(j)} = (\theta_{t-1}^{(j)}, d_{t-1}^{(j)})$,

$$x_t^{(j,1)} = (\theta_{t-1}^{(j)}, d_{t-1}^{(j)} + 1),$$
$$x_t^{(j,2)} = (\xi^{(j)}, 1).$$

Step 2. Reweight particles,
for $j = 1, \ldots, H$,

$$w_t^{(j,1)} = \pi_\delta(\delta \geq d_{t-1}^{(j)} + 1 | \delta \geq d_{t-1}^{(j)}) p(y_t | y_{1:t-1}, \theta_t = \theta_{t-1}^{(j)})$$
$$w_t^{(j,2)} = \pi_\delta(\delta = d_{t-1}^{(j)} | \delta \geq d_{t-1}^{(j)}) p(y_t | y_{1:t-1}, \theta_t = \xi^{(j)})$$

Step 3. 'Resample' the $2H$ particles with respect to the weights $w_t^{(j,1)}$, $w_t^{(j,2)}$, so as to obtain H resampled particles.
Step 4. Select a subset S of the resampled particles such that $\sum_{j\in S} d_t^{(j)} \leq C$ as explained above, and for each selected particle $x_t^{(j)} = (\theta_t^{(j)}, d_t^{(j)})$, replace $\theta_t^{(j)}$ by

$$\widetilde{\theta}_t^{(j)} \sim k_t^{(j)}(\theta_t^{(j)}, \cdot),$$

where $k_t^{(j)}$ is a MCMC kernel with invariant distribution $\eta_t^{(j)}$ as defined in (3).
Step 5. $t \leftarrow t + 1$. Go to Step 1.

3.4 Practical implementation of the MCMC moves

Recall that the MCMC move in Step 4 of our algorithm is built with respect to the time-series model restricted to the current period (since latest change point), that is, its invariant distribution is given by (3). An obvious choice, especially when Gibbs sampling cannot be implemented because full conditionals are not available, as in

our GARCH example in §5, is to update the parameter values through a Gaussian random walk Metropolis-Hastings, that is $\xi|\theta_t^{(j)} \sim q(\xi|\theta_t^{(j)}) = N(\theta_t^{(j)}, \widehat{\Sigma}_t)$ and

$$\widetilde{\theta}_t^{(j)} = \begin{cases} \xi & \text{with probability} 1 \bigwedge \frac{q(\theta_t^{(j)}|\xi)\eta_t^{(j)}(\xi)}{q(\xi|\theta_t^{(j)})\eta_t^{(j)}(\theta_t^{(j)})}, \\ \theta_t^{(j)} & \text{otherwise.} \end{cases}$$

The calibration of the random step in random walk algorithms is always an important issue, as too small steps slow down the exploration of the target distribution, and too large steps are rarely accepted. In this context however, we have the opportunity to scale our proposal distribution to the covariance matrix of the particle sample itself:

$$\widehat{\Sigma}_t = \gamma^2 \left[\frac{1}{H} \sum_{j=1}^{H} \theta_t^{(j)} \left(\theta_t^{(j)}\right)^{\mathrm{T}} - \left(\frac{1}{H} \sum_{j=1}^{H} \theta_t^{(j)} \right) \left(\frac{1}{H} \sum_{j=1}^{H} \theta_t^{(j)} \right)^{\mathrm{T}} \right],$$

where γ is a tuning parameter. This conveniently take into account the information contained in the particle system on the range (and correlations between components) of 'plausible' values for the $\theta_t^{(j)}$.

In our simulation experiments (see §5) we found that values between 0.5 and 1 for γ led to the best performance of the algorithm, in terms of Monte Carlo error. Interestingly, this led to an average acceptance rate over iterations of about 25%, which is considered as 'optimal' in standard implementations of random walk algorithms (Roberts et al. 1997).

We also experimented with a Langevin proposal strategy, namely,

$$q\left(\xi|\theta_t^{(j)}\right) = N\left(\theta_t^{(j)} + \frac{1}{2}\left\{H_t^{(j)}\right\}^{1/2} \nabla \log \eta_t^{(j)}\left(\theta_t^{(j)}\right), H_t^{(j)}\right) \tag{4}$$

where $\nabla \log \eta_t^{(j)}$ denotes the gradient function of $\log \eta_t^{(j)}$, and

$$H_t^{(j)} = -\gamma^2 \left\{ \nabla' \nabla \log \eta_t^{(j)}\left(\theta_t^{(j)}\right) \right\}^{-1},$$

that is $-\gamma^2$ times the inverse of the Hessian matrix of $\log \eta_t^{(j)}$ at point $\theta_t^{(j)}$. For $\gamma = 1$, this proposal density can be seen as a second-order approximation of target density $\eta_t^{(j)}$ around $\theta_t^{(j)}$ (Robert and Casella, 2004, p. 266). The gradient term pushes the exploration towards zones of higher posterior probability, whereas the Hessian term ensures that each proposed step is scaled with respect to the appropriate target density.

Our motivation for this second strategy was that it may lead to significant improvements at times where the particle sample is quite heterogeneous, especially when particles 'disagree' on whether a change has occurred recently or not. A different scaling for each particle would then be more appropriate. In our simulations however we did not observe substantial improvements, especially in regard of the increased computational cost. This may indicate that our simple random walk strategy performs 'well enough'.

3.5 Positive discrimination strategy

In our simulation studies we did notice that the filtered estimates could be less stable around change times, but for a reason independent of the implemented move strategy. Consider a time t when a change does occur. Typically the probability that this change is detected at time t is small, as a single observation is not enough to provide significant evidence of a change. Thus few particles that correspond to a change at time t, see §3.2, survive the resampling step, even if they may become predominant shortly afterwards. Therefore, and despite the MCMC step, there may be a temporary lack of diversity among particles in the following iterations. To avoid this, we propose to boost the population size of those 'young' particles through 'positive discrimination': at iteration t, for any particle such that its d_t-component is equal to d with $d \leq k$, multiply its weight before resampling by λ^{k-d+1}; then after the resampling and the move steps, assign weight $\lambda^{-(k-d+1)}$ to any particle such that its d_t-component equals d (while the other particles are assigned an unit weight). Note that since particles do not have equal weight after the move step anymore, these weights must be propagated appropriately in Step 2, that is

$$w_t^{(j,1)} = w_{t-1}^{(j)} \pi_\delta \left(\delta \geq d_{t-1}^{(j)} + 1 | \delta \geq d_{t-1}^{(j)}\right) p\left(y_t | y_{1:t-1}, \theta_t = \theta_{t-1}^{(j)}\right)$$
$$w_t^{(j,2)} = w_{t-1}^{(j)} \pi_\delta \left(\delta = d_{t-1}^{(j)} | \delta \geq d_{t-1}^{(j)}\right) p\left(y_t | y_{1:t-1}, \theta_t = \xi^{(j)}\right)$$

where $w_{t-1}^{(j)}$ stands for the weight of resampled (possibly moved) particle $\theta_t^{(j)}$. This positive discrimination strategy incurs virtually no additional computational cost, and did increase the stability of the filtered estimated in our simulations, at least locally around changes, see §5. We have taken $k = 10$ and $\lambda = 1.414$ in these simulations, but any values such that λ^k is between 10 and 100 seems to lead to equally acceptable results.

4 Particle smoothing

Our algorithm only requires a minor modification in order to provide simulated samples from the smoothing distribution $p(x_{1:T}|y_{1:T})$, that is the posterior distribution of the whole state trajectory until some final time T. As already said, this is useful when the data $y_{1:T}$ need to be processed as a whole rather than sequentially.

A first solution would be to carry forward the past values $x_{1:t-1}^{(j)}$ of each particle $x_t^{(j)}$ along iterations. Then the resampled trajectories obtained at the last iteration ($t = T$) should approximatively represent draws from the smoothing distribution. This is very inefficient in practice as these samples tend to be extremely correlated in the first dimensions. Typically even for a large number of particles, the $x_1^{(j)}$'s may all take the same value (Kitagawa, 1996).

Rather, we propose to reconstruct the simulated trajectories backwards, starting from the set of resampled particles obtained at the last iteration of our algorithm. Draw with replacement such a particle, and denote it $\widetilde{x}_T$, say $\widetilde{x}_T = (\xi, d)$. By virtue of resampling, this value can be considered as a draw from $p(x_T|y_{1:T})$. Clearly the previous states are already known up to time $T' = T - d + 1$, that is

$\widetilde{x}_t = (\xi, d - T + t)$, for $T' \leq t \leq T$. We then need to append some $x_{T'-1}$ drawn from distribution

$$p(x_{T'-1}|x_{T'} = (\xi, 1), y_{1:T}) = p(x_{T'-1}|u_{T'} = 1, y_{1:T'-1}),$$

where $u_{T'}$ is the indicator variable of a change point occurrence at time T'. Such a draw can be obtained at iteration T' of our filtering algorithm: for each resampled particle $x_{T'}^{(j)}$ such that $d_{T'}^{(j)} = 1$, store its 'ancestor' $x_{T'-1}^{(j')}$, that is the particle from which $x_{T'}^{(j)}$ has been simulated. Then draw with replacement one of these stored values, say $\widetilde{x}_{T'-1} = (\xi', d')$, reconstruct the sequence $\widetilde{x}_{T'-d':T'-1}$ up to time $T' - d$, and process by induction.

In summary, one has to add the following step to our algorithm to turn it into a smoothing algorithm:

Step 4b is. For each resampled particle $x_t^{(j)}$ such that $d_t^{(j)} = 1$, $j = 1, \ldots, H$, store a copy of its ancestor $x_{t-1}^{(j')}$.

Then one may obtain at the final stage of the algorithm as many smoothed trajectories as required, using the backward construction principle stated above. Note the full computational cost of this smoothing strategy is indeed $O(T)$, as our particle filter algorithm is designed in a way that ensures a constant cost across iterations, see §3.3. Also, the number of stored values at iteration t is proportional to the filtered probability of a change point occurrence at time t, which seems cost-effective in terms of storage. In contrast the particle smoothers of Kitagawa (1996) and Godsill et al. (2004) require to store the whole set of particles at every iteration, but these are general methods while our approach is specific to change point models.

5 Numerical illustration

5.1 The model

We consider a Gaussian GARCH change point model,

$$y_t \sim \mathcal{N}\left(0, \sigma_t^2\right),$$

where

$$\sigma_t^2 = m_t^{-1} + \alpha_t y_{t-1}^2 + \beta_t \sigma_{t-1}^2, \tag{5}$$

and $\theta_t = (m_t, \alpha_t, \beta_t)$ denotes the three-dimensional changing parameter, constrained to $m_t > 0$, $\alpha_t > 0$, $\beta_t > 0$, $\alpha_t + \beta_t < 1$, the last constraint ensuring stationarity (within each regime). The prior π_ξ for $\xi = (m, \alpha, \beta)$ is set to the product of Gamma(a, b) for m and Dirichlet$(1, 1, 1)$ for (α, β). One can prove that the posterior expectation and variance of m_t are infinite whatever t and whenever, respectively, $a \leq 0.5$, and $a \leq 1.5$; this is because, conditional on a change at t (an event with positive probability) the posterior of m_t turns into a GARCH posterior given one single observation, which have infinite moments for given values of a. One has therefore to select a slightly informative prior in such settings; say $a \in (1.5, 2.5]$ and $b = s^2(a-1)$, where s is the typical scale of the considered data,

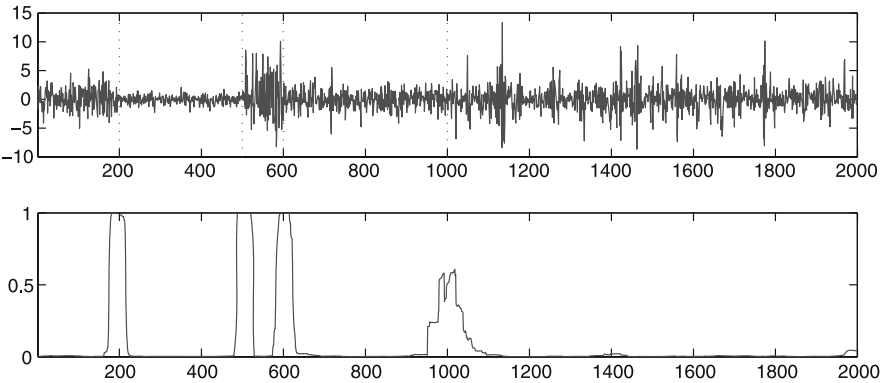

Fig. 1 From top to bottom: simulated GARCH data (*solid line*) and change points (*dotted lines*); estimated marginal probability of a change at time $t \pm 20$, conditional on complete data set

so that the prior mean of $1/m$ is s^2. Note such a prior becomes quite informative as a increases, as the prior variance of $1/m$ is $s^2/(a-2)$, for $a > 2$, and is $+\infty$ otherwise (we take $a = 2$ in our simulations). The prior π_δ is set to the uniform distribution on the set of integers between some values $\underline{\delta}$ and $\overline{\delta}$. The rationale for this prior is that this seems the simplest way to express prior information on the range of plausible durations. We shall see that this prior is very reasonable in our real data application, although other choices may be preferable in other settings. An alternative, popular choice for π_δ is the geometric distribution (with parameter ϵ being either fixed or random). This second choice seems however less appealing in applications where change occurs at a low frequency, as it requires to assign to ϵ either a very small fixed value, or a random distribution with a very narrow support. In doing so, one may still introduce a bias towards small durations (given the particular shape of a geometric distribution), and renders the analysis more sensitive to outliers.

Note (5) departs from the standard parameterisation of GARCH models, say (μ, α, β) with $\mu = m^{-1}$. This different parameterisation was initially motivated by the Langevin move strategy evoked in §3.4, as it ensures that $H_t^{(j)}$ in (4) is always positive definite. It turned out however that this was also beneficial for the random walk move strategy (as implemented in these simulations), possibly because it reduces the tails of the posterior density and therefore facilitates its exploration by a Gaussian random walk.

5.2 A simulated example

We simulated $T = 2{,}000$ data points from five successive regimes, of respective durations 300, 200, 100, 400 and 1,000. The successive parameter values are, respectively, (0.5, 3.33, 0.33, 1, 1) for m_t, (0.2, 0.2, 0.2, 0.3, 0.8) for α_t, (0.1, 0.1, 0.7, 0.2, 0.1) for β_t. Figure 1 plots the simulated data and indicates the change times by a vertical line. Hyperparameters were set to: $\underline{\delta} = 10$, $\overline{\delta} = 2{,}000$, $a = 2$, $b = 1$.

We executed our algorithm ten times, with $H = 50{,}000$ particles. Figure 2 reports the estimated filtered expectation of each component of θ_t as given by the first execution, and the standard deviation of these estimates over the ten runs, the latter quantity being of order 0.03 for the first component m_t, of order 0.01

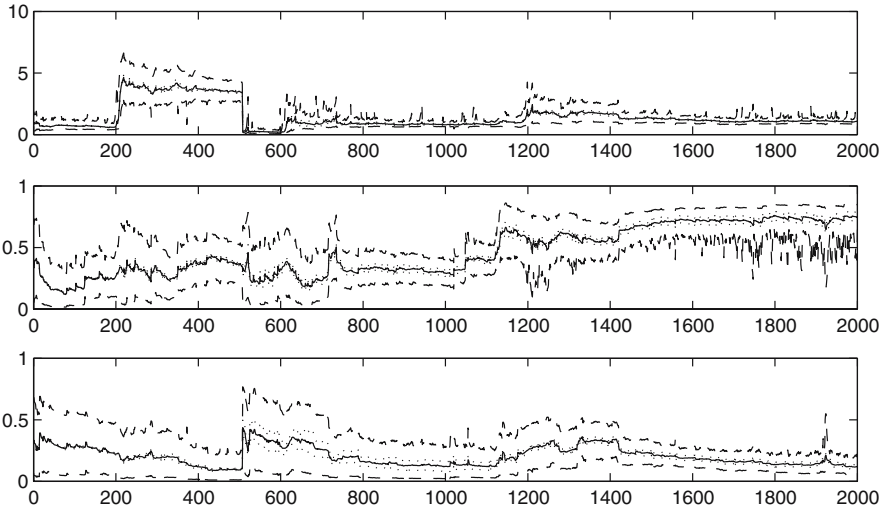

Fig. 2 From top to bottom, estimate of the filtered expectation of resp. m_t, α_t, β_t (*solid line*), same quantity $\pm$ twice its standard deviation over ten Monte Carlo exercises (*dotted lines*), estimated filtered 10% and 90%-quantiles (*dashed lines*)

for the other components α_t, β_t. This figure also provides the estimated 10 and 90%-quantiles of each marginal distribution of $\pi(\theta_t|y_{1:t})$.

From the output of the first run, we built 10,000 smoothing trajectories, out of which 79% featured five regimes, 18% six regimes, and 3% a different number. From the samples featuring five regimes, we constructed histograms of simulated values for the five regime parameters and the durations of the four first regimes, the last regime having not necessarily ended, see Fig. 3.

The constant C described in §3.3 was set to the number of particles H. This implies that the total computational cost of the move steps is roughly the same as the cost of H iterations of a random walk Hastings–Metropolis algorithm for a *single* GARCH model, without change point (and for the complete data set). The move steps accounted for 40% of the total computational load.

The tuning factor γ was set to 0.75, leading to an average acceptance rate of about 25%. Simulations with a different value of γ led to either similar results, for $0.5 \leq \gamma \leq 1$, or larger variability of the estimates, for values outside that interval. The 'positive discrimination' strategy described in §3.4 was applied, with the same constants as given in that section. The same simulations without positive discrimination gave similar results, except locally around changes where up to six times larger standard deviations of the filtering estimates were obtained. Each iteration of the algorithm took an average of 1.2 s on a 2.8 GHz desktop computer.

These results are more than satisfactory, given the challenging nature of the problem. GARCH models are notoriously difficult to estimate, as they produce rather flat likelihoods. Moreover some of the changes in the simulated data were deliberately small: for instance more than 150 observations after time $t = 1{,}000$ were necessary to detect the last change significantly, see Fig. 2. Despite this, the algorithm has been able to carry over a small number of particles that predict a change around 1,000 as long as necessary, then to make them evolve as a larger

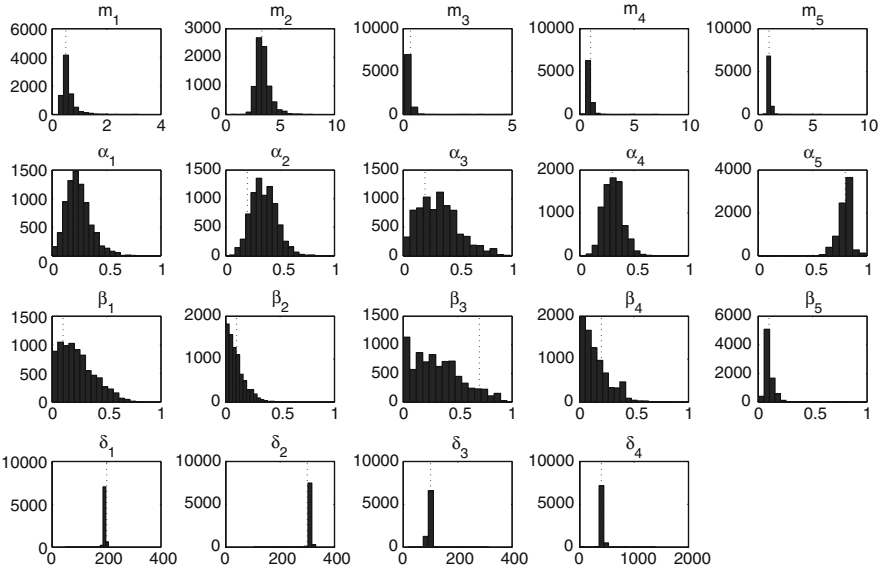

Fig. 3 Histograms of the simulated values of the regime parameters $\xi_k = (m_k, \alpha_k, \beta_k)$, $k = 1, \ldots, 5$, and of the regime durations δ_k, $k = 1, \ldots, 4$, from posterior distribution conditional on $y_{1:T}$ and on having five regimes; true values of the parameters are indicated by *dotted vertical lines*

and more diverse population as the estimated probability of a change around 1,000 increased.

5.3 A real data example

We applied the same analysis to the daily log-returns of the Standard and Poor 500 index between 1970.01.01 and 1979.12.31; for an economic discussion on the relevance of a GARCH change point model for these data, see Mikosch and Stărică (2003, 2004). Hyperparameters are set to: $\underline{\delta} = 10$, $\overline{\delta} = T = 2{,}526$, T being the sample size, $a = 2$, $b = 5 \times 10^{-5}$. Figure 4 plots these data, and the estimated probability of a change occurrence between $\max(t - \Delta, 1)$ and $\min(t + \Delta, T)$, with $\Delta = 20$. We selected particular time intervals around the four most important modes of the latter functions (modes are marked by dashed lines in top plot, intervals by dotted lines in bottom plot). The posterior probability of having exactly one change within these intervals are estimated as, respectively, 84, 58, 96, and 74%. The evidence in favour of having a change in the second interval is not very strong, and is not robust to prior specification, see below; thus, only the first, third and fourth change will be considered as significant from now on. The posterior probability of having k changes over these 10 years were estimated to be 0, 15, 27, 27.5, 18, 8.5% for $k = 2, \ldots, 7$, respectively, and smaller that 5% for all other values of k.

Even conditional on a fixed number k of changes, the posterior distribution is polymodal, as it assigns certain probability to non-comparable combinations

of change dates: for instance, a (slight) majority of simulations have their second change in the second interval mentioned above, but some have it in the third interval. For this reason, Fig. 5 gives histograms of only those simulations that feature exactly one change point within the first, the third and the fourth of the intervals represented in Fig. 4, and none outside (6% of the complete sample). One must bear in mind however that this figure represents only a part of the full complexity of the posterior distribution. Figure 5 also gives histograms corresponding to function $\tau_i = [(1 - \alpha_i - \beta_i)\mu_i]^{-1/2}$, the standard deviation of the stationary distribution corresponding to GARCH model with parameters $(\mu_i, \alpha_i, \beta_i)$.

Fit and predictive power of the model are reasonable: Figure 6 gives a QQ-plot of residuals estimated from the simulated paramater values with highest posterior density (within those featuring three changes). Figure 7 plots $\Phi^{-1}(\widehat{p}_t)$, where $\widehat{p}_t$ is an estimator of the probability $p(y_t > \widetilde{y}_t | y_{1:t-1})$, defined with respect to the predictive distribution of y_t, where $\widetilde{y}_t$ denotes the value actually observed at time t. Note the transformation $\Phi^{-1}(\cdot)$ is just a convenient way to represent more extreme values of $\widehat{p}_t$, as the marginal predictive distribution is not Gaussian. Note also that the marginal predictive density $p(y_t | y_{1:t-1})$ takes into account parameter incertainty; as such the fact that $\widehat{p}_t$ do not take too extreme values after change points indicates the prior is reasonable, and is not unduly non informative. Right plot in Figure 7 represents the three estimated quartiles of $p(e_t | y_{1:t})$, where $e_t = t + d_t - 1$ is the date where latest change has occured, at time t.

We conducted some prior sensitivity analysis: values of $\overline{\delta}$ corresponding to a duration between 5 and 20 years also lead to the same significant changes corresponding to first, third and four intervals in Fig. 4, while the change corresponding to the second interval drops significantly below 50% when $\delta > 15$ years. Basic economic insight and common sense suggest that values larger than 20 years for $\overline{\delta}$ are not appropriate. Similarly, for values smaller than 5 years, durations almost equal to $\overline{\delta}$ start to appear; a clear sign that values as small are not suitable either. Therefore, and despite its simplicity, an uniform prior for durations between changes seems convenient and reasonable in this particular application. A similar analysis for a varying within (1.5, 2.5] (while b is set to $s^2(a-1)$, see above for a justification of this particular interval) leads roughly to the same results, that is, the detection of a change in first, third and fourth intervals is prior robust, the detection of a change in second interval is not.

6 Concluding remarks

6.1 Extensions

We mention here some straightforward extensions for our algorithm. Sequential forecasting can be performed in the following way: after Step 1, draw $y_t^{(j)} \sim p(y_t | y_{1:t-1}, \theta_t^{(j,1)})$, for $j = 1, \ldots, H$, in order to obtain a weighted discrete representation of the marginal posterior distribution of y_t, conditional on $y_{1:t-1}$ and the event that there is not a change point at time t. Note it straightforward to adapt this to forecasting unconditional on the latter event, but that is arguably less useful and relevant in practice. The assumption of prior independence between the δ's and the ξ's may be relaxed: for instance, in Step 2 $\xi^{(j)}$ can be drawn from some prior

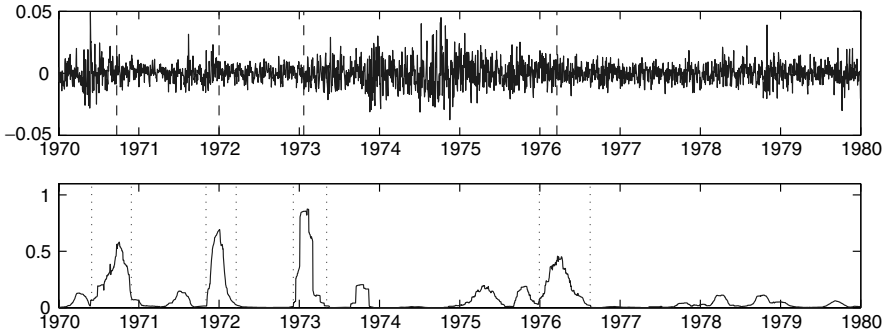

Fig. 4 From top to bottom, daily log-returns of the Standard and Poor 500 index between 1970.01.01 and 1979.12.31; estimated probability of a change between $\max(t-20,1)$ and $\min(t+20,T)$; *dashed lines* in top plot represent local maxima of the latter function; *dotted lines* represent interval in which exactly one change occurs with high probability

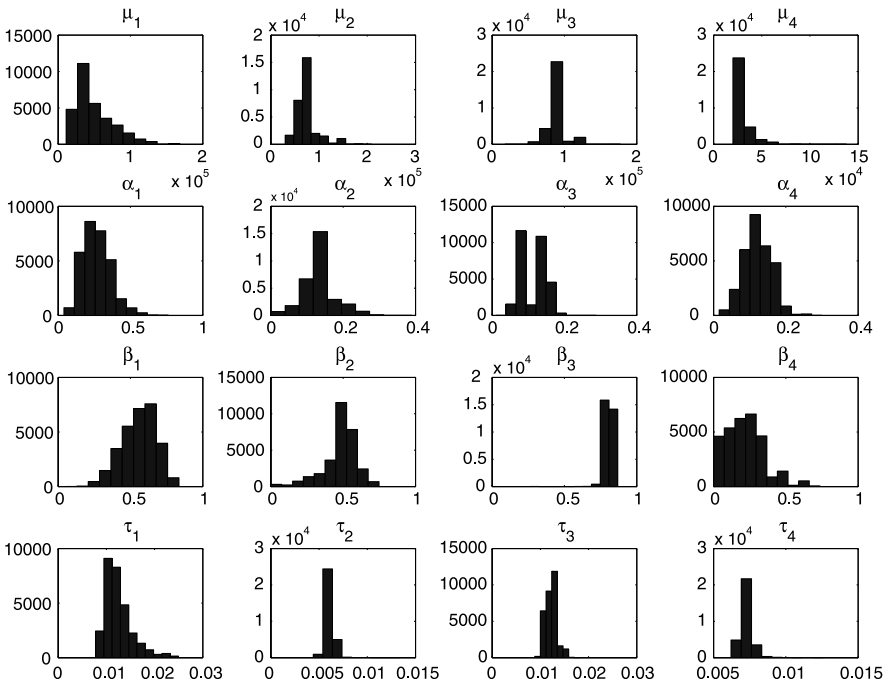

Fig. 5 Histograms of simulated values of the regime parameters $\xi_k = (m_k, \alpha_k, \beta_k), k = 1, \ldots, 4$, from posterior distribution conditional on $y_{1:T}$ and having exactly one change in first, third and fourth intervals of Fig. 4, for Standard and Poor example

conditional on the previous regime parameter $\theta_{t-1}^{(j)}$. One may penalise in this way successive regimes that are too similar, in the spirit of the discriminating factor in Chopin and Pelgrin (2004). Unfortunately, the extension to hierarchical priors seems more difficult.

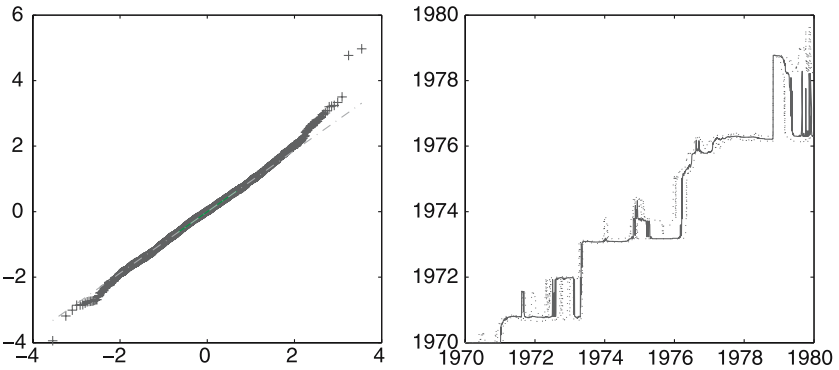

Fig. 6 From left to right: QQ-plot of the residuals estimated from simulation with highest posterior density (x-axis gives standard normal quantiles, y-axis gives residual quantiles); estimated quartiles of $p(d_t + t - 1|y_{1:t})$, e.g. date of last observed change at time t Standard and Poor example

6.2 Discussion

Let's comment on the computational efficiency of our algorithm. We have seen in previous section that its computational cost is comparable to that of a very reasonable number of iterations of a MCMC algorithm corresponding to the same model without change point; e.g. a single GARCH model in our example. This is mostly due to its sequential nature, which allows for breaking down the dimensionality of the problem: at a fixed time only one change needs to considered, or equivalently up to two competing models. Another appeal of particle filter algorithms is their unbiasedness: in our simulation experiments we could obtain reasonable estimates even with 5,000 particles, at the expense of course of larger Monte Carlo errors. This is particularly useful for experimentation and prior sensitivity purposes. In contrast, convergence is often an awkward issue with complex MCMC schemes, and, given the initial value of the chain, many iterations may be required before reaching the vicinity of the posterior mode. A third advantage of our particular approach is its ability to calibrate automatically its MCMC moves to an appropriate scale, while plain MCMC algorithms, such as random walk Hastings–Metropolis, require a manual tuning of the size the random step, which is sometimes cumbersome.

Beyond the computational aspects of this problem, we are convinced that change point modelling is a very promising way of dealing with non-stationarity. However it does not solve, and in some sense complicates, the issue of choosing an

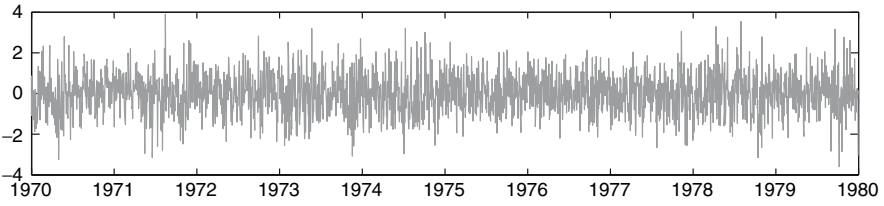

Fig. 7 Plot of $\Phi^{-1}(\widehat{p}_t)$, where $\widehat{p}_t$ is an estimate of $p(y_t > \widetilde{y}_t|y_{1:t-1})$, and $\widetilde{y}_t$ is the value of y_t actually observed, for Standard and Poor example

appropriate model within each period of time. To paraphrase George Box's famous statement: all models are wrong; some models fit longer.

Acknowledgements The author thanks Christophe Andrieu, Christian P. Robert, Peter J. Green, Antonietta Mira and two anonymous referees for insightful comments.

References

Barry, D., Hartigan, J. (1993). A Bayesian analysis for change point problems. *Journal of the American Statistical Association*, *88*, 309–319

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

Casella, G., Robert, C. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, *1*, 81–94.

Chen, R., Liu, J. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society Series B*, *62*, 493–508.

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

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

Chopin, N. (2004). Central Limit Theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Annals of Statistics*, *32*(6), 2385–2411.

Chopin, N., Pelgrin, F. (2004). Bayesian inference and state number determination for hidden Markov models: an application to the information content of the yield curve about inflation. *Journal of Econometrics*, *123*(2), 327–344.

de Jong, P., Shephard, N. (1995). The simulation smoother for time series models. *Biometrika*, *82*, 339–350.

Del Moral, P., Miclo, L. (2000). Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering. In J. Azéma, M. Emery, M. Ledoux, M. Yor, (Eds.) *Séminaire de Probabilités XXXIV*, (vol 1729 pp 1–145). Lecture Notes in Mathematics, Berlin Heidelberg New York: Springer.

Doucet, A., de Freitas, N., Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Berlin Heidelberg New York: Springer.

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

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

Gerlach, R., Carter, C., Kohn, R. (2000). Efficient Bayesian inference for dynamic mixture models. *Journal of the American Statistical Association*, *88*, 819–828.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society Series B*, *63*, 127–146.

Godsill, S., Doucet, A., West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, *99*, 156–168.

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

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, *82*, 711–732.

Kalman, R., Bucy, R. (1961). New results in linear filtering and prediction theory. *Transactions of the American Society of Mechanical Engineers*, *83*, 95–108.

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

Künsch, H. (2001). State space and hidden Markov models. In O. E. Barndorff-Nielsen, D. R. Cox, C. Klüppelberg (Eds.) *Complex stochastic systems*, (pp 109–173). London: Chapman and Hall.

Liu, J., Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, *93*, 1032–1044.

McCulloch, R., Tsay, R. (1993). Bayesian inference and prediction for mean and variance shifts in autoregressive time series. *Journal of the American Statistical Association*, *88*, 968–978.

Mikosch, T., Stărică, C. (2003). Long range dependence effects and ARCH modelling. In P. Doukhan, G. Oppenheim, M. Taqqu (Eds.) *Theory and applications of long range dependence*. Boston: Birkhauser.
Mikosch, T., Stărică, C. (2004). Non-stationarities in financial time series, the long range dependence and the IGARCH effects. *Review of Economics and Statistics*, *86*, 378–390.
Pitt, M., Shephard, N. (1999). Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, *94*, 590–599.
Robert, C. P., Casella, G. (2004). *Monte Carlo statistical methods, 2nd edn.* Berlin Heidelberg New York: Springer
Roberts, G., Gelman, A., Gilks, W. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied probability 7*, 110–120.
Stephens, D. A. (1994). Bayesian retrospective multiple-changepoint identification. *Applied Statistics*, *43*(1), 159–178.