BAYESIAN STATISTICS 9, pp. 317–360.
J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid,
D. Heckerman, A. F. M. Smith and M. West (Eds.)

Particle Learning for Sequential Bayesian Computation

Hedibert F. Lopes
The University of Chicago, USA
hlopes@chicagobooth.edu

Carlos M. Carvalho
The University of Chicago, USA
carlos.carvalho@chicagobooth.edu

Michael S. Johannes
Columbia University, USA
mj335@columbia.edu

Nicholas G. Polson
The University of Chicago, USA
ngp@chicagobooth.edu

Summary

Particle learning provides a simulation-based approach to sequential Bayesian computation. To sample from a posterior distribution of interest we use an essential state vector together with a predictive distribution and propagation rule to build a resampling-sampling framework. Predictive inference and sequential Bayes factors are a direct by-product. Our approach provides a simple yet powerful framework for the construction of sequential posterior sampling strategies for a variety of commonly used models.

Keywords and Phrases: Particle learning; Bayesian; Dynamic factor models; Essential state vector; Mixture models; Sequential inference; Conditional dynamic linear models; Nonparametric; Dirichlet.

Hedibert F. Lopes is Associate Professor of Econometrics and Statistics, The University of Chicago Booth School of Business. Carlos M. Carvalho is Assistant Professor of Econometrics and Statistics, University of Chicago Booth School of Business and Donald D. Harrington Fellow at The University of Texas, Austin. Michael S. Johannes is Roger F. Murray Associate Professor of Finance, Graduate School of Business, Columbia University. Nicholas G. Polson is Professor of Econometrics and Statistics, The University of Chicago Booth School of Business. We would like to thank Mike West, Raquel Prado and Peter Müller for insightful comments that greatly improved the article. We also thank Seung-Min Yae for research assistance with some of the examples. Part of this research was conducted while the first two authors were visiting the Statistical and Applied Mathematical Sciences Institute for the 2008-09 Program on Sequential Monte Carlo Methods. Carvalho would like to acknowledge the support of the Donald D. Harrington Fellowship Program and the IROM Department at The University of Texas at Austin.

1. THE PL FRAMEWORK

Sequential Bayesian computation requires the calculation of a set of posterior distributions $p(\theta \mid y^t)$, for $t = 1, \ldots, T$, where $y^t = (y_1, \ldots, y_t)$. The inability to directly compute the marginal $p(y^t) = \int p(y^t \mid \theta) p(\theta) d\theta$ implies that accessing the desired posterior distributions requires simulation schemes. This paper presents a sequential simulation strategy to calculate both $p(\theta \mid y^t)$ and $p(y^t)$ based on a *resample-sampling* framework called Particle Learning (PL). PL is a direct extension of the resample-sampling scheme introduced by Pitt and Shephard (1999) in the fixed-parameter, time series context.

Our new look at Bayes's theorem delivers a sequential, on-line inference strategy for effective posterior simulation strategies in a variety of commonly used models. These strategies are intuitive and easy to implement. In addition, when contrasted to MCMC methods PL delivers more for less as it provides

(i) posterior samples in a direct approximations of marginal likelihoods;

(ii) parallel environment, an important feature as more multi-processor computational power becomes available.

Central to PL is the creation of an *essential state vector* Z_t to be tracked sequentially. We assume that this vector is conditionally sufficient for the parameter of interest; so that $p(\theta \mid Z_t)$ is either available in closed-form or can easily be sampled from.

Given samples $\{Z_t^{(i)}, i = 1, \ldots, N\} \sim p(Z_t \mid y^t)$, or simply $\{Z_t^{(i)}\}$ by omitting N from the notation, then a simple mixture approximation to the set of posteriors (or moments thereof) is given by

$$p^N(\theta \mid y^t) = \frac{1}{N} \sum_{i=1}^{N} p(\theta \mid Z_t^{(i)}).$$

This follows from the Rao-Blackwellised identity,

$$p(\theta \mid y^t) = \mathbb{E}\left\{p(\theta \mid Z_t)\right\} = \int p(\theta \mid Z_t)\, p(Z_t \mid y^t)\, dZ_t.$$

If we require samples, we draw $\theta^{(i)} \sim p(\theta \mid Z_t^{(i)})$. See West (1992,1993) for an early approach to approximating posterior distributions via mixtures.

The task of sequential Bayesian computation is then equivalent to a filtering problem for the essential state vector, drawing $\{Z_t^{(i)}\} \sim p(Z_t \mid y^t)$ sequentially from the set of posteriors. To this end, PL exploits the following sequential decomposition of Bayes' rule

$$\begin{aligned} p(Z_{t+1} \mid y^{t+1}) &= \int p(Z_{t+1} \mid Z_t, y_{t+1})\, d\mathbb{P}(Z_t \mid y^{t+1}) \\ &\propto \int \underbrace{p(Z_{t+1} \mid Z_t, y_{t+1})}_{\text{propagate}} \underbrace{p(y_{t+1} \mid Z_t)}_{\text{resample}}\, d\mathbb{P}(Z_t \mid y^t). \end{aligned}$$

The distribution $d\mathbb{P}(Z_t \mid y^{t+1}) \propto p(y_{t+1} \mid Z_t) d\mathbb{P}(Z_t \mid y^t)$, where $\mathbb{P}(Z_t \mid y^t)$ denotes the distribution of the current state vector. In particle form this would be represented by $N^{-1} \sum_{i=1}^N \delta_{Z_t^{(i)}}$, where δ is the Dirac measure.

The intuition is as follows. Given $\mathbb{P}(Z_t \mid y^t)$ we find the smoothed distribution $\mathbb{P}(Z_t \mid y^{t+1})$ via resampling and then propagate forward using $p(Z_{t+1} \mid Z_t, y_{t+1})$ to find the new Z_{t+1}. Making an analogy to dynamic linear models this is exactly the Kalman filtering logic in reverse, first proposed by Pitt and Shephard (1999). From a sampling perspective, this leads to a very simple algorithm for updating particles $\{Z_t^{(i)}\}$ to $\{Z_{t+1}^{(i)}\}$ in 2 steps:

(i) *Resample:* with replacement from a multinomial with weights proportional to the predictive distribution $p(y_{t+1} \mid Z_t^{(i)})$ to obtain $\{Z_t^{\zeta(i)}\}$;

(ii) *Propagate:* with $Z_{t+1}^{(i)} \sim p(Z_{t+1} \mid Z_t^{\zeta(i)}, y_{t+1})$ to obtain $\{Z_{t+1}^{(i)}\}$.

The ingredients of particle learning are the essential state vector Z_t, a predictive probability rule $p(y_{t+1} \mid Z_t^{(i)})$ for resampling $\zeta(i)$ and a propagation rule to update particles $Z_t^{\zeta(i)}$ to $Z_{t+1}^{(i)}$. We summarize the algorithm as follows:

Particle Learning (PL)

Step 1. (Resample) Generate an index $\zeta \sim \text{Multinomial}(\omega, N)$ where

$$\omega^{(i)} = \frac{p(y_{t+1} \mid Z_t^{(i)})}{\sum_{i=1}^N p(y_{t+1} \mid Z_t^{(i)})};$$

Step 2. (Propagate)

$$Z_{t+1}^{(\zeta(i))} \sim p(Z_{t+1} \mid Z_t^{(\zeta(i))}, y_{t+1});$$

Step 3. (Learn)

$$p^N(\theta \mid y^{t+1}) = \frac{1}{N} \sum_{i=1}^N p(\theta \mid Z_{t+1}).$$

Example 1 (Constructing Z_t for the i.i.d. model). As a first illustration of the derivation of the essential state vector and the implementation of PL, consider the following simple i.i.d. model

$$\begin{aligned} y_t \mid \lambda_t &\sim N(\mu, \tau^2 \lambda_t) \\ \lambda_t &\sim IG(\nu/2, \nu/2) \end{aligned}$$

for $t = 1, \ldots, T$ and known ν and prior $\mu \mid \tau^2 \sim N(m_0, C_0 \tau^2)$ and $\tau^2 \sim IG(a_0, b_0)$.

Here the essential state vector is $Z_t = (\lambda_{t+1}, a_t, b_t, m_t, C_t)$ where (a_t, b_t) index the sufficient statistics for the updating of τ^2, while (m_t, C_t) index the sufficient statistics for the updating of μ. Set $m_0 = 0$ and $C_0 = 1$. The sequence of variables λ_{t+1} are

i.i.d. and so can be propagated directly from $p(\lambda_{t+1})$, whilst the conditional sufficient statistics (a_{t+1}, b_{t+1}) are deterministically calculated based on previous values (a_t, b_t) and parameters $(\mu_{t+1}, \lambda_{t+1})$. Here μ_{t+1} simply denotes draws for the parameter μ at time $t+1$. Given the particle set $\{(Z_0, \mu, \tau^2)^{(i)}\}$, PL cycles through the following steps:

Step 1. Resample $\{(\tilde{Z}_t, \tilde{\mu}, \tilde{\tau}^2)^{(i)}\}$ from $\{(Z_t, \mu, \tau^2)^{(i)}\}$ with weights

$$w_{t+1}^{(i)} \propto p(y_{t+1} \mid Z_n^{(i)}) = f_N(y_{t+1}; m_t^{(i)}, \tau^{2(i)}(C_t^{(i)} + \lambda_{t+1}^{(i)})), \qquad i = 1, \ldots, N;$$

Step 2. Propagate $a_{t+1}^{(i)} = \tilde{a}_t^{(i)} + 0.5$ and $b_{t+1}^{(i)} = \tilde{b}_t^{(i)} + 0.5 y_{t+1}^2/(1 + \tilde{\lambda}_{t+1}^{(i)})$, and sample $\tau^{2(i)}$ from $IG(a_{t+1}^{(i)}, b_{t+1}^{(i)})$, for $i = 1, \ldots, N$;

Step 3. Propagate $C_{t+1}^{(i)} = 1/(1/\tilde{C}_t^{(i)} + 1/\lambda_{t+1}^{(i)})$ and $(C_{t+1}^{(i)})^{-1} m_{t+1}^{(i)} = (\tilde{C}_t^{(i)})^{-1} \tilde{m}_t^{(i)} + y_{t+1}/\lambda_{t+1}^{(i)}$, and sample $\mu_{t+1}^{(i)}$ from $N(m_{t+1}^{(i)}, C_{t+1}^{(i)})$, for $i = 1, \ldots, N$;

Step 4. Sample $\lambda_{t+2}^{(i)}$ from $p(\lambda_{t+2})$ and let $Z_{t+1}^{(i)} = (\lambda_{t+2}, a_{t+1}, b_{t+1}, m_{t+1}, C_{t+1})^{(i)}$, for $i = 1, \ldots, N$.

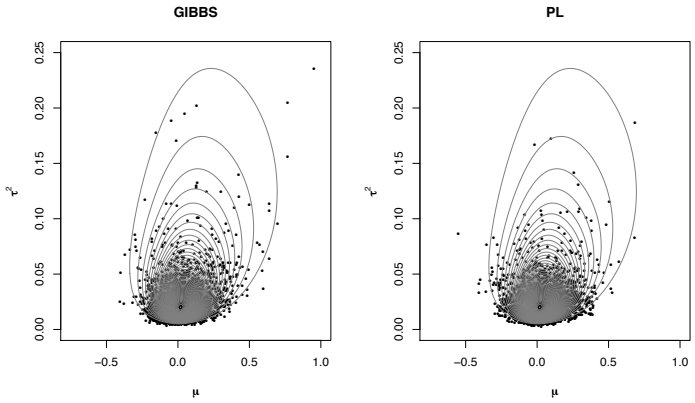

Figure 1: *i.i.d. model. Gibbs versus Particle Learning. Data* $y = (-15, -10, 0, 1, 2)$*, number of degrees of freedom* $\nu = 1$*, and hyperparameters* $a_0 = 5$*,* $b_0 = 0.05$*,* $m_0 = 0$ *and* $C_0 = 1$*. For the Gibbs sampler, the initial value for* τ^2 *is* $V(y) = 58.3$ *and 5000 draws after 10000 as burn-in. PL is based on 10000 particles. The contours represent the true posterior distribution.*

In step 2 $f_N(y; \mu, \sigma^2)$ denotes the density of $N(\mu, \sigma^2)$ evaluated at y. The posterior for μ and τ^2 could be approximated via a standard Gibbs sampler since the full conditionals are

$$\begin{aligned}
\mu \mid \lambda, \tau^2, y &\sim N(g_1(y, \lambda)/s(\lambda); \tau^2/s(\lambda)) \\
\tau^2 \mid \lambda, y &\sim IG(a_0 + T/2, b_0 + 0.5 g_2(y, \lambda)) \\
\lambda_t \mid \tau^2, y_t &\sim IG\left(\frac{\nu+1}{2}, \frac{\nu + (y_t - \mu)^2/\tau^2}{2}\right) \qquad t = 1, \ldots, T.
\end{aligned}$$

where $s(\lambda) = 1 + \sum_{t=1}^T \lambda_t^{-1}$, $g_1(y, \lambda) = \sum_{t=1}^T y_t/\lambda_t$, and $g_2(y, \lambda) = \sum_{t=1}^T y_t^2/(1 + \lambda_t)$. Figure 1 provides an illustration of both PL to the Gibbs sampler.

1.1. *Constructing the Essential State Vector*

At first sight, PL seems to be a rather simple paradigm. The real power, however, lies in the flexibility one has in defining the essential state vector. This may include: state variables, auxiliary variables, subset of parameters, sufficient statistics, model specification, among others. The dimensionality of Z_t can also be included in the particle set and increase with the sample size as, for example, in the non-parametric mixture of Dirichlet process discussed later.

In sum, the use of an essential state vector Z_t is an integral part of our approach, and its definition will become clear in the following sections. The propagation rule can involve either stochastic or deterministic updates and in many ways it is a modeling tool by itself. For example, in complicated settings (variable selection, treed models) the propagation rule $p(Z_{t+1} \mid Z_t, y_{t+1})$ suggests many different ways of searching the model space. It is our hope that with the dissemination of the ideas associated with PL there will be more cases where the use of Z_t leads to new modeling insights. The following represent examples of the form of Z_t in the models that will be addressed later in the chapter:

(i) *Mixture Regression Models:* Auxiliary state variable λ_t and conditional sufficient statistics s_t for parameter inference;

(ii) *State Space Models:* In conditionally Gaussian dynamic linear models Z_t tracks the usual Kalman filter state moments denoted by (m_t, C_t) and conditional sufficient statistics s_t for fixed parameters;

(iii) *Nonparametric Models:* Track an indicator of each mixture component k_t, the number n_t allocated to each component and the current number of unique components m_t. In a Dirichlet process mixture, for example, the particle vector can grow in time as there is a positive probability of adding a new mixture component with each new observation.

In the rest of the paper, we address each of these models and provide the necessary calculations to implement PL.

1.2. *Comparison with SIS and MCMC*

Particle filtering (Gordon, Salmond and Smith, 1993) and sequential importance sampling (Kong, Liu and Wong, 1994) have a number of features in common with PL. For example, one can view our update for the augmented vector Z_t as a fully-adapted version of Pitt and Shephard's (1999) the auxiliary particle filter (APF), with the additional step that the augmented variables can depend on functionals of the parameter. The additional parameter draw $\theta^{(i)} \sim p(\theta \mid Z_t^{(i)})$ is not present in the APF and is used in PL to replenish the diversity of the parameter particles.

Storvik (2002) proposed the use of sufficient statistics in state space models that are independent of parameters in a propagate-resampling algorithm. Chen and Liu (2000) work with a similar approach in the mixture Kalman filter context. PL differs in two important ways: *(i)* they only consider the problem of state filtering and *(ii)* they work on the propagate-resample framework. This is carefully discussed in Carvalho, Johannes, Lopes and Polson (2010). Again, our view of augmented variables Z_t is more general than Storvik's approach.

Another related class of algorithms are Rao-Blackwellised particle filters, which are typically propagate-resample algorithms where Z_{t+1} denotes missing data and

x_{t+1} a state and a pure filtering problem. Additionally they attempt to approximate the joint distribution $p\left(Z^t \mid y^t\right)$. This target increases in dimensionality as new data becomes available leading to unbalanced weights. In our framework, $p\left(Z^t \mid y^t\right)$ is not of interest as the filtered, lower dimensional $p\left(Z_t \mid y^t\right)$ is sufficient for inference at time t. Notice that, based on their work, one has to consider the question of "when to resample?" as an alternative to re-balance the approximation weights. In contrast, our approach requires re-sampling at every step as the pre-selection of particles in light of new observations is fundamental in avoiding a decay in the particle approximation for θ.

Another avenue of research uses MCMC steps inside a sequential Monte Carlo algorithm as in the resample-move algorithm of Gilks and Berzuini (2001). This is not required in our strategy as we are using a fully-adapted approach. Finally, see Lopes and Tsay (2011) for a recent review of particle filter methods with an emphasis on empirically contrasting propagate-resample and resample-propagate filters in financial econometrics problems.

1.3. *Smoothing*

At time T, PL provides the filtered distribution of the last essential state vector Z_T, namely $p(Z_T \mid y^T)$. If the smoothed distribution of any element k of Z, i.e, $p(k^T \mid y^T)$ is required, it can be obtained at the end of the filtering process. To compute the marginal smoothing distribution, we need the distribution

$$p(k^T \mid y^T) = \int p(k^T \mid Z_T, y^T) p(Z_T \mid y^T) dZ_T.$$

In the case where k_t is discrete and conditionally independent across time given Z_T this can further simplified as

$$\int p(k^T \mid Z_T, y) p(Z_T \mid y^T) dZ_T = \int \prod_{t=1}^{T} p(k_t \mid Z_T, y_t) p(Z_T \mid y^T) dZ_T$$

so that samples from $p(k^T \mid y^T)$ can be obtained by sampling (for each particle Z_T) each k_t independently from the discrete filtered mixture with probability proportional to

$$p(k_t = j \mid Z_T, y_t) \propto p(y_t \mid k_t = j, Z_T) p(k_t = j \mid Z_T).$$

This is the case, for example, in the mixture models consider later where k could represent the allocation of each observation to a mixture component.

When k_t has a Markovian evolution, as in state space models, the smoothing distribution can be expressed as

$$\int p(k^T \mid Z_T, y^T) p(Z_T \mid y^T) dZ_T = \prod_{t=1}^{T} p(k_t \mid k_{t+1}, Z_T) p(Z_T \mid y^T).$$

By noting that

$$p(k_t \mid k_{t+1}, Z_T) \propto p(k_{t+1} \mid k_t, Z_t) p(k_t \mid Z_t),$$

sequential backwards sampling schemes can be constructed using the transition equation of k_t as resampling weights.

This discussion is a generalization of the algorithm presented in Carvalho, Johannes, Lopes and Polson (2010) for state space models which is originally proposed as an extension of Godsill, Doucet and West (2004). It is important to point out that traditional SMC algorithms attempt to approximate $p(k^t \mid y^t)$ as part of the filtering process, *i.e.*, , attempting to sequentially approximate a posterior that is growing in dimension with t – this leads, as expected and extensively reported, to unbalanced weights. PL focus on the simpler, more stable problem of filtering $p(k_t \mid y^t)$ and observes that, in most models, smoothing can effectively be performed in the end.

1.4. *Marginal Likelihoods*

PL also provides estimates of the predictive distribution $p(y_{t+1} \mid y^t)$ and marginal likelihood $p(y^t)$ for model assessment and Bayes factors. Following our resampling-sampling approach, an on-line estimate of the full marginal likelihood can be developed by sequentially approximating $p(y_{t+1} \mid y^t)$. Specifically, given the current particle draws, we have

$$p^N(y_{t+1} \mid y^t) = \sum_{i=1}^{N} p(y_{t+1} \mid Z_t^{(i)}) \quad \text{and} \quad p^N(y^T) = \prod_{t=1}^{T} p^N(y_t \mid y^{t-1}).$$

Therefore we simplify the problem of calculating $p\left(y^T\right)$ by estimating a sequence of lower dimensional integrals. This also provides access to sequential Bayes factors necessary in many sequential decision problems.

1.5. *Choice of Priors*

At its simplest level the algorithm only requires samples $\theta^{(i)}$ from the prior $p(\theta)$. Hence the method is not directly applicable to improper priors. However, a natural class of priors are mixture priors on the form $p(\theta) = \int p(\theta \mid Z_0)p(Z_0)dZ_0$. The conditional $p(\theta \mid Z_0)$ is chosen to be naturally conjugate to the likelihood. If Z_0 is fixed, then we start all particles out with the same Z_0 value. More commonly, we will start with a sample $Z_0^{(i)} \sim p(Z_0)$ and let the algorithm resample these draws with the marginal likelihood $p(y_1 \mid Z_0^{(i)})$. This approach will lead to efficiency gains over blindly sampling from the prior. This method also allows us to implement non-conjugate priors together with vague "uninformative" priors such as Cauchy priors via a scale mixtures of normals.

1.6. *Monte Carlo Error*

Due to the sequential Monte Carlo nature of the algorithm, error bounds of the form $C_T/\sqrt{N}$ are available where N is the number of particles used. The constant C_T is model, prior and data dependent and in general its magnitude accumulates over T, see, for example, Brockwell, Del Moral and Doucet (2010). Clearly, these propagate errors will be greater for diffuse priors and for large signal-to-noise ratios as with many Monte Carlo approaches. To assess Monte Carlo standard errors we propose the convergence diagnostic of Carpenter, Clifford and Fearnhead (1999). By running the algorithm M independent times (based on N particles) one can calculate the Monte Carlo estimates of the mean and variance for the functional of interest. Then by performing an analysis of variance between replicates, the Monte Carlo error or effective sample size can be assessed. One might also wish to perform this measure over different data trajectories as some data realizations might be harder to estimate than others.

2. APPLICATIONS

2.1. *Mixture Regression Models*

In order to illustrate the efficiency gains available with our approach consider the most common class of applications: mixture or latent variable models

$$p(y \mid \theta) = \int p(y \mid \theta, \lambda) p(\lambda \mid \theta) d\lambda,$$

where $\lambda^T = (\lambda_1, \ldots, \lambda_T)$ is a data augmentation variable. For this model, with a conditionally conjugate prior, we can find a conditional sufficient statistic, s_t, for parameter learning. Therefore, we define our sufficient state vector as $Z_t = (\lambda_t, s_t)$. Under these assumptions, we can write

$$p\left(\theta \mid \lambda^{t+1}, y^{t+1}\right) = p\left(\theta \mid s_{t+1}\right) \quad \text{with} \quad s_{t+1} = \mathcal{S}\left(s_t, \lambda_{t+1}, y_{t+1}\right)$$

where $\mathcal{S}(\cdot)$ is a deterministic recursion relating the previous s_t to the next, conditionally on λ_{t+1} and y_{t+1}. Now, the propagation step becomes

$$\begin{aligned} \lambda_{t+1} &\sim p(\lambda_{t+1} \mid \lambda_n, \theta, y_{t+1}) \\ s_{t+1} &= \mathcal{S}\left(s_t, \lambda_{t+1}, y_{t+1}\right). \end{aligned}$$

More complicated mixture models appear in Section 2.3.

Example 2 (Bayesian lasso). Consider a sequential version of Bayesian lasso (Carlin and Polson, 1991, Hans, 2009) for a simple problem of signal detection. The model takes the form $y_t = \theta_t + \varepsilon_t$ and $\theta_t = \tau\sqrt{\lambda_t}\varepsilon_t^{\theta}$, where $\varepsilon_t \sim N(0,1)$, $\varepsilon_t^{\theta} \sim N(0,1)$, $\lambda_t \sim Exp(2)$ and $\tau^2 \sim IG(a_0, b_0)$. This leads to independent double exponential marginal priors for each θ_t with $p(\theta_t) = (2\tau)^{-1}\exp\left(-\left|\theta_t\right|/\tau\right)$. The natural set of latent variables is given by the augmentation variable λ_{t+1} and conditional sufficient statistics leading to $Z_{t+1} = (\lambda_{t+1}, a_t, b_t)$. The sequence of variables λ_{t+1} are i.i.d. and so can be propagated directly with $p(\lambda_{t+1})$, whilst the conditional sufficient statistics (a_{t+1}, b_{t+1}) are deterministically determined based on parameters $(\theta_{t+1}, \lambda_{t+1})$ and previous values (a_t, b_t).

Given the particle set $\{(Z_0, \tau)^{(i)}\}$, the resample-propagate algorithm cycles through the following steps:

i) Resample particles with weights $w_{t+1}^{(i)} \propto f_N(y_{t+1}; 0, 1 + \tau^{2(i)}\lambda_{t+1}^{(i)})$;

ii) Propagate $\theta_{t+1}^{(i)} \sim N(m_t^{(i)}, C_t^{(i)})$, $m_t^{(i)} = C_t^{(i)}\tilde{\tau}^{2(i)}\tilde{\lambda}_{t+1}^{(i)}y_{t+1}$ and $C_t^{-1} = 1 + \tilde{\tau}^{-2(i)}\tilde{\lambda}_{t+1}^{-1(i)}$;

iii) Update sufficient statistics $a_{t+1}^{(i)} = \tilde{a}_t^{(i)} + 1/2$ and $b_{t+1} = \tilde{b}_t^{(i)} + \theta_{t+1}^{2(i)}/(2\tilde{\lambda}_{t+1}^{(i)})$;

iv) Draw $\tau^{2(i)} \sim IG(a_{t+1}, b_{t+1})$ and $\lambda_{t+2}^{(i)} \sim Exp(2)$;

v) Let $Z_{t+1}^{(i)} = (\lambda_{t+1}^{(i)}, a_t^{(i)}, b_t^{(i)})$ and update $(Z_{t+1}, \tau)^{(i)}$.

We use our marginal likelihood (or Bayes factor) to compare lasso with a standard normal prior. Under the normal prior we assume that $\tau^2 \sim IG(a_1, b_1)$ and match the variances of the parameter θ_t. As the lasso is a model for sparsity we would expect the evidence for it to increase when we observe $y_t = 0$. We can sequentially estimate $p(y_{t+1} \mid y^t, \text{lasso})$ via $p(y_{t+1} \mid y^t, \text{lasso}) = N^{-1}\sum_{i=1}^{N} p(y_{t+1} \mid (\lambda_n, \tau)^{(i)})$ with a predictive

density $p(y_{t+1} \mid \lambda_t, \tau) \sim N(0, \tau^2\lambda_t + 1)$. This leads to a sequential Bayes factor $B_{t+1} = p(y^{t+1} \mid \text{lasso})/p(y^{t+1} \mid \text{normal})$.

Data was simulated based on $\theta = (0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1)$ and priors $\tau^2 \sim IG(2, 1)$ for the double exponential case and $\tau^2 \sim IG(2, 3)$ for the normal case, reflecting the ratio of variances between those two distributions. Results are summarized by computing sequential Bayes factor (figure not shown). As expected the evidence in favor of the lasso is increased when we observe $y = 0$ and for the normal model when we observe a signal $y = 1$.

PL can easily be extended to a lasso regression setting. Suppose that we have

$$y_{t+1} = X_t'\beta + \sigma\sqrt{\lambda_{t+1}}\epsilon_{t+1}$$

and $\theta = (\beta, \sigma^2)$ and conditionally conjugate prior is assumed, *i.e.*, $p(\beta \mid \sigma^2) \sim N(b_0, \sigma^2 B_0^{-1})$ and $p(\sigma^2) \sim IG(\nu_0/2, d_0/2)$. We track $Z_t = (s_t, \lambda_{t+1})$ where $s_t = (b_t, B_t, d_t)$ are conditional sufficient statistics for the parameters. The recursive definitions are

$$\begin{aligned} B_{t+1} &= B_t + \lambda_{t+1}^{-1} X_t' X_t \\ B_{t+1}b_{t+1} &= B_t b_t + \lambda_{t+1}^{-1} X_t' y_{t+1}, \text{and} \\ d_{t+1} &= d_t + b_t' B_t b_t + \lambda_{t+1}^{-1} X_{t+1}' y_{t+1} - b_{t+1}' B_{t+1} b_{t+1}. \end{aligned}$$

The conditional posterior $p(\theta \mid Z_{t+1})$ is then available for sampling and our approach applies.

We use this example to compare the accuracy in estimating the posterior distribution of the regularization penalty $p(\tau \mid y)$. We use the generic resample-move batch importance sampling developed by Gilks and Berzuini (2001) and Chopin (2002). The data is cut into batches parameterized by block-lengths (n, p). In the generic resample move algorithm, we first initialize by drawing from the prior $\pi(\theta, \tau)$ with $\theta = (\theta_1, \ldots, \theta_{15})$. The particles are then re-sampled with the likelihood from the first batch of observations $(y_1, \ldots, y_p)$. Then the algorithm proceeds sequentially.

There is no need to use the λ_t augmentation variables as this algorithm does not exploit this conditioning information. Then an MCMC kernel is used to move particles. Here, we use a simple random walk MCMC step. This can clearly be tuned to provide better performance although this detracts from the "black-box" nature of this approach. Chopin (2002) provides recommendations for the choice of kernel. Figure 2 provides the comparison with two separate runs of the algorithm both with $N = 10,000$ particles for $(n, p) = (3, 5)$ or $(n, p) = (15, 1)$. The performance is similar for the case $p = 1$. Our efficiency gains come from the extra conditioning information available in Z_t.

2.2. *Conditional Dynamic Linear Models*

We now explicitly derive our PL algorithm in a class of conditional dynamic linear models (CDLMs) which are an extension of the models considered in West and Harrison (1997). This follows from Carvalho, Johannes, Lopes and Polson (2010) and consists of a vast class of models embedding many of the commonly used dynamic models. MCMC via forward-filtering backwards-sampling (FFBS) or mixture Kalman filtering (MKF) (Chen and Liu, 2000) are the current methods of use for the estimation of these models. As an approach for filtering, PL has a number of advantages. First, our algorithm is more efficient as it is a perfectly-adapted filter (Pitt and Shephard, 1999). Second, we extend MKF by including learning about fixed parameters and smoothing for states.

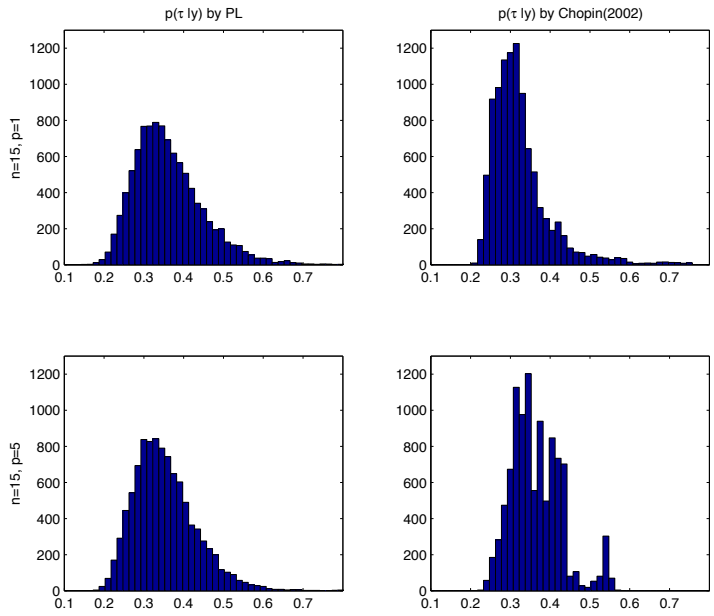

Figure 2: *Bayesian Lasso. Comparison to Chopin's (2002) batch sampling scheme.*

The CDLM defined by the observation and evolution equations takes the form of a linear system conditional on an auxiliary state λ_{t+1}

$$(y_{t+1} \mid x_{t+1}, \lambda_{t+1}, \theta) \sim N(F_{\lambda_{t+1}} x_{t+1}, V_{\lambda_{t+1}})$$
$$(x_{t+1} \mid x_t, \lambda_{t+1}, \theta) \sim N(G_{\lambda_{t+1}} x_t, W_{\lambda_{t+1}})$$

with θ containing the unknown elements of the quadruple $\{F_\lambda, G_\lambda, V_\lambda, W_\lambda\}$. The marginal distribution of observation error and state shock distribution are any combination of normal, scale mixture of normals, or discrete mixture of normals depending on the specification of the distribution on the auxiliary state variable $p(\lambda_{t+1} \mid \theta)$, so that,

$$p(y_{t+1} \mid x_{t+1}, \theta) = \int f_N(y_{t+1}; F_{\lambda_{t+1}} x_{t+1}, V_{\lambda_{t+1}}) p(\lambda_{t+1} \mid \theta) d\lambda_{t+1}.$$

Extensions to hidden Markov specifications where λ_{t+1} evolves according to the transition $p(\lambda_{t+1} \mid \lambda_t, \theta)$ are straightforward and are discussed below in the dynamic factor model with time varying loadings example.

In CDLMs the state filtering and parameter learning problem is equivalent to a filtering problem for the joint distribution of their respective sufficient statistics. This is a direct result of the factorization of the full joint $p(x_{t+1}, \theta, \lambda_{t+1}, s_{t+1}, s^x_{t+1} \mid y^{t+1})$ as a sequence of conditional distributions

$$p(\theta \mid s_{t+1}) p(x_{t+1} \mid s^x_{t+1}, \lambda_{t+1}) p(\lambda_{t+1}, s_{t+1}, s^x_{t+1} \mid y^{t+1}),$$

where s_t and s_t^x are the conditional sufficient statistics for parameters and states, respectively. Here, s_t^x and s_t satisfy deterministic updating rules

$$\begin{aligned} s_{t+1}^x &= \mathcal{K}\left(s_t^x, \theta, \lambda_{t+1}, y_{t+1}\right) \\ s_{t+1} &= \mathcal{S}\left(s_t, x_{t+1}, \lambda_{t+1}, y_{t+1}\right). \end{aligned}$$

More specifically, define $s_t^x = (m_t, C_t)$ as Kalman filter's first and second moments at time t. Conditional on θ, we then have $(x_{t+1} \mid s_{t+1}^x, \lambda_{t+1}, \theta,) \sim N(a_{t+1}, R_{t+1})$, where $a_{t+1} = G_{\lambda_{t+1}} m_t$ and $R_{t+1} = G_{\lambda_{t+1}} C_t G'_{\lambda_{t+1}} + W_{\lambda_{t+1}}$. Updating state sufficient statistics (m_{t+1}, C_{t+1}) is achieved by

$$m_{t+1} = G_{\lambda_{t+1}} m_t + A_{t+1}\left(y_{t+1} - e_t\right), \qquad C_{t+1}^{-1} = R_{t+1}^{-1} + F'_{\lambda_{t+1}} F_{\lambda_{t+1}} V_{\lambda_{t+1}}^{-1},$$

with Kalman gain matrix $A_{t+1} = R_{t+1} F_{\lambda_{t+1}} Q_{t+1}^{-1}$, predictive mean

$$e_t = F_{\lambda_{t+1}} G_{\lambda_{t+1}} m_t,$$

and predictive variance

$$Q_{t+1} = F_{\lambda_{t+1}} R_{t+1} F_{\lambda_{t+1}} + V_{\lambda_{t+1}}.$$

We are now ready to define the PL scheme for the CDLMs. First, assume that the auxiliary state variable is discrete with $\lambda_{t+1} \sim p(\lambda_{t+1} \mid \lambda_t, \theta)$. We start, at time t, with a particle approximation for the joint posterior of $(x_t, \lambda_t, s_t, s_t^x, \theta \mid y^t)$. Then we propagate to $t+1$ by first re-sampling the current particles with weights proportional to the predictive $p(y_{t+1} \mid (\theta, s_t^x))$. This provides a particle approximation to $p(x_t, \theta, \lambda_t, s_t, s_t^x \mid y^{t+1})$, the smoothing distribution. New states λ_{t+1} and x_{t+1} are then propagated through the conditional posterior distributions $p(\lambda_{t+1} \mid \lambda_t, \theta, y_{t+1})$ and $p(x_{t+1} \mid \lambda_{t+1}, x_t, \theta, y_{t+1})$, respectively. Finally, the conditional sufficient statistics are updated and new samples for θ are obtained from $p(\theta \mid s_{t+1})$. Notice that in the CDLMs all the above densities are available for evaluation or sampling. For instance, the predictive is computed via

$$p(y_{t+1} \mid (\lambda_t, s_t^x, \theta)^{(i)}) = \sum_{\lambda_{t+1}} p(y_{t+1} \mid \lambda_{t+1}, (s_t^x, \theta)^{(i)}) p(\lambda_{t+1} \mid \lambda_t, \theta)$$

where the inner predictive distribution is given by

$$p\left(y_{t+1} \mid \lambda_{t+1}, s_t^x, \theta\right) = \int p\left(y_{t+1} \mid x_{t+1}, \lambda_{t+1}, \theta\right) p(x_{t+1} \mid s_t^x, \theta) dx_{t+1}.$$

In the general case where the auxiliary state variable λ_t is continuous it might not be possible to integrate out λ_{t+1} from the predictive in step 1. We extend the above scheme by adding to the current particle set a propagated particle $\lambda_{t+1} \sim p(\lambda_{t+1} \mid (\lambda_t, \theta)^{(i)})$. Both algorithms can be combined with the backwards propagation scheme of Carvalho, Johannes, Lopes and Polson (2010) to provide a full draw from the marginal posterior distribution for all the states given the data, the smoothing distribution $p(x^T \mid y^T)$.

The next two examples detail the steps of PL for a dynamic factor models with time-varying loadings and for a dynamic logit models.

Example 3 (Dynamic factor model with time-varying loadings). Consider data $y_t = (y_{t1}, y_{t2})'$, $t = 1, \ldots, T$, following a dynamic factor model with time-varying loadings driven by a discrete latent state λ_t with possible values in $\{1, 2\}$. Specifically, we have

$$(y_{t+1} \,|\, x_{t+1}, \lambda_{t+1}, \theta) \sim N(\beta_{t+1} x_{t+1}, \sigma^2 I_2)$$
$$(x_{t+1} \,|\, x_t, \lambda_{t+1}, \theta) \sim N(x_t, \sigma_x^2)$$

with time-varying loadings $\beta_{t+1} = (1, \beta_{\lambda_{t+1}})'$ and initial state distribution $x_0 \sim N(m_0, C_0)$. The jumps in the factor loadings are driven by a Markov switching process $(\lambda_{t+1} \,|\, \lambda_t, \theta)$, whose transition matrix Π has diagonal elements $Pr(\lambda_{t+1} = 1 \,|\, \lambda_t = 1, \theta) = p$ and $Pr(\lambda_{t+1} = 2 \,|\, \lambda_t = 2, \theta) = q$. The parameters are, therefore, $\theta = (\beta_1, \beta_2, \sigma^2, \sigma_x^2, p, q)'$. See Carvalho and Lopes (2007) for related Markov switching models.

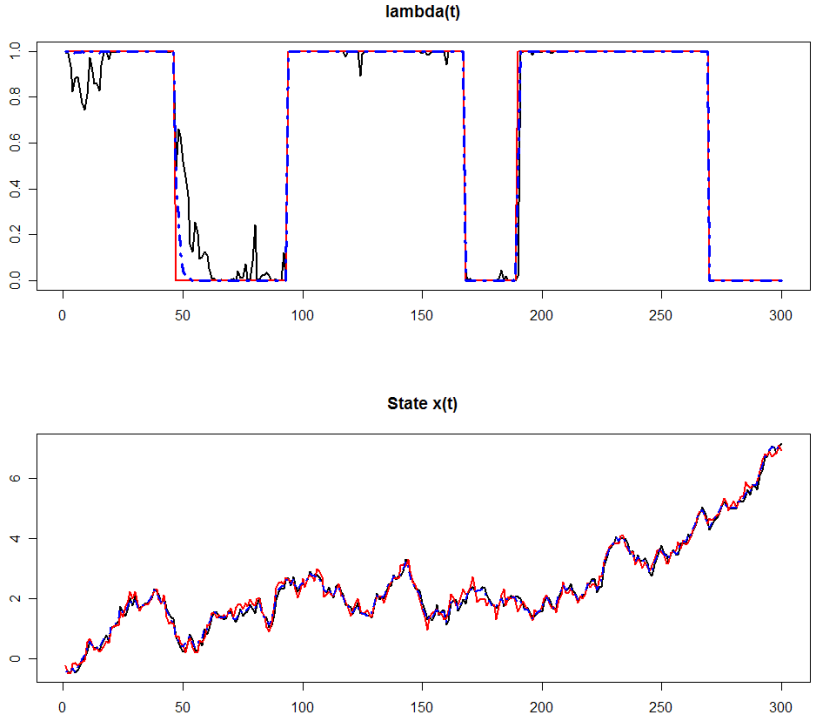

Figure 3: *Dynamic factor model - state filtering. Top panel: True value of* λ_t, $Pr(\lambda_t = 1 \,|\, y^t)$ *and* $Pr(\lambda_t = 1 \,|\, y^T)$ *Bottom panel: True value of* x_t, $E(x_t \,|\, y^t)$ *and* $E(x_t \,|\, y^T)$.

We are able to marginalize over both (x_{t+1}, λ_{t+1}) by using state sufficient statistics $s_t^x = (m_t, C_t)$ as particles. From the Kalman filter recursions we know that $(x_t \,|\, \lambda^t, \theta, y^t) \sim N(m_t, C_t)$. The mapping for state sufficient statistics $s_{t+1}^x = \mathcal{K}(s_t^x, \lambda_{t+1}, \theta, y_{t+1})$ is given by the one-step Kalman update equations. The prior distributions are conditionally conjugate where $(\beta_i \,|\, \sigma^2) \sim N(b_{i0}, \sigma^2 B_{i0})$ for $i = 1, 2$, $\sigma^2 \sim IG(\nu_{00}/2, d_{00}/2)$ and $\sigma_x^2 \sim IG(\nu_{10}/2, d_{10}/2)$.

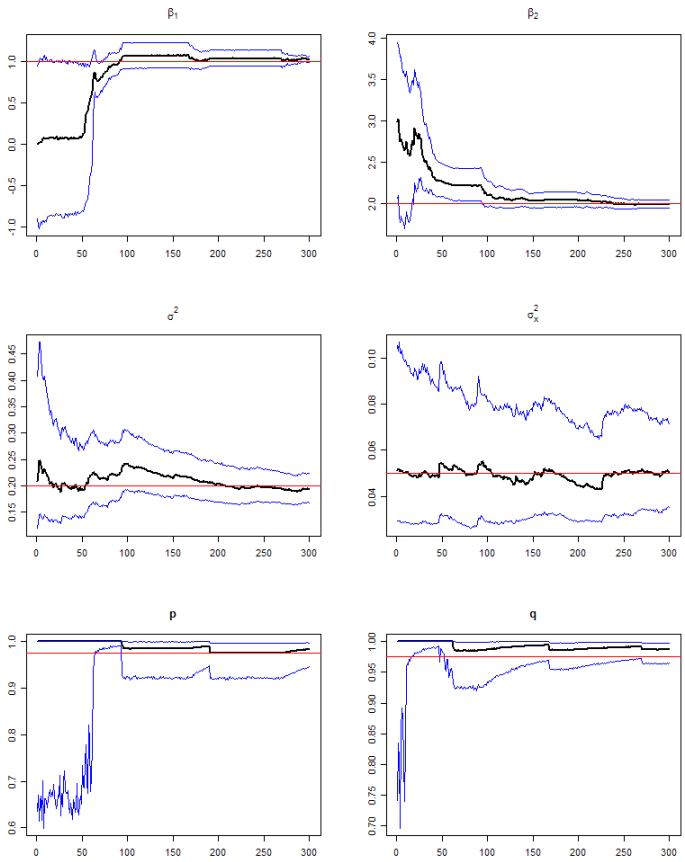

Figure 4: *Dynamic factor model–parameter learning. Sequential posterior median and posterior 95% credibility intervals for model parameters* β_1, β_2, σ^2, τ^2, p *and* q.

For the transition probabilities, we assume that $p \sim Beta(p_1, p_2)$ and $q \sim Beta(q_1, q_2)$. Assume that at time t, we have particles $\{(x_t, \theta, \lambda_t, s_t^x, s_t)^{(i)}\}$ approximating $p(x_t, \theta, \lambda_t, s_t^x, s_t \mid y^t)$. The PL algorithm can be described through the following steps:

(i) *Re-sampling:* Draw an index $k^i \sim \text{Multinomial}(w_t^{(1)}, \ldots, w_t^{(N)})$ with weights $w_t^{(i)} \propto p(y_{t+1} \mid (s_t^x, \lambda_t, \theta)^{(k^i)})$, where $p(y_{t+1} \mid s_t^x, \lambda_t, \theta)$ equals

$$\sum_{\lambda_{t+1}=1}^{2} f_N(y_{t+1}; \beta_{t+1} m_t, (C_t + \tau^2)\beta_{t+1}\beta_{t+1}' + \sigma^2 I_2) p(\lambda_{t+1} \mid \lambda_t, \theta);$$

(ii) *Propagating state* λ*:* Draw $\lambda_{t+1}^{(i)}$ from $p(\lambda_{t+1} \mid (s_t^x, \lambda_t, \theta)^{(k^i)}, y_{t+1})$, such that

$$p(\lambda_{t+1} \mid s_t^x, \lambda_t, \theta, y_{t+1}) \propto f_N(y_{t+1}; \beta_{t+1} m_t, (C_t + \tau^2)\beta_{t+1}\beta_{t+1}' + \sigma^2 I_2) p(\lambda_{t+1} \mid \lambda_t, \theta);$$

(iii) *Propagating state x:* Draw $x_{t+1}^{(i)}$ from $p(x_{t+1} \mid \lambda_{t+1}^{(i)}, (s_t^x, \theta)^{(k^i)}, y_{t+1})$;

(iv) *Propagating states sufficient statistics,* s_{t+1}^x*:* The Kalman filter recursions yield $m_{t+1} = m_t + A_{t+1}(y_{t+1} - \beta_{t+1} m_t)$ and $C_{t+1} = C_t + \tau^2 - A_{t+1} Q_{t+1}^{-1} A_{t+1}'$, where $Q_{t+1} = (C_t + \tau^2)\beta_{t+1}\beta_{t+1} + \sigma^2 I_2$ and $A_{t+1} = (C_t + \tau^2) Q_{t+1}^{-1} \beta_{t+1}$.

(v) *Propagating parameter sufficient statistics,* s_{t+1}*:* The posterior $p(\theta \mid s_t)$ is decomposed into $(\beta_i \mid \sigma^2, s_{t+1}) \sim N(b_{i,t+1}, \sigma^2 B_{i,t+1})$, $i = 1, 2$, $(\sigma^2 \mid s_{t+1}) \sim IG(\nu_{0,t+1}/2, d_{0,t+1}/2)$, $(\tau^2 \mid s_{t+1}) \sim IG(\nu_{1,t+1}/2, d_{1,t+1}/2)$, $(p \mid s_{t+1}) \sim Beta(p_{1,t+1}, p_{2,t+1})$, $(q \mid s_{t+1}) \sim Beta(q_{1,t+1}, q_{2,t+1})$ with $B_{i,t+1}^{-1} = B_{it}^{-1} + x_{t+1}^2 \mathbb{I}_{\lambda_{t+1}=i}$, $b_{i,t+1} = B_{i,t+1}(B_{it}^{-1} b_{it} + x_t y_{t2} \mathbb{I}_{\lambda_{t+1}=i})$ and $\nu_{i,t+1} = \nu_{i,t} + 1$, for $i = 1, 2$, $d_{1,t+1} = d_{1t} + (x_{t+1} - x_t)^2$, $p_{1,t+1} = p_{1t} + \mathbb{I}_{\lambda_t=1,\lambda_{t+1}=1}$, $p_{2,t+1} = p_{2t} + \mathbb{I}_{\lambda_t=1,\lambda_{t+1}=2}$, $q_{1,t+1} = q_{1t} + \mathbb{I}_{\lambda_t=2,\lambda_{t+1}=2}$, $q_{2,t+1} = q_{2t} + \mathbb{I}_{\lambda_t=2,\lambda_{t+1}=1}$, $d_{0,t+1} = d_{0t} + \sum_{j=1}^{2} [(y_{t+1,2} - b_{j,t+1} x_{t+1}) y_{t+1,2} + b_{j,t+1} B_{j0}^{-1} (y_{t+1,1} - x_{t+1})^2] \mathbb{I}_{\lambda_{t+1}=j}$,

Figure 3 and 4 illustrate the performance of the PL algorithm. The first panel of Figure 3 displays the true underlying λ process along with filtered and smoothed estimates whereas the second panel presents the same information for the common factor. Figure 4 provides the sequential parameter learning plots.

Example 4 (Dynamic logit models). Extensions of PL to non-Gaussian, non-linear state space models appear in Carvalho, Lopes and Polson (2010) and Carvalho, Johannes, Lopes and Polson (2010). We illustrate some these ideas in the context of a dynamic multinomial logit model with the following structure

$$P(y_{t+1} = 1 \mid \beta_{t+1}) = (1 + \exp\{-\beta_{t+1} x_{t+1}\})^{-1}$$
$$\beta_{t+1} = \phi \beta_t + \sigma_x \epsilon_{t+1}^{\beta}$$

where $\beta_0 \sim N(m_0, C_0)$ and $\theta = (\phi, \sigma_x^2)$. For simplicity assume that x_t is scalar. It is common practice in limited dependent variable models to introduce a latent continuous variable z_{t+1} to link y_{t+1} and x_t (see Scott, 2004, Kohn, 1997, and Frühwirth-Schnatter and Frühwirth, 2007). More precisely, the previous model, conditionally on z_{t+1}, where $y_{t+1} = \mathbb{I}(z_{t+1} \geq 0)$, can be rewritten as

$$\begin{aligned} z_{t+1} &= \beta_{t+1} x_{t+1} + \epsilon_{t+1}^z \\ \beta_{t+1} &= \phi \beta_t + \epsilon_{t+1}^{\beta}, \end{aligned}$$

where $\epsilon_{t+1}^{\beta} \sim N(0, \sigma_x^2)$, ϵ_{t+1}^z is an extreme value distribution of type 1, *i.e.*,

$$\epsilon_{t+1}^z \sim -\log \mathcal{E}(1),$$

where $\mathcal{E}(1)$ denotes an exponential with mean one.

Conditional normality can be achieved by rewriting the extreme value distribution as a mixture of normals. Frühwirth-Schnatter and Frühwirth (2007) suggest a 10-component mixture of normals with weight, mean and variance for component j given by w_j, μ_j and s_j^2, for $j = 1, \ldots, 10$. Hence conditional on the latent vector (z_{t+1}, λ_{t+1}), the previous representation leads to the following Gaussian dynamic linear model:

$$\begin{aligned} z_{t+1} &= \beta_{t+1} x_{t+1} + \epsilon_t \\ \beta_{t+1} &= \phi \beta_t + \epsilon_{t+1}^{\beta}, \end{aligned}$$

where $\epsilon_{t+1} \sim N(\mu_{\lambda_{t+1}}, s_{\lambda_{t+1}})$. Given λ_{t+1}, we have conditional state sufficient statistics (for β_t) and the Kalman filter recursions still hold as $s^{\beta}_{t+1} = \mathcal{K}(s^{\beta}_t, z_{t+1}, \lambda_{t+1}, \theta, y_{t+1})$. Similarly, for the parameter sufficient statistics s_t, which now involve λ_{t+1}. Moreover, as λ_{t+1} is discrete, it is straightforward to see that

$$Pr(y_{t+1} = 1 \,|\, s^{\beta}_t, \theta, \lambda_{t+1}) = 1 - \Phi(-\phi m_t x_{t+1}((\phi^2 C_t + \sigma^2_x)x^2_{t+1} + s^2_{\lambda_{t+1}})^{-1/2})$$

leading to the predictive

$$Pr(y_{t+1} = 1 \,|\, s^{\beta}_t, \theta) = \sum_{j=1}^{10} w_j Pr(y_{t+1} = 1 \,|\, s^{\beta}_t, \theta, \lambda_{t+1} = j),$$

which plays an important role in the resample step. The propagation step requires one to be able to sample λ_{t+1} from $p(\lambda_{t+1} \,|\, s^{\beta}_t, \theta, y_{t+1})$, z_{t+1} from $p(z_{t+1} \,|\, s^{\beta}_t, \theta, \lambda_{t+1}, y_{t+1})$ and β_{t+1} from $p(\beta_{t+1} \,|\, s^{\beta}_t, \theta, \lambda_{t+1}, z_{t+1}, y_{t+1})$. The final step of PL is the deterministic updating for conditional sufficient statistics.

2.3. *Nonparametric Mixture Models*

We now develop PL for discrete nonparametric mixture models and Bayesian nonparametric density estimation. Details appear in Carvalho, Lopes, Polson and Taddy (2010). Our essential state vector now depends on the (random) number of unique mixture components. The posterior information can be summarized by *(i)* the number of observations allocated to each unique component, $n_t = (n_{t,1}, \ldots, n_{t,m_t})$, *(ii)* the the conditional sufficient statistics, $s_t = (s_{t,1}, \ldots, s_{t,m_t})$, for the set of m_t distinct components in θ^t, $\theta^\star_t = \{\theta^\star_1, \ldots, \theta^\star_{m_t}\}$, and *(iii)* $k^t = (k_1, \ldots, k_t)$, the associated latent allocation such that $\theta_t = \theta^\star_{k_t}$. Therefore, the state vector to be tracked by PL can then be defined as $Z_t = (k_t, m_t, s_t, n_t)$. PL will not directly provide the full joint posterior distribution of the allocation vector k^t. If this is required either a particle smoothing or an MCMC step is incorporated in the algorithm.

For infinite mixture models, PL proceeds through the two familiar steps: Resample: $(s_t, n_t, m_t) \propto p(y_{t+1} \,|\, s_t, n_t, m_t)$ and Propagate: $k_{t+1} \sim p(k_{t+1} \,|\, s_t, n_t, m_t, y_{t+1})$. The filtered posterior for (s_T, n_T, m_T) can be used for inference via the posterior predictive density $\mathrm{p}(y \,|\, s_T, n_T, m_T)$, which is a Rao-Blackwellized version of $\mathbb{E}[f(y; G) \mid y^T]$ for many nonparametric priors (including the Dirichlet Process mixture, DP). Alternatively, since $p(G \,|\, y^T) = \int p(G \mid s_T, n_T, m_T)\; dp(s_T, n_T, m_T \,|\, y^T)$, the filtered posterior provides a basis for inference about the full random mixing distribution.

The DP characterizes a prior over probability distributions and is most intuitively represented through its constructive definition (Perman, Pitman and Yor, 1992): a random distribution G generated from $\mathrm{DP}(\alpha, G_0(\psi))$ is almost surely of the form

$$dG(\cdot) = \sum_{l=1}^{\infty} p_l\, \delta_{\vartheta_l}(\cdot) \quad \text{with} \quad \vartheta_l \overset{iid}{\sim} G_0(\vartheta_l; \psi), \quad p_l = \left(1 - \sum_{j=1}^{l-1} p_j\right) v_l,$$

and $v_l \overset{iid}{\sim} \mathrm{beta}(1, \alpha)$, for $l = 1, 2$, where $G_0(\vartheta; \psi)$ is the centering distribution function, parametrized by ψ, and the sequences $\{\vartheta_l, l = 1, 2, \ldots\}$ and $\{v_k : l = 1, 2, \ldots\}$ are independent. The discreteness of DP realizations is explicit in this definition.

The DP mixture model is then $f(y_r; G) = \int \mathrm{k}(y_r; \theta) dG(\theta)$ for $r = 1, \ldots, t$, where $G \sim \mathrm{DP}(\alpha, G_0)$. Alternatively, in terms of latent variables, the hierarchical model is that for $r = 1, \ldots, t$, $y_r \sim k(y_r; \theta_r)$, $\theta_r \sim G$ and $G \sim \mathrm{DP}(\alpha, G_0)$.

Two properties of the DP are particularly important for sequential inference. First, the DP is a conditionally conjugate prior, *i.e.*, given θ^t (or, equivalently, $\theta_t^\star$ and n_t), the posterior distribution for G is characterized as a $\mathrm{DP}(\alpha + t, G_0^t)$ where,

$$dG_0^t(\theta; \theta_t^\star, n_t) = \frac{\alpha}{\alpha + t} dG_0(\theta) + \sum_{j=1}^{m_t} \frac{n_{t,j}}{\alpha + t} \delta_{[\theta = \theta_j^\star]}.$$

Second, this Pólya urn density dG_0^t is also $\mathbb{E}[\, dG \mid \theta^t \,] = \int dG(\theta) dp(G \mid \theta_t^\star, n_t)$, and provides a finite predictive probability function for our mixture model:

$$p(y_{t+1} \mid \theta^t) = \int k(y_{t+1}; \theta) dG_0^t(\theta).$$

A Rao-Blackwellized version of the standard Pólya urn mixture serves as a density estimator:

$$p\left(\mathbb{E}[f(y; G)] \mid y^t\right) = \int p\left(\mathbb{E}[f(y; G)] \mid s_t, n_t, m_t\right) dp(s_t, n_t, m_t \mid y^t),$$

where $p\left(\mathbb{E}[f(y; G)] \mid s_t, n_t, m_t\right) = \int p(y \mid \theta_t^\star, n_t) dp(\theta_t^\star \mid s_t, n_t, m_t)$. If either α or ψ are assigned hyperpriors, we include this in Z_t and sample off-line for each particle conditional on $(n_t, s_t, m_t)^{(i)}$ at each iteration. This is of particular importance in the understanding of the generality of PL.

PL for DP mixture models

Step 1. (Resample) Generate an index $\zeta(i) \sim \mathrm{Multinomial}(\omega, N)$ where

$$\omega^{(i)} \propto p(y_{t+1} \mid (s_t, n_t, m_t)^{(i)})$$

Step 2. (Propagate)

Step 2.1. $k_{t+1} \sim p(k_{t+1} \mid (s_t, n_t, m_t)^{\zeta(i)}, y_{t+1})$;

Step 2.2. $s_{t+1} = \mathcal{S}(s_t, k_{t+1}, y_{t+1})$;

Step 2.3. n_{t+1}

$n_{t+1,j} = n_{t,j}$, for $j \neq k_{t+1}$,
$n_{t+1,k_t} = n_{t,k_t} + 1$ and $m_{t+1} = m_t$, if $k_{t+1} \leq m_t$,
$n_{t,m_{t+1}} = 1$, $m_{t+1} = m_t + 1$ and , if $k_{t+1} > m_t$;

Step 3. (Estimation)

$$p(\mathbb{E}[f(y; G)] \mid y^t) = \frac{1}{N} \sum_{i=1}^{N} p(y \mid (s_t, n_t, m_t)^{(i)})$$

Example 5 (The DP mixture of multivariate normals). In the particular case of the d-dimensional DP multivariate normal mixture (DP-MVN) model, we have

$$f(y_t; G) = \int \mathrm{N}(y_t \mid \mu_t, \Sigma_t) dG(\mu_t, \Sigma_t), \quad \text{and} \quad G \sim DP(\alpha, G_0(\mu, \Sigma)),$$

with conjugate centering distribution $G_0 = N(\mu; \lambda, \Sigma/\kappa)\,\mathrm{W}(\Sigma^{-1}; \nu, \Omega)$, where $\mathrm{W}(\Sigma^{-1}; \nu, \Omega)$ denotes a Wishart distribution such that $\mathbb{E}[\Sigma^{-1}] = \nu\Omega^{-1}$ and $\mathbb{E}[\Sigma] = (\nu - (d+1)/2)^{-1}\Omega$. Conditional sufficient statistics for each unique mixture component $s_{t,j}$ are

$$\bar{y}_{t,j} = \sum_{r:k_r=j} y_r/n_{t,j} \quad \text{and} \quad S_{t,j} = \sum_{r:k_r=j} y_r y_r' - n_{t,j}\bar{y}_{t,j}\bar{y}_{t,j}'.$$

The initial sufficient statistics are $n_1 = 1$ and $s_1 = \{y_1, 0\}$, such that the algorithm is populated with N identical particles. Conditional on existing particles $\{(n_t, s_t)^{(i)}\}$, uncertainty is updated through the familiar resample/propagate approach. The resampling step is performed by an application of the predictive probability function

$$p(y_{t+1} \mid s_t, n_t, m_t + 1) = \frac{\alpha}{\alpha + t}\mathrm{St}(y_{t+1}; a_0, B_0, c_0) + \sum_{j=1}^{m_t} \frac{n_{t,j}}{\alpha + t}\mathrm{St}(y_{t+1}; a_{t,j}, B_{t,j}, c_{t,j}),$$

with hyperparameters $a_0 = \lambda$, $B_0 = \frac{2(\kappa+1)}{\kappa c_0}\Omega$, $c_0 = 2\nu - d + 1$,

$$\begin{aligned} a_{t,j} &= \frac{\kappa\lambda + n_{t,j}\bar{y}_{t,j}}{\kappa + n_{t,j}}, & B_{t,j} &= \frac{2(\kappa + n_{t,j} + 1)}{(\kappa + n_{t,j})c_{t,j}}(\Omega + 0.5D_{t,j}), \\ c_{t,j} &= 2\nu + n_{t,j} - d + 1, \quad \text{and} & D_{t,j} &= S_{t,j} + \frac{\kappa n_{t,j}}{(\kappa + n_{t,j})}(\lambda - \bar{y}_{t,j})(\lambda - \bar{y}_{t,j})'. \end{aligned}$$

In the propagation step, we then sample the component state k_{t+1} such that,

$$\begin{aligned} p(k_{t+1} = j) &\propto \frac{n_{t,j}}{\alpha + t}\mathrm{St}(y_{t+1};\ a_{t,j}, B_{t,j}, c_{t,j}) \\ p(k_{t+1} = m_t + 1) &\propto \frac{\alpha}{\alpha + t}\mathrm{St}(y_{t+1};\ a_0, B_0, c_0), \end{aligned}$$

for $j = 1, \ldots, m_t$. If $k_{t+1} = m_t + 1$, the new sufficient statistics are defined by $m_{t+1} = m_t + 1$ and $s_{t+1,m_{t+1}} = [y_{t+1}, 0]$. If $k_{t+1} = j$, $n_{t+1,j} = n_{t,j} + 1$ and we update $s_{t+1,j}$ such that $\bar{y}_{t+1} = (n_{t,j}\bar{y}_{t,j} + y_{t+1})/n_{t+1,j}$ and $S_{t+1,j} = S_{t,j} + y_{t+1}y_{t+1}' + n_{t,j}\bar{y}_{t,j}\bar{y}_{t,j}' - n_{t+1,j}\bar{y}_{t+1,j}\bar{y}_{t+1,j}'$. The remaining sufficient statistics are the same as time t.

We can also assign hyperpriors to the parameters of G_0. In this case, a parameter learning step for each particle is added to the algorithm. Assuming a $\mathrm{W}(\gamma_\Omega, \Psi_\Omega^{-1})$ prior for Ω and a $\mathrm{N}(\gamma_\lambda, \Psi_\lambda)$ prior for λ, the sample at time t is augmented with draws for the auxiliary variables $\{\mu_j^\star, \Sigma_j^\star\}$, for $j = 1, \ldots, m_t$, from their posterior full conditionals,

$$p(\mu_j^\star, \Sigma_j^\star \mid s_t, n_t) \equiv N\left(\mu_j^\star;\ a_{t,j}, \frac{1}{\kappa + n_{t,j}}\Sigma_j^\star\right) W(\Sigma_j^{\star -1};\ \nu + n_{t,j}, \Omega + D_{t,j}).$$

The parameter updates are then

$$\lambda \sim N\left(R(\gamma_\lambda\Psi_\lambda^{-1} + \kappa\sum_{j=1}^{m_t}\Sigma_j^{\star -1}\mu_j^\star), R\right) \quad \text{and} \quad \Omega \sim \mathrm{W}(\gamma_\Omega + m_t\nu, R^{-1}),$$

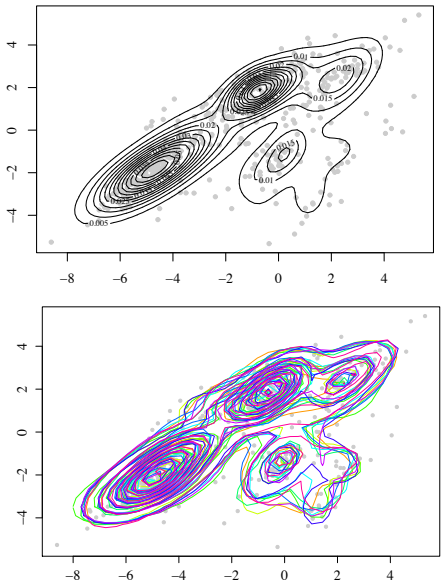

Figure 5: *DP mixture of multivariate normals. Data and density estimates for PL fit with 1000 particles (top) and each of ten PL fits with 500 particles (bottom), to a random ordering of the 1000 observations of bivariate data.*

where $R^{-1} = \sum_{j=1}^{m_t} \Sigma_j^{\star -1} + \Psi_\Omega^{-1}$. Similarly, if α is assigned the usual gamma hyperprior, it can be updated for each particle using the auxiliary variable method from Escobar and West (1995).

To illustrate the PL algorithm, a dataset was simulated with dimension $d = 2$ and sample size $T = 1000$. The bivariate vector of y_t was generated from a $\mathrm{N}(\mu_t, \mathrm{AR}(0.9))$ density, where $\mu_t \sim G_\mu$ and AR(0.9) denotes the correlation matrix implied by an autoregressive process of lag one and correlation 0.9. The mean distribution, G_μ, is the realization of a $\mathrm{DP}(4, N(0, 4\mathrm{I}))$ process. Thus the simulated data is clustered around a set of distinct means, and highly correlated within each cluster. The parameters are fixed at $\alpha = 2$, $\lambda = 0$, $\kappa = 0.25$, $\nu = 4$, and $\Omega = (\nu - 1.5)\mathrm{I}$, was fit to this data. Figure 5 shows the data and bivariate density estimates, which are the mean Rao-Blackwellized posterior predictive $\mathrm{p}(y \mid s_T, n_T, m_T)$; hence, the posterior expectation for $f(y; G)$. Marginal estimates are just the appropriate marginal density derived from mixture of Student's t distributions.

3. OTHER APPLICATIONS

Successful implementations of PL (and hybrid versions of PL) have appeared over the last couple of years. Taddy, Gramacy and Polson (2010) show that PL is the best alternative to perform online posterior filtering of tree-states in dynamic regression tree models, while Gramacy and Polson (2010) use PL for online updating of Gaussian process models for regression and classification. Shi and Dunson (2009) adopt a PL-flavored scheme for stochastic variable selection and model search in linear regression and probit models, while Mukherjee and West (2009) focus on model comparison for applications in cellular dynamics in systems biology.

With a more time series flavor, Rios and Lopes (2010), for example, propose a hybrid LW-Storvik filter for the Markov switching stochastic volatility model that outperforms Carvalho and Lopes (2007) filter. Lund and Lopes (2010) sequentially estimate a regime switching macro-finance model for the postwar US term-structure of interest rates, while Prado and Lopes (2010) adapt PL to study state-space autoregressive models with structured priors. Chen, Petralia and Lopes (2010) propose a hybrid PL-LW sequential MC algorithm that fully estimates non-linear, non-normal dynamic to stochastic general equilibrium models, with a particular application in a neoclassical growth model. Additionally, Dukić, Lopes and Polson (2010) use PL to track flu epidemics using Google trends data, while Lopes and Polson (2010) use PL to estimate volatility and examine volatility dynamics for financial time series, such as the S&P500 and the NDX100 indices, during the early part of the credit crisis.

4. FINAL THOUGHTS

4.1. *Historical Note*

Since the seminal paper by Gordon, Salmond and Smith (1993), and subsequently Kong, Liu and Wong (1994), Liu and Chen (1998) and Doucet, Godsill and Andrieu (2000), to name but a few, the sequential Monte Carlo literature is growing continuously. The first generation of SMC methods is well summarized in the compendium edited by Doucet, de Freitas and Gordon (2001) where several strategies for improving existing particle filters are discussed as well as about a dozen applications in various areas (see also Ristic, Arulampalam and Gordon, 2004, and the 2002 special issue of IEEE Transactions on Signal Processing on sequential Monte Carlo methods).

The vast majority of the literature defining the first generation focuses on sample-resample schemes, but the resample-sample particle filter introduced by Pitt and Shephard (1999) is the key initiator of the second stage of development in the SMC literature. APF with parameter learning was introduced by Liu and West (2001) and builds on earlier work by West (1992, 1993) who is the first published adaptive importance sampling scheme using mixtures (via kernel shrinkage) in sequential models. Our PL approach is a direct extension of Pitt and Shephard's (1999) APF. Carvalho, Johannes, Lopes and Polson (2010) show that APF and PL, both resample-sample schemes, outperform the standard sample-resample filters.

The second wave in the SMC literature occurred over the last decade, with recent advances in SMC that focus on, amongst other things, *i)* parameter learning, *ii)* similarities and differences between propagate-resample and resample-propagate filters; *iii)* computational viable particle smoothers and *iv)* the merge of SMC and MCMC tools towards more efficient sequential schemes. See Cappé, Godsill and Moulines (2007) and Doucet and Johansen (2008) for thorough reviews. See also, Prado and West (2010, chapter 6) and Lopes and Tsay (2011).

For example, important contributions to parameter learning were brought up, either for online or batch sampling, by Liu and West (2001), as mentioned above, Pitt (2002), Storvik (2002), Fearnhead (2002), Polson, Stroud and Müller (2008), Doucet and Tadić (2003), Poyiadjis, Doucet and Singh (2005) and Olsson, Cappé, Douc and Moulines (2006), to name by a few, while SIS and APF similarities are the focus of Doucet and Johansen (2008) and Douc, Moulines and Olsson (2009).

4.2. *PL and the Future*

Particle Learning provides a simulation-based approach to sequential Bayesian inference. It combines the features of data augmentation that is prevalent in MCMC with the resample-propagate auxiliary particle filter of Pitt and Shephard (1999). In many ways there is a parallel between the proliferation of data augmentation in Gibbs sampling and its potential role in expanding the PL framework. This combination of factors provides new insights on sequential learning for static parameters. In the case of trees, for example, using the essential state vector Z_t is itself a modeling tool suggesting many different ways of searching model space and specifying prior distributions in complicated spaces.

This leads to a fruitful direction for future modeling. There is a many open areas for future implementation of the framework: (i) Nonlinear and nonnormal panels (econometrics); (ii) REML (econometrics); (iii) Structural equations model; (iv) Dynamic factor models; (v) Multivariate extensions (challenging); (vi) Space-time models (ensemble Kalman filters).

Finally, we emphasize that there are differences in the way that Monte Carlo errors accumulate in MCMC and PL and this is clearly another fruitful area for future research both from a theoretical and empirical perspective. As with MCMC methods the usual word of caution of relying heavily on asymptotic central limit theorem results carried over to the PL framework.

REFERENCES

Cappé, O., Godsill, S. and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Trans. Signal Process.* **95**, 899-924.

Carpenter, J., Clifford, P. and Fearnhead, P. (1999). An improved particle filter for non-linear problems. *IEE Proc. Radar, Sonar and Navigation* **146**, 2–7.

Carlin, B. P. and Polson, N. G. (1991). Inference for nonconjugate Bayesian models using the Gibbs sampler. *Can. J. Statist.* **19**, 399–405.

Carvalho, C. M., Johannes, M., Lopes, H. F. and Polson, N. G. (2010). Particle learning and smoothing. *Statist. Science* **25**, 88–106.

Carvalho, C. M. and Lopes, H. F. (2007). Simulation-based sequential analysis of Markov switching stochastic volatility models. *Comput. Statist. Data Anal.* **51**, 4526–4542.

Carvalho, C. M., Lopes, H. F., Polson (2010). Particle learning for generalized dynamic conditionally linear models. *Tech. Rep.*, University of Chicago, USA.

Carvalho, C. M., Lopes, H. F., Polson, N. G. and Taddy, M. (2010). Particle learning for general mixtures. *Bayesian Analysis* **5**, 709–740.

Chen, H., Petralia, F. and Lopes, H. F. (2010). Sequential Monte Carlo estimation of DSGE models. *Tech. Rep.*, University of Chicago, USA.

Chen, R. and Liu, J. S. (2000). Mixture Kalman filters. *J. Roy. Statist. Soc. B* **62**, 493–508.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika* **89**, 539–551.

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Ann. Statist.* **32**, 2385–2411.

Douc, R., E. Moulines, and J. Olsson (2009). Optimality of the auxiliary particle filter. *Probab. Math. Statist.* **29**, 1–28.

Doucet, A., De Freitas, N. and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Berlin: Springer.

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

Doucet, A. and Johansen, A. (2008). A note on auxiliary particle filters. *Statist. Probab. Lett.* **78**, 1498–1504.

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

Dukić, V., Lopes, H. F. and Polson, N. G. (2010). Tracking flu epidemics using Google trends and particle learning. *Tech. Rep.*, University of Chicago, USA.

Escobar, M. and West, M. (1995). Bayesian density estimation and inference using mixtures. *J. Amer. Statist. Assoc.* **90**, 577–588.

Fearnhead, P. (2002). Markov chain Monte Carlo, sufficient statistics and particle filter. *J. Comp. Graphical Statist.* **11**, 848-862.

Gamerman, D. and Lopes, H. F. (2006). *Chain Monte Carlo: Stochastic Simulation for Bayesian Inference* (2nd ed.) Boca Raton: Chapman and Hall/CRC.

Gilks, W. and Berzuini, C. (2001). Following a moving target: Monte Carlo inference for dynamic Bayesian models. *J. Roy. Statist. Soc. B* **63**, 127–146.

Godsill, S. J., A. Doucet, and M. West (2004). Monte Carlo smoothing for nonlinear time series. *J. Amer. Statist. Assoc.* **99**, 156–168.

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

Gramacy, R. and Polson, N. G. (2010). Particle learning of Gaussian process models for sequential design and optimization. *Tech. Rep.*, University of Chicago, USA.

Hans, C. (2009). Bayesian lasso regression. *Biometrika* **96**, 835–845.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian non-linear state space models. *J. Comp. Graphical Statist.* **5**, 1–25.

Kong, A., Liu, J. S. and Wong, W. (1994). Sequential imputation and Bayesian missing data problems. *J. Amer. Statist. Assoc.* **89**, 590–599.

Liu, J. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.* **93**, 1032–1044.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation based filtering. *Sequential Monte Carlo in Practice*. (A. Doucet, J. F. G. de Freitas and N. J. Gordon, eds.) New York: Springer, 197–223.

Lopes, H. F. (2000). *Bayesian Analysis in Latent Factor and Longitudinal Models*. Ph.D. Thesis, Duke University, USA.

Lopes, H. F. and Polson, N. G. (2010). Extracting SP500 and NASDAQ volatility: The credit crisis of 2007-2008. *The Oxford Handbook of Applied Bayesian Analysis*. (A. O'Hagan and M. West, eds.) Oxford: University Press, 319–342.

Lopes, H. F. and Tsay, R. S. (2011). Bayesian analysis of financial time series via particle filters. *J. Forecast.* **30**, 168–209.

Lund, B. and Lopes, H. F. (2010). Learning in a regime switching macro-finance model for the term structure. *Tech. Rep.*, University of Chicago, USA.

Mukherjee, C. and West, M. (2009). Sequential Monte Carlo in model comparison: Example in cellular dynamics in systems biology. *Tech. Rep.*, Duke University, USA.

Olsson, J., Cappé, O., Douc, R. and Moulines, E. (2008). Sequential Monte Carlo smoothing with application to parameter estimation in non-linear state space models. *Bernoulli* **14**, 155-179.

Perman, M., Pitman, J. and Yor, M. (1992). Size-biased sampling of Poisson point processes and excursions. *Probab. Theory Related Fields* **92**, 21–39.

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

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *J. Amer. Statist. Assoc.* **94**, 590–599.

Polson, N. G., Stroud, J. R. and Müller, P. (2008). Practical filtering with sequential parameter learning. *J. Roy. Statist. Soc. B* **70**, 413–428.

Poyiadjis, G., Doucet, A. and Singh, S. S. (2005). Particle methods for optimal filter derivative: Application to parameter estimation. *Proc, IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing* **5**, 925–928.

Prado, R. and Lopes, H. F. (2010). Sequential parameter learning and filtering in structured autoregressive models. *Tech. Rep.*, University of Chicago, USA.

Prado, R. and West, M. (2010). *Time Series: Modelling, Computation and Inference*. London: Chapman and Hall.

Rios, M. P. and Lopes, H. F. (2010). The extended Liu and West filter: Parameter learning in Markov switching stochastic volatility models. *Tech. Rep.*, University of Chicago, USA.

Ristic, B., Arulampalam, S. and Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Radar Library.

Shi, M. and Dunson, D. (2009). Bayesian variable selection via particle stochastic search. *Tech. Rep.*, Duke University, USA.

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

Taddy, M., Gramacy, R. and Polson, N. G. (2010). Dynamic trees for learning and design. *Tech. Rep.*, University of Chicago, USA.

West, M. (1992). Modelling with mixtures. *Bayesian Statistics 4* (J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith, eds.) Oxford: University Press, 503–524 (with discussion).

West, M. (1993). Mixture models, Monte Carlo, Bayesian updating and dynamic models. *Computing Science and Statistics* **24**, 325–333.

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models* (2nd ed.) Berlin: Springer.

DISCUSSION

MICHAEL PITT (*University of Warwick, UK*)

I would like to congratulate the authors on an interesting paper and presentation. The paper covers a lot of statistical ground ranging from dynamic models through to non-parametric mixture models. The ambition of the paper is to provide a unified approach to estimating such models via sequential Monte Carlo methods.

There are three main approaches which are used in the paper. Firstly, the authors typically consider systems which can be represented by low dimensional representations in the states, or latent variables. Specifically, in the dynamic models context, this means the authors examine models which are mixtures of state space form in which case the particles which are propagated are the sufficient Kalman filter statistics (the prediction mean and variance). Secondly, the authors also use an auxiliary particle filter approach in the fully adapted form. Thirdly, the models are chosen to exploit a conjugate system so that the parameters θ are conjugate to the states, or latent variables.

I shall concentrate most of my remarks on the dynamic model examined in Section 2.2 and illustrated using Example 3. This is,

$$\begin{aligned} &p(y_t|x_t, \lambda_t; \theta), \\ &p(x_{t+1}|x_t, \lambda_t; \theta), \end{aligned} \tag{1}$$

a Gaussian state space form (GSSF) model, conditional upon $\{\lambda_t\}$ which arises according to its own process. Efficient MCMC procedures have been developed

for such models, for example Carter and Kohn (1996). Efficient PF methods for conditionally GSSF model have been proposed by Chen and Liu (2000) and Andrieu and Doucet (2002). However, these PF papers fix the parameters θ and are simply concerned with signal extraction.

This is a useful model for two reasons. Firstly, the authors are able to employ all three of the above tricks in order to obtain efficient inference. Secondly, unlike the static models considered by the authors, there are not really any viable competing approaches to online Bayesian (or frequentist) inference. Clearly, recursively naïvely applying MCMC could be attempted but this will be an $O(T^2)$ algorithm and would quickly become computationally prohibitive.

The first technique exploited by the authors is to represent propagate the sufficient terms arising from the Kalman filter, namely the filtered mean and variance matrices. That is we can, following Chen and Liu (2000), propagate the sufficient terms $S_t^k = \{\widehat{x}_{t|t}^k; P_{t|t}^k\}$, with associated π_t^k, for $k = 1, .., N$. We therefore have a mixture representation of the true filter with the conditional density given by

$$p(x_t|S_t^k) = N(x_t \mid \widehat{x}_{t|t}^k, P_{t|t}^k).$$

This should allow far fewer particles, for a given level of precision, to be used than if the state or signal were itself propagated. The authors use this conjugacy idea also for the Dirichlet process mixture model of Section 2.3. When it is possible to place a model in a conditionally GSSF this results in considerable improvements. The factor model with changing weight structure, in Example 3, is clearly in this form. It should be noted that it is also possible to very efficiently, and unbiassedly, estimate the likelihood arising from conditionally GSSF models. This allows offline MCMC techniques to be used, see Andrieu, Doucet and Holenstein (2010). This could also be considered by the authors.

The second approach is to use an auxiliary particle filter, see Pitt and Shephard (1999), approach in the fully adapted form. In simple notation, and considering a single time step, if $\{\pi_t^k, x_t^k\}$ represents the filter density at time t, where again π_t^k is the mass and x_t^k the state, then (Gordon *et al.*, 1993) update corresponding to,

$$p(x_{t+1}|y_{1:t+1}) \propto \underbrace{p(y_{t+1}|x_{t+1})}_{reweight/resample} \underbrace{\sum_{k=1}^{M} p(x_{t+1}|x_t^k)\pi_t^k}_{simulate}.$$

That is they simulate from the mixture on the right hand side of the expression and then reweight with respect to the measurement density on the left hand side of the expression. This will be inefficient when $p(y_{t+1}|x_{t+1})$ is peaked. When full adaption is possible, *i.e.*, when we can simulate from $p(x_{t+1}|x_t; y_{t+1})$ and evaluate $p(y_{t+1}|x_t)$, then we can rewrite $p(x_{t+1}|y_{1:t+1})$ above as,

$$\begin{aligned} p(x_{t+1}|y_{1:t+1}) &\propto \sum_{k=1}^{M} p(x_{t+1}|x_t^k; y_{t+1}) p(y_{t+1}|x_t^k)\pi_t^k \\ p(x_{t+1}|y_{1:t+1}) &= \underbrace{\sum_{k=1}^{M} p(x_{t+1}|x_t^k; y_{t+1})\pi_{t+1}^k}_{simulate}, \end{aligned}$$

where $\pi_{t+1}^k \propto p(y_{t+1}|x_t^k)\pi_t^k$. This is the fully adapted procedure in Pitt and Shephard (1999) and when it is possible to do this it is almost always preferable in terms

of efficiency, sometimes dramatically so depending on the strength of the signal and whether outliers are present. The auxiliary representation used in Pitt and Shephard (1999) in which x_t^k, or equivalently, the index k is introduced above is not necessary to consider until approximations are used. The approximations in Pitt and Shephard (1999) attempt to get a close as possible to this fully adapted system i.e to approximate $p(y_{t+1}|x_t)$ and $p(x_{t+1}|x_t; y_{t+1})$.

The authors are able to fully adapt in their models as they can write down

$$p(y_{t+1}|S_t^k; \lambda_t) = \sum_{\lambda_{t+1}} p(y_{t+1}|S_t^k, \lambda_{t+1}) q_{\lambda_{t+1}|\lambda_t},$$

integrating, or summing, out the mixing component λ_{t+1}. In this case the reweighting only involve the sufficient, *e.g.*, Kalman filter, statistics for the system and the mixing component from the previous time period, λ_t. This means that the weighting function is relatively flat and so the performance of the resulting filter should be very efficient in the state and the mixing process λ_t, see Figure 3.

These two approaches, sufficient representations for the system combined with fully adapted filters are used well in conjuction with one another. The third main approach the authors use is to exploit the conjugacy of the states with the parameters if this exists. That is we have $p(\theta|S_\theta(x_{1:t}, y_{1:t}, \lambda_{1:t})) = p(\theta|S_t^\theta)$. In a similar manner to the conditionally GSSF models this allows the sufficient quantities associated with θ, S_t^θ, to be propagated through time rather than the parameters themselves. Therefore through time the PF records

$$\widehat{p}(\theta|y_{1:t}) = \sum p(\theta|S_t^{\theta(k)}) \pi_t^k. \tag{2}$$

This gives an estimator of the posterior as we sequentially pass through time. The idea was first introduced by Storvik (2002). A more robust and more general, though more computationally expensive, approach has been proposed and used by Polson, Stroud and Müller (2008). We can therefore never collapse to a single point for θ as we always have a mixture representation.

The parameter learning is not something for nothing in that there are potential pitfalls. Particle filters provide poor representations of smoothing densities. Implicitly, the sufficient statistic $S_t^{\theta(k)}$ is a function of the past state trajectory and whilst there may be many distinct copies of x_t^k for $k = 1, .., N$ there will be very few distinct copies of x_{t-h}^k as h becomes large. Essentially we are considering the parentage of the current particles where there are only very few ancestors going back in time. As a consequence, for a single run of this filter the left hand side of (2) may become too tight and centred at the wrong value relative to the true posterior $p(\theta|y_{1:t})$ as t becomes large. How quickly this degeneracy happens depends upon the length of the times series and the signal to noise ratio. In the models that the authors examine this degeneracy appears to be benign, see Figure 4. This is in part due to the authors integrating out the states when this is possible and using fully adapted procedures when available.

How quickly any degeneracy happens will depend on how informative the measurement density is and also on the particular resampling scheme employed. The authors use a multinomial sampling scheme to do the resampling. As they have taken great care to design methods in which the states are integrated out and full

adaption is used, their measurement density should be less informative than it would be otherwise. In this case they may be able to get much more efficient inference by not unnecessarily throwing particles away. This is the idea of the stratified approach of Kitagawa (1996) and Carpenter, Clifford, Fearnhead (1999). When the observations are relatively uninformative these approaches provide large gains and also will lead to less paths from the past being discarded.

The authors are careful to choose models or to arrange models into a form which allows statistically efficient procedure to be used. I think this is a sensible choice and there are a wide selection of interesting models available in such a form. The approach can clearly be used for a wide variety of existing models estimated currently be MCMC. The techniques could be used for example on the Stochastic Volatility model when written as a mixture SSF following Kim, Shepherd and Chib (1998).

NICOLAS CHOPIN (*CREST, France*) and
CHRISTIAN ROBERT (*Université Paris-Dauphine, France*)

In this discussion, we consider the performance of the particle learning technique of the authors in a limiting case, in order to illustrate the fact that a particle system cannot but degenerate, even when considering sufficient statistics Z_t with fixed dimensions.

Particle system degeneracy. When the authors state that $p(Z^t|y^t)$ *is not of interest as the filtered, low dimensional* $p(Z_t|y^t)$ *is sufficient for inference at time* t, they seem to implicitly imply that the restriction of the simulation focus to a low dimensional vector is a way to avoid the degeneracy inherent to all particle filters (see, *e.g.*, del Moral *et al.*, 2006). However, the degeneracy of particle filters is an unavoidable consequence of the explosion of the state vector Z^t and the issue does not vanish because one is only interested in the marginal

$$p(Z_t|y^t) = \int p(Z^t|y^t)\, \mathrm{d}Z^{-t}\,.$$

Indeed, as shown by the pseudo-code rendering in the paper, the way PL produces a sample from $p(Z_t|y^t)$ is by sequentially simulating Z^t and by extracting Z_t as the final output from this sequence.The PL algorithm therefore relies on an approximation of $p(Z^t|y^t)$ and the fact that this approximation quickly degenerates as t increases, as discussed below and in the companion discussion by Robert and Ryder, obviously has an impact on the approximation of $p(Z_t|y^t)$.

Inherently, particle learning (PL) is at its core an auxiliary particle filter (Pitt and Shephard, 1999) applied in settings where there exists a sufficient statistic (Darmois, 1935) of reduced (or, even better, with fixed) dimension. The simulation scheme thus relies on resampling (Rubin, 1988; Kitagawa, 1996) for adjusting the distribution of the current particle population to the new observation y_{t+1}. Because of this continual resampling, the number of different values of Z_p ($p \geq 1$) contributing to the sufficient statistic Z_t ($t > p$) is decreasing in t at an exponential rate for a fixed p. Therefore, unless the size of the particle population exponentially increases with t (see Douc *et al.*, 2002, and the companion discussion by Chopin and

N. Chopin and C.P. Robert are supported by the 2007–2010 grant ANR-07-BLAN "SP Bayes".

Schäfer), the sample of Z_t's will not be distributed as an iid sample from $p(Z_t|y^t)$. The following section very clearly makes this point through a simple if representative example.

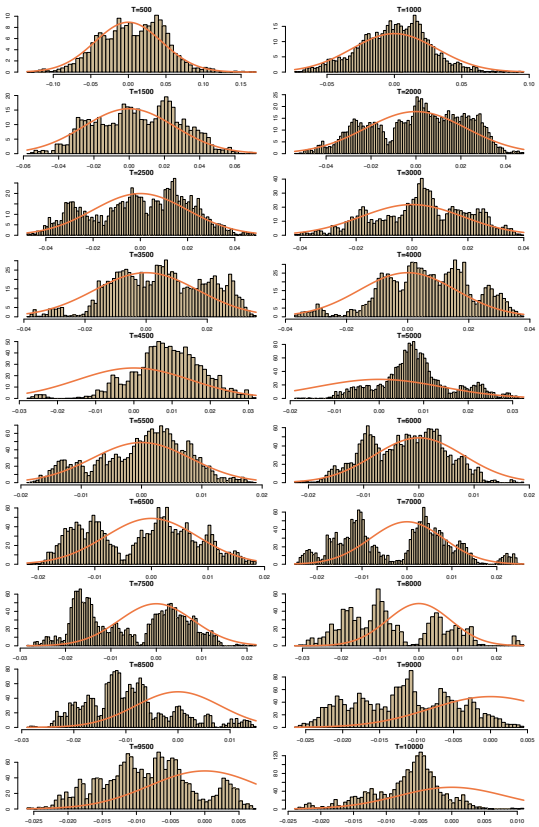

Figure 6: *Evolution of the particle learning sample against the target distribution in terms of the number* $T = 500, \ldots, 10000$ *of iterations, for a particle population of fixed size* 10^4.

A simple particle learning example. Consider the ultimate case where the z_t's are completely independent from the observations y_t, $z_t \sim \mathcal{N}(0, 1)$, and where the empirical average of the z_t's is the sufficient statistic. In this setting, the PL algorithm simplifies into the following iteration t:

(i) Resample uniformly from $(Z_1^t, \ldots, Z_n^t)$ to produce $(\mathfrak{Z}_1^t, \ldots, \mathfrak{Z}_n^t)$;

(ii) Generate $z_{it} \sim \mathcal{N}(0, 1)$;

(iii) Update $Z_i^{t+1} = (t\mathfrak{Z}_i^t + z_{it})/(t + 1)$

The target distribution of the (sufficient) empirical average $Z^t = (z_1 + \ldots + z_t)/t$ is obviously the normal $\mathcal{N}(0, 1/t)$ distribution. A straightforward simulation of the above particle system shows how quickly the degeneracy occurs in the sample: Figure 6 shows a complete lack of fit to the target distribution as early as $t = 500$ simulations when using $10,000$ particles.

Conclusion. The paper fails to mention the well-documented issue of particle degeneracy (Cappé *et al.*, 2004, del Moral *et al.*, 2006), thus giving the impression that PL escapes this problem. Our simple example shows that a particle system cannot be expected to withstand an indeterminate increase in the number of observations without imposing a corresponding exponential increase in the particle size.

NICOLAS CHOPIN (*CREST, France*) and
CHRISTIAN SCHÄFER (*CREST* and *Université Paris-Dauphine, France*)

Much of the confusion around the degeneracy of particle learning and similar algorithms (Fearnhead, 2002; Storvik, 2002) seems related to the lack of formal results regarding the degeneracy of path functional in Sequential Monte Carlo. We would like to report here some preliminary investigation on this subject.

Consider a standard state-space model, with observed process (y_t), and hidden Markov process (x_t), and a basic particle filter, which would track the complete trajectory $x_{1:t}$, *i.e.*, which would produce, at each iteration t, N simulated trajectories $x_{1:t}^{(n)}$, with some weight $w_t^{(n)}$, so as to approximate $p(x_{1:t}|y_{1:t})$. It is well known that the Monte Carlo error regarding the expectation of $\varphi(x_{1:t})$ (a) remains bounded over time if $\varphi(x_{1:t}) = x_t$, (the filtering problem), and (b) blows away, at an exponential rate, if $\varphi(x_{1:t}) = x_1$ (the smoothing problem). Chopin (2004) formalises these two statements by studying the asymptotic variance that appears in the central limit theorem for the corresponding particle estimates.

As mentioned above, and to the best of our knowledge, there is currently no formal result on the divergence of the asymptotic variance for test functions like $\varphi(x_{1:t}) = t^{-1} \sum_{i=1}^t x_i$, *i.e.*, some symmetric function with respect to the complete trajectory. (The fact that this function is a sufficient statistic should not play any role in this convergence study.) One difficulty is that the iterative definition of the asymptotic variance given by Chopin (2004) leads to cumbersome calculations.

We managed however to compute this asymptotic variance exactly, for the Gaussian local level model

$$x_{t+1} \mid x_t \sim N(x_t, 1), \quad y_t \mid x_t \sim N(x_t, 1)$$

and the functional $\varphi(x_{1:t}) = t^{-1} \sum_{i=1}^t x_i$. In this case, the asymptotic variance diverges at rate $O(e^{ct}/t^2)$. Exact calculations may be requested from the authors. We plan to extend these results to a slightly more general model, *e.g.*, with unknown variances, and a function φ which would be a sufficient statistic for such parameters. We conjecture that this exponential divergence occurs for many models: basically, in an average like $\varphi(x_{1:t}) = t^{-1} \sum_{i=1}^t x_i$, the Monte Carlo error attached to x_1/t should be $O(e^{ct}/t^2)$, and should dominate all the other terms. This is at least what one observes in toy examples. After, say, 100 iterations of a particle filter, the

N. Chopin is supported by the 2007–2010 grant ANR-07-BLAN-0237-01 "SP Bayes".

number of distinct values within all the simulated trajectories (that have survived so far) for the component x_1 is typically very small, and the degeneracy in the x_1 dimension seems sufficient to endanger the accuracy of any estimate based on the complete trajectory $x_{1:t}$.

PAUL FEARNHEAD (*Lancaster University, UK*)

One criticism of the idea behind particle learning (and also the earlier, related methods of using MCMC within particle filters) is that the sufficient statistic stored by a particle will depend on the whole history of that particle (*i.e.*, it is defined in terms of all previous values of the state vector). Now theory for particle filters suggest that while they can be efficient for filtering problems, where interest is in the current state, they are not efficient for smoothing problems, where interest is in all previous states. In smoothing problems, the Monte Carlo error will increase, perhaps even exponentially, as the length of the time-series increases.

These results suggest that particle learning could suffer when analysing long-time series. In these cases the approximation of the distribution of sufficient statistics could be poor – intuitively we would expect the approximation to lack diversity relative to the true distribution – which will in turn affect both the updates of the parameters, and the approximation of the posterior distribution of the parameters. These effects are not observed in the simulation examples considered, but do the authors have experience of analysing longer time series, and do they observe the performance of particle learning deteriorating in these cases? One way of counteracting any loss of diversity in the approximation of the distribution of sufficient statistics would be to use kernel density methods (*e.g.*, Liu and West, 2001) when resampling sufficient statistic values.

Finally, on the specific application to nonparametric mixture models. There is some history of applying particle learning-type algorithms to models such as these (*e.g.*, MacEachern *et al.*, 1998, Chen and Liu, 2000, Fearnhead, 2004, 2008). In particular, there may be more efficient resampling algorithms than the one considered in this paper. The key idea is that keeping multiple copies of the same particle is wasteful – you can store the same information through having at most one copy of each particle, and adjusting the weights accordingly. A resampling approach the obeys this principle is used in Fearnhead (2004), and it is shown that such a resampling method can be substantially more efficient than more standard resampling approaches. In some cases the resulting particle filter can be super-efficient – *i.e.*, a particle filter with N particles can be more accurate than inference based on N iid draws from the true posterior.

ALESSANDRA IACOBUCCI (*Université Paris-Dauphine, France*)
CHRISTIAN ROBERT (*Université Paris-Dauphine, France*)
JEAN-MICHEL MARIN (*Université Montpellier 2, France*) and
KERRIE MENGERSEN (*Queensland University of Technology, Australia*)

We now consider the performances of the particle learning (PL) technique in the specific setting of mixtures of distributions and for the approximation of the

J.-M. Marin and C.P. Robert are supported by the 2009–2012 grant ANR-09-BLAN-0218 "Big'MC".

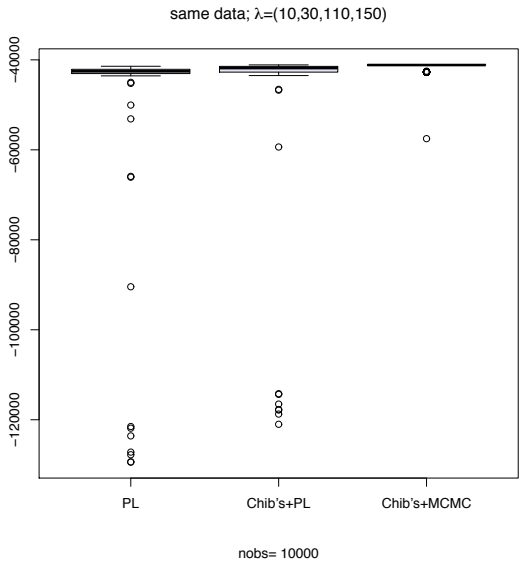

Figure 7: *Range of the evidence approximation based on a PL sample and Lopes et al. (2010) approximation, on a PL sample and Chib's (1995) approximation, on an MCMC sample and Chib's (1995) approximation, for a particle population of size* 10.000*, a mixture with 4 components and scale parameters* $\lambda = (10, 50, 110, 150)$*, and* 683 *replications.*

'evidence', $\mathfrak{Z}_i = \int_{\Theta_i} \pi_i(\theta_i) f_i(y|\theta_i)\, \mathrm{d}\theta_i$, aka the marginal likelihood. Through a simulation experiment, we examine how much the degeneracy that is inherent to particle systems impacts this approximation (We refer the reader to Chen *et al.* (2000), for a general approach to the approximation of evidence and to both Chopin and Robert (2010), and Marin and Robert (2010), for illustrations in the particular setting of mixtures.)

Approximation of the evidence. In the case of a mixture of k Poisson distributions,

$$f(x|\omega, \mu) = \sum_{i=1}^{k} p_i g(x|\lambda_i)\,,$$

taken as an example in Lopes *et al.* (2010), and studied in Carvalho *et al.* (2009) the integrated predictive can be obtained in closed form, as derived in the discussion of Mengersen et al. This implies that the product approximation to the evidence

$$p(y^t) = \prod_{r=1}^{t} p(y_r|y^{r-1}) \approx \prod_{r=1}^{t} \frac{1}{N} \sum_{i=1}^{N} p(y_r|\mathfrak{Z}_{r-1}^{(i)})$$

proposed in Carvalho *et al.* (2009) and Lopes *et al.* (2010) can be implemented here. We thus use the setting of Poisson mixtures to evaluate this PL approximation of the

evidence and we re-evaluate Carvalho *et al.* (2009) assessment that this *"approach offers a simple and robust sequential Monte Carlo alternative to the traditionally hard problem of approximating marginal predictive densities via MCMC output"*.

We note that, since the PL sample is considered as an approximate sample from the posterior $\pi(p, \lambda|y^t)$ it is possible to evaluate the evidence using Chib's (1995) (1995) formula rather than the above proposal of the authors. The availability of an alternative estimator of the evidence allows for a differenciation between the evaluation of approximation [of the target posterior distribution] resulting from the particle system (seen through a possible bias in Chib's, 1995, version) and the evaluation of the approximation [of the evidence] resulting from the use of the product marginal in Lopes *et al.* (2010). Thus, in contrast to the other discussions of ours, we evaluate here the specific degeneracy of the evidence approximation due to using a product of approximations.

A Monte Carlo Experimentation. In order to evaluate the performances of the PL algorithm when compared with the vanilla Gibbs sampler (Diebolt and Robert, 1990, 1994), we simulated 250 samples of size 10^4 from Poisson mixtures with 4 and 5 components and with either widely spaced or close components, $\lambda = (10, 50, 110, 150, 180, 210)$ and $\lambda = (10, 15, 20, 25, 30, 35)$, respectively, and with slightly decreasing weights p_i. We ran a 10^4 iteration Gibbs sampler for Figure 8, performing a further 10^6 iterations as a check of the stability of the MCMC approximation. (For Chib's approximation to perform correctly, as noted in Berkhof *et al.* (2003) and Marin and Robert (2010), it is necessary to average over all $k!$ permutations of the component indices for both the original PL sample and the MCMC sample in order to escape label switching issues.)

The first interesting outcome of our experiment is that the PL sample does not suffer from degeneracy for a small enough number of observations, since the ranges of the Chib's (2005) approximations for both PL and MCMC samples (represented by the second and third columns in the boxplots) are then the same. However, as predicted by the theory (see the discussions by Chopin and Robert, and by Robert and Ryder), increasing the number of observations without simultaneously and exponentially increasing the number of particles necessarily leads to the degeneracy of the simulated sufficient statistic paths. In our experiment, this degeneracy always occurs between $5,000$ and $10,000$ observations. The phenomenon clearly appears on Figure 8 where both the range and the extremes of the evidence approximations significantly differ on the right hand side boxplot graph. (Again, the stability of the MCMC range was tested by running the Gibbs sampler for much longer and observing no variation.) This divergence is to be contrasted with Figure 1 in Carvalho *et al.* (2009) which concludes to an agreement between all approximations to the Bayes factor.

The second result that is relevant for our discussion is that the new approximation to the evidence proposed by the authors suffers from a severe bias as one proceeds through the observations. This issue is apparently unrelated to the degeneracy phenomenon observed above in that the discrepancy starts from the beginning, the closest approximation occuring for $n = 1,000$ observations. Note that Carvalho *et al.* (2009) mention that the evidence approximation based on particle learning was less variable. While this feature is not visible in our experiment, it is not necessarily a positive feature in any case, as shown in the current experiment. (In order to provide a better rendering of the comparison between the PL and the MCMC algorithms, we excluded the outliers from all boxplots. We however stress

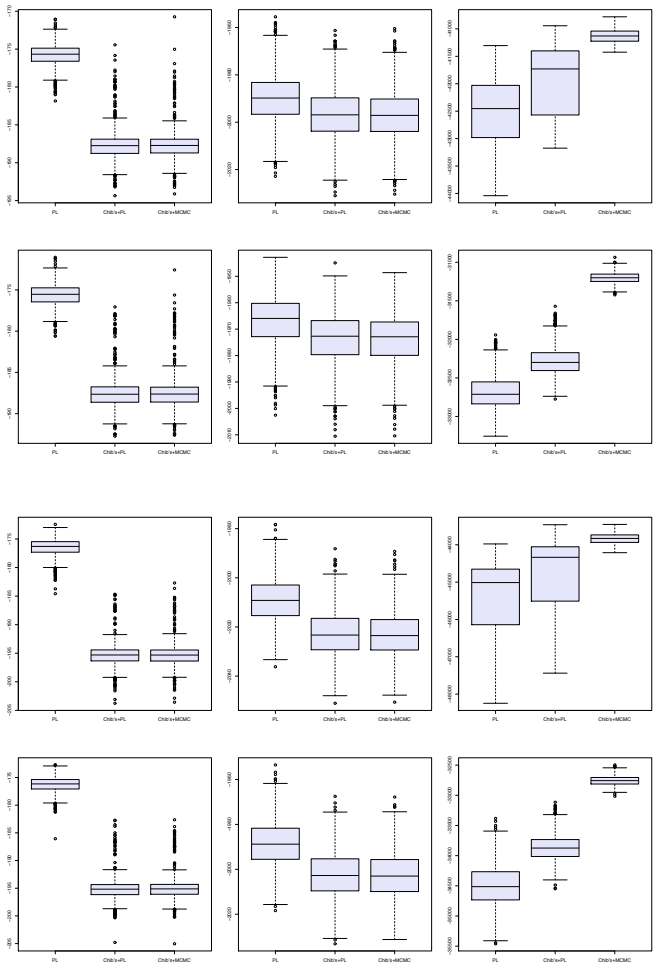

Figure 8: *Evolution against the number of observations ($n = 100$, $n = 1,000$ and $n = 10,000$, from left to right) of the evidence approximation based on a PL sample and Lopes et al. (2010) approximation, on a PL sample and Chib's (1995) approximation, on an MCMC sample and Chib's (1995) approximation, for a particle population of size* $10,000$*, a mixture with* 4 *components and scale parameters* $\lambda = (10, 50, 110, 150)$ *(first row) and* $\lambda = (10, 15, 20, 25)$ *(second row), and for* 5 *components (last two rows).*

that both PL approaches had a higher propensity to outlying behaviour.) In the strongest case of discrepancy between PL and MCMC found in our experiment, Figure 9 illustrates the departure between the three approaches from a particularly influential observation, since the graphs are compared in terms of evidence *per*

observation.

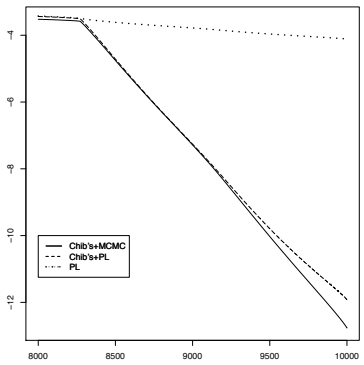

Figure 9: *Evolution of the three approximations of the evidence* per observation *against the number of observations for a specific sample simulated from the same Poisson mixture as in the top panel of Figure 8.*

We thus conclude at the lack of robustness of the new approximation of evidence suggested in both Carvalho *et al.* (2009) and Lopes *et al.* (2010) (besides providing a reinforced demonstration of the overall difficulty with degeneracy).

Following the floor discussion at the conference, we want to point out here that the divergence between the evidence evaluations observed in the discussion of Iacobucci et al. is not the result of an outlying Monte Carlo experiment but indeed a distributional property. This can be seen on Figure 7 on the variation of the evidence in the specific setting of mixtures of Poisson distributions. For two given data sets, we repeated 683 times the three evidence approximations using the method proposed in this paper and the method in Chib (1995) applied to both the PL and MCMC samples. The divergence between the three evaluations is consistent across simulations, so repeating simulations does not help in eliminating this divergence.

DANIEL MERL (*Lawrence Livermore National Laboratory, USA*)

I would like to congratulate Lopes, Carvalho, Johannes, and Polson on their paper, which in my opinion makes an important contribution to the literature on computationally efficient Bayesian inference. As an industrial statistician concerned primarily with the successful deployment of modern predictive modeling techniques to live information systems, my main observation about PL is that it enables Bayesian inference in settings where few or no alternative inference algorithms exist. A generic but important example of such a system is a continuously observed, very high frequency time series. In this type of setting, the frequency and duration of observation may be such that indefinite storage of the incoming data is not possible, thus eliminating retrospective inference methods such as MCMC. Additionally, the indefinite duration of observation implies that SMC methods in which the particles involve ever-expanding state space representations will eventually overflow any finite memory computing platform. PL, however, is ideally suited for such systems due to its ability to achieve posterior inference in a single pass over the observations and its

ability to concisely represent the model state space via sufficient statistics that grow in dimension much more slowly than O(N) (where N is the number of observations).

It seems worth mentioning that the PL algorithm as described in these proceedings should be regarded as the *vanilla* PL, and I assume the authors would agree that there exist the usual opportunities for algorithmic enhancement. In particular, enhancements that confer computational benefits are especially necessary in limited storage settings, which obviously require not just sequential but *real time* inference. Several such improvements to the vanilla PL algorithm that have proven useful in practice include stratified resampling rather than multinomial resampling, effective sample size (ESS) based resampling rather than constant resampling, and multithreaded rather than single threaded implementations.

A final note regarding the latter: it is often remarked that SMC approaches are "embarrassingly parallel" and therefore trivially parallelized. While it is true that the most computationally intensive components of SMC algorithms can be computed independently and in parallel, such as the evaluations of weights and the particle propagations, the resampling step, which sits squarely between these components, can not. It is this break in the parallelism that makes SMC approaches such as PL in general *not* ideally suited for implementation on graphical processing units. In his talk during an ISBA session at this meeting, Professor Chris Holmes described an approximately 30 fold improvement for a GPU enabled SMC algorithm (as opposed to over two orders of magnitude improvement for MCMC). Although it is possible to gain more substantial improvements for SMC algorithms through a GPU implementation incorporating the adaptive resampling techniques described above (see Lee *et. al.* 2009), in the meantime, it is all but trivial to employ shared memory multithreading via tools like `OpenMP` to effect an order of magnitude improvement to even the vanilla PL simply by utilizing the multiple processor cores of modern desktop computers.

CHRISTIAN ROBERT (*Université Paris-Dauphine, France*)
ROBIN RYDER (*Université Paris-Dauphine and CREST, France*) and
NICOLAS CHOPIN (*CREST, France*)

In connection with the discussion by Chopin and Robert, we detail here how the degeneracy dynamics of the particle learning technique presented in this paper impacts the distribution of the sufficient (or "essential state vector") statistics.

The authors focus on the distribution of a sufficient statistic, $p(Z_t|y^t)$, at time t. By insisting both on the low dimensionality of Z_t and on the sufficiency, they give the reader the impression that the poor approximation of the state vector Z^t resulting from the resampling propagation scheme does not impact $p(Z_t|y^t)$, since their statement *"at time T, PL provides the filtered distribution of the last essential state vector Z_T, namely $p(Z_T|y^T)$"* (Section 1.2) does not mention any deterioration in the approximation—this is how we understand *filtered*—provided by PL. Because particle learning is inherently a particle filter (Pitt and Shephard, 1999), this intuition is unfortunately wrong, as shown below in the case of an empirical average of the past auxiliary variables Z_t. Contrary to the belief that *"resampling (...)*

C.P. Robert and N. Chopin are supported by the 2007–2010 grant ANR-07-BLAN "SP Bayes". Robin Ryder is funded by a postdoctoral fellowship from the Fondation des Sciences Mathématiques de Paris.

is fundamental in avoiding a decay" (Section 1.2), resampling necessarily leads to degeneracy unless the size of the particle population increases exponentially with t.

We thus consider again the case introduced by Chopin and Robert in their discussion, when the auxiliary variables $z_t \sim \mathcal{N}(0,1)$ are independent from the observations y_t and where the essential state vector statistic is the empirical average of the z_t's. In this case, the distribution of the empirical average

$$Z^t = (z_1 + \ldots + z_t)/t$$

is the normal $\mathcal{N}(0, 1/t)$ distribution, but the particle population degenerates into a single path from the point of view of this sufficient statistic. In other words, degeneracy occurs much faster than the root T forgetting of the past of the particle path that is due to the averaging. In order to support this perspective, we provide here a derivation of the variance of the particle population after t iterations.

Using the same notations as in Chopin and Robert, since $\mathbb{E}[Z_i^t] = 0$,

$$\mathrm{Var}(Z_i^t) = 1/t, \quad Z_i^t = \frac{t-1}{t}\mathfrak{Z}_i^t + \frac{z_{it}}{t},$$

we consider

$$\begin{aligned}
\mathbb{E}[Z_i^t Z_j^t] &= \left(\frac{t-1}{t}\right)^2 \mathbb{E}[\mathfrak{Z}_i^t \mathfrak{Z}_j^t] \\
&= \left(\frac{t-1}{t}\right)^2 \left(\mathbb{P}[\mathfrak{Z}_i^{t-1} = \mathfrak{Z}_j^{t-1}]\mathbb{E}[(\mathfrak{Z}_i^{t-1})^2] + \mathbb{P}[\mathfrak{Z}_i^{t-1} \neq \mathfrak{Z}_j^{t-1}]\mathbb{E}[\mathfrak{Z}_i^{t-1}\mathfrak{Z}_j^{t-1}]\right) \\
&= \left(\frac{t-1}{t}\right)^2 \left(\frac{1}{n}\frac{1}{t-1} + \frac{n-1}{n}\mathbb{E}[Z_i^{t-1} Z_j^{t-1}]\right).
\end{aligned}$$

Now let $u_t = t^2\mathbb{E}[Z_i^t Z_j^t] - t + n$. The last line becomes $u_t = \frac{n-1}{n}u_{t-1}$. Since $u_1 = n-1$, we have

$$\begin{aligned}
\mathbb{E}[Z_i^t Z_j^t] &= \frac{u_t + t - n}{t^2} = \frac{\left(\frac{n-1}{n}\right)^{t-1}(n-1) + t - n}{t^2} \\
&= \frac{1}{t^2 n^{t-1}}\left\{(n-1)^t - n^t + tn^{t-1}\right\} = \frac{t-1}{2t}\frac{n^{t-2}}{n^{t-1}} + \cdots = \mathrm{O}_n(n^{-1}).
\end{aligned}$$

In conclusion,

$$\begin{aligned}
\mathrm{var}(\overline{Z}_i^t) &= \frac{1}{nt} + \frac{n(n-1)}{n^2}\frac{1}{t^2 n^{t-1}}\left\{(n-1)^t - n^t + tn^{t-1}\right\} \\
&= \frac{1}{nt}\left[1 + \frac{n(n-1)}{t}\left\{(1-1/n)^t - 1 + t/n\right\}\right].
\end{aligned}$$

For n fixed, and $t \to +\infty$, $t\mathrm{var}(\overline{Z}_i^t) \to 1$, a limit that does not depend on n, *i.e.*, the system eventually degenerates to a single path. If we set $n = ct$, then $nt\mathrm{var}(\overline{Z}_i^t) \to C$, for some $C > 0$. Bearing in mind that the actual posterior variance should be $O(t^{-1})$, this means that, to bound the *relative error* uniformly over a given time interval, *i.e.*, for $t = 1, \ldots T$, one must take $n = O(T)$.

REPLY TO THE DISCUSSION

We would like to enthusiastically thank the discussants Mike Pitt, Christian Robert's multinational team, Paul Fearnhead and Dan Merl for their contributions. Hopefully our comments will make PL's scope, strengths and weaknesses clear, particularly to those readers interested in sequential parameter Bayesian computation. We would like to organize our comments into the following topics: approximating predictive densities, outliers and model misspecification, sufficient statistics, MC error accumulation, PL and MCMC and resampling. Pitt's discussion is mainly focused on PL for dynamic models (Carvalho *et al.*, 2010, Lopes and Tsay, 2011). Similarly, several of Robert *et al.* discussion are based on the mixture of Poisson distributions from Carvalho *et al.* (2009). Therefore, some readers might benefit from browsing through those papers before engaging in our comments. Fearnhead's and Merl's discussion are solely based on our chapter.

Approximating predictive densities. Iacobucci, Robert, Marin and Mergensen and Iacobucci, Marin and Robert suggest alternative approximations, still based on PL samples, to the predictive density. This is clearly a good idea. Examples A and B below proved some simulation evidence: PL (based on the product estimate) and MCMC (based on Chib's method) produce relatively similar results either for small or large samples. Chib's method – as it uses extra "analytical" information – might outperform the product estimator[1] in some scenarios with the well-known caveat of its potential high variability (See Example A). Neal (1999) and Polson (2007) point out that Chib's method and variants thereof can have poor MC properties which are exacerbated when MCMC convergence is prohibitively slow. The product estimate is naturally sequential and easy to implement but is potentially biased. Appealing to alternatives that exploit functional forms and/or the conditional structure of the model such as Savage-Dickey density ratio estimates or Rao-Blackwellized estimates, amongst others, is clearly preferable when available.

Outliers and model misspecification. One well-known fact is that all particle methods breakdown when there are outliers (misspecified models). A known drawback, also shared by all alternative particle filters, is the accumulation of MC error (for example, in the presence of sequences of outliers). We show in Carvalho *et al.* (2010) that PL has better properties than alternative filters in the presence of outliers. Example C clearly shows in an extreme situation that even $N = 100,000$ particles will not overcome a large outlier – even though PL vastly outperforms standard alternatives.

Sufficient statistics. One area where we strongly disagree with Chopin, Robert and colleagues is our use of the essential state vector, Z_t. Our view is that this is key to sequential parameter Bayesian computation as it converts the sequential learning problem to a filtering problem for Z_t, *i.e.*, find $p(Z_t|y^t)$ for $1 \leq t \leq T$. Without this extra structure, we feel that black-box sequential importance sampling algorithms and related central limit theorems are of little use in practice.

It appears that one source of confusion is that the calculation of the marginal filtering distribution $p(Z_T|y^T)$ is alligned with the full posterior smoothing problem,

[1] In a simple example of the application of Chib's method, also known as the candidate estimator (Besag, 1989), the predictive $p(y)$ is approximated by $\hat{p}(y) = p(y|\tilde{\theta})p(\tilde{\theta})/p(\tilde{\theta}_1|\tilde{\theta}_2, y)\hat{p}(\tilde{\theta}_2|y)$, where $\tilde{\theta} = (\tilde{\theta}_1, \tilde{\theta}_2)$ is any value of θ, say the posterior mode or the posterior mean, and $\hat{p}(\tilde{\theta}_2|y)$ is a Monte Carlo approximation to $p(\tilde{\theta}_2|y)$, say $N^{-1}\sum_i p(\tilde{\theta}_2|\theta_{1i}, y)$, where $\theta_{11}, \ldots, \theta_{1N}$ are draws from $p(\theta_1|y)$.

$p(x_1, \ldots, x_T|y^T)$. Clearly, if one solves the smoothing problem (a T dimensional joint posterior), the distribution of Z_T follows as a marginal. The converse is clearly not true – one might be able to accurately estimate the functional $p(Z_T|y^T)$ whilst having no idea about the full joint. For example, from the forward filtering PL algorithm $p(Z_1|Y^T)$ will have collapsed on one particle. We note that Carvalho *et al.* (2010) also provide a smoothing algorithm with parameter learning – extending Godsill, Doucet and West (2004) – but this is $O(N^2)$ (see discussions by Pitt and Fearnhead).

Pitt, Chopin and Robert, Chopin and Schäfer, Robert, Ryder and Chopin and Fearnhead all comment on the potential particle degeneracy of the parameter sufficient statistics. Our view is that you have to separate the concepts of degeneracy and accumulation of MC error. Now we will provide two standard examples (including the local level model of Chopin and Schäfer) illustrating how PL does in fact accurately learns Z_T. In example D, PL is implemented with conditional parameter sufficient statistics for a large sample size $n = 5000$ and same order of magnitude particle size $N = 1000$. Despite the very simplistic nature of the example, PL and MCMC produce fairly similar approximations. We carefully revisit the first order dynamic linear model discussed by Chopin and Schäfer in example E. It appears then that for PL to "degenerate" as the discussants suggest the time series length n will have to be many orders of magnitude larger than N. Robert *et al.* seem intent on using $N = 1000$ particles in 5000 dimensional problems and showing poor Monte Carlo performance – there really shouldn't be surprised at all with some of their findings. Addressing real problems and showing when large Monte Carlo samples are needed is clearly an area for future research much in the same way that the MCMC convergence literature evolved.

One of the main criticisms running through the discussions, as well as the literature (*e.g.*, , Kantas *et al.*, 2009), is that the parameter estimation problem with sufficient statistics is equivalent to learning additive functionals of the states of the form $s_{n+1} = s_n + \phi(x_{n+1}) = \sum_{t=1}^{n} \phi(x_t, x_{t-1})$. The line of argument continues that well known limiting results, such as those in Olsson *et al.* (2008), indicate that the variance of the Monte Carlo estimates of $E(s_n|\theta, y^n)$ increases quadratically with time, since it involves approximating $p(x^n|y^n)$, the smoothing distribution. Thus, PL inherently 'degenerate', in the sense that the Monte Carlo variance will 'blow-up', and thus is unreliable. This argument appears repeatedly in the literature.

This argument is incorrect and extremely misleading for two reasons. First, what appears in the posteriors that we sample from, $p(\theta|s_n)$, are not terms like $s(x^n) = \sum_{t=1}^{n} \phi(x_t, x_{t-1})$, but rather *time-averaged* terms like $\widetilde{s}(x^n) = \sum_{t=1}^{n} \phi(x_t, x_{t-1})/n$. This point was mentioned in the discussion by Chopin and Schäfer and, in our view, is crucial. For example, think about learning the mean α in the local level model: $y_t|x_t \sim N(\alpha + x_t, \sigma^2)$ and $x_t|x_{t-1} \sim N(x_{t-1}, \tau^2)$. Here, the posterior for α will depend on $\sum_{t=1}^{n}(y_t - x_t)/n = \sum_{t=1}^{n} y_t/n - \sum_{t=1}^{n} x_t/n$, and the first term is observed. More generally, the terms that appear in the posteriors are $\sum_{t=1}^{n} x_t/n$, $\sum_{t=1}^{n} x_t^2/n$, and $\sum_{t=1}^{n} x_t x_{t-1}/n$, all of which are time averaged.

Second, time-averaging matters. Targets like $\sum_{t=1}^{n} \phi(x_t, x_{t-1})/n$ do not grow for large n, at least in stationary models. Because of that, they are easier to estimate than a moving target because, for example, its variance does not increase with time (in population). Potentially, it is even easier than estimating $E(x_n|y^n)$. This can actually be seen from figures 2 and 3 in Olsson *et al.* (2008). They show the Monte Carlo error in estimating $s_2(x^n) = \sum_{t=1}^{n} x_t^2/n$, holding the number of

particles fixed at $N = 1000$ (a very small number). It is obvious that the Monte Carlo variance decreases over time. For the local level model, we repeat these calculations in example E. Again, it is obvious the Monte Carlo variance associated with estimating $s_n = \sum_{t=1}^{n} x_t/n$ decreases with n (even though this model is non-stationary). See figures (c) and (d) of example E. This holds more generally, and we have verified this for a range of models and sufficient statistics. We could imagine if the model were strongly non-stationary, that time-averaging might not mitigate the error accumulation. Our conjecture is that the Monte Carlo variance decreases provided the errors in estimating the current state do not increase too rapidly. This seems to hold in common specifications.

PL parameter particles do not degenerate (as they are drawn offline if need be). Particles in PL, per se, never degenerate – we draw exactly from the mixture approximation and resampling first avoids degeneracy problems that plagued previous parameter learning attempts. This is the main advantage of PL over previous attempts where θ is part of the particle set and, after degeneration, would have to be rejuvenated (with an MCMC step).

Accumulation of MC error. The more interesting problem (as with MCMC convergence checks) is how MC errors accumulate in PL. General bounds, such as those provided by Chopin and Schäfer, seem to be of little use. Due to the simplicity of implementation, it is quite straightforward to address this via simulation. Consider the first order dynamic linear model of Chopin and Schäfer with $p(y_t|x_t) \sim N(x_t, \sigma^2)$, $p(x_t|x_{t-1}) \sim N(x_{t-1}, \tau^2)$ and $p(x_0) \sim N(0, C_0)$, for known variances σ^2, τ^2 and C_0. The predictive and propagation distributions needed for PL are $p(y_{t+1}|x_t) \sim N(x_t, \sigma^2 + \tau^2)$ and $(x_{t+1}|x_t, y_{t+1}) \sim N\left(Ay_{t+1} + (1-A)x_t, A\sigma^2\right)$, respectively, where $A = \tau^2/(\tau^2+\sigma^2)$. It is instructive to analyze the MC error at the first step and then argue by induction (see *e.g.*, Godsill et al, 2004). Here we have $p(y_1|x_0) \sim N(x_0, \sigma^2+\tau^2)$ and $p(y_1) \sim N(0, \sigma^2+\tau^2+C_0)$ and $p(x_1) \sim N(0, \tau^2+C_0)$. There is the usual relative MC error bound to approximate the marginal distribution $p^N(y_1)$ to $p(y_1)$ (functionals $\phi(x_t)$ can be analyzed in a similar fashion). We need to compare the bounds produced by PL and SIS, *i.e.*, compare the right hand side of $Var_{PL}\left(p^N(y_1)/p(y_1)\right) \leq (Np^2(y_1))^{-1} E_{p(x_0)}[p^2(y_1|x_0)]$ to the right hand side of $Var_{SIS}\left(p^N(y_1)/p(y_1)\right) \leq (Np^2(y_1))^{-1} E_{p(x_1)}[p^2(y_1|x_1)]$, or simply study the behavior of the ratio $E_{p(x_0)}[p^2(y_1|x_0)]/E_{p(x_1)}[p^2(y_1|x_1)]$. Example F shows that, in this context, PL bounds are always smaller than SIS bounds. The only situation where PL and SIS behave similarly is when τ^2 is small relative to σ^2 and, simultaneously, C_0 is large, *i.e.*, when the state evolution process informs very little about the observation evolution process and ones current information about where the state is moving to is rather vague.

PL versus MCMC. MCMC methods have proven to be very effective in a large number of highly complex and structured frameworks, some of which studied by us in our papers and books. Our claim, mistakenly interpreted as dismissive of MCMC in the discussion by Mergensen, Iacobucci and Robert, is that PL is an attractive alternative to MCMC schemes in certain classes of models and, more importantly, MCMC is inherently non-sequential. As Pitt, one of the proponents of the APF, properly says, "the approach can clearly be used for a wide variety of existing models estimated currently by MCMC." The literature we cite in the paper include several serious applications of PL to situations other than the illustrative and pedagogical ones we decided to include. One particular example is the PL implementation for general mixture models in Carvalho *et al.* (2009).

Resampling schemes. Pitt, Fearnhead and Merl all suggested stratified sampling over naïve multinomial sampling. Clearly this has advantages. We support and magnify their advise and suggest that more clever resampling schemes, normalized by their computational cost, should be the norm, not the exception. This has shown to be drastically important particularly when using (partially) blind particle filters, such as the sequential importance sampling with resampling filter.

Recommendations.

(i) (G_0, G): MCMC schemes depend upon the not so trivial task of assessing convergence. How long should the burn-in G_0 be? (Polson, 1996). Besides, MCMC schemes produce G dependent draws.

(ii) (T, N): PL schemes, as well as all particle filters, have to increase the number of particles N with the sample size T. Monte Carlo error is usually of the form $C_T/\sqrt{N}$, with $1/\sqrt{N}$ representing the particle filter's main strength and C_T its main weakness.

(iii) *Propagation-resampling* schemes, such as the bootstrap filter and SIS filters, are generally outperformed by *resampling-propagation* schemes, such as AP filters and PL schemes.

(iv) What seems, at first glance, to be a drawback of PL, *i.e.*, the existence of several different essential vectors Z_ts for any single problem, is in fact PL's comparative advantage. The clever investigation of which essential vector to choose in a given situation can potentially lead to realistically more efficient PL schemes.

Example A: PL versus Chib's+MCMC.

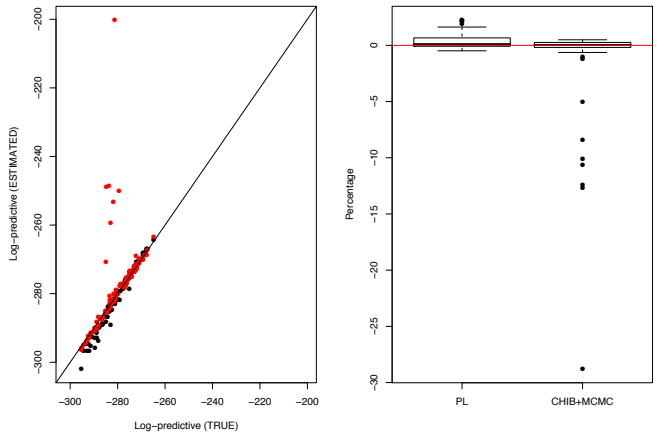

Figure 10: *Example A*

Comparison of PL and Chib's + MCMC when approximating $p(y)$, the predictive likelihood of a two-component mixture of Poisson distributions.

For $t = 1, \ldots, n$, from $y_t \sim \alpha Poi(\gamma_1) + (1-\alpha) Poi(\gamma_2)$, where $n = 100$, $(\gamma_1, \gamma_2) = (10, 15)$ and $\alpha = 0.75$. Model is fit via PL and MCMC with prior $p(\gamma_1, \gamma_2, \alpha) = p_G(\gamma_1; 1, 0.1)$ $p_G(\gamma_2; 1.5, 0.1)$, for $\gamma_1, \gamma_2 > 0$ and $\alpha \in (0, 1)$. The particle size for PL is $N = 1000$, while MCMC is run for 2000 iterations with the 2nd half kept for inference. Both MC schemes are run for each one of $S = 100$ data sets. PL seems slightly more robust than MCMC when $n = 100$, where MCMC percentage error can be as big as 30%. MCMC dominates PL when $n = 1000$, however the percentage error is below 1%.

Example B: PL versus Chib's+PL. In this example we show that PL and Chib's PL produce comparable results for samples of size up to $n = 200$, which we consider large for the complexity of the model. We simulate $S = 50$ samples with n i.i.d. $N(0, 1)$ observations. The sample size n varies in $\{20, \ldots, 100, 200\}$, leading to 500 samples. For each sample we fit the simple normal model with conjugate prior for the mean and variance parameters, *i.e.*, $y_t \sim N(\theta, \sigma^2)$ $(t = 1, \ldots, n)$, $\theta|\sigma^2 \sim N(0, \sigma^2)$ and $\sigma^2 \sim IG(10, 9)$. In this case the exact value of $p(y)$ is easily obtained since the marginal distribution of y is $t_{20}(0_n, 1.8I_n)$. We run $R = 50$ times PL, each time based on $N = 500$ particles, *i.e.*, the same order of magnitude of the sample size. PL does not take advantage of prior conjugacy, so that during propagation θs is propagated based on resample σ^2s, which is then used to propagate σ^2s. By doing that we show that the *essential* state vector depends on both σ^2 (when propagating θ) and θ (when propagating σ^2). For any given sample size n, we compute the mean absolute error (in percentage) as $MAE(n) = \frac{100}{SR} \sum_{s=1}^{S} |\sum_{r=1}^{R} \log p_{pl}^r(y_s) / \log p(y_s) - R|$, where $\log p_{pl}^r(y_s)$ is r^{th} PL approximation to $p(y_s)$ and y_s is the s^{th} sample of size n. PL is slightly better than Chib's+PL.

Table 1: *Example B*

	Mean absolute deviation					
n	20	40	60	80	100	200
PL	3.222	1.750	0.980	0.752	0.774	0.276
Chib's+PL	3.311	1.782	1.019	0.765	0.769	0.279

Example C: PL versus SISR. Let consider the basic local level model, *i.e.*, $y_t|x_t \sim N(x_t, 1)$ and $x_t|x_{t-1} \sim N(x_{t-1}, 1)$, for $t = 1, \ldots, n$ and $x_0 \sim N(0, C_0)$. The MC study shows that PL has smaller MC error than SISR when approximating $\log p(y_1, y_2)$ in the presence of an outlier in the observation equation when $C_0 = 1$, $n = 2$, $y_2 = 0$ and $y_1 = 2$ (panel (a)) or $y_1 = 20$ (panel (b)).

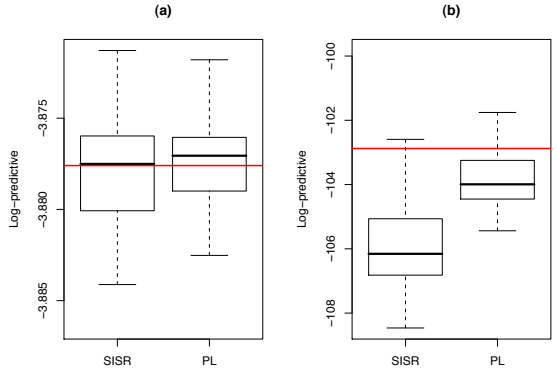

Figure 11: *Example C*

Example D: PL versus MCMC.

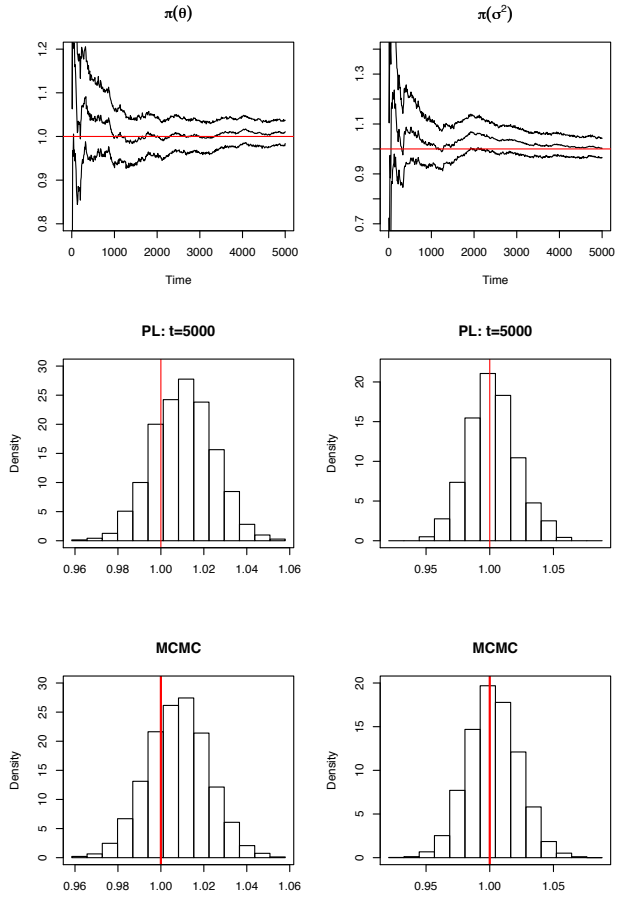

Figure 12: *Example D*

We simulate $n = 5000$ data points from $y_t \sim N(1, 1)$, and fit the model $y_t \sim N(\theta, \sigma^2)$ and $(\theta, \sigma^2) \sim N(m_0, C_0)IG(a_0, b_0)$, where $m_0 = 0$, $C_0 = 10$, $a_0 = 3$ and $b_0 = 2$ (relatively vague prior information). MCMC is a Gibbs sampler with full conditionals $\theta|\sigma^2, y \sim N(m_n, C_n)$ and $\sigma^2|\theta, y \sim IG(a_n, b_n)$, for $C_n = 1/(1/C_0 + n/\sigma^2)$, $m_n = C_n(m_0/C_0 + n\bar{y}/\sigma^2)$, $a_n = a_0 + n/2$ and $b_n = b_0 + \sum_{t=1}^{n}(y_t - \theta)^2/2$. The Gibbs sampler started at $\sigma^{2(0)} = 1.0$ and was run for 20,000 draws discarding the first half. PL runs from $t = 1$ to $t = n$ as follows: 1) Let $\{(m_{t-1}, C_{t-1}, a_{t-1}, b_{t-1}, \sigma^2)^{(i)}\}_{i=1}^{N}$ be the particle set at time $t - 1$, with $s_1 = 1/C_{t-1}$ and $s_2 = m_{t-1}/C_{t-1}$; 2) resample the set with weights $w_t^{(i)} \propto f_N(y_t; m_{t-1}^{(i)}, C_{t-1}^{(i)} + \sigma^{2(i)})$; 3) compute $s_1^{(i)} = \tilde{s}_1^{(i)} + 1/\tilde{\sigma}^{2(i)}$, $s_2^{(i)} = \tilde{s}_2^{(i)} + y_t/\tilde{\sigma}^{2(i)}$, $a_t = a_{t-1} + 1/2$, $C_t^{(i)} = 1/s_1^{(i)}$ and $m_t^{(i)} = C_t^{(i)} s_2^{(i)}$; 4) draw $\theta^{(i)} \sim N(m_t^{(i)}, C_t^{(i)})$; 5) compute $b_t^{(i)} = \tilde{b}_{t-1}^{(i)} + (y_t - \theta^{(i)})^2/2$; and 6) draw $\sigma^{2(i)} \sim IG(a_t^{(i)}, b_t^{(i)})$. PL results are based on $N = 1000$ particles.

Example E: Sufficient statistics.

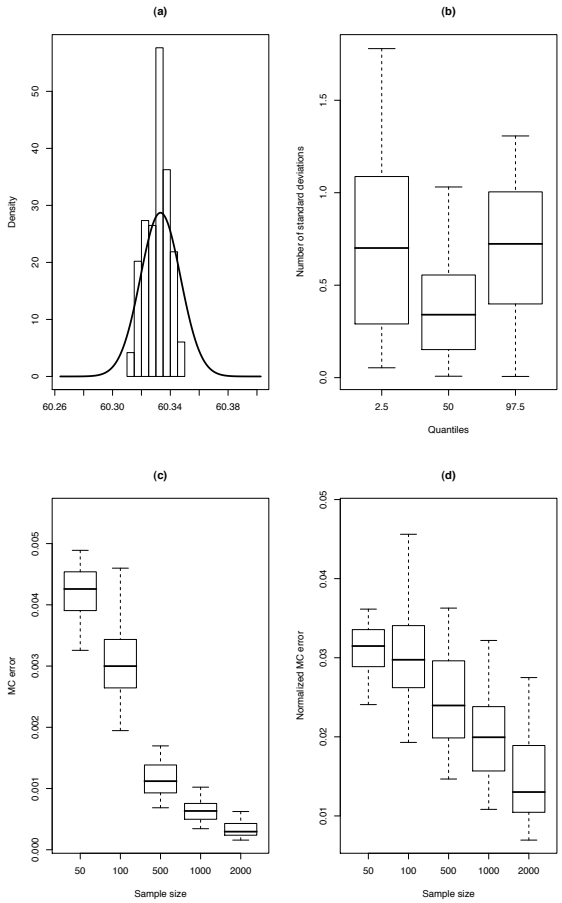

Figure 13: *Example E*

For $t = 1, \ldots, n$, let us consider the local level model where $y_t|x_t, \sigma^2 \sim N(x_t, \sigma^2)$, $x_t|x_{t-1}, \sigma^2 \sim N(x_{t-1}, \sigma^2)$, $x_0|\sigma^2 \sim N(m_0, \sigma^2)$ and $\sigma^2 \sim IG(c_0, d_0)$. It is easy to see that the joint prior of $x = (x_1, \ldots, x_n)'$ is multivariate normal with mean $\mu_0 = 1_n m_0$ and precision $\sigma^{-2}\Phi_0$, where $\Phi_{0,ij} = 0$ for all $|i-j| > 1$, $\Phi_{0,ij} = -1$ for all $|i-j| = 1$, $\Phi_{0,ii} = 2$ for all $i = 1, \ldots, n-1$ and $\Phi_{0,nn} = 1$. Combining this (improper) prior with the normal model for $y = (y_1, \ldots, y_n)$, $y|x, \sigma^2 \sim N(x, \sigma^2 I_n)$, leads to the joint posterior of x being normal with mean $\mu_n = \Phi_n^{-1}(\Phi_0\mu_0 + y)$ and variance $\sigma^2\Phi_n^{-1}$, for $\Phi_n = \Phi_0 + I_n$. Therefore, conditional on σ^2, the posterior distribution of $s_n = \sum_{t=1}^n x_t/n = 1_n'x/n$ is normal with mean $a_n = 1_n'\mu_n/n$ and variance $\sigma^2 b_n$, where $b_n = 1_n'\Phi_n^{-1}1_n/n^2$. It is also easy to see that $\sigma^2|y \sim IG(c_n, d_n)$ where $c_n = c_0 + n/2$ and $d_n = d_0 + (y'y + \mu_0'\Phi_0\mu_0 - \mu_n'\Phi_n\mu_n)/2$, so that $s_n|y \sim t_{2c_n}(a_n, b_n d_n/c_n)$. In addition, it is easy to see that $(\sigma^2|y^t, x^t) \sim IG(c_t, d_t)$, where $y^t = (y_1, \ldots, y_t)$, $c_t = c_{t-1} + 1$ and $d_t = d_{t-1} + (y_t - x_t)^2 + (x_t - x_{t-1})^2$.

In this exercise, the sample size is $n = 5000$ and particle size $N = 10000$, for $m_0 = x_0 = 0$, $c_0 = 10$, $d_0 = 9$ and $R = 50$ runs of PL. (a) Histogram approximating $p(s_n|y)$ for one of the runs. (b) Box-plots of distances (in number of standard deviations) between approximate quantiles based on the $R = 50$ histograms and the true Student's t quantiles for $p(s_n|y)$. (c) MC error measured as the standard deviation PL's estimate of $E(s_n|y^n)$ over the $R = 50$ runs and different sample sizes. (d) Same as (c) but normalized by the true value of $\sqrt{V(s_n|y^n)}$.

Example F: PL versus SIS bounds. Surface ratio $E_{p(x_0)}[p^2(y_1|x_0)]/E_{p(x_1)}[p^2(y_1|x_1)]$ for $\sigma^2 = 1$, $\tau^2 \in \{0.01, 1, 10\}$ (panels (a) through (c), respectively), $C_0 = 1$ (panel (d)), $x_0 \sim N(0, C_0)$, $(y_1|x_1) \sim N(x_1, \sigma^2)$, $(y_1|x_0) \sim N(x_0, \sigma^2 + \tau^2)$, $y_1 \sim N(0, \sigma^2 + \tau^2 + C_0)$ and $x_1 \sim N(0, \tau^2 + C_0)$. It is easy to shown that

$$E_{p(x_0)}[p^2(y_1|x_0)] = [2\pi(\sigma^2 + \tau^2)(2C_0 + \sigma^2 + \tau^2)]^{-1} \exp\{-y_1^2/(2C_0 + \sigma^2 + \tau^2)\}$$

and

$$E_{p(x_1)}[p^2(y_1|x_1)] = [2\pi\sigma^2(2C_0 + \sigma^2 + 2\tau^2)]^{-1} \exp\{-y_1^2/(2C_0 + \sigma^2 + 2\tau^2)\}.$$

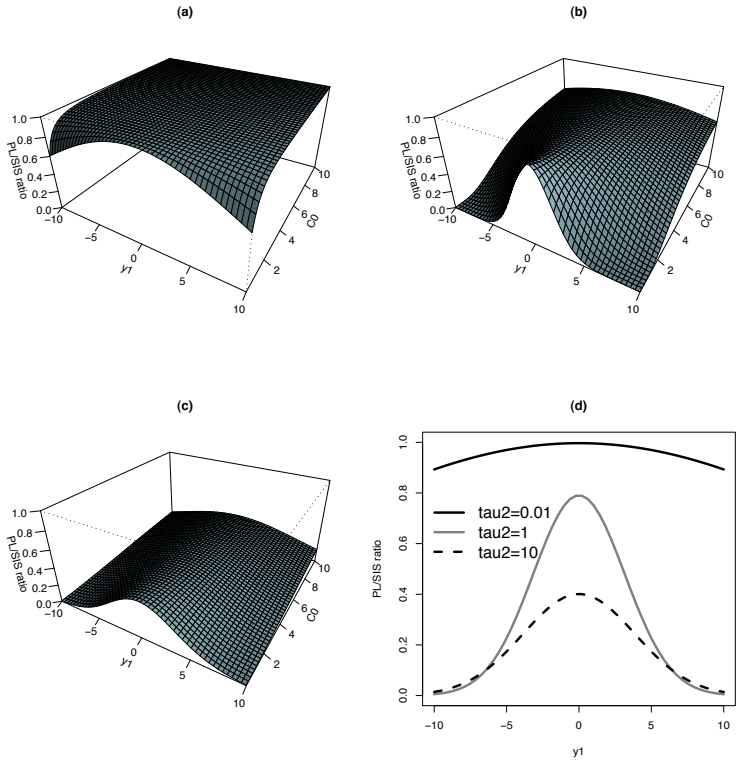

Figure 14: *Example F*

ADDITIONAL REFERENCES IN THE DISCUSSION

Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *J. Roy. Statist. Soc. B* **64**, 827–836.

Andrieu, C., Doucet, A. and Holenstein, R. (2010). Particle Markov Chain Monte Carlo *J. Roy. Statist. Soc. B* (to appear).

Berkhof, J., I. van Mechelen, and Gelman, A. (2003). A Bayesian approach to the selection and testing of mixture models. *Statist. Science* **13**, 423–442.

Besag, J. (1989). A candidate's formula: a curious result in Bayesian prediction. *Biometrika* **78**, 183–183.

Cappé, O., Moulines, E. and T. Ryden, T. (2004). *Hidden Markov Models*. New York: Springer

Carter, C. and Kohn, R. (1996). Markov Chain Monte Carlo in conditionally Gaussian state space models. *Biometrika* **83**, 589–601.

Chen, M., Shao, Q. and Ibrahim, J. (2000). *Monte Carlo Methods in Bayesian Computation*. New York: Springer

Chib, S. (1995). Marginal likelihood from the Gibbs output. *J. Amer. Statist. Assoc.* **90**, 1313–1321.

Chopin, N. and Robert. C. (2010). Properties of nested sampling. *Biometrika* **97**, 741–755

Darmois, G. (1935). Sur les lois de probabilité à estimation exhaustive. *Comptes Rendus Acad. Sciences Paris* **200**, 126–1266.

del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers. *J. Roy. Statist. Soc. B* **68**, 41–436.

Diebolt, J. and Robert, C. (1990). Estimation des paramètres d'un mélange par échantillonnage bayésien. *Notes aux Comptes–Rendus de Acad. Sciences Paris* **311**, 653–658.

Diebolt, J. and C. Robert. (1994). Estimation of finite mixture distributions by Bayesian sampling. *J. Roy. Statist. Soc. B* **56**, 363–375.

Douc, R., Cappé, O., Moulines, E. and Robert, C. (2002). On the convergence of the Monte Carlo maximum likelihood method for latent variable models. *Scandinavian J. Statist.***29**, 61–636.

Fearnhead, P. (2004). Particle filters for mixture models with an unknown number of components. *Statistics and Computing* **14**, 11–21.

Fearnhead, P. (2008). Computational methods for complex stochastic systems: A review of some alternatives to MCMC. *Statistics and Computing* **18**, 151–171.

Kantas, N., Doucet, A., Singh, S. S. and Maciejowski, J. M. (2009) An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. *15th IFAC Symposium on System Identification*, Saint-Malo, France.

Kim, S., Shepherd, N. and Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with arch models. *Review of Economic Studies* **65**, 361–394.

Lee, A., Yau, C., Giles, M. B., Doucet, A. and Holmes, C. C. (2009). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *arXiv* 0905.2441v3.

MacEachern, S. N., Clyde, M. A. and Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *Can. J. Statist.* **27**, 251–267.

Marin, J.-M. and Robert, C. P. (2010). Importance sampling methods for Bayesian discrimination between embedded models. *Frontiers of Statistical Decision Making and Bayesian Anaysis. In Honor of James O. Berger* (M.-H. Chen, D. K. Dey, P. Müller, D. Sun and K. Ye, eds.) New York: Springer, 513–527.

Neal, R. M. (1999). Erroneous results in "Marginal likelihood from the Gibbs output", *Unpublished letter*. `http://www.cs.toronto.edu/ radford/ftp/chib-letter.pdf`. University of Toronto.

Polson, N. G. (1996). Convergence of Markov Chain Monte Carlo algorithms. *Bayesian Statistics 5* (J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith, eds.) Oxford: University Press, 297–321 (with discussion).

Polson, N. G. (2007). Discussion of Raftery *et al.* (2007). *Bayesian Statistics 8* (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West, eds.) Oxford: University Press, 401–403.

Raftery, A. E., Newton, M. A., Satagopan, J. M. and Krivitsky, P. N. (2007). Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. *Bayesian Statistics 8* (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West, eds.) Oxford: University Press, 371–416 (with discussion).

Rubin, D. (1988). Using the SIR algorithm to simulate posterior distributions. *Bayesian Statistics 3* (J. M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith, eds.) Oxford: University Press, 395–402 (with discussion).