BAYESIAN STATISTICS 9,
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 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, The University of Chicago Booth School of Business. 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.

1. THE PL FRAMEWORK

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

- posterior samples in a direct approximations of marginal likelihoods;
- parallel environment, an important feature as more multi-processor computational power becomes available.

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

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

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

This follows from the Rao-Blackwellised identity,

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

If we require samples, we draw $\theta^{(i)} \sim p(\theta \,|\, Z_t^{(i)})$. 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 $\{Z_t^{(i)}\}_{i=1}^N \sim p(Z_t \,|\, y^t)$ sequentially from the set of posteriors. To this end, PL exploits the following sequential decomposition of Bayes' rule

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

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

The intuition is as follows. Given $\mathbb{P}(Z_t \,|\, y^t)$ we find the smoothed distribution $\mathbb{P}(Z_t \,|\, y^{t+1})$ via resampling and then propagate forward using $p(Z_{t+1} \,|\, Z_t, y_{t+1})$ 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 $\{Z_t\}_{i=1}^N$ to $\{Z_{t+1}\}_{i=1}^N$ in 2 steps:

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

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

The ingredients of particle learning are the essential state vector Z_t, a predictive probability rule $p(y_{t+1} \,|\, Z_t^{(i)})$ 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

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

Step 2. (Propagate)

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

Step 3. (Learn)

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

Example 1 (Constructing Z_n 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

$$\begin{aligned} y_i \,|\, \lambda_i &\sim N(\mu, \tau^2 \lambda_i) \\ \lambda_i &\sim IG(\nu/2, \nu/2) \end{aligned}$$

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

Here the essential state vector is $Z_n = (\lambda_{n+1}, a_n, b_n, m_n, C_n)$ where (a_n, b_n) index the sufficient statistics for the updating of τ^2, while (m_n, C_n) index the sufficient statistics for the updating of μ. Set $m_0 = 0$ and $C_0 = 1$. The sequence of variables λ_{n+1} are i.i.d. and so can be propagated directly from $p(\lambda_{n+1})$, whilst the conditional sufficient statistics (a_{n+1}, b_{n+1}) are deterministically calculated based on previous values (a_n, b_n) and parameters $(\mu_{n+1}, \lambda_{n+1})$. Here μ_{n+1} simply denotes draws for the parameter μ at time $n + 1$. Given the particle set $\{(Z_0, \mu, \tau^2)^{(i)}\}_{i=1}^N$, PL cycles through the following steps:

Step 1. Resample $\{(\tilde{Z}_n, \tilde{\mu}, \tilde{\tau}^2)^{(i)}\}_{i=1}^N$ from $\{(Z_n, \mu, \tau^2)^{(i)}\}_{i=1}^N$ with weights

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

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

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

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

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 through a Gibbs sampler based on the full conditionals:

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

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

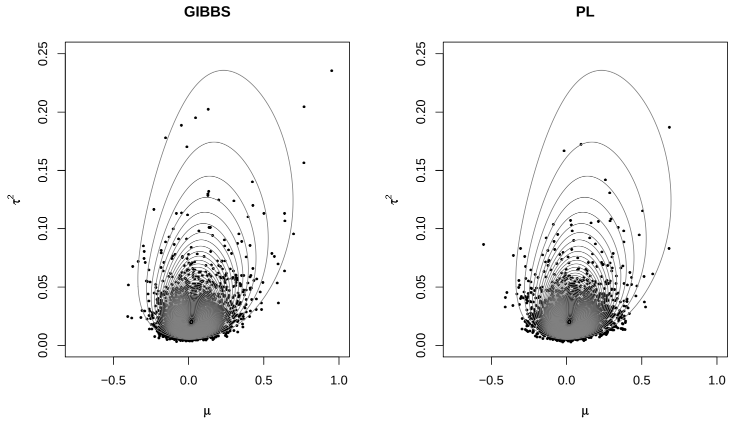

Figure 1: i.i.d. model. *Gibbs versus Particle Learning. Data* $y = (-15, -10, 0, 1, 2)$*, number of degrees of freedom* $\nu = 1$*, and hyperparameters* $a_0 = 5$*,* $b_0 = 0.05$*,* $m_0 = 0$ *and* $C_0 = 1$*. For the Gibbs sampler, the initial value for* τ^2 *is* $V(y) = 58.3$ *and 5000 draws after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

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 $\theta_{(-i)}$, 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 nonparametric 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 set ups (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 our hope that dissimination 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:

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

- *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;

- *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, the particle vector can grow in time as there's 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 $\theta^{(i)} \sim p(\theta \mid Z_n^{(i)})$ is not present in the APF and is used in PL to replenish the diversity of the parameter particles.

Storvik (2002) proposed the use 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 the approximation of 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 (2010) for a recent review of particle filter methods with an emphasis on contrasting propagate-resample and resample-propagate filters.

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

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

In the case where k_t is 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.

In other models, where k_t has a Markovian evolution (as in state space models) the smoothing distribution can be expressed by

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

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 by using the transition equation of k_t as 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 can also provide estimates of the predictive distribution $p(y_{n+1} \mid y^n)$ and marginal likelihood $p(y^n)$ 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 $p(y_{n+1} \mid y^n)$. Specifically, given the current particle draws, we have

$$p^N(y_{n+1} \mid y^n) = \sum_{i=1}^{N} p(y_{n+1} \mid Z_n^{(i)}) \quad \text{and} \quad p^N(y^n) = \prod_{i=1}^{n} p^N(y_i \mid y^{i-1}).$$

Therefore we simplify the problem of calculating $p\left(y^n\right)$ by estimating a sequence of small 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, the 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 worse 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 time (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 $\lambda^n = (\lambda_1, \ldots, \lambda_n)$ is a data augmentation variable. For this model, with a conditionally conjugate prior, we can find a conditional sufficient statistic, s_n, for parameter learning. Therefore, we define our sufficient state vector as $Z_n = (\lambda_n, s_n)$. Under these assumptions, we can write

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

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

$$\begin{array}{rcl} \lambda_{n+1} & \sim & p(\lambda_{n+1} \mid \lambda_n, \theta, y_{n+1}) \\ s_{n+1} & = & \mathcal{S}\left(s_n, \lambda_{n+1}, y_{n+1}\right). \end{array}$$

More complicated mixture models appear in Section 2.3.

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

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

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

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

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

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

v) Let $Z_{n+1}^{(i)} = (\lambda_{n+1}^{(i)}, a_n^{(i)}, b_n^{(i)})$ and update $(Z_{n+1}, \tau)^{(i)}$.

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 we match the variances of the parameter θ_i. 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 $p(y_{n+1} \mid y^n, \text{lasso})$ via $p(y_{n+1} \mid y^n, \text{lasso}) = N^{-1}\sum_{i=1}^N p(y_{n+1} \mid (\lambda_n, \tau)^{(i)})$ with predictive $p(y_{n+1} \mid \lambda_n, \tau) \sim N(0, \tau^2\lambda_n + 1)$. This leads to a sequential Bayes factor $B_{n+1} = p(y^{n+1} \mid \text{lasso})/p(y^{n+1} \mid \text{normal})$.

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 $p(\theta \mid Z_n)$ 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_i 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_n.

2.2. *Conditional Dynamic Linear Models*

We now explicitly derive our PL algorithm in a class of conditional dynamic linear models 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 that embeds many of the commonly used dynamic models. MCMC via Forward-filtering Backwards-sampling 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.

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

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

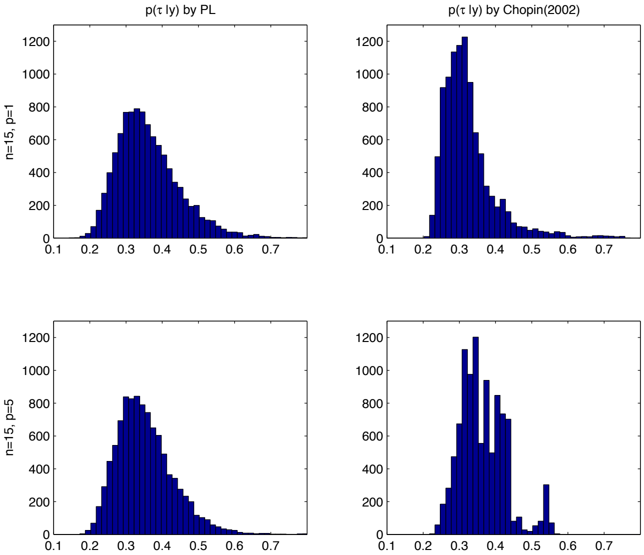

Figure 2: Bayesian Lasso. *Comparison to Chopin's (2002) batch sampling scheme.*

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 in the dynamic factor model example below.

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^x_t are the conditional sufficient statistics for parameters and states respectively. Here the conditional sufficient statistics for states (s^x_t) and parameters

(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 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 $m_{t+1} = G_{\lambda_{t+1}} m_t + A_{t+1}\left(y_{t+1} - e_t\right)$ and $C_{t+1}^{-1} = R_{t+1}^{-1} + F'_{\lambda_{t+1}} F_{\lambda_{t+1}} V_{\lambda_{t+1}}^{-1}$, with Kalman gain matrix $A_{t+1} = R_{t+1} F_{\lambda_{t+1}} Q_{t+1}^{-1}$, $e_t = F_{\lambda_{t+1}} G_{\lambda_{t+1}} m_t$, and $Q_{t+1} = F_{\lambda_{t+1}} R_{t+1} F_{\lambda_{t+1}} + V_{\lambda_{t+1}}$.

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 $p(x_{t+1} \mid \lambda_{t+1}, x_t, \theta, y_{t+1})$. Finally the conditional sufficient statistics are updated according to (1) and (1) and new samples for θ are obtained from $p(\theta \mid s_{t+1})$. Notice that in the conditional dynamic linear models all the above densities are available for evaluation and 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} form 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, namely the smoothing distribution $p(x_1, \ldots, x_T \mid y^T)$.

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 $\{1, 2\}$. Specifically, we have

$$\begin{aligned} (y_{t+1} \mid x_{t+1}, \lambda_{t+1}, \theta) &\sim N(\beta_{t+1} x_{t+1}, \sigma^2 I_2) \\ (x_{t+1} \mid x_t, \lambda_{t+1}, \theta) &\sim N(x_t, \sigma_x^2) \end{aligned}$$

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 $(\lambda_{t+1} \mid \lambda_t, \theta)$, whose transition matrix Π has diagonal elements $Pr(\lambda_{t+1} = 1 \mid \lambda_t = 1, \theta) = p$ and

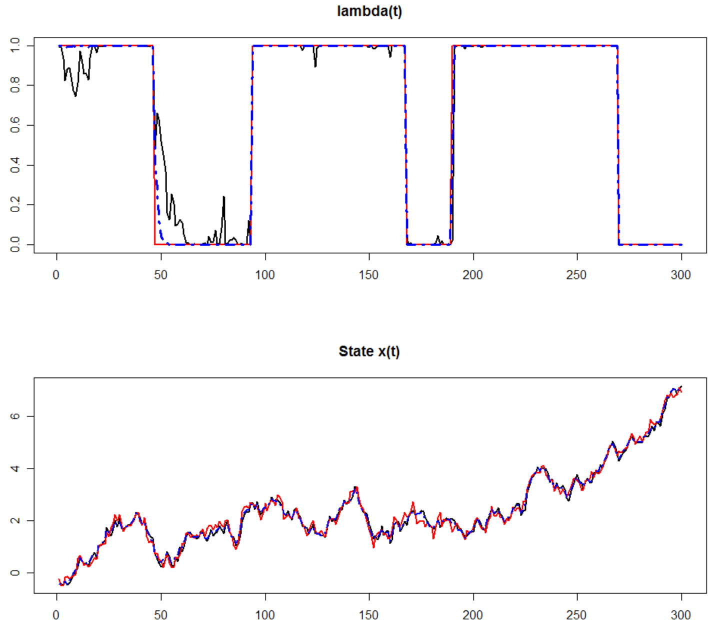

Figure 3: Dynamic factor model - state filtering. *Top panel: True value of* λ_t *(red line),* $Pr(\lambda_t = 1 \mid y^t)$ *(black line) and* $Pr(\lambda_t = 1 \mid y^T)$ *(blue line) Bottom panel: True value of* x_t *(red line),* $E(x_t \mid y^t)$ *(black line) and* $E(x_t \mid y^T)$ *(blue line).*

$Pr(\lambda_{t+1} = 2 \mid \lambda_t = 2, \theta) = q$. The parameters are $\theta = (\beta_1, \beta_2, \sigma^2, \tau^2, p, q)'$. See Carvalho and Lopes (2007) for related Markov switching models.

We are able to marginalizing 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 $(x_t \mid \lambda^t, \theta, y^t) \sim N(m_t, C_t)$. 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 $(\beta_i \mid \sigma^2) \sim N(b_{i0}, \sigma^2 B_{i0})$ for $i = 1, 2$, $\sigma^2 \sim IG(\nu_{00}/2, d_{00}/2)$ and $\tau^2 \sim IG(\nu_{10}/2, d_{10}/2)$. 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 $\{(x_t, \theta, \lambda_t, s_t^x, s_t)^{(i)}, i = 1, \ldots, N\}$ 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 $p(\lambda_{t+1} \mid (s_t^x, \lambda_t, \theta)^{(k^i)}, y_{t+1})$, so the density

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

(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 $(\beta_i \mid \sigma^2, s_{t+1}) \sim N(b_{i,t+1}, \sigma^2 B_{i,t+1})$, $i = 1, 2$, $(\sigma^2 \mid s_{t+1}) \sim IG(\nu_{0,t+1}/2, d_{0,t+1}/2t)$, $(\tau^2 \mid s_{t+1}) \sim IG(\nu_{1,t+1}/2, d_{1,t+1}/2)$, $(p \mid s_{t+1}) \sim Beta(p_{1,t+1}, p_{2,t+1})$, $(q \mid s_{t+1}) \sim Beta(q_{1,t+1}, q_{2,t+1})$ 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 illustrates 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. $\epsilon_{t+1}^z \sim -\log \mathcal{E}(1)$, with $\mathcal{E}(1)$ denoting 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:

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

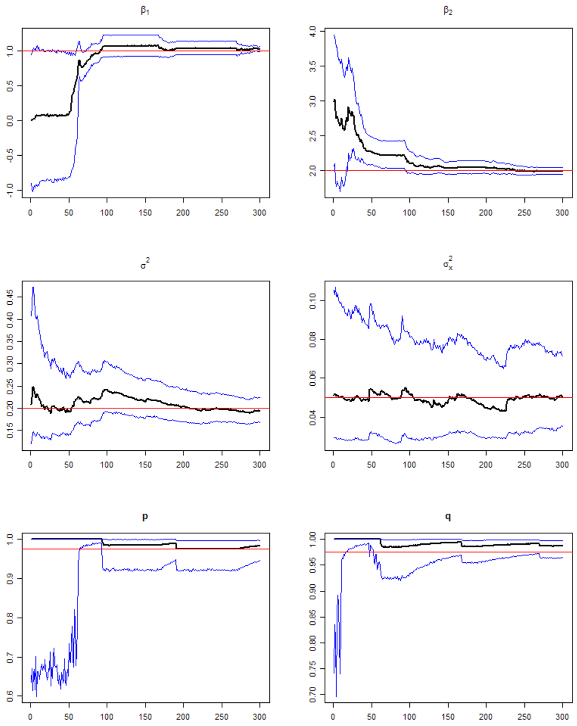

Figure 4: Dynamic factor model - parameter learning. *Sequential posterior median (black line) and posterior 95% credibility intervals (blue lines) for model parameters β_1, β_2, σ^2, τ^2, p and q. True values are the red lines.*

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 involves 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 re-sampling 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 $n_t = (n_{t,1}, \ldots, n_{t,m_t})$, the number of observations allocated to each unique component, and $s_t = (s_{t,1}, \ldots, s_{t,m_t})$, the conditional sufficient statistics for the component parameters. The state vector to be tracked by PL can then be defined as $Z_t = (k_t, m_t, s_t, n_t)$.

By definition $\theta_t^\star = \{\theta_1^\star, \ldots, \theta_{m_t}^\star\}$ is the set of m_t distinct components in θ^t, k^t is the associated latent allocation such that $\theta_t = \theta_{k_t}^\star$, $n_t = (n_{t,1}, \ldots, n_{t,m_t})$ is the number of observations allocated to each unique component, and $\boldsymbol{s}_t = (s_{t,1}, \ldots, s_{t,m_t})$ is the set of conditional sufficient statistics for each $\theta_j^\star$. Once again, we can define the state vector as $Z_t = (k_t, m_t, s_t, n_t)$. PL will not directly provide the full joint posterior of the allocation vector $k^t = (k_1, \ldots, k_t)$. If this is required either a particle smoothing or an MCMC step is required.

For infinite mixture models particle learning 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 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

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

and $v_l \overset{iid}{\sim} \text{beta}(1, \alpha)$, 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 $\{v_k : l = 1, 2, ...\}$ 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 $r = 1, \ldots, t$, $y_r \overset{ind}{\sim} k(y_r; \theta_r)$, $\theta_r \overset{iid}{\sim} G$ 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: 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 $\mathbb{E}[\; dG \mid \theta^t \;] = \int dG(\theta) dp(G \,|\, \theta_t^\star, n_t)$, and provides a finite predictive probability function for our mixture model: $p(y_{t+1} \,|\, \theta^t) = \int k(y_{t+1}; \theta) dG_0^t(\theta)$.

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

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

and $p\left(\mathbb{E}[f(y;G)] \,|\, s_t, n_t, m_t\right) = \int p(y \,|\, \theta_t^\star, n_t) dp(\theta_t^\star \,|\, s_t, n_t, m_t)$. 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 $\zeta \sim \text{Multinomial}(\omega, N)$ where

    $\omega^{(i)} = \frac{p(y_{t+1} \,|\, (s_t, n_t, m_t)^{(i)})}{\sum_{i=1}^N p(y_{t+1} \,|\, (s_t, n_t, m_t)^{(i)})};$

Step 2. (Propagate)

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

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

Example 5 (The DP mixture of multivariate normals). In the particular case of the d-dimensional DP multivariate normal mixture (DP-MVN) model has density function

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

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 deterministically $n_1 = 1$ and $s_1 = \{y_1, 0\}$, such that the algorithm is populated with N identical particles. Conditional on existing particles

$\{(n_t, s_t)^i\}_{i=1}^N$, uncertainty is updated through the familiar resample/propagate approach. The resampling step is performed by an application of the predictive probability function

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

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, for $j = 1, \ldots, m_t$,

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

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 $S_{t+1,j} = S_{t,j} + y_{t+1}y'_{t+1} + n_{t,j}\bar{y}_{t,j}\bar{y}^{i\prime}_{t,j} - n_{t+1,j}\bar{y}_{t+1,j}\bar{y}^{i\prime}_{t+1,j}$. The remaining sufficient statistics are the same as at 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,

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

The parameter updates are then

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

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 $\mathrm{p}(y \,|\, s_T, n_T, m_T)$; hence, the posterior expectation for $f(y; G)$. Marginal estimates are just the appropriate marginal density derived from mixture of Student's t distributions.

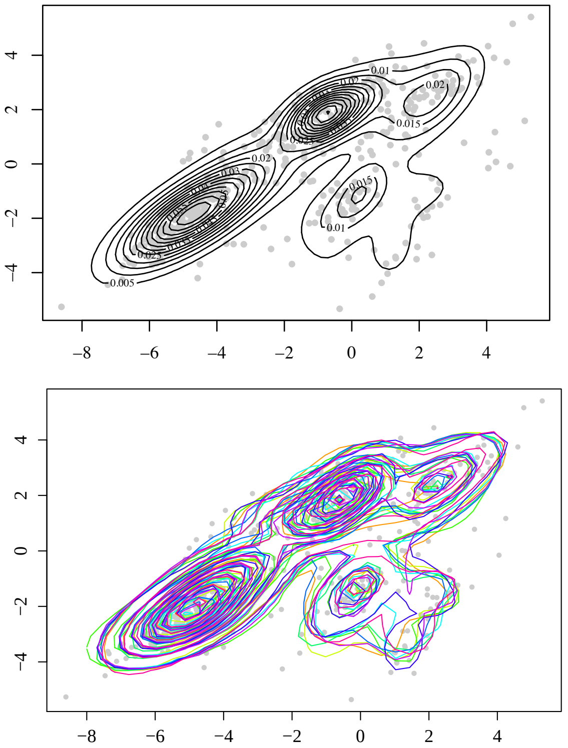

Figure 5: DP mixture of multivariate normals. *Data and density estimates for PL fit with 1000 particles (left) and each of ten PL fits with 500 particles (right), to a random ordering of the 1000 observations of bivariate data.*

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 (2009), 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 it is the resample-sample particle filter introduced by Pitt and Shephard (1999) 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 (2010).

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:

- Nonlinear and nonnormal panels (econometrics);
- REML (econometrics);
- Structural equations model;
- Dynamic factor models;
- Multivariate extensions (challenging);
- Space-time models (ensemble Kalman filters).

Finally, we emphasis 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* (to appear).

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. *Working Paper.* University of Chicago Booth School of Business.

Carvalho, C. M., Lopes, H. F., Polson, N. G. and Taddy, M. (2009). Particle learning for general mixtures. *Working Paper.* University of Chicago Booth School of Business.

Chen, H., Petralia, F. and Lopes, H. F. (2010). Sequential Monte Carlo estimation of DSGE models. *Working Paper.* The University of Chicago Booth School of Business.

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.

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. *Working Paper.* The University of Chicago Booth School of Business.

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 edition). 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. *Working Paper.* The University of Chicago Booth School of Business.

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 parameters and state estimation in simulation based filtering. In *Sequential Monte Carlo Methods in Practice* (A. Doucet, N. de Freitas and N. Gordon, eds.), 197–223. Berlin: Springer.

Lopes, H. F. (2000). *Bayesian analysis in latent factor and longitudinal models.* Unpublished Ph.D. Thesis. Institute of Statistics and Decision Sciences, Duke University, USA.

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

Lopes, H. F. and Tsay, R. S. (2010). Bayesian analysis of financial time series via particle filters. *J. Forecast.* (to appear).

Lund, B. and Lopes, H. F. (2010). Learning in a regime switching macro-finance model for the term structure. *Working Paper.* The University of Chicago Booth School of Business.

Mukherjee, C. and West, M. (2009). Sequential Monte Carlo in Model Comparison: Example in Cellular Dynamics in Systems Biology. *Working paper.* Department of Statistical Science, Duke University.

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., J. Pitman, and M. Yor (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. *Working paper.* Department of Economics, 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. In *Proceedings of the IEEE International Conference 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. *Working Paper.* The University of Chicago Booth School of Business.

Prado, R. and West, M. (2010). Time Series: Modelling, Computation and Inference. Boca Raton: Chapman & Hall/CRC Press.

Rios, M. P. and Lopes, H. F. (2009). Sequential parameter estimation in stochastic volatility models. *Working Paper.* The University of Chicago Booth School of Business.

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. *Working paper.* Department of Statistical Science, Duke University.

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. *Working Paper.* The University of Chicago Booth School of Business.

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

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 Edition). Berlin: Springer.