Biometrika (2002), **89,** 3, *pp.* 539–551

Printed in Great Britain

A sequential particle filter method for static models

BY NICOLAS CHOPIN

Laboratoire de Statistique, Centre de Recherche en Economie et Statistique, Ecole Nationale de la Statistique et de l'Administration Economique, Timbre J120, 75675 Paris cedex 14, France

chopin@ensae.fr

SUMMARY

Particle filter methods are complex inference procedures, which combine importance sampling and Monte Carlo schemes in order to explore consistently a sequence of multiple distributions of interest. We show that such methods can also offer an efficient estimation tool in 'static' set-ups, in which case $\pi(\theta | y_1, \ldots, y_N)$ $(n < N)$ is the only posterior distribution of interest but the preliminary exploration of partial posteriors $\pi(\theta | y_1, \ldots, y_n)$ makes it possible to save computing time. A complete algorithm is proposed for independent or Markov models. Our method is shown to challenge other common estimation procedures in terms of robustness and execution time, especially when the sample size is important. Two classes of examples, mixture models and discrete generalised linear models, are discussed and illustrated by numerical results.

Some key words: Batch importance sampling; Generalised linear model; Importance sampling; Markov chain Monte Carlo; Metropolis–Hastings; Mixture model; Parallel processing; Particle filter.

1. INTRODUCTION

Markov chain Monte Carlo methods are now a common tool for Bayesian inference. Unfortunately, in dynamic set-ups, when a sequence of posterior distributions π_t is involved, Markov chain Monte Carlo techniques may be unusable as they need to generate a different chain run for each posterior π_t, and do not take into account the previous generations from π_{t-1}. Instead, some authors have been developing more efficient methods based on importance sampling iterative strategies. In this context, an inference about π_{t-1} can be easily used to draw an inference on π_t by a 'reweighting' operation. These methods are usually referred to as 'particle filter methods' (Doucet et al., 2001, Ch. 1).

The purpose of this paper is to show that such methods can also improve estimation in static scenarios, when a single posterior density $\pi(\theta | y_1, \ldots, y_N)$ is involved. For instance, common simulation procedures are not practicable on huge datasets, because they consist of numerous iterations, each of which requires the whole sample to be processed. We propose an alternative structure, which provides significant computational savings by performing preliminary explorations of partial distributions $\pi(\theta | y_1, \ldots, y_n)$ $(n < N)$: an inference is first drawn from the n first observations, and this inference is then updated through importance sampling to incorporate the following p observations. This strategy can be iterated several times to provide finally inference based on the whole sample. We create artificial dynamics that correspond to the way that the data are processed

recursively, in batches: a particle filter is applied to a sequence of partial posterior distributions, $\pi_t(\theta) = \pi(\theta \mid y_1, \ldots, y_{n_t})$, with $n_1 < \ldots < n_t < \ldots < n_T = N$.

From now on, the observations y_n are assumed to be drawn from a parametric family $\{\mathscr{P}_\theta;\ \theta \in \Theta\}$, where Θ is an open subset of $\mathbb{R}^K$ ($K \geqslant 1$). The notation $y_{n:n+p}$ will refer to the sequence of observations $y_n, \ldots, y_{n+p}$. A partial posterior distribution $\pi(\theta \mid y_{1:n})$ will be denoted by π_n, so that π_N will stand for the 'complete' posterior distribution.

Section 2 recalls the main properties of particle filter methods. Section 3 presents the batch importance sampling scheme, which consists of importance sampling applied to partial posteriors π_n. Section 4 details the iterated batch importance sampling algorithm, a particle filter designed for estimating expectations $E_{\pi_N}\{h(\theta)\}$, in cases where the observations are either independent or Markov. This algorithm is a special case of the resample-move particle filter of Gilks & Berzuini (2001). Section 5 describes application to generalised linear models and mixture models.

2. PARTICLE FILTERS

2·1. *Importance sampling*

A particle system is a sequence (θ_j, w_j) of weighted random variables in Θ, θ_j being a particle with weight w_j, which targets a distribution of interest π over Θ in the sense that

$$\lim_{H \to +\infty} \frac{\sum_{j=1}^H w_j h(\theta_j)}{\sum_{j=1}^H w_j} = E_\pi\{h(\theta)\},$$

almost surely, for any measurable h such that $E_\pi\{h(\theta)\}$ exists.

When the particles are drawn independently from a distribution g, weights proportional to $\pi(\theta_j)/g(\theta_j)$ provide a particle system with target π: this is importance sampling. The ratio $\pi(\theta_j)/g(\theta_j)$ is often known only up to a multiplicative constant, which is cancelled by the denominator $\sum_{j=1}^H w_j$. The resulting estimator $\hat{\mu}(h) = \sum_{j=1}^H w_j h(\theta_j)/\sum_{j=1}^H w_j$ is biased but consistent, and we have

$$H^{\frac{1}{2}}\{\hat{\mu}(h) - E_\pi(h)\} \to \mathscr{N}\{0, V_{\pi/g}(h)\},$$

in distribution, where $V_{\pi/g}(h) = \operatorname{var}_g[\{\pi(\theta)/g(\theta)\}\{h(\theta) - E_\pi(h)\}]$.

Thus, it is sensible to consider the quantity $V_{\pi/g}(h)$ as a measure of the efficiency of importance sampling based on the instrumental g. This quantity can be 'normalised' by dividing by $\operatorname{var}_\pi\{h(\theta)\}$, in order to remove the variability caused by the studied phenomenon itself. We define

$$\tau_{\pi/g}(h) = V_{\pi/g}(h)[\operatorname{var}_\pi\{h(\theta)\}]^{-1}.$$

This normalised measure of efficiency is directly related to the concept of effective sample size of Carpenter et al. (1999), which is the sample size required to attain the same precision as with particles drawn directly from the target distribution π: the effective sample size for a function h is equal to $H\tau_{\pi/g}(h)^{-1}$ in an univariate setting.

We will also use the following properties, as sufficient finiteness conditions for $V_{\pi/g}(h)$

and $\tau_{\pi/g}(h)$ (Geweke, 1989):

$$\sup_{\theta \in \Theta} \left(\frac{\pi(\theta)}{g(\theta)} \right) < +\infty, \quad E_g\{h(\theta)h(\theta)'\} < +\infty.$$

2·2. *Iterated applications of importance sampling*

A major advantage of particle systems is their flexibility: a simple reweighting operation $w'_j = w_j \pi_2(\theta_j)/\pi_1(\theta_j)$ shifts the target of a particle system from π_1 to π_2 (Liu & Chen, 1998; Gilks & Berzuini, 2001). The ratio $\pi_2(\theta_j)/\pi_1(\theta_j)$ is called the incremental weight.

When the distribution of interest π_t is evolving through time, this reweighting scheme can be iterated for each π_t. Unfortunately, each reweighting phase introduces a bit more variability in the estimates, as π_t moves away from π_1: fewer and fewer particles retain significant weights, and the mass of the considered distribution π_t can even leave the support of the distribution π_1 from which particles are initially drawn. The particle system suffers from progressive 'impoverishment' or 'degeneracy'.

The reweighting steps can be alternated with resampling steps in which each particle θ_j is replaced by a number n_j of its replicates; n_j may be zero. Each new particle is assigned unit weight. The n_j's can be determined in various ways, the most famous being multinomial selection (Gordon et al., 1993), in which a resampled particle $\theta^r_{j'}$ is drawn in such a way that

$$\mathrm{pr}(\theta^r_{j'} = \theta_j) = w_j \Big/ \sum w_j \quad (j = 1, \ldots, H).$$

This method keeps the estimators unbiased; note that the n_j's are random. More elaborate unbiased selection schemes include residual resampling (Liu & Chen, 1998) and stratified resampling (Carpenter et al., 1999). Liu & Chen (1998) and Doucet et al. (2001, Ch. 4) provide good reviews.

It must be stressed that resampling does not protect from degeneracy: it just saves further calculation time by getting rid of particles with insignificant weights. Moreover, resampling artificially conceals impoverishment, by replacing high weights with numerous replicates of an unique particle, thereby introducing high correlations between particles.

Gilks & Berzuini (2001) added a rejuvenation step to the resampling step. Resampled particles at stage t are then 'moved' according to a Markov chain transition kernel with stationary distribution π_t; that is $\theta^m_j \sim K_t(\theta^r_j, .)$. This operation does not change the system target, but it may strongly reduce impoverishment, since identical replicates of a single particle are replaced by new 'fresh' values. Efficiency of the rejuvenation step obviously relies on a sensible choice of K_t, while assessing the efficiency for a given kernel may be a difficult task in practice.

The term particle filter methods will refer to algorithms which provide consistent inferences from a sequence of distributions π_t, by iterating the following steps.

PARTICLE FILTER ALGORITHM

Step 1. *Reweighting*: *update the weights*, $w_j \leftarrow w_j \times \pi_{t+1}(\theta_j)/\pi_t(\theta_j)$, *for* $j = 1, \ldots, H$.

Step 2. *Resampling*: *resample* $(\theta_j, w_j)_{j=1,\ldots,H} \rightarrow (\theta^r_j, 1)_{j=1,\ldots,H}$, *according to a given selection scheme.*

Step 3. *Move*: *draw* $\theta^m_j \sim K_{t+1}(\theta^r_j, .)$ *for* $j = 1, \ldots, H$, *where* K_{t+1} *is a transition kernel with stationary distribution* π_{t+1}.

Step 4. *Loop*: $t \leftarrow t + 1$, $(\theta_j, w_j)_{j=1,\ldots,H} \leftarrow (\theta^m_j, 1)_{j=1,\ldots,H}$, *and return to Step* 1.

Recent results about the convergence of such algorithms, when $H \to \infty$, can be found in a Cambridge University technical report by D. Crisan and A. Doucet.

3. BATCH IMPORTANCE SAMPLING

Our aim is to estimate a fixed-dimension parameter $\theta_0 \in \Theta$, from N observations $y_1, \ldots, y_N$ drawn from the 'static' model $\mathscr{P}(\theta_0)$. The posterior distribution of interest is therefore $\pi(\theta \mid y_{1:N})$.

It is possible to endow this static model with a dynamical structure. Suppose only the first n observations ($n < N$) are immediately available. Inference can be based on simulations from the posterior $\pi(\theta \mid y_{1:n})$, or more generally on a particle system with target $\pi(\theta \mid y_{1:n})$. Then assume that p new observations are available. Provided p is not too large, $\pi(\theta \mid y_{1:n})$ and $\pi(\theta \mid y_{1:n+p})$ are likely to be similar, and hence we can use the results about $\pi(\theta \mid y_{1:n})$ in constructing a proper reweighting of the particles by the incremental weight

$$w_{n,p}(\theta) \propto \frac{\pi(\theta \mid y_{1:n+p})}{\pi(\theta \mid y_{1:n})} \propto \frac{p(y_{1:n+p} \mid \theta)}{p(y_{1:n} \mid \theta)} = p(y_{n+1:n+p} \mid y_{1:n}, \theta).$$

We refer to this particular case of importance sampling as batch importance sampling. We now describe important properties of $w_{n,p}(\theta)$.

First, if $p(y_{n+1:n+p} \mid y_{1:n}, \theta)$ is bounded from above in θ, clearly a weak condition, then $w_{n,p}(\theta)$ is also bounded, which ensures that $V_{\pi_{n+p}/\pi_n}(h)$ and $\tau_{\pi_{n+p}/\pi_n}(h)$ are finite, for any h such that $E_{\pi_n}\{h(\theta)h(\theta)'\}$ exists; see the end of § 2·1.

Moreover, the impoverishment of the particle system induced by batch importance sampling can be approximately evaluated by the following asymptotic result.

THEOREM 1. *Under some regularity conditions, listed in the Appendix, and for any thrice continuously differentiable function* $h : \Theta \to \mathbb{R}^L$, *for* $L \in \mathbb{N}$, *such that the integrals*

$$\int h(\theta)\pi_{n+p}(\theta)^2/\pi_n(\theta)\, d\theta, \quad \int h(\theta)h(\theta)'\pi_{n+p}(\theta)^2/\pi_n(\theta)\, d\theta$$

exist for all $n, p \in \mathbb{N}$, *we have that*

$$\|\tau_{\pi_{n+p}/\pi_n}(h)\| = O(1),$$

as $n \to \infty$, $p/n \to r > 0$, *where* $\|.\|$ *stands for the Euclidean norm on* $\mathbb{R}^L$.

A proof is given in an unpublished Centre de Recherche en Economie et Statistique technical report by the author.

We see that, when n is large enough, the relative precision of batch importance sampling depends only on the proportion of new data, and, given that proportion, any relative precision may be attained with a sufficiently large number H of particles, where H does not depend on n.

Finally, in two important cases, the reweighting step only operates on the new data $y_{n+1:n+p}$, thus avoiding a second complete browse of $y_{1:n}$; when the observations are independent, we have

$$w_{n,p}(\theta) \propto p(y_{n+1:n+p} \mid y_{1:n}, \theta) = \prod_{i=1}^{p} p(y_{n+i} \mid \theta),$$

and, when they are Markov of order m, that is there exists an integer m such that

$$p(y_{n+1} \mid y_{1:n}, \theta) = p(y_{n+1} \mid y_{n+1-m:n}, \theta),$$

we have

$$w_{n,p}(\theta) \propto p(y_{n+1:n+p} \mid y_{1:n}, \theta) = \prod_{i=1}^{p} p(y_{n+i} \mid y_{n+i-m:n+i-1}, \theta).$$

In both cases, the reweighting step makes possible a quick update of the particle system, provided the likelihood $p(y_{n+i} \mid \theta)$ or $p(y_{n+i} \mid y_{n+i-m:n+i-1}, \theta)$ is computable. In other cases, batch importance sampling is less interesting, given that the calculation of $p(y_{n+1:n+p} \mid y_{1:n}, \theta)$ may become too intensive as n grows. We will therefore assume in the sequel that the observations are either Markov or independent, and we will adopt a common notation for both cases, that is $p(y_{n+1} \mid y_{1:n}, \theta) = p(y_{n+1} \mid y_{n+1-m:n}, \theta)$, with $m = 0$ if the y_i's are independent, in which case $y_{n+1:n}$ will stand for $\varnothing$.

4. The iterated batch importance sampling algorithm

4·1. *Statement of the algorithm*

The iterated batch importance sampling algorithm is a particle filter method that iterates the following steps.

Iterated batch importance sampling algorithm

Step 0. *Initialisation*: *generate a particle system* $(\theta_j, w_j)_{j=1,\ldots,H}$ *which targets the initial distribution* π_{n_0}.

Step 1. *Reweighting*: *update the weights*, $w_j \leftarrow w_j \times w_{n,p}(\theta_j)$ *with*

$$w_{n,p}(\theta_j) \propto p(y_{n+1:n+p} \mid y_{1:n}, \theta_j) \quad (j = 1, \ldots, H).$$

Step 2. *Resampling*: *resample*, $(\theta_j, w_j)_{j=1,\ldots,H} \rightarrow (\theta_j^r, 1)_{j=1,\ldots,H}$, *according to a given selection scheme*; *see* § 2·2.

Step 3. *Move*: *draw* $\theta_j^m \sim K_{n+p}(\theta_j^r, .)$ *for* $j = 1, \ldots, H$, *where* K_{n+p} *is a transition kernel with stationary distribution* π_{n+p}.

Step 4. *Loop*: $n \leftarrow n + p$, $(\theta_j, w_j)_{j=1,\ldots,H} \leftarrow (\theta_j^m, 1)_{j=1,\ldots,H}$, *and return to Step* 1.

The algorithm stops when $n = N$, that is when the particle system targets the distribution of interest $\pi(\theta \mid y_{1:N})$. Sections 4·2–4·4 discuss components of the algorithm whose specification can influence the algorithm efficiency, both in terms of execution time and robustness of the resulting estimates. The aim is also to make the algorithm a true 'black box', that is an algorithm whose internal machinery is not model-dependent; the practitioner's task then reduces to supplying a likelihood-computation routine, and a correctly initialised particle system.

4·2. *Choice of the move kernel*

Efficiency in the move step is critical, since it is the most computationally demanding step, in requiring a complete browsing of the past observations $y_{1:n}$. Markov chain Monte Carlo techniques usually involve numerous applications of a transition kernel over a single 'particle'. In contrast, the move step of particle filter methods consists of a single application of a given kernel, for a large set of particles. Thus the choice of a kernel must rely on different criteria from those usually mentioned in Markov chain Monte Carlo settings: for instance, theoretical convergence does not ensure that the particles move efficiently in a single application of the kernel. Moreover it is not easy in practice to assess correctly how much the system really has been rejuvenated. For example, using a Metropolis–

Hastings kernel, one may assess the rejuvenation through the acceptance rate, regarding the 'accepted' particles as new particles introduced into the system, and acknowledging that the greater the number of new particles, the better. If the 'proposed' value is generated from a random walk, that is $\theta_j^p = \theta_j + \varepsilon_j$, high acceptance rates can be achieved artificially by using ε_j with a very low variance: identical replicates of a single particle will then be replaced by an equivalent number of particles taking distinct but very similar values, which are therefore mutually highly correlated. System impoverishment remains high but is no longer detectable.

It therefore seems sensible to select a transition kernel which depends weakly on the previous value of the moved particle. Such is the case with an independent Metropolis–Hastings kernel, in which the proposed particle is generated independently from an instrumental distribution g, and the moved particle only depends on its previous value through the acceptance probability. If we use such a kernel, the acceptance rate seems a far more reliable indicator of efficiency. Note that, when the rejuvenation is performed by such a kernel, a browse through the whole past subsample is needed in the computation of $\pi_{n+p}(\theta_j^p)$, which appears in the acceptance probability.

Provided we use such a kernel, high acceptance rates will be obtained with an instrumental distribution g close to the target distribution π_{n+p}. However, our only information about π_{n+p} is given by the particle system itself, which can provide at the current stage an estimate of any expectation $E_{\pi_{n+p}}\{h(\theta)\}$, say

$$\hat{\mu}(h) = \frac{\sum_{j=1}^{H} w_j h(\theta_j)}{\sum_{j=1}^{H} w_j}.$$

In particular, the estimates

$$\hat{E}_{n+p} = \frac{\sum_{j=1}^{H} w_j \theta_j}{\sum_{j=1}^{H} w_j}, \quad \hat{V}_{n+p} = \frac{\sum_{j=1}^{H} w_j \{\theta_j - \hat{E}_{n+p}\}\{\theta_j - \hat{E}_{n+p}\}'}{\sum_{j=1}^{H} w_j},$$

give a rough idea of the location of the mass of π_{n+p}. Thus the instrumental distribution $\mathcal{N}(\hat{E}_{n+p}, \hat{V}_{n+p})$ seems a reasonable choice: it is simple, in that, even when the dimension of Θ is important, we can still easily take correlations between components of θ into account; it is not 'model-dependent', and hence it fulfils our black-box requirement; and it is asymptotically justified, because, as $n \to \infty$, π_n tends towards a normal distribution. The last point is essential: as we have said, the rejuvenation step becomes more and more demanding as $n \to \infty$, but fortunately simultaneously becomes more and more efficient.

However, for finite n, a complex posterior distribution can differ significantly from a Gaussian distribution, possibly possessing local modes or thick tails. In the latter context, note that the posterior distribution tails can easily be reduced by a proper reparametrisation: if, for instance, the density of π_n decreases towards infinity like $\exp(-K\theta)$, for $\theta > 0$, we obtain thinner, Gaussian-like tails by replacing θ by $\theta' = \theta^{\frac{1}{2}}$. Also, reparameterising the model is likely to be less time-consuming than deriving a distinct instrumental distribution with tails of the same order as those of the considered distribution. This is another appeal of our black-box approach. For these reasons, we regard $\mathcal{N}(\hat{E}_{n+p}, \hat{V}_{n+p})$ as a convenient all-purpose instrumental distribution.

4·3. *Incorporation schedule*

The distribution π_{n+p} is expected to move progressively away from π_n as p increases, so that the impact of a reweighting step, from π_n to π_{n+p}, on system degeneracy should

increase with p. Since the reweighting can be operated recursively, by incorporating one observation after the other through the transitive operation

$$w_{n,n+k+1}(\theta) = w_{n,n+k}(\theta) w_{n+k,n+k+1}(\theta),$$

it is possible to increase p, the number of added observations, while checking regularly for degeneracy, and to stop this process when a given level of degeneracy is attained. A common indicator of degeneracy is the empirical variance of the weights (Liu & Chen, 1998).

We argue however that the current representation of the weights cannot express system degeneracy properly, since it does not take the level of (in)effectiveness of the previous move step into account. When a given particle θ_j gives birth to n_j replicates in the resampling step, the extent to which these replicates should be seen as independent particles after the move step is highly dependent on the efficiency of the chosen kernel. In particular, with an independent Metropolis–Hastings kernel, it is likely that a number n'_j of the n_j replicates, $0 \leqslant n'_j \leqslant n_j$, are not moved, but keep the same value θ_j. It is therefore more sensible to replace these n'_j identical particles by a single representative with weight n'_j: this leads to a new representation of the weights, which gives a better idea of the actual degeneracy. In particular, the empirical variance of these new weights may take considerably greater values than with the previous representation, if the previous move steps are weakly efficient. Furthermore, this new representation must also be adopted when implementing the algorithm in practice, since it avoids useless repetitions of the same computations for identical particles. We therefore propose to resample and move the particles whenever the empirical variance of these new weights exceeds a given threshold d.

Theorem 1 indicates that the level of degeneracy induced by the reweighting step is asymptotically given by the proportion p/n of new points. Thus, if we apply the same criterion at each stage, and assume a constant level of efficiency for the move steps, we expect the number of new points to increase geometrically. In fact, the move steps are likely to become more and more efficient as n increases, see § 4·2, so that the incorporation may be even faster.

4·4. *Number of particles and other calibrations*

Particle filter methods are only justified asymptotically, as $H \to +\infty$. It would therefore be tempting to run the iterated batch importance sampling algorithm with very large values of H, but this would imply a tremendous computational cost. In practice, we must determine a reasonable value of H, in computational terms, that still leads to robust estimates. One way is to run the algorithm several times to measure the variability of the results obtained for a given number of particles. In the independent case, the observations should be shuffled on each run, to check that the results do not depend on the order of incorporation of the observations. Note however that, in our numerical work, we never faced any case of degeneracy or high variability of estimates, provided H exceeded 10 000, even though we dealt with posterior distributions as complex as mixture distributions over a parameter space of high dimension; see § 5·2. Note also that, if the acceptance rate of the last move step is close to one, the final particles will be nearly independent realisations of π_N, and therefore the quantity $\mathrm{var}_{\pi_N}\{(h(\theta)\}/H$ will provide a good approximation of the variance of the estimator of $E_{\pi_N}\{(h(\theta)\}$ at the first execution.

In the independent case, the algorithm may perform very poorly if the order of incorporation of the observations is unfortunate. If the data are suspected of showing some partial ordering, we recommend randomising their labels before the algorithm is implemented.

Finally, note that the particle system may degenerate strongly in the very early stages, when the evolving target π_n changes the most, while the move steps do not yet show a high efficiency. In particular, if the initial particles are simulated from a flat prior, only a few of them will be in a region of interest for the following stages. In that case, it is preferable to initialise the algorithm with some partial posterior π_{n_0} $(n_0 > 0)$, based on a small number of the observations. This is also necessary when an improper prior is specified.

4·5. *Computational considerations and the issue of parallel processing*

The iterated batch importance sampling algorithm explores the parameter space Θ in a distinctive way. It tracks numerous particles while smoothly browsing the sample of observations $y_{1:N}$, whereas most estimation algorithms track a single particle through numerous iterations, each requiring a complete browsing of the dataset $y_{1:N}$; 'particle' here refers to a single parameter value varying at each iteration.

This strategy strongly limits the number of accesses to the dataset. In fact, our algorithm can roughly locate the region of interest by drawing information from a small part of the sample before exploring it more precisely, whereas its competitors would need to perform computations over the whole sample.

Moreover, the strategy facilitates parallel processing. If K processors are available, we can partition the particle system into K subsets S_k $(k = 1, \ldots, K)$, and implement computations for particles in S_k in processor k. This partitioning must occur during the reweighting step, when processor k computes the incremental weights of particles in S_k, and the move step, when the processor k 'moves' the particles in S_k. The resampling step is much less intensive, and does not really need this partitioning. After the reweighting and move steps, results must be gathered together before moving onto the next step. Since the algorithm can deal with new data at a nearly geometric rate, the frequency of move steps quickly decreases as n grows, and therefore the frequency of required information exchanges between processors also decreases at a rate that is exponential in n, making the parallel processing of the reweighting and move steps quickly highly efficient.

4·6. *Economies of scale*

The algorithm has to be supplied with a routine for computing the partial likelihood $p(y_{n+1} | y_{n+1-m:n}, \theta)$ for any value of $y_{n+1-m:n+1}$ and θ. This routine is called for each particle and each added observation during the reweighting step in order to compute the corresponding incremental weights and during the move step in order to compute the acceptance probabilities of the Metropolis–Hastings kernel. The computational feasibility of the algorithm therefore directly depends on the execution time of this routine. Although $p(y_{n+1} | y_{n+1-m:n}, \theta)$ may in some cases appear complex, particularly if it is expressed as an integral, economies of scale are often easily achievable, permitting x evaluations of $p(y_{n+1} | y_{n+1-m:n}, \theta)$ in less time than x times the time needed for one evaluation. For example, if the likelihood involves a computationally 'costly' function $\Psi : \mathbb{R} \to \mathbb{R}$, one can store preliminary evaluations of Ψ and its first derivatives $\Psi^{(l)}$ over a grid of points z_k, and then replace any further computation of $\Psi(z)$ by an appropriate Taylor approximation; see § 5·1. Also suppose that the likelihood involves an integral $\int \phi(z)\varphi_i(z)\, dz$, where the function φ_i differs for each evaluation i. This integral can be evaluated through a given set of evaluations of its integrand $\phi(.)\varphi_i(.)$. The $\phi(.)$ may be stored the first time they are computed, and re-used as often as necessary.

4·7. *Iterated batch importance sampling in a latent variable context*

Suppose the observation y_n is related to an unobserved, latent z_n, such that

$$y_n|\{z_n = z\} \sim \mathscr{P}(\theta_0; z).$$

For such models, one can use the Gibbs sampler or the EM algorithm, both of which explore the augmented space of parameters and latent variables. However, because of its much greater dimension, it may be more difficult to explore the augmented space efficiently. Moreover, since the intermediary movements along the latent variables' dimensions are specific to a given model, implementation of a Gibbs sampler, for instance, involves the practitioner in problem-specific programming, in contrast to the black-box nature of the iterated batch importance sampling algorithm. Finally, the necessary complete browsing of the sample at each iteration renders it extra-difficult to handle large databases by these methods even in an age of great computational power. In such cases our algorithm seems more to be recommended, provided that the marginal likelihood

$$p(y_n|\theta) = \int p(y_n, z_n|\theta)\, dz_n$$

can be computed.

5. Examples

5·1. *Generalised linear models*

Generalised linear models (McCullagh & Nelder, 1989) relate independent observations (y_i) to their covariates $(x_i) = (x_i^1, \ldots, x_i^K)$ through an exponential-family density of the form

$$p(y_i|\theta_i) = m(y_i) \exp\{y_i\theta_i - \psi(\theta_i)\}, \quad y_i \in \mathscr{Y},$$

where the canonical parameter θ_i is determined by the linear relation

$$h\{E_{\theta_i}(y_i)\} = h\{(\psi'(\theta_i)\} = x_i'\beta,$$

in which h is the link function. The row vectors of covariates x_i form a full-rank matrix X, and $\mathscr{Y}$ may be either discrete or continuous.

Application of the Gibbs sampler often appears natural, especially when a missing data structure can be exhibited, as in binary or polytomous regression, for instance. Unfortunately, direct simulation of these missing components is feasible only in some special cases, such as the probit model (Albert & Chib, 1993). For other models, the hybrid algorithm of Dellaportas & Smith (1993), relying on both Gibbs sampling and rejection techniques, can be used provided the considered posterior distribution is log-concave, which is often the case.

The iterated batch importance sampling algorithm can handle directly any generalised linear model for which $p(y|x, \beta)$ is computable. Moreover, as a result of the exponential nature of the density, such a model often leads to a very regular and unimodal posterior distribution, enabling the Gaussian instrumental distribution of the algorithm to move the particles efficiently in the early stages; see Fig. 1(a).

Consider for instance the probit model, in which

$$y|x, \beta \sim \text{Bi}\{1, \Phi(x'\beta)\}, \quad \mathscr{Y} = \{0, 1\}.$$

In this case, $p(y|x, \beta) = \Phi(x'\beta)^y \Phi(-x'\beta)^{1-y}$, and the algorithm must perform numerous

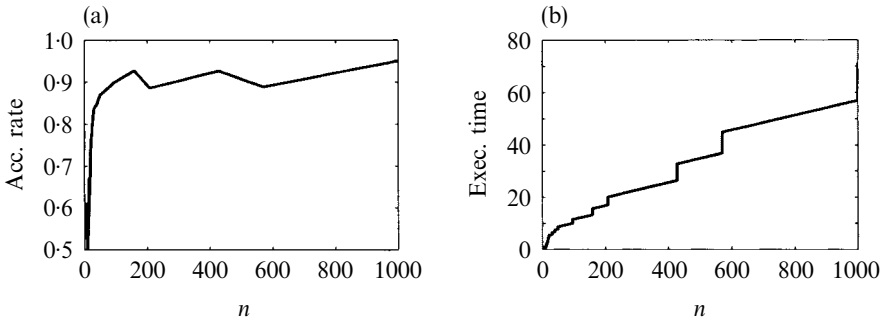

Fig. 1. (a) Acceptance rate against n, and (b) execution time, in seconds, against n, where n is the number of observations incorporated for simulated probit data.

evaluations of the function Φ. In order to reduce the execution time, we implement our Taylor approximation method, see § 4·6, for $\phi(z) = \log \Phi(z)$; it is better to compute ϕ instead of Φ, since the logarithm transforms products into sums.

This method approximates $\phi(z)$, for any $z \in [-5, 5]$, up to an absolute error of 10^{-6}, by performing only two sums and two products, with only a few initial evaluations of ϕ, ϕ' and ϕ'', over a grid of 500 points. Note that an absolute error in ϕ corresponds to a relative error of the same magnitude in Φ.

We considered a probit model with $K = 5$ covariates. The $N = 1000$ observations were simulated from the true model, for a given parameter β_0. Covariates were simulated from independent standard normal distributions, except for the first row, which was constant. A Gaussian prior distribution $\mathcal{N}(\mu_0, \sigma_0^2)$ was assigned to β, with $\mu_0 = 0$ and $\sigma_0 = 5$. Table 1 gives the resulting estimate, along with its mean squared error over 10 runs. We can see that with a moderately large number of particles, $H = 2000$, estimates are relatively robust. Each execution of the algorithm took approximatively one minute on a 500 MegaHertz desktop computer.

Table 1. *Results for simulated data from a probit model, with* $H = 2000$ *and* $N = 1000$. *Shown are the true value* β_0 *of the parameter, which is a vector of dimension* 5, *the estimate* $\hat{E}_{\pi_N}(\beta)$ *of the posterior mean provided by the algorithm and the mean squared error of this estimate over* 10 *runs*

	$\beta_0 = -1$	$\beta_0 = 0{\cdot}7$	$\beta_0 = -0{\cdot}5$	$\beta_0 = -0{\cdot}1$	$\beta_0 = -0{\cdot}3$
$\hat{E}_{\pi_N}(\beta)$	$-0{\cdot}94$	$0{\cdot}63$	$-0{\cdot}49$	$-0{\cdot}15$	$-0{\cdot}34$
MSE	10^{-6}	$7{\cdot}8 \times 10^{-6}$	4×10^{-6}	2×10^{-6}	4×10^{-6}

MSE, mean squared error.

Figure 1 shows how the acceptance rate and the execution time, in seconds, evolve with n, the amount of data already incorporated. As mentioned in § 4·2, the frequency of move steps decreases very quickly, the move steps corresponding to the vertical parts of the plot in Fig. 1(b), while their acceptance rates are already close to one at early stages. On the other hand, move steps become more and more intensive, i.e. the vertical segments get longer, since they involve computations over all the previously incorporated observations $y_{1:n}$.

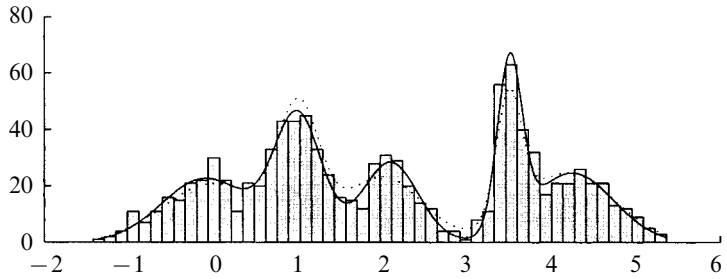

Fig. 2. Mixture model example. Histogram, true density $p(y|\theta_0)$ of the simulated observations, dotted line, and the density estimate $p(y|\hat{\theta})$, solid line, where $\hat{\theta}$ is the estimate produced by the algorithm, with $N = 1000$, $k = 5$ and $H = 10\,000$.

5·2. *Mixtures*

Mixture models take the following form:

$$z_i \sim \text{MN}(k;\, p_1, \ldots, p_k), \quad y_i|\{z_i = l\} \sim \mathscr{P}(\xi_l) \quad (i = 1, \ldots, N),$$

where the z_i's are unobserved and independent discrete variables, $\text{MN}(k;\, p_1, \ldots, p_k)$ stands for the k-cell multinomial distribution, with probabilities $p_1, \ldots, p_k$, and the $\mathscr{P}(\xi_l)$'s are distinct models belonging to some parametric family $\{\mathscr{P}(\xi), \xi \in \Xi\}$. The aim is to estimate $\theta = (p_1, \ldots, p_{k-1}, \xi_1, \ldots, \xi_k)$. In our example, we will take $\mathscr{P}(\xi_l) = \mathscr{N}(\mu_l, \sigma_l^2)$. A constraint on the parameters, such as $p_1 < \ldots < p_{k-1}$, $\mu_1 < \ldots < \mu_k$ or $\sigma_1 < \ldots < \sigma_k$, is needed to identify the model fully.

Since the latent variables z_i are discrete, the likelihood $p(y_{n+1:n+p}|y_{1:n}, \theta)$ is easily computable, as a sum over the possible states of the z_i. The greatest difficulties with mixture models arise from the complexity of the posterior distribution $\pi(\theta|y_{1:N})$, which often involves many local modes spread over a high-dimensional space, so that standard algorithms do not converge. Methods such as the Gibbs sampler theoretically converge to the region of highest probability, but can become trapped in suboptimal modes in practice (Celeux et al., 2000). It is therefore of interest to see if our algorithm can deal with such a polymodal target distribution, even when the move steps are performed by a normal instrumental distribution.

The identifiability constraint was chosen to be $\mu_1 < \ldots < \mu_k$. The standard errors σ_j were reparameterised to $s_j = \log(\sigma_j)$, in order to lower the posterior distribution tails in σ; see § 4·2. Correct choice of prior distribution is still a sensitive issue in mixture modelling; see for instance Robert & Mengersen (1999). However, since it was not our aim to address this problem, we just used a simple yet reasonable prior, given by the uniform distribution over the compact set corresponding to the following constraints:

$$p_1, \ldots, p_{k-1} \geqslant 0, \quad p_1 + \ldots + p_{k-1} \leqslant 1,$$
$$\bar{\mu}_L < \mu_1 < \ldots < \mu_k < \bar{\mu}_U, \quad s_1, \ldots, s_k \in [\bar{s}_L, \bar{s}_U],$$

where $\bar{\mu}_L$, $\bar{\mu}_U$, $\bar{s}_L$ and $\bar{s}_U$ are data-based; we took $\bar{\mu}_L = \min(y_i)$, $\bar{\mu}_U = \max(y_i)$, $\bar{s}_U = \log\{(\bar{\mu}_U - \bar{\mu}_L)/6\}$ and $\bar{s}_L = \log(\min|y_i - y_j|/2)$.

Our algorithm was applied to $N = 1000$ observations drawn from a Gaussian mixture model with $k = 5$ components, for which the corresponding parameter

$$\theta_0 = (p_1^{(0)}, \ldots, p_4^{(0)}, \mu_1^{(0)}, \ldots, \mu_5^{(0)}, s_1^{(0)}, \ldots, s_5^{(0)})$$

is a vector of 14 dimensions. Figure 2 compares the density $p(y|\hat{\theta})$, where $\hat{\theta} = \hat{E}_{\pi_N}(\theta)$ is

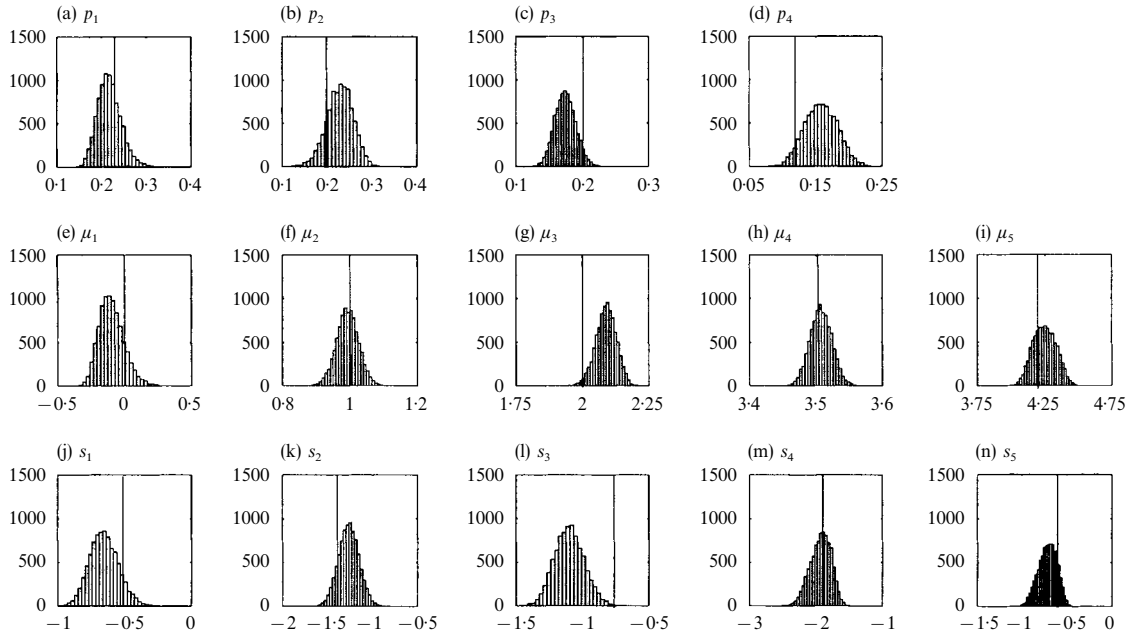

Fig. 3: Mixture model example. Weighted histograms of the final particle system, $H = 10\,000$, with the true values θ_0 of the parameters indicated by vertical lines.

the estimate provided by the iterated batch importance sampling algorithm, with the true density $p(y|\theta_0)$. The density estimate clearly fits the data well. Figure 3 gives the weighted histograms of the resulting particle system, which targets π_N; for each parameter, the corresponding true value is identified by a vertical line. The number of particles was set to $H = 10\,000$, which gave a mean squared error of $\hat{E}_{\pi_N}(\theta)$ of the order of 10^{-5} over 10 runs.

ACKNOWLEDGEMENT

I am grateful to C. P. Robert for his guidance, to J. Rousseau for her comments on Theorem 1, and to the editor, the associate editor and the referee for their helpful suggestions. This paper is the first part of my Ph.D. thesis at Université Pierre et Marie Curie, Paris, and was partially supported by the European Union Training and Mobility of Researchers network on 'Statistical and computational methods for the analysis of spatial data' and by the European Science Foundation Highly Structured Stochastic Systems programme via the research kitchen workshop at Villard de Lans, in January 2000.

APPENDIX

Regularity conditions for Theorem 1

The quantity $L_n(\theta)$ refers to the likelihood $p(y_{1:n}|\theta)$and $l_n(\theta)$ to the corresponding loglikelihood. The observations $y_1, \ldots, y_n$ are drawn from the distribution corresponding to $\theta = \theta_0$. The following conditions are assumed to be fulfilled almost surely.

Condition 1. The maximum $\hat{\theta}_n$ of $L_n(\theta)$ exists for each n and converges to θ_0 as $n \to \infty$.

Condition 2. The matrix

$$\Sigma_n = -\left\{ \frac{1}{n} \frac{\partial^2 l_n(\hat{\theta}_n)}{\partial\theta\, \partial\theta'} \right\}^{-1}$$

is positive definite and converges to $I(\theta_0)$, the Fisher information matrix evaluated at θ_0.

Condition 3. There exists $\Delta > 0$ such that

$$0 < \delta < \Delta \Rightarrow \limsup_{n \to +\infty} \left[\frac{1}{n} \sup_{|\theta - \hat{\theta}_n| > \delta} \{ l_n(\theta) - l_n(\hat{\theta}_n) \} \right] < 0.$$

Condition 4. The function $l_n(\theta)$ is thrice continuously differentiable, the quantity

$$\sup_{\theta \in \Theta'} \left\{ \frac{1}{n} \frac{\partial^3 l_n}{\partial\theta^3}(\theta) \right\}$$

is bounded from above in n, for any compact set $\Theta' \subset \Theta$, and the bound does not depend on the observations.

References

Albert, J. H. & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *J. Am. Statist. Assoc.* **88**, 669–79.

Carpenter, J., Clifford, P. & Fearnhead, P. (1999). An improved particle filter for nonlinear problems. *IEE Proc. Radar, Sonar Navig.* **146**, 2–7.

Celeux, G., Hurn, M. & Robert, C. P. (2000). Computational and inferential difficulties with mixture posterior distributions. *J. Am. Statist. Assoc.* **95**, 957–70.

Dellaportas, P. & Smith, A. F. M. (1993). Bayesian inference for generalised linear and proportional hazards models via Gibbs sampling. *Appl. Statist.* **42**, 443–59.

Doucet, A., De Freitas, N. & Gordon, N. J. (2001). *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* **57**, 1317–39.

Gilks, W. R. & Berzuini, C. (2001). Following a moving target: Monte Carlo inference for dynamic Bayesian models. *J. R. Statist. Soc.* B **63**, 127–46.

Gordon, N. J., Salmond, D. J. & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. Radar Sig. Proces.* **140**, 107–13.

Liu, J. & Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *J. Am. Statist. Assoc.* **93**, 1032–44.

McCullagh, P. & Nelder, J. A. (1989). *Generalized Linear Models*, 2nd ed. London: Chapman and Hall.

Robert, C.P. & Mengersen, K. L. (1999). Reparameterisation issues in mixture modelling and their bearing on MCMC algorithms. *Comp. Statist. Data Anal.* **29**, 325–43.

[*Received December* 2000. *Revised July* 2001]