Bayesian Analysis (2006) **1**, Number 2, pp. 345–362

A One-Pass Sequential Monte Carlo Method for Bayesian Analysis of Massive Datasets

Suhrid Balakrishnan[*] and David Madigan[†]

Abstract. For Bayesian analysis of massive data, Markov chain Monte Carlo (MCMC) techniques often prove infeasible due to computational resource constraints. Standard MCMC methods generally require a complete scan of the dataset for each iteration. Ridgeway and Madigan (2002) and Chopin (2002b) recently presented importance sampling algorithms that combined simulations from a posterior distribution conditioned on a small portion of the dataset with a reweighting of those simulations to condition on the remainder of the dataset. While these algorithms drastically reduce the number of data accesses as compared to traditional MCMC, they still require substantially more than a single pass over the dataset. In this paper, we present "1PFS," an efficient, one-pass algorithm. The algorithm employs a simple modification of the Ridgeway and Madigan (2002) particle filtering algorithm that replaces the MCMC based "rejuvenation" step with a more efficient "shrinkage" kernel smoothing based step. To show proof-of-concept and to enable a direct comparison, we demonstrate 1PFS on the same examples presented in Ridgeway and Madigan (2002), namely a mixture model for Markov chains and Bayesian logistic regression. Our results indicate the proposed scheme delivers accurate parameter estimates while employing only a single pass through the data.

Keywords: Sequential Monte Carlo, One-Pass,Massive Datasets

1 Introduction

Routine Bayesian data analysis relies on Monte Carlo algorithms that perform thousands or even millions of laps through the data. This can preclude Bayesian analyses of massive datasets. The Bayesian approach does, however, lend itself naturally to sequential algorithms. The posterior after the ith observation becomes the prior for the $i + 1$st observation, and so on. For analyses that adopt conjugate prior distributions, this can provide a simple, scalable approach for dealing with massive datasets. However, this conjugate setup describes only a small fraction of today's Bayesian applications.

In this paper we propose a general algorithm that performs a rigorous Bayesian computation on a small, manageable portion of the dataset and then sequentially adapts those calculations with the remaining observations. The algorithm loads each of these remaining observations into memory only once yet maintains inferential fidelity.

There exists a small literature focussed on scaling up Bayesian methods to massive

[*] Department of Computer Science, Rutgers University, Piscataway, NJ, http://sol.rutgers.edu/staff/marianth/suhrid.html

[*] Department of Statistics, Rutgers University, Piscataway, NJ, http://www.stat.rutgers.edu/~madigan/

datasets. A number of authors have proposed large-scale Bayesian network learning algorithms, although most of this work is not actually Bayesian per se (see, for example, Friedman et al. 1999) and none to our knowledge is one-pass. Posse (2001) presents an algorithm for large-scale Bayesian mixture modelling. DuMouchel (1999) presents an algorithm for learning a Gamma-Poisson empirical Bayes model from massive frequency tables.

Our work improves and extends the previously proposed scheme in Ridgeway and Madigan (2002) (which is essentially the same as that outlined for static models in Chopin 2002b) and formulates a one-pass method of analysis.

2 Bayesian computation for massive datasets

The outputs of Bayesian data analyses often take the form of estimates of expectations. Specifically, we compute the expected value of the quantity of interest, $h(\theta)$, using

$$E(h(\theta)|x_1,\ldots,x_N) = \int h(\theta)f(\theta|x_1,\ldots,x_N)d\theta \tag{1}$$

where $f(\theta|\mathbf{x})$, is the posterior density of the parameters given the observed data. Analytic expressions for such integrals exist only in the simplest of cases leading to a de facto reliance on Monte Carlo methods. Monte Carlo integration methods sample from the posterior, $f(\theta|\mathbf{x})$, and then estimate $E(h(\theta)|x_1,\ldots,x_N)$ as $\frac{1}{M}\sum_{i=1}^{M} h(\theta_i)$ where $\theta_1,\ldots,\theta_M$ comprise a sample of M "particles" from $f(\theta|\mathbf{x})$. The law of large numbers ensures convergence:

$$\lim_{M\to\infty} \frac{1}{M}\sum_{i=1}^{M} h(\theta_i) = \int h(\theta)f(\theta|x_1,\ldots,x_N)d\theta. \tag{2}$$

Importance sampling methods sample from a different density, say $g(\theta)$, and take weighted averages:

$$\begin{aligned}\int h(\theta)f(\theta|x_1,\ldots,x_N)d\theta &= \int h(\theta)\frac{f(\theta|\mathbf{x})}{g(\theta)}g(\theta)d\theta & (3)\\ &= \lim_{M\to\infty}\frac{1}{M}\sum_{i=1}^{M} w_i h(\theta_i) & (4)\end{aligned}$$

where θ_i is now a draw from $g(\theta)$ and $w_i = f(\theta_i|\mathbf{x})/g(\theta_i)$, is a weight associated with a particular particle θ_i. Since the expected value of w_i under $g(\theta)$ is 1, we need only compute weights up to a constant of proportionality and then normalize, leading to:

$$\widehat{E}(h(\theta|x_1,\ldots,x_N)) = \frac{\sum_{i=1}^{M} w_i h(\theta_i)}{\sum_{i=1}^{M} w_i}. \tag{5}$$

Geweke (1989) provides conditions under which these estimates are asymptotically consistent.

The algorithm in Ridgeway and Madigan (2002), essentially consists of partitioning the data $\{x_1, \ldots, x_N\}$ into two pieces, a manageable portion $D_{1:n} = x_1, \ldots, x_n$ where $n \ll N$, and the remainder of the data, $D_{n+1:N} = x_{n+1}, \ldots, x_N$, and then applying importance sampling with $g(\theta) = f(\theta|x_1, \ldots, x_n)$. Now, if the observations are conditionally independent given the parameters θ, i.e. $f(x_1, \ldots, x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta)$, the corresponding importance sampling weights have a particularly simple form:

$$w_i \propto f(D_{n+1:N}|\theta_i) = \prod_{x_j \in D_{n+1:N}} f(x_j|\theta_i). \qquad (6)$$

This is just the likelihood of the observations in $D_{n+1:N}$ evaluated at each particle.

Unfortunately, the Monte Carlo variance of the resulting importance sampling estimates grows quickly. Since all of the terms are positive in:

$$Var(\theta|D_{1:n}) = E(Var(\theta|D_{1:n}, D_{n+1:N})) + Var(E(\theta|D_{1:n}, D_{n+1:N})), \qquad (7)$$

the posterior variance with the additional observations in $D_{n+1:N}$ is, in expectation, smaller than the posterior variance conditioned only on $D_{1:n}$. Therefore, although the location of the sampling density should be close to the target density, its spread will most likely be wider than that of the target. As additional observations become available, $f(\theta|D_{1:n}, D_{n+1:N})$ becomes much narrower than $f(\theta|D_{1:n})$. The result of this narrowing is that the weights of many of the original draws from the sampling density approach zero and so we have few effective draws from the target density, a phenomenon also known as degeneracy of the sample. Figure 1 demonstrates the problem schematically. The wider density represents the sampling density $f(\theta|D_{1:n})$ that generates the particles. However, the target density, $f(\theta|D_{1:n}, D_{n+1:N})$, shown as a dashed curve, is shifted and narrower. About half of the draws from $f(\theta|D_{1:n})$ will have importance weight near zero.

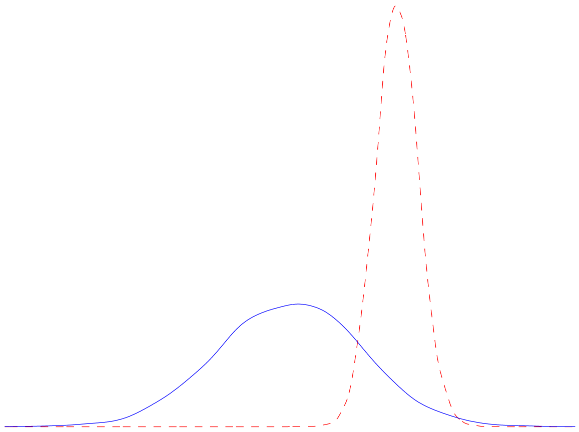


Figure 1: Comparison of $f(\theta|D_{1:n}, D_{n+1:N})$ (dashed) and $f(\theta|D_{1:n})$ (solid).

Ridgeway and Madigan (2002) monitor this degeneracy via the so-called effective sample size or ESS. Kong et al. (1994) provided a simple approximation for the ESS:

$$ESS = \frac{M}{1 + Var(w)} = \frac{(\sum w_i)^2}{\sum w_i^2}.$$

When the ESS drops below some pre-specified level, Ridgeway and Madigan (2002) counter the degeneracy via a resample-move or "rejuvenation" step, an idea that borrows from the particle filtering/sequential Monte Carlo literature – see, for example Doucet et al. (2001) and Gilks and Berzuini (2001). The resample-move step first resamples the particles with probabilities proportional to the weights and then applies a single Metropolis-Hastings "move" step to each particle. Each move requires a complete scan of all the data. Figure 2 provides an outline of the algorithm.

1. Load as much data into memory as possible to form $D_{1:n}$
2. Draw M times from $f(\theta|D_{1:n})$ via Monte Carlo or Markov chain Monte Carlo
3. Iterate through the remaining observations (those that comprise $D_{n+1:N}$). For each observation, x_j, update the log-weights on all of the draws from $f(\theta|D_{1:n})$.

```
Set $j = n + 1$ and $p \in (0, 1)$ to the allowable decrease in ESS
While $j < N$
    Set $w_i = 1$, $i = 1, \ldots, M$
    While ESS $> pM$
        $j \leftarrow j + 1$
        for $i$ in $1, \ldots, M$ do $w_i \leftarrow w_i \times f(x_j|\theta_i)$
    Resample M times with replacement from $\theta_1, \ldots, \theta_M$
        with probability proportional to $w_i$
    for $i$ in $1, \ldots, M$ do one MCMC step to move $\theta_i$
        conditioned on $n + j$ observations
```

Figure 2: The Ridgeway and Madigan (2002) algorithm for massive datasets

While the Ridgeway and Madigan (2002) scheme decreases the number of data accesses by up to 99% as compared to vanilla MCMC, it is not a one-pass algorithm since the move portion of the rejuvenation scheme, i.e. the single MCMC step, involves conditioning on all of the data processed thus far. Although Ridgeway and Madigan provide arguments that show that rejuvenations become less frequent as more data accumulate, this is still a significant flaw since the number of repeated data accesses per observation grows without bound.

3 One-pass Particle Filtering for Massive Datasets

Our proposed one-pass particle filtering algorithm 1PFS (One-pass Particle Filter with Shrinkage) differs from the Ridgeway and Madigan algorithm of Figure 2 only in the rejuvenation step. The new rejuvenation step uses a "shrinkage" kernel smoothing approximation to the current importance sampling distribution.

Thus 1PFS starts out with Steps 1 and 2 of the Ridgeway and Madigan (2002) algorithm (see Figure 2). Then, as with Ridgeway and Madigan, 1PFS iterates the outer loop of Step 3 until the ESS deteriorates below some tolerance limit (10% of M, say). Assuming that this occurs after absorbing n_1 observations, 1PFS then resamples M times with replacement the particles from the posterior conditioned on the first $n+n_1$ data points. The resample selects each particle θ_i with probability proportional to w_i.

Note that these draws still represent a sample, albeit a dependent one, from the posterior conditioned on the first $n + n_1$ data points. Several of the θ_i will appear multiple times in this new sample. For the most part this refreshed sample will be devoid of those θ_i not supported by the data.

In order to rejuvenate the sample, we propose to approximate the distribution of the resampled θ_i using kernel smoothing. Liu and West (2001) note that centering the kernels at the standard locations (i.e., at the existing particles) systematically results in a density estimate that is over-dispersed. Liu and West (2001) propose a shrinkage scheme to correct for this over-dispersion via a weighted shift of the location of the particles towards the sample mean.

More precisely, 1PFS uses kernel smoothing to approximate the importance sampling posterior density $f(\theta|D_{1:n+n_1})$ of the parameters as per:

$$\widehat{f}(\theta|D_{1:n+n_1}) \quad = \quad \sum_{i=1}^{M} K(\theta;\widetilde{\theta}_i, b^2V) \tag{8}$$

where $K(\theta; s, T)$ is the value at θ of the kernel function (e.g., Gaussian) with mean s and variance matrix T. $\widetilde{\theta}_i$ and V are the shifted sample/particle values and the sample Monte Carlo variance respectively with b being the kernel bandwidth. Note that the w_i's, are all identically equal to 1 in the above formula because a resample step preceded the rejuvenation step. The shrinkage rule specifies the shifted sample locations as:

$$\widetilde{\theta}_i = a\theta_i + (1-a)\overline{\theta} \tag{9}$$

where $a = \sqrt{1-b^2}$ and $\overline{\theta}$ is the current Monte Carlo mean θ_i value. The sample drawn from the kernels placed at the shrinkage locations will not only have the correct mean (which is the original sample mean, $\overline{\theta}$ and is unchanged) but also the correct variance (the sample variance, V).

Therefore, to rejuvenate the sample, for each of the new θ_i's we simply sample from the shrinkage kernel density based approximation to the importance sample distribution that we currently have, $\widehat{f}(\theta|D_{1:n+n_1})$. Our rejuvenated θ_i's now represent a more diverse set of parameter values with an effective sample size closer to M again. Figure 4 graphically walks through the resample-move process for this "smooth bootstrap" step-by-step and Figure 3 shows the new reweighting step to replace step 3 of the Ridgeway and Madigan (2002) algorithm.

```
3. Iterate through the remaining observations (those that comprise $D_{n+1:N}$). For
   each observation, $x_j$, update the log-weights on all of the draws from $f(\theta|D_{1:n})$.
   Set $j = n+1$, $p \in (0,1)$ to the allowable decrease in ESS and $b$, the kernel
   bandwidth (Note: enables computation of $a = \sqrt{1-b^2}$)
   While $j < N$
       Set $w_i = 1$, $i = 1, \ldots, M$
       While ESS $> pM$
           $j \leftarrow j+1$
           for $i$ in $1, \ldots, M$ do $w_i \leftarrow w_i \times f(x_j|\theta_i)$
       Resample M times with replacement from $\theta_1, \ldots, \theta_M$
           with probability proportional to $w_i$
       Compute $\overline{\theta}$, $V$
       for $i$ in $1, \ldots, M$ do
           Compute $\widetilde{\theta}_i = a\theta_i + (1-a)\overline{\theta}$
           Sample new $\theta_i$ from $K(\widetilde{\theta}_i, b^2V)$
```

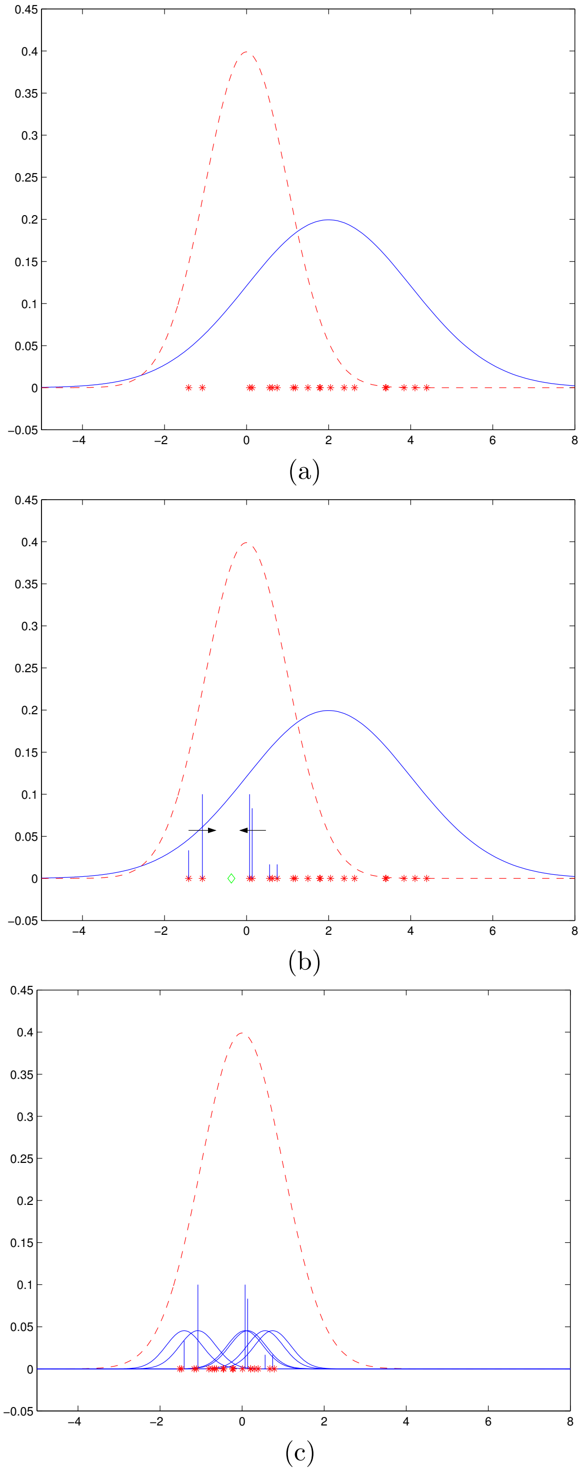
Figure 3: One-pass Particle Filtering for Massive Datasets

After rejuvenating the set of θ_i, we can continue where we left off, on observation $n + n_1 + 1$, absorbing additional observations until either we include the entire dataset or the ESS again has dropped too low and we need to preform a new rejuvenation step.

3.1 Convergence of the Smooth Bootstrap; Bandwidth Selection

There exist established asymptotic (as the number of particles tends to infinity, i.e., $M \rightarrow \infty$) Central Limit Theorems for Sequential Monte Carlo methods – see Moral and Guionnet (1999), Gilks and Berzuini (2001), and Chopin (2002a). These results deal with the more general version of the sequential inference problem involving unseen state variables in addition to static model parameters. These results also apply to the simpler version of the problem we are concerned with, namely filtering for static model parameters. Indeed, the static-only case is better behaved and more tractable than the general problem (Chopin 2002a). Further, since we don't have a general state-space model (and the sequential updating is only an artifact to reduce the number of data accesses) we are concerned solely with the convergence properties of the final posterior distribution estimate that our algorithm returns.

These Central Limit Theorems hold true for sequential Monte Carlo methods involv-



(a)

(b)

(c)

Figure 4: The resample-move step. (a) Generate an initial sample from $f(\theta|D_{1:n})$ (the solid curve). The stars mark the particles, the sampled θ_i. (b) Weight based on $f(\theta|D_{1:n}, D_{n+1:N})$ (the dashed density) and resample, the length of the vertical lines indicate the number of times resampled. Shrink these locations towards $\overline{\theta}$ (the open diamond) 3) For each θ_i sample from the now shifted kernel density distribution and thus diversify and obtain the new sample (the stars mark these locations).

ing sampling-importance resampling and MCMC (rejuvenation) moves as per Ridgeway and Madigan. 1PFS, however, employs an "extra" approximation step involving the kernel smoothing approximation to the importance sampling posterior distribution, $\widehat{f}(\theta|x)$. Intuitively, this extra approximation will be inconsequential if, in the limit as the number of particles tend to infinity, the particles sampled from the kernel smoothed approximation to the posterior distribution will resemble random samples from the importance sampling posterior $f(\theta|x)$.

Stavropoulos and Titterington (2001) prove a restricted version of the above statement formally. Their theorem states:

Theorem 1 *Under mild conditions, for univariate θ and the Normal kernel K, the cumulative distribution function of the values generated by the kernel approximation to the posterior distribution $\widehat{f}(\theta|x)$, converges to that of the target density, $f(\theta|x)$.*

The proof has an easy multivariate generalization and can be adapted for non-Normal K as well. The assumptions under which theorem 1 holds are fairly mild, essentially the same as those required by Geweke (1989) for the importance sample estimates to converge, with the additional requirement that the kernel functions variance should shrink to zero as the number of particles tends to infinity.

For Normal kernels (where $K(s,T) = \varphi(s,T)$, the Gaussian density function), kernel density estimation literature Silverman (1986) suggests a choice of $T = V{b_M}^2$, with

$$b_M = \left(\frac{4}{(d+2)M}\right)^{\frac{1}{d+4}} \tag{10}$$

where d is the dimensionality of the samples, M is the number of samples and V is the sample Monte Carlo variance estimate. This choice of bandwidth is asymptotically optimal if the density being approximated is multivariate-Normal, and the samples had been obtained from this distribution Stavropoulos and Titterington (2001).

The following sections of the paper present two examples (both of which were previously analyzed in Ridgeway and Madigan 2002) that elucidate the application of the algorithm in practice, followed by a discussion of the method and conclusions.

4 Example I - Fully Bayes Logistic Regression

The first example we consider concerns Bayesian logistic regression. The training data comprise vectors $\mathbf{x_i} = [x_{i_1}, \ldots, x_{i_d}]^T$ in $\mathbf{R^d}$ and $y_i \in \{0,1\}$, $i = 1, \ldots, N$. We consider a model of the form:

$$p(y=1|\mathbf{x}) = \psi(\boldsymbol{\beta}^T\mathbf{x}) \tag{11}$$

where $\boldsymbol{\beta}$ is a vector of regression coefficients and $\psi(\cdot)$ is the logistic link function. Following some recent literature on so-called "sparse" models (Tibshirani 1995; Figueiredo 2001) we use an independent Laplace prior for each component of $\boldsymbol{\beta}$:

$$\pi(\beta_i|\gamma) = \frac{1}{2}\sqrt{\gamma}e^{-\gamma|\beta_i|}, \lambda > 0, i = 1, \ldots, d.$$

This prior typically results in posterior modes of zero for many parameters and as such accomplishes simultaneous shrinkage and variable selection. Our interest here however is not in obtaining the posterior mode (see Genkin et al. 2004) but rather in fully Bayesian inference for arbitrary characteristics of the posterior distribution of $\boldsymbol{\beta}$. In the example below we set $\gamma = 5$; Genkin et al. (2004) discuss approaches for selecting γ.

Following Ridgeway and Madigan (2002) we report fully Bayesian logistic regression analysis of the "outpic data" comprising 744,963 customer records, 57 Mb when stored in double precision. Thus, in our notation, $N = 744,963$. These data originated in a major telecommunications company. The binary response variable identifies customers who have switched to a competitor. There are seven predictor variables. Five of these are continuous and two are 3-level categorical variables. Thus for logistic regression there are $d = 10$ parameters. This dataset is small enough that regular MCMC to compute $f(\boldsymbol{\beta}|D_{1:N})$, while cumbersome, is still feasible. We also used MCMC to generate the initial particles from $f(\boldsymbol{\beta}|D_{1:n})$. In both cases we used a straightforward Metropolis-within-Gibbs sampler. Ridgeway and Madigan (2002) describe this sampler in detail. The key point is that the MCMC sampler requires one complete pass through the data per iteration.

1PFS used a Gaussian kernel function. Formula 10 defines the kernel bandwidth. Conditioning on the first 10,000 observations (i.e., $n = 10,000$), we generated 25,000 initial particles using the MCMC algorithm, dropping the first 5,000 (i.e. $M = 20,000$). Thus we accessed each of the first 10,000 observations 25,000 times. 1PFS executed a rejuvenation step whenever the ESS dropped below 10,000 (which occurred 51 times until the whole dataset was processed) corresponding to the tolerance limit $p = 0.5$. By construction, 1PFS accessed observations 10,001-744,963 just once. Ridgeway and Madigan's algorithm with the same rejuvenation schedule also accessed the first 10,000 observations 25,000 times each, but then accessed the remaining 734,963 observations a total of 2,352,460 times or 3.2 times per observation.

In addition to 1PFS, we also fit the logistic regression model using maximum likelihood and using a full MCMC run on the entire dataset. Table 1 shows the resulting estimates.

	$\beta(1)$	$\beta(2)$	$\beta(3)$	$\beta(4)$	$\beta(5)$	$\beta(6)$	$\beta(7)$	$\beta(8)$	$\beta(9)$	$\beta(10)$
MLE	-0.574	0.155	0.056	0.220	-0.087	0.361	-0.358	-0.204	0.079	0.079
MCMC	-0.574	0.155	0.056	0.220	-0.087	0.360	-0.358	-0.204	0.080	0.078
1PFS	-0.574	0.156	0.056	0.221	-0.087	0.360	-0.357	-0.204	0.079	0.079

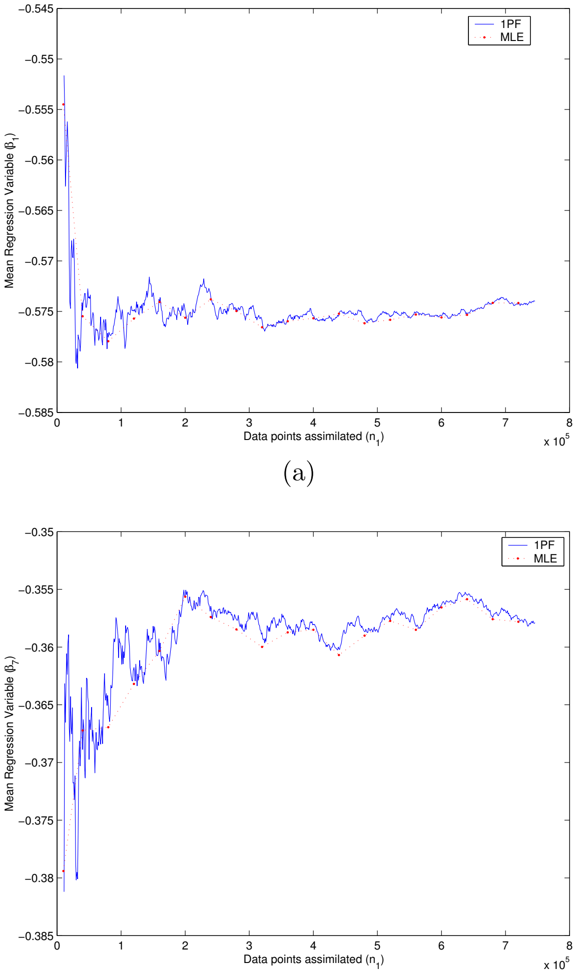
Table 1: Mean $\boldsymbol{\beta}$ estimates obtained from Bayesian logistic regression analysis of the outpic data.

Table 2 shows the total number of data accesses.

1PFS not only scans (most of) the dataset only once but also produces parameter estimates that are very close to the desired values. Indeed, on smaller subsets of the data, the maximum likelihood estimates for the parameter values and those obtained via 1PFS are very similar as well (Figure 5).

Algorithm	first 10,000	next 734,963
MCMC	2.5 $\times 10^8$	1.8 $\times 10^{10}$
R&M	2.5 $\times 10^8$	2.4 $\times 10^6$
1PFS	2.5 $\times 10^8$	7.3 $\times 10^5$

Table 2: Total number of data accesses for MCMC (Markov chain Monte Carlo on the entire dataset), R&M (Ridgeway and Madigan's Particle Filter), and 1PFS (One-pass Kernel-based Particle Filter).



(a)

(b)

Figure 5: Plot showing the representative mean posterior parameter values ($\beta(1)$ and $\beta(7)$) determined via 1PFS as a function of the amount of data processed. Also shown on the plot is the corresponding MLE for the same amount of data.

5 Example II - Mixtures of Transition Models

Mixtures of first-order transition models (i.e., finite state Markov chains) have attracted recent attention in a variety of applications such as web user modelling (Cadez et al. 2000; Ridgeway 1997) and unsupervised training of robots (Ramoni et al. 2002). This mixture model assumes that the data comprise N state sequences of random length and that one of C transition matrices generated each sequence. However, neither the transition matrices nor the mixing proportions are known. Thus the unknown parameters of this model are the C transition matrices (each $S \times S$, where S is the size of the state space), $P_1, \dots, P_C$, the mixing vector of length C, and the N cluster assignments, $z_j \in \{1, \dots, C\}, j = 1, \dots, N$. A Bayesian analysis estimates the posterior distribution of these unknowns given a set of observed sequences. We assume that both C and S are fixed.

Ridgeway and Madigan (2002) describe a simple-to-implement Gibbs sampler that generates draws from this posterior distribution. However, each iteration requires two scans of the entire dataset, one for the matrix update and one for the cluster assignment update. Consequently, computation time becomes prohibitive for any N much bigger than a few tens of thousands. Ridgeway and Madigan (2002) propose a particle filtering algorithm for this model and, as with the logistic regression example, the number of data accesses decreases dramatically compared to the Gibbs sampler. However each rejuvenation step requires a complete scan of the data to that point. Here we apply the 1PFS algorithm to this model in the context of a particular example.

Specifically we generated $N = 1$ million sequences of length between 5 and 20 from two 4×4 transition matrices. We used the first $n = 1000$ sequences to obtain the initial sample of $M = 1000$ particles. For this example 1PFS executes a rejuvenation step each time ESS drops below 100 (i.e. $p = 0.1$). Because both the rows of the transition matrices and the vector of mixing proportions must sum to one, the kernel function, $K(.,.)$, we chose for this example was Dirichlet. Following Aitchison and Lauder (1985), we choose the bandwidth b that maximizes the pseudo-likelihood (the average leave-one-out cross validation approximated likelihood).

The use of the Dirichlet kernel involves one extra detail. The Liu and West (2001) shrinkage rule requires a parametrization of the kernel $K(\widetilde{\theta}_i, b^2V)$ in terms of its mean $\widetilde{\theta}_i$ and variance b^2V. Unfortunately, starting from a mean and variance, a closed-form expression for the corresponding Dirichlet distribution Dirichlet(α) does not exist. Following Ronning (1989) we compute an approximation to α by matching first and second moments. Specifically, the parameter values α_{isc} for the i^{th} row of the transition matrix P_c are:

$$\alpha_{isc} = \widetilde{\theta}_{isc} \sum_s \alpha_{isc} \tag{12}$$

$$\log \sum_s \alpha_{isc} = \frac{1}{S-1} \sum_{s=1}^{S-1} \log \left(\frac{\widetilde{\theta}_{isc}(1-\widetilde{\theta}_{isc})}{b^2 V_{isc}} - 1 \right) \tag{13}$$

Here we model each row independently. A similar set of equations exists for the mixing vector's parameters (implying a total of $d = 25$ independent parameters).

Except for the first 1000 observations, which generate the initial set of particles, 1PFS accesses each of the remaining observations once. Once again, this represents a substantial computational savings as compared to the Ridgeway and Madigan (2002) scheme. A Gibbs sampler, conditioned on the entire dataset, would need to access each observation 2000 times.

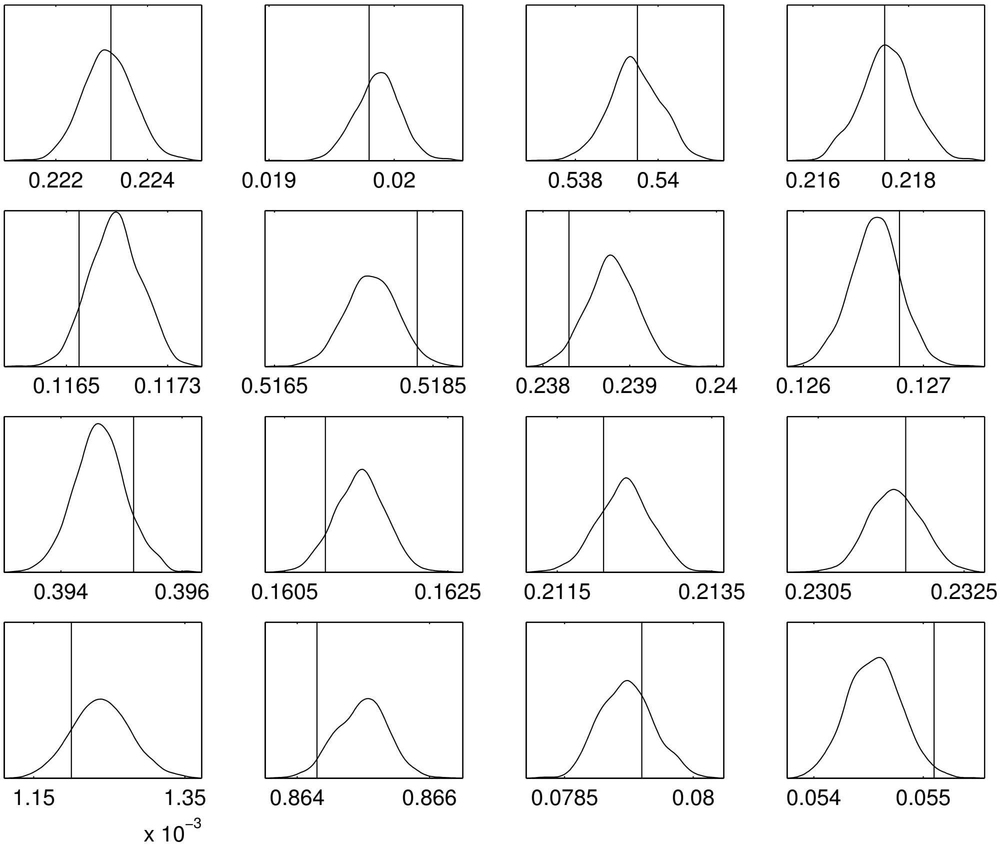


Figure 6: The posterior distribution of the transition probabilities for one of the transition matrices. 1PFS generated these densities. The vertical line marks the true value used to simulate the dataset.

While efficiency as measured by the number of data accesses is important in the analysis of massive datasets, precision of parameter estimates is also important. Figure 6 shows the marginal posterior densities for the 16 transition probabilities from the first mixture component's transition matrix. The smooth density plot is based on the $M = 1000$ 1PFS particles. The figure also marks the location of the parameter value that generated the data. All of these values are within the region with most of the posterior mass.

With 1,000,000 observations, a Gibbs sampler here is prohibitively expensive (scaling computational time linearly and disregarding memory constraints leads to an estimate of around 2 months CPU time). On a subset of the data comprising the first 10,000 observations, the two methods produced nearly identical posterior distributions – see Figure 7.

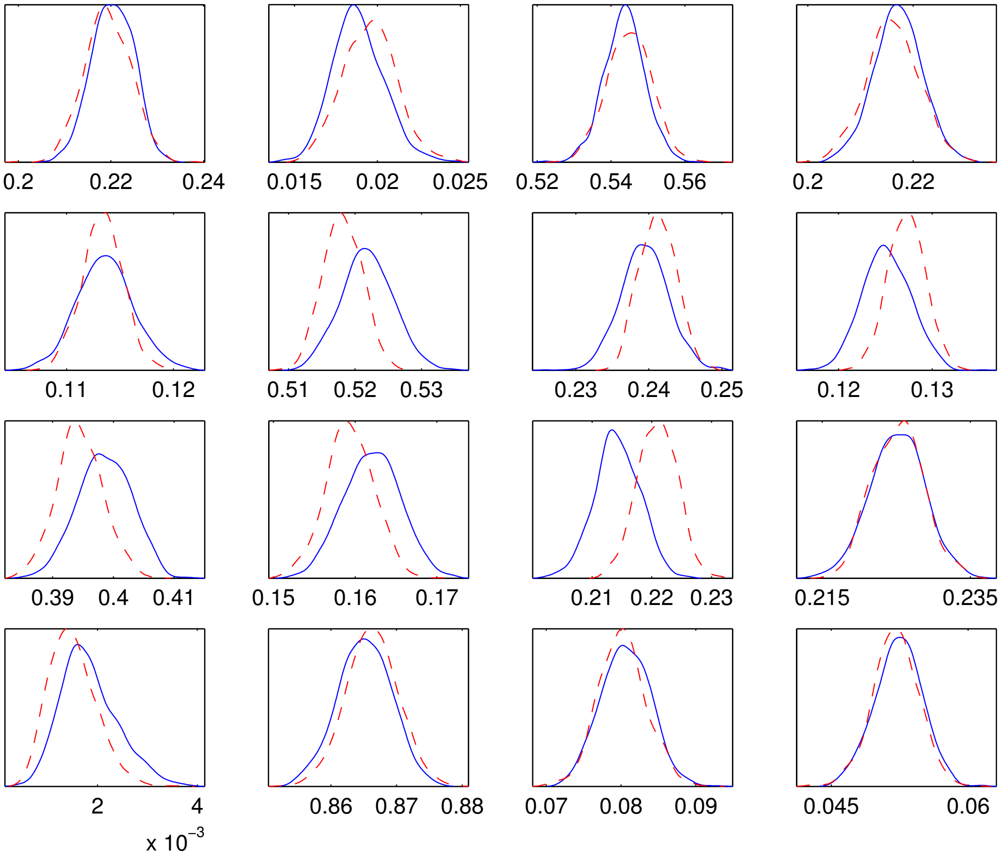


Figure 7: The posterior distribution of the transition probabilities for one of the transition matrices for a smaller subset of the whole dataset (the first 10,000 observations). The posterior density found via MCMC is represented by the blue solid line and that of the particle filter by the red dashed line.

Note that increasing M does not change the number of data accesses for 1PFS while each additional draw represents yet another scan for the standard implementation.

6 Discussion

MCMC has established itself as a standard tool for the statistical analysis of complex models. Data mining research, by contrast, rarely features MCMC. Indeed, for data mining applications involving massive datasets, computational barriers essentially preclude routine use of MCMC. MCMC is however an indispensable tool for Bayesian analysis, especially in those applications where the inferential targets are more complex than posterior means or modes. As such we contend that extension of MCMC methods to larger datasets is an important research challenge. We have presented one particular line of attack. Working with samples from massive datasets represents an alternative strategy for large-scale Bayesian data analysis and may be viable for some applications. In high dimensional applications, however, throwing data away may be too costly.

Indeed, high dimensionality is well known to be the bane of all kernel density based approximation methods. However, it should be pointed out here that our proposed scheme utilizes kernel smoothing (and not density estimation per se) and thus should be applicable even in medium dimension problems (Liu and West 2001, have examples where they apply kernel smoothing in dimension around 30).

By reducing the number of data accesses by a huge amount (one-pass for all but a small fraction of the data), MCMC becomes viable for a large class of models useful in data mining. We note that 1PFS can bypass the exhaustive analysis of an initial portion of the training data by sampling initial particles from the prior distribution of the parameters. While this doesn't affect any of the analysis, we have seen in practice, that it is often a good idea to start the particle filter with a reasonable set of particles. The sequential nature of the algorithm also allows the analyst to stop when uncertainty in the parameters of interests has dropped below a required tolerance limit. Parallelization of the algorithm is straightforward. Each processor manages a small set of the weighted draws from the posterior and is responsible for updating their weights and computing the refresh step.

Appendix

A Empirical Justification of Bandwidth Selection Rule

As stated in the manuscript, the bandwidth selection rule in Equation 10 is (on average) an optimal finite sample size bandwidth selection rule (minimizes the AIMSE) when the distribution being estimated is known to be multivariate Normal and we are using a Gaussian kernel. Since the addition of observations forces the posterior distribution to asymptotically approach a multivariate Normal, and as we know that for a Gaussian kernel, no other bandwidth selection rule will behave better uniformly (because then it would also be better on average) we believe the choice of bandwidth to be very reasonable (and indeed continually improving as more data is assimilated).

In order to empirically validate this claim, we performed 10 simulations of 1PFS with parameters exactly as specified in the manuscript, except for the bandwidth parameter, which we varied uniformly in the [0-1] interval in increments of 0.1. We then calculated the MSE error between the mean posterior parameter values obtained from regular MCMC and those obtained from the use of 1PFS. The following table (Table 3) shows the mean error (averaged over the 10 runs per bandwidth) and associated standard deviations.

Bandwidth (b_M)	Mean MSE	STD MSE
0.10	0.0436	0.0169
0.20	0.0151	0.0029
0.30	0.0074	0.0012
0.40	0.0063	0.0007
0.50	0.0062	0.0008
0.60	0.0057	0.0008
0.70	0.0062	0.0007
0.80	0.0064	0.0006
0.90	0.0065	0.0008

Table 3: The effect of bandwidth b_M on the mean MSE error estimates (and standard deviations) obtained between regular MCMC and 1PFS for Bayesian logistic regression analysis of the outpic data.

Figure 8 shows the same data on a plot. The results of our limited number of simulations point to the fact that while the bandwidth specified per Equation 10, $b_M = 0.4557$, doesn't give the lowest error (that appears to be at $b_M = 0.6$), it still clearly does provide a reasonable choice.

References

Aitchison, J. and Lauder, I. J. (1985). "Kernel Density Estimation for Compositional Data." *Applied Statistics*, 34(2): 129 – 137. 355

Cadez, I., Heckerman, D., Meek, C., Smyth, P., and White, S. (2000). "Visualization of navigation patterns on a Web site using model-based clustering." Technical report, Microsoft Research. Technical Report MSR-TR-00-18. 355

Chopin, N. (2002a). "Central Limit Theorem for Sequential Monte Carlo Methods and its Applications to Bayesian Inference." Technical report, CREST.
URL http://www.crest.fr/doctravail/document/2002-44.pdf 350

— (2002b). "A sequential particle filter method for static models." *Biometrika*, 89(3): 539 – 552. 345, 346

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer-Verlag. 348

DuMouchel, W. (1999). "Bayesian data mining in large frequency tables, with an application to the FDA spontaneous reporting system (with discussion)." *The American Statistician*, 53(3): 177 – 190. 346

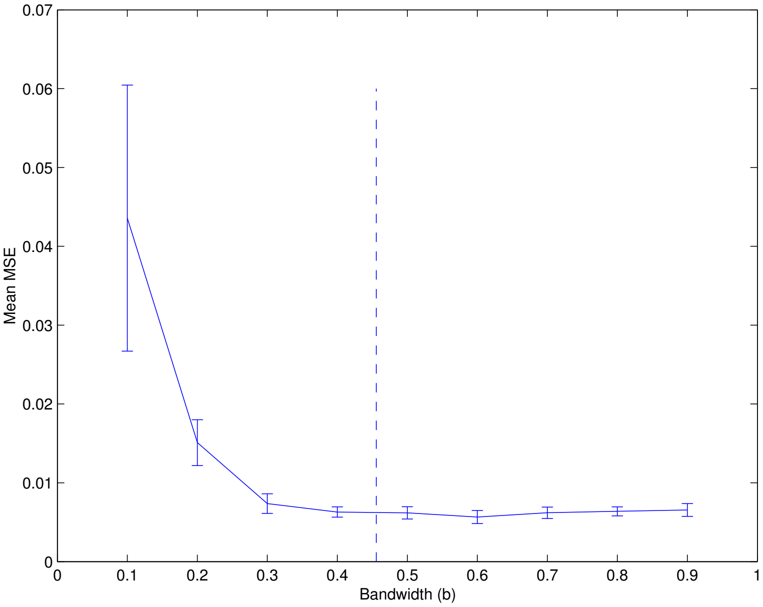


Figure 8: Plot of the MSE and associated standard deviation of Table 3. The dashed vertical line indicates the bandwidth choice recommended by Equation 10.

Figueiredo, M. (2001). "Adaptive sparseness using Jeffreys prior." In T. G. Dietterich, S. B. and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems, (NIPS 14)*. Vancouver, Canada: MIT Press. 352

Friedman, N., Nachman, I., and Peer, D. (1999). "Learning Bayesian Network Structures from Massive Datasets: The Sparse Candidate Algorithm." In *15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, 206 – 215. San Francisco, USA: Morgan Kaufmann. 346

Genkin, A., Lewis, D. D., and Madigan, D. (2004). "Bayesian Logistic Regression for Text Categorization." *In preparation*. 353

Geweke, J. (1989). "Bayesian inference in econometric models using Monte Carlo integration." *Econometrica*, 24: 1317 – 1399. 346, 352

Gilks, W. and Berzuini, C. (2001). "Following a moving target - Monte Carlo inference for dynamic Bayesian models." *Journal of the Royal Statistical Society B*, 63(1): 127 – 146. 348, 350

Kong, A., Liu, J., and Wong, W. (1994). "Sequential imputation and Bayesian missing data problems." *Journal of the American Statistical Association*, 89: 278 – 288. 348

Liu, J. and West, M. (2001). *Combined parameter and state estimation in simulation-based filtering*, 197 – 224. New York, NY: Springer-Verlag. 349, 355, 358

Moral, P. D. and Guionnet, A. (1999). "A central limit theorem for nonlinear filtering using interacting particle systems." *Annals of Applied Probability*, 9: 275 – 297. 350

Posse, C. (2001). "Hierarchical Model-based Clustering For Large Datasets." *Journal of Computational and Graphical Statistics*, 10(3): 464 – 486. 346

Ramoni, M., Sebastiani, P., and Cohen, P. (2002). "Bayesian Clustering by Dynamics." *Machine Learning*, 47(1): 91 – 121. 355

Ridgeway, G. (1997). "Finite discrete Markov process clustering." Technical report, Microsoft Research. Technical Report MSR-TR-97-24. 355

Ridgeway, G. and Madigan, D. (2002). "A Sequential Monte Carlo Method for Bayesian Analysis of Massive Datasets." *Journal of Knowledge Discovery and Data Mining*, 7: 301 – 319. 345, 346, 347, 348, 349, 350, 352, 353, 355, 356

Ronning, G. (1989). "Maximum likelihood estimation of dirichlet distributions." *Journal of Statistical Computation and Simulation*, 32(4): 215 – 221. 355

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis. Monographs on Statistics and Applied Probability*. New York, NY: Chapman Hall. 352

Stavropoulos, P. and Titterington, D. M. (2001). "Improved particle filters and smoothing." In A. Doucet, N. d. F. and Gordon., N. (eds.), *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer-Verlag. 352

Tibshirani, R. (1995). "Regression selection and shrinkage via the lasso." *Journal of the Royal Statistical Society, Series B*, 57: 267 – 288. 352

Acknowledgments

We thank Greg Ridgeway and David Scott for helpful discussions. We also acknowledge the anonymous referees for insightful reviews. Software for both examples is available on the Web at:

`http://paul.rutgers.edu/~suhrid/code/1PFS`