

ON-LINE INFERENCE FOR DATA STREAMS

PETER CLIFFORD

Statistics Department

Oxford University

E-mail: clifford@stats.ox.ac.uk

Rapid accumulation of substantial datasets is now common in many data processing applications. For example in monitoring and examining Internet traffic; analysing high-frequency financial data in market trading; voice and video capture; data logging in numerous areas of scientific enquiry. Markov Chain Monte Carlo (MCMC) methods revolutionised statistical analysis in the 1990s by providing practical, computationally-feasible access to the flexible and coherent framework of Bayesian inference. However, massive datasets have produced difficulties for these methods since, with a few simple exceptions, MCMC implementations require a complete scan of what might be several gigabytes of data at each iteration of the algorithm. For time-series data, progress is possible using modern sequential Monte Carlo methods (known as particle filters). With suitable modifications the techniques can be adapted to deal with more general data catalogues.

1. Bayesian Analysis

The basic components in the Bayesian analysis of a statistical problem are:

- Data: y
- Parameters: θ , functions of interest: $g(\theta)$
- Likelihood: $L(\theta; y)$
- Prior density: $\pi(\theta)$

The prior density and the likelihood are used to calculate integrals of the form

$$\frac{\int g(\theta)\pi(\theta)L(\theta; y)d\theta}{\int \pi(\theta)L(\theta; y)d\theta} = \int g(\theta)\pi(\theta|y)d\theta$$

$$= E\{g(\theta)|y\} \text{ (posterior expectation),}$$

where $\pi(\theta|y)$ is the posterior density of the parameter θ given the data y . Markov chain Monte Carlo (MCMC) methods can be used to construct a chain with successive values, $\theta^1, \theta^2, \dots, \theta^n$, simulated from the equilibrium density $\pi(\theta|y) \propto \pi(\theta)L(\theta; y)$, estimating $E\{g(\theta)|y\}$ by

$$\bar{g} = \frac{\sum_{i=1}^n g(\theta^i)}{n}.$$

If $\{\theta^i\}$ are independent then $\text{Var}(\bar{g}) = \sigma_g^2/n$. Typically $\text{Var}(\bar{g}) = \tau\sigma_g^2/n$ with ‘correlation time’, τ , greater than 1 and ‘effective sample size’ n/τ .

Suppose now that the observational framework expands, giving additional data and an expanded parameter set.

- Data: y, y^+
- Parameters: θ, θ^+ , functions of interest $g(\theta, \theta^+)$

- Likelihood: $L(\theta, \theta^+; y, y^+)$
- Joint prior density: $\pi(\theta)\pi(\theta^+|\theta)$

Question: Can we use the simulations from $\pi(\theta|y)$ to simulate from $\pi(\theta, \theta^+|y, y^+)$ or do we have to start completely afresh using MCMC, for example, on the expanded problem?

2. Time-series (signal processing)

In many applications, time-series data are noisy observations of an unobserved underlying process of interest (the signal). The data, $(y_1, \dots, y_t) = \mathbf{y}_{1:t}$, expand with time, and the parameters (the history of the underlying process) expand correspondingly, $(\theta_1, \dots, \theta_t)$.

In on-line analysis, a basic objective is to maintain knowledge about the current state θ_t , for example to allow estimation of $E\{g(\theta_t)|\mathbf{y}_{1:t}\}$. In signal processing terms, this is the *filtering* problem. Applications include: medical monitoring, robotics, finance.

For simplicity *structural assumptions* are made about the evolving data set.

- $\pi(\theta_1, \dots, \theta_t) = \pi(\theta_1)\pi(\theta_2|\theta_1)\dots\pi(\theta_t|\theta_{t-1})$
(underlying state is Markov)
- $L(\theta_1, \dots, \theta_t; \mathbf{y}_{1:t}) = \prod_{k=1}^t h(y_k|\theta_k)$
(current observations depend only on the current state).

In general, the underlying state process will depend on unknown (hyper)parameters that must be incorporated into a full Bayesian model. The Markov as-

sumption is not severely restrictive and the observational assumptions can be relaxed.

With these assumptions, the current state of knowledge can be updated by

$$\pi(\theta_{t+1}|\mathbf{y}_{1:t}) = \int \pi(\theta_{t+1}|\theta_t)\pi(\theta_t|\mathbf{y}_{1:t})d\theta_t \quad (1)$$

$$\pi(\theta_{t+1}|\mathbf{y}_{1:t+1}) = \frac{h(y_{t+1}|\theta_{t+1})\pi(\theta_{t+1}|\mathbf{y}_{1:t})}{p(y_{t+1}|\mathbf{y}_{1:t})} \quad (2)$$

where

$$p(y_{t+1}|\mathbf{y}_{1:t}) = \int h(y_{t+1}|\theta_{t+1})\pi(\theta_{t+1}|\mathbf{y}_{1:t})d\theta_{t+1}. \quad (3)$$

The first integral is crucial. If θ_t is high-dimensional, evaluation of this integral at each stage will present problems.

When there is a linear Gaussian model for the evolution of the underlying state and when the noise is additive and Gaussian, the integrals can be evaluated explicitly. The posterior distributions then turn out to be Gaussian too. This is the basis of the *Kalman filter* (which basically just updates the means and covariances of the state θ_t). Since the posterior distributions can be obtained explicitly in the linear Gaussian model, it is comparatively straightforward to draw inferences about any unknown parameters involved in the underlying state process and the error model.

In many practical applications, these assumptions are implausible. In particular, the observation process will often be non-linear. An alternative approach in such cases is the *extended Kalman filter* (EKF), in which the updated measurements are linearised about the predicted state, permitting the Kalman filter to be applied approximately. This algorithm and its refinements have proved popular, particularly in the field of object tracking. However, the Gaussian approximation to the density of the underlying state, inherent in the EKF, will often prove to be inadequate, causing the update procedure to become unstable.

Other methods involve approximating distributions by mixtures of Gaussians (the Gaussian sum filter); approximating the first two moments of the density; evaluating the required probability density function over a grid in the state space. However, each of these techniques has to be extensively modified to tackle the particular problem in hand. For

example, methods that evaluate the probability density over a grid in the state space first require the grid to be specified, which is a non-trivial problem in a multi-dimensional space. To avoid misleading results, a large number of grid points will in general be necessary. In addition, a non-trivial computation must be performed at each point.

2.1. Sequential Monte Carlo (Particle filters)

Recall that the current state of knowledge is updated via equations 1, 2 and 3. We need a way of carrying out these integrals successively for $t = 1, 2, \dots$

Poor Man's Bayes: Rubin¹⁵ devised a simple way of obtaining an approximate sample from a Bayesian posterior distribution.

- Simulate a sample $\tilde{\theta}^1, \tilde{\theta}^2, \dots, \tilde{\theta}^n$ from $\pi(\theta)$.
- Calculate weights $q_i \propto L(\tilde{\theta}^i; y)$; $\sum q_i = 1$
- Sample n times (with replacement) from the discrete θ -distribution with

$$P(\theta = \tilde{\theta}^i) = q_i.$$

The resulting sample $\theta^1, \theta^2, \dots, \theta^n$ is an ‘‘approximate’’ sample from $\pi(\theta|y)$. The sample obtained is approximate in the sense that $n^{-1} \sum_{i=1}^n g(\theta^i)$ converges in probability to $E\{g(\theta)|y\}$, as $n \rightarrow \infty$.

The *Sampling Importance Resampling (SIR/particle filter)*^{10, 6} is based on Rubin's sampler. It proceeds as follows. Assume that you have a sample $(\theta_t^i)_{i=1, \dots, n}$ from $\pi(\theta_t|\mathbf{y}_{1:t})$:

- Sampling:** Independently simulate $\tilde{\theta}_{t+1}^i$, using the state transition density $\pi(\theta_{t+1}|\theta_t^i)$, for each $i = 1, \dots, n$,
- Importance:** Upon receipt of observation y_{t+1} , for each value $\tilde{\theta}_{t+1}^i$ calculate the corresponding likelihood $h(y_{t+1}|\theta_{t+1}^i)$. Denote the set of likelihood values, normalised to sum to 1, by $(q_{t+1}^i)_{i=1, \dots, n}$.
- Resampling:** Draw a random sample of size n from the discrete distribution taking values $(\tilde{\theta}_{t+1}^i)_{i=1, \dots, n}$ with probabilities $(q_{t+1}^i)_{i=1, \dots, n}$. This is an approximation to a sample from $\pi(\theta_{t+1}|\mathbf{y}_{1:t+1})$.

The algorithm can be thought of as propagating a swarm of particles in the underlying state space. At

time t the particles are assumed to be an approximate sample from the posterior distribution of θ_t , given the observations so far. At time $t+1$ each particle moves to a new location in the state space. The likelihood of this location given x_{t+1} is evaluated and a multinomial sample of particles is then drawn from the discrete distribution with *support points* given by the *particle locations* and *probabilities* proportional to the *likelihoods*. The process is a form of *Genetic Algorithm* where the ‘fitness’ of a speculative parameter value is proportional to its likelihood. In its simplest form the SIR filter has various weaknesses.

Outliers: The effect of an outlying observation is to produce a likelihood which is centered in the tail of the prior distribution. Since this tail is represented only sparsely by sample points in the SIR filter, an exceptionally large sample from the prior will be needed to yield a good support for the posterior distribution.

Sample Impoverishment: Lack of diversity: particles may be highly correlated, localised into a restricted region of parameter space, acting as one. The particle system may collapse to a singleton (extreme lack of diversity).

Track loss: Particles become trapped in ‘impossible’ regions of state space (evolutionary dead-ends).

Jittering: There are various *ad hoc* fixes for these problems. In order to alleviate the problem of sample impoverishment, Gordon *et al.*¹⁰ suggested adding a small amount of Gaussian noise, or jitter, to each sample point at each time step. If one point is replicated in the posterior r times, it is now replaced by r closely adjacent points. Jittering therefore smooths out the posterior density, using a Gaussian kernel. Choosing the jitter variance is thus equivalent to choosing the smoothing parameter in density estimation, and there is a corresponding variance/bias trade-off to be made. Standard rules of thumb can be used to choose the degree of smoothing.

Prior Boosting: This approach to sample depletion was originally proposed by Rubin¹⁵. At the prediction stage of the SIR filter, instead of generating the usual n points, we generate κn points. The likelihood of each of these is calculated, and then n

are resampled in the update step in the usual way. Typically $\kappa = 10$.

3. Fundamentals

Particle filters work by providing a discrete approximation to the PDF which can be easily updated to incorporate new information as it arrives. More generally our interest will be in approximations which consist of a set of random locations in the state space $(s^i)_{i=1,\dots,n}$, termed the *support*, and a set of associated weights $(m^i)_{i=1,\dots,n}$ summing to 1. The support and the weights together form a *random measure*.

The objective is to choose measures so that

$$\sum_{i=1}^n g(s^i) m^i \approx \int g(\theta) \pi(\theta) d\mu(\theta) \quad (4)$$

for typical functions g of the state space, in the sense that the left-hand side converges (in probability) to the right-hand side as $n \rightarrow \infty$.

The simplest example of a random measure is obtained by sampling $(s^i)_{i=1,\dots,n}$ independently from $\pi(\theta)$, and giving equal weights $m^i = n^{-1}$; $i = 1, \dots, n$. The estimate of the expected value of $g(\theta)$ is then the sample average $\sum_{i=1}^n g(s^i)/n$. Importance sampling provides a more general example by sampling $(s^i)_{i=1,\dots,n}$ from another PDF $f(y)$ and attaching importance weights $m^i = A\pi(s^i)/f(s^i)$, where $A^{-1} = \sum_{i=1}^n \pi(s^i)/f(s^i)$.

Before attempting to improve the SIR algorithm, it is worth emphasizing that our fundamental objective is to produce accurate Monte Carlo approximations to the *succession of integrals* that arise in Bayesian calculations. For accurate Monte Carlo integration, it is essential to eliminate unnecessary randomness and to make careful choices for proposals in importance sampling.

For example, the purpose of resampling is to produce a set of points with a histogram that approximates a particular probability mass function. The standard SIR algorithm achieves this with a multinomial sample $(N_i)_{i=1,\dots,n}$. But with the following algorithm the variables N_i never differ from their required expected value by more than 1.

Algorithm: Randomised circular sampling.

```

T = unif(0, n-1); j = 1; Q = 0; i = 0
do while T < 1
  if Q > T then

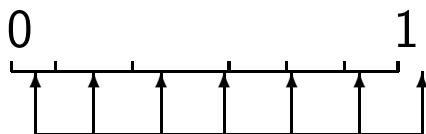
```

```

     $T = T + 1/n$ ; output  $s^i$ 
  else
    pick  $k$  in  $\{j, \dots, \ell\}$ 
     $i = s^k$ 
     $Q = Q + m^i$ 
    switch  $(s^k, m^k)$  with  $(s^j, m^j)$ 
     $j = j + 1$ 
  end if
end do

```

The algorithm treats the weights as contiguous intervals of $(0, 1)$. These intervals are randomly ordered, and the number of grid points $\{T + k/n\}$ in each interval is then counted. It randomly translates a ‘comb’ with equally spaced teeth as follows:



The objective of the previous sampler is to ensure that N_i has expected value nm_i for $i = 1, \dots, \ell$, while ensuring that the variances of the N_i are as small as possible. Crisan and Lyons⁵ proposed that each N_i should be chosen to be the integer part of nm_i plus a Bernoulli variable with probability equal to the fractional remainder. Liu and Chen¹² have a similar method where each N_i is again chosen to be the integer part of nm_i but with the addition of a multinomial variable based on the fractional remainders. They call their method *residual sampling*. In practice, these methods produce similar effects on sampling efficiency.

Since resampling introduces noise, this raises the question, when should we resample, and when should we carry forward the weights? The question has been addressed by Liu and Chen¹² who propose an *ad hoc* rule based on the variance of the weights $(\tilde{m}_i^i)_{i=1, \dots, n}$. In general, if the weights are roughly even, and the system noise is small compared to the variance of the posterior at the previous time step, then it is better not to resample. In particular, if there is no system noise, resampling is always inefficient.

3.1. Assessing sample depletion

To compare refinements of the SIR algorithm, it is helpful to have a measure of the effective sample size

(ESS). This is the sample size that would be required for a simple random sample from the target posterior density to achieve the same estimating precision as the random measure provided by the particle filter.

Liu¹² has suggested using $ESS = n/(1 + V)$, where V is the variance of the importance weights. The result should be used with caution, since in practice some properties of the state distribution may be estimated well, and some poorly. In general, the effective sample size will depend on the quantity being estimated and not just the weight distribution.

In principle, a Bayesian filter should be assessed by looking at its performance averaged over the population of trajectories generated by the system model. However, for non-linear problems it may happen that most of the trajectories are simple to filter and only a few are ‘difficult cases’. It is therefore helpful to see how the filter performs for typical examples of these difficult cases. The integrated correlation time in MCMC calculations in non-dynamic problems and the ESS play similar roles. Neither of these diagnostics is designed to check for convergence to the *right* distribution. A noisy biased filter may have a large ESS but the sample will not have come from the correct distribution. To check for bias, the proposed particle filter will need to be compared with filters which are known to perform correctly.

We should note that there is intermediate ground between resampling and carrying forward the weights. Resampling can be carried out using modified weights: for example, using modified weights proportional to the square root of the original, i.e. $w_t^i \propto \sqrt{m_t^i}$. The resampled points are then carried forward with weights proportional to m_t^i/w_t^i . Similar techniques have been proposed in MCMC sampling to avoid problems in sampling from highly peaked densities.

Although it is unrealistic to use MCMC to sample the posterior distribution of the complete state history, under certain circumstances MCMC moves can be introduced in particle filtering. These moves may be successful in preventing sample impoverishment. In general, to accommodate arbitrary transitions it is necessary to store the whole history of the process up to time t . As we shall see in the next example, this can be avoided if the transition kernel only depends on a fixed set of summary statistics, or only upon the last τ time points.

3.2. Example: Bearings only tracking

A classic example of non-linear filtering is *bearings only tracking*. An observer (either fixed or moving) observes the bearing of a moving ship. The bearing is the angle of the observation relative to a fixed direction. The crucial problem is that we are trying to reconstruct the two-dimensional coordinates of the ship from a single non-linear observation. This type of non-linearity in tracking problems usually causes difficulties for the Extended Kalman Filter.

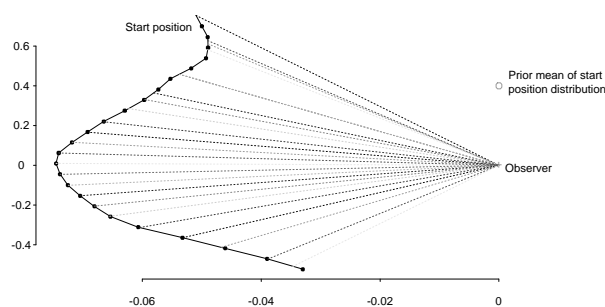


Fig. 1. Typical simulated trajectory. Dotted lines show observed bearings

We want to reconstruct the trajectory given the system model, observed bearings and a prior distribution on the initial position and velocity. In particular, suppose we observe t bearings. Notice that scaling the track toward the observer by a constant λ does not affect the likelihood since none of the angles change. It affects some of the parameters in a simple way. These factors can be incorporated into the filter by extending the *signature* of each particle. The MCMC scale move, when it is made, is a Gibbs move sampling from a truncated Gamma distribution.

3.3. Hidden Markov Models

By way of illustration we will work through a specific example. The problem is typical in the sense that the observation process is driven by a hidden Markov process.

Well-logs are records of the physical and mineralogical characteristics of underground rocks obtained by drilling in a region of geological interest. In traditional applications, a probe (called a sonde) is lowered into an existing well-bore by a cable, and

acoustical, electrical, nuclear-magnetic or thermal properties of the surrounding rock types are recorded as the sonde descends. In this example, the measurements are of nuclear magnetic response taken at 4500 time points. The underlying signal is piecewise constant; each constant segment relating to a stratum of a single rock type with constant physical properties. The jump discontinuities in the signal occur at times when a new rock stratum is first met.

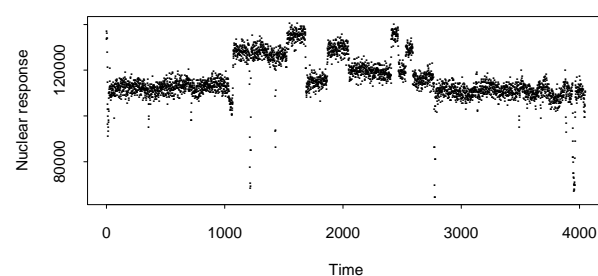


Fig. 2. The measurements of nuclear magnetic response taken at 4500 time points.

There is increasing interest in the possibility of ‘measurement-while-drilling’ (MWD) rather than the retrospective measurement of rock characteristics in existing boreholes. To detect changes in rock strata as drilling proceeds, data need to be collected from the vicinity of the drill-head. There are severe technical difficulties both in obtaining useful measurements and in the transmission of these data to the surface. Progress is currently being made with these problems in the gas and oil drilling industry. Other areas in which the use of traditional sondes is inadequate include the exploration of leakage below buried waste. These investigations are carried out by horizontal drilling making it impossible to lower a sonde into the borehole. Attachment of recording devices to the drill head may be the only way in which data can be collected – thus enabling drilling to be steered towards areas of high contamination.

3.3.1. Batch processing

The well-log data of Figure 2 have been analysed previously¹³. The whole dataset was batch-processed using a Gibbs (MCMC) sampler. Outliers were removed by hand and the number of change-points was

fixed prior to the analysis. It only remained to locate the change-points as accurately as possible.

When scanning the data as a whole (by eye) the detection of change-points appears straightforward. However when the data are only available incrementally, differentiating between outliers and true change-points is difficult. Successive MCMC sampling, even when outliers have been eliminated and number of change-points is known, is too time-consuming for real-time inference. By contrast, as we shall see, particle filter methods are computationally efficient and enable uncertainty about the number of change-points and outliers to be incorporated automatically.

3.3.2. On-line analysis

We use a hidden Markov model to model regime switching in the well-log data. The (underlying) state is the expected nuclear magnetic response for the current rock strata. The hidden Markov chain allows for both changes in the rock strata, and the possibility that the current measurements are outliers. The conjugacy in the assumed model means that conditional on knowing the history of the hidden Markov chain, the posterior distribution of the history of the measurable state can be calculated analytically using the Kalman filter.

The posterior distribution can be written as a mixture distribution, with each term in the mixture referring to a single possible value of the history of the hidden Markov chain. Liu and Chen¹² show that for such problems, the efficiency of the particle filter can be greatly improved if, instead of each particle representing a possible value of the history of the state, each particle represents a possible history of the hidden Markov chain (or a suitable summary of that history). This technique is called *marginalisation* or *collapsing*.

With such an approach, the posterior can be calculated exactly using a finite number of particles. Unfortunately, the number of particles needs to increase exponentially with the number of measurements, and becomes unfeasibly large for even small data sets (let alone the data set shown in Figure 2, where there are 4050 measurements). To restrict the number of particles used by the particle filter, resampling must be used. At each time stage a smaller, but hopefully representative, sample of par-

ticles are chosen from the large number of current particles^{16, 12, 7}.

We assume a two-dimensional Hidden Markov Model, with states $I_t = (S_t, O_t)$, where S_t and O_t both taking values in $\{1, 2\}$. Conditional on I_t , the underlying state (the expected nuclear magnetic response) satisfies

$$\theta_t = \begin{cases} \theta_{t-1} & \text{if } S_t = 1, \\ \mu + \sigma Z_t & \text{if } S_t = 2. \end{cases} \quad (5)$$

and the measurements satisfy

$$Y_t = \begin{cases} \theta_t + \tau_1 Z_t^* & \text{if } O_t = 1, \\ \nu + \tau_2 Z_t^* & \text{if } O_t = 2. \end{cases} \quad (6)$$

The error terms $\{Z_t, Z_t^*\}_{t=1, \dots}$ are uncorrelated, standard Gaussian random variables, and $\mu, \nu, \sigma, \tau_1, \tau_2$ are suitably chosen hyperparameters. The system equation (5) allows for jumps in the underlying signal, while the measurement equation (6) allows for clusters of outliers. Such a model produces the step function form for the underlying signal that is evident from the data.

A number of outliers below the main body of data are apparent. This motivated the model that we have used (see Equation 6). When the Markov chain, O_t , is in state 1, the observations will be modelled as the true state corrupted by additive noise. State 2 will represent an outlier state, and, for simplicity, the observation will be modelled as a draw from a Gaussian random variable whose parameters are independent of the true state. There are around 70 observations that appear to be outlying. These occur in 16 clusters. This suggests that suitable values of the transition probabilities would be approximately $P(O_t = 2 | O_{t-1} = 1) = 0.004$ and $P(O_t = 2 | O_{t-1} = 2) = 0.75$. The outlier distribution was taken to have a mean, ν , of 85000 and a standard deviation, τ_2 , of 12500. The standard deviation τ_1 of non-outlying observations was taken to be 2500.

Previous analyses¹³ have assumed additive Laplacian noise for the data. The analysis of the well-log data by a particle filter under such a model can be found in Fearnhead's thesis. More complicated models, which include more detailed modelling of the outliers, and allowing for correlated noise, were also considered there. For all these models the posterior distribution of the history of the state, θ_t , conditional on the history of the hidden state, I_t , could be calculated analytically.

Results: The main aim of analysing the well-log data is to detect the change-points in the signal on-line. So after processing each measurement, the probability of a jump having occurred during the last k time points was estimated. The results we present are for $k = 5$, but similar results were obtained with slightly different values of k .

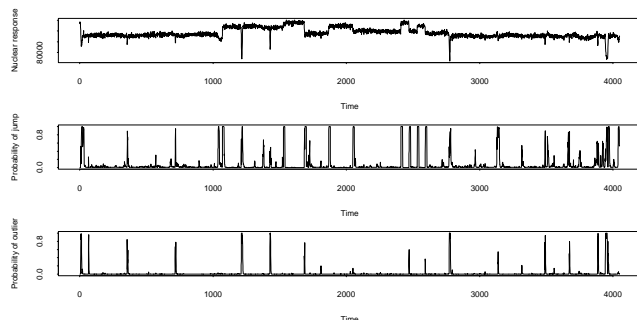


Fig. 3. Results of on-line analysis of the well-log data (top) by the new particle filter. The particle filter used 100 particles. The estimates of the probabilities of a recent change-point (middle), and the probability of the measurement being an outlier (bottom) are both shown.

The filter appears to have performed well, with all obvious change-points being given a posterior probability close to one. In a few cases, the filter appears to have misclassified outliers as change-points. An easier evaluation of the performance of the filter can be gained from looking at an estimate of the underlying signal (see Figure 4).

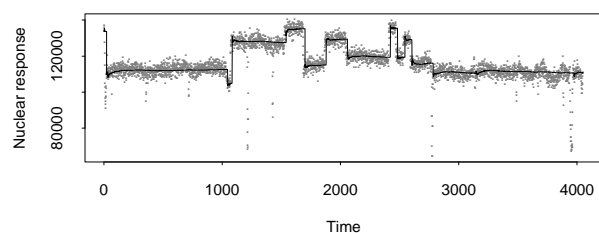


Fig. 4. An estimate of the underlying signal for the well-log data.

The estimate was obtained from the output of the particle filter. The change-points were fixed to be at times where the posterior probability of a recent jump was greater than 0.9, of which there were 16. The value of the state between each pair of adja-

cent change-points was estimated by the mean of all measurements in that time period which had negligible probability of being an outlier.

4. Using particle filters to analyse large datasets

In Bayesian statistical analysis, our aim is to find the posterior density $\pi(\theta|y_{1:N})$ of the parameter given the data $y_{1:N} = \{y_1, \dots, y_N\}$. The parameter θ may be of high dimension θ . In a standard non-dynamic (static) problem all the data $y_{1:N}$ are available at once, and we know that

$$\pi(\theta|y_{1:N}) \propto \pi(\theta)L(\theta; y_{1:N})$$

where $L(\theta; y_{1:N})$ is the likelihood. Markov chain Monte Carlo is one method of analysis.

To use MCMC we need to construct a Markov chain $(\theta^{(1)}, \theta^{(2)}, \dots)$ on the space of possible θ values with $\pi(\theta|y_{1:N})$ as its equilibrium. By running the chain for a long period of time, values from the equilibrium can be harvested and used to summarise the target distribution.

Problems: Suppose for example that the Metropolis sampler is used. At each step r in the Markov chain the current value of $\theta^{(r)}$ is modified by proposing a new value, θ , sampled from a proposal density $g(\theta|\theta^{(r)})$. The new value θ is accepted and becomes $\theta^{(r+1)}$ with acceptance probability

$$A(\theta^{(r)}, \theta; y_{1:N}) = \min \left\{ 1, \frac{g(\theta^{(r)}|\theta)\pi(\theta|y_{1:N})}{g(\theta|\theta^{(r)})\pi(\theta^{(r)}|y_{1:N})} \right\}.$$

The problem is that that $A(\theta^{(r)}, \theta; y_{1:N})$ depends on the whole of $y_{1:N}$. When the data set is massive, computing the acceptance probability is a non-trivial calculation, since it involves scanning through the whole dataset. When N is of the order of millions this can be a very time-consuming task, and furthermore the task has to be repeated until the MCMC algorithm has converged, which may take several thousand steps.

There have been various attempts to use particle filters for the Bayesian analysis of large datasets. The papers by Ridgeway and Madigan¹⁴ and Fearnhead⁸ provide a simple introduction.

4.1. Simple use of sub-sampling

The basic idea is to use a sub-sample of the data $y_{1:n}$ where $n \ll N$. If n is small enough then MCMC can

be run on the subsample, to yield $\{\theta_i\}, i = 1, \dots, M$, a sample of values from $\pi(\theta|y_{1:n})$. Each of these values then receives an importance weight w_i from the rest of the sample, given by

$$w_i = \frac{\pi(\theta_i|y_{1:N})}{\pi(\theta_i|y_{1:n})}.$$

A simplification occurs when observations are independent, since

$$\frac{\pi(\theta|y_{1:N})}{\pi(\theta|y_{1:n})} = \frac{\pi(\theta)L(\theta, y_{1:N})}{\pi(\theta)L(\theta, y_{1:n})} = L(\theta, y_{n+1:N}).$$

Similar simplifications occur when the observations have Markov dependence. The practical impact is that the remainder of the dataset only needs to be scanned once.

4.2. Successive sub-samples

Unfortunately, the set of weights produced by this procedure may be highly skewed and concentrated on only a few of the values in the set $\{\theta_i\}$. To remedy this Ridgeway and Madigan¹⁴ consider a succession of values of n , say n_1, n_2, \dots, N and apply a modified particle filter to the successively augmented datasets, proceeding as if these form a time series. The modified particle filter has two components, sampling/resampling and refreshment. MCMC transitions are introduced to refresh the particle support set. The decision on when to refresh is based on the distribution of particle weights. If the distribution is highly skewed then refreshment is carried out.

Unless the statistical model has special structure that can be exploited, these MCMC steps are computationally expensive. However, we expect that as the data are successively augmented, the distribution of particle weights will become less skewed, so moves are made less often. In Ridgeway and Madigan we see that the refresh times occur frequently at the beginning and less so toward the end of the data reading process.

4.3. Model-based clustering

Fearnhead's paper⁸ is about model-based clustering. The data are assumed to come from a mixture distribution where the distributions of the mixture components have some known parametric form. For example, it could be assumed that each observation is from one of K possible multivariate normal distributions. We don't know the means and covariances of

the distributions, how many different distributions there are or which distribution each observation is from.

The data are $y_{1:n} = \{y_1, \dots, y_n\}$. Under the model each y_i comes from one of the mixture components. For any given component the observations are considered to be independent. An assignment $z_{1:n} = \{z_1, \dots, z_n\}$ is a vector of component labels and k is the number of components identified, so $z_i \in \{1, \dots, k\}$. The component distributions have densities $f(y; \theta)$ where θ is different for each component. The joint density of these variables is

$$p(y_{1:n}, z_{1:n}, \theta_{1:k}) \propto \pi(z_{1:n}) \prod_{j=1}^k \pi(\theta_j) \prod_{i=1}^n f(y_i; \theta_{z_i}).$$

The Dirichlet prior $\pi(z_{1:n})$ is parametrised by α , with a recursive definition:

$$\pi(z_{i+1} = j | z_{1:i}) = \begin{cases} n_j / (i + \alpha) & \text{for } j = 1, \dots, k_i \\ \alpha / (i + \alpha) & j = k_i + 1 \end{cases}$$

where k_i is the number of clusters in the assignment $z_{1:i}$ and n_j is the number of observations that $z_{1:i}$ assigns to cluster j . With classical conjugate prior distributions for the parameters of a multivariate normal density some special simplifications occur. In particular and most importantly, once the assignment vector $z_{1:n}$ is known, it is possible to evaluate the posterior distribution of the parameters explicitly. The special form of the Dirichlet prior also leads to simplifications enabling the posterior probabilities of the mixture weights to be assessed when $z_{1:n}$ is known. So we can use a particle filter where each particle is tagged with its own assignment vector $z_{1:i}$ at stage i . See Fearnhead⁸ for further details.

5. Data sketching for large datasets

The purpose of using sequential statistical methods (particle filters) on static datasets is to reduce demands on data access. An entirely independent approach to related problems has been developed in the computer science literature. Key authors are Indyk¹¹, Cormode and Muthukrishnan⁴ and Flajolet⁹. The first three authors exploit an ingenious combination of random projections (using stable law distributions) and universal hashing³ to produce sketches of large datasets that enable questions concerning the distributional properties of the values in the dataset to be answered rapidly.

Flajolet⁹ develops an ingenious way of counting large numbers with a tiny amount of memory. This is related to the Additive-increase multiplicative-decrease processes studied by Bertoin-Biane-Yor¹.

5.1. Projection methods

The data come in a stream (no particular order), $(a_1, w_1), (a_2, w_2), \dots$ where $a_i \in A$ is the type of the i th item in the stream and w_i is the multiplicity. The problem is that the amount of data can be vast. How can you answer questions about the stream, for example, to find out how many different types there are? How many different users are there on the Internet?

We suppose that there is a pseudo-random mapping $h : A \rightarrow R$ such that

$$P(h(a) < x) = F_p(x),$$

where F_p is the distribution function of a symmetric stable distribution with parameter p , and where

$$P(|h(a) - h(b)| < \epsilon) = O(\epsilon), a \neq b.$$

(universal hash function)

Now calculate

$$S = \sum_{i=1}^n h(a_i)w_i = \sum_{j=1}^m x_j \sum_{i:h(a_i)=x_j} w_i = \sum_{j=1}^m x_j c_j,$$

and note that, using the property of stable distributions,

$$S \sim X \left(\sum_{j=1}^m |c_j|^p \right)^{1/p},$$

where X has a symmetric stable distribution with parameter p .

The *projection sketch* consists of R independent replicates of S . The median (for example) of the S values is then used to estimate the scaling term $\sum_{j=1}^m |c_j|^p$. When p is small this gives an estimate of the number of distinct items.

5.2. Other types of data sketches:

Sketches based on small p projections enable us to assess whether the profile of occurrences in two data streams is the same — just subtract the sketches. They also allow for removal of items (stock control). Techniques for maintaining histogram sketches are of particular interest for statistical applications.

6. Concluding remarks

The Bayesian analysis of massive datasets remains a challenging problem. MCMC methods are not feasible for these datasets. Particle filters are promising. They are particularly effective when

- distributional conjugacy can be exploited (c.f. Section 3.3.2),
- sufficient statistics are available, permitting occasional MCMC moves to be made at low computational cost (c.f. Section 3.2).

For complex problems, particles need to be tagged with extensive information. The design of efficient database management systems for these data is an open problem.

Data sketches have the potential for summarising both the data and the particle systems that represent the posterior distribution.

References

1. J. Bertoin, P. Biane and M. Yor, *Tech. Rep. PMA-705, Lab. de Probab., Univ. Paris VI* (2002).
2. J. Carpenter *et al.*, *IEE Proc. Radar Sonar Navigation* **146**, 2 (1999).
3. T. Cormen, C.E. Leiserson and R. Rivest, *Introduction to Algorithms*. MIT Press, London (1990).
4. G. Cormode and S. Muthukrishnan, *IEEE Trans. Know. Data Eng.* **15(3)**, 529 (2003).
5. D. Crisan and T. Lyons, *Probab. Th. and Rel. Fields* **109**, 217 (1997).
6. A. Doucet *et al.*, *Sequential Monte Carlo Methods in Practice*. Springer, NY.(2000).
7. P. Fearnhead and P. Clifford, *J. Royal Statist. Soc.* **65**, 887 (2003).
8. P. Fearnhead, *Statistics and Computing* **14**, 11 (2004).
9. P. Flajolet and G.N. Martin, *J. Comp. Sys. Sc.* **31**, 182 (1985).
10. N.J. Gordon *et al.*, *IEE Proc. F Radar Sig. Proc.* **140**, 107 (1993).
11. P. Indyk, *Proc. 40th Symp. Found. Comp. Sc.* 189-197 (2000).
12. J. Liu and R. Chen, *J. Amer. Statist. Assoc.* **93**, 1032 (1998).
13. J.K. O'Ruanaidh *et al.*, *Numerical Bayesian Methods*. Springer, NY.(1993).
14. G. Ridgeway and D. Madigan, *KDD02 Proc. 8th ACM SIGKDD*, 5 (2002).
15. D.B. Rubin, in *Bayesian Statistics*, Vol.3, Oxford University Press (1988).
16. J.K. Tugnait, *Automatica* **18**, 607 (1982).