Particle smoothing using Markov chain Monte Carlo sampling for backwards simulation

Pete Bunch, Simon Godsill, *Member, IEEE,*

***Abstract*—The abstract goes here.**

***Index Terms*—**

I. Introduction

THE objective of sequential Bayesian inference is the estimation of an unknown, time-varying quanitity from incomplete or inaccurate observations. This is commonly achieved through the use of probabilistic models for the evolution of the latent state and the measurement process.

$$x_k \sim p(x_k|x_{k-1}) \tag{1}$$
$$y_k \sim p(y_k|x_k) \tag{2}$$

Here, the random variable x_k denotes the value of the latent state at the k^{th} instant, and y_k the value of the observation. A set of random variables will be written as $x_{1:k} = \{x_1, x_2, \dots, x_k\}$. The state evolution is assumed to be Markovian, i.e. x_k depends only on the previous value, x_{k-1}.

Two problems are generally of interest; first, the inference of the *filtering* distribution, $p(x_k|y_{1:k})$, an estimate of the current state given all previous observations; second, that of the *smoothing* distribution, $p(x_{1:K}|y_{1:K})$ (where K is the number of time steps), an estimate of the entire state sequence given all the observations. Note that a related problem is that of estimating the *marginal smoothing* distributions, $p(x_k|y_{1:K})$ for each k, which is not considered in this paper.

In the case where the state and observation processes of equations 1 and 2 are linear and Gaussian, the filtering problem may be solved analytically using the Kalman filter [1]. For nonlinear or non-Gaussian models, tractable closed-form solutions are rare. In such cases, numerical approximations are often employed, including the particle filter algorithm, first introduced by [2]. (See [3], [4] for a thorough introduction.)

In a particle filter, the filtering distribution is approximated by a set of weighted samples (or "particles") drawn from that distribution.

$$\hat{p}(x_k|y_{1:k}) = \sum_i w_k^{(i)} \delta_{x_k^{(i)}}(x_k) \tag{3}$$

Here, $\delta_a(x)$ indicates a discrete probability mass at the point $x = a$. $\hat{p}$ means an approximation to p.

A similar story applies to the problem of smoothing. In the linear Gaussian case, the entire state sequence, $x_{1:K}$, may be estimated in closed-form using the a Rauch-Tung-Striebel (RTS) smoother [5]. This works by modifying the ordinary Kalman filter results in a backward processing pass through the observations. For nonlinear or non-Gaussian models, it is possible again to resort to numerical methods. Now, the particles of the approximation are drawn from the smoothing distribution.

$$\hat{p}(x_{1:K}|y_{1:K}) = \sum_i w_K^{(i)} \delta_{x_{1:K}^{(i)}}(x_{1:K}) \tag{4}$$

Each particle is an entire realisation of the state process, with K values corresponding to the state at each time step. The conventional method for generating these samples was described in [6]. In this paper a new method is presented for drawing samples from the smoothing distribution, using Markov chain Monte Carlo (MCMC). This novel procedure has computational advantages and greater flexibility over the previous methods.

In section **??**, we review briefly review the basic particle filter and smoother algorithms. The new MCMC-based smoothing method is presented in section **??**, with supporting simulations in section **??**.

II. Particle Filtering and Smoothing

A. The Particle Filter

The particle filter is a recursive numerical algorithm for estimation of the filtering distribution. At the k^{th} processing step, a particle estimate of the joint distribution over x_{k-1} and x_k is made by drawing samples from a proposal distribution.

$$\{x_{k-1}^{(i)}, x_k^{(i)}\} \sim q(x_x|x_{k-1})q(x_{k-1}) \tag{5}$$

The $k-1$ state proposal distribution, $q(x_{k-1})$, is constructed from the particles of the filtering distribution from the previous time step. The choice of weights and sampling procedure here determines what from of "resampling" is to be used. Here we use arbitrary weights for generality.

$$q(x_{k-1}) = \sum_i v_{k-1}^{(i)} \delta_{x_{k-1}^{(i)}}(x_{k-1}) \tag{6}$$

Note that the choice $v_{k-1}^{(i)} = w_{k-1}^{(i)}$ corresponds to ordinary resampling, and other choices of $v_{k-1}^{(i)}$ represent auxiliary sampling schemes [7]. For the least computational expense, the proposal distribution with weights $v_{k-1}^{(i)} = 1/N_P$ (where N_p is the number of particles) can be "sampled" by simply

P. Bunch and S. Godsill are with the Department of Engineering, Cambridge University, UK. email: {pb404,sjg30}@cam.ac.uk

Manuscript received January 01, 1901; revised January 02, 1901.

keeping the set of particles generated in the last step, with no resampling. (See [3], [4] for further discussion of resampling.)

Next, the particles are assigned an importance weight according to the ratio of the targeted joint distribution and the proposal.

$$\begin{aligned} w_k^{(i)} &= \frac{p(x_{k-1}^{(i)}, x_k^{(i)}|y_{1:k})}{q(x_x^{(i)}|x_{k-1}^{(i)})q(x_{k-1}^{(i)})} \\ &\propto \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})p(x_{k-1}^{(i)}|y_{1:k-1})}{q(x_x^{(i)}|x_{k-1}^{(i)})q(x_{k-1}^{(i)})} \\ &\approx \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_x^{(i)}|x_{k-1}^{(i)})} \times \frac{w_{k-1}^{(i)}}{v_{k-1}^{(i)}} \end{aligned} \tag{7}$$

Normalisation is enforced by scaling the weights so that they sum to 0. Finally, x_{k-1} is marginalised from the distribution by simply discarding the values from each particles.

Algorithm 1 Particle filter algorithm

Initialise particles from prior, $x_0^{(i)} \sim p(x_0)$.
for $k = 1 \dots K$ **do**
  for $i = 1 \dots N_P$ **do**
    Sample $\{x_{k-1}^{(i)}, x_k^{(i)}\} \sim \sum_j v_{k-1}^{(j)} q(x_k|x_{k-1}^{(j)})$.
    Weight $w_k^{(i)} \propto \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_x^{(i)}|x_{k-1}^{(i)})} \times \frac{w_{k-1}^{(i)}}{v_{k-1}^{(i)}}$.
    Discard $x_{k-1}^{(i)}$.
  end for
  Scale weights so that $\sum_i w_k^{(i)} = 1$.
end for

B. Existing Algorithms for Particle Smoothing

The particle filter generates a numerical approximation to each filtering distribution, $p(x_k|y_{1:k})$ by simulating a set of samples from each. A similar sampling-based method may be used to estimate the smoothing distribution.

The most basic particle smoother was presented in [**?**], and is a trivial extension of the particle filter. Rather than marginalising the previous states from each particle, these past values are stored as well. Thus, at each step the algorithm generates an approximation to $p(x_{1:k}|y_{1:k})$. The output from the final processing step is an approximation to the desired smoothing distribution.

The weakness of the Kitigawa smoother (KS) is degeneracy. Every time the particles are resampled, those with low weights do not appear in the approximation at the next step, while those with high weights appear multiple times. The result is that many, or even all, particles will have the same values at early time steps. This effect is illustrated in figure **??** in section **??**. If the objective is only to produce a single point estimate then this might be sufficient, but to characterise the complete smoothing distribution the approximation must be rejeuvenated.

Algorithm 2 Kitigawa smoother algorithm

Initialise particles from prior, $x_0^{(i)} \sim p(x_0)$.
for $k = 1 \dots K$ **do**
  for $i = 1 \dots N_P$ **do**
    Sample $\{x_{1:k-1}^{(i)}, x_k^{(i)}\} \sim \sum_j v_{k-1}^{(j)} q(x_k|x_{k-1}^{(j)})$.
    Weight $w_k^{(i)} \propto \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_x^{(i)}|x_{k-1}^{(i)})} \times \frac{w_{k-1}^{(i)}}{v_{k-1}^{(i)}}$.
  end for
  Scale weights so that $\sum_i w_k^{(i)} = 1$.
end for

- Kitigawa smoother
- Doucet re-weighting smoother
- Godsill re-sampling smoother
- Complexities
- Algorithms

III. New MCMC Particle Smoother

- Motivate - high complexity, low speed of other smoothers
- MCMC avoids need to calculate all weights at each step
- maths
- algorithm
- complexity

A. Performance Control

- The effect of varying M, the number of MH steps
- Situations where the new smoother works well/poorly.
- Backward sampling weight ESS.

IV. Simulations

V. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal Of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.6247&rep=rep1&type=pdf

[2] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107 –113, apr 1993.

[3] O. Cappé, S. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[4] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2009. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.772&rep=rep1&type=pdf

[5] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[6] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, 2004. [Online]. Available: http://pubs.amstat.org/doi/abs/10.1198/016214504000000151

[7] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. pp. 590–599, 1999. [Online]. Available: http://www.jstor.org/stable/2670179

PLACE PHOTO HERE

Pete Bunch Biography text here.

Simon Godsill Biography text here.