Adaptive Sequential MCMC for Combined State and Parameter Estimation

Zhanglong Cao, David Bryant, Matthew Parry

Department of Mathematics & Statistics, University of Otago, Dunedin, New Zealand

Abstract

We use a two-phase MCMC sampler for the estimation of the unknown state and parameters of a linear state space model. An approximation to the posterior density over the parameters is acquired in a learning phase, which is then used in a delayed-acceptance algorithm in the estimation phase. In online mode, the algorithm is adaptive to maintain sampling efficiency and uses a sliding window approach to maintain sampling speed.

Filtering for States and Parameters

Let $X_t = \{x_t, u_t\}$ be the unknown position and velocity of an object at time t and $Y_t = \{y_t = x_t + \varepsilon_t, v_t = u_t + \varepsilon'_t\}$ be the observed position and velocity, where $\varepsilon_t \sim N(0, \sigma^2)$ and $\varepsilon'_t \sim N(0, \tau^2)$, respectively.

The forward map for the states is based on an Ornstein-Uhlenbeck process,

$$\begin{cases} \mathrm{d}u_t = -\gamma u_t \mathrm{d}t + \lambda \mathrm{d}W'_t, \\ \mathrm{d}x_t = u_t \mathrm{d}t + \xi \mathrm{d}W_t, \end{cases} \tag{1}$$

so that γ^{-1} is roughly the time scale over which the velocity remains informative in the absence of subsequent observations. In our application, $\gamma^{-1} \approx 60$s.

The set of parameters to be estimated is $\theta = \{\gamma, \xi^2, \lambda^2, \sigma^2, \tau^2\}$.

The filtering for states is

$$p(X_t \mid Y_{1:t}) = \int p(X_t \mid Y_{1:t}, \theta) p(\theta \mid Y_{1:t}) \mathrm{d}\theta \tag{2}$$

and $p(\theta \mid Y_{1:t}) \propto p(Y_{1:t} \mid \theta) p(\theta)$.

The preceding generalises to more than one spatial dimension in a straightforward way.

Learning Phase

In the learning phase, a cheap Gaussian surrogate $\hat{p}(\theta)$ is obtained that will be used for a *delayed-acceptance Metropolis-Hastings* sampler [1] in the estimation phase. Specifically, a self-tuning random walk Metropolis-Hastings algorithm, in which the parameters are updated one at a time, is used to obtain the mean and covariance structure of $\hat{p}(\theta)$ [2]. The covariance structure is also used to inform the proposal density.

Sampling Efficiency

The optimal step size for the proposal density is found by maximising the sampling efficiency

$$\text{ESSUT} = \text{ESS}/T, \tag{3}$$

i.e. the *Effective Sample Size* (ESS) [3] per unit computing time T.

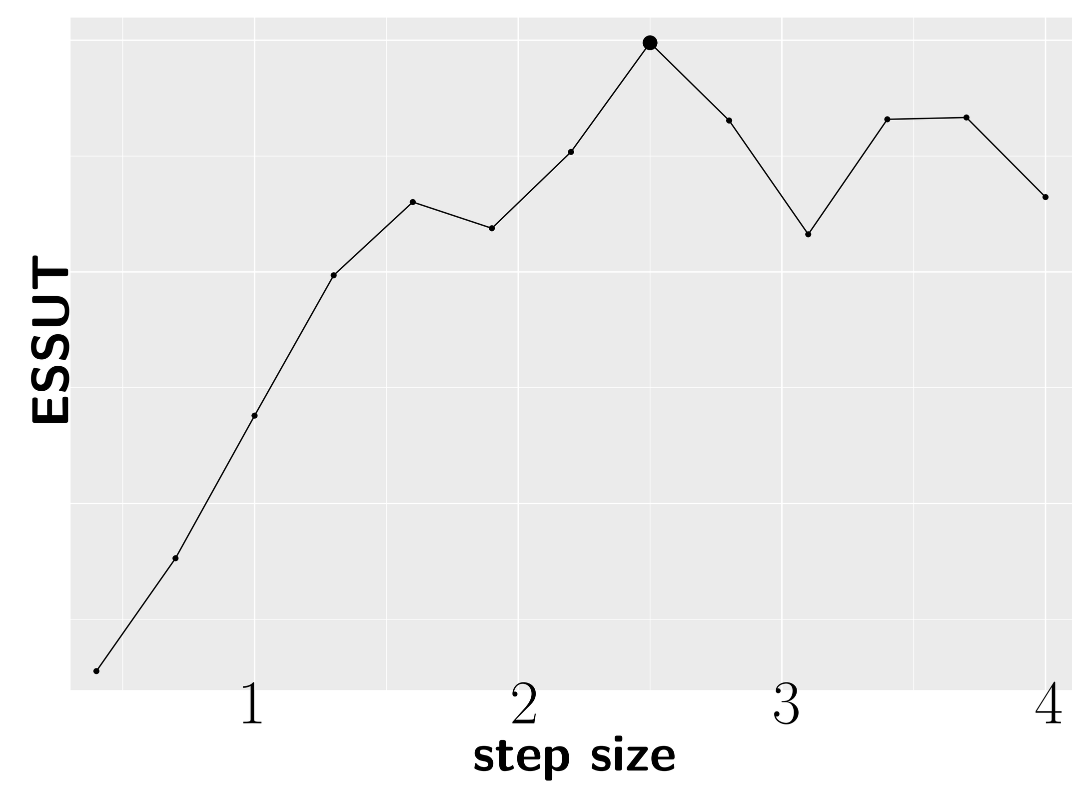


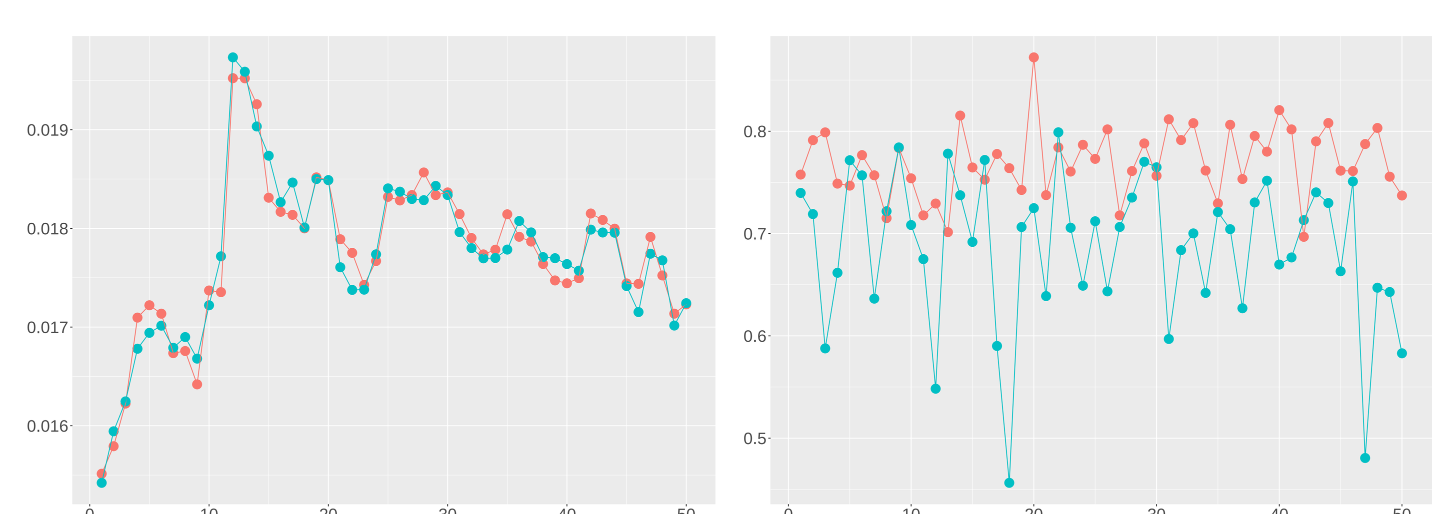
Figure: The optimal step size is 2.5. In contrast, the ESS is optimised with step size $2.4/\sqrt{d} \approx 1$ when $d = 5$ [4].

Estimation Phase

We use a Monte Carlo estimate of the target in (2):

$$p(X_t \mid Y_{1:t}) \doteq \frac{1}{N} \sum_{i=1}^{N} p(X_t \mid Y_{1:t}, \theta^{(i)}). \tag{4}$$

To maintain sampling speed, we introduce a *Sliding Window MCMC* algorithm that retains only the last L observations. This is also advantageous in our application as the parameters are often slowly varying in time. To maintain sampling efficiency, if the acceptance ratio α drops below a predefined threshold, the mean of $\hat{p}(\theta)$ is updated (the covariance is kept the same).



(a) Posterior mean of γ (b) Acceptance ratio α

Figure: Comparison over time of batch MCMC (orange) and sliding window MCMC (green) for γ and α. Differences are only minor.

Application

A tractor moving on an orchard is mounted with a GPS-enabled unit, which records and transmits data to a remote server. The data is an irregularly spaced time series of longitude, latitude, speed and bearing. In online mode, the sliding window algorithm is able to infer the tractor's position within seconds with an uncertainty of ≈ 0.5m. A window L of 100 is chosen to maintain a balance between sampling speed and acceptable estimation errors.

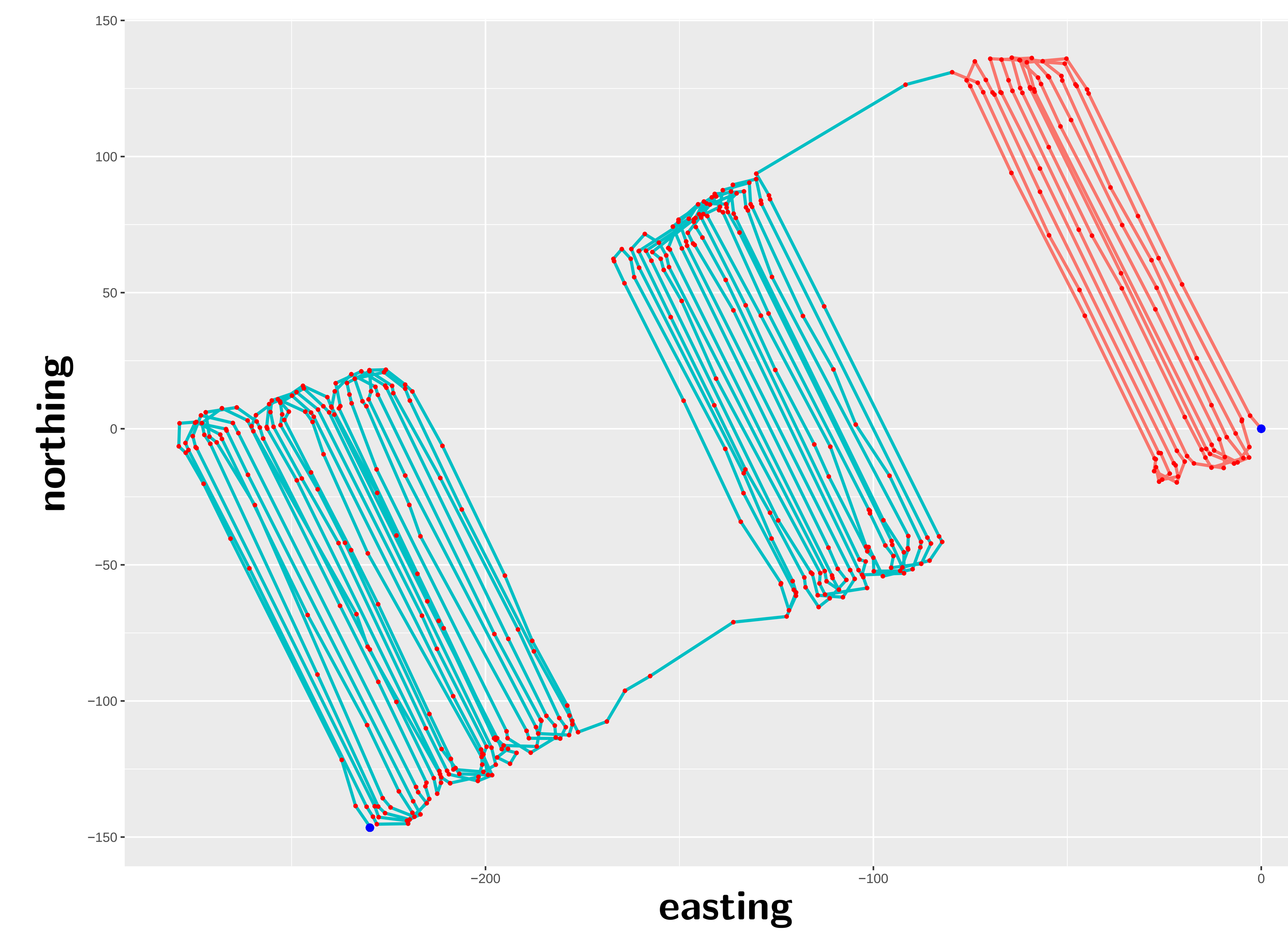


Figure: Posterior mean trajectory in two dimensions. The orange line is the reconstruction from the learning phase and the green line is the filtering from the estimation phase. Red dots are the measurements, and the two blue dots are the start and end points.

References

[1] J Andrés Christen and Colin Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005.

[2] Rui Vieira and Darren J Wilkinson. Online state and parameter estimation in dynamic generalised linear models. *arXiv preprint arXiv:1608.08666*, 2016.

[3] Christian P Robert. *Monte Carlo methods*. Wiley Online Library, 2004.

[4] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

Acknowledgements

This work was funded by grant UOOX1208 from the Ministry of Business, Innovation & Employment (NZ). GPS data was provided by TracMap (NZ).