Replica Conditional Sequential Monte Carlo

Alexander Y. Shestopaloff and Arnaud Doucet

The Alan Turing Institute

ICML 2019, June 13th 2019

State space models

We would like to model the distribution of an observed sequence $y_{1:T} = (y_1, \ldots, y_T)$.

- In the state space framework, the Y_t are drawn from an observation density $g(y_t|x_t, \theta)$.
- X_t is an unobserved Markov process with initial density $\mu(x_1|\theta)$ and transition density $f(x_t|x_{t-1}, \theta)$.

This talk will focus on inferring the realized values of the Markov process $x_{1:T} = (x_1, \ldots, x_T)$, assuming that θ is known.

State space models

State space models are a very widely used class of models. Some examples where state space models have been successfully applied are

- Stochastic volatility models, e.g. Guarniero, Lee and Johansen (2016).
- Population dynamics models, e.g. Finke et al (2017).
- Partially observed queueing systems, Shestopaloff and Neal (2013).
- Oceanography, e.g. modelling variations in global sea levels, Markos et al (2015).
- Computational neuroscience, e.g. decoding neural spike train data (Paninski et al (2010)).

Bayesian inference for state space models

In a Bayesian approach, we infer $x_{1:T}$ by sampling from the posterior density of $x_{1:T}$ given $y_{1:T}$,

$$p(x_{1:T}|y_{1:T}) \propto \mu(x_1)g(y_t|x_t)\prod_{t=2}^{T} f(x_t|x_{t-1})g(y_t|x_t).$$

This sampling problem has no exact solution, except for linear Gaussian models or models with a finite state space.

- In these cases, we can use the Kalman filter or the forward-backward algorithm to compute posterior marginals.

For general, i.e. non-linear, non-Gaussian cases, approximate methods such as Markov Chain Monte Carlo (MCMC) must be used.

MCMC with replicas of state

Running a Markov chain on multiple copies of a space has previously been used to improve MCMC, e.g. parallel tempering, also see Leimkuhler et al (2018).

Sharing information between different replicas can improve exploration of the space.

For our scenario, the replica target is a product density over K copies of the latent space, for some $K > 2$,

$$\bar{\pi}\left(x_{1:T}^{(1)}, \ldots., x_{1:T}^{(K)}\right) = \prod_{k=1}^{K} p\left(x_{1:T}^{(k)} | y_{1:T}\right).$$

We can draw samples from $\bar{\pi}$ by updating each replica in turn.

- This is computationally more expensive but can be beneficial in practice.

The replica cSMC sampler

Consider updating replica k, with the other replicas fixed.

Key idea: For each replica $x_{1:T}^{(k)}$, use

$$x_{t+1}^{(-k)} = (x_{t+1}^{(1)}, \ldots, x_{t+1}^{(k-1)}, x_{t+1}^{(k+1)}, \ldots, x_{t+1}^{(K)})$$

to construct an estimate of the backwards information filter $\hat{p}^{(k)}(y_{t+1:T}|x_t)$.

Then, use iterated cSMC with the sequence of targets

$$\hat{p}^{(k)}\left(x_{1:t}|y_{1:T}\right) \propto p\left(x_{1:t}|y_{1:t-1}\right) \hat{p}^{(k)}(y_{t+1:T}|x_t)$$

to update replica $x_{1:T}^{(k)}$. The optimal proposal at $t \geq 2$ now is

$$q_t^{\text{opt}}(x_t|x_{t-1}) \propto g(y_t|x_t) f(x_t|x_{t-1}) \hat{p}^{(k)}\left(y_{t+1:T}|x_t\right).$$

- The full update consists of updating all replicas in turn.

Estimating the backward information filter

The replica cSMC sampler requires an estimator $\hat{p}^{(k)}\left(y_{t+1:T}|x_t\right)$ of the backward information filter based on $x_{t+1}^{(-k)}$.

We propose to use a Monte Carlo approximation built using the other replicas,

$$\hat{p}^{(k)}\left(y_{t+1:T}|x_t\right) \propto \sum_{j\neq k} \frac{f\left(x_{t+1}^{(j)}|x_t\right)}{p\left(x_{t+1}^{(j)}|y_{1:t}\right)}.$$

Here, $p\left(x_{t+1}|y_{1:t}\right)$ denotes the predictive density of x_{t+1}.

- In practice, the predictive is unknown, and we also need to approximate it with some $\hat{p}(x_{t+1}|y_{1:t})$.
- However, this turns out to be easier.

Approximating the predictive density

- If we have informative observations, the posterior will tend to be much more concentrated than the predictive.

- We can approximate the predictive by its mean with respect to the posterior density,

$$
\begin{aligned}
&\int \frac{f\left(x_{t+1}|x_t\right)}{p\left(x_{t+1}|y_{1:t}\right)} p\left(x_{t+1}|y_{1:T}\right) dx_{t+1} \\
&\approx \frac{\int f\left(x_{t+1}|x_t\right) p\left(x_{t+1}|y_{1:T}\right) dx_{t+1}}{\int p\left(x_{t+1}|y_{1:t}\right) p\left(x_{t+1}|y_{1:T}\right) dx_{t+1}} \\
&\approx \frac{\frac{1}{K}\sum_{k=1}^{K} f\left(x_{t+1}^{(k)}|x_t\right)}{\frac{1}{K}\sum_{k=1}^{K} p\left(x_{t+1}^{(k)}|y_{1:t}\right)}.
\end{aligned}
$$

Approximating the predictive density

Using a constant approximation can reduce the variance of the mixture weights. Suppose the predictive is $\mathcal{N}(\mu, \sigma_0^2)$ and the posterior is $\mathcal{N}(0, \sigma_1^2)$, where $\sigma_1^2 < \sigma_0^2$. Then,

$$\begin{aligned}
&\operatorname{Var}\left(\frac{1}{p(x_{t+1}|y_{1:t})}\right) \\
&= \frac{2\pi\sigma_0^2}{\sqrt{2\sigma_1^2\nu_1}} \exp\left[\mu^2\left(\frac{1}{\sigma_0^2} + \frac{1}{(\sigma_0^2)^2\nu_1}\right)\right] \\
&- \frac{2\pi\sigma_0^2}{\sigma_1^2\nu_2} \exp\left[\mu^2\left(\frac{1}{\sigma_0^2} + \frac{1}{(\sigma_0^2)^2\nu_2}\right)\right].
\end{aligned} \tag{1}$$

where

$$\nu_1 = \left(\frac{1}{2\sigma_1^2} - \frac{1}{\sigma_0^2}\right) \qquad \nu_2 = \left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_0^2}\right). \tag{2}$$

- The weight variance can grow quickly with the difference of predictive and posterior means.
- This can reduce the effective number of replicas used.

Examples - Latent Process

$X_1 \sim \mathcal{N}(0, \Sigma_{\mathrm{b}})$, $X_t | \{X_{t-1} = x\} \sim \mathcal{N}(\Phi x, \Sigma)$.

Here, $X_t = (X_{1,t}, \dots, X_{d,t})'$, $\sigma^2_{\mathrm{b},i} = 1/(1-\phi_i^2)$ and

$$\Phi = \begin{pmatrix} \phi_1 & 0 & \cdots & 0 \\ 0 & \phi_2 & \ddots & \vdots \\ \vdots & \ddots & \phi_{d-1} & 0 \\ 0 & \cdots & 0 & \phi_d \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \rho \\ \rho & \cdots & \rho & 1 \end{pmatrix},$$

$$\Sigma_{\mathrm{b}} = \begin{pmatrix} \sigma^2_{\mathrm{b},1} & \rho\sigma_{\mathrm{b},1}\sigma_{\mathrm{b},2} & \cdots & \rho\sigma_{\mathrm{b},1}\sigma_{\mathrm{b},d} \\ \rho\sigma_{\mathrm{b},2}\sigma_{\mathrm{b},1} & \sigma^2_{\mathrm{b},2} & \ddots & \vdots \\ \vdots & \ddots & \sigma^2_{\mathrm{b},d-1} & \rho\sigma_{\mathrm{b},d-1}\sigma_{\mathrm{b},d} \\ \rho\sigma_{\mathrm{b},d}\sigma_{\mathrm{b},1} & \cdots & \rho\sigma_{\mathrm{b},d}\sigma_{\mathrm{b},d-1} & \sigma^2_{\mathrm{b},d} \end{pmatrix}.$$

Example 1: A Linear Gaussian Model

We use the latent autoregressive process as described previously.

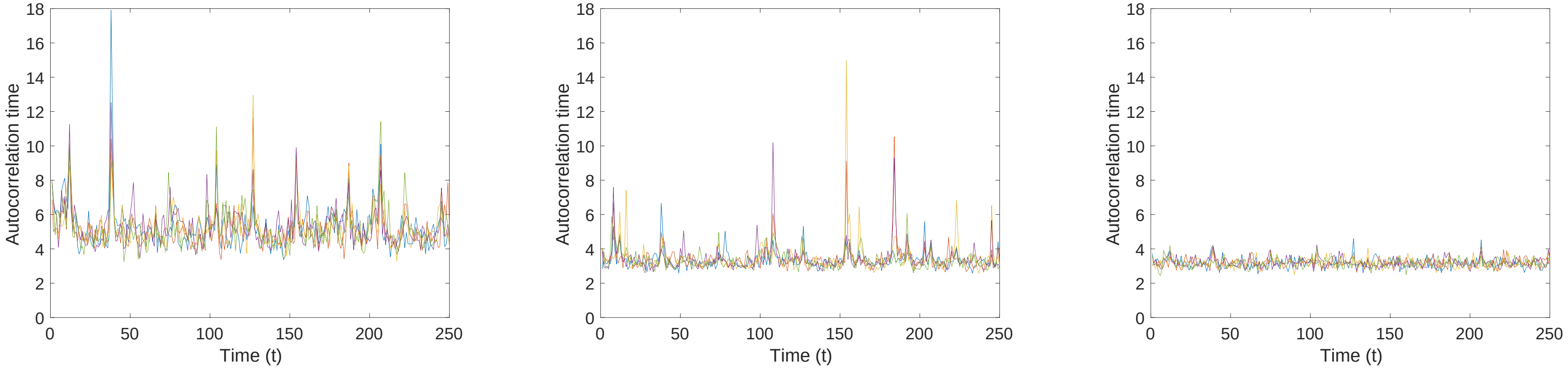
The observation process is $Y_{i,t}|\{X_{i,t} = x_{i,t}\} \sim \mathcal{N}(x_{i,t}, 1)$ for $i = 1, \ldots, d$, $t = 1, \ldots, T$.

We set $T = 250, d = 5$ and the model's parameters to $\rho = 0.7$ and $\phi_i = 0.9$ for $i = 1, \ldots, d$.

Example 1. A Linear Gaussian Model

We use this model to investigate the effects of the following.

1. Increasing the number of replicas K.
2. Using a constant approximation to the predictive density, since it can be computed exactly.



(a) 2 replicas.

(b) 75 replicas.

(c) 75 replicas, constant predictive.

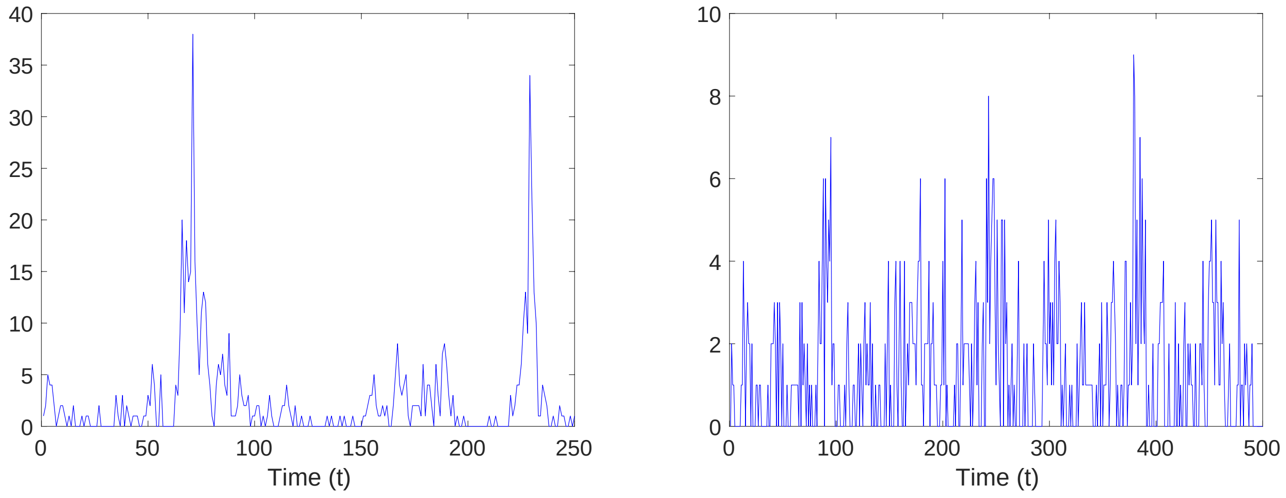
Figure 1: Estimated autocorrelation times for each latent variable. Different coloured lines correspond to different latent state components.

Example 2. Two Benchmark Models

We use the same autoregressive latent process as earlier.

Model 1: $T = 250$, $d = 10$ and $Y_{i,t}|\{X_{i,t} = x_{i,t}\} \sim \text{Poisson}(\exp(c + \sigma x_{i,t}))$ where $c = -0.4$ and $\sigma = 0.6$.

Model 2: $T = 500$, $d = 15$ and $Y_{i,t}|\{X_{i,t} = x_{i,t}\} \sim \text{Poisson}(\sigma|x_{i,t}|))$ where $\sigma = 0.8$.

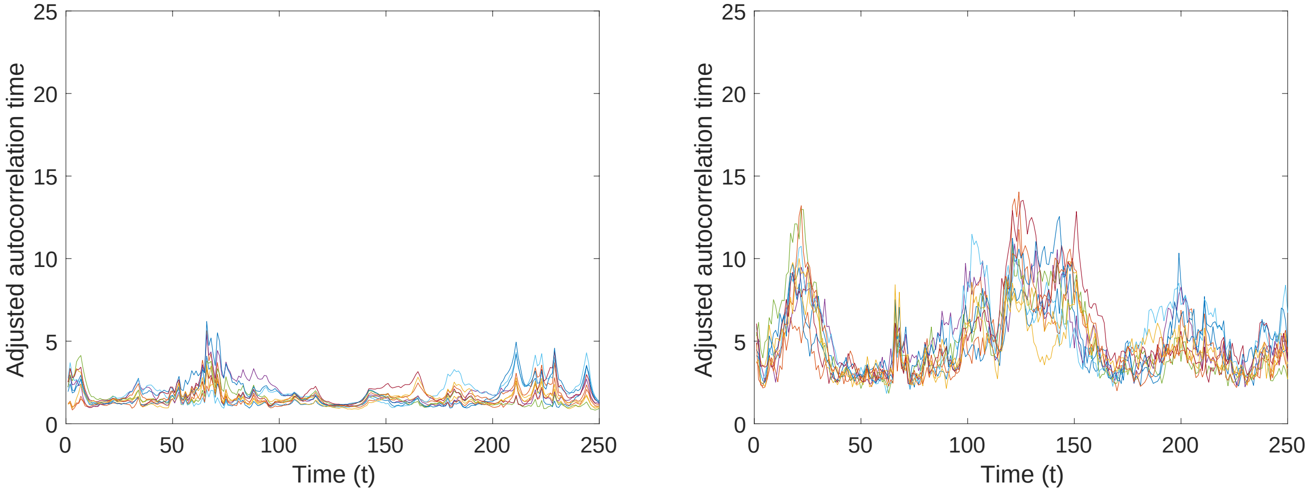


(a) Data for Model 1, $i = 1$. (b) Data for Model 2, $i = 1$.

Figure 2: Simulated data from the Poisson-Gaussian models.

Example 2. Two Benchmark Models

- For model 1, we use replica cSMC with two replicas, and update one replica conditional on the other.
- We compare to the best method in Shestopaloff and Neal (2018).



(a) Iterated cSMC with Metropolis.

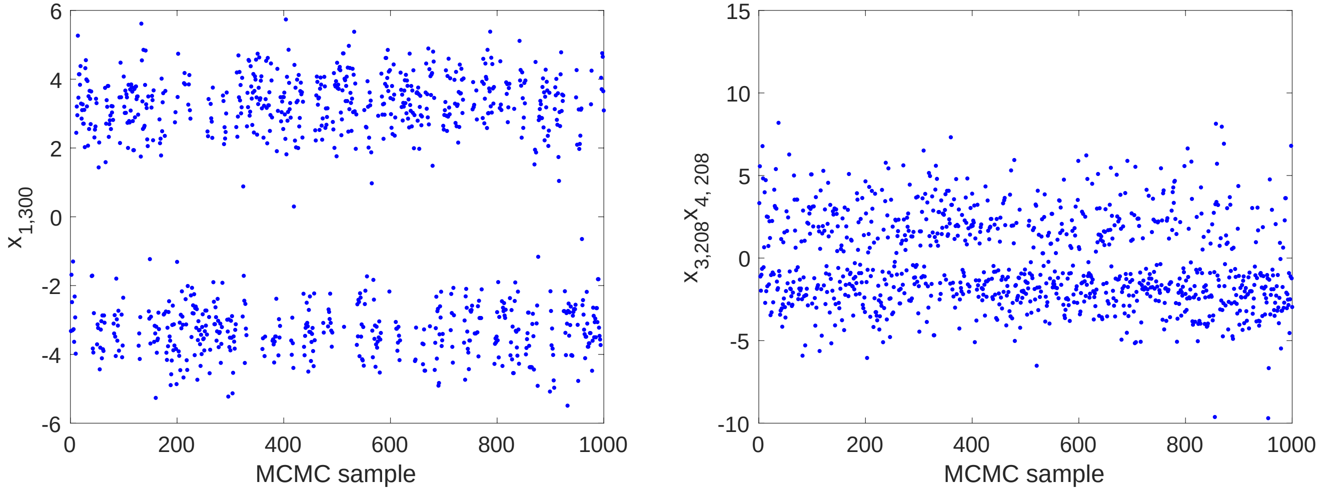
(b) Replica cSMC.

Figure 3: Model 1. Estimated autocorrelation times for each latent variable, adjusted for computation time. Different coloured lines corresponds to different latent state components.

Example 2. Two Benchmark Models

- For this model, the challenge is to move between the many different modes of the latent state.

- We use a total of 15 replicas and update 14 of the 15 replicas with iterated cSMC and one replica with replica cSMC.



(a) Trace plot for $x_{1,300}$. (b) Trace plot for $x_{3,208}x_{4,208}$.

Figure 4: Model 2. Replica + ordinary iterated cSMC.

Good performance relies on replicas being well-distributed.

Future Work

- Are the other ways to use to estimate the predictive density, i.e. improvement on using a constant, without resulting in mixture weights with high variance?
- How do we improve the estimate of the backward information filter in the multimodal case?
- How do we choose the number of replicas? Better guidance needed for this.
- Can we apply these methods to scenarios that have a sequential structure but do not involve time series?