particle MCMC as a pseudo-marginal method for exact-approximate Bayesian inference

MVE187-MSA101 “Computational methods for Bayesian statistics”, 2022

Umberto Picchini
@uPicchini, picchini@chalmers.se

Chalmers University of Technology and University of Gothenburg
Sweden

Missing file

There was a file missing that is needed to run my demo_bootstrap.m and demo_pmcmc.m. Now `resampling.m` has been added on Canvas (performs resampling with replacement for the bootstrap filter).

Goals

- We have previously obtained likelihood approximations for state-space models.
- We constructed particle MCMC to perform Bayesian inference for the parameters using an approximate likelihood.
- We are going to look at the surprising results showing that we are actually doing **exact Bayesian** inference.
- More topics may follow (if time allows) in a different set of slides, ie ABC (approximate Bayesian computation).

When we constructed the bootstrap filter, essentially we found that

$$\hat{p}(y_t|y_{1:t-1};\theta) = \frac{1}{N}\sum_{i=1}^{N} w_t^i = \frac{1}{N}\sum_{i=1}^{N} p(y_t|x_t^i), \qquad x_t^i \sim p(x_t|x_{t-1}), i = 1,...,N$$

this implying the approximate likelihood:

$$\hat{p}(y_{1:T}|\theta) = \prod_{t=1}^{T} \hat{p}(y_t|y_{1:t-1};\theta) = \prod_{t=1}^{T} (\frac{1}{N}\sum_{i=1}^{N} w_t^i)$$

and of course this means that $\hat{p}(y_{1:T}|\theta)$ is a random draw (repeated evaluations of $\hat{p}(y_{1:T}|\theta^*)$ will give different results at the same θ^* value).

This unlike the true $p(y_{1:T}|\theta^*)$ which will deterministically give you the same value at a given θ^* value.

So what are we really sampling from?

When producing $\hat{p}(y_{1:T}|\theta)$ how many things have been simulated from some probability distribution?

Surely the particles are randomly simulated. And these particles are functions of pseudo-random numbers. Say we call ξ these pseudo-random numbers.

Here follow examples.

Our basic example:

$$\begin{cases} y_t = x_t + \epsilon_t^{(1)}, & \epsilon_t^{(1)} \sim_{iid} N(0, 0.3^2) \\ x_t = x_{t-1} + \epsilon_t^{(2)}, & \epsilon_t^{(2)} \sim_{iid} N(0, 1) \end{cases}$$

Here $\epsilon_t^{(1)} \sim_{iid} N(0, 0.3^2)$ can alternatively be thought as $0.3 \cdot \xi_t$, where $\xi_t \sim N(0, 1)$.

So, in practice to simulate the model we only need access to standard Gaussian samplers.

Example: (this is finally a nonlinear model)

$$\begin{cases} x_t = 0.5x_{t-1} + 25\frac{x_{t-1}}{(1+x_{t-1}^2)} + 8\cos(1.2(t-1)) + N(0,q), \\ y_t = 0.05x_t^2 + N(0,r), \end{cases}$$

or alternatively written as

$$\begin{cases} x_t = 0.5x_{t-1} + 25\frac{x_{t-1}}{(1+x_{t-1}^2)} + 8\cos(1.2(t-1)) + \sqrt{q}\cdot\xi_t^{(1)} \\ y_t = 0.05x_t^2 + \sqrt{r}\cdot\xi_t^{(2)} \end{cases}$$

with $\xi_t^{(1)} \sim N(0,1)$ and $\xi_t^{(2)} \sim N(0,1)$.

What about other distributions?

It is remarkable that to simulate from very many distributions it is enough to be able to sample from the uniform U(0,1) distribution and the standard N(0,1) distribution.

For example, the inverse transform theorem shows that if X is a continuous random variable with an invertible cdf $F_X(x)$, then you can sample an x from its distribution by using

$$x := F_X^{-1}(u), \qquad u \sim U(0,1)$$

Ex: say we want to sample from the exponential Exp(λ) distribution. We know $F_X(x) = 1 - e^{-\lambda x}$ for $x > 0$, then since any cdf is uniform distributed $U := F_X(x) \sim U(0,1)$, we set $U = 1 - e^{-\lambda x}$ meaning that $x = -\frac{1}{\lambda}\log(1-U)$.

So if you draw a $u^* \sim U(0,1)$, then $x^* = -\frac{1}{\lambda}\log(1-u^*)$ is $x^* \sim \text{Exp}(\lambda)$

The inverse transform theorem is very general. It implies that if a random variable X has an invertible cdf then you only need uniforms in (0,1) to simulate from the distribution of X.

Another example (check Wikipedia): the Box-Muller method shows that to generate N(0,1) random draws, you can make use of uniform draws.

I could go on and on but you got the gist. Often, to simulate even complex systems, if you go look into the details, these complex systems are (possibly complicated) transformations of very basic random numbers such as N(0,1) and U(0,1).

Therefore, our particles x_t^i are generally functions of "simple" random numbers that we collectively denote with ξ, where often ξ are N(0,1) or U(0,1).

So for our simple model, certainly

$$\xi = \left((\xi_1^{(1)}, \xi_1^{(2)}), ..., (\xi_T^{(1)}, \xi_T^{(2)}), ...?\right)$$

and what other random variates can we also include?

Resampling! Performing resampling involves the generation of uniform pseudo-random numbers so we finally have

$$\xi = \Big((\xi_1^{(1)}, \xi_1^{(2)}), ..., (\xi_T^{(1)}, \xi_T^{(2)}),$$
plus ALL the pseudo-random numbers produced during resampling$\Big)$

In particle MCMC, Metropolis-Hasting actually samples from an artificially extended posterior

$$\hat{\pi}(\theta, \xi | y_{1:T}) \propto \hat{p}(y_{1:T} | \xi, \theta) \cdot p(\xi) \cdot \pi(\theta)$$

1. current value is θ_r, propose a new $\theta^* \sim q(\theta^* | \theta_r)$, e.g. $\theta^* \sim N(\theta_r, \Sigma_\theta)$ for some covariance matrix Σ_θ.
2. Sample $\xi^* \sim p(\xi)$ (useful for propagation and resampling)
3. compute

$$A = \frac{\hat{p}(y_{1:T} | \xi^*, \theta^*)}{\hat{p}(y_{1:T} | \xi_r, \theta_r)} \times \frac{p(\xi^*)}{p(\xi_r)} \times \frac{\pi(\theta^*)}{\pi(\theta_r)} \times \frac{q(\theta_r | \theta^*)}{q(\theta^* | \theta_r)}$$

   Draw a uniform $u \sim U(0,1)$ and if $u < A$ **accept** (θ^*, ξ^*) and set $(\theta_{r+1}, \xi_{r+1}) := (\theta^*, \xi^*)$.
   Otherwise, **reject**, and set $(\theta_{r+1}, \xi_{r+1}) := (\theta_r, \xi_r)$.
4. Set $r := r + 1$, go to 1 and repeat.

However in practice we do not need to bother storing the ξ, as we are not interested in those.

If we only keep the sampled θ and disregard the ξ, we are implicitly sampling from

$$\hat{\pi}(\theta|y_{1:T}) = \int \hat{\pi}(\theta, \xi|y_{1:T}) d\xi$$

which is the marginal of $\hat{\pi}(\theta, \xi|y_{1:T})$ and the object we are **actually** interested in sampling from.

Incredible result

Quite astonishingly Andrieu and Roberts[1] proved that using an *unbiased and non negative* estimate of the likelihood function into the MCMC routine is *sufficient* to obtain exact Bayesian inference for θ

That is using the Metropolis-Hastings acceptance probability

$$\min\left\{1, \frac{\hat{p}(y_{1:T}|\xi^*,\theta^*)}{\hat{p}(y_{1:T}|\xi,\theta)} \times \frac{p(\xi^*)}{p(\xi)} \times \frac{\pi(\theta^*)}{\pi(\theta)} \times \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}\right\}$$

will return a Markov chain with stationary distribution $\pi(\theta|y_{1:T})$ *regardless the* ***finite*** *number N of particles used to approximate the likelihood!*.

The good news is that $\mathbb{E}_\xi(\hat{p}(y_{1:T}|\xi,\theta)) = p(y_{1:T}|\theta)$ with $\hat{p}(y_{1:T}|\xi,\theta)$ obtained via SMC.

[1] Andrieu and Roberts (2009), Annals of Statistics, 37(2) 697–725.

The previous result is, in my opinion, one of the most important statistical results of the last 30 years.

In fact, it offers an "exact-approximate" approach, where because of computing limitations we can only produce $N < \infty$ particles, while still be reassured to obtain *exact* (Bayesian) inference under minor assumptions.

But let's give a rapid (technically informal) look at why it works.

Key result: unbiasedness (del Moral 2004[2])
We have that

$$\mathbb{E}_{\xi}(\hat{p}(y_{1:T}|\xi,\theta)) = \int \hat{p}(y_{1:T}|\theta,\xi)p(\xi)d\xi = p(y_{1:T}|\theta)$$

with $\xi \sim p(\xi)$ vector of *all* random variates generated during SMC (both to *propagate forward* the state and to perform particles resampling).

---

[2]Easier to look at Pitt, Silva, Giordani, Kohn. J. Econometrics 171, 2012 or page 87 of Naesseth's PhD thesis 2018.

To prove the exactness of the approach we look at the (easier and less general) argument in sec. 2.2 of Pitt, Silva, Giordani, Kohn. J. Econometrics 171, 2012 or my even more introductive blog post.

To simplify the notation take $y := y_{1:T}$.

- $\hat{\pi}(\theta, \xi|y)$ approximate joint posterior of (θ, ξ) obtained via SMC

$$\hat{\pi}(\theta, \xi|y) = \frac{\hat{p}(y|\theta, \xi)p(\xi)\pi(\theta)}{p(y)}$$

(notice ξ and θ are assumed a-priori independent)

Notice we put $p(y)$ not $\hat{p}(y)$ at the denominator: this follows from the unbiasedeness assumption as we obtain $\int\int \hat{p}(y|\theta, \xi)p(\xi)\pi(\theta)d\xi d\theta = \int \pi(\theta)\{\int \hat{p}(y|\theta, \xi)p(\xi)d\xi\}d\theta = \int \pi(\theta)p(y|\theta)d\theta = p(y)$.

The exact (unavailable) posterior of θ is

$$\pi(\theta|y) = \frac{p(y|\theta)\pi(\theta)}{p(y)}$$

therefore the marginal likelihood (evidence) is

$$p(y) = \frac{p(y|\theta)\pi(\theta)}{\pi(\theta|y)}$$

and

$$\begin{aligned}\hat{\pi}(\theta, \xi|y) &= \frac{\hat{p}(y|\theta, \xi)p(\xi)\pi(\theta)}{p(y)} \\ &= \frac{\pi(\theta|y)\hat{p}(y|\theta, \xi)p(\xi)\cancel{\pi(\theta)}}{p(y|\theta)\cancel{\pi(\theta)}}\end{aligned}$$

Now, we know that applying an MCMC targeting $\hat{\pi}(\theta, \xi|y)$ then discarding the output pertaining to ξ corresponds to *integrate-out* ξ from the posterior

$$\int \hat{\pi}(\theta, \xi|y)d\xi = \frac{\pi(\theta|y)}{p(y|\theta)} \underbrace{\int \hat{p}(y|\theta, \xi)p(\xi)d\xi}_{\mathbb{E}(\hat{p}(y|\theta))=p(y|\theta)} = \pi(\theta|y)$$

We are thus performing a *pseudo-marginal* approach: "marginal" because we disregard ξ; *pseudo* because we use $\hat{p}(\cdot)$ not $p(\cdot)$.

Therefore we proved that, using MCMC on an (artificially) augmented posterior, then discard from the output all the random variates ξ created during SMC, returns samples from the **true** posterior. Exact Bayes!

Notice that discarding the ξ is something that we naturally do in Metropolis-Hastings hence nothing strange is happening here. The ξ are just instrumental, uninteresting, variates independent of θ and independent of $\{X_t\}$.

Actually...

When I wrote that we obtain samples from the true posterior, that's true, but you explore the true posterior only as long as the number of iterations $\to \infty$.

When we run a "small enough" number of iterations, not an infinite number, it can still happen that we struggle to explore the whole posterior surface thoroughly.

So the pseudomarginal method is not a silver-bullet. It comes with the usual problems of MCMC, eg tuning issues and difficulties with exploring multimodal surfaces.

Example: SSM with latent SDE

$$dx_t = f(x_t;\theta)dt + g(x_t;\theta)dB_t, \quad dB_t \sim_{iid} N(0, dt)$$
$$y_t = x_t + e_t, \quad e_t \sim_{iid} N(\cdot,\cdot)$$

We consider an Ornstein-Uhlenbeck (OU) process for the latent dynamics:

$$dx_t = -\beta(x_t - \alpha)dt + \sigma \cdot dB_t,$$
$$y_t = x_t + e_t, \quad e_t \sim_{iid} N(0, 0.316^2)$$

where

- $\alpha \in \mathbb{R}$ is the *stationary mean* of the process;
- $\beta > 0$ is the growth rate;
- $\sigma > 0$ diffusion coefficient (intensity of the intrinsic noise).

OU has known (Gaussian) transition densities, however for our purposes it is more useful to write *how* we simulate a path exactly:

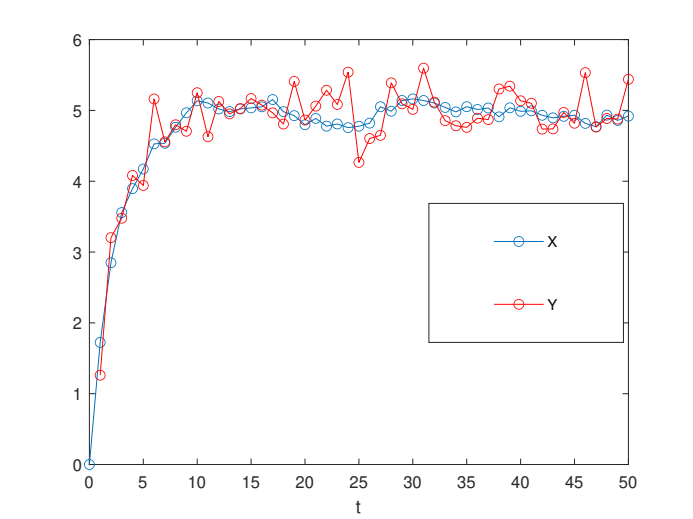
$$x_{t+\Delta} = \alpha + (x_t - \alpha)e^{-\beta\Delta} + \sqrt{\frac{\sigma^2}{2\beta}(1 - \exp(-2\beta\Delta))} \times \xi_{t+\Delta}$$

with $\xi_t \sim N(0,1)$ iid.

Simulation setup

$$dx_t = -\beta(x_t - \alpha)dt + \sigma \cdot dB_t,$$
$$y_t = x_t + e_t, \quad e_t \sim_{iid} N(0, 0.316^2)$$

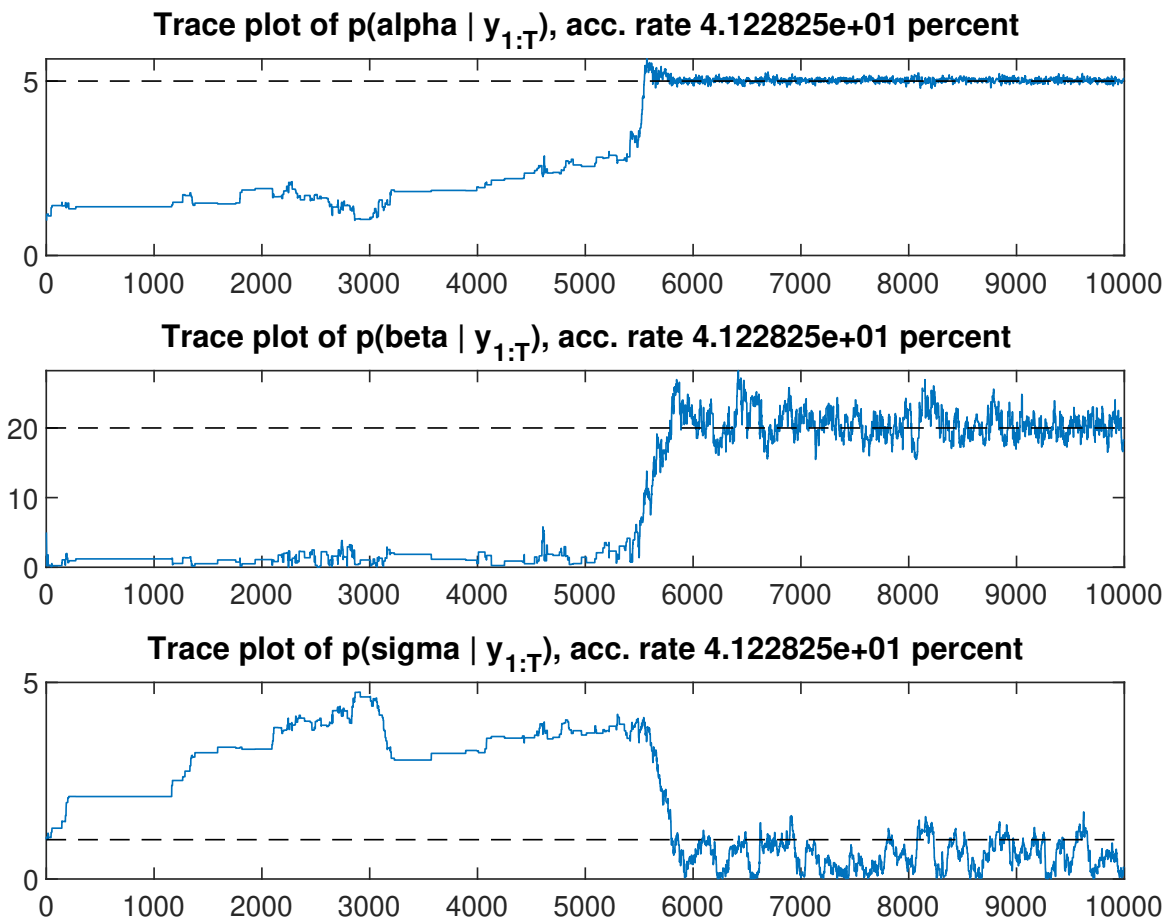
- $T = 50$ observations at equispaced integer times $t = 1, 2, ..., T$, so $\Delta = 1$.
- ground-truth parameters: $\alpha = 5$, $\beta = 20$, $\sigma = 1$.



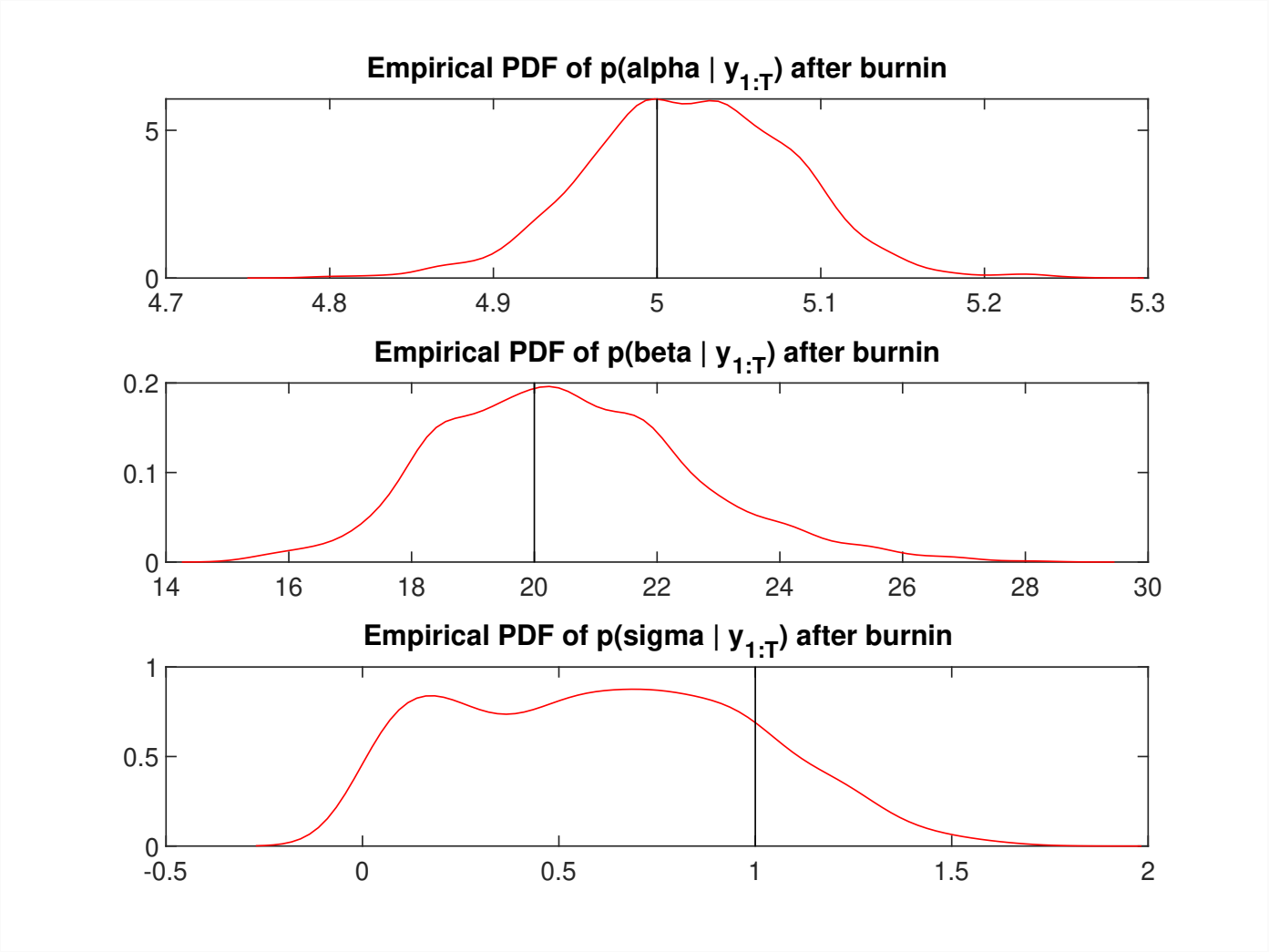
Inference setup

For SMC we use the bootstrap filter with $N = 50$ particles.

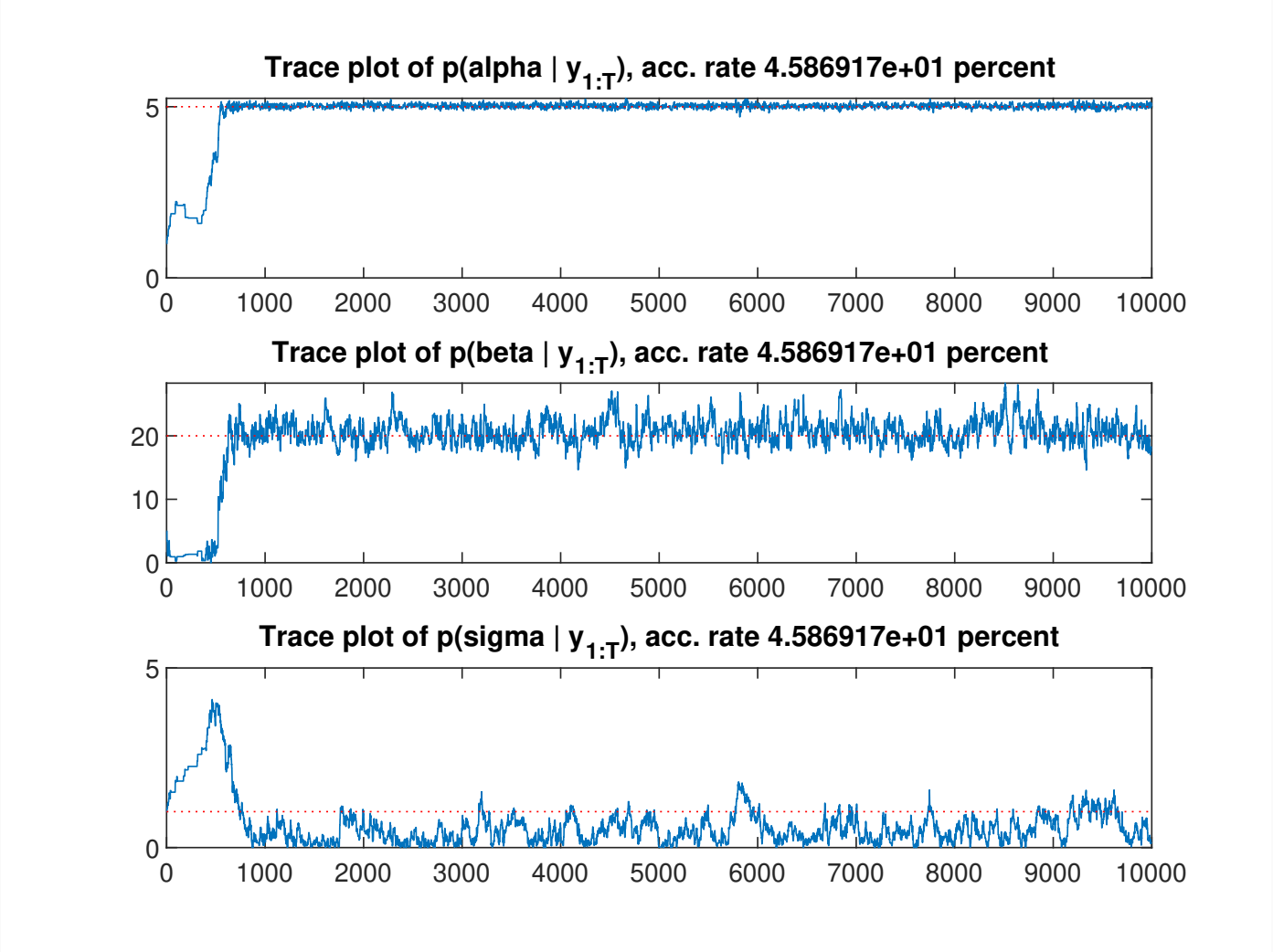
Priors: $\alpha \sim U(1, 10)$, $\beta \sim \text{InvGamma}(3, 50)$, $\sigma \sim \text{InvGamma}(3, 4)$.



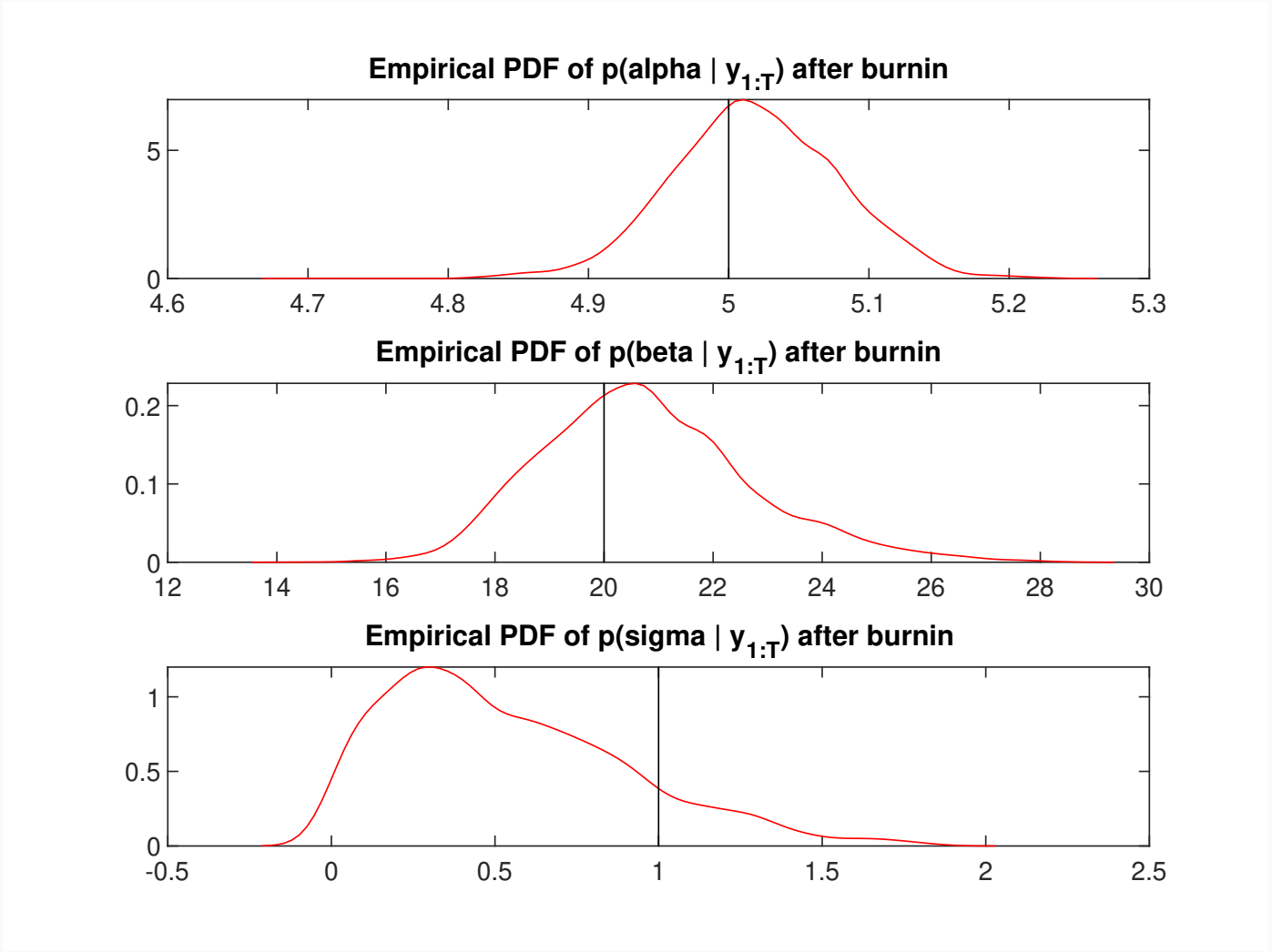
Marginal posteriors.



We obtain the similar inference with $N = 500$ (instead of N=50) but faster convergence:



With N=500:


Empirical PDF of p(alpha | $y_{1:T}$) after burnin
5
0
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
Empirical PDF of p(beta | $y_{1:T}$) after burnin
0.2
0.1
0
12
14
16
18
20
22
24
26
28
30
Empirical PDF of p(sigma | $y_{1:T}$) after burnin
1
0.5
0
-0.5
0
0.5
1
1.5
2
2.5

The effect of selecting N

What's the effect of choosing a small or a large number of particles N?

Think about it: the estimated likelihood used a stochastic procedure, it's not a deterministic approximation.

$\hat{p}(y_{1:T}|\theta)$ from either importance sampling or SMC is a random variable (variability induced by Monte Carlo).

The smaller the N the larger the variance of $\hat{p}(y_{1:T}|\theta)$.

The larger N the more precise the approximation (ie the smaller the variance is).

Here I fix the values of α and β to their true values, and instead consider the equispaced grid for $\sigma \in (0.1, 0.2, ..., 10)$.

I estimate the likelihood via bootstrap filter for each σ value in the grid, and repeat the estimation independently for 50 times.

I used N=10 particles. Below are loglikelihood values for the procedure. Recall the true $\sigma = 1$.

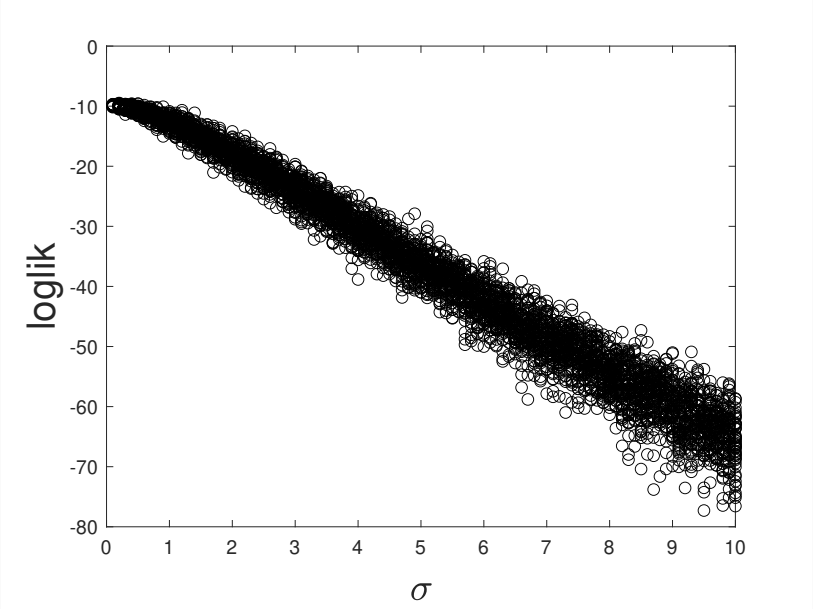


Figure 1: N=10

Same but with N=100.

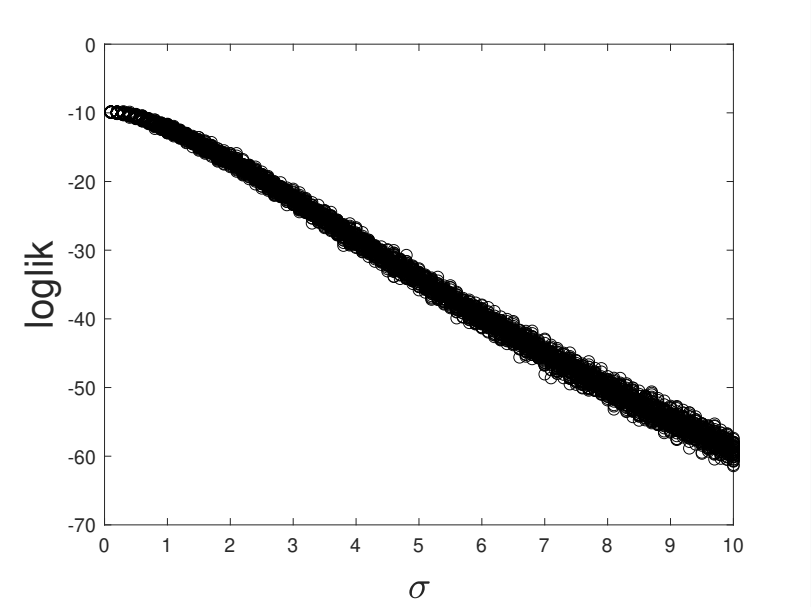
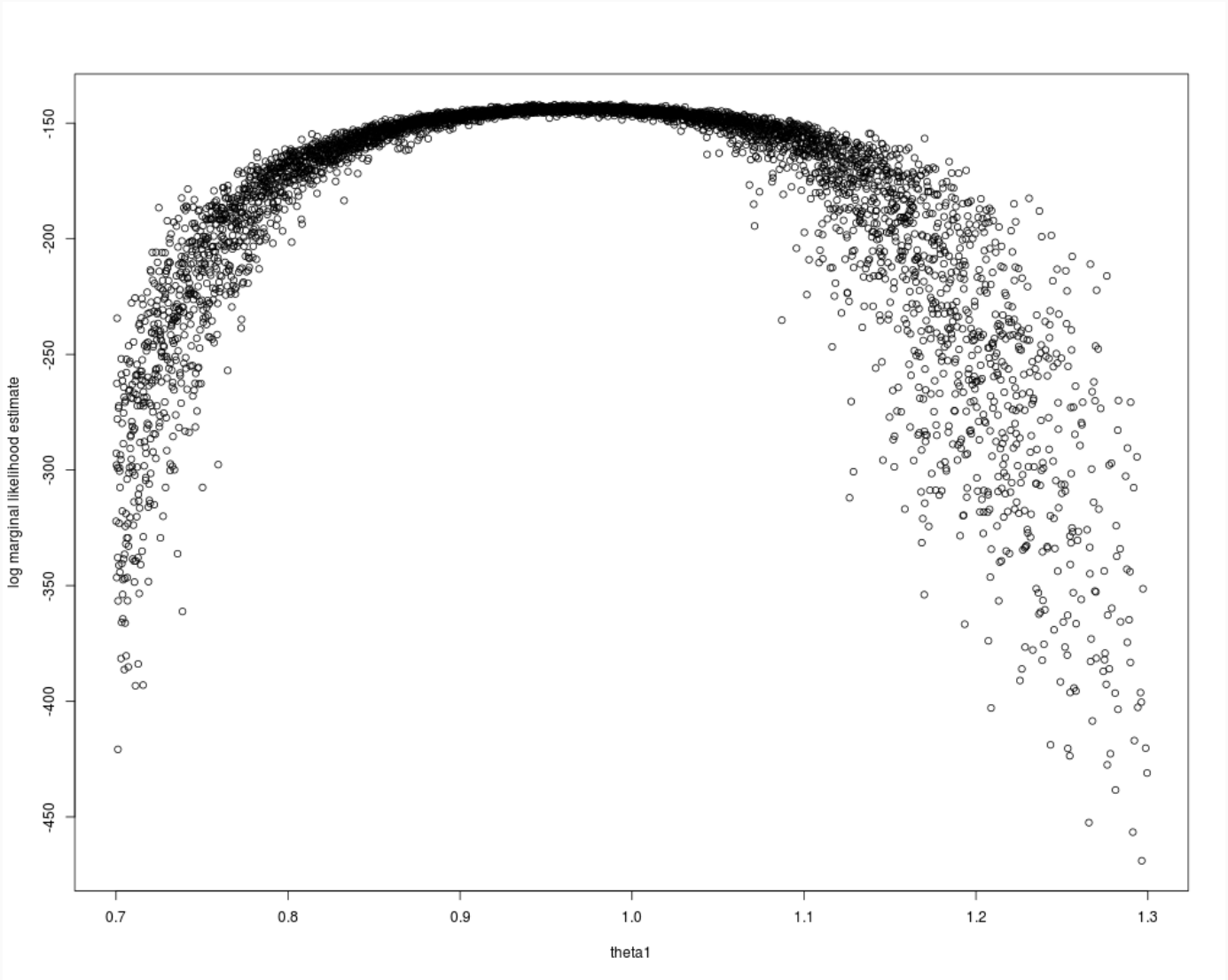


Figure 2: N=100

A more dramatic figure for a different model (see https://tinyurl.com/4h7k3utv)



The variability of the approximation depends also on θ, not only N.

If the current θ value is implausible, for the given data, the likelihood approximation gets noisy because many particles end-up in unimportant regions.

If the likelihood approximation via SMC gets very variable, it can occur that in Metropolis-Hastings (MH) **we occasionally accept an overestimated likelihood** $\rightarrow$ goes in the denominator of the MH ratio $\rightarrow$ difficult to accept further proposals $\rightarrow$ many rejections $\rightarrow$ chain slowly moving.

Easiest solution: increase the number of particles N but this will increase the computational effort.

Plug-and-play strategies

The use of the bootstrap filter into MCMC is sometimes denoted a "plug-and-play" strategy.

This means you can write a generic code that you can reuse for pretty much any state-space model without analytic calculations involved.

This is because you only need to define in your code how the states X advance/propagate from x_t to x_{t+1}.

And then we assume we can evaluate $p(y_t|x_t;\theta)$ pointwisely for any of its arguments.

That's it! You *plug* the model equations in your code, and you *play*.

Other plug-and-play methods

There exist several other plug-and-play methods. These are all examples of *simulation-based inference* methods, in that they are very generic and only require model simulations to get around the **intractability of the likelihood function**.

The R package pomp supports a number of plug-and-play methods:

- particle marginal methods
- iterated filtering
- approximate Bayesian computation (ABC)
- synthetic likelihoods
- ...and more.

Notice pseudomarginal methods, ABC and synthetic likelihoods are not only working with state-space models! They are very general methods.

Conclusions

- We have outlined a powerful methodology for **exact Bayesian inference**. Theoretically exact regardless the number of particles N.
- In practice, a too small N will have a negative impact on chain mixing $\rightarrow$ many rejections, *sticky chain*.
- the methodology is perfectly suited for state-space modelling. However it can deal with more general models.

Downsides when using bootstrap filter

- Recall, in general we wanted to propose particles $x_t^i \sim h(x_t^i | x_{0:t-1}^i, y_{1:t})$, i=1,...,N;
- the above (if implemented) allows particles at time t-1 to be able to "lookahead" to the next datapoint y_t;
- the bootstrap filter has $x_t^i \sim h(x_t | x_{0:t-1}^i, y_{1:t}) \equiv p(x_t | x_{t-1}^i)$. It is "myopic";
- if the dimension $\dim(x_t)$ increases, we might need a very very large N (computationally intensive);
- There are more intelligent SMC filters. Such as the "auxiliary particle filter" (Pitt& Shephard), or the use of "bridges" and "guided proposals" when discretizing an SDE numerically (lots of work by Moritz Schauer, and also Golightly-Wilkinson).

Important issues

How to tune the number of particles?

- Doucet, Pitt, and Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. arXiv:1210.1871 (2012).
- Pitt, dos Santos Silva, Giordani and Kohn. On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. Journal of Econometrics 171, no. 2 (2012): 134-151.
- Sherlock, Thiery, Roberts and Rosenthal. On the efficiency of pseudo-marginal random walk Metropolis algorithms. arXiv:1309.7209 (2013).

More suggestions for further reading in my blog post:

https://tinyurl.com/4964pesp