Bayesian non-Gaussian state space models in R

Random-walk Metropolis versus Hamiltonian Monte Carlo

Jouni Helske

University of Jyväskylä, Department of Mathematics and Statistics

jouni.helske@iki.fi

UNIVERSITY OF JYVÄSKYLÄ

Introduction

We compare the computational efficiency of different Markov chain Monte Carlo (MCMC) algorithms for Bayesian inference of state space models using `rstan`, `bssm`, and `stannis` packages

- **rstan**: Interface for Stan, a probabilistic modelling language with focus on Hamiltonian Monte Carlo (HMC) methods [6]
- **bssm**: State space modelling with MCMC methods based on adaptive random walk Metropolis algorithm (RWM) and particle filters [4]
- **stannis**: Implementation of hybrid IS-HMC algorithm [3]

We compare the following algorithms:

- **HMC**: HMC using NUTS algorithm (Stan) [5]
- **PM**: Pseudo-marginal MCMC (bssm) [1]
- **DA**: Delayed acceptance PM (bssm) [2]
- **IS-RWM**: IS type correction of MCMC with RWM (bssm) [7]
- **IS-HMC**: IS type correction of MCMC with HMC (stannis) [3]

Model

We simulated time series of length $n = 100$ from the following model:

$$\begin{aligned} y_t &\sim \text{Poisson}(\exp(\mu_t)) \\ \mu_{t+1} &= \mu_t + \nu_t + \sigma_\eta \eta_t, \qquad \eta_t \sim N(0,1), \\ \nu_{t+1} &= \nu_t + \sigma_\xi \xi_t, \qquad \xi_t \sim N(0,1), \end{aligned}$$

with $\sigma_\eta = \sigma_\xi = 0.01$, and $\mu_1 = \nu_1 = 0$.

Results

We ran all algorithms $N = 500$ times with 20,000 + 20,000 iterations. As a measure of efficiency, we use inverse relative efficiency (IRE), defined as

$$IRE = 100 \frac{\bar{T}}{N} \sum_{i=1}^{N} (\hat{\theta}_i - \theta)^2,$$

where $\bar{T}$ is the average running time, $\hat{\theta}_i$ is the estimate from ith run, and θ is the parameter estimate based on 1,000,000 iterations of pseudo-marginal MCMC.

	σ_η			μ_1			
method	mean	SE	**IRE**	mean	SE	**IRE**	time (s)
IS-RWM	0.07	0.0016	0.004	0.02	0.008	0.10	17
DA	0.07	0.0016	0.006	0.02	0.008	0.14	24
PM	0.07	0.0016	0.010	0.02	0.008	0.22	39
IS-HMC	0.07	0.0006	0.031	0.02	0.003	0.76	927
HMC	0.07	0.0006	0.049	0.02	0.003	1.02	1166

Conclusions

- We demonstrated the computational differences between different MCMC algorithms
- RWM based algorithms were clearly superior in terms of IRE
- Hybrid IS-HMC improved the efficiency of the standard HMC approach
- Full study is available as a vignette of `stannis`

Inference with bssm

```
model <- ng_bsm(y, P1 = diag(c(10, 0.1)),
  sd_level = halfnormal(0.01, 1),
  sd_slope = halfnormal(0.01, 0.1),
  distribution = "poisson")
run_mcmc(model, n_iter = 10000, nsim_states = 10, method = "isc")
Call:
run_mcmc.ng_bsm(object = model, n_iter = 10000, nsim_states = 10,
  method = "isc")

Iterations = 5001:10000
Thinning interval = 1
Length of the final jump chain = 1160

Acceptance rate after the burn-in period:  0.2318

Summary for theta:

                Mean          SD        SE-IS        SE-AR
sd_level 0.076987129 0.058232429 0.0023210821 0.0032319932
sd_slope 0.007553367 0.005316024 0.0001751029 0.0002196252

Effective sample sizes for theta:

           ESS-IS   ESS-AR
sd_level 454.8747 324.6302
sd_slope 524.3069 585.8809

Summary for alpha_100:

            Mean         SD       SE-IS       SE-AR
level 2.32278892 0.18689398 0.007498485 0.007501719
slope 0.05661408 0.03421972 0.001254904 0.001255445

Effective sample sizes for alpha:

        ESS-IS   ESS-AR
level 537.8590 620.6818
slope 447.9824 742.9461

Run time:
  user  system elapsed
 2.916   0.000   2.929
```

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. URL `http://dx.doi.org/10.1111/j.1467-9868.2009.00736.x`.

[2] J. A Christen and C. Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005. URL `http://dx.doi.org/10.1198/106186005X76983`.

[3] J Helske. stannis: Importance sampling type correction of Markov chain Monte Carlo with Stan, 2017. URL `http://github.com/helske/stannis`. R package version 0.0.1.

[4] J. Helske and Vihola M. bssm: Bayesian inference of exponential family state space models, 2017. URL `http://github.com/helske/bssm`. R package version 0.1.0.

[5] M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:15931623, 2014.

[6] Stan Development Team. RStan: the R interface to Stan, 2016. URL `http://mc-stan.org/`. R package version 2.14.1.

[7] M. Vihola, J. Helske, and J. Franks. Importance sampling type correction of Markov chain Monte Carlo and exact approximations. *ArXiv e-prints*, 2016. URL `https://arxiv.org/abs/1609.02541`.

Acknowledgements

This research was funded by the Academy of Finland research grant 284513.