Numerical methods in stochastic modeling and simulations:

Miniproject 3

Stefan Engblom*

May 2, 2016

In this miniproject you will implement the Metropolis MCMC method to sample the posterior distribution of the parameters of an SDE-model. The idea is to give a feeling for the steps required to form such an MCMC method and how it can be expected to behave. You are to submit the (draft) report on the mandatory part before the next seminar.

If you wish to also take the extended part you are to present your results towards the end of the course (but feel invited to actively help me schedule this presentation on a suitable occasion where the material still feels fresh!).

Note: to pass the course you need to take *one* extended part from one of the three miniprojects during the course.

Mandatory part

Reading For this project I suggest to read and implement at the same time. Refine your understanding and formulations when preparing the written report.

- I have based the present project to some extent on [Press et al., 2007, §15.8].
- From the course web-page I can recommend taking a look at the draft textbook by Zwanzig & Mahjani.

Bayesian setup We are given a set of data D and wishes to estimate the parameters x used to generate D. The Bayesian understanding is to

*Division of Scientific Computing, Department of Information Technology, Uppsala university, SE-751 05 Uppsala, Sweden. stefane@it.uu.se

investigate the *posterior* distribution $P(x|D)$, for example, by producing independent samples $(x_i) \sim P(x|D)$.

We shall suppose that the likelihood $P(D|x)$ is available, either in closed form, or via some kind of numerical approximation. *Try to explain to yourself why $P(D|x)$ is easier to handle than $P(x|D)$!*

According to Baye's theorem, $P(x|D) \propto P(D|x)P(x)$, where $P(x)$ is the prior distribution on the parameters. The constant of proportionality is generally unknown.

Ignoring the prior $P(x)$ for now, at Boltzmann temperature 1 and defining a Hamiltonian $H(x) = -\log P(D|x)$, the Metropolis algorithm will produce samples $(x_i) \sim \exp(-H(x)) = P(D|x) \propto P(x|D)$.

Implementation

- Firstly, you should implement a log-likelihood $L(x) = \log P(D|x)$ for your model of the data.
- Secondly, decide on a suitable set of proposal moves from the current state x to some new state y.
- Thirdly, given a proposed transition $x \to y$ (and $L(x) \to L(y)$), formulate the Metropolis-Hastings acceptance criterion.

Some pieces of advice.

- Test your log-likelihood and your data by finding the maximum likelihood estimator $\hat{x} = \arg\max_x L(x)$. This is usually very fast and should give sane answers. If you increase data D, does the ML estimator gets closer to the true parameters?
- The proposal moves can be pretty much anything you want as long as a series of steps in principle can reach any region of relevance in parameter space.
- If your factored proposal distribution is $P(x \to y) = g(x \to y)A(x \to y)$, then the Metropolis-Hastings acceptance probability is

$$A(x \to y) = \min\left(1, \exp(\Delta L(x \to y))\frac{g(y \to x)}{g(x \to y)}\right),$$

  where $\Delta L(x \to y) = L(y) - L(x)$. For symmetric proposals which depend only on the absolute difference $|y - x|$, we have $g(y \to x) = g(x \to y)$. For example, such proposals are generated in a Brownian walk in parameter space. Another useful proposal is a log-normal proposal, say $y = qx$, where $q \sim \exp(\mathcal{N}(0, \sigma^2))$. For this proposal $g(y \to x)/g(x \to y) = q$. Log-normal proposals have the benefit that they will not change the sign of the states.

Start your Markov chain from some suitable initial value x_0 and then walk. Try to select scaling parameters such that about 10–40% of the proposals are accepted. Inspect data, parameter samples, and also monitor the autocorrelation. Increase the data and run again: compare the results.

Task 1: identify a GBM Create a couple of trajectories from the geometric Brownian motion model, $dX_t = \mu X_t\, dt + \sigma X_t dW_t$, with $X(0) = 1$. Note that you can produce *exact* samples from this model. With (μ, σ) given, try to reason about a suitable sampling rate Δt and a suitable sampling time interval $[0, T]$. With Δt too large and/or T too small, it might be very difficult to identify the model.

Present the ingredients of your MCMC implementation and how you arrived at them.

Test your implementation by exploring the posterior density over (μ, σ). Try to produce solid estimators with error bounds. -How does the estimators respond to eg. a 4-fold increase of data?

Task 2: the GBM challenge On the course web-page you will find data from a GBM model. -What is the best you can say about the parameters for this model?

Task 3: identify your favorite SDE Instead of the basic GBM, pick your own favorite SDE-model and try to identify the parameters given observations. For a one-dimensional SDE, a closed-form solution of the likelihood $L(x)$ is not necessary, a numerical solution should work just as fine. For two dimensions and higher, numerical likelihoods are going to be more challenging (see Task 7).

Extended part

For the extended part, perform and report on *at least one* of the tasks below.

The tasks are very openly described and the intent here is that you undertake a bit of research and take the role of a critical reviewer. *Take your job seriously!*

Task 4: adaptive MCMC There are many different ideas for how to form an *adaptive* MCMC, which typically adapts the proposal in some clever way as the Markov walk progresses. For several examples, see the draft book by Zwanzig & Mahjani (there are others). Implement an adaptive MCMC algorithm and test on an example of interest. -How much efficiency is gained?

Task 5: hidden observations The examples above assumed that you directly observe the process over which you are performing inference. Depending on the application, other indirect setups are more suitable. *Investigate such a situation and try it out!* There might be reasons the MCMC-approach does not work at all and other ideas must be pursued. *Explain this situation!* Or maybe some parts of the MCMC-methodology is in fact re-used (perhaps in disguise), while others must be abandoned?

This track should preferably start with an application relevant to yourself.

Task 6: identify a non-trivial stochastic model The above discussion circled mainly around Wiener SDEs. -What about models with jumps? Try to identify a model driven by Poissonian processes! Or a Lévy-flight!

Task 7: numerical likelihoods Try a setup with a numerical likelihood. -Do some research: what is the impact of the numerical errors in the likelihood? How does the approach scale to higher dimensional state variables?

References

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes*. Cambridge University Pres, Cambridge, 3rd edition, 2007.