Stat314 Term 4: Summary

Patrick Graham

October, 2021

The problem

We want to compute

$$p(\boldsymbol{\theta}|\mathrm{data}) = \frac{p(\mathrm{data}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int (p(\mathrm{data}|\boldsymbol{\theta})p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}}. \tag{1}$$

We know the form of $p(\mathrm{data}|\boldsymbol{\theta})$ - usually something like $\prod_i p(Y_i|\boldsymbol{\theta})$ - and $p(\boldsymbol{\theta})$ and hence can compute the numerator in (1), so we know:

$$p(\boldsymbol{\theta}|\mathrm{data}) \propto p(\mathrm{data}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2}$$

The rhs of (2) is sometimes referred to as the "unnormalized posterior density," $q(\boldsymbol{\theta}|\mathrm{data})$. But to actually compute the posterior we need be able to integrate rhs of (2)- e.g (i) just to get the proportionality constant; (ii) to compute moments; (iii)to compute interval and tail probabilities, etc For non-conjugate problems such integrations can be difficult, and, in realistic problems, they are high-dimensional.

Our approach

We have studied several Monte Carlo algorithms for approximating posterior distributions, without having to explicitly evaluate the normalising constant:

- Rejection sampling - sample from an approximation $g(\boldsymbol{\theta})$ and correct by accepting draws with probability $q(\boldsymbol{\theta})/Mg(\boldsymbol{\theta})$, where $M = \max\left(q(\boldsymbol{\theta})g(\boldsymbol{\theta})\right).$
- Importance sampling - sample from an approximation and "weight up" to the posterior by applying the importance weights $w_i = q(\theta_i)/g(\theta_i)$ to the sampled values.
- Metropolis-Hastings sampler - jump through the parameter space, guided to by the (unnormalised) posterior. Always move to a point with higher posterior density; move to a point of lower posterior density with probability $r_{MH}()$.
- Gibbs sampler - alternately sample from the full conditional posterior distributions (missing data can be treated like a parameter)

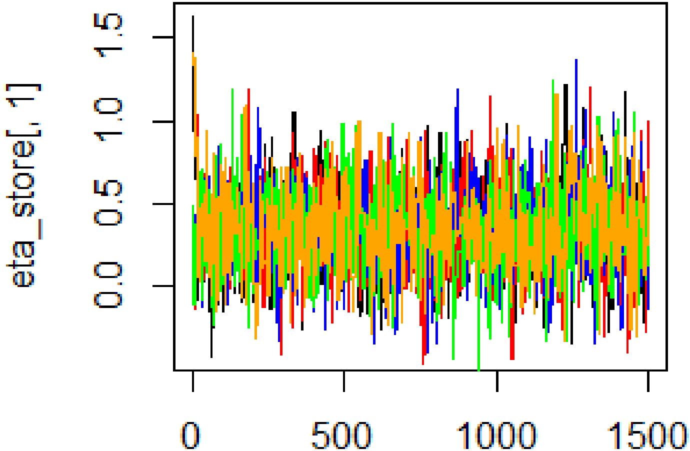
Monte Carlo (MC) methods are approximations

- If we can sample N draws directly from the posterior, for some parameter θ for which the posterior standard deviation is σ the Monte Carlo error for approximation of the posterior mean is $\mathrm{MCerror} = \sigma/\sqrt{N}$. This is a best case and is often pretty small.
- A useful concept in evaluating the efficiency of MC methods, relative to sampling directly from the posterior is the effective Monte Carlo sample size, Neff. Neff is the number of direct, independent draws from the posterior that would give the same Monte Carlo error as achieved by a given algorithm
- For rejection sampling Neff is the number of accepted draws.
- For importance sampling $\mathrm{Neff} \approx 1/\sum_i \tilde{w}_i^2$, which is smaller the more variable are the weights ($\tilde{w}_i$. denotes the *normalised* weights.)
- For MCMC, computation of Neff is complicated due to the correlated nature of the draws. We have seen that it is often considerably smaller than the nominal MC sample size (i.e actual number of draws in the posterior sample).

MCMC

- Main workhorse of modern Bayesian computation. Better suited to higher dimensional problems than acceptance or rejection sampling.
- Initially does not sample from the posterior, but converges to the posterior after some number of iterations.
- Checking convergence is therefore an important part of MCMC practice.
  - multiple chains
  - traceplots
  - $\hat{R}$ (Gelman-Rubin diagnostic)
- Many of our MCMC examples have converged very quickly. This will not always be the case.
- Choice of jumping (proposal) distribution can affect convergence - moving too slowly through the space will lead to high correlations, slow convergence and low Neff. Remember: `MH_example1_binomlogit_2021.r`

Metropolis-Hastings with optimal jumps



Metropolis-Hastings with poor choice of jumping distribution