QUEEN MARY, UNIVERSITY OF LONDON

MTH6102: Bayesian Statistical Methods

Practical 10

2023-2024

1 Symmetric Metropolis-Hastings algorithm

1.1 Binomial data/beta prior

The R code file "practical 10 R code beta-binomial.R" implements the symmetric Metropolis-Hastings algorithm for the binomial example with beta prior distribution that we saw in the lecture 9B

- Data: $k = 12 \sim \text{binomial}(40, q)$, where q is the probability of success.
- Prior: $p(q) \sim \text{beta}(2, 2)$.

Our goal is to construct a sample $q_1, q_2 \ldots$ from the posterior density, which is $\text{beta}(14, 30)$.

For each part of the code, make sure you understand what each line is doing. You can change the code, and output intermediate quantities to help understand how it works. You can output quantities inside a loop using `print(...)`:

```
for(i in 1:5){
    x = i^2
    print(x)
}
```

The variable `M` in the code file is M, the length of the sequence of random variables $q_1, \ldots, q_M$ that the algorithm generates. First run the code with some small M so that you can look at the individual steps to understand how the algorithm behaves. Plot the vector of q values against the iteration number to visualize the behaviour, as in the lecture code. (The iteration numbers are the sequence $1, 2, \ldots, M$, which in R is given by `1:M`.)

Then run the code to see that when M is large enough, the sample generated is approximately from the posterior density that we are aiming for, which is a beta distribution.

Plot the histogram of the sample and the exact beta posterior density on the same graph. Check that summaries of the posterior distribution are similar to what you would expect for the exact beta posterior.

2 Working on the log scale

2.1 Binomial data/beta prior

In order to get accurate results with large datasets, we would do all the calculations in the Metropolis algorithm using log pdfs or log probabilities. The R file "practical 10 R code beta-binomial log scale.R" implements the Binomial data/beta prior on the log scale. In this case, the probability of acceptance changes to

$$\delta = \min\{0, \mathcal{L}(\psi) - \mathcal{L}(\theta_i)\},$$

where for any θ,

$$\mathcal{L}(\theta) = \log p(\theta) + \log p(y \mid \theta),$$

and $p(\theta)$ is the prior and $p(y \mid \theta)$ is the likelihood. Then, the algorithm accepts the newly proposed random variable ψ if $\log u < \delta$, where $u \sim U[0, 1]$.

The option `log=TRUE` can be included in any of the probability density or probability mass function commands, such as `dbinom`, to return the log pdf or log probability, $\log p(y \mid \theta)$.
`?dbinom`