STAT40380/STAT40390/STAT40850 Bayesian Analysis

Dr Niamh Russell

School of Mathematics and Statistics
University College Dublin

niamh.russell@ucd.ie

March 2016

The Gibbs sampler and MCMC

- The *Gibbs sampler* algorithm is a neat way to simulate values from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{x})$ when we observe only $q(\boldsymbol{\theta}|\mathbf{x})$
- It works by updating parameter k (of K total parameters) at iteration t from the distribution $q(\theta_k|\boldsymbol{\theta}^{t-1}_{-k}, \mathbf{y})$ for each parameter in turn
- When $q(\theta_k|\boldsymbol{\theta}^{t-1}_{-k}, \mathbf{y})$ is a standard probability distribution we can sample directly
- When $q(\theta_k|\boldsymbol{\theta}^{t-1}_{-k}, \mathbf{y})$ is not a standard probability distribution we can use another method (such as rejection sampling)
- When sampling from $q(\theta_k|\boldsymbol{\theta}^{t-1}_{-k}, \mathbf{y})$ is very hard, the Gibbs sampler may not be an appropriate tool
- Instead we can use a generalisation of the Gibbs sampler known as the *Metropolis-Hastings algorithm*

The Metropolis alogrithm

- uses a random walk (ie Markov) acceptance/rejection rule to converge to the posterior distribution
- Below are the steps for the univariate version (ie only one parameter) but the method easily extends to multiple parameter problems
- Steps:
  1. Choose starting values θ^0 for which $q(\theta^0|\mathbf{x}) > 0$.
  2. For iterations $t = 1, 2, 3 \ldots$
     (a) Sample a *proposal value* θ^* from a *proposal* or *jumping distribution* at time t: $J_t(\theta^*|\theta^{t-1})$. For the Metropolis algorithm, this distribution must be *symmetric*, so that $J_t(\theta^*|\theta^{t-1}) = J_t(\theta^{t-1}|\theta^*)$
     (b) Calculate
     $$r = \frac{q(\theta^*|\mathbf{x})}{q(\theta^{t-1}|\mathbf{x})}$$
     (c) Set
     $$\theta^t = \begin{cases} \theta^* & \text{with probability } min(1, r) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

Notes about the Metropolis algorithm

- The algorithm requires the ability to calculate the ratio *r* for all possible values of θ
- Similarly, we must be able to draw a θ^* for all possible values of θ
- If the jump is not accepted, so that $\theta^t = \theta^{t-1}$, this counts as an iteration so we move on with the algorithm
- A simple version of the Metropolis algorithm is:
  1. If the jump increases the posterior density, set $\theta^t = \theta^*$
  2. If the jump decreases the posterior density, set $\theta^t = \theta^*$ with probability *r*
- Thus the Metropolis algorithm is very similar to other optimisation procedures, with an extra step to occasionally accept values of lower probability

Example

Suppose that $x_i \sim Po(e^{\lambda})$ for $i = 1, \ldots, n$ with prior $\lambda \sim N(0, 2)$. Some data are observed such that $n = 10$ and $\sum x_i = 22$. Write out the steps to produce posterior samples of λ using the Metropolis algorithm .

Why does the Metropolis algorithm work?

- Two parts to proof:
  - First, that there is a unique stationary distribution for the Markov chain
  - Second, that the atationary distribution is the posterior distribution
- Remember: a Markov chain has a unique stationary distribution if it is irreducible, aperiodic and not transient
  - Irreducible: it is possible to get from any θ to any other θ^*
  - Aperiodic: it may return to θ at any time (irregularly)
  - Not transient: will not get stuck at a particular value of θ

Target distributions

- Consider starting the algorithm at time $t-1$ with a draw θ^{t-1} from the target distribution $p(\theta|\mathbf{x})$
- Now consider two such points θ_a and θ_b drawn from p so that $p(\theta_b|\mathbf{x}) \geq p(\theta_a|\mathbf{x})$. The transition probability from θ_a to θ_b is:

$$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b) = p(\theta_a|\mathbf{x}) J_t(\theta_b|\theta_a)$$

  where the acceptance probability is 1 because $p(\theta_b|\mathbf{x}) \geq p(\theta_a|\mathbf{x})$
- Conversely, the probability of transition from θ_b to θ_a is:

$$\begin{aligned} p(\theta^t = \theta_a, \theta^{t-1} = \theta_b) &= p(\theta_b|\mathbf{x}) J_t(\theta_a|\theta_b) \frac{p(\theta_a|\mathbf{x})}{p(\theta_b|\mathbf{x})} \\ &= p(\theta_a|\mathbf{x}) J_t(\theta_a|\theta_b) \end{aligned}$$

- Since these probabilities are the same, and that they are both drawn from p, *p must be the stationary distribution of the Markov chain*

The Metropolis-Hastings alogrithm

- A generalisation of the Metropolis algorithm which allows for non-symmetric jumping distributions
- Steps:
  1. Choose starting values θ^0 for which $q(\theta^0|\mathbf{x}) > 0$.
  2. For iterations $t = 1, 2, 3 \ldots$
     (a) Sample a *proposal value* θ^* from a *proposal distribution* at time t: $J_t(\theta^*|\theta^{t-1})$.
     (b) Calculate
     $$r = \frac{q(\theta^*|\mathbf{x})/J_t(\theta^*|\theta^{t-1})}{q(\theta^{t-1}|\mathbf{x})/J_t(\theta^{t-1}|\theta^*)}$$
     (c) Set
     $$\theta^t = \begin{cases} \theta^* & \text{with probability } min(1, r) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

Notes on the Metropolis-Hastings algorithm

- Relaxing the jumping rule can be convenient in certain situations (eg when a parameter is bounded)
- The M-H algorithm can also be useful in speeding up the random walk to produce samples from the posterior distribution
- The proof of the M-H algorithm is identical to that of the Metropolis algorithm

The M-H algorithm

Example

Suppose that $x_i \sim Po(\gamma)$ for $i = 1, \ldots, n$ with prior $\log \gamma \sim N(0, 2)$. Some data are observed such that $n = 10$ and $\sum x_i = 22$. Write out the steps to produce posterior samples of γ using the Metropolis-Hastings algorithm