

## Lecture Note 12: Bayesian Inference using MCMC: Metropolis Algorithm and Examples

Geweke, Gowrisankaran, and Town (2003)

$i = 1, \dots, n$ : patients with pneumonia in LA County

$j = 1, \dots, J$ : hospitals in LA County

$x_i$ : a  $k \times 1$  vector of patient characteristics

$z_{ij}$ : a  $q \times 1$  vector of patient-hospital characteristics, including distance of patient  $i$  to hospital  $j$ .

$m_i$ : mortality indicator, =1 if patient dies.

$c_i$ :  $J \times 1$  vector of indicators for whether patient  $i$  was admitted to a hospital.

Model:

$$m_i^* = c_i' \beta + x_i' \gamma + \epsilon_i, \\ \epsilon_i \stackrel{\text{iid}}{\sim} N(0, 1).$$

Here  $\beta = (\beta_1, \beta_2, \dots, \beta_J)'$ .

Interpretation: if patient  $i$  were randomly assigned to hospital  $j$ , then

$$Pr(m_i = 1) = \Phi(\beta_j + x_i' \gamma).$$

However, we suspect that hospital choice is not random, so that  $c_i$  is correlated with  $\epsilon_i$ .

Let

$$Z_i = \begin{pmatrix} z_{i1}' \\ \vdots \\ z_{iJ}' \end{pmatrix}, \\ c_i^* = Z_i \alpha + \eta_i.$$

The hospital indicators are formed from the latent  $J \times 1$  vector  $c_i^*$  by

$$c_i = \begin{pmatrix} c_{i1} \\ \vdots \\ c_{iJ} \end{pmatrix}, \quad \text{with } c_{ij} = 1(c_{ij}^* \geq c_{ik}^* \forall k).$$

Normalize  $c_{iJ}^* = 0$ , so there are  $J - 1$  latent variables. Assume

$$\begin{pmatrix} \epsilon_i \\ \eta_{i1} \\ \vdots \\ \eta_{i,J-1} \end{pmatrix} \sim N\left(0, \begin{bmatrix} 1 & \pi' \\ \pi & \Sigma \end{bmatrix}\right).$$

This allows correlation between the elements of  $\eta$  and  $\epsilon$ , which makes  $\eta$  and  $c_i$  correlated.

Exclusion restriction: variables like distance to hospital are assumed to affect hospital choice, but not have a direct effect or mortality given hospital choice. Thus there is “exogenous variation” in hospital choice.

Here  $J$  is large:  $J = 114$ . This means that  $\Sigma$  has 6441 free parameters. In order to obtain a tractable model, the authors make some simplifying assumptions on the form of  $\Sigma$ .

### Metropolis Algorithm

In the previous applications of data augmentation and Gibbs sampling, it was possible to draw from each of the full conditional distributions because they fell into well-known families of distributions (e.g. normal, Wishart, truncated normal).

In some cases, this is not possible. A simple example is the logit model.

### Example: Logit Model

$$Pr(y_i = 1|x_i, \beta) = \Lambda(x_i'\beta),$$

where

$$\Lambda(z) = \frac{\exp(z)}{1 + \exp(z)}.$$

The conditional likelihood function is

$$p(y_1, \dots, y_n|x_1, \dots, x_n, \beta) = \prod_{i=1}^n \Lambda(x_i'\beta)^{y_i} (1 - \Lambda(x_i'\beta))^{1-y_i}.$$

So for a prior  $p(\beta)$ :

$$p(\beta|y, x) \propto p(\beta) \prod_{i=1}^n \Lambda(x_i'\beta)^{y_i} (1 - \Lambda(x_i'\beta))^{1-y_i}.$$

If  $x_i$  is anything other than a constant, there is no family of priors such that the posterior has a known form.

One way to calculate the posterior is to use the discretization method you explored in HW5. However, if  $x_i$  is high-dimensional, this becomes computationally infeasible. There are other numerical methods for evaluating integrals (which is what you need to calculate the normalizing constant in

the posterior expression above). These methods typically work well for dimensions up to about 4. One alternative that can be used in many cases is the Metropolis algorithm.

### Metropolis-Hastings Algorithm

Suppose we wish to draw  $w$  from a density  $\pi(w)$ . Define a “candidate generating density”  $q(w, w')$ , which will be used to generate a potential draw for  $w'$  given  $w$ .

1. Initialize  $w^0$  at some value.
2. Generate a draw  $w' \sim q(w^0, w')$ . Calculate

$$\alpha = \begin{cases} \min \left\{ \frac{\pi(w')q(w', w^0)}{\pi(w^0)q(w^0, w')}, 1 \right\} & \text{if } \pi(w^0)q(w^0, w') > 0 \\ 1 & \text{if } \pi(w^0)q(w^0, w') = 0 \end{cases}$$

Accept the candidate draw  $w'$  with probability  $\alpha$ . If accept, we set  $w^1 = w'$ . If we reject, we set  $w^1 = w^0$ .

3. Generate a draw  $w' \sim q(w^1, w')$ . Calculate

$$\alpha = \begin{cases} \min \left\{ \frac{\pi(w')q(w', w^1)}{\pi(w^1)q(w^1, w')}, 1 \right\} & \text{if } \pi(w^1)q(w^1, w') > 0 \\ 1 & \text{if } \pi(w^1)q(w^1, w') = 0 \end{cases}$$

Accept the candidate draw  $w'$  with probability  $\alpha$ . If accept, we set  $w^2 = w'$ . If we reject, we set  $w^2 = w^1$ .

- 4.
5. ...

At first glance, it appears that we need to be able to calculate  $\pi(w)$  to be able to implement this procedure. Note however, that we really only need to be able to calculate  $\pi(w)$  *up to a normalizing constant*. This is handy because when working with posterior distributions, we can avoid calculating the normalizing constant which is the main source of difficulty.

What candidate generating density to use? A convenient one is to draw

$$w' = w + \epsilon,$$

where  $\epsilon$  has a symmetric density, e.g.  $N(0, \sigma^2)$ . This makes  $w' \sim N(w, \sigma^2)$ , or more concretely,

$$q(w', w) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(w - w')^2\right).$$

Notice that  $q(w', w) = q(w, w')$  by the symmetry of the normal distribution. Therefore, the factor  $\frac{q(w', w^j)}{q(w^j, w')}$  drops out of the Metropolis probability calculation, so that we only need to calculate  $\frac{\pi(w')}{\pi(w^j)}$  to get  $\alpha$ .

This special case of the Metropolis-Hastings algorithm is sometimes called the Metropolis algorithm or “random-walk Metropolis-Hastings.”

The Metropolis-Hastings algorithm is a Markov chain and converges to the target distribution  $\pi(w)$ . Notice that it some of the time it will stay at the same value for  $w$  for two or more steps.

A key issue is what candidate generating density  $q(\cdot, \cdot)$  to use. When using the random-walk Metropolis algorithm, this amounts to choosing  $\sigma^2$ .

If  $\sigma$  is too high, then candidate draws for  $w$  will be very far away from the current draw, and in a region of the parameter space that has low probability under  $\pi$ . This would lead to a high rejection rate, and a Markov chain that doesn't move very often.

If  $\sigma$  is too low, then candidate draws for  $w$  will be very close to the current draw. Even if the algorithm accepts frequently, the chain will move slowly around the parameter space.

So we need to strike a balance. Often, it's a good idea to experiment with different values for  $\sigma$ .

As with Gibbs samplers, it is a good idea to run multiple chains from disperse starting values, to help assess convergence.

### Logit Example: Metropolis Algorithm

Back to logit. This can be handled with a simple Metropolis algorithm:

1. Initialize the vector  $\beta^0$ .
2. Generate a candidate draw  $\tilde{\beta} = \beta^0 + \sigma \cdot \epsilon$ , where  $\epsilon \sim N(0, 1)$ .

Then calculate the ratio  $\alpha$ :

$$\alpha = \min \left\{ \frac{g(\tilde{\beta})}{g(\beta^0)}, 1 \right\},$$

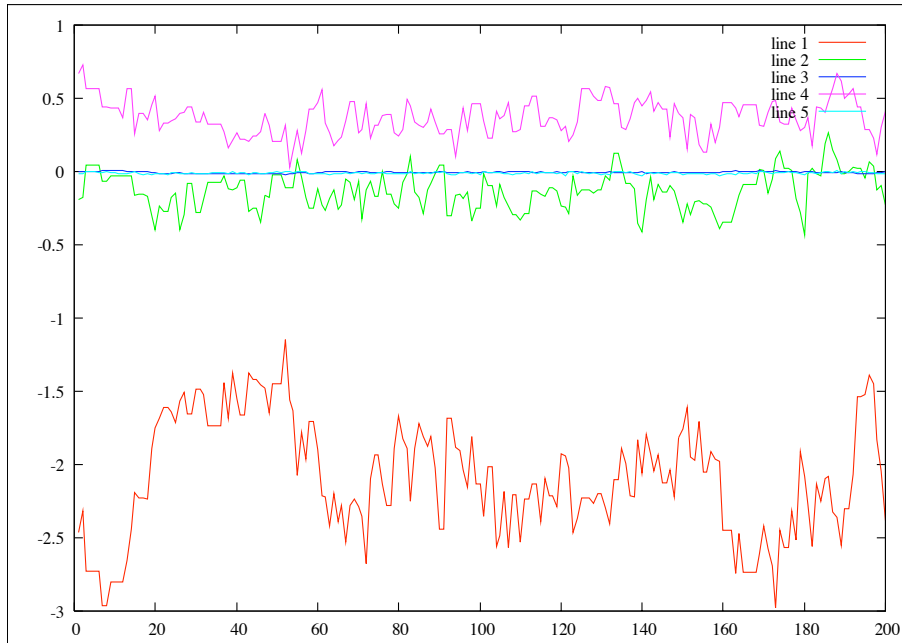
where

$$g(\beta) = p(\beta) \prod_{i=1}^n \Lambda(x'_i \beta)^{y_i} (1 - \Lambda(x'_i \beta))^{1-y_i}.$$

Draw  $U \sim Unif[0, 1]$ . If  $U \leq \alpha$ , form  $\beta^1 = \tilde{\beta}$ . If  $U > \alpha$ , form  $\beta^1 = \beta^0$ .

3. Generate a candidate draw  $\tilde{\beta} = \beta^1 + \sigma \cdot \epsilon \dots$

Here is a graph showing the Metropolis draws using this algorithm for some data from Hirano, Imbens, Rubin, and Zhou (2000):



### Metropolis-within-Gibbs

Suppose we have a  $k \times 1$  parameter  $\theta$ , with prior  $p(\theta)$ , and the likelihood is  $p(z|\theta)$ .

One way to simulate the posterior distribution  $p(\theta|z)$  is the Metropolis algorithm.

An alternative is to split the parameter  $\theta$  into subvectors

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_M \end{pmatrix}.$$

We can then form a Gibbs-like algorithm:

1. Initialize  $\theta^0 = (\theta_1^0, \theta_2^0, \dots, \theta_M^0)$ .
2. Draw  $\theta_1^1$  given  $\theta_2^0, \dots, \theta_M^0$  and  $z$ .
3. Draw Draw  $\theta_2^1$  given  $\theta_1^1, \theta_3^0, \dots, \theta_M^0$  and  $z$ .
4. etc.

For each of the draws of  $\theta_m^j$ , we can either use the full conditional distribution as in regular Gibbs sampling, OR use a Metropolis draw.

This is particularly useful if some of the full conditionals have a known form, but some of them do not.