Lecture 10 - Key

Bayesian GLMs and Metropolis-Hastings Algorithm

We have seen that with conjugate or semi-conjugate prior distributions the Gibbs sampler can be used to sample from the posterior distribution. In situations, such as Generalized Linear Models (GLMs), where conjugate prior distributions are not available the Gibbs sampler cannot be used. Instead, the Metropolis-Hastings algorithm will be used to sample from the posterior distribution.

Poisson Regression Model

Example. (Hoff p. 171) A sample was taken of 52 female sparrows to study their reproductive habits. The number of offspring and the age of the sparrow are recorded. The response (number of offspring) is a non-negative integer. A logical distribution for this data is the Poisson distribution. The conditional expection from a Poisson regression model, as a function of age, can be formulated as:

$$\log E[Y|x] = \log(\theta_x) = \beta_0 + \beta_1 x + \beta_2 x^2.$$

This implies

$$E[Y|x] = \theta_x = \exp(\beta_0 + \beta_1 x + \beta_2 x^2),$$

where θ_x is the mean term in the Poisson model.

Q: This forces the expectation to be positive, but not an integer. Is this a problem?

Formallly, the Poisson regression model can be expressed as:

$$Y|x \quad \sim \quad Poisson(\theta_x) = Poisson(\exp(\beta_0 + \beta_1 x + \beta_2 x^2))$$

In a generalized linear model there are three defining components:

1. A probability distribution from an exponential family, in this case the Poisson distribution,
2. a linear predictor $(\beta_0 + \beta_1 x + \beta_2 x^2)$ is linear in the predictors, and
3. a link function, which is the log link in the Poisson case.

Note it is important to consider the restriction of the Poisson distribution, namely that the mean and variance are the same. **Q:** If this is not appropriate, what do we do?

One option is to consider the negative binomial distribution. The natural parameterization of the negative binomial distribution has one parameter corresponding to the probability of a sucess and the other as the number of successes until stopping. However, it can be parameterized such that one parameter represents the mean of the distribution and the other as the variance of the distribution.

Up until now, we haven't talked about anything inherently Bayesian. So what do we need to think about this model from a Bayesian persepective?

We need priors on β_0, β_1, and β_2. So how do we come up with a prior on $\tilde{\beta}$? In other settings we have used a multivariate normal distribution, say $p(\tilde{\beta}) = MVN(\tilde{\beta}_0, \Sigma_0)$. Then the posterior distribution is defined as:

$$\begin{aligned} p(\tilde{\beta}|\tilde{y}, X) &\propto p(\tilde{y}|\tilde{\beta}, X, \tilde{y}) \times p(\tilde{\beta}) \\ &\propto \prod_{i=1}^{n} \exp(\tilde{\beta}^T \tilde{x}_i)^y \exp\left(-\exp(\tilde{\beta}^T \tilde{x}_i)\right)/y_i! \times \\ & \quad (2\pi)^{-p/2} |\Sigma_0|^{-1} \exp\left(-\frac{1}{2}(\tilde{\beta} - \tilde{\beta}_0)^T \Sigma_0^{-1} (\tilde{\beta} - \tilde{\beta}_0)\right) \end{aligned}$$

hmm... we don't see a kernel of a distribution and the integration looks nasty, so what now?

Metropolis-Hastings, coming soon...

Logistic Regression Model

Another common GLM is logistic regression, which is used for modeling binary outcomes. The logistic regression model uses the logistic link function where:

$$\begin{aligned} y_i|\tilde{x}_i, \tilde{\beta} &\sim Bernoulli(\theta_i) \\ \tilde{\beta}^T \tilde{x}_i &= \log \frac{\theta_x}{1-\theta_x} \\ \text{which implies } \theta_i &= \frac{\exp(\tilde{\beta}^T \tilde{x}_i)}{1+\exp(\tilde{\beta}^T \tilde{x}_i)}. \end{aligned}$$

The logistic link function restricts θ_x to the interval $(0, 1)$.

Let's consider the structure of the posterior distribution as a function of the prior for $\tilde{\beta}$.

$$\begin{aligned} p(\tilde{\beta}|X, y_i) &\propto p(y_i|\tilde{x}_i, \tilde{\beta}) \times p(\tilde{\beta}) \\ &\propto \left(\frac{\exp(\tilde{\beta}^T \tilde{x}_i)}{1+\exp(\tilde{\beta}^T \tilde{x}_i)}\right)^{y_i} \left(1 - \frac{\exp(\tilde{\beta}^T \tilde{x}_i)}{1+\exp(\tilde{\beta}^T \tilde{x}_i)}\right)^{1-y_i} \times p(\tilde{\beta}) \end{aligned}$$

Again there is not a conjugate prior for easy computation or sampling from the posterior distribution.

Note, often Bayesians will use the probit link that enables the use of a Gibbs sampler on a latent variable, where $p_i = \Phi(\tilde{\beta}^T \tilde{x}_i)$, where $\Phi(.)$ is the cumulative distribution function of a standard normal random variable.

Metropolis Algorithm

Before considering the sampling for the GLMs detailed above, first consider a generic case. Assume we have a sampling model $p(y|\theta)$ and a prior distribution $p(\theta)$. In general:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta}$$

is hard to sample from due to the integration. However, given samples from $p(\theta|y)$ we can compute functions of the posterior distribution.

The challenge is drawing samples from $p(\theta|y)$, so far we have done this using Monte Carlo and Markov Chain Monte Carlo procedures (Gibbs sampler).

The goal is to create a sampler where the empirical distribution of the samples corresponds to the true samples. In otherwords,

$$\frac{\#\theta^{(s)} \in \theta_A}{\# \text{ of samples}} \approx p(\theta_A|y).$$

and if $p(\theta_A|y) > p(\theta_B|y)$ then we want more samples in θ_A than θ_B and specifically

$$\frac{\#\theta^{(s)} \in \theta_A}{\#\theta^{(s)} \in \theta_B} \approx \frac{p(\theta_A|y)}{p(\theta_B|y)}.$$

Q: So how do we design a sampler to meet this requirement?

Assume the current value of θ is denoted as $\theta^{(s)}$, suppose we propose a new value θ^*. The question is should we include this in our samples, or in otherwords should we move from $\theta^{(s)}$ to θ^*?

Ideally we would evaluate the ratio of:

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)}$$

however in many cases computing this value is difficult due to the required integration. Fortunately, it is not necessary as:

$$r = \left(\frac{p(y|\theta^*)p(\theta^*)}{p(y)}\right) \times \left(\frac{p(y)}{p(y|\theta^{(s)}p(\theta^{(s)}}\right) = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(s)})p(\theta^{(s)})}.$$

Now the question is, what do we do when:

- $r \geq 1$. In this case the proposed value θ^* is more attractive than the current value $\theta^{(s)}$ so we accept the next value $\theta^{(s+1)} = \theta^*$.

- $r < 1$. Now the current value is less attractive than the current value. However, the relative frequency of samples of θ^* to $\theta^{(s)}$ should be r. So with probability r set $\theta^{(s+1)} = \theta^*$, otherwise $\theta^{(s+1)} = \theta^{(s)}$.

This intuitive algorithm that we have devised is known as the Metroplis Algorithm, which is a special case of the Metropolis-Hastings algorithm where the proposal distribution (to select θ^*) is symmetric.

Formally, the Metropolis algorithm follows as: 1. Sample $\theta^*|\theta^{(s)} \sim J(\theta^*|\theta^{(s)})$. Typically $J(.)$ is a random walk function such as $J(\theta^*|\theta^{(s)}) \sim N(\theta^{(s)}, \gamma^2)$, where γ is thought of as the step size. The symmetric requirement means $J(\theta^*|\theta^{(s)}) = J(\theta^{(s)}|\theta^*)$, this restriction is not necessary in a slightly more complicated algorithm (Metropolis-Hastings).

2. Compute the acceptance ratio:

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)})} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(s)})p(\theta^{(s)})}.$$

3. Let

$$\theta^{(s+1)} = \begin{cases} \theta^* & \text{with probability min (r,1)} \\ \theta^{(s)} & \text{with probability 1 0 min (r,1)} \end{cases}$$

In practice to complete step 3, sample $u \sim Unif(0,1)$ and set $\theta^{(s+1)} = \theta^*$ if $u < r$.

Metropolis for Poisson - with R code

Note: in many cases we need to consider $\log(r)$ as the likelihood calculations can easily end up being numerically zero.

```
set.seed(11112018)
library(mnormt) # rmnorm

# Simulate Data
n <- 1000
p <- 3
beta.true <- c(2,.5,.5)
X <- matrix(c(rep(1,n),rnorm(n*2)),nrow=n,ncol=p)
theta <- exp(X %*% beta.true)
y <- rpois(n,theta)
hist(y,breaks='FD')
```

```
# Run Metropolis Algorithm

num.mcmc <- 10000
step.size <- .0001
accept.ratio <- rep(0,num.mcmc)
beta.mcmc <- matrix(0,num.mcmc,p)
beta.prior.var <- diag(p) * 100 # b ~n(0,100*I)

for (i in 2:num.mcmc){
  beta.star <- beta.mcmc[i-1,] + rmnorm(1,0,step.size * diag(p))

  #compute r
  theta.current <- exp(X %*% beta.mcmc[i-1,])
  theta.star <- exp(X %*% beta.star)

  log.p.current <- sum(dpois(y,theta.current,log=T)) + dmnorm(beta.mcmc[i-1,],0,beta.prior.var,log=T)
  log.p.star <- sum(dpois(y,theta.star,log=T)) + dmnorm(beta.star,0,beta.prior.var,log=T)

  log.r <- log.p.star - log.p.current

  if (log(runif(1)) < log.r){
    beta.mcmc[i,] <- beta.star
    accept.ratio[i] <- 1
  } else{
    beta.mcmc[i,] <- beta.mcmc[i-1,]
  }
}

#mean(accept.ratio)
#colMeans(beta.mcmc)

plot(beta.mcmc[,1],type='l')
abline(h=beta.true[1],lwd=2,col='gray')
plot(beta.mcmc[,2],type='l')
abline(h=beta.true[2],lwd=2,col='gray')
plot(beta.mcmc[,3],type='l')
abline(h=beta.true[3],lwd=2,col='gray')
```

Note the step size in as important consideration in a Metropolis-Hastings Algorithm. If the proposal is too large, the algorithm will tend to stay in one location for a large number of iterations as $\tilde{\beta}^*$ will be unattractive. If the step size is too small, virtually all of the proposals will be accepted, but the sampler will not efficiently explore the space. These can be seen visually and as a product of

Consider three figures below for an example of what happens as this varies

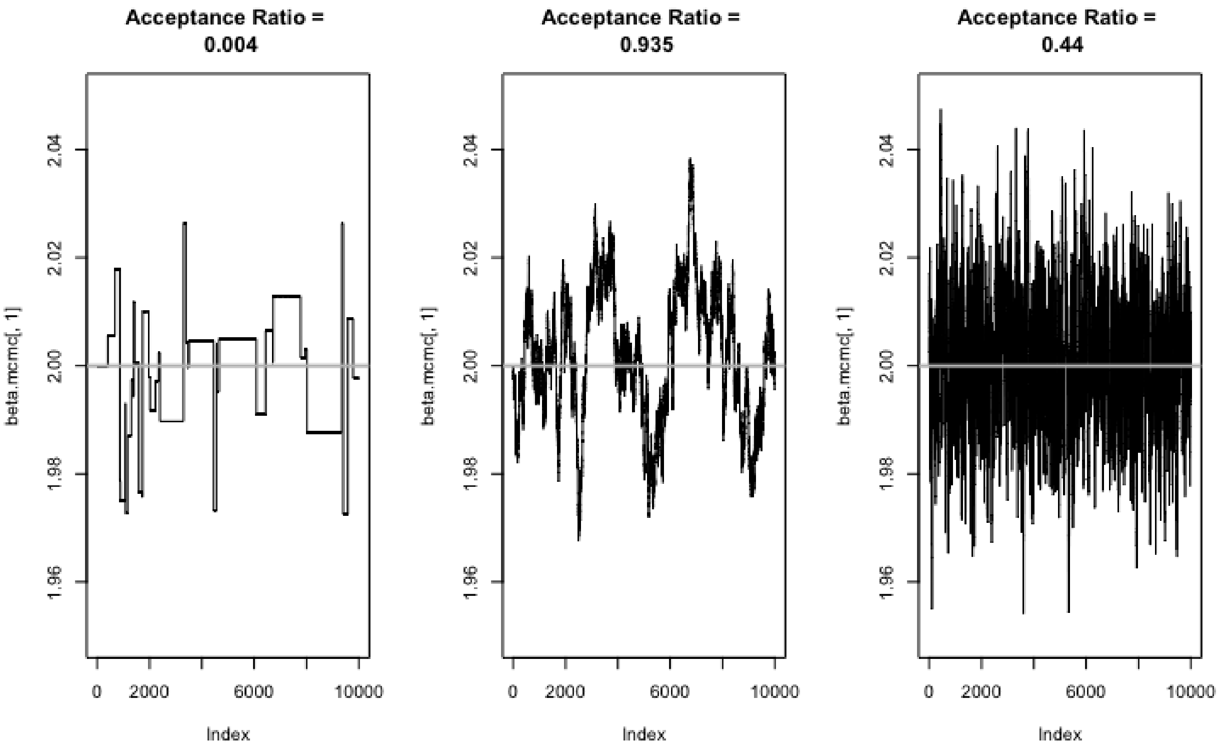

Figure 1: Trace plots for a step size that is too large, too small, and just right.

Metropolis-Hastings

Example. Assume we observe data from a negative-binomial distribution where the probability of a success (or failure) p is known. Use the following parametrization,

$$Pr(X = x) = \binom{x + r - 1}{x}(1-p)^r p^x \tag{1}$$

where r is the number of successes, x is the number of failures, and p is the probability of failure. The goal is to make inferences about r.

Assume that the parents favorite Halloween candy are Reese's peanut butter cups. Unbeknownst to their children they have decided to continue visiting homes until r more Reese's peanut butter cups have been obtained. In this example the probability of visiting a home and not getting a peanut butter cup (a failure) is p. The child is allowed to trick-or-treat until receiving r Reese's peanut butter cups.

Luckily for you the child keeps meticulous records and has recorded the number of homes visited in the last 4 years that did not have Reese's peanut butter cups.

- Consider using a Metropolis algorithm to learn the value of r. What will you use as a proposal distribution $J(\theta^*|\theta)$?

- Is your proposal distribution symmetric? In other words, does the $Pr(\theta^* \rightarrow \theta) = Pr(\theta \rightarrow \theta^*)$ for all θ^*, θ?

- Assume, you have devised a non-symmetric proposal, where:

$$\frac{J(1|0)}{J(0|1)} \approx 2.$$

In other words, you are twice as likely to propose a move from 0 to 1 than from 1 to 0. This could be due to a random step proposal near the end of the support for r. What implications do you suppose this has on the posterior probabilities ($Pr(r = 1|x)$ and $Pr(r = 0|x)$) using the standard Metropolis algorithm, with an acceptance ratio of proportional to $min(1, \alpha)$ where :

$$\alpha = \frac{p(\theta^*|y)}{p(\theta^{(s)})} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(s)})p(\theta^{(s)})}.$$

The Metropolis-Hastings algorithm permits non-symmetric proposals, by 'correcting' for values that are proposed more frequently. The acceptance ratio now can be written as:

$$\alpha = \frac{\pi(\theta^*)}{\pi(\theta^{(s)})} \times \frac{J(\theta|\theta^*)}{J(\theta^*|\theta)}, \tag{2}$$

where *pi* denotes the target distribution (typically a posterior probability factorized as the product of sampling model the prior in this course).

Q: What does the second term in the acceptance ratio do? Consider the case from above where $\frac{J(1|0)}{J(0|1)} \approx 2$ what impact does this have on moving from $\theta^{(s)} = 1$ to $\theta^* = 0$?

A: We are less likely to propose a step from 1 to 0 than vice versa, so the the resulting acceptance probability from 1 to 0 includes an inflating factor for the acceptance probability.

It is obvious that the Metropolis algorithm is a special case of Metropolis-Hastings, but how about a Gibbs sampler? Assume we are interested in moving from $\theta_1^{(s)}$ to θ_1^*.

$$\begin{aligned}
r &= \frac{\pi(\theta_1^*, \theta_2^{(s)})}{\pi(\theta_1^{(s)}, \theta_2^{(s)})} \times \frac{J(\theta_1^{(s)}|\theta_1^*, \theta_2^{(s)})}{J(\theta_1^*|\theta_1^{(s)}, \theta_2^{(s)})} \\
\text{the proposal is the full conditional} &= \frac{\pi(\theta_1^*, \theta_2^{(s)})}{\pi(\theta_1^{(s)}, \theta_2^{(s)})} \times \frac{\pi(\theta_1^{(s)}|\theta_2^{(s)})}{\pi(\theta_1^*|\theta_2^{(s)})} \\
&= \frac{\pi(\theta_1^*|\theta_2^{(s)})\pi(\theta_2^{(s)})}{\pi(\theta_1^{(s)}|\theta_2^{(s)})\pi(\theta_2^{(s)})} \times \frac{\pi(\theta_1^{(s)}|\theta_2^{(s)})}{\pi(\theta_1^*|\theta_2^{(s)})} \\
&= \frac{\pi(\theta_2^{(s)})}{\pi(\theta_2^{(s)})} = 1
\end{aligned}$$

So the Gibbs sampler is a very specific Metropolis-Hastings algorithm where the acceptance probability is always 1.

MCMC Theory

We have seen some empirical results that suggest these MCMC algorithms are reasonable, but what about theoretical guarantees? Recall the first (MC) chunk stand for Markov chain. The **Markov property**, in this case, is that each iteration of the sampler is only dependent on on the current values. First we establish some general properties of Markov chains that are useful for convergence of our MCMC algorithms.

1. **irreducible**. A reducible chain will have non-overlapping sets between which the algorithm is unable to move. The textbook cites a proposal mechanism that proposes values ±2. Hence, the chain would not be able to move from even to odd numbers and vice versa.

2. **aperiodic**. An aperiodic chain has values that can only be visited every k observations, which implies that $Pr(x^{(s)} = x) = 0$ for some values.

3. **recurrent**. A recurrent chain states that if $x^{(s)} = x$ then $\pi(x^{(s)} > 0$ so the chain must be guaranteed to return to x, eventually.

Recall, our Monte Carlo algorithms use central limit theorem ideas to show convergence of quantities compute from the posterior samples. However, given the dependence in the MCMC samples, we use theory of Markov Chains.

Ergodic Theorem. *If $\{x^{(1)}, x^{(2)}, \dots\}$ is an irreducible, aperiodic, and recurrent Markov chain, then there is a unique probability distribution π such that as $s \to \infty$, then*

1. $Pr(x^{(s)} \in A) \to \pi(A)$ for any set A;

2. $\frac{1}{S} \sum g(x^{(s)}) \to \int g(x)\pi(x)dx$.

Here $\pi()$ is denoted as the stationary distribution of the Markov Chain and has the following property: if $x^{(s)} \sim \pi$ and $x^{(s+1)}$ comes from that Markov Chain started by $x^{(s)}$ then $Pr(x^{(s+1)} \in A) = \pi(A)$. In other words, if a sample comes from the stationary distribution and used to generate more realizations from the Markov chain, then those appear according to the probabilities of π.

Now we need to show that $\pi(.) =$ the target distribution, $p_0()$ (joint posterior) for our MCMC examples. To verify this, assume $x^{(s)}$ comes from the target distribution and $x^{(s)}$ is generated from $x^{(s)}$ via the M-H algorithm, then we need to show $Pr(x^{(s+1)} = x) = p_0(x)$. Let x_a and x_b be any two x values. WLOG assume $p_0(x_a)J(x_b|x_a) \geq p_0(x_b)J(x_a|x_b)$. Then for MH, the probability of transitioning from $x^{(s)} = x_a$ to $x^{(s+1)} = x_b$ is equal to the probability of: 1. sampling $x^{(s)} = x_a$ from p_0,

2. proposing $x^* = x_b$ from $J(x^*|x^{(s)})$

3. accepting $x^{(s+1)} = x_b$.

The probability of these three steps is:

$$\begin{aligned} Pr(x^{(s)} = x_a, x^{(s+1)} = x_b) &= p_0(x_a) \times J(x_b|x_a) \times \frac{p_0(x_b)J(x_a|x_b)}{p_0(x_b)J(x_b|x_a)} \\ &= p_0(x_b)J(x_a|x_b). \end{aligned}$$

To go the other direction, where $x^{(s)} = x_b$ and $x^{(s+1)} = x_a$ the acceptance probability is 1 (as $p_0(x_a)J(x_b|x_a) \geq p_0(x_b)J(x_a|x_b)$). So $Pr(x^{(s)} = x_b, x^{(s+1)} = x_a) = p_0(x_b)J(x_a|x_b)$. This implies that the joint probability of observing $x^{(s)}$ and $x^{(s+1)}$ is the same for any order of x_b and x_a. The final step of the proof is to show that $Pr(x^{(s+1)} = x) = p_0(x)$.

$$\begin{aligned} Pr(x^{(s+1)} = x) &= \sum_{x_a} Pr(x^{(s+1)} = x, x^{(s)} - x_a) \\ &= \sum_{x_a} Pr(x^{(s+1)} = x_a, x^{(s)} = x) \\ &= Pr(x^{(s)} = x) \end{aligned}$$

Hence as $Pr(x^{(s)} = x) = p_0(x)$ then $Pr(x^{(s+1)} = x) = p_0(x)$.

Metropolis with Gibbs - Bayesian Kriging

Often a Gibbs sampler and a more vanilla Metropolis-Hastings style proposal can be used together in the same algorithm.

Recall the Bayesian spatial model:

$$y \sim N(X\tilde{\beta}, \sigma^2 H(\phi) + \tau^2 I),$$

Where $H(\phi)$ is a correlation matrix, such as $h_{ij} = \exp(-d_{ij}/\phi)$ where h_{ij} is the correlation between sites i and j and d_{ij} is the distance between sites i and j.

Sketch out an algorithm to sample the following parameters: $\tilde{\beta}$, σ^2, ϕ, and τ^2.