Stat 515- Take-home Exam

Brad Thompson

April 29, 2015

Let Y_i $EMG(\beta_0 + \beta_1 X, \sigma_i, \lambda)$ for the following problems.

1 Part 1

Let $\beta_0 = 5$, $\lambda = .4$, and $\sigma_i = 1$ for all i. Assume that the prior for β_1 is $N(0, 10)$. The goal is to construct a Metropolis-Hastings algorithm to approximate the posterior distribution $\pi(\beta_1|Y, X)$ To do this, we performed the following steps.

1. Create a chain that is the length of the number of iterations. Set an initial value for the chain. The initial value is somewhat arbitrary, but runtime can be decreased with smart choice of initial value. Initially, I used a N(0,10) random variable as the start point. I

2. For each subsequent spot in the chain do the following: Set the current value of β_1 to be the chain's value in the previous step. Draw a proposal β_1 from a distribution q. For this problem, I used a random walk, so q is a normal distribution centered at the current β_1 and with a variance τ^2: $q \sim N(curr\beta_1, \tau^2)$

3. Accept-reject step:
a). Determine alpha for the proposed value of β_1. Let y be the proposed value of β_1 while x is the current value.
$\alpha = min(1, \frac{h(y)q(y,x)}{h(x)q(x,y)})$. $h(x)$ is a distribution that is proportional to π up to a normalizing constant. Also, note that since q is symmetric, $q(y, x) = q(x, y)$ so α depends solely on the ratio of $h(y)$ to $h(x)$. It is often helpful to do this in log-scale, which is what I've done in my algorithm.
b). Draw a Unif(0,1) random variable (let's call it U). If $U < \alpha$, then we accept the proposed value of β_1 and set it as the new spot in the chain. If $U > \alpha$, we reject the proposed value and stay in the current state.
For this problem h(x) after taking logs is
$nlog(\frac{\lambda}{2}) + (\lambda(n\beta_0 + \beta_1 \sum x_i) + \frac{n}{2}\lambda^2\sigma^2 - \lambda * n * mean(Y) + \sum log(erfc) - \frac{1}{20}\beta_1^2$

From this algorithm, I obtained an estimate for β_1 equal to 7.25441. A simple 95 percent confidence interval obtained by looking at the 2.5 and 97.5 percentiles of the samples was (6.42, 7.87). See Figure 1-a in the plot section of the report for a smoothed density plot of the β_1 samples.

Now, for a few words on these approximations. First off, they do not appear to be entirely correct. The mean of the samples is around 7.25. Since the mean of an EMG distribution is $\frac{\mu+1}{\lambda}$, this value for β_1 corresponds to an expected value of about 21.88, whereas the actual mean of the given Y values for dataset 1 is 11.05. The distribution definitely appears to be off, but it is the best one I obtained and I am not entirely sure where I went wrong. I'm guessing there might be an error in calculating the $h(x)$ function, although I could not find an obvious place where I went wrong. Ignoring the supposed accuracy errors for now, there were a few things I looked for when evaluating the algorithm. The first was an ACF plot that dropped off after the first few lags. See Figure 1-b for the ACF plot of the chain. As we would like to see, the The second thing I looked at was acceptance rate. In class, it was mentioned that with an accurate variance parameter for the proposal distribution, the acceptance rate should be around 40

2 Part 2

Now assume we only know that $\sigma = 1$. The following is the steps I performed when constructing the M-H algorithm.
1. Create a chain that is the number of iterations by 3. The first column contains the lambda values, the second contains the β_0 values, and the third column contains the β_1 values. Set an initial value for the first row of the chain.

2. For each subsequent row in the chain, do variable-at-a-time M-H. Find the full conditional distributions for each variable and apply all-at-once M-H to each variable in turn. In more detail, sample from the full conditional distribution of lambda and then find the acceptance probability as outlined in the first problem. Again, the proposal distribution q is symmetric, so the acceptance probability depends only on $h(x)$.

For both this problem and the next, I initially also wrote a function that did all-at-once M-H, but I found that although the estimates were similar, acf plots, MCMC standard errors, and acceptance rates were better with variable-at-a-time. Therefore, only results and code from that function have been put into the final report.

Estimates (MCMCse in parentheses after):
λ: $E(\lambda) = .815(.0125)$
β_0: $E(\beta_0 = 2.369(.0135)$

β_1: $E(\beta_1) = 3.432(.0186)$

95 Percent Confidence Intervals (Found from 2.5th percentile and 97.5th percentile of draws):
λ: (.7127,.94325)
β_0: (2.1407, 2.6352)
β_1: (2.995,3.78664)

Using the R function cor, an estimate of the correlation between β_0 and β_1 is -.749
See Figures 2-a-2-c for density plots for each of the three variables.

Although the estimates are much closer than in problem 1, they are still off by a little bit. . Looking at the expected value of Y, which is $\frac{\mu+1}{\lambda}$, we get an expected value of 6.3, which is pretty close to the actual mean of Y, 5.3. The acceptance rates are somewhat lower than we'd like with an accurate proposal variance, but the ACF plots for each variable are good, showing dependence only on low lag values. ESS is also relatively low because run time is higher than ideal.

3 Part 3

Estimates (MCMCse in parentheses after): λ: $E(\lambda)$= .164 (.00354)
β_0: $E(\beta_0)$=.1545(.0125)
β_1: $E(beta_1)$= 2.465 (.0213)

95 Percent Confidence Intervals:
λ: (.1471,.1728)
β_0: (-.1874, .4791)
β_1: (1.9245, 3.025)

See Figures 3-a through 3-c in the plots section for density plots for the three variables.
The one thing I looked at was going between V-MH and A-MH, but better estimates were obtained with V-MH, as well as better heuristics for the algorithm. I changed the proposal variance slightly to get an acceptance rate closer to .4 and changed the start values, but kept the meat of the algorithm the same as for problem 2.

Distributions used in problems 2 and 3:
Joint Posterior Distribution- Product of the priors with the EMG pdf:
$nlog(\frac{\lambda}{2}) + \lambda(n\beta_0 + \beta_1 \sum x_i) + \frac{n}{2}\lambda^2\sigma^2 - \lambda * n * mean(Y) + \sum log(erfc) - \frac{1}{20}\beta_0^2 - \frac{1}{20}\beta_1^2 - .99log(\lambda) - .01\lambda$

Full Conditional Distributions:

λ: $nlog(\frac{\lambda}{2}) + \lambda(n\beta_0 + \beta_1 \sum x_i) + \frac{n}{2}\lambda^2\sigma^2 - \lambda * n * mean(Y) + \sum log(erfc) - .99log(\lambda) - .01\lambda$

β_0: $\lambda(n\beta_0 + \beta_1 \sum x_i) + \sum log(erfc) - \frac{1}{20}\beta_0^2$

β_1: $\lambda(n\beta_0 + \beta_1 \sum x_i) + \sum log(erfc) - \frac{1}{20}\beta_1^2$