Metropolis Algorithm

and Logistic Regression

Yingbo Li

Clemson University

MATH 9810

Metropolis Algorithms

- Not all Gibbs samplers have "nice" full conditional distributions
- MCMC technique called the Metropolis algorithm for "not nice" full conditionals
- Chapter 10 in Hoff covers this and Metropolis Hastings algorithm, which is a generalization of Metropolis algorithm
- Today we apply Metropolis algorithm to logistic regression

Metropolis Algorithm

Suppose we want to estimate $p(\theta|Y)$ for some scalar θ

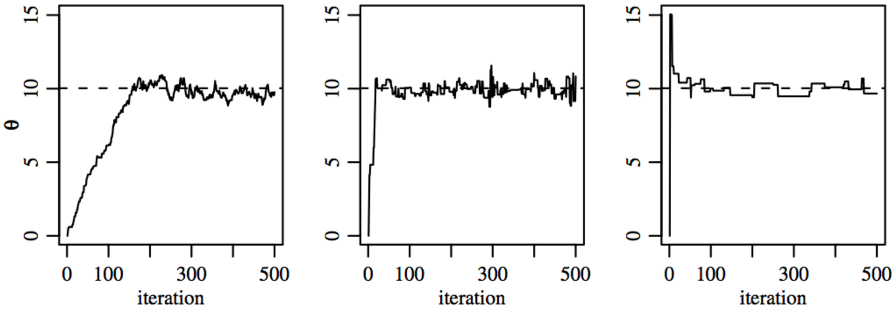
1. Start with an initial guess at θ, say $\theta^{(1)}$.
2. Given $\theta^{(s)}$, generate a value $\theta^{(s+1)}$ as follows:
   1. Draw plausible value of θ from some symmetric distribution $J(\theta|\theta^{(s)})$ that is easy to simulate, like a $N(\theta^{(s)}, c^2)$, i.e.,
   $$\theta^* \sim J(\theta \mid \theta^{(s)})$$
   2. If θ^* is more likely under $p(\theta|Y)$ than $\theta^{(s)}$, then we keep it as a plausible value of θ, i.e., $\theta^{(s+1)} = \theta^*$.
   3. If θ^* is less likely under $p(\theta|Y)$ than $\theta^{(s)}$, then we let $\theta^{(s+1)} = \theta^*$ with probability
   $$r = \frac{p(\theta^*|Y)}{p(\theta^{(s)}|Y)} = \frac{p(Y|\theta^*)p(\theta^*)}{p(Y|\theta^{(s)})p(\theta^{(s)})}$$
3. Repeat Step 2 until MCMC convergence (or for a large number of iterations, say $S = 10^5$).

Features of Jumping Distribution

- $J(\theta|\theta^{(s)})$ is called the proposal distribution
- $J(\theta \mid \theta^{(s)})$ must depend only on $\theta^{(s)}$ and not previous values of θ in the chain
- $J(\theta \mid \theta^{(s)})$ must be a symmetric density, i.e., $J(\theta^{(s+1)} \mid \theta^{(s)}) = J(\theta^{(s)} \mid \theta^{(s+1)})$
- $J(\theta|\theta^{(s)})$ must be such that you can get to any value of the parameter space for θ eventually from any $\theta^{(s)}$
- $J(\theta|\theta^{(s)})$ must be such that you don't return periodically to any particular value of θ

Tuning Metropolis Algorithm

- You get to specify $J(\theta|\theta^{(s)})$, e.g., proposal variance.
- Small proposal steps: high acceptance rate, but the moves are never very large so the Markov chain is sticky and highly correlated.
- Large proposal steps: quickly moves to posterior mode but gets "stuck" for long periods, since proposed values are usually far away from the mode.



Tuning Metropolis Algorithm

- Goal is to select one that leads to roughly 35% of new proposed $\theta^{(s+1)}$ accepted (or at least between 20% to 50%).
- Tuning: try short runs and record percentage of acceptances, and reset J as necessary to achieve near 35%.
- For example, with a normal jumping distribution, reset the variance c^2 until you get about 35% acceptances.
- Alternatively, use tuning-free algorithms, e.g., adaptive MCMC.

Multi-parameter MCMC

- With multiple parameters, a common strategy is to set up a Gibbs sampler overall, and update each parameter using
  - Draws from the full conditional when they are readily available
  - Draws from a Metropolis step otherwise

Note: there are lots of other types of MCMC algorithms that are variants of the Metropolis algorithm. We will talk about Metropolis-Hastings next time (when proposal is not symmetric).

Application: Logistic Regression

- Often the goal of analysis is to predict a binary outcome.
- Logistic regression is useful tool for doing this.
- Posterior distributions for logistic regression parameters not amenable to direct simulation via MC
- Full conditional distributions also messy, so Gibbs sampler not option
- Use Metropolis algorithm

Example: Pima Indian diabetes data

- Contains records of 532 independent patients
- Binary response: whether the patient has diabetes according to WHO criteria.
- In this dataset, 7 predictors were collected. Here, we just use BMI.
- Learn association between obesity and diabetes.
- Could we have predicted the probability of diabetes from BMI?

Can We Use Linear Regression?

Why not use a linear model of Diabetes on BMI to estimate prediction equation?

- Outcomes are binary, not normally distributed
- Could get predicted values less than zero or greater than one

Logistic regression models the probability of diabetes, π_i, where i indexes a patient. For $i = 1, \ldots, n$,

- Let $y_i = 1$ if diabetes, and let $y_i = 0$ if healthy
- Let x_i be the BMI

Logistic Regression Model

The model is

$$
\begin{aligned}
y_i &\sim Bin(\pi_i, 1) \\
log(\frac{\pi_i}{1-\pi_i}) &= \beta_0 + \beta_1 x_i
\end{aligned}
$$

Why use log of odds of π_i?

- $\pi_i = \beta_0 + \beta_1 x_i$ could imply π_i not in [0,1]
- $log(\pi_i) = \beta_0 + \beta_1 x_i$ could imply $\pi_i > 1$
- $log\left(\frac{\pi_i}{1-\pi_i}\right)$ can take on values between $(-\infty, \infty)$

To convert to probability scale, use $\pi_i = \frac{\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}$

Interpretation of Coefficients

Consider odds $\frac{\pi_i}{1-\pi_i} = \omega_i = \exp(\beta_0 + \beta_1 x_i)$

- When all explanatory variables equal zero, the odds of diabetes are $\exp(\beta_0)$
- If you center x_i first by subtracting $\bar{x}$ from each x_i, then $\exp(\beta_0)$ is odds of diabetes at average temperature
- The ratio of odds (or odds ratio) at $x_i = A$ to odds at $x_i = B$ for fixed values of any other explanatory variables is

$$\text{Odds ratio} = \frac{\omega_A}{\omega_B} = \exp(\beta_i(A - B))$$
$$\text{Odds}(x_j = A) = \exp(\beta_i(A - B))\text{Odds}(x_i = B)$$

- Coefficients are log odds ratios

Frequentist Estimation of Coefficients

- Need a numerical differentiation algorithm to find MLEs, $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)$
- MLEs are approximately normally distributed in large samples
  - $\hat{\beta}_j$ has mean β_j
  - $\hat{\beta}_j$ has estimated variance $\mathsf{SE}^2_{\beta_j}$
- Asymptotic distribution for $\hat{\beta}_j$ is $N(\beta_j, \mathsf{SE}^2_{\beta_j})$
- $(1-\alpha)100\%$ CI based on normal theory:

$$\hat{\beta}_j \pm Z_{\alpha/2}\mathsf{SE}_{\beta_j}$$

- Exponentiate to obtain interval for odds ratio.

Bayesian Estimation

First, write down the likelihood function $f(Y|X,\beta)$

$$\begin{aligned} f(Y|X,\beta) &= \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} \\ &= \prod_{i=1}^{n} \left(\frac{\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}\right)^{y_i} \left(\frac{1}{1+\exp(\beta_0+\beta_1 x_i)}\right)^{1-y_i} \end{aligned}$$

We also need priors for β_0 and β_1. No conjugate priors here, so we'll have to use MCMC. Let's use a bivariate normal:

$$p(\beta_0,\beta_1) \sim \mathsf{N}_2(\mu,\Sigma)$$

where $\mu = (0,0)$ and Σ has variances of 100 and covariances of zero.

Metropolis Algorithm

- Work on the log scale to avoid computational problems
- Proposal distribution: bivariate normal distribution around current values, with diagonal covariance matrix with variance c^2
- MCMC sampler shows high autocorrelations: run for LONG time (100,000) to get decent effective sample sizes