MCMC

Paul E. Johnson[1] [2]

[1]Department of Political Science

[2]Center for Research Methods and Data Analysis, University of Kansas

2011

Overview

- Restate the Problem
- MH
- MCMC

Bayes Rule

- Recall we want the posterior distribution, the probabilty that a particular hypothesis "*hyp*" is correct, in light of the "*data*".

$$Bayes\ Rule:\ p(hyp|data) = \frac{p(data|hyp)p(hyp)}{p(data)} \tag{1}$$

- And we often throw away the denominator because it is a "constant" in this context

$$Bayes\ Rule:\ p(hyp|data) = p(data|hyp)p(hyp) \tag{2}$$

- Use θ for the hypthesized parameter values

$$p(\theta|data) = p(data|\theta)p(\theta) \tag{3}$$

- Recall "*data*" is a collection of observations in a sample

$$data = (data_1, data_2, data_3 \ldots, data_N) \tag{4}$$

Likelihood $\times$ prior

- $p(data|\theta)$ is a likelihood function.
- Assuming "independence",

$$p(data|\theta) = \Pi_{i=1}^{N} p(data_i|\theta) \tag{5}$$

- So Bayes theorem means we need

$$p(\theta|data) = \left(\Pi_{i=1}^{N} p(data_i|\theta)\right) \times prior(\theta) \tag{6}$$

What does p(θ|data) Look Like?

- That's the million dollar question. What outcomes are most likely? How "widely spread" is it.
- In Jim Albert's book, one approximate approach is the Laplace approximation. This finds the "mode" of the posterior, approximately.
- Before high speed (parallel) computing, that was about as good as we can do (and it is still a useful "pedagogical" approach).

Remember "acceptance sampling"

- In my lecture on "drawing random samples", it was shown that one can draw random cases from a distribution by choosing values from a candidate distribution and then accepting "the right proportion" of them.
- If θ is a one dimensional thing–a single parameter–then we could sample from $p(\theta|data)$ by ordinary acceptance sampling.
- As long as the proposal distribution covers the whole range of θ's possible values, this is a manageable project.

If θ is Complicated...

- Suppose the parameter vector is larger

$$\theta = (\theta_1, \theta_2, \ldots, \theta_m) \tag{7}$$

- Problem: find a "good" multidimensional proposal distribution
- Draw a reasonably large sample (and do so reasonably quickly, within our lifetimes)

Metropolis Algorithm

- Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. "Equation of State Calculations by Fast Computing Machines." The Journal of Chemical Physics 21(6): 1087.
- Collect a sequence of values $\theta = \{\theta^1, \theta^2, \ldots, \theta^T\}$ that will approximate $p(\theta|data)$
- In order to make sure that we "explore" the space, use a Markov Chain to draw new suggested points.
- Recall Markov Chain: "one step dependence"

$$\theta^t = some\ function\,(\theta^{t-1}, any\ other\ info\ avail.\ at\ t-1) \qquad (8)$$

- The power of general theorems on Markov Chains comes into play, so that the Metropolis algorithm does not have to prove everything completely from scratch.

Implementation

- Metropolis, et al, proposed to begin at some arbitrary point θ^0 . Calculate $p(\theta^0|data)$.
- Then draw a new point for consideration at random by perturbation

$$\theta^* = \theta^0 + \text{noise} \tag{9}$$

- Then calculate $p(\theta^*|data)$.
- If $p(\theta^*|data) > p(\theta^0|data)$, that means hypothesis θ^* is "more likely" to be correct. So we accept θ^* into our collection of points. Call that θ^1.

Mountain Climbing is Overrated

- If we only accept points such that $p(\theta^*|data) > p(\theta^{t-1}|data)$, then we are "hill climbing".
- Suppose we are lucky, and there is just one "global maximum" (no local maxima), then this algorithm will find the "most likely value" of θ and it will stay there forever.
- That's not enough because
  - It does not "explore" the full extent of possible values of θ
  - We would like to say "95% of the outcomes from $p(\theta|data)$ are between points A and B, and this does not allow such statements.

Go Sideways, or Down (sometimes)

- Metropolis et al proposed to accept some values of θ^* for which

$$p(\theta^*|data) < p(\theta^{t-1}|data) \tag{10}$$

- The chance of accepting a "lower" step was

$$r_m = \frac{p(\theta^*|data)}{p(\theta^{t-1}|data)} \tag{11}$$

- So if θ^* is "almost as likely" as θ^{t-1}, then θ^* is very likely to get added as θ^t.
- Even if θ^* is far less likely than θ^{t-1}, it has a chance of getting selected.
- Thus, there is at least "a chance" that even very unlikely spots will be visited.

Tweaks

- Fiddle around with the procedure for drawing suggested points: Proposal density.
  - Random Walk (depends on θ^{t-1})
  - Independent draws (does not depend on θ^{t-1})
- Fiddle around with the criterion for deciding whether to accept points into the chain.
  - Hastings's proposal (1970) (In Jackman's notation)

$$r = \frac{p(\theta^*|data)}{p(\theta^{t-1}|data)} \cdot \frac{J_t(\theta^*, \theta^{t-1})}{J_t(\theta^{t-1}, \theta^*)} \tag{12}$$

  - W. K. Hastings (1970) Monte Carlo Sampling Methods Using Markov Chains and Their Applications, Biometrika, 57(1): 97-109
  - J_t is the jumping distribution
  - Hastings (p. 100) notes this is same as Metropolis if J_t is reversible,

$$J_t(\theta^*, \theta^{t-1}) = J_t(\theta^{t-1}, \theta^*) \tag{13}$$

  - Use of right J_t may improve "mixing" (exploration of parameter space) without raising the number of wasted (rejected) proposals.

Example Usage of MH

Count model from MCMCpack in Jackman

Practical Problems of MH

- Slow: creating draws in m dimensional space
- Slow: rejection rate high
- Autocorrelation: Must aggressively "thin" (throw away observations)

Gibbs Sampling

- Recall that
$$Pr(x, y, z) = Pr(x|y, z) \cdot Pr(y, z) \tag{14}$$
- and
$$Pr(y, z) = Pr(y|z) \cdot P(z) \tag{15}$$
- So the 3-tuple's (x, y, z) distribution can be thought of as a series of conditional distributions.

Gibbs Sampling

- The posterior distribution

$$p(\theta|data) \propto p(data|\theta)p(\theta) \tag{16}$$

- Draw each parameter conditionally on all the others

$$\begin{aligned} &draw\ \theta_1\ from\ g_1(\theta_1|\theta_{-1}, data) \\ &draw\ \theta_2\ from\ g_2(\theta_2|\theta_{-2}, data) \\ &\ldots \\ &draw\ \theta_m\ from\ g_m(\theta_m|\theta_{-m}, data) \end{aligned} \tag{17}$$

Gibbs Sampling

- The distribution of θ from those draws eventually converges to $p(\theta|data)$
- At the start, $g(\theta)$ does not resemble $p(\theta|data)$, so it is necessary to throw away a chunk of observations. ("burn in" iterations)

When This is Done

- We have a sample from the multivariate density $(\theta_1, \theta_2, \theta_3 \ldots, \theta_m)$
- That's m "columns" of estimates, each of which is an "exact sampling distribution".
- How is that different from Maximum Likelihood (?)
- Can treat each column as a "marginal posterior density", (Jackman, p 220).
- King's Clarify software uses these columns to calculate predicted values

Why Doesn't a Metropolis Algorithm Require a Burn In Period?

- MH can use every sample drawn
- Gibbs sampling cannot. Why the difference
- MH accepts suggestions in proportion to the desired probability (acceptance sampling)
- Gibbs accepts all draws, without checking that any particular one matches the desired distribution
- The premise is that Gibbs will be more efficient because it is so much simpler to work with one parameter at a time, even though some must be discarded.

An Ordinary Regression Model

	Garbage Can Regression	
	Estimate	S.E.
(Intercept)	-29.565*	(8.66)
V045117L	-10.788	(7.931)
V045117SL	2.375	(7.933)
V045117M	5.612	(7.819)
V045117SC	10.141	(8.257)
V045117C	17.499*	(8.341)
V045117EC	26.398*	(9.783)
V043116WD	24.605*	(4.032)
V043116ID	22.365*	(3.765)
V043116I	40.605*	(5.165)
V043116IR	65.212*	(4.59)
V043116WR	67.239*	(4.515)
V043116SR	82.348*	(4.722)
V043210No	7.911*	(2.615)
V043210Med	6.781	(5.84)
V0432133. Worse	-25.083*	(3.278)
V0432135. The same	-7.382*	(3.317)
V045145X2. Very good	-7.623*	(2.528)
V045145X3. Somewhat good	-14.505*	(3.387)
V045145X4. Not very good	-14.672*	(6.141)
V045145X7. Don't feel anything VOL	-26.238*	(8.668)
V041109AF	0.284	(2.19)
N	803	
RMSE	29.95	
R^2	0.712	
AIC	7762.133	

Treat some predictors as Numeric

	Garbage Can Regression	
	Estimate	S.E.
(Intercept)	-87.134*	(6.888)
Ideology	7.47*	(1.063)
Party ID	15.003*	(0.692)
AntiGay	8.892*	(2.294)
Economy	-0.94	(1.714)
Flag Love	-7.447*	(1.301)
V041109AF	-0.504	(2.266)
N	803	
RMSE	31.831	
R^2	0.669	
AIC	7845.258	

* $p \leq 0.05$

MCMCpack has regression

Inteface:

```
MCMCregress(formula, data = NULL, burnin = 1000,
    mcmc = 10000,            thin = 1, verbose = 0,
    seed = NA, beta.start = NA,            b0 = 0, B0
    = 0, c0 = 0.001, d0 = 0.001,
    marginal.likelihood = c("none", "Laplace", "
    Chib95"), ...)
```

Count Regression as a Hierarchical Bayesian Model

- Suppose some "count" model follows a Poisson distribution. The *i*'th case:
-
$$f(y_i|X,\beta) = \frac{X_i\beta^{y_i}}{y_{i!}} exp(-X_i\beta) \tag{18}$$
- β is a vector of parameters, X_i is a row of observations for the $i'th$ case
- Across a sample of N cases, that leads to a likelihood
$$f(y|X,\beta) = \Pi_{i=1}^{N} \frac{X_i\beta^{y_i}}{y_{i!}} exp(-X_i\beta) \tag{19}$$

Frailty

- Throw in ϵ_i like so:

$$f(y|X,\beta) = \Pi_{i=1}^{N} \frac{X_i \beta^{y_i}}{y_{i!}} exp(-X_i\beta + \epsilon_i) \tag{20}$$

- If 0, then this is just the same old model.
    - However, if ϵ_i has some noise in it, then it will cause the observations to fluctuate more.
- Any probability model for which $E[\epsilon_i] = 0$ can be used.

Rewrite Like This

- Rearrange

$$f(y|X,\beta) = \Pi_{i=1}^{N} \frac{X_i \beta^{y_i}}{y_{i!}} exp(-X_i\beta) exp(\epsilon_i) \tag{21}$$

- Now think of the multiplicative error $\delta_i = exp(\epsilon_i)$ as something that has expected value 1.
- The benefit here is that the terms are multiplicatively separated

$$f(y|X,\beta) = \Pi_{i=1}^{N} \frac{X_i \beta^{y_i}}{y_{i!}} exp(-X_i\beta)\delta_i \tag{22}$$

- If $\delta \sim Gamma(\alpha,\alpha)$, Recall $E[\delta] = \alpha/\alpha{=}1$. However, the Variance can differ, $Var[\delta] = 1/\alpha$.
- That gives y a Negative Binomial distribution. (same expected value as Poisson, bigger variance.)

Estimation

- α is a "hyper parameter"
- We need to estimate β and α
- Because of Gibbs sampling, we can alternate between drawing values of α and β from this posterior.

MCMCpoisson MCMCpack

```
Markov Chain Monte Carlo for Poisson Regression
Description:
     This function generates a sample from the
        posterior distribution      of a Poisson
        regression model using a random walk
        Metropolis     algorithm. The user
        supplies data and priors, and a sample
        from     the posterior distribution is
        returned as an mcmc object, which     can
         be subsequently analyzed with functions
        provided in the coda     package.
Usage:
```

MCMCpoisson MCMCpack

```
MCMCpoisson(formula, data = NULL, burnin = 1000,
    mcmc = 10000,           thin = 1, tune = 1.1,
    verbose = 0, seed = NA,  beta.start = NA,
            b0 = 0, B0 = 0, marginal.likelihood = c
    ("none", "Laplace"), ...)
```

Priors: Prior on β is $MVN(b_0, B_0^{-1})$ (B_0 is the prior's "precision", the reciprocal of variance).

Phony Example I

```
library(MCMCpack)
counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
outcome <- gl(3, 1, 9)
treatment <- gl(3, 3)
posterior <- MCMCpoisson(counts ~ outcome + treatment)
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The Metropolis acceptance rate for beta was 0.27318
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
summary(posterior)
```

```
Iterations = 1001:11000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                 Mean     SD  Naive SE Time-series SE
(Intercept)  3.025836 0.1770 0.001770       0.008396
outcome2    -0.450066 0.1965 0.001965       0.008228
outcome3    -0.284293 0.1913 0.001913       0.008052
```

Phony Example II

```
treatment2    0.001680 0.2008 0.002008        0.007786
treatment3   -0.006043 0.2021 0.002021        0.008560

2. Quantiles for each variable:

                2.5%     25%       50%     75%    97.5%
(Intercept)  2.6618  2.9059  3.030113  3.1535  3.35518
outcome2    -0.8284 -0.5782 -0.456604 -0.3190 -0.05947
outcome3    -0.6764 -0.4115 -0.285502 -0.1569  0.08712
treatment2  -0.3878 -0.1381 -0.000305  0.1427  0.39398
treatment3  -0.3999 -0.1459 -0.003043  0.1302  0.38832
```

```
plot(posterior)
```

Default Plot