

# Introduction to Bayesian Modeling

Richard Condit

National Center for Ecological Analysis and Synthesis

March, 2008

Accompanying files: *RcmdBayesWorkshop.r*, *RfuncBayesWorkshop.r*

This is a workshop on Bayesian statistics illustrated with the computer programming and statistical language *R*. The following notes give a brief overview of the posterior probability distribution, explain how it is used in Bayesian statistics, and then show how the Metropolis and Gibbs methods can be used to estimate posterior distributions. The main goal is to use *R* to illustrate the ideas, showing probability distributions and likelihood functions graphically and walking command-by-command through a Gibbs sampler. There are two companion files with the *R* commands and functions described here.

*R* commands are in the file *RcmdBayesWorkshop.r*, ready to run, one at a time, by pasting in an *R* command window. One change will be necessary for use with either Windows or Macintosh: they are designed for Unix, with the `X11()` function for opening graphics windows. These must be changed for a Windows or a Mac. The Windows function is included with a `#` in front of it. The other file is *RfuncBayesWorkshop.r*, which has several functions called by the commands. A user need not ever look inside this file, but is welcome to do so. There is a function running a Gibbs sampler which might be useful to someone trying to learn to do Bayesian programming in *R*.

## The Posterior Distribution

Likelihood models are based on the sampling distribution or likelihood function, defined as the probability of observing data given a model or hypothesis:  $p(y|H)$ , where  $y$  represents the data and  $H$  the hypothesis or model. If the hypothesis is represented by parameters  $\theta$  for a specified model, the likelihood function is  $p(y|\theta)$ .

The posterior distribution is exactly the opposite, the probability of alternate hypotheses given the data,  $p(H|y)$  or  $p(\theta|y)$ . This is the staple of the Bayesian approach: the posterior distribution is the truly useful probability, since it begins with what is known and provides inference about what is not. The likelihood function gives the probability of seeing what we already saw.

### The Binomial Distribution and its Posterior

Consider a population of  $N$  individuals, and then a recensus some time later that finds  $S$  survivors; the remaining  $N - S$  died. Given the probability that any one individual survives,  $\theta$ , then the binomial provides a likelihood function for observing  $S$  of  $N$ ,

$$prob(S) = Dbinom(S, N, \theta) = \binom{N}{S} \theta^S (1 - \theta)^{N - S}.$$

A graph of  $prob(S)$  vs.  $S$  is the likelihood function. To create a posterior, make the same calculation for one value of  $S$  but for many different values of  $\theta$ . A graph of  $prob(S)$  vs.  $\theta$  illustrates the notion of the posterior distribution:  $S$  and  $N$  are known,  $\theta$  is not. What values of  $\theta$  are most likely to account for the observations? These calculations are illustrated in the section of *R* commands headed *Binomial Probability Distribution* in the file *RcmdBayesWorkshop.r*, with a graph of  $\theta$  vs. the result called *postlike*, which is calculated from the *R* function *dbinom*.

Execute lines 7 through 51 of *RcmdBayesWorkshop.r* to illustrate the binomial likelihood and its posterior distribution.

The graph of *postlike* vs.  $\theta$  is not precisely the posterior distribution. It is not a probability distribution: it does not integrate to 1. The integral of this distribution is illustrated in *RcmdBayesWorkshop.r* by calculating  $k$  as the sum of *postlike*.

Execute lines 56 through 78 of *RcmdBayesWorkshop.r* to illustrate normalization of the posterior distribution by finding the sum *postlike*.

Beyond this, the posterior as defined by the likelihood function ignores prior probabilities of  $\theta$ . Bayes' rule gives the precise relationship between likelihood and posterior:

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

The left side is the posterior, and the likelihood  $p(y|\theta)$  appears on the right side;  $p(\theta)$  is the prior probability distribution. The denominator is the total probability of observing  $y$ , given all possible  $\theta$ , which is a constant. If we ignore the prior, by assuming that all  $\theta$  are equally likely, then Bayes' rule is  $p(\theta|y) = kp(y|\theta)$ , where  $k$  is a constant. This means that the posterior distribution is proportional to the likelihood function.

The graph of *postlike* as a function of  $\theta$  is thus a good representation of the posterior, even if it is not exact. Assuming no prior information, it has the same shape as the posterior. This makes it very useful for drawing inference about  $\theta$ . The simplest inference, widely used in model fitting, is to find the value of  $\theta$  with the highest probability. In the binomial example, it is always at  $\theta = \frac{S}{N}$ . That is the maximum-likelihood estimate of  $\theta$ .

The main tool of a Bayesian analysis is to use more than the maximum and examine the complete posterior distribution to draw inference about how well we know  $\theta$ . In the binomial example, with the graph of *postlike* vs.  $\theta$ , this can be illustrated qualitatively: values of  $\theta$  for which the binomial probability is near 0 are unlikely to be true, whereas those near the peak are.

In the specific example of the binomial, the posterior distribution,  $p(\theta|y)$ , can be calculated exactly. It turns out to be the beta-distribution with parameters  $S+1$  and  $N-S+1$ . The function *dbeta* with those two parameters can thus be used instead of the surrogate  $p(S|\theta)$ . These are illustrated with use of the R commands *dbeta* and *qbeta* in *RcmdBayesWorkshop.r*.

Execute lines 83 through 101 of *RcmdBayesWorkshop.r* to use the beta distribution as the posterior of the binomial.

### **Random Draws on the Posterior**

One interesting approach widely used in Bayesian statistics is to make random draws on the posterior distribution. In the binomial example, this means creating random values of the  $\theta$  whose probability matches the posterior distribution (as illustrated by the graph of *postlike* vs.  $\theta$  and the graph of *dbeta* as a function of  $\theta$ ). The R function *rbeta* is illustrated in *RcmdBayesWorkshop.r*, with a histogram to show that the random draws really do match the form of the posterior distribution.

Execute lines 104 through 110 of *RcmdBayesWorkshop.r* to use the R function *rbeta* to make random draws on the posterior of the binomial distribution.

There are two main uses for random draws. One is to propagate a parameter forward in predictions or simulations. Imagine needing the parameter  $\theta$  in a population model that predicts future population size. The maximum likelihood estimate of  $\theta = \frac{S}{N}$  could be used, but this ignores uncertainty in the parameter. Instead, the predictions could be run repeatedly, each time with a different random draw from the posterior. This precisely propagates the uncertainty.

The second reason for random draws of the posterior applies in cases where there is no closed form for the posterior. The binomial example is unusual: more often, the likelihood function can be used as a surrogate for the posterior, but it is not a known probability distribution. In these circumstances, there is a clever approach known as the Metropolis-Hastings algorithm that creates random draws even without knowing the exact posterior distribution. It only requires the likelihood function as a surrogate: the function must be proportional to the posterior, but need not be exact.

### **The Metropolis-Hastings Method for Sampling the Posterior**

This is a widely useful approach for drawing inference on a posterior distribution, because it can be used in nearly any circumstance, no matter the form of the likelihood function, and it easily

incorporates prior probabilities in any form. It simply assumes that a likelihood function that is proportional to the posterior can be generated.

The precise algorithm is illustrated by the function *showBinomialMCMC*, which is defined in *RfuncBayesWorkshop.r*. The first call to this function in *RcmdBayesWorkshop.r* offers a graphical walk through a Metropolis search. The search involves 3 key steps: 1) start with any permitted value of the parameter of interest, here  $\theta$  of the binomial; 2) draw a second  $\theta$  at random, but with equal probability of being  $>$  or  $<$  the first; 3) decide whether to accept the new  $\theta$  based only on the likelihood comparison between it and the previous  $\theta$ .

Execute lines 117 through 123 of *RcmdBayesWorkshop.r* to start a step-by-step walk through a Metropolis-Hastings sampler of the binomial likelihood. Each newly chosen  $\theta$  appears first as a red circle. If it is accepted, it changes to blue and then is filled in with a blue point. If it is rejected, it disappears.

The exact algorithm for step 2 is to draw a random normal variable centered on the initial  $\theta$ , using a predetermined standard deviation called the *step size*. Ideally, the step size should be chosen so that about 25% of all steps are accepted, following the acceptance algorithm given below.

The exact acceptance algorithm used in step 3 is based on the ratio of the new likelihood to the old likelihood. The likelihood is the probability of observing  $\theta$  given the pre-defined likelihood function. If the ratio of new to old is  $> 1$ , then the new value is always accepted. If the ratio  $< 1$ , then the new value is accepted with probability exactly equal to the ratio. Thus, steps downward in likelihood are sometimes accepted.

The second and third calls to the function *showBinomialMCMC* in *RcmdBayesWorkshop.r* illustrate long runs of the Metropolis algorithm based on the binomial likelihood. They demonstrate that the sequence of values of  $\theta$  produced exactly match the form of the likelihood function: they are random draws on the posterior distribution.

Execute line 124 of *RcmdBayesWorkshop.r* to illustrate a 100-step Metropolis-Hastings sampler of the binomial likelihood. Execute lines 125 through 133 to run a much longer sampling (4000 steps), then illustrate that it produces a distribution identical to *rbeta*.

### **Step-Size Adjustment in Metropolis-Hastings**

Getting the step size correct is important. Although the algorithm works regardless of step size, steps that are too big or too small make for very inefficient runs. The general recommendation is to aim for an acceptance rate of 0.25, that is, one in four Metropolis steps should be accepted (with only 2 or 3 parameters, the acceptance rate is supposed to be slightly higher, 0.3–0.4, although this probably makes little difference). Any acceptance rate can be achieved with a clever adjustment of step size.

If steps are too big, then too many will be rejected. To overcome this, step size can be shrunken slightly after each rejection.

Conversely, if steps are too small, too many are accepted. This can be overcome by making the step size bigger after each acceptance.

The precise procedure is to reduce step size by a factor 1.01 on each rejection but to increase by a factor 1.01<sup>3</sup> on each acceptance. The step size then tends toward a value producing 3 rejections for every acceptance. The power 3 can be adjusted to get a different acceptance rate, and the number 1.01 can be any number slightly above 1.

I have been trying to find a proof that step-size adjustment does not foil the goal of MCMC: reproducing the posterior distribution. But this problem can be avoided by running step-size adjustment during a burn-in of the MCMC, then maintaining a constant step size afterward.

### **Estimating the Mean and SD of a Normal Distribution: a Gibbs Sampler**

Fitting a normal distribution to observations requires estimating both mean  $\mu$  and standard deviation,  $\sigma$ . In this situation, the likelihood can be graphed as a function of  $\mu$  while  $\sigma$  is held constant, or vice versa. That is, for any given set of data  $x$ , set  $\mu$  to some fixed value (usually a reasonable value, perhaps the observed mean of  $x$ ). Then calculate the Gaussian probability of observing the data  $x$  for all possible values of  $\sigma$ . This is illustrated in *RcmdBayesWorkshop.r*.

Execute line 143 to 194 of *RcmdBayesWorkshop.r* to first create a simulated set of data (*weights*), then graph the likelihood first as a function of  $\mu$  (with  $\sigma$  held constant at 25), then as a function of  $\sigma$  (with  $\mu$  held constant at 125).

The Metropolis-Hastings algorithm can be used in this example, with two parameters, as long as one parameter is adjusted at a time. That is, start with any  $\mu$  and  $\sigma$ . Holding  $\sigma$  constant, make a Metropolis adjustment of  $\mu$ . Then holding  $\mu$  at this new value, make a Metropolis adjustment of  $\sigma$ . And so on and so forth. Working with one parameter at a time, while holding all others constant, is known as a Gibbs sampler. The value of this approach is that it allows models with large numbers of parameters to be fitted.

The function *normalposterior.Gibbs* in *RfuncBayesWorkshop.r* runs a Gibbs sampler to estimate  $\mu$  and  $\sigma$  for a series of observations that are normally distributed. It can be paused at every step to let the user observe details of the procedure.

Execute line 202 to 233 of *RcmdBayesWorkshop.r* to use the function *normalposterior.Gibbs* to run the Gibbs sampler with the simulated data *weights*. Then line 238 to 248 graph the likelihood surface as estimated by the sampler, showing the highest likelihood in dark blue, etc.

### **Gibbs Sampling and the General Likelihood Model**

In likelihood models, a single likelihood function defines how well a model's prediction describes the data. All parameters are involved in the same likelihood calculation. These basic models are the sort routinely amenable to an optimization approach, which is the method of maximum likelihood.

With large numbers of parameters, optimization can become difficult, though. A Gibbs sampler can handle large numbers of parameters easily, simply adjusting one at a time.

The approach is a straightforward extension of the example shown with the two-parameter normal distribution:

The log-likelihood function  $L(y|\theta) = \log(p(y|\theta))$ , where  $\theta = \theta_1, \theta_2, \dots, \theta_k$  are  $k$  parameters and  $y$  includes all the data.

Step 1: Assign all  $k$  parameters an initial value. Any value at which  $L$  can be calculated should work.

Step 2: The posterior for  $\theta_1$  is  $L(\theta_1|y, \theta_{-1})$ . That is, write  $L$  as a function of  $\theta_1$ , assuming all other parameters  $\theta_{-1}$  are held constant ( $\theta$  with subscript  $-1$  means all parameters except the first). Carry out a Metropolis step for  $\theta_1$ . This is exactly like the binomial example illustrated above, because only one parameter is varied. This produces a new value for  $\theta_1$ , the second in the MCMC chain. (Remember that the second value could be the same as the first, if the Metropolis step was rejected; also, step size adjustment is a good idea.)

Step 3: Identical to step 2, but for  $\theta_2$  instead of  $\theta_1$ . Write  $L$  as a function of  $\theta_2$ , assuming all other parameters ( $\theta_{-2}$ ) are held constant.  $\theta_1$  is fixed at its second value, while the rest are still at their first. The Metropolis step produces the second value in the MCMC chain of  $\theta_2$ .

Step 4 etc: Repeat the same procedure for each of the  $k$  parameters. The MCMC chain then includes two values for all parameters.

Then return to  $\theta_1$  and loop through all parameters a second time, and so on for 1000s of steps. The output chain for each parameter should be graphed to determine the burn-in period and to make sure the chain moves quickly up and down through its full range (as opposed to long sequences near one end or the other).

## **Special Priors**

In the example of estimating a posterior distribution for the binomial, it turned out that the posterior has a closed form solution: the beta distribution. This held for a flat prior (all parameters  $\theta$  equally likely), but it holds more generally too: if the prior of  $\theta$  is any beta distribution, then the posterior remains a beta.

Generally, one way of creating a posterior which has a precise formulation is to choose the prior correctly. This works if the prior multiplied by the likelihood function produces a form that can be integrated. One widely used example of this is the posterior of the standard deviation  $\sigma$  of the normal distribution. Above, we sampled this using Metropolis because a flat prior does not yield a closed form for the posterior. However, if the prior probability is  $p(\sigma) \propto \frac{1}{\sigma^2}$ , the posterior is an inverse-gamma distribution. The functions *Gibbs.normalmean* and *Gibbs.normalSD* in *RfuncBayesWorkshop.r* invoke this method for fitting a Gaussian. Lots of books and web pages show proofs and give the formulae (ie, <http://www.biostat.jhsph.edu/~fdominic/teaching/BM/3-4.pdf>).

## **Problems and Trouble-Shooting**

**Likelihood function.** The most common problem I have with Gibbs samplers, or any model based on likelihood, is getting the likelihood function right. Failure of the model to achieve reasonable results is usually due to errors in the likelihood. Sometimes the errors are obvious, but complicated likelihood functions can be difficult to debug. It is sometimes necessary to browse inside the function with a known set of parameters, and one trick is to check the likelihood contributed by every individual in a population for outliers. This can show, for instance, that one mismeasurement is controlling the summed likelihood.

**Failure to converge.** Models may still fail to converge, even with accurate likelihood functions. In my experience, the adjustment of step-size in the Gibbs sampler is very effective at demonstrating failure to converge, because the step-size will blow up and take parameter values with it (ie, reaching numbers like  $10^{40}$ ). Without adjusting step-size, it may take a very long run before it is evident that parameter values are climbing without bound. Failure to converge means the model must be reconsidered.

**Poorly mixed chain.** It is typically advised that chains of parameter values from a Gibbs sampler (or any Monte Carlo method) be examined graphically. Well-mixed means that successive parameter values are uncorrelated, so that the chain wanders up and down in a random way. Poorly mixed means that a parameter remains at one end of its range for a stretch, then gradually climbs toward the other end; the graph of the chain for one parameter may look sinusoidal. Step-size adjustment in the Metropolis algorithm is very effective at producing good mixing. But if parameters are correlated, it may fail. A poorly mixed chain may mean that parameter values were not explored widely enough, and this can produce erroneous results.

Parameter correlation. Another routine piece of advice with Gibbs samplers (or other MCMC) is to check pairwise graphs of the chains of all (or as many as practical) of the parameters. Correlation among pairs is common, but weak correlations are not much of a concern. Strong correlations, though, can prevent chains from mixing. As one common example, regression models often show correlation between intercept and slope parameters, but this can be avoided by redefining the intercept to refer to the mean of the independent variable(s). If other situations arise where parameters are strongly correlated, it would be wise to carefully consider what the parameters mean and whether the model can be written differently.