# Introduction to Markov Chain Monte Carlo

Jim Albert

May 28, 2014

## A Selected Data Problem

Here is an interesting problem with "selected data". Suppose you are measuring the speeds of cars driving on an interstate. You assume the speeds are normally distributed with mean $\mu$ and standard deviation $\sigma$. You see 10 cars pass by and you only record the minimum and maximum speeds. What have you learned about the normal parameters?

First we focus on the construction of the likelihood. Given values of the normal parameters, what is the probability of observing minimum $= x$ and the maximum $= y$ in a sample of size n?

Essentially we're looking for the joint density of two order statistics which is a standard result. Let $f$ and $F$denote the density and cdf of a normal density with mean $\mu$ and standard deviation $\sigma$. Then the joint density of $(x, y)$ is given by

$$f(x, y|\mu, \sigma) \propto f(x)f(y)[F(y) - F(x)]^{n-2}, x < y$$

After we observe data, the likelihood is this sampling density viewed as function of the parameters. Suppose we take a sample of size 10 and we observe $x = 52, y = 84$. Then the likelihood is given by

$$L(\mu, \sigma) \propto f(52)f(84)[F(84) - F(52)]^8$$

## Defining the log posterior

First I write a short function `minmaxpost` that computes the logarithm of the posterior density. The arguments to this function are $\theta = (\mu, \log \sigma)$ and data which is a list with components `n`, `min`, and `max`. I'd recommend using the R functions `pnorm` and `dnorm` in computing the density – it saves typing errors.

```
> minmaxpost <- function(theta, data){
+   mu <- theta[1]
+   sigma <- exp(theta[2])
+   dnorm(data$min, mu, sigma, log=TRUE) +
+     dnorm(data$max, mu, sigma, log=TRUE) +
```

```
+     (data$n - 2) * log(pnorm(data$max, mu, sigma) -
+     pnorm(data$min, mu, sigma))
+ }
```

## Normal approximation to posterior

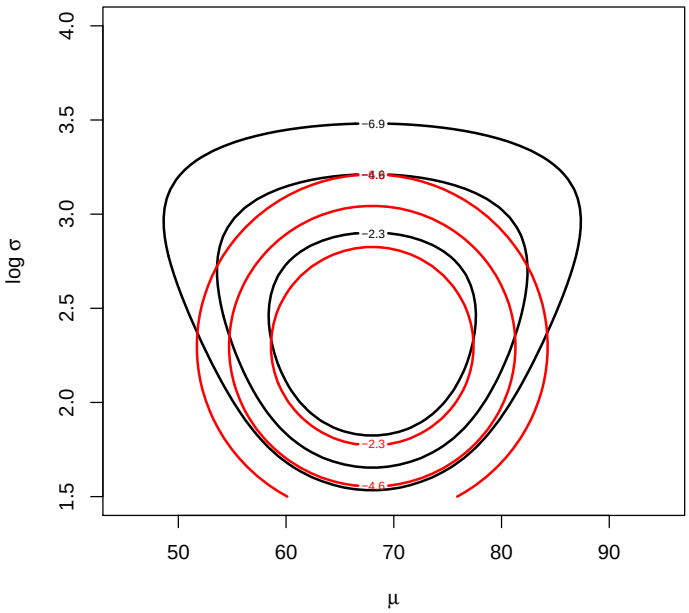
We work with the parameterization $(\mu, \log \sigma)$ which will give us a better normal approximation. A standard noninformative prior is uniform on $(\mu, \log \sigma)$.

The function `laplace` is used to summarize this posterior. The arguments to `laplace` are the name of the log posterior function, an initial estimate at $\theta$, and the data that is used in the log posterior function. The output of laplace includes mode, the posterior mode, and var, the corresponding estimate at the variance-covariance matrix.

```
> data <- list(n=10, min=52, max=84)
> library(LearnBayes)
> fit <- laplace(minmaxpost, c(70, 2), data)
> fit

$mode
[1] 67.999960  2.298369

$var
              [,1]          [,2]
[1,]  1.920690e+01 -1.901459e-06
[2,] -1.901459e-06  6.031533e-02

$int
[1] -8.020139

$converge
[1] TRUE
```

In this example, this gives a pretty good approximation in this situation. The `mycontour` function is used to display contours of the exact posterior and overlay the matching normal approximation using a second application of `mycontour`.

```
> mycontour(minmaxpost, c(45, 95, 1.5, 4), data,
+           xlab=expression(mu), ylab=expression(paste("log ",sigma)))
> mycontour(lbinorm, c(45, 95, 1.5, 4),
+           list(m=fit$mode, v=fit$var), add=TRUE, col="red")
```




## Random Walk Metropolis Sampling

The `rwmetrop` function implements the M-H random walk algorithm. There are four inputs: (1) the function defining the log posterior, (2) a list containing var, the estimated var-cov matrix, and scale, the M-H random walk scale constant, (3) the starting value in the Markov Chain simulation, (4) the number of iterations of the algorithm, and (5) any data and prior parameters used in the log posterior density.

Here we use `fit$v` as our estimated var-cov matrix, use a scale value of 3, start the simulation at $(\mu, \log \sigma) = (70, 2)$ and try 10,000 iterations.

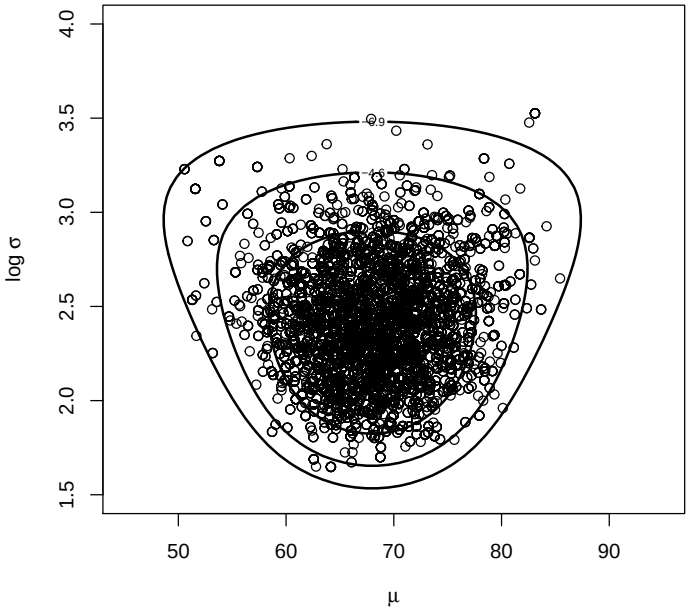
```
> mcmc.fit <-  rwmetrop(minmaxpost,
+              list(var=fit$v, scale=3),
+              c(70, 2),
+              10000,
+              data)
```

I display the acceptance rate – here it is 19% which is a reasonable value.

```
> mcmc.fit$accept

[1] 0.1889
```

We display the contours of the exact posterior and overlay the simulated draws.

```
> mycontour(minmaxpost, c(45, 95, 1.5, 4), data,
+           xlab=expression(mu),
+           ylab=expression(paste("log ",sigma)))
> points(mcmc.fit$par)
```




It appears like we have been successful in getting a good sample from this posterior distribution.

## Random Walk Metropolis Sampling

To illustrate simulation-based inference, suppose one is interested in learning about the upper quartile

$$P.75 = \mu + 0.674 \times \sigma$$

of the car speed distribution. For each simulated draw of $(\mu, \sigma)$ from the posterior, we compute the upper quartile $P.75$. We use the `density` function to construct a density estimate of the simulated sample of $P.75$.

```
> mu <- mcmc.fit$par[, 1]
> sigma <- exp(mcmc.fit$par[, 2])
```

```
> P.75 <- mu + 0.674 * sigma
> plot(density(P.75),
+      main="Posterior Density of Upper Quartile")
```