Chapter 9.4 Example:Cauchy/Normal Problem

Jim Albert and Monika Hu

Chapter 9 Simulation by Markov Chain Monte Carlo

Normal Sampling

- Suppose you are planning to move to Buffalo, New York and you are wondering about the snowfall in Buffalo in the following winter season.
- Collect data for the last 20 seasons in January. Assume that these observations of January snowfall are Normally distributed with mean μ and standard deviation σ.
- Assume that the sampling standard deviation σ is equal to the observed standard deviation s.
- The observed sample mean $\bar{y}$ and corresponding standard error are given by $\bar{y} = 26.785$ inches and se $= s/\sqrt{n} =$ 3.236

A Cauchy Prior

- Have prior beliefs about μ, the average snowfall during the month of January.
- After some reflection, you are 50 percent confident that μ falls between 8 and 12 inches.
- The 25th percentile of your prior for μ is 8 inches and the 75th percentile is 12 inches.
- Match this prior information with a Cauchy density with location parameter 10 and scale parameter 2.

The Posterior Density

- The posterior density of μ is proportional to

$$\pi(\mu \mid y) \propto \frac{1}{1 + \left(\frac{\mu - 10}{2}\right)^2} \times \exp\left\{-\frac{n}{2\sigma^2}(\bar{y} - \mu)^2\right\}.$$

- There are four inputs to this posterior:
- Data: the mean $\bar{y}$ and corresponding standard error $\sigma/\sqrt{n}$
- Prior; the location parameter 10 and the scale parameter 2 for the Cauchy prior.

Program the logarithm of the posterior

- Define a short function `lpost()` defining the logarithm of the posterior density

$$\log \pi(\mu \mid y) = -\log\left\{1 + \left(\frac{\mu - 10}{2}\right)^2\right\} - \frac{n}{2\sigma^2}(\bar{y} - \mu)^2.$$

```
lpost <- function(theta, s){
    dcauchy(theta, s$loc, s$scale, log = TRUE) +
    dnorm(s$ybar, theta, s$se, log = TRUE)
}
```

- A list named `s` is defined that contains these inputs for this particular problem.

```
s <- list(loc = 10, scale = 2,
          ybar = mean(data$JAN),
          se = sd(data$JAN) / sqrt(20))
```

Running the Metropolis Algorithm

- The Metropolis algorithm as coded in the function `metropolis()`.
- The inputs to this function are the log posterior function `lpost`, the starting value $\mu = 5$, the width of the proposal density $C = 20$, the number of iterations 10,000, and the list `s` that contains the inputs to the log posterior function.

```
out <- metropolis(lpost, 5, 20, 10000, s)
```

- The output variable `out` has two components: `S` is a vector of the simulated draws and `accept_rate` gives the acceptance rate of the algorithm.

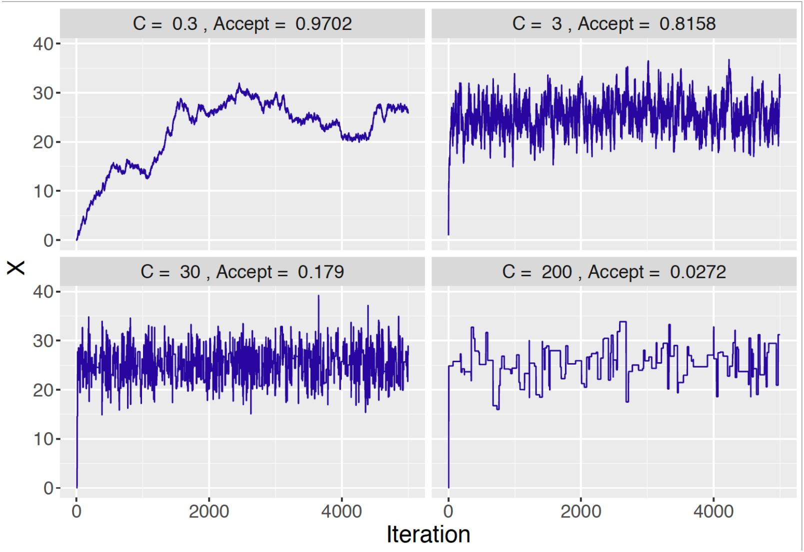
Choice of starting value and proposal region

- The user has to make two choices: a starting value and a value of C which determines the width of the proposal region.
- Choice of the starting value is usually not critical.
- Choice of the constant C is more critical.
- If C is chosen too small, then the simulated values tend to be strongly correlated and it takes a relatively long time to explore the entire probability distribution.
- If C is chosen too large, proposal values tend not to be accepted and the simulated values tend to get stuck at the current values.
- One monitors the choice of C by computing the acceptance rate, the proportion of proposal values that are accepted.

Try Different Choices of C

- Start with the value $\mu = 20$ and try the C values 0.3, 3, 30, and 200.
- Simulate 5000 values of the MCMC chain.
- Figure on the next slide shows in each case a line graph of the simulated draws against the iteration number and displays the acceptance rate.

Try Different Choices of C



Comments

- When one chooses a small value $C = 0.3$ (top-left panel), the graph of simulated draws has a snake-like appearance. The sampler does a relatively poor job of exploring the posterior distribution.
- Iff one uses a large value $C = 200$ (bottom-right panel), the flat-portions in the graph indicates there are many occurrences where the chain will not move from the current value.
- More moderate values of $C = 3$ and $C = 30$ (top-right and bottom-left panels) produce more acceptable streams of simulated values.
- In practice, it is recommended that the Metropolis algorithm has an acceptance rate between 20% and 40%.

Collecting the simulated draws

- Using MCMC diagnostic methods, one sees that the simulated draws are a reasonable approximation to the posterior density of μ.
- One displays the posterior density by computing a density estimate of the simulated sample.
- In Figure plots the prior, likelihood, and posterior density for the mean amount of Buffalo snowfall μ using the Cauchy prior.

Prior, Likelihood, Posterior Plot

- The posterior density resembles the likelihood. The posterior is most influenced by the data.

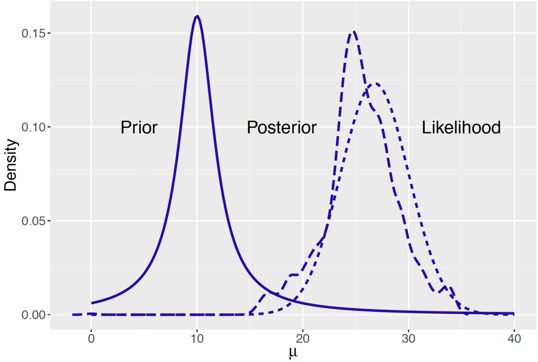


Figure 1: Prior, likelihood, and posterior of a Normal mean with a Cauchy prior.