Chapter 9.6 MCMC Inputs and Diagnostics

Jim Albert and Monika Hu

Chapter 9 Simulation by Markov Chain Monte Carlo

General Advice - Burn-In

- In theory, the Metropolis and Gibbs sampling algorithms will produce simulated draws that converge to the posterior distribution of interest.
- But in typical practice, it may take a number of iterations before the simulation values are close to the posterior distribution.
- So it is recommended that one run the algorithm for a number of "burn-in" iterations before one collects iterations for inference.

General Advice - Starting Values

- We have illustrated running a single “chain” where one has a single starting value
- It is possible that the MCMC sample will depend on the choice of starting value.
- So we recommend running the MCMC algorithm several times using different starting values.
- In this case, one will have multiple MCMC chains.
- By comparing the inferential summaries from the different chains one explores the sensitivity of the inference to the choice of starting value.

Diagnostics

- Output of a single chain from the Metropolis and Gibbs algorithms is a vector or matrix of simulated draws.
- Before one believes that a collection of simulated draws is a close approximation to the posterior distribution, some special diagnostic methods should be initially performed.

Trace plot

- A trace plot which is a line plot of the simulated draws of the parameter of interest graphed against the iteration number.
- Figure on the next slide displays a trace plot of the simulated draws of μ from the Metropolis algorithm for our Buffalo snowfall example for Normal sampling with a Cauchy prior.
- It is undesirable to have a snake-like appearance in the trace plot indicating a high acceptance rate.
- It is also undesirable to see a trace plot with many flat portions that indicates a sampler with a low acceptance rate.
- The trace plot on the next slide indicates the sampler is efficiently sampling from the posterior distribution.

Example Trace Plot for Snowfall Example

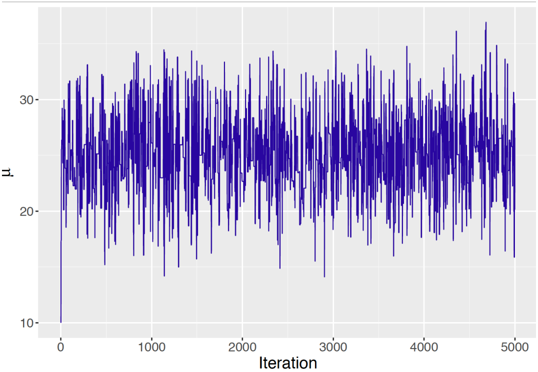

Figure 1: Trace plot of simulated draws of normal mean using the Metropolis algorithm with $C = 20$.

Autocorrelation plot

- One is concerned about the possible strong correlation between successive draws of the sampler.
- One visualizes this dependence by computing the correlation of the pairs $\{\theta^{(j)}, \theta^{(j+l)}\}$ and plotting this "lag-correlation" as a function of the lag value l.
- This autocorrelation plot of the simulated draws from our example is displayed on a Figure on the next slide.
- If there is a strong degree of autocorrelation, we see a large correlation even for large values of the lag value.
- Here the lag correlation values quickly drop to zero. This autocorrelation graph indicates the Metropolis algorithm is providing an efficient sampler of the posterior.

Autocorrelation Graph

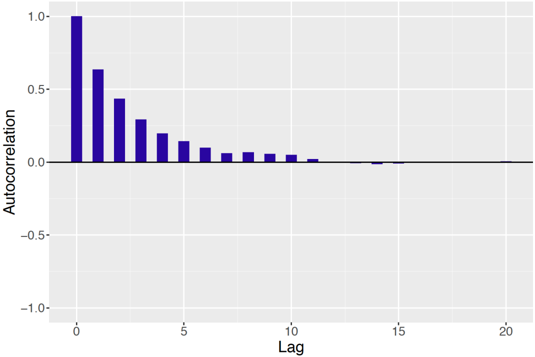

Figure 2: Autocorrelation plot of simulated draws of normal mean using the Metropolis algorithm with $C = 20$.

Graphs and summaries

- If the trace plot or autocorrelation plot indicate issues with the Metropolis sampler, the width of the proposal C should be adjusted and the algorithm run again.
- Since we believe that Metropolis with the use of the value $C = 20$ is suitable, then one uses a histogram of simulated draws, as displayed on the Figure on the next slide to represent the posterior distribution.
- Alternatively, a density estimate of the simulated draws can be used. Figure places a density estimate on top of the histogram of the simulated values of the parameter μ.

Graph of simulated draws from posterior

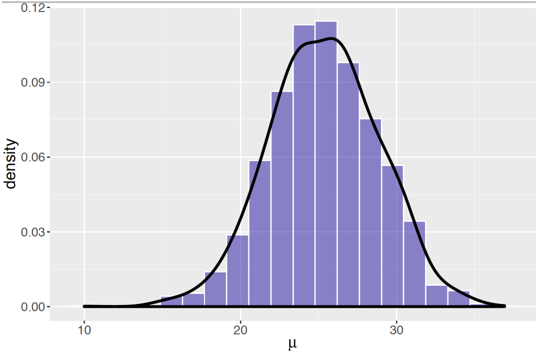

Figure 3: Histogram of simulated draws of the normal mean using the Metropolis algorithm with $C = 20$. The solid curve is a density estimate of the simulated values.

Posterior Summaries

- One estimates different summaries of the posterior distribution by computing different summaries of the simulated sample.
- In our Cauchy-Normal model, the posterior mean of μ is estimate by computing the mean of the simulated posterior draws:

$$E(\mu \mid y) \approx \frac{\sum_{j=1}^{S} \mu^{(j)}}{S}.$$

Monte Carlo Standard Error of Estimate

- If simulated draws are independent, a Monte Carlo standard error of this posterior mean estimate is given by the standard deviation of the draws divided by the square root of the simulation sample size:

$$se = \frac{sd(\{\mu^{(j)}\})}{\sqrt{S}}.$$

- However, this estimate of the standard error is not correct since the MCMC sample is not independent
- One obtains a more accurate estimate of Monte Carlo standard error by using time-series methods.
- This standard error estimate will be larger than the "naive" standard error estimate assuming independent MCMC sample values.