STA 250: Statistics

Notes 12. Stochastic Computing

Book chapters: 12.2, 12.4

1 Sampling and Monte Carlo

Laplace approximation is a wonderful mathematical tool, but it has several limitations. It fails to provide an adequate approximation when the target pdf is not symmetric around its mode, or has multiple modes. Worse still, we have no way to know how good the approximation is for a particular posterior pdf we are interested in. Moreover, the approximation does not work for non-regular statistical models.

Ever since the advent of computers, stochastic methods have become popular to approximate summaries of a target pdf. If we have a large number of draws $\theta^{(1)}, \cdots, \theta^{(M)}$ from a target pdf $h(\theta)$, then we can approximate the quantiles of $h(\theta)$ by the sample quantiles of the $\theta^{(j)}$'s. We can approximate the probability $h(\theta)$ assigns to any set A, i.e., $\int_A h(\theta)d\theta$ by $\#\{\theta^{(j)} \in A\}/M$, and so on. The approximations here all depend on laws of large number and apply to any pdf/pmf $h(\theta)$ (symmetric, irregular does not matter). Moreover, we can make the approximation arbitrarily better by working with a larger M, because the errors decrease at a rate $1/\sqrt{M}$ (due to the Central Limit Theorem). Such sample based approximations are known as Monte Carlo approximations.

While the laws of large number (and CLT error decay rate) apply directly when $\theta^{(j)}$'s are IID draws from $h(\theta)$ (resulting in IID Monte Carlo), they continue to hold under certain types of dependence between the sampled values. A very popular approach is to generate a sequence or a "chain" of draws that are serially dependent (a Markov chain) but still provides the desired approximation results. Such Monte Carlo approximations are known as Markov chain Monte Carlo approximations. We will see an IID Monte Carlo based on "rejection sampling" and a Markov chain Monte Carlo based on "Gibbs sampling".

2 Rejection sampling

Suppose we want to sample draws from a pdf $h(\theta) = e^{q(\theta)}/c_h$ where $q(\theta)$ is available in closed form, but we don't know the normalizing constant c_h. But suppose we have access to another pdf $g(\theta) = e^{u(\theta)}/c_g$ such that (a) we know how to sample from $g(\theta)$ and (b) $q(\theta) \le u(\theta)$ for all θ (the function $u(\theta)$ is called an envelop of $q(\theta)$). Rejection sampling uses the following strategy to convert draws from $g(\theta)$ into draws from $h(\theta)$:

1. Draw a $\theta \sim g(\theta)$ and set $p = \exp\{q(\theta) - u(\theta)\}$
2. Draw a $U \sim \mathsf{Uniform}(0, 1)$. If $U < p$ stop and output θ, otherwise go back to step 1.

Step 2 may result in a "rejection" and one may have to make several draws of θ from $g(\theta)$ before finding one that is accepted in step 2. Hence the name rejection sampling. Fewer rejections will be made if $u(\theta) - q(\theta)$ is small, i.e., the envelop is tight. Indeed, probability of rejection at any try is precisely:

$$\int g(\theta)P(U > e^{q(\theta)-u(\theta)})d\theta = \int g(\theta)\{1 - e^{q(\theta)-u(\theta)}\}d\theta = 1 - \frac{c_h}{c_g}$$

because $e^{q(\theta)-u(\theta)} = (c_h/c_g) \times h(\theta)/g(\theta)$. Let θ_{rej} denote the output of the rejection sampler. For any set A,

$$\begin{aligned} P(\theta_{\text{rej}} \in A) &= \sum_{k=0}^{\infty} P(\text{after } k \text{ rejections, I got a } \theta \in A \text{ and a } U < e^{q(\theta)-u(\theta)}) \\ &= \sum_{k=0}^{\infty} \left(1 - \frac{c_h}{c_g}\right)^k \int_A g(\theta)e^{q(\theta)-u(\theta)}d\theta = \sum_{k=0}^{\infty} \left(1 - \frac{c_h}{c_g}\right)^k \frac{c_h}{c_g} \int_A h(\theta)d\theta \\ &= \int_A h(\theta)d\theta \sum_{k=0}^{\infty} \left(1 - \frac{c_h}{c_g}\right)^k \frac{c_h}{c_g} = \int_A h(\theta)d\theta \end{aligned}$$

because the geometric series sums to one. But saying "$P(\theta_{\text{rej}} \in A) = \int_A h(\theta)d\theta$ for any arbitrary set A" is same as saying "$\theta_{\text{rej}} \sim h(\theta)$".

So rejection sampling makes a draw from the target pdf $h(\theta)$. To make several draws, we will have to make several (parallel) runs of the sampler and assemble the sampled values as IID draws from $h(\theta)$, and can do IID Monte Carlo approximation with them. We saw an example of this in Lab 6.

3 Gibbs sampling

Gibbs sampling is a technique of gathering samples from a multivariate pdf. It creates a chain of draws, that are serially correlated. However, the chain is built in such a way that laws of large numbers and CLT apply to the sampled values, and hence we can perform (Markov chain) Monte Carlo with them. For our discussion we will consider sampling from a bivariate pdf.

Suppose $h(\theta_1, \theta_2)$ is a bivariate pdf we want to sample from. For $(\theta_1, \theta_2) \sim h(\theta_1, \theta_2)$, denote the conditional pdf of θ_1 given θ_2 as $h_1(\theta_1|\theta_2)$ and denote the conditional pdf of θ_2 given θ_1 as $h_2(\theta_2|\theta_1)$. We assume that it is easy to sample from either conditional pdf.

Gibbs sampling from $h(\theta_1, \theta_2)$ proceeds as follows:

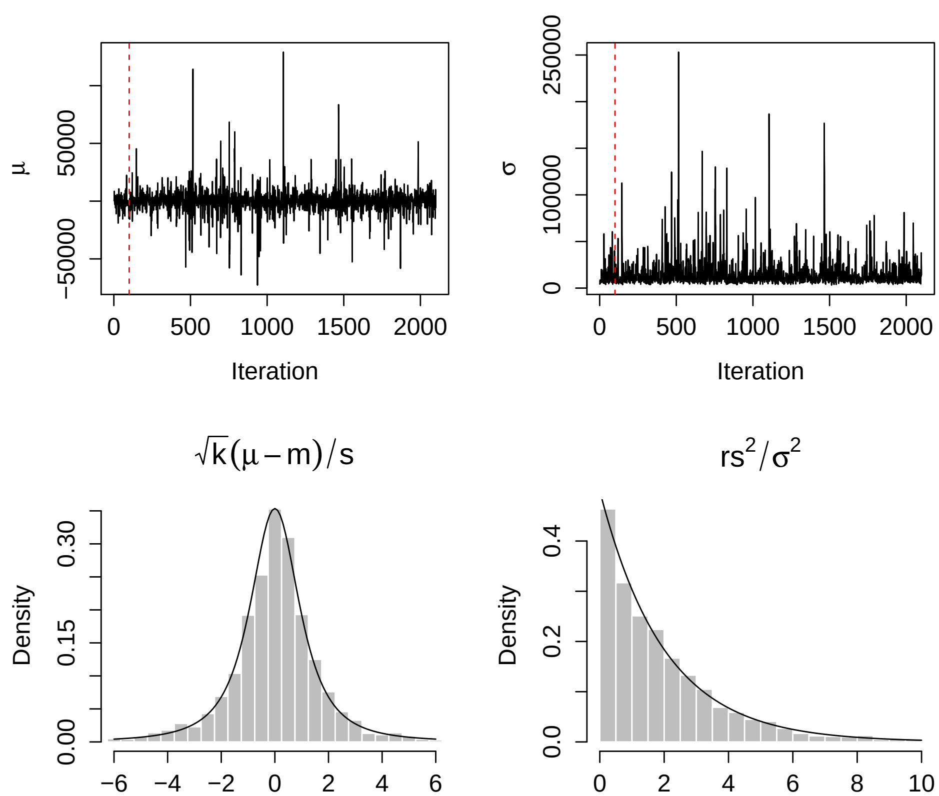
1. Start with initial values $\theta_1 = \theta_1^0, \theta_2 = \theta_2^0$.
2. For $t = 1, 2, \cdots$
   (a) Draw $\theta_1^t \sim h_1(\theta_1|\theta_2 = \theta_2^{t-1})$
   (b) Draw $\theta_2^t \sim h_2(\theta_2|\theta_1 = \theta_1^t)$
3. Discard some initial B many draws and present $(\theta_1^{B+j}, \theta_2^{B+j})$, $j = 1, \cdots, M$ as your sampled draws from $h(\theta_1, \theta_2)$.

Showing that LLN and CLT apply to this chain requires sophisticated tools from Markov chain theory. We will see this in action with a simple example.

Example (Sampling from $\mathsf{N}\chi^{-2}(m, k, r, s^2)$)**.** Let $h(\mu, \sigma^2) = \mathsf{N}\chi^{-2}(m, k, r, s^2)$ with $m = 0, k = 4, r = 2$ and $s = 9000$. Although we already know how to sample from the pdf, we will use Gibbs sampling to see how it performs for this target pdf. We already know that under this pdf, $\mu|\sigma^2 \sim \mathsf{Normal}(m, \sigma^2/k)$. Figuring out $\sigma^2|\mu$ requires some additional work. But it can be shown that the conditional pdf of $\{rs^2+k(\mu-m)^2\}/\sigma^2$ given μ equals $\chi^2(r+1)$. So our Gibbs sampler looks like:

1. Start with initial values $\mu = \mu^0, \sigma = \sigma^0$.
2. For $t = 1, 2, \cdots$
   (a) Draw $\mu^t \sim \mathsf{Normal}(m, (\sigma^{t-1})^2/k)$
   (b) Draw $u \sim \chi^2(r + 1)$ and set $\sigma^t = \sqrt{\{rs^2 + k(\mu^t - m)^2\}/u}$.
3. Discard some initial B many draws and present $(\mu^{B+j}, (\sigma^{B+j})^2)$, $j = 1, \cdots, M$ as your sampled draws from $\mathsf{N}\chi^{-2}(m, k, r, s^2)$.

The code below runs the Gibbs sampler, and shows the chains that were drawn. Then it compares the histograms of the sampled values of $\sqrt{k}(\mu - m)/s$ and rs^2/σ^2 against their theoretical pdfs $t(r)$ and $\chi^2(r)$.



```
m <- 0; k <- 4; r <- 2; s <- 9000

## run lengths and storage ##
n.discard <- 1e2; n.samp <- 2e3; n.iter <- n.discard + n.samp
mu.samp <- rep(NA, n.iter); sigma.samp <- rep(NA, n.iter)

## initialize and run Gibbs sampler ##
mu <- 0; sigma <- 5000
for(iter in 1:n.iter){
  mu <- rnorm(1, m, sigma / sqrt(k))
  u <- rgamma(1, (r + 1)/2, 1/2)
  sigma <- sqrt((r * s^2 + k * (mu - m)^2) / u)
  mu.samp[iter] <- mu; sigma.samp[iter] <- sigma
}

par(mfrow = c(2,2))
## plot chains ##
plot(mu.samp, ty = "l", ann = FALSE, ylab = expression(mu))
abline(v = n.discard, lty = 2, col = "red")
plot(sigma.samp, ty = "l", ann = FALSE, ylab = expression(sigma))
abline(v = n.discard, lty = 2, col = "red")

## discard and save ##
mu.samp <- mu.samp[-(1:n.discard)]
sigma.samp <- sigma.samp[-(1:n.discard)]

## compare sqrt(k) * (mu - m) / s against t(r)
hist(sqrt(k) * (mu.samp - m) / s, freq = FALSE, col = "gray", border = "white",
     breaks = c(-1e16, seq(-6.25,6.25,.5), 1e16), xlim = c(-6,6), ann = FALSE)
lines(seq(-6,6,.1), dt(seq(-6,6,.1), df = r))
title(main = expression(sqrt(k) * (mu - m) / s), ylab = "Density")

hist(r * s^2 / sigma.samp^2, freq = FALSE, col = "gray", border = "white",
     breaks = c(seq(0, 10, .5), 1e16), xlim = c(0,10), ann = FALSE)
lines(seq(0,10,.1), dgamma(seq(0,10,.1), r/2, 1/2))
title(main = expression(rs^2/sigma^2), ylab = "Density")
```