

# MTH 511a - 2020: Lecture 30

Instructor: Dootika Vats

*The instructor of this course owns the copyright of all the course materials. This lecture material was distributed only to the students attending the course MTH511a: “Statistical Simulation and Data Analysis” of IIT Kanpur, and should not be distributed in print or through electronic media without the consent of the instructor. Students can make their own copies of the course materials for their use.*

## 1 Metropolis-Hastings continued

In this video, we will continue discussing the Metropolis-Hastings algorithm via a few examples.

*Example 1* (Normal distribution). Suppose our target distribution is the standard normal distribution with density

$$\pi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \propto e^{-x^2/2}.$$

We do not need to know the proportionality constant, since that gets canceled out in the MH ratio. We want to first choose a proposal distribution.

Let’s choose proposal distribution be a a  $\text{Unif}[x - h, x + h]$  for some window  $h$ . So that

$$q(y|x) = \frac{1}{2h} I(x - h < y < x + h).$$

This is a symmetric proposal since  $q(x|y) = q(y|x)$  for particular values of  $x$  and  $y$ .

Next, we need to choose a starting value: we know that the center of a normal distribution is 0, so let’s start there and let’s set  $h = 1$ .

1. Set  $x_1 = 0$ .
2. Draw a proposed value  $y^* \sim U[x_t - 1, x_t + 1]$ .
3. Calculate

$$\alpha(x_t, y^*) = \min \left\{ 1, \frac{\pi(y^*)}{\pi(x_t)} \right\} = \min \left\{ 1, \exp \left\{ -\frac{y^{*2}}{2} - \frac{x_t^2}{2} \right\} \right\}$$

4. Draw  $U \sim U[0, 1]$ . If  $U < \alpha(x_t, y^*)$ . Set  $x_{t+1} = y^*$ .
5. Else, set  $x_{t+1} = x_t$ .
6. Stop when  $t = T$

Notice that we set the starting value to be 0 in the above example, since the mode/mean of the target distribution is 0. Thus, 0 is an area of high probability and a reasonable starting point for the Markov chain. Choosing starting values for Bayesian problems can be tricky, but the following can be helpful:

- Start from the MLE or the method of moments estimator of the parameter.
- If you believe the prior distribution is chosen well, then you can start from the prior distribution.

*Example 2* (Continuing  $t$  Bayesian example). The resulting Bayesian posterior distribution is

$$\pi(\mu|y) = c e^{-\frac{\mu^2}{2}} \prod_{i=1}^n \left( 1 + \frac{(y_i - \mu)^2}{\nu} \right)^{-\frac{\nu+1}{2}},$$

where the  $c$  is unknown.

We will use a Normal distribution proposal, centered at the current value. So  $q(y|x)$  will be the density of the distribution  $N(x, h)$  where  $h$  is similar to stepsize. Note that this is also a symmetric density.

We can choose a starting value for  $\mu$ .  $\mu$  is the mean of the  $t$  likelihood. So a good  $x_1 = n^{-1} \sum y_i$ . We set  $h = 2$  for now.

1. Set  $x_1 = n^{-1} \sum y_i$ .
2. Draw a proposed value  $y^* \sim N(x_t, 2)$ .
3. Calculate

$$\alpha(x_t, y^*) = \min \left\{ 1, \frac{\pi(y^*)}{\pi(x_t)} \right\} = \min \left\{ 1, \frac{e^{-\frac{y^{*2}}{2}} \prod_{i=1}^n \left( 1 + \frac{(y_i - y^*)^2}{\nu} \right)^{-\frac{\nu+1}{2}}}{e^{-\frac{x_t^2}{2}} \prod_{i=1}^n \left( 1 + \frac{(y_i - x_t)^2}{\nu} \right)^{-\frac{\nu+1}{2}}} \right\}$$

4. Draw  $U \sim U[0, 1]$ . If  $U < \alpha(x_t, y^*)$ . Set  $x_{t+1} = y^*$ .
5. Else, set  $x_{t+1} = x_t$ .
6. Stop when  $t = T$

```
#####
## MH samples from posterior distribtuon
# from t likelihood and normal prior
#####
set.seed(10)
log.post <- function(y, mu, nu)
{
  -mu^2/2 - (nu + 1)/2 *sum( log( 1 + (y-mu)^2/nu ) )
}

# Generate the data set.
# No need for small n and mu close to zero (the prior)
nu <- 3
n <- 50
mu <- 4
y <- mu + rt(n, df = nu) # Generate the data
```

Unlike accept-reject, the MCMC algorithm will not be limited by the size of the data. Thus, we can generate more data. Now we implement the Metropolis-Hastings algorithm above.

```
# Now compare to MCMC for the same target distribution
T <- 1e5
mc.samp <- numeric(length = T)
mc.samp[1] <- mean(y)
acc <- 0
foo <- proc.time()
for(t in 2:T)
{
  prop <- rnorm(1, mean = mc.samp[t-1], sd = .5)
  ratio <- log.post(y = y, mu = prop, nu = nu) - log.post(y = y, mu =
    mc.samp[t-1], nu = nu)

  if(runif(1) < exp(ratio))
  {
    mc.samp[t] <- prop
    acc <- acc + 1
  }else{
    mc.samp[t] <- mc.samp[t-1]
  }
}
proc.time() - foo
# user system elapsed
# 0.786 0.018 0.805
```

```

# Acceptance probability
# One dimensional problem, so the acceptance is good.
print(acc/T)
#[1] 0.4453

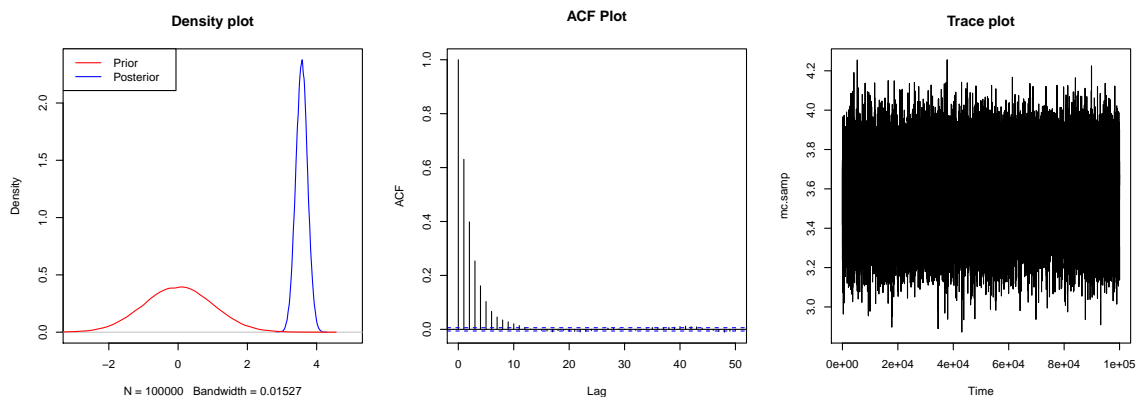
```

It is always a good idea to first look at some diagnostic plots, like the density plot, the acf plot, and the trace plot.

```

par(mfrow = c(1,3))
plot(density(mc.samp), col = "blue", xlim = c(-3,5), main = "Density plot")
lines(density(rnorm(1e5)), col = "red")
legend("topleft", col = c("red", "blue"), legend = c("Prior", "Posterior"),
      lty = 1)
acf(mc.samp, main = "ACF Plot")
plot.ts(mc.samp, main = "Trace plot")

```



From the plots above, we learn the following:

- The estimated posterior density plot is fairly smooth, indicating a reasonable quality of sampling. This makes sense since we generated  $T = 10^5$  samples, which is a large number of samples for this problem.
- Our acceptance probability was 44% which was well-chosen and the ACF plots indicate reasonably low autocorrelation. Thus our samples are of a reasonable quality.
- The trace plot looks like white noise with no discernible patterns. This is what we are looking for.

We conclude that the sampler performs reasonably well and note that it is fast as well.

```

# Since the sampler looks good, we can now
# estimate the posterior mean and quantiles
mean(mc.samp)

```

```
# [1] 3.568152
quantile(mc.samp, c(.025, .975))
#    2.5%    97.5%
#3.235619 3.902179
```

---

For the same problem, if we start from a bad starting value, the Markov chain will look first tend towards an area of high probability.

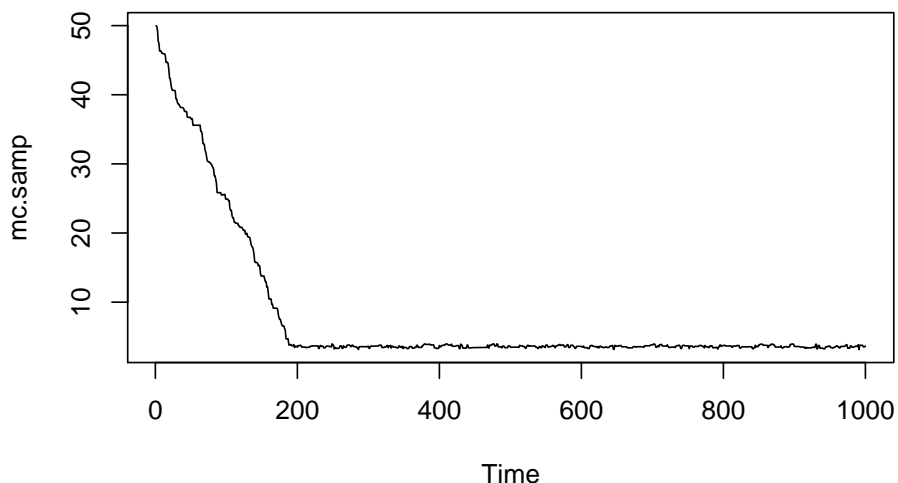
---

```
# a short run again to indicate what happens
# with a bad starting value
# But choose a BAD starting values
T <- 1e3
mc.samp <- numeric(length = T)
mc.samp[1] <- 50 ## bad starting value
acc <- 0
foo <- proc.time()
for(t in 2:T)
{
  prop <- rnorm(1, mean = mc.samp[t-1], sd = .5)
  ratio <- log.post(y = y, mu = prop, nu = nu) - log.post(y = y, mu =
    mc.samp[t-1], nu = nu)

  if(runif(1) < exp(ratio))
  {
    mc.samp[t] <- prop
    acc <- acc + 1
  }else{
    mc.samp[t] <- mc.samp[t-1]
  }
}
par(mfrow = c(1,1))
plot.ts(mc.samp, main = "Trace plot with bad starting value")
```

---

### Trace plot with bad starting value



There are multiple questions to still answer in MCMC that we will not be able to get to here. For example:

- Even though, we have non-iid samples, how do we still have a law of large numbers:

$$\frac{1}{T} \sum_{t=1}^T X_t \xrightarrow{a.s.} E_{\pi}[X].$$

- Does a Markov chain version of a central limit theorem hold? If yes, what is the variance in this CLT?
- Note that other posterior quantities pertaining to the target distribution can be found. For example, the variance of the target distribution can be estimated by

$$\frac{1}{T-1} \sum_{t=1}^T (X_t - \bar{X})^2 \xrightarrow{a.s.} \text{Var}_{\pi}(X).$$

Similarly target quantiles can be computed by sample quantiles.

- How many samples are needed in MCMC. That is, what is  $T$ ? For now, you should obtain enough samples so that the marginal density plots are relatively smooth.
- Tuning the MCMC sampler can take a lot of work. There are some other rules, guidelines, that are outside the scope of the course.
- Aside from Metropolis-Hastings algorithm, there are multiple other MCMC algorithms that we will not get to discuss here.

## 2 Questions to think about

- Implement the MCMC sampler for the normal target distribution example. Change proposal variance  $h$  to see how the sampler changes.
- How would you choose the proposal distribution when the target is higher-dimensional?
- When starting from bad starting values, like the in the plot above, what do you think we should do having observed that trace plot?