Exploring an Independence Sampler

Dipika Gawande | 12 November 2021

A simple special case of Metropolis-Hastings sampling is called the independence sampler, where proposals are made according to a fixed pmf or pdf independently of the current state. That is, we choose a pmf or pdf g and propose candidates according to $q(x, y) = g(y)$, so that the probability (or density) of proposing y is just $g(y)$ and does not depend on the current state x.

(Part a)

Show that for independence sampling with proposal density g, the Metropolis-Hastings acceptance probability takes the form $A(x, y) = min\left\{1, \frac{f(y)g(x)}{f(x)g(y)}\right\}$.

Ans: In general, the Metropolis-Hastings acceptance probability is calculated as $A(x, y) = min\left\{1, \frac{f(y)Q(y,x)}{f(x)Q(x,y)}\right\}$, where:

- $x =$ current state
- $y =$ proposed next state
- $f =$ the desired probability density (from which we want to sample)
- $Q =$ the proposal density. Therefore, $Q(x, y)$ is the transition probability from x to y using the proposal distribution.

In this case, the proposal density Q doesn't depend on x (the current state) at all. We are told that:

- $Q(x, y) = g(y)$ <- depends only on proposed state. Here x is the current state and y is the proposed state. Therefore this implies that:
- $Q(y, x) = g(x)$ <- depends only on proposed state. Here y is the current state and x is the proposed state.

Therefore, subbing these quantities into the general Metropolis-Hastings acceptance equation from above gives

$A(x, y) = min\left\{1, \frac{f(y)g(x)}{f(x)g(y)}\right\}$. The Hastings correction is altered by our substitution.

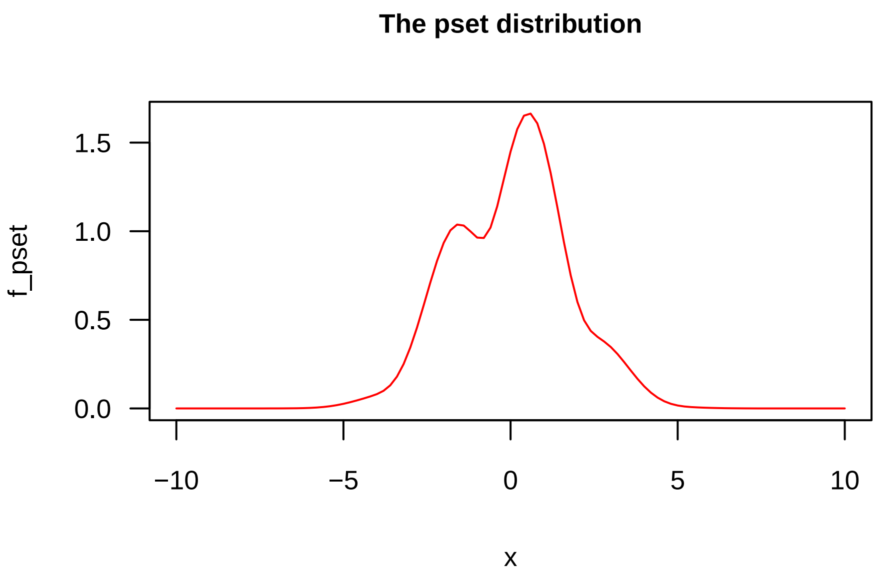
QED.

(Part b)

Say we have a density with parameters (7, 2021) given by:

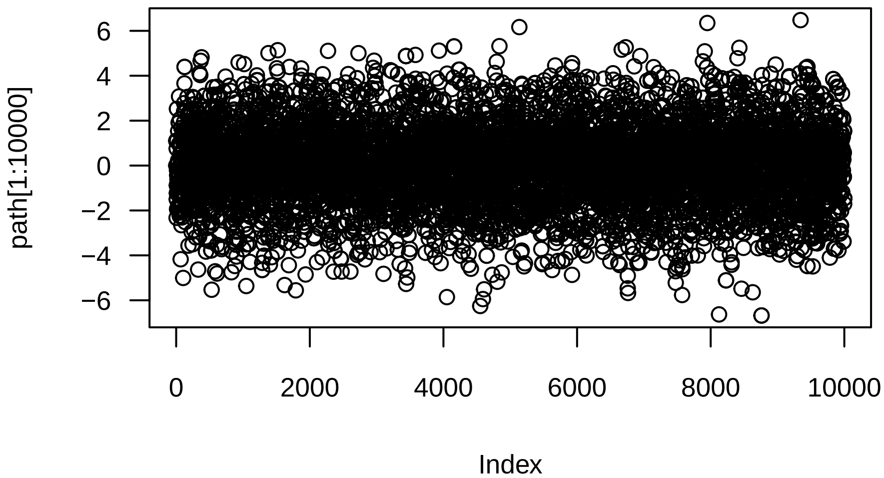
$f(x) \propto \sqrt{2.1 + sin(2.1x)}e^{\frac{-|x|^{2.1}}{7}}$ for $-10 < x < 10$ and and $f(x) = 0$ for $|x| > 10$. Generate an MCMC sample of size 100,000 from f by using an independence sampler with proposals generated from the Cauchy density. Draw a histogram (on density scale, not frequency) and overlay the density curve f.

```r
f_pset <- function(x){
  sqrt(2.1 + sin(2.1*x)) * exp((-1/7) * (abs(x)^2.1)) * (x > -10) * (x < 10)
}

plot(f_pset, -10, 10,
     col = "red", las = 1,
     main = "The pset distribution",
     cex.main = 0.8, cex.axis = 0.8, cex.lab = 0.8)
```

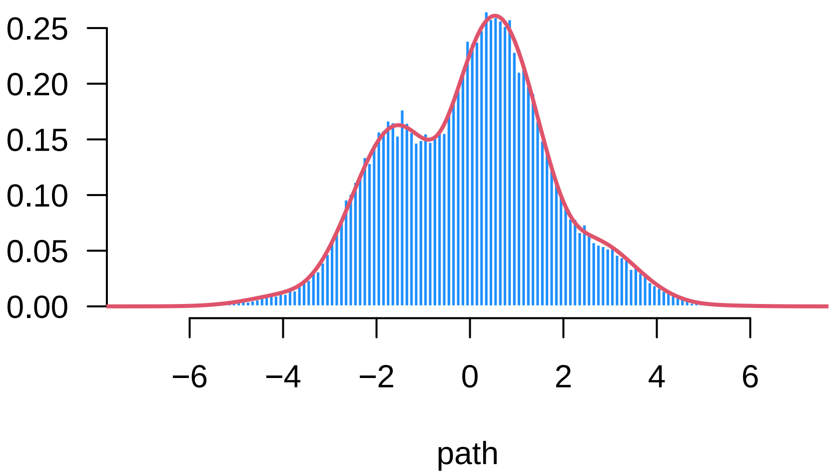



```r
set.seed(1)
f <- f_pset

nit <- 100000
path <- rep(0, nit)

state <- 0
path[1] <- state

for(i in 2:nit){
  #angle <- runif(1, -pi/2, pi/2)
  #candidate <- tan(angle)
  candidate <- rcauchy(n = 1)
  ratio <- (f(candidate) * dcauchy(state)) / (f(state) * dcauchy(candidate))
  u <- runif(n = 1, min = 0, max = 1)
  if(u < ratio){state <- candidate}
  path[i] <- state
}
#path[1:100]
```

```r
plot(path[1:10000],
     las = 1, cex.main = 0.8, cex.axis = 0.8, cex.lab = 0.8)
```




```r
library(MASS)
truehist(path,
         col = "dodgerblue", border = "white", las = 1)
con <- integrate(f_pset, -10, 10)$value
fpset_normalized <- function(x){f_pset(x)/con}
plot(fpset_normalized, -10, 10, n=1000, add = T, lwd=2, col=2)
```



(Part c)

Use your sample to approximate the probability $P\{X < 2.38\}$ where $X \sim f$

```r
mean(path < 2.38)
```

```
## [1] 0.91631
```