Lab 5: Rejection Sampling

Rebecca C. Steorts

January 2019

We can often end up with posterior distributions that we only know up to a normalizing constant. For example, in practice, we may derive

$$p(\theta \mid x) \propto p(x \mid \theta)p(\theta)$$

and find that the normalizing constant $p(x)$ is very difficult to evaluate. Such examples occur when we start building non-conjugate models in Bayesian statistics.

Given such a posterior, how can we appropriate it's density? One way is using rejection sampling. As an example, let's suppose our resulting posterior distribution is

$$f(x) \propto sin^2(\pi x), x \in [0, 1].$$

```r
set.seed(1)
```

In order to understand how to approximate the density (normalized) of f, we will investigate the following tasks:

Task 1

Plot the densities of $f(x)$ and the Unif(0,1) on the same plot.

Let's first create a sequence of points from 0 to 1, so that we can have a grid of points for plotting both of the proposed functions.

```r
# grid of points
x <- seq(0, 1, 10^-2)
```

```r
fx <- function(x) sin(pi * x)^2
plot(fx, xlim = c(0,1), ylim = c(0,1.5), ylab = "f(x)", lwd = 2)
curve(dunif, add = TRUE, col = "blue", lwd = 2)
curve(dbeta(x,2,2), add = TRUE, col = "red", lwd = 2)
legend("bottom", legend = c(expression(paste("sin(",pi,"x)"^"2")),"Unif(0,1)",
"Beta(2,2)"), col = c("black", "blue", "red"), lty = c(1,1,1), bty = "n", cex = 1.1, lwd = 2)
```

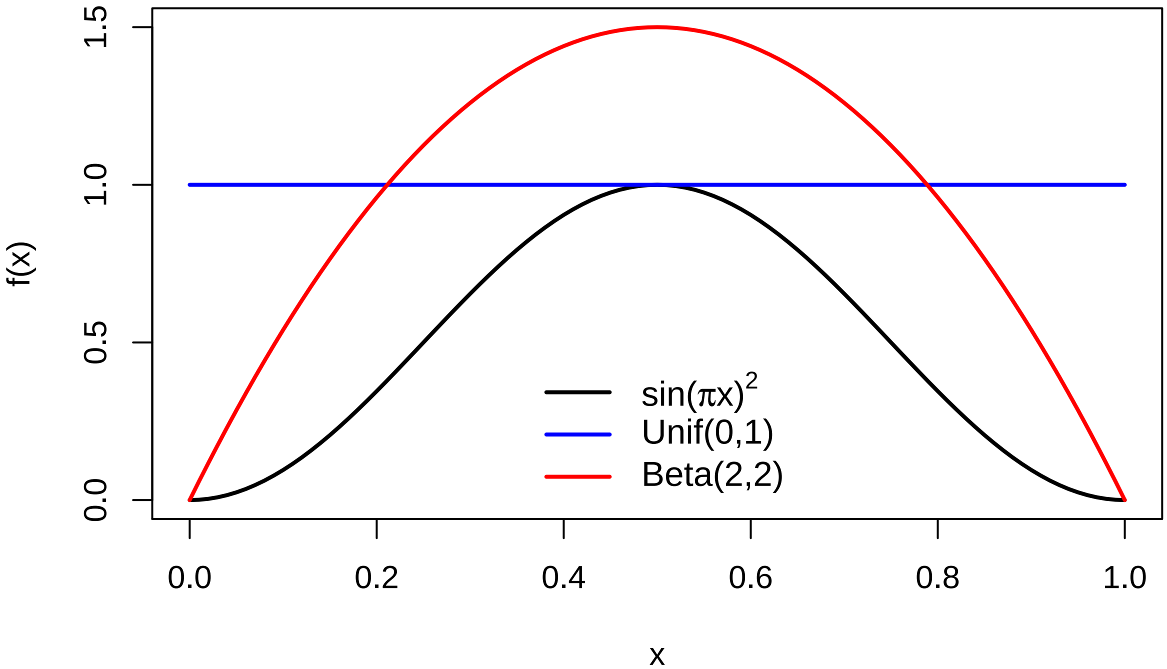

Figure 1: Comparision of the target function and the Unif(0,1) and the Beta(2,2) densities on the same plot.

Tasks 2 – 4

According to the rejection sampling approach sample from f(x) using the Unif(0,1) pdf as an enveloping function. In order to do this, we write a general rejection sampling function that also allows us to plot the historams for any simulation size. Finally, our function also allows us to look at task 4 quite easily.

```
sim_fun <- function(f, envelope = "unif", par1 = 0, par2 = 1, n = 10^2, plot = TRUE){

  r_envelope <- match.fun(paste0("r", envelope))
  d_envelope <- match.fun(paste0("d", envelope))
  proposal <- r_envelope(n, par1, par2)
  density_ratio <- f(proposal) / d_envelope(proposal, par1, par2)
  samples <- proposal[runif(n) < density_ratio]
  acceptance_ratio <- length(samples) / n
  if (plot) {
    hist(samples, probability = TRUE,
         main = paste0("Histogram of ",
                       n, " samples from ",
                       envelope, "(", par1, ",", par2,
                       ").\n Acceptance ratio: ",
                       round(acceptance_ratio,2)),
                       cex.main = 0.75)
  }
  list(x = samples, acceptance_ratio = acceptance_ratio)
}
```

```
par(mfrow = c(2,2), mar = rep(4, 4))
unif_1 <- sim_fun(fx, envelope = "unif", par1 = 0, par2 = 1, n = 10^2)
unif_2 <- sim_fun(fx, envelope = "unif", par1 = 0, par2 = 1, n = 10^5)
# ATTN: You will need to add in the Beta(2,2) densities on your own to finish task 4.
```

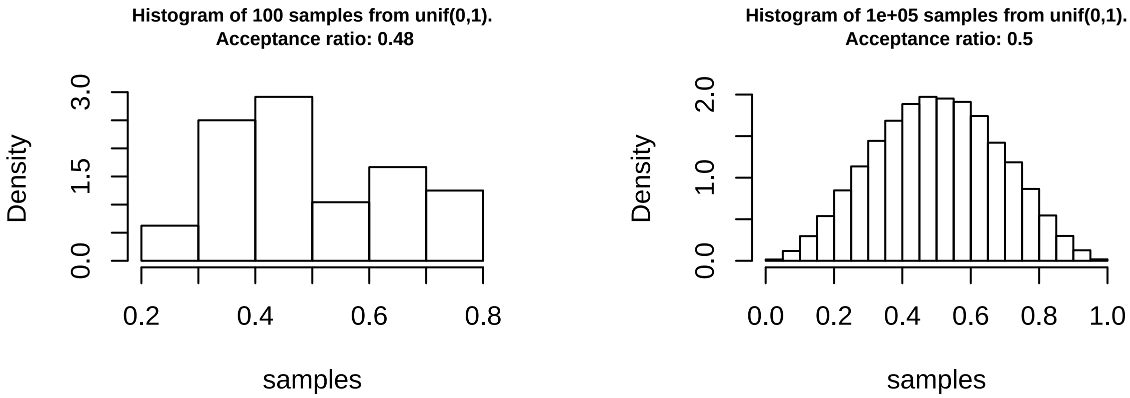

Figure 2: Comparision of the output of the rejection sampling for 100 versus 100,000 simulations with uniform and beta distributions as envelope functions.

```r
par(mfrow = c(1,1))
```