Lab 6 Solutions, STA 360/602

Rebecca C. Steorts

1. Naive Monte Carlo

Actual Value For the sake of comparison, we can derive the true value using substitution and the gamma function. We will use the substitution $u = x^4$ to derive the result:

$$\begin{aligned}
\int_{-\infty}^{\infty} \exp(-x^{-4})\,dx &= 2\int_{0}^{\infty} \exp(-x^{-4})\,dx \\
&= 2\int_{0}^{\infty} \frac{\exp(-u)}{4u^{3/4}}\,du \\
&= 2^{-1/2}\int_{0}^{\infty} u^{1/4-1}e^{-u}\,du \\
&= \frac{\Gamma(1/4)}{2} \\
&= 1.813.
\end{aligned}$$

Three Monte Carlo simulations designed to approximate this value are explained below.

Evaluation using substitution Let $y = \sqrt{2}x^2$. We will perform u-substitution to evaluate the integral with Monte Carlo:

$$\begin{aligned}
I &= \int_{-\infty}^{\infty} \exp(-x^4)\,dx \\
&= 2^{-1/2}\int_{0}^{\infty} \frac{2^{1/4}\exp(-y^2/2)}{\sqrt{y}}\,dy \\
&= 2^{-1/4}\int_{0}^{\infty} \frac{\sqrt{2\pi}\exp(-y^2/2)}{\sqrt{2\pi}\sqrt{y}}\,dy \\
&= 2^{-1/4}\int_{0}^{\infty} \sqrt{\frac{2\pi}{y}}\phi(y)\,dy \\
&= 2^{-5/4}\int_{0}^{\infty} \sqrt{\frac{2\pi}{y}}2\phi(y)\,dy.
\end{aligned}$$

The function $2\phi(y)$ is the density of the normal distribution truncated (or folded) at zero. We can sample from this distribution by taking samples from the standard normal distribution and then taking their absolute value. Note that if $X \sim N(0,1)$ we see for any $c > 0$

$$\begin{aligned}
P(|X| < c) &= P(-c < X < c) \\
&= 2(\Phi(c) - 1/2) \\
&= 2\Phi(c) - 1,
\end{aligned}$$

the derivative of which is the pdf we are sampling from.

Inducing the Normal Expectation We can multiply and divide the integral by a density that has a support equal to the area over which we are integrating. An obvious choice is the standard normal density, ϕ:

$$\int_{-\infty}^{\infty} \exp(-x^4)\, dx = \int_{-\infty}^{\infty} \frac{\exp(-x^4)}{\phi(x)} \phi(x)\, dx.$$

We can therefore evaluate the integral by sampling from a standard normal and averaging the values evaluated in $\exp(-x^4)/\phi(x)$.

Truncating the Integral Note that in `R`, we find $\exp(-(10)^4) = 0$. This means that the function decreases so quickly that it is essentially zero by the time $x = 10$. We can reasonably assume, therefore, that truncating the tails of the integral will not significantly affect the value. In other words, we have

$$\begin{aligned} \int_{-\infty}^{\infty} \exp(-x^4)\, dx \approx \int_{-10}^{10} \exp(-x^4)\, dx \\ = \int_{-10}^{10} \frac{20 \exp(-x^4)}{20}\, dx, \end{aligned}$$

which represents the expecation of $20 \exp(-x^4)$ over the uniform distribution on $[-10, 10]$.

Comparison of Methods The results of 10,000 simulations using the three methods described above are summarized in Table 1; the true value is included for comparison. Of these methods, multiplying and dividing by the standard normal density and then sampling from this density seems to yield the best estimate, which is both closer to the true value and has lower standard error. The other two methods are comparable in both their estimates and standard errors.

	Mean	SE
True value	1.81	
Truncated Normal	1.79	0.06
Full Normal	1.81	0.01
Truncated Uniform	1.78	0.05

Table 1: Comparison of the Monte Carlo estimate for the value of the integral using various methods with 10,000 draws for each method.

```r
knitr::opts_chunk$set(cache=TRUE)
library(xtable)
# define target functions for MC estimate
mc1 <- function(x){
  # folded normal target
  return(sqrt(2*pi/abs(x)))
}

mc2 <- function(x){
  # normal target
  exp(-x^4)/dnorm(x)
}

mc3 <- function(x){
  # uniform target
  return(20*exp(-x^4))
}
mcSE <- function(values){
  # this function calculates the standard error
  # of an MC estimate with target function values
  # eg values should be vector of f(x_i)
  return(sqrt(var(values)/length(values)))
}
# folded normal
draws <- rnorm(10000)
values <- sapply(draws,mc1)
# normal
draws2 <- rnorm(10000)
values2 <- sapply(draws2,mc2)
# uniform
draws3 <- runif(10000,min=-10,max=10)
values3 <- sapply(draws3,mc3)
# plot histograms
pdf(file="mc1hist.pdf")
hist(values*2^(-5/4), xlab = expression(f(X)),
     main = "Folded Normal")
dev.off()
```

```
## pdf
##   2
```

```
pdf(file="mc2hist.pdf")
hist(values2, xlab = expression(f(X)),
     main = "Normal")
dev.off()
```

```
## pdf
##   2
```

```
pdf(file="mc3hist.pdf")
hist(values3, xlab = expression(f(X)),
     main = "Uniform")
dev.off()
```

```
## pdf
##   2
```

```
# put results in a data frame and display with xtable
mc.mean <- mean(values)*2^(-5/4)
mc.sd<- 2^(-10/4)*sd(values)/sqrt(length(values))
means <- c(gamma(1/4)/2,mc.mean,mean(values2),mean(values3))
SEs <- c(NA, mcSE(values),mcSE(values2),mcSE(values3))
mc.df <- data.frame(Mean = means, SE = SEs,
                    row.names = c("True value", "row.names
                                  Normal", "Full Normal",
                                  "Truncated Uniform"))
xtable(mc.df)
```

```
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Fri Feb 23 11:04:59 2018
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrr}
##   \hline
##  & Mean & SE \\
##   \hline
## True value & 1.81 &  \\
##   row.names
##                                    Normal & 1.73 & 0.04 \\
##   Full Normal & 1.83 & 0.01 \\
##   Truncated Uniform & 1.89 & 0.05 \\
##    \hline
## \end{tabular}
## \end{table}
```

The histograms of the samples are give in Figure 1.

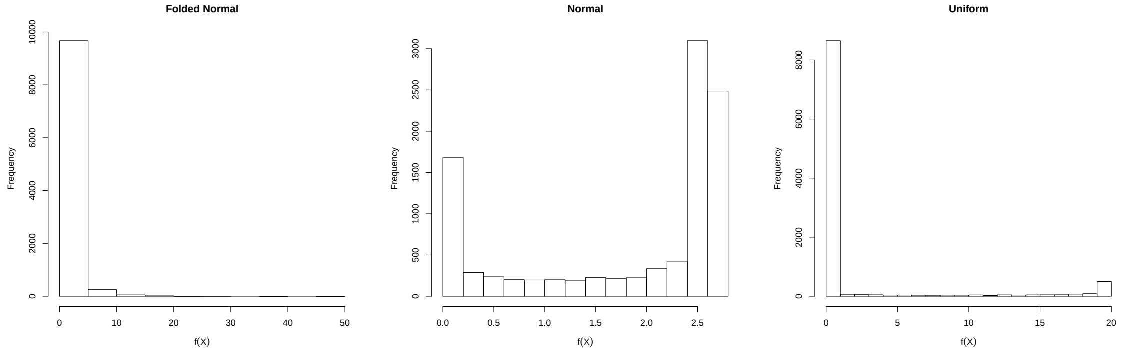

Figure 1:   Histograms for the various Monte Carlo simulations.

An ideal histogram would be highly centered around the true value of 1.81. For the folded normal, we see that there are some very large observations that skew the distribution. The normal method also results in a strange histogram, with values concentrated near the edges. The uniform has many values close to zero and a large number near 20. The values near 20 result from draws that are close to zero ($20e^0 = 20$). None of these histograms suggest that the standard error will be low. To get a lower standard error, we could use a method like importance sampling.