Homework 8

Louis Bensard

April 18, 2018

Problem 1:

(a)

Let $g(x) = \frac{e^{-x}}{1+x^2}$ and $X_1, \ldots, X_n \sim unif(0,1)$. Consider the control function $h(x) = \frac{e^{-1/2}}{1+x^2}$. Clearly $h(X_1)$ and $g(X_1)$ are correlated , thus we can estimate θ by:

$$\hat{\theta}_{c^*} = \frac{1}{n}\sum_{i=1}^{n} g(X_i) + c^* \cdot (h(X_i) - \mu)$$

With $\int_0^1 h(x)dx = \mu = \frac{\pi e^{-1/2}}{4}$ and $c^* = \frac{-cov(g(X_1),h(X_1))}{var(h(X_1))}$.

```r
g <- function(x){

    return(exp(-x)/(1+x^2))
}

h <- function(x){

    return(exp(-0.5)/(1+x^2))
}

n= 20000
X = runif(n,0,1)

c_star = -(cov(g(X), h(X)))/var(h(X))
mu = exp(-0.5)*(pi/4)

theta_hat_c = (1/n)*sum(g(X) + c_star*(h(X)-mu))
print(theta_hat_c)
```

```
## [1] 0.525292
```

Thus $\hat{\theta}_{c^*} = 0.525292$.

(b)

The % reduction variance by using $\hat{\theta}_{c^*}$ in place of $\hat{\theta}_{MC}$ gives us:

$$\frac{var(\hat{\theta}_{c^*}) - var(\hat{\theta}_{MC})}{var(\hat{\theta}_{MC})} = -corr(h(X_1), g(X_1))^2$$

Therefore, we get:

$$ste(\hat{\theta}_{c^*}) = ste(\hat{\theta}_{MC}) \cdot \sqrt{1 - corr(h(X_1), g(X_1))^2}$$

```
rho = cor(g(X), h(X))

g_bar_MC = (1/n)*sum(g(X))
ste_MC = sqrt((1/(n*(n-1)))*sum((g(X)-g_bar_MC)^2))

ste_c = ste_MC*sqrt(1-rho^2)
print(ste_c)
```

```
## [1] 0.00039698
```

Thus, $ste(\hat{\theta}_{c^*}) = 3.9698 \times 10^{-4}$.

(c)

```
ste_c_sample = sd(g(X) + c_star*(h(X)-mu))/sqrt(n)
print(ste_c_sample)
```

```
## [1] 0.00039698
```

The standard error of $\hat{\theta}_{c^*}$ obtained from the 20000 values we simulated is exactly the standard error obtained in b). Therefore, I am guessing that R must be doing the same thing as we do and is sampling from $unif(0, 1)$ by default.

(d)

Let $f(x) = 2(1-x)$, $(0 \leq x \leq 1)$, we are now drawing $X_1, \ldots, X_n$ from $f(x)$. Thus, we have:

$$\theta = \int_0^1 \frac{g(x)}{f(x)} f(x) dx$$

Thus, the control variate estimate of θ is now:

$$\hat{\theta}_{c^*} = \frac{1}{n} \sum_{i=1}^{n} \frac{g(X_i)}{f(X_i)} + c^* \cdot (\frac{h(X_i)}{f(X_i)} - \mu)$$

With

$$\mu = \frac{\pi e^{-1/2}}{4} \text{ and } c^* = \frac{-cov(\frac{g(X_1)}{f(X_1)}, \frac{h(X_1)}{f(X_1)})}{var(\frac{h(X_1)}{f(X_1)})}$$

.

Sampling the X's:

Note that the cdf of $X \sim f(x)$ is $F(x) = x(2-x)$, $(0 \leq x \leq 1)$, thus we have $F^{-1}(x) = 1 - \sqrt{1-x}$, $(0 \leq x \leq 1)$. Therefore, we sample the $X_i's$ the following way.
Let $U_1, \ldots, U_n \sim unif(0, 1)$, then $F^{-1}(U_i) \sim X$.

```
f <- function(x){

    return(2*(1-x))
}
```

```
n= 20000
U = runif(n,0,1)
X = 1 - sqrt(1-U)
c_star2 = -(cov(g(X)/f(X), h(X)/f(X)))/var(h(X)/f(X))
mu = exp(-0.5)*(pi/4)

theta_hat_c2 = (1/n)*sum(g(X)/f(X) + c_star2*(h(X)/f(X)-mu))
print(theta_hat_c2)
```

```
## [1] 0.524992
```

Thus $\hat{\theta}_{c^*} = 0.524992$.

(e)

```
rho2 = cor(g(X)/f(X), h(X)/f(X))

g_bar_MC2 = (1/n)*sum(g(X)/f(X))
ste_MC2 = sqrt((1/(n*(n-1)))*sum(((g(X)/f(X))-g_bar_MC2)^2))

ste_c2 = ste_MC2*sqrt(1-rho2^2)
print(ste_c2)
```

```
## [1] 0.000369893
```

Thus, $ste(\hat{\theta}_{c^*}) = 3.698932 \times 10^{-4}$. The standad error here is a little lower than the standard error we obtained in part c), so the different sampling does decreases variance.

Problem 2:

(a)

Let $f(x) = \frac{e^{-x}}{1-e^{-1}}$, $(0 \le x \le 1)$, initially we are drawing $X_1, \ldots, X_n$ from $f(x)$. Thus, we have:

$$\theta = \int_0^1 h(x)f(x)dx = \int_0^1 \frac{e^{-x}}{1+x^2}dx, \text{ with } h(x) = \frac{1-e^{-1}}{1+x^2}$$

Now assume we are drawing the $X's$ from $g(x) = \frac{1}{\pi(1+x^2)}$, $-\infty < x < \infty$ (Cauchy(0,1) distribution). So now, we can estimate θ by:

$$\hat{\theta^*}_{IS} = \frac{1}{n}\sum \frac{h(X_i) \cdot f(X_i)}{g(X_i)}$$

But note that g has support over all $\mathbb{R}$ while we are only integrating over $[0,1]$. Therefore, we need to weight the X values generated that are not in $[0,1]$ so that if $X_i \notin [0,1]$, then $\frac{h(X_i) \cdot f(X_i)}{g(X_i)} = 0$.

To do so, I decided to set every X_i outside of that interval equal to 100, therefore $\frac{h(X_i) \cdot f(X_i)}{g(X_i)}$ gets to 0 (because of the e^{-x}).

```r
g <- function(x){

    return(1/(pi*(1+x^2)))
}

f <- function(x){

    return(exp(-x)/(1-exp(-1)))
}

h <- function(x){

    return((1-exp(-1))/(1+x^2))
}

n= 20000
X = rcauchy(n,0,1)
X[(X<=-0.00001)] = 100; X[(X>=1.00001)] = 100

theta_hat_IS = (1/n)*sum(h(X)*f(X)/g(X))
print(theta_hat_IS)
```

```
## [1] 0.532704
```

```r
#standard error
hfg_bar = theta_hat_IS

ste_IS = (1/(n*(n-1)))*sum(((h(X)*f(X)/g(X)) - hfg_bar)^2)
print(ste_IS)
```

```
## [1] 4.59533e-05
```

Thus $\hat{\theta^*}_{IS} = 0.532704$ and $ste(\hat{\theta^*}_{IS}) = 4.595326 \times 10^{-5}$

(b)

Now, we are going to repeat the same process as a) but with $g(x) = \frac{4}{\pi(1+x^2)}$, $(0 \leq x \leq 1)$. So instead of adding weights, we are simply going to remove the $X_i's$ that are not in $[0, 1]$ after generating them from $Cauchy(0, 1)$.

```r
g <- function(x){

    return(4/(pi*(1+x^2)))
}



n = 20000
X = rcauchy(n,0,1)
Y = X[which((X>=0) & (X<=1))]

n_y = length(Y) #the resulting vector has a smaller length now

theta_hat_IS2 = (1/n_y)*sum(h(Y)*f(Y)/g(Y))
print(theta_hat_IS2)
```

```
## [1] 0.522795
```

```r
#standard error
hfg_bar2 = theta_hat_IS

ste_IS2 = (1/(n_y*(n_y-1)))*sum(((h(Y)*f(Y)/g(Y)) - hfg_bar2)^2)
print(ste_IS2)
```

```
## [1] 4.02658e-06
```

Thus $\hat{\theta^*}_{IS} = 0.522795$ and $ste(\hat{\theta^*}_{IS}) = 4.026576 \times 10^{-6}$.

Note that we are throwing away about three fourth of the values we generate ($n_y = 4883$ out of 20000). Therefore if we really want to have a effective sample size of 20000, we should generate 80000 values instead:

```r
n = 80000
X = rcauchy(n,0,1)
Y = X[which((X>=0) & (X<=1))]

n_y = length(Y) #the resulting vector has a smaller length now

theta_hat_IS2 = (1/n_y)*sum(h(Y)*f(Y)/g(Y))
print(theta_hat_IS2)
```

```
## [1] 0.524007
```

```r
#standard error
hfg_bar2 = theta_hat_IS

ste_IS2 = (1/(n_y*(n_y-1)))*sum(((h(Y)*f(Y)/g(Y)) - hfg_bar2)^2)
print(ste_IS2)
```

```
## [1] 9.92972e-07
```

Note that in both cases, the standard error obtained is very small (smaller than sampling from $f(x)$) so resampling from Cauchy was easier and decreased the variance.

Problem 3:

We know that if $f(x|\Sigma)$ is the density of a tri-variate t distribution, then if
$X = (X_1, X_2, X_3) \sim f(x|\Sigma)$, then:

$$\theta = P(-\infty < X_1 < 1, -\infty < X_2 < 4, -\infty < X_3 < 2) = \int_{-\infty}^{1}\int_{-\infty}^{4}\int_{-\infty}^{2} h(x)f(x)dx1dx2dx3$$

With $h(x) = 1_{\{x_1<1,\, x_2<4,\, x_3<2\}}$

Now, instead of generating the $X's$ from the tri-variate t distribution , we are going to generate the $X's$ from the $Normal(0, \Sigma)$ distribution and use the Importance Sampling method.

Let $Y_1, \ldots, Y_n \sim g(x|\Sigma)$, then:

$$w(Y_i) = \frac{w^*(Y_i)}{\sum_{i=1}^{n} w^*(Y_i)} = \frac{\frac{f(Y_i|\Sigma)}{g(Y_i|\Sigma)}}{\sum_{i=1}^{n} \frac{f(Y_i|\Sigma)}{g(Y_i|\Sigma)}}$$

Moreover, we know that $f(x|\Sigma) \propto f_\propto = (5 + x^T\Sigma^{-1}x)^{-4}$, thus we get:

$$w(Y_i) = \frac{\frac{f_\propto(Y_i|\Sigma)}{g(Y_i|\Sigma)}}{\sum_{i=1}^{n} \frac{f_\propto(Y_i|\Sigma)}{g(Y_i|\Sigma)}}$$

As a result, we can approximate θ by:

$$\hat{\theta}_{IS} = \sum_{i=1}^{n} h(Y_i) \cdot w(Y_i)$$

```
library(MASS)
library(emdbook)
```

```
## Warning: package 'emdbook' was built under R version 3.4.4
```

```
f_a <-function(x, Sigma){

    return((5+ t(x) %*% solve(Sigma) %*% x)^(-4))
}

g <- function(x,mu,Sigma){

    return(dmvnorm(x, mu, Sigma))
}

h <- function(x){

    if((x[1]<1) & (x[2]<4) & (x[3]<2)) return(1)
    else return(0)
}

n = 20000
mu = c(0,0,0)
Sigma = matrix(c(1,3/5,1/3,3/5,1,11/15,1/3,11/15,1),3,3)

Y = mvrnorm(n, mu, Sigma)

s = 0
s2 = 0
for(i in 1:n){

    w_yi = f_a(Y[i,], Sigma)/g(Y[i,],mu,Sigma)
    s = s+w_yi

    w_yi_h = h(Y[i,])*f_a(Y[i,], Sigma)/g(Y[i,],mu,Sigma)
    s2 = s2 + w_yi_h
}

theta_hat_IS = s2/s
print(theta_hat_IS)
```

```
##          [,1]
## [1,] 0.805218
```

Thus, $\hat{\theta}_{IS} = 0.805218$. The approximation is a little off the true value $\theta^* = 0.79145379$, that is because we are resampling from a function $g(x)$ that has lighter tails than the initial sampling function $f(x)$. It should be the other way around to perform a good an accurate Importance Sampling method. Indeed, the density of a 5-df t-distribution has heavier tails than the Normal distribution. We would have a more accurate approximation if the t distribution had more degrees of freedom.