Joseph High
9E1FDC

EN.550.633 Homework # 10

Problem 5.14.) Prove that the solution of $\min_g \mathrm{Var}_g\left(H(x)\frac{f(x)}{g(x)}\right)$

is $g^*(x) = \dfrac{|H(x)|f(x)}{\int |H(x)|f(x)dx}$

Pf: Suppose that $g(x) = \dfrac{|H(x)|f(x)}{\int |H(x)|f(x)dx}$.

Claim: $g(x) = g^*(x)$; There are 2 Cases

Case I: $H(x) \geq 0$;

$$\ell = \int H(x)f(x)dx = \int |H(x)|f(x)dx$$

then

$$g(x) = \frac{|H(x)|f(x)}{\int |H(x)|f(x)dx} = \frac{H(x)f(x)}{\int H(x)f(x)dx} = \frac{H(x)f(x)}{\ell}$$

$$\Rightarrow \mathrm{Var}_g\left[H(x)\frac{f(x)}{g(x)}\right] = \mathrm{Var}_g\left[\frac{H(x)f(x)}{(H(x)f(x))/\ell}\right] = \mathrm{Var}_g[\ell]$$

$$= 0$$

Since $\mathrm{Var}_g(\cdot) \geq 0$ for $\forall g, f, H$, then this implies that

$$g(x) = \arg\min_g \mathrm{Var}_g\left[H(x)\frac{f(x)}{g(x)}\right]$$

i.e. $g(x) = g^*(x)$ for $H(x) \geq 0$

Case II: $H(x) < 0$ (Equivalent to proving for arbitrary $H(x)$ since $H(x) \geq 0$ already proven.)

Then,

$$\mathrm{Var}_g\left[H(x)\frac{f(x)}{g(x)}\right] = \mathbb{E}_g\left[\left(H(x)\cdot\frac{f(x)}{g(x)}\right)^2\right] - \left(\mathbb{E}_g\left[H(x)\cdot\frac{f(x)}{g(x)}\right]\right)^2$$

$\Rightarrow$

Joseph High

Problem A: (Refer to R code attached for code, solutions, answers, etc.)

(a.) The true value of $\mathbb{P}\{|X| \geq 3.5\}$ for $X \sim N(0,1)$ was computed in R and found to be

$$\ell = \mathbb{P}\{|X| \geq 3.5\} = 0.0004652582$$

~~(Refer to R code)~~

(b.) Since we want to estimate $\ell = \mathbb{P}\{|X| \geq 3.5\}$ for $X \sim N(0,1)$, then

$$\ell = \mathbb{E}_f[H(X)] = \mathbb{E}_f[\mathbb{I}_{\{|X| \geq 3.5\}}]$$

$$\Rightarrow H(X) = \mathbb{I}_{\{|X| \geq 3.5\}} = \begin{cases} 1, & X \in (-\infty, -3.5] \cup [3.5, \infty) \\ 0, & X \in (-3.5, 3.5) \end{cases}$$

Using R, the 10 estimates $\hat{\ell}$ using $N = 100{,}000$ were found to be:

	1	2	3	4	5	6
$\hat{\ell} =$	0.00047	0.00045	0.00054	5×10^{-4}	0.00044	3×10^{-4}

	7	8	9	10
$\hat{\ell} =$	0.00037	0.00051	0.00058	0.00044

(Refer to R code)

Now, to compute $Var(\hat{\ell})$: $\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} H(X_i)$

Since $H(X_i)$ generated independently:

$$Var[\hat{\ell}] = Var\left[\frac{1}{N}\sum_{i=1}^{N} H(X_i)\right] = \frac{1}{N^2}\sum_{i=1}^{N} Var[H(X_i)]$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\left(\mathbb{E}_f[(H(X_i))^2] - (\mathbb{E}_f[H(X_i)])^2\right) =$$

$\Longrightarrow$

Joseph High

Using the var function in R, the variance of the 10 estimates was found to be (Refer to R code)

$$Var(\hat{\ell}) = 7.958055 \times 10^{-12}$$

which is significantly lower than the variance of the standard CMC estimates.

Part (d): Using R, the relative error for each method was computed and found to be:

$$RE_{\substack{\text{Standard}\\\text{Method}}} = \frac{\sqrt{Var(\hat{\ell})}}{\mathbb{E}(\hat{\ell})} = \frac{\sqrt{4.650417 \times 10^{-9}}}{\ell} = \frac{\sqrt{4.650417 \times 10^{-9}}}{0.0004652582}$$

(Since $\hat{\ell}$ unbiased estimate of ℓ)

$$= 0.1465723$$
$$\sim 14.66\%$$

$$RE_{\substack{\text{IS}\\\text{Method}}} = \frac{\sqrt{Var(\hat{\ell})}}{\mathbb{E}[\hat{\ell}]} = \frac{\sqrt{7.958055 \times 10^{-12}}}{0.0004652582} = 0.006063306$$
$$\sim 0.61\%$$

The Relative error for the IS method is significantly smaller than the relative error for the standard CMC method. Thus, the relative accuracy of the IS estimates to the true value is much greater than that of the standard CMC estimates — that is, the IS estimates are much more accurate than the standard CMC estimates. Indeed, the IS method uses a standard normal proposal density centered at 3.5, significantly increasing the probability that samples from $\{X : |X| \geq 3.5\}$ are generated. //

Joseph Hugh

(Extra credit cont'd...)

$$= \frac{1}{\sqrt{2\pi}}\left(\int_{3.5}^{\infty}(-x-\mu)e^{-\frac{1}{2}x^2-\mu x+\frac{1}{2}\mu^2}dx + \int_{-\infty}^{-3.5}(-x-\mu)e^{-\frac{1}{2}x^2-\mu x+\frac{1}{2}\mu^2}dx\right) = 0$$

$$= \frac{e^{\frac{1}{2}\mu^2}}{\sqrt{2\pi}}\left(\int_{3.5}^{\infty}(-x-\mu)e^{-\frac{1}{2}(x^2+2\mu x)} + \int_{-\infty}^{-3.5}(-x-\mu)e^{-\frac{1}{2}x^2-\mu x}dx\right) = 0$$

$$= \int_{3.5}^{\infty}(-x-\mu)e^{-\frac{1}{2}x^2-\mu x}dx + \int_{-\infty}^{-3.5}(-x-\mu)e^{-\frac{1}{2}x^2-\mu x}dx = 0$$

let $u = -\frac{1}{2}x^2-\mu x$

$du = (-x-\mu)dx$

$$\Rightarrow \int_{u_1(3.5)}^{u_2(\infty)} e^u du + \int_{u_1(-\infty)}^{u_2(-3.5)} e^u du = 0$$

$$= e^{-\frac{1}{2}x^2-\mu x}\Big|_{3.5}^{\infty} + e^{-\frac{1}{2}x^2-\mu x}\Big|_{-\infty}^{-3.5} = 0$$

$$= -e^{-\frac{1}{2}(3.5)^2-\mu(3.5)} + e^{-\frac{1}{2}(3.5)^2+\mu(3.5)} = 0$$

$$\Rightarrow \cancel{-\frac{1}{2}(3.5)^2}-\mu(3.5) = \cancel{-\frac{1}{2}(3.5)^2}+\mu(3.5)$$

$$2\mu(3.5) = 0$$

Almost had it!

Must have made a mistake somewhere ☹

Joseph High
9E1FDC

HW 10 R Code and Results

```
> #Problem A
> #Part (a)
> P = pnorm(3.5, mean = 0, sd = 1, lower.tail = FALSE)  #Probability X
> 3.5 (Uppertail only)
> L = 2*P  #Since N(0,1) is a symmetric distribution, P(abs(X) > 3.5)
= 2*P(X > 3.5)
> #Thus, l = P(abs(X) > 3.5) = 0.0004652582
>
>
> #Part (b)  - Estimating using CMC
> set.seed(1)
> N = 10^5                  #number of independent samples
> x = matrix(NA, N, 10);  #storage for the 10 estimates
> H = matrix(NA, N, 10);  #storage for the 10 estimates
> L.hat = matrix(NA, 1, 10);
> for(j in 1:10){
+   x[,j] = rnorm(N, mean = 0, sd = 1)  # sampling from f
+   for (i in 1:N) {
+     if (x[i,j] >= qnorm(1-P)) {
+       H[i,j] = 1
+     }
+     else if (x[i,j] <= qnorm(P)) {
+       H[i,j] = 1
+     }
+     else {
+       H[i,j] = 0
+     }
+   }
+   for(k in 1:10) {
+     L.hat[,k] = (1/N)*sum(H[,k])
+   }
+ }
>
> show(L.hat)  #list of all 10 estimates of Prob(abs(X) >= 3.5)
        [,1]    [,2]    [,3]  [,4]    [,5]  [,6]    [,7]    [,8]
[,9]
[1,] 0.00047 0.00045 0.00054 5e-04 0.00044 3e-04 0.00037 0.00051
0.00058
       [,10]
[1,] 0.00044
>
> #Variance of L.hat:
> var.Lhat = (1/N)*(L - L^2)
>
> var.Lhat  #Var(L.hat) = ~ 4.650417e-09
[1] 4.650417e-09
>
>
> #Part (c)  - Estimating using IS with g = N(3.5, 1):
```

Joseph High
9E1FDC

```
> y = matrix(NA, N, 10);
> h = matrix(NA, N, 10);
> w = matrix(NA, N, 10);
> f = matrix(NA, N, 10);
> g = matrix(NA, N, 10);
> Lhat.IS = matrix(NA, 1, 10);
> for(j in 1:10){
+   y[,j] = rnorm(N, mean = 3.5, sd = 1) #sampling from g
+   f[,j] = dnorm(y[,j], mean = 0, sd = 1)
+   g[,j] = dnorm(y[,j], mean = 3.5, sd = 1)
+   w[,j] = f[,j]/g[,j]   #weights
+   for (i in 1:N) {
+     if (y[i,j] >= qnorm(1-P)) {
+       h[i,j] = 1
+     }
+     else {
+       h[i,j] = 0
+     }
+   }
+   for(k in 1:10) {
+     Lhat.IS[,k] = 2*(1/N)*sum(h[,k]*w[,k])  #Since g is symmetric,
multiply
+                                             #by 2 to attain an
estimate for
+                                             #both tails.
+     }
+ }
> show(Lhat.IS)
             [,1]         [,2]        [,3]         [,4]         [,5]
[1,] 0.0004628242 0.0004625311 0.000467592 0.0004650986 0.0004611431
            [,6]         [,7]         [,8]         [,9]        [,10]
[1,] 0.000460414 0.0004627089 0.0004642176 0.0004686603 0.0004671747
>
> #Variance of Lhat.IS:
> var.LhatIS = var(Lhat.IS[,])
> var.LhatIS  #Variance of Importance Sampling estimate = ~ 7.958055e-
12
[1] 7.958055e-12
>
> #Part (d)
> #Relative Error, RE = sqrt(Var(L.hat))/E(L.hat) = sqrt(Var(L.hat))/L
> RE.Standard = sqrt(var.Lhat)/L
> RE.IS = sqrt(var.LhatIS)/L
> RE.Standard  #0.1465723 ~ 14.66% error relative to the true value
[1] 0.1465723
> RE.IS  #0.006063306 ~ 0.61% error relative to the true value
[1] 0.006063306
>
> #The Relative error for the Importance Sampling method is
significantly
> #smaller than the Relative Error for the standard CMC method. Thus,
the
```

Joseph High
9E1FDC

```
> #relative accuracy of the IS estimates to the true value is much
greater
> #than that of the standard CMC estimates. Indeed, the IS method uses
a
> #standard normal distribution centered at 3.5, significantly
increasing the
> #probability that samples of X >=3.5 are generated.
>
> #Extra Credit  Part (e)
y = matrix(NA, N, 10);
h = matrix(NA, N, 10);
w = matrix(NA, N, 10);
f = matrix(NA, N, 10);
g = matrix(NA, N, 10);
Lhat.IS = matrix(NA, 1, 10);
for(j in 1:10){
  y[,j] = rnorm(N, mean = mu*, sd = 1) #sampling from g
  f[,j] = dnorm(y[,j], mean = 0, sd = 1)
  g[,j] = dnorm(y[,j], mean = mu*, sd = 1)
  w[,j] = f[,j]/g[,j]   #weights
  for (i in 1:N) {
    if (y[i,j] >= qnorm(1-P)) {
      h[i,j] = 1
    }
    else {
      h[i,j] = 0
    }
  }
  for(k in 1:10) {
    Lhat.IS[,k] = 2*(1/N)*sum(h[,k]*w[,k])  #Since g is symmetric,
multiply
                                            #by 2 to attain an
estimate for
                                            #both tails.
    }
}
```