Project 2

Computer Intensive Statistical Methods

Erling og Christian

03 02 2022

Contents

This is from Fab

```
generate_from_normal <- function(n) {
    u1 <- runif(n)
    u2 <- runif(n)
    r <- sqrt(-2 * log(u1))
    x_1 <- 2 * pi * u2
    x = r * cos(x_1)
    return(x)
}
```

Problem C: Monte Carlo integration and variance reduction

Intro, do we need it?\ Here we will consider Monte Carlo integration to estimate $\theta = P(X > 4)$ when $X \sim N(0,1)$. Then we will compare the variance reduction in importance sampling and antithethic sampling.

1 Monte Carlo integration

Let $h(X) = I(X > 4)$, where I is the indicator function, such that

$$\begin{aligned} E[h(X)] &= \int_{-\infty}^{\infty} h(x)f(x)dx \\ &= \int_{-\infty}^{\infty} I(x > 4)f(x)dx \\ &= P(X > 4) \\ &= \theta. \end{aligned}$$

Then, the Monte Carlo Estimate of θ is given by

$$\hat{\theta}_{MC} = \frac{1}{n}\sum_{i=1}^{n} h(x_i).$$

Now we will find a $1-\alpha$ confidence interval for θ based on our sample set. First we need the expected value of the Monte Carlo estimator

$$\begin{aligned} E[\hat{\theta}] &= E[\frac{1}{n}\sum_{i=1}^{n} h(x_i)] \\ &= \frac{1}{n}\sum_{i=1}^{n} \theta \\ &= \theta, \end{aligned}$$

and it's variance

$$\begin{aligned} Var(\hat{\theta}) &= Var(\frac{1}{n}\sum_{i=1}^{n} h(x_i)) \\ &= \frac{1}{n^2}\sum_{i=1}^{n} Var(h(x_i)) \\ &= \frac{1}{n}\frac{1}{n-1}\sum_{i=1}^{n}(h(x_i) - \hat{\theta})^2. \end{aligned}$$

Then we get the statistic

$$T_{MC} = \frac{\hat{\theta}_{MC} - \theta}{\sqrt{\hat{Var}(\hat{\theta}_{MC})}} \sim t_{n-1}.$$

Below there is an implementation with $n = 100000$ samples of X which we use to find the Monte Carlo estimate θ. Is there something wrong with this variance?

```
set.seed(321)  # Seed for reproducibility.
n = 1e+05
x = generate_from_normal(n)  # Drawing n samples fron N(0,1)
h = x > 4
MCest = mean(h)  # Monte Carlo estimate
theta = pnorm(4, lower.tail = F)  # True theta
# Confidence interval and results
svMC = sum((h - MCest)^2)/(n * (n - 1))  # Sample Variance
alpha = 0.05
t = qt(alpha/2, n - 1, lower.tail = F)  # (1-alpha) significance
lwrUpr = sqrt(svMC) * t  # lower and upper deviation from mean
ciMC = MCest + c(-lwrUpr, lwrUpr)
resultMC = c(Estimator = MCest, Confint = ciMC, Var = svMC, error = abs(theta - MCest))
resultMC
```

```
##     Estimator      Confint1      Confint2           Var         error
## 5.000000e-05 6.174219e-06 9.382578e-05 4.999800e-10 1.832876e-05
```

```
theta
```

```
## [1] 3.167124e-05
```

Here we see and error in the $1 \cdot 10^{-5}$ decimal and that the true value coincide with the 95% confidence interval.

2 Importance sampling

Here we will use importance sampling on the same problem as in C1 to try to reduce the variance of the Monte Carlo integration. The proposal distribution is

$$g(x) = \begin{cases} cxe^{-x^2/2} & , \ x > 4 \\ 0 & , \ \text{otherwise}, \end{cases}$$

where c is a normalizing constant. Now, let $x_1, ..., x_n \overset{i.i.d}{\sim} g(x)$ and let $w_i = w(x_i) = f(x_i)/g(x_i) = f_i/g_i$ be the weights. (where w_i, f_i and g_i are function evaluations at x_i). Then, the importance sampling estimator of θ is

$$\hat{\theta}_{IS} = \frac{\sum_{i=1}^{n} h_i w_i}{n}.$$

In order to use inversion sampling on the proposal distribution g we need its cdf

$$\begin{aligned} G(x) &= \int_4^x cye^{-y^2/2} dy \\ &= \int_8^{x^2/2} ce^{-u} du \\ &= \left[-ce^{-u}\right]_8^{x^2/2} \\ &= c(e^{-8} - e^{-x^2/2}). \end{aligned}$$

Since g is a distribution, and therefore $\int_4^\infty g(x)dx = 1$, we can find c by solving

$$\begin{aligned} c(e^{-8} - e^{-x^2/2})\Big|_{x=\infty} &= 1 \\ c &= e^8. \end{aligned}$$

Then we have $G(x) = 1 - e^{8 - x^2/2}$. Now we can sample from g by solving $U = G(x) \sim Unif(0,1)$ for x, that is,

$$\begin{aligned} U &= 1 - e^{8 - x^2/2} \\ -2ln(1-U) &= x^2 - 16 \\ x &= \sqrt{16 - 2ln(1-U)}. \end{aligned}$$

Thus, our samples are generated by inserting randomly selected $U \sim Unif(0,1)$ admits samples from $X \sim g$.

We also need the expected value,

$$
\begin{aligned}
E[\hat{\theta}_{IS}] &= E\left[\frac{\sum_{i=1}^{n} h_i w_i}{n}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_0^\infty h_i \frac{f_i}{g_i} g_i dx \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_0^\infty h_i f_i dx \\
&= \frac{1}{n}\sum_{i=1}^{n} E[h_i] \\
&= \frac{1}{n} n\theta \\
&= \theta,
\end{aligned}
$$

and the sample variance,

$$
\begin{aligned}
Var(\hat{\theta}_{IS}) &= Var\left(\frac{\sum_{i=1}^{n} h_i w_i}{n}\right) \\
&= \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(h_i w_i - \sum_{i=1}^{n}\frac{h_i w_i}{n}\right)^2 \\
&= \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(h_i w_i - \hat{\theta}_{IS}\right)^2,
\end{aligned}
$$

of the importance sample estimator to compute the $(1-\alpha)$ confidence interval. Below we have implemented the computation of the importance sample estimate along with the 95% confidence interval.

```
expSampler <- function(n) {
    # Samples from proposal distribution g
    u = runif(n)
    return(sqrt(16 - 2 * log(1 - u)))
}

w <- function(x) {
    # Weight function
    f = dnorm(x)
    g = ifelse(x > 4, x * exp(8 - 0.5 * x^2), 0)
    return(f/g)
}
set.seed(321)
gx = expSampler(n)  # Sample from proposal
gh = (gx > 4) * 1
ISest = mean(gh * w(gx))
svIS = sum((gh * w(gx) - ISest)^2)/(n * (n - 1))
ISconfint = ISest + c(-t * sqrt(svIS), t * svIS)
# Results
resultIS = c(ISestimate = ISest, confint = ISconfint, var = svIS, error = abs(theta -
    ISest))
results = rbind(MC = resultMC, IS = resultIS)
results
```

```
##         Estimator      Confint1      Confint2           Var         error
```

```
## MC 5.000000e-05 6.174219e-06 9.382578e-05 4.999800e-10 1.832876e-05
## IS 3.167611e-05 3.166649e-05 3.167611e-05 2.410122e-17 4.866683e-09
```

```
theta
```

```
## [1] 3.167124e-05
```

Here we see that importance sampling has reduced the variance by a factor of $Var(\hat{\theta}_{MC})/Var(\hat{\theta}_{IS}) =$ svMC/svIS $= 2.074501 \times 10^7$. Also, the importance sample estimator is a much more precise estimate.

Not sure how to compute how many more samples we would need, since the error influences.

3 Antithetic Sampling

Now we will combine the importance sampling with the use of antithetic variates. We start by modifying the sample generator for g in task C2 so that it produces n pairs, $X = x_i$ and $Y = y_i$, of antithetic variates by using u_i and $1 - u_i$ as input to G^{-1}, respectively.

```
expSamplerAnti <- function(n) {
    # Pairwise sampling from proposal distribution evaluated at u and 1-u.
    u = runif(n)
    return(data.frame(x = sqrt(16 - 2 * log(1 - u)), y = sqrt(16 - 2 * log(u))))
}
# xy <- expSamplerAnti(n) gx <- ifelse(xy$x>4, xy$x*exp(8-0.5*xy$x^2),0) gy <-
# ifelse(xy$y>4, xy$y*exp(8-0.5*xy$y^2),0) plot(xy$x,gx) points(xy$y,gy,
# col='cyan3',cex=1)
```

Then, the importance sample estimates for each of the pairs are

$$\hat{\theta}_X = \frac{1}{n}\sum_{i=1}^{n} h(x_i)w(x_i),$$
$$\hat{\theta}_Y = \frac{1}{n}\sum_{i=1}^{n} h(y_i)w(y_i),$$

and the antithetic sample estimator is

$$\hat{\theta}_A = \frac{\hat{\theta}_X + \hat{\theta}_Y}{2}.$$

We also need the expected value Should we show that each expectation is theta?

$$\begin{aligned}
E[\hat{\theta}_{AS}] &= E\left[\frac{\hat{\theta}_X + \hat{\theta}_Y}{2}\right] \\
&= \frac{1}{2n}\sum_{i=1}^{n}(E[h(x_i)w(x_i)] + E[h(y_i)w(y_i)]) \\
&\overset{*}{=} \frac{1}{2n}\sum_{i=1}^{n} 2\theta \\
&= \theta,
\end{aligned}$$

where $*$ since the proposal distribution evaluations cancel in each expectation. not true or not needed?. The variance of the estimator is

$$
\begin{aligned}
Var(\hat{\theta}_{AS}) &= \frac{1}{4}(Var(\hat{\theta}_X) + Var(\hat{\theta}_Y) + 2Cov(\hat{\theta}_X, \hat{\theta}_Y)) \\
\Downarrow & \quad Var(\hat{\theta}_X) = Var(\hat{\theta}_Y) \\
&= \frac{(1+\rho_{XY})S^2_{XY}}{2n},
\end{aligned}
$$

where $\rho_{XY} = Cov(\hat{\theta}_X, \hat{\theta}_Y)$ and S^2_{XY} is the sample variance of either estimator $\hat{\theta}_X$ or $\hat{\theta}_Y$.

```
set.seed(321)
n = 50000
xy = expSamplerAnti(n)
hxy = (xy > 4) * 1
hwx = hxy[, "x"] * w(xy$x)
hwy = hxy[, "y"] * w(xy$y)
hwxy = (hwx + hwy)/2
ASest = mean(hwxy)
var(hwxy)
```

```
## [1] 2.851883e-13
```

```
svAS = (var(hwx) + var(hwy) + 2 * cov(hwx, hwy))/4
lwrUprAS = c(-t, t) * sqrt(svAS)
confintAS = ASest + lwrUprAS
resultAS <- c(ASest, confintAS, svAS, abs(theta - ASest))
rbind(results, AS = resultAS)
```

```
##       Estimator     Confint1     Confint2          Var        error
## MC 5.000000e-05 6.174219e-06 9.382578e-05 4.999800e-10 1.832876e-05
## IS 3.167611e-05 3.166649e-05 3.167611e-05 2.410122e-17 4.866683e-09
## AS 3.167262e-05 3.062593e-05 3.271931e-05 2.851883e-13 1.378996e-09
```