Computational Statistics Homework 2

吴嘉骜 21307130203

2023 年 10 月 4 日

1 ex 5.2

Refer to Example 5.3. Compute a Monte Carlo estimate of the standard normal cdf, by generating from the Uniform$(0, x)$ distribution. Compare your estimates with the normal cdf function `pnorm`. Compute an estimate of the variance of your Monte Carlo estimate of $\Phi(2)$, and a 95% confidence interval for $\Phi(2)$.

Solution:

The R code and result are as follows:

```
# Monte Carlo estimate of the standard normal cdf
x <- seq(0.1,2.5,length=10)
m <- 10000
cdf <- numeric(length(x))
for(i in 1:length(x)){
    j <- runif(m,0,x[i])
    A <- exp(-j^2/2)
    cdf[i] <- 0.5 + x[i]*mean(A)/sqrt(2*pi)
}
Phi <- pnorm(x)
print(round(rbind(x,cdf,Phi),3))


> print(round(rbind(x,cdf,Phi),3))
    [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
x   0.10 0.367 0.633 0.900 1.167 1.433 1.700 1.967 2.233 2.500
cdf 0.54 0.643 0.737 0.816 0.878 0.923 0.955 0.977 0.983 0.993
Phi 0.54 0.643 0.737 0.816 0.878 0.924 0.955 0.975 0.987 0.994
```

From the result, we can see that the Monte Carlo estimate is very close to the normal cdf function `pnorm`.
To estimate the variance of the Monte Carlo estimate of $\theta = \Phi(2)$, notice that the estimator $\hat{\theta}$ is computed by $\hat{\theta} = 0.5 + \frac{x}{m}\sum_{i=1}^{m} g(x_i)$, where $g(t) = \frac{1}{\sqrt{2\pi}}e^{-t^2/2}$ and $x_i \sim$ Uniform$(0,x)$, iid.
So $Var(\hat{\theta}) = \frac{x^2}{m}Var(g(x))$. We can estimate $Var(g(x))$ by the sample variance of $g(x_i)$, $i = 1, 2, \ldots, m$.
To compute a 95% confidence interval for θ, by the Central Limit Theorem, we have $\hat{\theta} \sim N(\theta, Var(\hat{\theta}))$ approximately for large m. Then an approximate 95% confidence interval for θ is given by

$$\hat{\theta} \pm z_{0.025}\sqrt{\widehat{Var(\hat{\theta})}}.$$

we can use the following R code:

```
# an estimate of the variance of Monte Carlo estimate of Φ(2)
x <- 2
m <- 10000
j <- runif(m,0,x)
g <- exp(-j^2/2)
theta <- x*mean(g)/sqrt(2*pi) + 0.5
varg <- mean((g - mean(g))^2)
varhat <- x^2*varg/(m*2*pi)
c(theta, varhat)
pnorm(2)
# a 95% confidence interval for Φ(2)
theta + qnorm(c(0.025, 0.975)) * sqrt(varhat)

> c(theta, varhat)
[1] 9.770137e-01 5.327914e-06
> pnorm(2)
[1] 0.9772499
> # a 95% confidence interval for Φ(2)
> theta + qnorm(c(0.025, 0.975)) * sqrt(varhat)
[1] 0.9724897 0.9815377
```

2 ex 5.3

Compute a Monte Carlo estimate $\hat{\theta}$ of

$$\theta = \int_0^{0.5} e^{-x} dx$$

by sampling from Uniform(0, 0.5), and estimate the variance of $\hat{\theta}$. Find another Monte Carlo estimator θ^* by sampling from the exponential distribution. Which of the variances (of $\hat{\theta}$ and θ^*) is smaller, and why?

Solution:

First we estimate θ by sampling from Uniform(0, 0.5). The R code and results are as follows:

```
# Monte Carlo estimate from Uniform(0, 0.5).
m <- 10000
u <- runif(m,0,0.5)
g <- exp(-u)
theta <- 0.5*mean(g)
theta
1 - exp(-0.5)
# estimate variance1
var1 <- 0.5*0.5* mean((g-mean(g))^2)/m
var1

> theta
[1] 0.3935569
> 1 - exp(-0.5)
[1] 0.3934693
> # estimate variance1
> var1 <- 0.5*0.5* mean((g-mean(g))^2)/m
> var1
[1] 3.221585e-07
```

From the result, we can see that the Monte Carlo estimate $\hat{\theta}$ is very close to the exact value.
Then we estimate θ by sampling from the exponential distribution. The R code and results are as follows:

```
# sampling from the exponential distribution
m <- 10000
v <- rexp(m,1)
theta <- mean(v<=0.5)
theta
p <- 1 - exp(-0.5)
p
# estimate variance2
var2 <- theta*(1-theta)/m
var2
accvar <- p*(1-p)/m
accvar

> theta
[1] 0.3938
> p <- 1 - exp(-0.5)
> p
[1] 0.3934693
> # estimate variance2
> var2 <- theta*(1-theta)/m
> var2
[1] 2.387216e-05
> accvar <- p*(1-p)/m
> accvar
[1] 2.386512e-05
```

We compare the variances of $\hat{\theta}$ and θ^* as follows:

```
> var1/var2
[1] 0.01349516
```

The result shows that $Var(\hat{\theta}) < Var(\theta^*)$. The reason is that the exact value of $Var(\hat{\theta})$ is smaller than the accurate value of $Var(\theta^*)$, which is $\theta(1-\theta)/m = 2.386512e-05$.
Now we calculate the exact value of $Var(\hat{\theta})$ to clarify the reason.
Var($\hat{\theta}$) $= \frac{0.5^2}{m} Var(g(U))$, where $U \sim$ Uniform(0,0.5).

$$
\begin{aligned}
Var(g(U)) &= E(g^2(U)) - E(g(U))^2 \\
&= \int_0^{0.5} 2e^{-2x}dx - (\int_0^{0.5} 2e^{-x}dx)^2 \\
&= 1 - e^{-1} - 4(1 - e^{-0.5})^2 \\
&= 8e^{-0.5} - 5e^{-1} - 3
\end{aligned}
$$

So $Var(\hat{\theta}) = \frac{0.5^2}{m}(8e^{-0.5} - 5e^{-1} - 3) = 3.212018e{-}07$.
Then $\frac{Var(\hat{\theta})}{Var(\theta^*)} = 0.01345905$.

3 ex 5.6

In Example 5.7 the control variate approach was illustrated for Monte Carlo integration of

$$\theta = \int_0^1 e^x dx$$

Now consider the antithetic variate approach. Compute $Cov(e^U, e^{1-U})$ and $Var(e^U + e^{1-U})$, where $U \sim$ Uniform(0,1). What is the percent reduction in variance of $\hat{\theta}$ that can be achieved using antithetic variates (compared with simple MC)?

Solution:

$Cov(e^U, e^{1-U}) = E(e^U e^{1-U}) - E(e^U)E(e^{1-U}) = e - (e-1)^2 = -0.2342106$.

$Var(e^U) = Var(e^{1-U}) = E(e^{2U}) - (E(e^U))^2 = 0.5(e^2 - 1) - (e-1)^2 = 0.2420356$.

$Var(e^U + e^{1-U}) = 2Var(e^U) + 2Cov(e^U, e^{1-U}) = e^2 - 1 - 4(e-1)^2 + 2e = 0.01564999$.

If we use the antithetic variate approach, we have $Var(\hat{\theta}_{ant}) = Var(\frac{1}{2}(e^U + e^{1-U})) = 0.003912497$.

Otherwise, using the simple Monte Carlo method, we have $Var(\hat{\theta}) = Var(\frac{1}{2}(e^U + e^V)) = \frac{1}{2}Var(e^U) = 0.1210178$, where $V \sim$ Uniform(0,1) and U and V are independent.

Then the percent reduction in variance that can be achieved using antithetic variates is

$$\frac{Var(\hat{\theta}) - Var(\hat{\theta}_{ant})}{Var(\hat{\theta})} = 0.96767 = 96.767\%.$$

4 ex 5.7

Refer to Exercise 5.6. Use a Monte Carlo simulation to estimate θ by the antithetic variate approach and by the simple Monte Carlo method. Compute an empirical estimate of the percent reduction in variance using the antithetic variate. Compare the result with the theoretical value from Exercise 5.6.

Solution:

The R code and results are as follows:

```
# estimate theta by the antithetic variate approach
m <- 10000
u <- runif(m/2)
v <- 1-u
g1 <- (exp(u)+exp(v))/2
anti <- mean(g1)
v1 <- mean((g1-anti)^2)/(m/2)

# estimate theta by the simple Monte Carlo method
u2 <- runif(m)
g2 <- exp(u2)
smc <- mean(g2)
v2 <- mean((g2-smc)^2)/m

c(anti, smc)
c(v1,v2)
# reduction
(v2 - v1)/v2

> c(anti, smc)
[1] 1.717577 1.709982
```

```
> c(v1,v2)
[1] 7.730742e-07 2.387231e-05
> # reduction
> (v2 - v1)/v2
[1] 0.9676163
```

The result shows that the empirical estimate of the percent reduction in variance is 96.762%, which is very close to the theoretical value 96.767% in Exercise 5.6.

5 ex 5.8

Let $U \sim$ Uniform(0,1), $X = aU$, and $X' = a(1-U)$, where a is a constant. Show that $\rho(X, X') = -1$. Is $\rho(X, X') = -1$ if U is a symmetric beta random variable?

Proof:

For all U with finite mean and variance, let $\mu = E(U)$, $\sigma^2 = Var(U)$.

$Var(X) = Var(aU) = a^2Var(U) = a^2\sigma^2$.

$Var(X') = Var(a(1-U)) = a^2Var(1-U) = a^2Var(U) = a^2\sigma^2$.

$Cov(X, X') = Cov(aU, a(1-U)) = a^2Cov(U, 1-U) = a^2[E(U(1-U)) - E(U)E(1-U)] = a^2[\mu - (\mu^2 + \sigma^2) - \mu(1-\mu)] = -a^2\sigma^2$.

$\rho(X, X') = \frac{Cov(X,X')}{\sqrt{Var(X)Var(X')}} = -1$.

In particular, the conclusion holds when $U \sim$ Uniform(0,1) or U is a symmetric beta random variable.

6 ex 5.10

Use Monte Carlo integration with antithetic variables to estimate

$$\theta = \int_0^1 \frac{e^{-x}}{1+x^2}dx,$$

and find the approximate reduction in variance as a percentage of the variance without variance reduction.

Solution:

The R code and results are as follows:

```
# estimate   by the antithetic variate approach
m <- 10000
u <- runif(m/2)
v <- 1-u
g1 <- (exp(-u)/(1 + u^2)+exp(-v)/(1 + v^2))/2
anti <- mean(g1)
v1 <- mean((g1-anti)^2)/(m/2)

# estimate   by the simple Monte Carlo method
u2 <- runif(m)
g2 <- exp(-u2)/(1 + u2^2)
smc <- mean(g2)
v2 <- mean((g2-smc)^2)/m
```

```
c(anti, smc)
c(v1,v2)
# reduction
(v2 - v1)/v2

> c(anti, smc)
[1] 0.5243512 0.5232506
> c(v1,v2)
[1] 2.142252e-07 6.061683e-06
> # reduction
> (v2 - v1)/v2
[1] 0.9646591
```

The approximate reduction in variance as a percentage of the variance without variance reduction is 96.4659%.

7

Monte Carlo method can be used to approximate the fraction of a d-dimensional hypersphere which lies in the inscribed d-dimensional hypercube. Simulate with different dimensions $d = 2, 3, 4, \ldots, 10$. (Hint: use `apply` function.)

(1) Derive the formula for the EXACT values for the above problem for each d-dimension.

(2) Using the above formula, approximate the value of π. Find the sample size needed to approximate π to its 5th digit, i.e., the first time when you have an estimate as 3.14159x, for each dimension d. Set the random seed with `set.seed(123)` at the beginning of your R code.

Solution:

(1) The volume of a d-dimensional hypersphere with radius r is given by:

$$V_d = \frac{\pi^{d/2}}{\Gamma(d/2+1)} r^d.$$

where Γ is the gamma function.

We assume that the hypersphere is inscribed in a hypercube with side length 1, then $r = 0.5$.

The volume of a d-dimensional hypercube with side length l is $V_d' = l^d$, and for our problem, $l = 1$.

The fraction f_d of the d-dimensional hypersphere that lies in the inscribed d-dimensional hypercube is:
$f_d = \frac{V_d}{V_d'} = \frac{\pi^{d/2}}{\Gamma(d/2+1)} 2^{-d}$,
which is the desired formula.

(2) Using the method of Monte Carlo, we can approximate π by randomly generating points in the d-dimensional hypercube and checking how many of those points lie within the hypersphere. The ratio of points inside the hypersphere to the total number of points gives an approximation of f_d, which we can then use to solve for π.

The R code is as follows:

```
set.seed(123)

# use Monte Carlo to approximate pi for dimension d
# N: total sample number
```

```r
# prevcount: previous count
mcpi <- function(d,N,prevcount=0){
  count <- prevcount

  if (count==0){
    for(i in 1:N){
      u <- runif(d,-0.5,0.5)
      if(sum(u^2)<=0.5*0.5)
        count <- count+1
    }
  }
  # skip the previous samples
  else{
    u <- runif(d,-0.5,0.5)
    if(sum(u^2)<=0.5*0.5)
      count <- count+1
  }
  f <- count/N
  pi_est <- (f * 2^d * gamma(d/2 + 1))^(2/d)
  return(list(pi_est = pi_est, count = count))
}

# determine the minimum n to approximate   to its 5th digit
sizeto5 <- function(d){
  N <- 2000
  prevresult <- mcpi(d,N)
  while (TRUE) {
    if (trunc(prevresult$pi_est*1e5) == 314159)
      return(c(Dimension = d, Sample_Size = N, Pi_Approximation = round(prevresult$pi_est, 6)))
    N <- N+1
    prevresult <- mcpi(d, N, prevresult$count)
  }
}

dims <- 2:10
results <- lapply(dims, sizeto5)

results_df <- as.data.frame(do.call(rbind, results))
print(results_df)
```

In the code above, the function `mcpi` is used to approximate π for dimension d, and the function `sizeto5` is used to determine the minimum n to approximate π to its 5th digit. Empirically, we choose to start with $N = 2000$ samples, and then increase N by 1 until we get the desired approximation. The results are as follows:

```
> print(results_df)
  Dimension Sample_Size Pi_Approximation
1         2        9753         3.141597
2         3        4089         3.141597
3         4       20141         3.141595
4         5       37126         3.141596
5         6        4025         3.141590
6         7       23461         3.141595
7         8        5109         3.141594
8         9        5588         3.141590
9        10        5220         3.141599
```