STAT 428 Statistical Computing

Homework 2 Solutions

Department of Statistics, University of Illinois at Urbana-Champaign

February 2, 2017

Ex 1

(a)

Let $g(x) = cos(\pi x/2)$ and $f(x) = 1$, $0 < x < 1$. Then we have

$$\theta = \int_0^1 cos(\pi x/2)dx = \int_0^1 f(x)g(x)dx = E_f[g(X)],$$

where $X \sim Uniform(0,1)$. Thus,

$$\hat{\theta} = \frac{1}{M}\sum_{i=1}^{M} g(X_i)$$

and

$$se(\hat{\theta}) = \sqrt{Var(\frac{1}{M}\sum_{i=1}^{M} g(X_i))} = \sqrt{Var(g(X_i))/M}$$

can be estimated by following steps.

```
set.seed(183)
m<-1000
g <- function(x) cos(pi*x/2)
hat_sample <- function(m){
  x<-runif(m,0,1)
  gx <- g(x)
  gx
}
gx = hat_sample(m)
theta_hat<-mean(gx)
theta_hat_se<-sd(gx)/sqrt(m)
cat(paste("Estimate:",round(theta_hat,4),sep=" "),
    paste("Standard Error:",round(theta_hat_se,4),sep=" "),sep="\n")
```

```
## Estimate: 0.6477
## Standard Error: 0.0096
```

(b)

Set $0 = x_0 < x_1 < ... < x_k = 1$ $(k = 4)$. For stratified Monte Carlo estimatation, we have

$$\hat{\theta} = \sum_{j=1}^{k} \hat{\theta}_j,$$

$$\hat{\theta}_j = \frac{x_j - x_{j-1}}{M_j} \sum_{i=1}^{M_j} g(X_i)$$

where $X \sim Uniform(x_j, x_{j-1})$.

```
m <- 1000 #number of replicates
k <- 4 #number of strata
r <- m / k #replicates per stratum
theta_st <- numeric(k)
for (i in 1:k)
  theta_st[i]<-sum(g(runif(r, (i-1)/k, i/k)))*(i/k-(i-1)/k)/r
stratified_theta_hat<-sum(theta_st)
cat(paste("Estimate:",round(stratified_theta_hat,4),sep=" "))
```

```
## Estimate: 0.6352
```

(c)

We will use the mid-point rule to approximate the integral. Set $0 = x_0 < x_1 < ... < x_k = 1$. Then the value of θ can be approximated by

$$\int_0^1 cos(\pi x/2)dx = \int_0^1 g(x)dx = \sum_{j=1}^{k} \int_{x_{j-1}}^{x_j} g(x)dx \approx \sum_{j=1}^{k} (x_j - x_{j-1}) f(\frac{x_j + x_{j-1}}{2})$$

```
num_integration<-function(l,u,sep){
  x<-seq(l,u,sep)
  I<-sum(sep*g(x[-length(x)]+sep/2))
  return(I)
}
theta_ni<-num_integration(0,1,0.01)
cat(paste("Estimate:",round(theta_ni,4),sep=" "))
```

```
## Estimate: 0.6366
```

(d)

The upper bound of second derivative of g(x) is given by $g''(x) = g'(-\pi/2 \cdot \sin(\pi x/2)) = -\pi^2/4 \cdot \cos(\pi x/2) \leq \pi^2/4$.The number of nodes used in integration is $n = (b-a)/\Delta = 1/sep$. The upper bound of error can be given by

$$err \leq \frac{n\Delta^3}{24} f''(\epsilon) \leq \frac{sep^2}{24} \cdot \frac{\pi^2}{4} = \frac{sep^2\pi^2}{96}$$

Reference: https://en.wikipedia.org/wiki/Rectangle_method

```
n = 1000
sep = (1-0)/n
sep^2*pi^2/96
```

```
## [1] 1.028084e-07
```

There is also a much looser upper bound for the intergral. Since cosine function is decreasing, the value of middle point is between the values of two ends. T

$$
\begin{aligned}
err &= \sum_{i=1}^{n}(x_i - x_{i-1})(f(\xi_i) - f(m_i)) \qquad \text{where } \xi_i \in [x_i, x_{i+1}] \text{ and } m_i = \frac{x_i + x_{i-1}}{2} \text{ is the middle point} \\
&\le \sum_{i=1}^{n}(x_i - x_{i-1})(f(x_{i-1}) - f(x_i)) \\
&= \sum_{i=1}^{n} sep \cdot (f(x_{i-1}) - f(x_i)) \\
&= sep \cdot \sum_{i=1}^{n}(f(x_{i-1}) - f(x_i)) \\
&= sep \cdot (f(x_0) - f(x_n)) \\
&= sep \cdot (1 - 0) \\
&= sep
\end{aligned}
$$

```
sep
```

```
## [1] 0.001
```

Ex 2

(a)

Let $g(x) = cos(\pi x/2)$ and $f(x) = 1$. For importance sampling we have

$$\theta = \int_0^1 cos(\pi x/2)dx = \int_0^1 \frac{f(x)g(x)}{\phi(x)}\phi(x)dx = E_\phi[\frac{f(x)g(x)}{\phi(x)}],$$

where $X \sim \phi(x) = 3(1 - x^2)/2$.
Here we use rejection method to sample X from $\phi(x)$ and $\theta^\star$ will be estimated similarly as the way in Question 1.

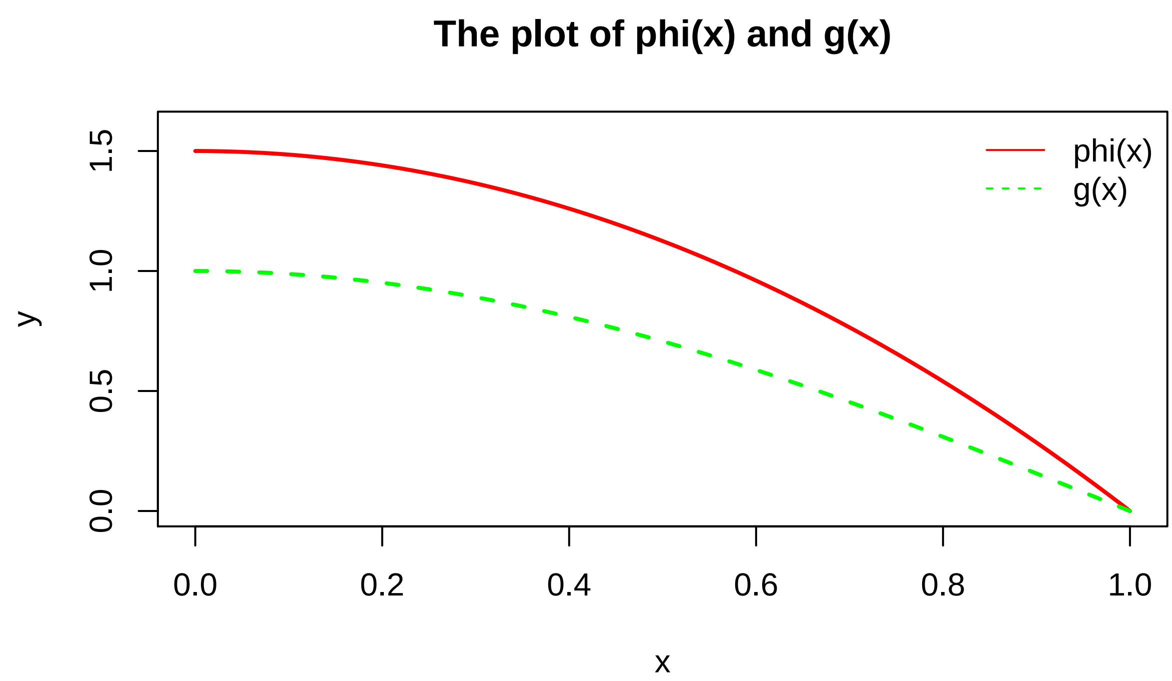
```
## Use rejection method sample from phi(x)
phi<-function(x){
  3*(1-x^2)/2
}
## phi(x)/unif(x)<=1.5 make c=2
sample.rej<-function(n){
  x=integer(n)
  i=0
  while(i<n)
  {
    y<-runif(1)
    u<-runif(1)
    ratio<-phi(y)/2
    if (u<ratio)
    {
      i<-i+1
      x[i]=y
    }
  }
```

```
  x
}
m<-1000
x<-sample.rej(m)
theta_star<-mean(g(x)/phi(x))
cat(paste("Estimate:",round(theta_star,4),sep=" "))
```

```
## Estimate: 0.6356
```

(b)

```
x<-seq(0,1,0.001)
title<-paste("The plot of",expression(phi(x)),"and",expression(g(x)),sep=" ")
plot(x,phi(x),type="l",lwd=2,col="red",lty=1,ylim=c(0,1.6),ylab="y",main=title)
lines(x,g(x),lwd=2,col="green",lty=2)
legend("topright",lty=c(1,2), col=c("red","green"), legend=c("phi(x)","g(x)"),cex=1,bty="n")
```



Comment:
The importance function $\phi(x)$ shares a similar shape with $cos(\pi x/2)$ and also covers $cos(\pi x/2)$ over the support $(0, 1)$.

Ex 3

```
m<- 1000
n<- 100
theta_hats = apply(replicate(n,hat_sample(m)),2,mean)
```

```
theta_stars = numeric(n)
for(i in 1:n){
  x<-sample.rej(m)
  theta_stars[i] <-mean(cos(pi*x/2)/phi(x))
}
cbind(hat_obs_var = var(theta_hats),star_obs_var = var(theta_stars))
```

```
##       hat_obs_var star_obs_var
## [1,] 9.699702e-05   9.3674e-07
```

The sample variances of the observations of θ^* is much smaller

Ex 4

(a)

```
mvn_pdf<-function(x,mu,sigma){
 (2*pi)^(-1)*det(sigma)^(-1/2)*exp(-0.5*t(x-mu)%*%solve(sigma)%*%(x-mu))
}
mvn_mcintegration<-function(a,b,c,d,mu,sigma,m){
  X<-cbind(runif(m,a,b),runif(m,c,d))
  I<-sum(apply(X,1,mvn_pdf,mu,sigma))*(b-a)*(d-c)/m
  return(I)
}
```

(b)

The covariance matrix equals to I means that these joint pdf can be factored into two independent parts

$$
\begin{aligned}
f(x_1, x_2) &= (2\pi)^{-k/2}|\Sigma|^{-1/2}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \\
&= \frac{1}{2\pi}e^{-\frac{1}{2}x^Tx} \\
&= \frac{1}{2\pi}e^{-\frac{1}{2}x_1^2+x_2^2} \\
&= \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x_1^2} \cdot \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x_2^2} \\
\theta &= \int_c^d \int_a^b f(x_1, x_2)dx_1dx_2 \\
&= (\int_0^1 \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}dx)^2
\end{aligned}
$$

```
theta_hat = mvn_mcintegration(0,1,0,1,c(0,0),diag(2),10000)
theta = (pnorm(1)-pnorm(0))^2
cbind(theta_hat,theta)
```

```
##      theta_hat     theta
## [1,] 0.1168281 0.1165162
```

The estimated value is pretty close to the real value

(c)

```
require('mvtnorm')
MC_mvtnorm <- function(a,b,c,d,mu,sigma,m){
  X<-matrix(rmvnorm(m,c(0,0),diag(2)),ncol=2)
  indicator_X<-as.numeric(X[,1]>a & X[,1]<b & X[,2]>c & X[,2]<d)
  indicator_X
}
```

(d)

```
mean(MC_mvtnorm(0,1,0,1,c(0,0),diag(2),10000))
```

```
## [1] 0.1158
```

Ex 5 (Rizzo 5.2)

The Standard Normal cdf is

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt.$$

Note that the standard normal distribution is symmetric about zero, thus we have $F(0) = 0.5$. Let $g(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$ and $f(t) = 1/x$. Let $\Phi(x) = \int_0^x g(t)dt$, $x \geq 0$.

When $x \geq 0$,

$$F(x) = 0.5 + \int_0^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = 0.5 + \int_0^x g(t)f(t)x dt = 0.5 + xE_f[\Phi(x)]$$

When $x < 0$,

$$F(x) = 1 - F(-x) = 0.5 - xE_f[\Phi(-x)]$$

```
std_ncdf<-function(x,m){
    set.seed(183)
    t<-runif(m,0,abs(x))
    gt<-(1/sqrt(2*pi))*exp(-t^2/2)
    return(0.5+x*mean(gt))
}
## Compare with pnorm()
x<-c(-2,-1.5,-1,-0.5,0,0.5,1,1.5,2)
m = 10000
comp_table<-rbind(pnorm(x),apply(matrix(x,ncol=1),1,std_ncdf,m))
rownames(comp_table)<-c("pnorm","Monte Carlo")
colnames(comp_table)<-as.character(x)
print(round(comp_table,4))
```

```
##                 -2   -1.5     -1   -0.5   0    0.5      1    1.5      2
## pnorm       0.0228 0.0668 0.1587 0.3085 0.5 0.6915 0.8413 0.9332 0.9772
## Monte Carlo 0.0206 0.0656 0.1582 0.3085 0.5 0.6915 0.8418 0.9344 0.9794
```

```
## The variance of estimate and 95% CI
set.seed(183)
x = 2
t<-runif(m,0,x)
```

```
gt<-(1/sqrt(2*pi))*exp(-t^2/2)
est<-0.5+x*mean(gt)
var<-abs(x)^2*var(gt)/m ## variance
CI<-c(est-1.96*sqrt(var),est+1.96*sqrt(var)) ## 95% CI
cat(paste("Variance:",round(var,4),sep=" "),
    paste("95% CI: ","[",round(CI[1],4),", ",round(CI[2],4),"]",sep=""),sep="\n")
```

```
## Variance: 0
## 95% CI: [0.9749, 0.9839]
```

Ex 6 (Rizzo 5.3)

```
#By sampling from U[0,0.5]
m = 10000
U = runif(m,0,0.5)
g = exp(-U)
theta_hat = mean(g)/2
var_hat = (0.5)^2*var(g)/m

#By sampling from exponential distribution
exp_samp <- function(n,lambda){
  x = rexp(n,lambda)
  gfphi = exp((lambda-1)*x)/lambda*(x<0.5)
  theta_exp = mean(gfphi)
  var_exp = var(gfphi)/n
  return(list(theta_exp,var_exp))
}
exp1 = exp_samp(m,1)
exp2 = exp_samp(m,2)
exp5 = exp_samp(m,5)
tab = rbind(theta_hat = c(theta_hat,exp1[[1]],exp2[[1]],exp5[[1]]),var_hat =
              c(var_hat,exp1[[2]],exp2[[2]],exp5[[2]]))
colnames(tab) = c('uniform(0,0.5)','exp(1)','exp(2)','exp(5)')
tab
```

```
##           uniform(0,0.5)      exp(1)       exp(2)       exp(5)
## theta_hat   3.940256e-01 3.92800e-01 3.919484e-01 3.956545e-01
## var_hat     3.256494e-07 2.38532e-05 9.585036e-06 7.793653e-06
```

We have tried several choices of lambda for exponential sampling but the uniform sampling still producing a smaller variance for estimating theta. The reason is that uniform(0,0.5) has a support on the same the interval of the integral.

Ex 7 (Rizzo 5.4)

The cdf of $\beta(a,b)$

$$F(x) = \int_0^x \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} t^{a-1}(1-t)^{b-1}dt$$

We substitute x by setting y=t/x, the equation is transformed as

$$F(y) = \int_0^1 \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}(xy)^{a-1}(1-xy)^{b-1}xdy$$

```
betaMC <-function (x, a = 1,b = 1, m = 1000000){
  u = runif(m,0,1)
  gx = factorial(a+b-1)/(factorial(a-1)*factorial(b-1))*(x*u)^(a-1)*(1-x*u)^(b-1)*x
  thetahat=mean(gx)
  thetahat
}
sv = numeric(l=9)
tv = numeric(l=9)
for (i in seq(.1,.9,.1)){
  sv[10*i] = betaMC(i,3,3)
  tv[10*i] = pbeta(i,3,3)
}
print(data.frame("Simulated" = sv,"Theoretical" = tv))
```

```
##     Simulated Theoretical
## 1 0.008556259     0.00856
## 2 0.057919329     0.05792
## 3 0.162879401     0.16308
## 4 0.317658207     0.31744
## 5 0.500103798     0.50000
## 6 0.682741806     0.68256
## 7 0.837175780     0.83692
## 8 0.942610070     0.94208
## 9 0.990774957     0.99144
```

Ex 8 (Rizzo 5.14)

The supports of some continuous random variable we use below have a support on $[1, \infty]$, so we have to shift them right a little bit manually.

```
# Solution 1(Shifted Exponential Distribution)
m = 1000000
x = rexp(m,1) + 1
gx1 = dnorm(x)*(x^2)
fx1 = dexp(x-1)
gfphi1 = gx1/fx1
theta_hat_1 = mean(gfphi1)
sd_hat_1 = sd(gfphi1)/sqrt(m)

# Solution 2(Shifted Gamma Distribution)
x2 = rgamma(m,3,1)+1
gx2 = x2^2/sqrt(2*pi)*exp(-x2^2/2)
fx2 = dgamma(x2-1,3,1)
gfphi2 = gx2/fx2
theta_hat_2 = mean(gfphi2)
sd_hat_2 = sd(gfphi2)/sqrt(m)

# Solution 3(Shifted Chi-squared Distribution)
x3 = rchisq(m,3)+1
```

```r
gx3 = x3^2/sqrt(2*pi)*exp(-x3^2/2)
fx3 = dchisq(x3-1,3)
gfphi3 = gx3/fx3
theta_hat_3 = mean(gfphi3)
sd_hat_3 = sd(gfphi3)/sqrt(m)

tab = rbind(theta_hat = c(theta_hat_1,theta_hat_2,theta_hat_3),sd_hat =
              c(sd_hat_1,sd_hat_2,sd_hat_3))
colnames(tab) = c('exp','gamma','chi-squared')
tab
```

```
##                     exp       gamma  chi-squared
## theta_hat 0.4006445839 0.394314572 0.4001399974
## sd_hat    0.0001575023 0.002564185 0.0005850642
```

Bonus

In this problem, we give a numerical estimate of the integral

$$\theta = \int_c^d \int_a^b f(x_1, x_2) dx_1 dx_2$$

where $a = 0$, $b = 1$, $c = 0$, $d = 1$, and $f(x_1, x_2)$ is the density function of the multivariate normal distribution with mean $\mu = (0, 0)^T$ and variance-covariance matrix $\Sigma = I$.

```r
#Multivariate Normal PDF
mvn.pdf<-function(x,mu,sigma){
(2*pi)^(-1)*det(sigma)^(-1/2)*exp(-0.5*t(x-mu)%*%solve(sigma)%*%(x-mu))
}
#True value of theta
theta = (pnorm(1) - pnorm(0))^2

mvn.numeric.intg <- function(a,b,c,d,k){
    #Equally distributed grid of k points
    xvals1 = seq(a,b, length.out = k)
    xvals2 = seq(c,d, length.out = k)
    midpoints1 = (xvals1[1:k-1] + xvals1[2:k])/2
    midpoints2 = (xvals2[1:k-1] + xvals2[2:k])/2
    delta = xvals1[2] - xvals1[1]
    midpoints = cbind(rep(midpoints1,k-1),matrix(apply(as.matrix(midpoints2),1,rep,k-1),ncol = 1))
    fx = apply(midpoints, MARGIN = 1, FUN = mvn.pdf, mu = c(0,0), sigma = diag(2))
    theta_hat = sum(fx*delta^2)
    return(theta_hat)
}
```