# Statistical Computing - Midterm

Wei-Chen Chang r12227118

2024-04-10

Consider the estimation of

$$\theta = \int_1^\infty \frac{x^2}{\sqrt{2\pi}} \exp(\frac{-x^2}{2}) dx$$

Using the following methods to estimate $\theta$ and make a comparison of them. For each method, drawing a plot to show the convergence of the approximation.

1. Direct Monte Carlo method (standard normal distribution as the sampling distribution).
2. Importance sampling method and the self-normalized importance sampling methods. Is there any difference between two methods? Which one is an unbiased estimator?
3. Variable transformation of the integrand to a interval of $[0, 1]$
4. Two control variates.
5. Stratified Sampling (Divide the range of integration into n equal probable segments). Discuss the effect of $n$.
6. Use the tile density of standard normal distribution as the importance function given by

$$f_t(x) = \frac{e^{tx}\phi(x)}{M(t)},$$

   where $\phi$ is the standard normal distribution and $M(t)$ is the moment generating function of $\phi$. Discuss the effect of $t$, $-\infty < t < \infty$.

## Answer:

In the following simulation, for each estimator, total $N = 1500$ MC samples were generated and repeated $r = 3$ times. This enable us to visualize the convergence and efficient between different estimators in the trace plot.

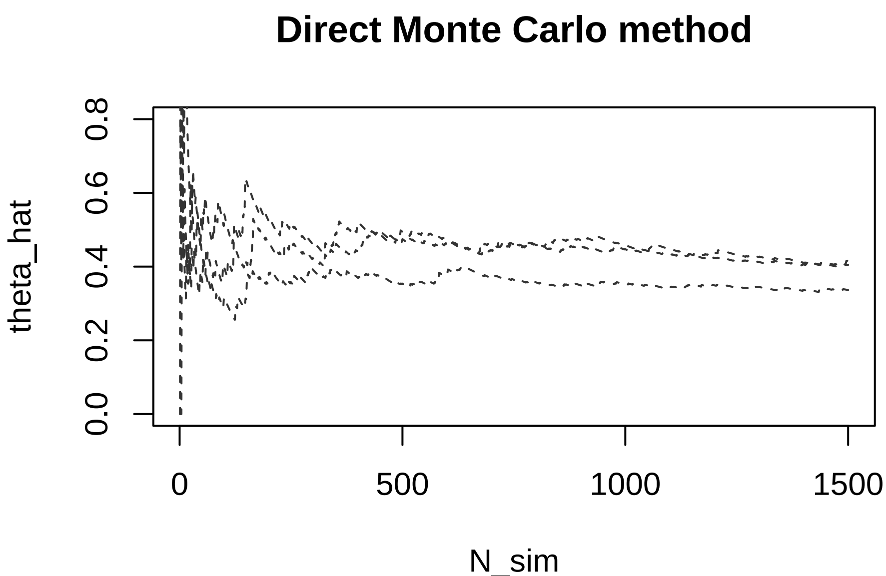
### 1. Direct Monte Carlo

let $\phi(x)$ be the pdf of standard normal distribution, $\theta$ can be rewrite as:

$$\begin{aligned}\theta &= \int_1^\infty x^2\phi(x)dx \\ &= \int_{-\infty}^\infty I_{\{x\geq 1\}} x^2 \phi(x) dx \\ &= E_\phi[I_{\{X\geq 1\}}X^2]\end{aligned}$$

And the MC estimator is: $\hat{\theta} = \frac{1}{n}\Sigma_{i=1}^n I_{\{X_i\geq 1\}} X_i^2$, where $X_i \sim N(0, 1)$.

```r
set.seed(.seed)
for (i in 1:r) {
  MC_samp <- rnorm(N)
  MC_samp <- ifelse(MC_samp>=1, MC_samp^2, 0) # indicator and return X^2
  if (i==1){
    plot(x= 1:length(MC_samp), y= cumsum(MC_samp)/1:length(MC_samp), type = "l",
         xlab = "N_sim", ylab = "theta_hat", ylim = c(0,.8), lty=2, col = "grey20",
         main= "Direct Monte Carlo method")
  }else{
    lines(x= 1:length(MC_samp), y= cumsum(MC_samp)/1:length(MC_samp), lty=2, col = "grey20")
  }
}
```




## 2. Importance Sampling and Self-Normalized Importance Sampling

Choose half-normal density shifted to start at 1 as importance function $f(x)$:

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{2\pi}}\exp(\frac{-(x-1)^2}{2})/\int_1^\infty \frac{1}{\sqrt{2\pi}}\exp(\frac{-(x-1)^2}{2})dx \\ &= \frac{2}{\sqrt{2\pi}}\exp(\frac{-(x-1)^2}{2}), \end{aligned}$$

And $\theta$ and its estimator $\hat{\theta}_{imp}$ can be expressed as:

$$\begin{aligned} \theta &= \int_1^\infty x^2[\frac{\phi(x)}{f(x)}]f(x)dx \\ &= E_f[X^2\frac{\phi(X)}{f(X)}], \\ \hat{\theta}_{imp} &= \frac{1}{n}\Sigma_{i=1}^n X_i^2\frac{\phi(X_i)}{f(X_i)}, \text{where } X_i \sim f. \end{aligned}$$

$X_i$ can be generated by the following relationship: $X = |Z| + 1$, where $Z \sim N(0, 1)$.

**Self-Normalized Importance Sampling Estimator:**

We wish to find some $C$ such that $\int_1^\infty \frac{\phi(x)}{C}dx = 1$, then the integral can be self normalized. One can solve that $C = 1 - \Phi(1)$, where $\Phi$ is the cdf of standard normal distribution. Thus,

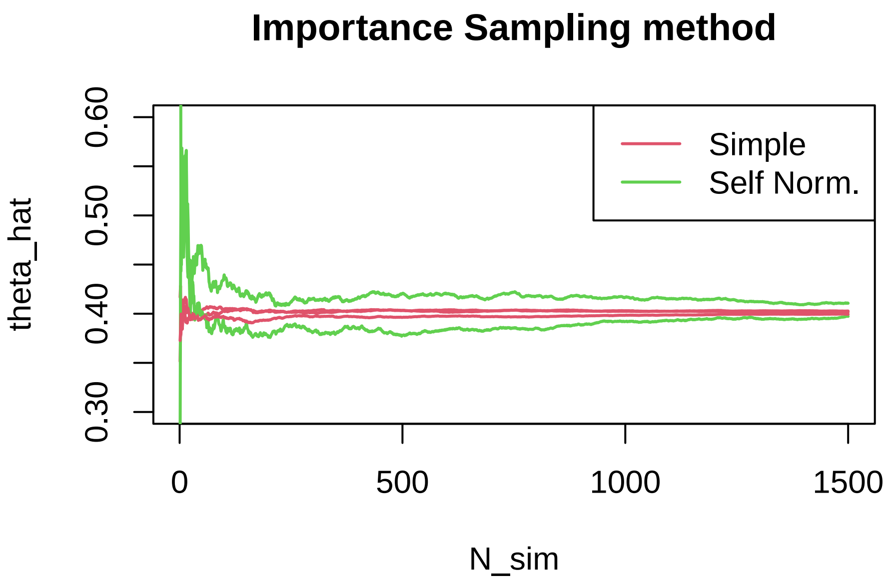
$$\begin{aligned}\theta &= C\int_1^\infty x^2 \frac{\phi(x)}{Cf(x)} f(x)dx \\ &= C\int_1^\infty x^2 \frac{\phi(x)}{Cf(x)} f(x)dx / \int_1^\infty \frac{\phi(x)}{Cf(x)} f(x)dx,\end{aligned}$$

and the Self-Normalized Estimator is:

$$\hat{\theta}_{SN} = C\,\Sigma_{i=1}^n X_i^2 [\frac{\phi(X_i)}{f(X_i)} / \Sigma_{j=1}^n \frac{\phi(X_j)}{f(X_j)}], \text{ where } X_i \sim f$$

Note this is a biased estimator.

```
set.seed(.seed)
hNorm <- function(x){
  return( ifelse(x>=1, 2/sqrt(2*pi)*exp(-(x-1)^2/2), 0))
}
target <-  function(x){
  return(ifelse(x>=1, x^2*dnorm(x) ,0))
}
for (i in 1:r){
  hNorm_samp <- abs(rnorm(N))+1
  import_samp <- target(hNorm_samp)/hNorm(hNorm_samp)
  #self-normalized
  weight <- dnorm(hNorm_samp)/hNorm(hNorm_samp)
  SN_samp <-  pnorm(1, lower.tail = F)* cumsum(hNorm_samp^2*weight)/cumsum(weight)
  if (i==1){
    plot(x= 1:N, y= cumsum(import_samp)/1:N, type = "l", lwd=1.5, col = 2,
         xlab = "N_sim", ylab = "theta_hat", ylim = c(.3,.6),
         main= "Importance Sampling method")
  }else{
    lines(x= 1:N, y= SN_samp, lwd=1.5,col = 3)
    lines(x= 1:N, y= cumsum(import_samp)/1:N, lwd=1.5,col = 2)
  }
}
legend("topright", c("Simple", "Self Norm."), col = c(2,3), lwd=1.5, lty=1)
```




Surprisingly, the self-normalized estimator performs worse than simple one.

## 3. Variable Transformation

Consider such variable transformation: $u = \frac{1}{x}$, then $du = -x^{-2}dx$, the integral turns into:

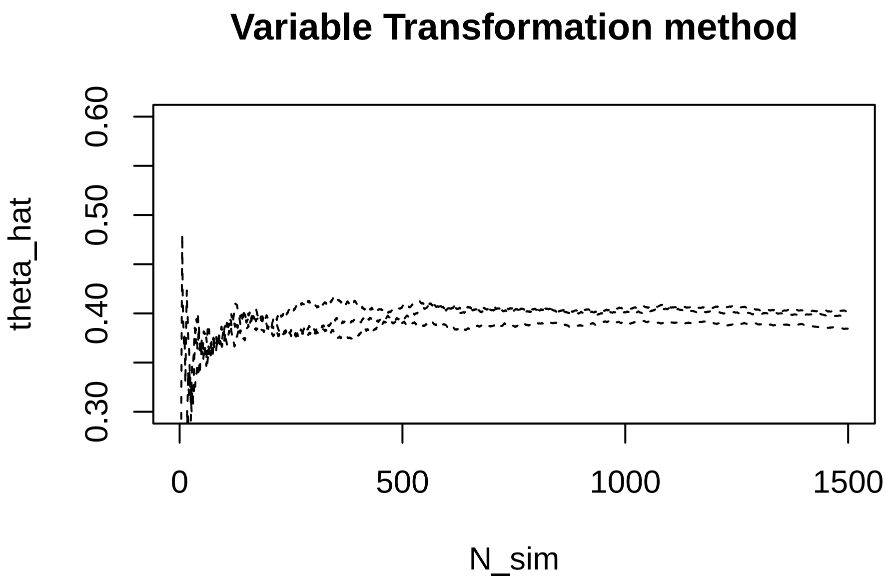
$$
\begin{aligned}
\theta &= \int_1^\infty \frac{x^2}{\sqrt{2\pi}} \exp(\frac{-x^2}{2})dx \\
&= \int_0^1 \frac{1}{u^4\sqrt{2\pi}} \exp(\frac{-1}{2u^2})du \\
&= E_U[\frac{1}{u^4\sqrt{2\pi}} \exp(\frac{-1}{2u^2})] \\
&= E_U[g(u)], \text{ where } U \sim \text{Unif}(0,1).
\end{aligned}
$$

And its MC estimator $\hat{\theta}_{vt}$ is:

$$\hat{\theta}_{vt} = \frac{1}{n}\Sigma_{i=1}^n g(U_i), \text{ where } U_i \sim \text{Unif}(0,1)$$

```
# N=1000
set.seed(.seed)
target_tf <- function(u){
  return(1/(u^4*sqrt(2*pi))*exp(-1/(2*u^2)))
}
for(i in 1:r) {
  uf <- runif(N)
  vtran_samp <- target_tf(uf)
  if(i==1){
    plot(x= 1:N, y= cumsum(vtran_samp)/1:N, type = "l", lty = 2,
         xlab = "N_sim", ylab = "theta_hat", ylim = c(.3,.6),
```

```
          main= "Variable Transformation method")
  }else{
    lines(x= 1:N, y= cumsum(vtran_samp)/1:N, lty = 2)
    }
}
```




### 4. 2 Control Variates

Following the the variable transformation and its MC estimator $\hat{\theta}_{vt} = E[g(u)]$, here we choose two control variates:

$$V_1 \sim \text{Unif}(0,1),\ V_2 = (V_1 - \frac{1}{2})^2.$$

Both expectation can be easily derived ($E(V_1) = \frac{1}{2}$, $E(V_2) = \frac{1}{12}$) and they can be generated by MC samples of $g(U_i)$. Furthermore, these two covariates should have low(linear) correlation. This make coefficient estimation in multiple regression estimation feasible.

Thus the Control Variate Estimator $\hat{\theta}_{cv}$ is :

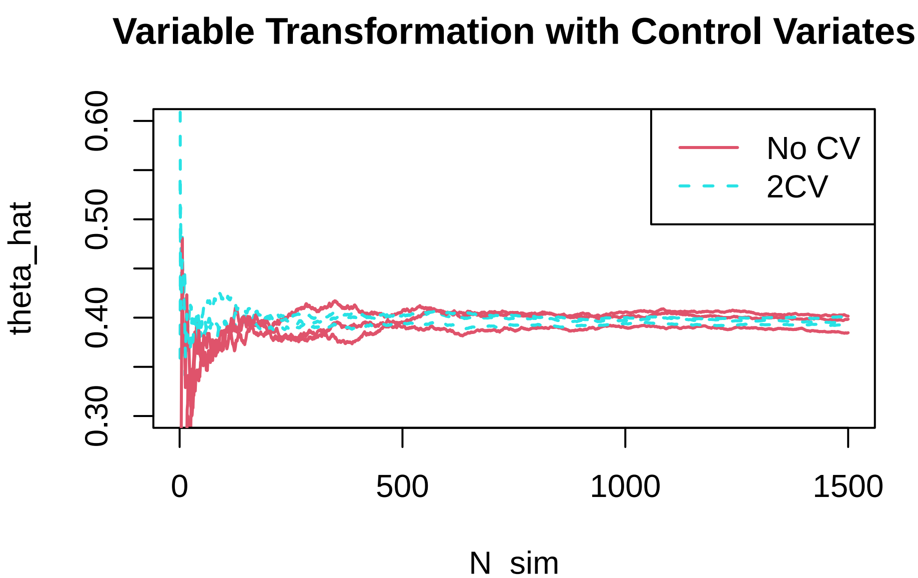
$$\hat{\theta}_{cv} = \frac{1}{n}\Sigma_{i=1}^{n}\{g(U_i) + c_1(U_i - \frac{1}{2}) + c_2[(U_i - \frac{1}{2})^2 - \frac{1}{12}]\},$$

where $U_i \sim \text{Unif}(0,1)$.

$c_1, c_2$ can be estimated by the coefficients regressing $g(U_i)$ on both $U_i$ and $(U_i - \frac{1}{2})^2$.

```
# N=1000
set.seed(.seed)
for (i in 1:r){
  v1 <- runif(N)
  gv1 <- target_tf(v1)
```

```r
  v2 <- (v1-.5)^2
  c <- lm(gv1~v1+v2)$coef
  cv_samp <- gv1 - c[2]*(v1-.5)-c[3]*(v2-(1/12))
  # cv1_samp <- gv1 - lm(gv1~v2)$coef[2]*(v2-(1/12))
  if (i==1){
    plot(x= 1:N, y= cumsum(gv1)/1:N, type = "l", col=2, lwd=1.5,
         xlab = "N_sim", ylab = "theta_hat", lty = 1, ylim = c(.3,.6),
         main= "Variable Transformation with Control Variates")
  }else{lines(x= 1:N, y= cumsum(gv1)/1:N, col=2, lty=1, lwd=1.5)
    }
  lines(x= 1:N, y= cumsum(cv_samp)/1:N,col=5, lty=2, lwd=1.5)
  # lines(x= 1:N, y= cumsum(cv1_samp)/1:N,col=7, lty=2, lwd=1.5)
}
legend("topright", c("No CV", "2CV"), col = c(2,5), lwd=1.5, lty=c(1,2))
```




Note that `No CV` (red line) traces is just the variable transformed MC estimator in 3.

## 5. Stratified Sampling

As the integral range is unbounded, which is hard to divided into stratifies.
Here apply the variable transformation in 3. first, setting the integral range to $[0, 1]$, then divided it in to $n$ stratifies.

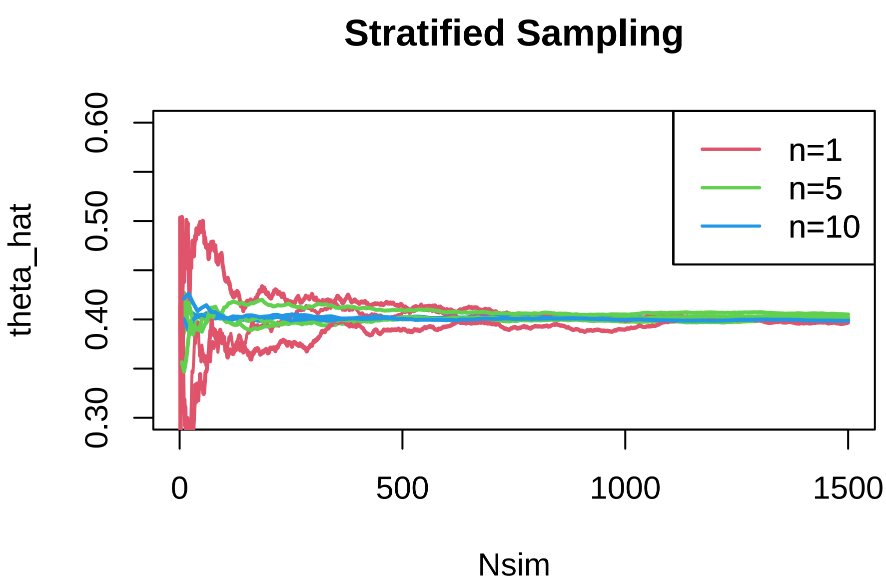
```r
# N = 1000
set.seed(.seed)
n = c(1, 5, 10)
stra_samp = matrix(0, N, 3)
for (i in 1:r)  {
  for (j in 1:length(n)){
    t <- c()
```

```r
    for (k in 1:n[j]) {
      .t <- target_tf(runif(N/n[j], (k-1)/n[j], k/n[j]))
      t <- c(t, .t)
    }
    stra_samp[, j] <- t
  }

  str5 <- stra_samp[,2] %>% # 5 stratifies
    matrix(ncol = n[2]) %>%
    apply(MARGIN = 2, FUN = cumsum) %>%  # each stratifies has N/n sample
    rowMeans() %>%
    {./1:(N/n[2])}

  str10 <- stra_samp[,3] %>% # 10 stratifies
    matrix(ncol = n[3]) %>%
    apply(MARGIN = 2, FUN = cumsum) %>%  # each stratifies has N/n sample
    rowMeans()  %>%
    {./1:(N/n[3])}

  if(i==1){
    plot(1:N, cumsum(stra_samp[,1])/1:N, type ="l",lty=1, col = 2,
         ylab = "theta_hat", xlab = "Nsim", ylim = c(.3,.6),lwd=1.5,
         main = "Stratified Sampling")
  }else{
    lines(1:N, cumsum(stra_samp[,1])/1:N,lty=1, col = 2,lwd=2)
  }
  lines(seq(n[2],N, by=n[2]), str5, lty=1, col = 3,lwd=2)
  lines(seq(n[3],N, by=n[3]), str10, lty=1, col = 4,lwd=2)
  legend("topright", paste0("n=",n), col = 2:4,lty=1, lwd=1.5)
}
```



Note that when n=1, the estimator is just variable transformed MC estimator in 3., but with different MC sample. As n increases, the estimator converges faster, more efficient.

## 6. Choose $f_t(x)$ as Importance Function

First note that $M(t) = e^{\frac{t^2}{2}}$, thus $f_t(x) = e^{t(x-\frac{t}{2})}\phi(x)$. One can see its is just pdf of $N(t, 1)$.

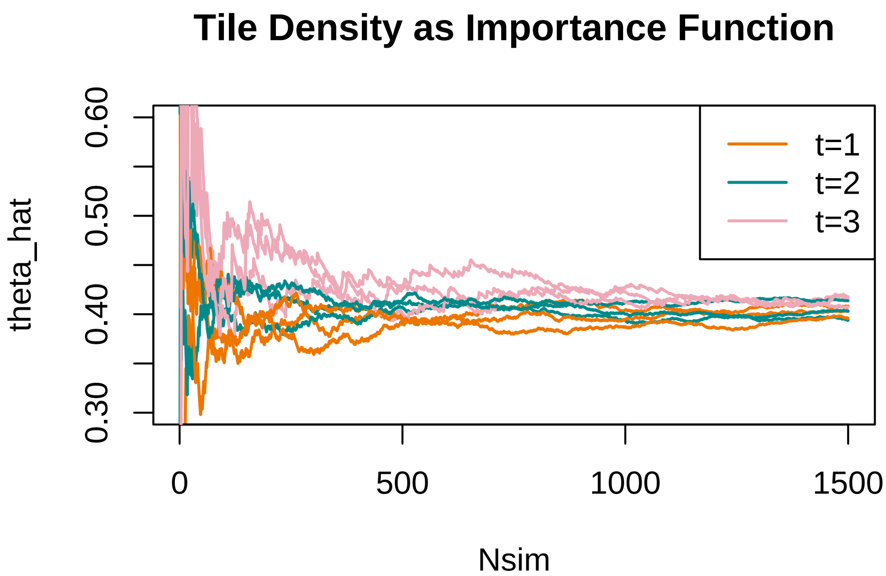
So $\theta$ can be rewrite as:

$$\begin{aligned}
\theta &= \int_1^\infty x^2 \frac{\phi(x)}{f_t(x)} f_t(x) dx \\
&= \int_1^\infty x^2 e^{-t(x-\frac{t}{2})} f_t(x) dx \\
&= \int_{-\infty}^\infty I_{x\geq 1} x^2 e^{-t(x-\frac{t}{2})} f_t(x) dx. \\
\hat{\theta}_t &= \frac{1}{n}\Sigma_{i=1}^n h(X_i),
\end{aligned}$$

$$\text{where } X_i \sim f_t(x),\ h(x) = I_{x\geq 1} x^2 e^{-t(x-\frac{t}{2})}.$$

```
# N = 5000
set.seed(.seed)
t.val = 1:3
.col = c("darkorange2","darkcyan","pink2")
for (i in 1:r){
  samp.mat <- matrix(rnorm(N*length(t.val), mean = t.val),
                     nrow =N, ncol=length(t.val), byrow = T) # gen f_t samples

  for (j in 1:length(t.val)){
   xi <- samp.mat[,j]
   samp.mat[,j] <- target(xi)/dnorm(xi, mean = t.val[j]) #x^2phi/f_t(x)
   if (i==1 & j==1){
     plot(1:N, cumsum(samp.mat[,j])/1:N, lwd=1.5, type = "l", col = .col[j],
          ylab = "theta_hat", xlab = "Nsim", ylim = c(.3,.6), lty= 1,
          main = "Tile Density as Importance Function")
   }else{
     lines(1:N, cumsum(samp.mat[,j])/1:N, lwd=1.5, lty= 1, col = .col[j])
   }
  }
}
legend("topright", paste0("t=",t.val), col = .col,lwd=1.5, lty= 1)
```




One can see when t =1,2 the estimate performs well (set t =2 even better than t=1), but when t=3 the estimate start to become less efficient. It may related to these factors: (1) the shape of target function (2) the range of integral.

## Brief Summary and Comparison Between Estimators

The Table below shows the mean and variance of each estimator.
% latex table generated in R 4.2.3 by xtable 1.8-4 package % Tue Apr 9 16:37:41 2024

| | est | var |
|---:|---|---|
| SimpMC | 0.4155 | 1.446 |
| ImpSamp | 0.3995 | 0.0019 |
| ImpSamp(SN) | 0.3973 | NA |
| VarTran | 0.3847 | 0.0901 |
| CV | 0.3925 | 0.0235 |
| Stra(n=1) | 0.4006 | 0.0948 |
| Stra(n=5) | 0.4003 | 0 |
| Stra(n=10) | 0.4009 | 0 |
| TDens(t=1) | 0.3959 | 0.1634 |
| TDens(t=2) | 0.403 | 0.1068 |
| TDens(t=3) | 0.4072 | 0.6013 |

Table 1: Summary of MC Estimators