2019-11-13

computational Statistics

HW#7

192STG11 우나영

computational Statistics

HW#7

1. Problem

Monte Carlo(MC) simulation 은 target 함수에서 iid 한 샘플을 뽑아 통계량을 계산하는 기법이다. 우리의 목표는 $X_1, \dots, X_n$이 target 함수 f 를 iid 로 따를 때 함수 h(x)의 expected value 를 구하는 것이다. 예를 들면 MC 방법을 사용하여 아래와 같이 구할 수 있다.

$$\widehat{\mu_{MC}} = \frac{1}{n}\sum_{i=1}^{n} h(X_i) \approx E[h(X)]$$

이때 두 가지 고려해야할 이슈가 발생한다. 첫 번째, target 함수 f 의 iid 샘플을 뽑는 방법이다. 이와 같은 문제를 해결하는 알고리즘은 지난번 과제에서 다루었다. 두 번째, 위와 같은 간단한 estimator 보다 더 정확한 estimator 을 구하는 방법이다. 이때, 정확도(precision)의 통계량은 분산의 역수로 분산이 작을수록 정확도가 높아지기 때문에 더 정확한 estimator 란 분산이 더 작은 estimator 를 말한다. 위와 같은 간단한 MC estimator 를 Naïve MC 라고 한다. 본 과제에서는 Naïve MC 보다 분산을 작게 만드는 Importance Sampling MC 와 Antithetic MC 을 소개하고자 한다.

Importance Sampling 은 원래의 density f 를 따르는 X_i가 굉장히 rare 한 event 인 경우 ISD(Importance Sampling Distribution) g 에서 X_i를 oversample 하여 Importance Weights(IW)로 bias 를 교정해줌으로써 estimator 의 분산을 줄일 때 유용한 방법이다. 아래는 Importance Sampling 의 아이디어를 수식으로 정리한 것이다.

$$\mu = \int h(x)f(x)dx = \int h(x)\frac{f(x)}{g(x)}g(x)dx \quad -(*)$$

$$where\ \ w^*(x_i) = \frac{f(x_i)}{g(x_i)} \quad \widehat{\mu_{IS}^*} = \frac{1}{n}\sum h(x_i)w^*(x_i)$$

$$\text{or } \mu = \frac{\int h(x)f(x)dx}{\int f(x)dx} = \frac{\int h(x)\frac{f(x)}{g(x)}g(x)dx}{\int \frac{f(x)}{g(x)}g(x)dx} \quad -(**)$$

$$where\ w(x_i) = \frac{w^*(x_i)}{\sum w^*(x_i)} \quad \widehat{\mu_{IS}} = \sum h(x_i)w(x_i)$$

원래의 분포 f 대신 g 라는 ISD 에서 X_i 에서 sample 을 생성하여 원래는 rare 하게 발생하는 event 의 수를 oversample 한다. 이때 g 는 아래의 두가지 조건을 만족해야 한다.

$$(1)\ supp(g) \geq supp(f)$$

$$(2)\ \frac{f}{g} bounded\ and\ g\ have\ heavier\ tail\ than\ f$$

g 에서 샘플을 생성해 발생하는 bias 를 IW 로 교정해주는데 (*)은 unstandardized weight 그리고 (**)는 standardized weight 을 각각 사용한 경우이다.

다음으로 Antithetic MC 는 동일한 분포를 따르지만 negative correlation 을 보여주는 두 개의 unbiased estimator $\widehat{\mu_1}$과 $\widehat{\mu_2}$의 mean 이다.

$$\widehat{\mu_{AS}} = \frac{\widehat{\mu_1} + \widehat{\mu_2}}{2}$$

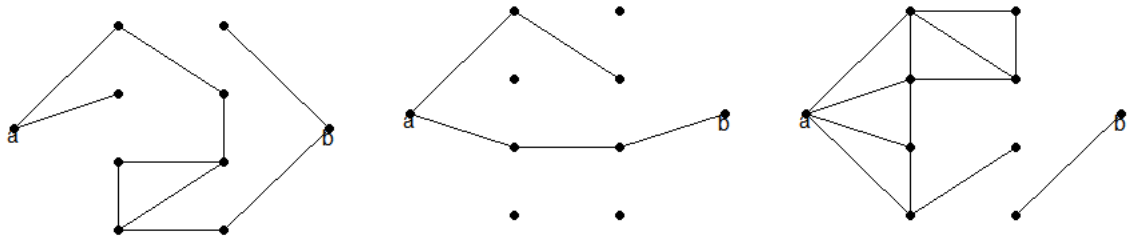
따라서 $\widehat{\mu_{AS}}$의 분산은

$$var(\widehat{\mu_{AS}}) = \frac{1}{4}\left(var(\widehat{\mu_1}) + var(\widehat{\mu_2})\right) + \frac{1}{2}cov(\widehat{\mu_1}, \widehat{\mu_2}) = \frac{(1+\rho)\sigma^2}{2n}$$

이때 ρ는 $\widehat{\mu_1}$과 $\widehat{\mu_2}$의 correlation 이다. $\widehat{\mu_1}$과 $\widehat{\mu_2}$가 negative correlated 이기 때문에 Antithetic MC 의 분산은 Naïve MC w/ 2n sample size 보다 작다. 이때 Naïve MC 의 분산은

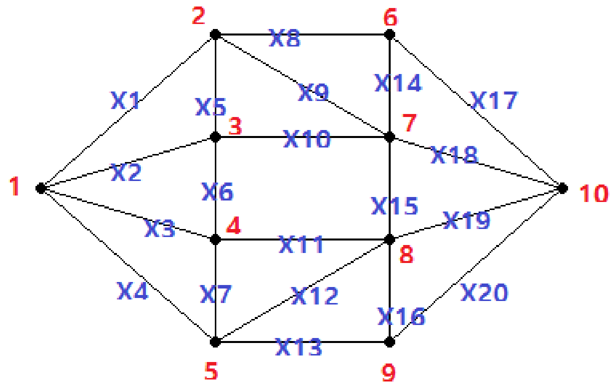
$$var(\widehat{\mu_{MC}}) = var\left(\frac{1}{2n}\sum h(x_i)\right) = \frac{1}{4n^2}2n\sigma^2 = \frac{\sigma^2}{2n}$$

본 과제에서는 Naïve MC 와 비교해 Importance Sampling, Antithetic MC 그리고 Mixture of Importance Sampling and Antithetic MC 가 얼마나 분산을 줄일 수 있는지 다음의 Network Failure 예제를 통해 알아볼 것이다.



많은 시스템들이 위의 그림과 같이 연결된 그래프 형태를 띈다. 이러한 그래프들은 노드(점)과 edge(선)으로 구성 되어있다. a 에서 시작한 신호가 b 까지 도달하기 위해서는 a 에서 b 까지 연결된 edge 만 있으면 가능하다. 하지만 세번째 그림의 경우 a 와 b 사이를 연결하는 중요한 edge 들이

broken 되어 신호가 도달하는 것에 fail 한 경우를 보여준다. 이와 같은 Network 시스템에서 중요한 것은 각각의 edge 가 broken 될 확률이 주어질 때 Network 가 fail 할 확률을 계산하는 것이다. 본 과제에서는 아래의 그림과 같이 10 개의 노드와 20 개의 edge 로 구성된 Network failure 를 MC 를 이용해 구하볼 것이다.



$$X = (X_{i,1}, \dots, X_{i,20}) : state\ of\ edge\ fail\ or\ not\ if\ fail\ X_{i,m} = 1$$

$$where\ i = 1, \dots, n\ \ and\ m = 1, \dots, 20$$

$$\mathrm{b(X)} = \text{\# of broken edges in X}$$

$$\mathrm{h(X)} = \begin{cases} 1 & if\ a\ \ and\ b\ disconnected \\ 0 & o.w. \end{cases}$$

$$[E(h(x)] = \mu\ \ which\ is\ the\ probability\ of\ network\ failure$$

Naïve MC

$\widehat{\mu_{MC}} = \frac{1}{n}\sum h(x_i)\ \ var(\widehat{\mu_{MC}}) = \frac{1}{n^2}n\mu(1-\mu) = \frac{\mu(1-\mu)}{n}$

Importance Sampling MC with $p^* > p$

Unstandardized IW = $w^*(X_i^*) = \frac{p^{b(X_i^*)}(1-p)^{20-b(X_i^*)}}{p^{*b(X_i^*)}(1-p^*)^{20-b(X_i^*)}} = \left(\frac{1-p}{1-p^*}\right)^{20} \left(\frac{p(1-p^*)}{p^*(1-p)}\right)^{b(X_i^*)}$

$\widehat{\mu_{IS}} = \frac{1}{n}\sum h(x_i^*)w^*(x_i^*)\ \ \ \mathrm{var}(\widehat{\mu_{IS}}) = \frac{1}{n}var\left(h(x_i^*)w^*(x_i^*)\right) = \frac{1}{n}\left(\sum_{X\in F} w^*(x)p^{b(x)}(1-p)^{20-b(x)} - \mu^2\right)$

Antithetic MC

$\mathrm{U} = (U_{i,1}, \dots, U_{i,20}) \sim unif(0,1)\ \ i = 1, \dots, n$

$$\begin{cases} X_i = 1 \;\; if \; U_i < p \\ X_i^* = 1 \;\; if \; U_i > 1 - p \end{cases} \; where \; 1 \; means \; broken \; edge$$

$$\widehat{\mu_{AS}} = \frac{\sum h(X_i) + h(X_i^*)}{2}$$

이때 X_i^*는 X_i를 생성할 때 사용한 U_i를 이용해 샘플링해야 negative correlation 을 만족한다.

<u>Mixture of Importance Sampling and Antithetic MC with $p^* > p$</u>

U $= (U_{i,1}, \dots, U_{i,20}) \sim unif(0,1) \;\; i = 1, \dots, n$

$$\begin{cases} X_i = 1 \;\; if \; U_i < p^* \\ X_i^* = 1 \;\; if \; U_i > 1 - p^* \end{cases} \; where \; 1 \; means \; broken \; edge$$

Unstandardized IW = $w^*(X_i^*) = \frac{p^{b(X_i^*)}(1-p)^{20-b(X_i^*)}}{p^{*b(X_i^*)}(1-p^*)^{20-b(X_i^*)}} = \left(\frac{1-p}{1-p^*}\right)^{20} \left(\frac{p(1-p^*)}{p^*(1-p)}\right)^{b(X_i^*)}$

$$=\mathrm{w}(X_i) = \frac{p^{b(X_i)}(1-p)^{20-b(X_i)}}{p^{*b(X_i)}(1-p^*)^{20-b(X_i)}} = \left(\frac{1-p}{1-p^*}\right)^{20} \left(\frac{p(1-p^*)}{p^*(1-p)}\right)^{b(X_i)}$$

$$\widehat{\mu_{MIX}} = \frac{\sum h(X_i w_i) + h(X_i^* w_i^*)}{2}$$

Antithetic 과 Mixture 의 분산의 경우 알려진 공식을 사용하지 않고 simulation 을 통해 구한 estimator 들의 분산으로 계산하였다.

2. Result

<u>Naive MC vs Importance Sampling</u>

# of samples=100,000 p=0.05	Naïve	IS with $p^* = 0.25$	IS with $p^* = 0.2$
$\hat{\mu}$	1e-05	7.7e-06	7.6e-06
$var(\hat{\mu})$	9.9e-11	9.4e-12	7.2e-12
# of failures	1	1320	476
$var(\hat{\mu}_{MC})/var(\hat{\mu}_{IS})$		10.64	13.89

Importance Sampling 은 rare 한 event 를 ISD 로 oversampling 한 후 IW 로 correction 해주는 방법이다. 따라서 원래의 density 에서 뽑은 failure 수는 100,000 번 중 1 번인 반면 ISD 로 뽑은 failure 수는 p 가 0.25 일 때 1320 번이고 p 가 0.2 일 때 476 번으로 커진다. 이때 p 는 edge 가 independent 하게 broken 될 확률이다. IS MC 의 variance 는 Naïve MC 의 variance 보다 p 가 0.25 일 때 10.64 배 줄고 p 가 0.2 일 때 13.89 배 줄어든다.

Naïve MC Sampling vs Antithetic Sampling vs Mixture of Antithetic and Importance Sampling

n=100,000 # of simulation = 10	Naïve	Antithetic	Mixture with $p^* = 0.2$
1	0e+00	1e-05	5.2e-06
2	1e-05	1e-05	8.3e-06
3	2e-05	0e+00	6.7e-06
4	0e+00	3e-05	6.9e-06
5	3e-05	2e-05	1.2e-05
6	0e+00	1e-05	8.7e-06
7	1e-05	2e-05	1.0e-05
8	0e+00	0e+00	8.8e-06
9	2e-05	1e-05	8.6e-06
10	1e-05	1e-05	9.3e-06
$var(\hat{\mu})$	1.1e-10	8.4e-11	3.4e-12
$var(\hat{\mu}_{MC})/var(\hat{\mu}_{IS\ or\ Mix})$		1.32	32.73

uniform 분포에 n 개의 표본을 생성해 $X_i = 1\ if\ U_i < p$ 으로 샘플링하는 방법으로 10 번을 반복하여 10 개의 Naïve MC 를 구했다. Antithetic 은 uniform 분포의 n/2 개의 표본에서 negative correlation 을 만족하는 X_i를 샘플링하는 것을 10 번 반복하여 10 개의 AS MC 를 구했다. Mixture 는 uniform 분포의 n/2 개의 표본에서 negative correlation 을 만족하면서 동시에 ISD 을 따르는 X_i를 샘플링하는 것을 10 번 반복하여 10 개의 Mixture MC 를 구했다. AS MC 의 variance 는 Naïve MC 의 variance 보다 1.32 배 줄어들어 거의 차이가 없음을 확인할 수 있다. 반면에 Mixture MC 의 variance 는 Naïve MC 의 variance 보다 p^*가 0.2 일 때 32.73 배 줄어들어 굉장히 효과적으로 estimator 의 정확도를 높인다는 것을 확인할 수 있다.

3. Discussion

이번 과제에서는 MC 의 두 번째 이슈인 분산을 줄이는 방법으로 두 가지 Importance Sampling 과 Antithetic 을 다루었다. 분산의 역수는 정확도의 통계량으로 분산이 작을수록 estimator 의 정확도가 높아진다. 따라서 기존의 간단한 Naïve MC 보다 정확도가 더 높은 estimator 를 계산하는 다양한 방법이 연구되었다. 그중 Importance Sampling 은 rare 한 event 를 sampling 할 때 원래의 f 분포 대신 ISD 분포를 따르는 표본을 생성하여 oversampling 한 후 IW 로 bias 를 correction 해주는 방식으로 분산을 줄인다. 다음으로 Antithetic 은 동일한 분포를 따르지만 negative correlated 한 두 개의 estimator 의 평균을 취하는 방법으로 분산을 줄인다.

Network failure 의 확률을 계산하는 예제를 통해 두 가지 방법을 구현해 보았다. Importance Sampling 은 분산을 10 배 정도 줄이지만 Antithetic 은 예상했던 것보다 분산을 줄이지 못했다. 하지만

Importance Sampling 과 Antithetic 방법을 mix 한 방법으로 estimator 를 구했을 때 30 배 이상 분산이 줄어들어 분산을 줄이는데 매우 효과적이라는 것을 알 수 있었다.

4. Appendix

```
pacman::p_load(ggplot2, tidyverse, dplyr, plotly, processx)
rm(list=ls(all=TRUE))

## Calculating Expected Value of Network Failure

## Check function to check if a network fails or not(엄수연선배님코드)
## 1 - connected 0 - disconnected : 교재와 반대
myroute<-function(x){
  m<-matrix(c(    1,x[1],x[2],x[3],x[4],0,0,0,0,0,
               x[1],1,x[5],0,0,x[8],x[9],0,0,0,
               x[2],x[5],1,x[6],0,0,x[10],0,0,0,
               x[3],0,x[6],1,x[7],0,0,x[11],0,0,
               x[4],0,0,x[7],1,0,0,x[12],x[13],0,
               0,x[8],0,0,0,1,x[14],0,0,x[17],
               0,x[9],x[10],0,0,x[14],1,x[15],0,x[18],
               0,0,0,x[11],x[12],0,x[15],1,x[16],x[19],
               0,0,0,0,x[13],0,0,x[16],1,x[20],
               0,0,0,0,0,x[17],x[18],x[19],x[20],1),ncol=10,byrow=T)
  res<-ifelse((m%*%m%*%m%*%m%*%m%*%m%*%m%*%m%*%m)[1,10]>0,1,0)
  return(res)
}

myplot<-function(x){
  plot(c(0,1,1,1,1,2,2,2,2,3),c(0,1.5,0.5,-0.5,-1.5,1.5,0.5,-0.5,-
1.5,0),axes=FALSE,xlab="",ylab="",pch=16)
  text(0,-0.1,"a")
  text(3,-0.1,"b")
  if(x[1]==1){lines(c(0,1),c(0,1.5))}
  if(x[2]==1){lines(c(0,1),c(0,0.5))}
  if(x[3]==1){lines(c(0,1),c(0,-0.5))}
  if(x[4]==1){lines(c(0,1),c(0,-1.5))}
  if(x[5]==1){lines(c(1,1),c(1.5,0.5))}
  if(x[6]==1){lines(c(1,1),c(0.5,-0.5))}
  if(x[7]==1){lines(c(1,1),c(-0.5,-1.5))}
  if(x[8]==1){lines(c(1,2),c(1.5,1.5))}
  if(x[9]==1){lines(c(1,2),c(1.5,0.5))}
  if(x[10]==1){lines(c(1,2),c(0.5,0.5))}
  if(x[11]==1){lines(c(1,2),c(-0.5,-0.5))}
  if(x[12]==1){lines(c(1,2),c(-1.5,-0.5))}
  if(x[13]==1){lines(c(1,2),c(-1.5,-1.5))}
  if(x[14]==1){lines(c(2,2),c(1.5,0.5))}
  if(x[15]==1){lines(c(2,2),c(0.5,-0.5))}
  if(x[16]==1){lines(c(2,2),c(-0.5,-1.5))}
  if(x[17]==1){lines(c(2,3),c(1.5,0))}
  if(x[18]==1){lines(c(2,3),c(0.5,0))}
  if(x[19]==1){lines(c(2,3),c(-0.5,0))}
  if(x[20]==1){lines(c(2,3),c(-1.5,0))}
}
```

```
##############################################
myroute(c(1,1,0,0,0,0,1,0,1,0,1,1,1,0,1,0,1,0,0,1))
myplot(c(1,1,0,0,0,0,1,0,1,0,1,1,1,0,1,0,1,0,0,1))
myroute(c(1,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,1,0))
myplot(c(1,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,1,0))
myroute(c(1,1,1,1,1,1,1,1,1,1,0,1,0,1,0,0,0,0,0,1))
myplot(c(1,1,1,1,1,1,1,1,1,1,0,1,0,1,0,0,0,0,0,1))
##############################################


## Naive MC
nsim <- 10^5
p <- 0.05
set.seed(19960129)
x <- matrix((1 - rbinom(20*nsim, 1, p)), ncol = 20, nrow=nsim)
hx <- 1 - apply(x, 1, myroute)
(failure_mc <- sum(hx))
(mu_mc <- mean(hx))
(var_mc <- mu_mc*(1-mu_mc)/nsim)

## Importance Sampling
IS <- function(pstar=0.25){
  set.seed(19960129)
  nsim <- 10^5
  p <- 0.05

  x <- matrix((1 - rbinom(20*nsim, 1, pstar)), ncol = 20, nrow=nsim)
  b <- apply(1-x, 1, sum)
  w <- ((1-p)/(1-pstar))^20*(p*(1-pstar)/pstar*(1-p))^b
  hx <- 1 - apply(x, 1, myroute)
  failure_is <- sum(hx)
  mu_is <- mean(hx*w)
  var_is <- (sum(hx*w*p^b*(1-p)^(20-b)) - mu_is^2)/nsim
  return(list(mu_is=mu_is, var_is=var_is, failure_is=failure_is))
}
# 1. pstar = 0.25
result <- IS()
result
var_mc/result$var_is

# 2. pstar = 0.2
result2 <- IS(pstar=0.2)
result2
var_mc/result2$var_is

## Antithetic
ANT <- function(nsim=10^5){
  p <- 0.05

  u <- matrix(runif(20*nsim), ncol=20, nrow=nsim)
  xx <- ifelse(u < p, 0, 1)
  naive <- 1 - apply(xx, 1, myroute)
  mu_naive <- mean(naive)

  u.half <- u[1:(nsim/2),]
  x <- ifelse(u.half < p, 0, 1)
  xstar <- ifelse(u.half > (1-p), 0, 1)
  hx <- 1 - apply(x, 1, myroute)
  hxstar <- 1 - apply(xstar, 1, myroute)
  mu_ant <- (mean(hx) + mean(hxstar))/2
  return(list(naive=mu_naive, ant=mu_ant))
```

```
}
N <- 10; res_naive <-0; res_ant <- 0
for(i in 1:N){
  res_naive[i] <- ANT()$naive
  res_ant[i] <- ANT()$ant
}
var(res_naive); var(res_ant)
res_naive; res_ant


## Mixture of Antithetic and Importance Sampling
MIX <- function(nsim=10^5){
  p <- 0.05; pstar <- 0.2

  u <- matrix(runif(20*nsim), ncol=20, nrow=nsim)
  u.half <- u[1:(nsim/2), ]
  x <- ifelse(u.half < pstar, 0, 1)
  b <- apply(1-x,1,sum)
  w <- ((1-p)/(1-pstar))^20*(p*(1-pstar)/pstar*(1-p))^b
  hx <- 1 - apply(x, 1, myroute)

  xstar <- ifelse(u.half > (1-pstar), 0, 1)
  bstar <- apply(1-xstar,1,sum)
  wstar <- ((1-p)/(1-pstar))^20*(p*(1-pstar)/pstar*(1-p))^bstar
  hxstar <- 1 - apply(xstar, 1, myroute)

  mu_mix <- (mean(hx*w) + mean(hxstar*wstar))/2

  return(mu_mix)
}
N <- 10; res_mix <- 0
for(i in 1:N){
  res_mix[i] <- MIX()
}
var(res_mix)
res_mix
```