HW#09

Zijian Huang

11/13/2020

E 7.5.1

a

$$
\begin{aligned}
E(X^2) &= \int_{-\infty}^{\infty} x^2 \frac{1}{5\sqrt{2\pi}} x^2 e^{-\frac{(x-2)^2}{2}} dx \\
&= \int_{-\infty}^{\infty} x^2 \frac{1}{5} x^2 e^{-(2-2x)} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \\
&= \int_{-\infty}^{\infty} h(x) \frac{f(x)}{g(x)} g(x) dx \\
&= \int_{-\infty}^{\infty} h(x) w(x) g(x) dx
\end{aligned}
$$

So we have:

$$
h(x) = x^2, \quad w(x) = \frac{f(x)}{g(x)} = \frac{1}{5} x^2 e^{-(2-2x)}, \quad g(x) : \text{is the standard normal pdf}
$$

```r
set.seed(123)
sim_imp <- function(n){
  # importance ratio
  w <- function(x){
    x^2 * exp(2*x - 2)/5
  }
  x <- rnorm(n)
  y <- mean(x^2*w(x))
  return(y)
}

# replicate the estimate 1000 times
nrep <- 1000
# 1000 samples
est1 <- replicate(nrep, sim_imp(1000))
# 10000 samples
est2 <- replicate(nrep, sim_imp(10000))
# 50000 samples
est3 <- replicate(nrep, sim_imp(50000))
all_est <- list(est1, est2, est3)
mean <- sapply(all_est, mean)
```

```r
variance <- sapply(all_est, var)
df <- data.frame(samples=c(1000, 10000, 50000), cbind(mean, variance))
knitr::kable(df, booktabs = TRUE,
             caption = "Importance sampling mean and variance estimation")
```

Table 1: Importance sampling mean and variance estimation

samples	mean	variance
1000	8.875260	266.04652
10000	8.653025	34.40799
50000	8.556730	5.82669

b

$$
\begin{aligned}
E(X^2) &= \int_{-\infty}^{\infty} x^2 \frac{1}{5\sqrt{2\pi}} x^2 e^{-\frac{(x-2)^2}{2}} dx \\
&= \int_{-\infty}^{\infty} x^2 \frac{1}{5} x^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-2)^2}{2}} dx \\
&= \int_{-\infty}^{\infty} h(x) \frac{f(x)}{g(x)} g(x) dx
\end{aligned}
$$

So we have:

$$
h(x) = x^2, \quad w(x) = \frac{f(x)}{g(x)} = \frac{1}{5}x^2, \quad g(x): \text{is the pdf of N(2, 1)}
$$

c

```r
set.seed(0)
sim_imp <- function(n){
  # importance ratio
  w <- function(x){
    x^2/5
  }
  x <- rnorm(n, 2, 1)
  y <- mean(x^2*w(x))
  return(y)
}

# replicate the estimate 1000 times
nrep <- 1000
# 1000 samples
est1 <- replicate(nrep, sim_imp(1000))
# 10000 samples
est2 <- replicate(nrep, sim_imp(10000))
# 50000 samples
est3 <- replicate(nrep, sim_imp(50000))
all_est <- list(est1, est2, est3)
mean <- sapply(all_est, mean)
```

```r
variance <- sapply(all_est, var)
df <- data.frame(samples=c(1000, 10000, 50000), cbind(mean, variance))
knitr::kable(df, booktabs = TRUE,
             caption = "Importance sampling mean and variance estimation with better g(x)")
```

Table 2: Importance sampling mean and variance estimation with better g(x)

samples	mean	variance
1000	8.601180	0.2073631
10000	8.597461	0.0203899
50000	8.601717	0.0041690

d

Firstly, as the sample size increasing, the variance decreases in the two method. Moreover, the g(x) in (c) turn out to have less variance of than the method in (a). The reason is g(x) in (c) is more proportional to h(x)f(x) than g(x) in (a). It means the estimates will be more spread out in method in (a), on the other hand, the estimates is more stable for the method in (c). The results is suitable for different sample sizes.

E 7.5.2

a

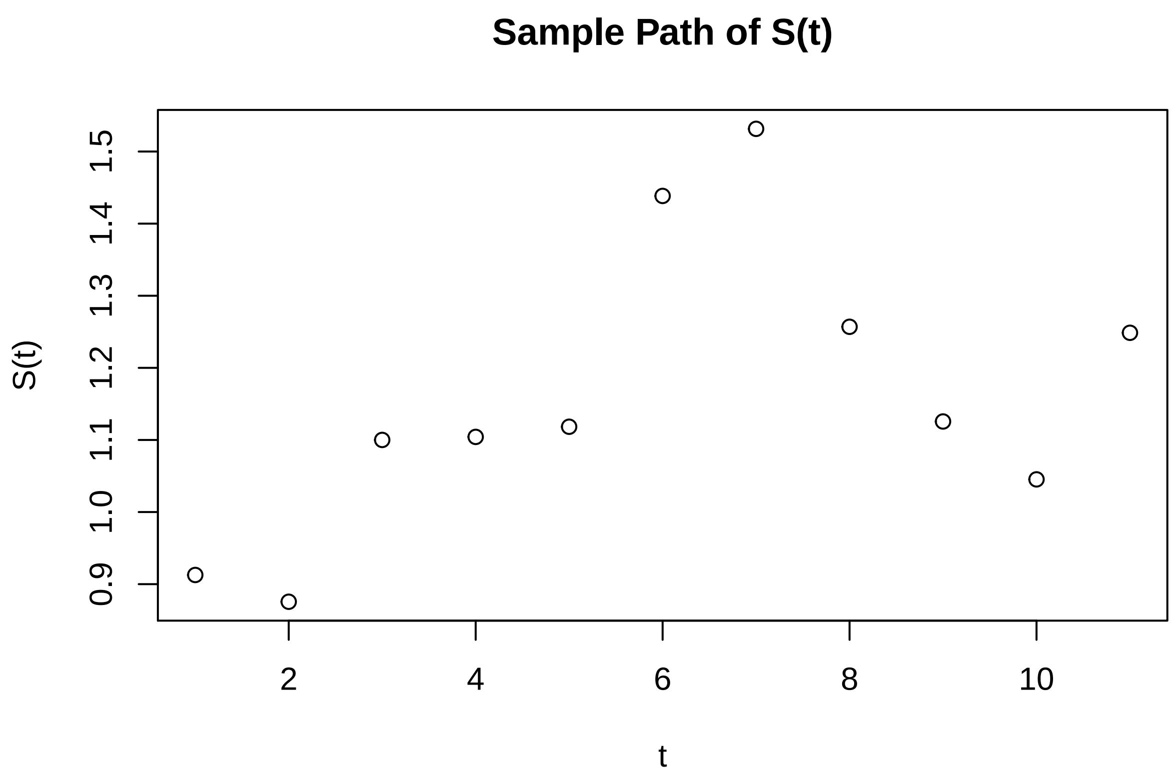
$$\text{Firstly, we may guess: } S(t) = S(0)e^{at+bW(t)} = f(t, W(t)),\ \ f(t,x) = S(0)e^{at+bx}$$

Then by Ito's formula:

$$S(t) = S(0)e^{(r-\sigma^2/2)t+\sigma W(t)}, \quad W(t) \sim N(0,t)$$

```r
set.seed(123)
rBM <- function(n, tgrid, sigma, r, S0) {
  tt <- c(0, tgrid)
  dt <- diff(tt)
  nt <- length(tgrid)
  dw <- matrix(rnorm(n * nt, sd = sigma * sqrt(dt)), n, nt, byrow = TRUE)
  wt <- t(apply(dw, 1, cumsum))
  St <- S0 * exp((r - sigma^2 / 2) * matrix(tgrid, n, nt, byrow = TRUE) + wt)
  return(St)
}

S0 <- 1; r <- 0.05; sigma <- 0.5; n <- 1
tgrid <- seq(0, 1, length = 12)[-1]
plot(drop(rBM(n, tgrid, sigma, r, S0)), ylab = 'S(t)', xlab = "t",
     main = 'Sample Path of S(t)')
```



b

```
set.seed(123)
optValueAppr <- function(K) {
  S0 <- 1; r <- 0.05; sigma <- 0.5; n <- 5000
  tgrid <- seq(0, 1, length = 12)[-1]
  ## payoff of call option on arithmetic average
  nt <- length(tgrid)
  TT <- tgrid[nt]
  St <- rBM(n, tgrid, sigma, r, S0)
  pAri <- pmax(rowMeans(St) - K, 0) * exp(-r * TT)
  vAri <- mean(pAri)
  ## underlying asset price
  ST <- St[, nt]
  ## value of standard option
  pStd <- pmax(ST - K, 0) * exp(-r * TT)
  ## payoff of call option on geometric average
  pGeo <- pmax(exp(rowMeans(log(St))) - K, 0) * exp(-r * TT)
  list(pAri = pAri, ST = ST, pStd = pStd, pGeo = pGeo )
}
p1 <- optValueAppr(1.1); p2 <- optValueAppr(1.2); p3 <- optValueAppr(1.3)
p4 <- optValueAppr(1.4); p5 <- optValueAppr(1.5)
cc1 <- cbind(cor(p1$pAri,p1$ST),cor(p1$pAri,p1$pStd),cor(p1$pAri,p1$pGeo))
cc2 <- cbind(cor(p2$pAri,p2$ST),cor(p2$pAri,p2$pStd),cor(p2$pAri,p2$pGeo))
```

```r
cc3 <- cbind(cor(p3$pAri,p3$ST),cor(p3$pAri,p3$pStd),cor(p3$pAri,p3$pGeo))
cc4 <- cbind(cor(p4$pAri,p4$ST),cor(p4$pAri,p4$pStd),cor(p4$pAri,p4$pGeo))
cc5 <- cbind(cor(p5$pAri,p5$ST),cor(p5$pAri,p5$pStd),cor(p5$pAri,p5$pGeo))
df = data.frame(K=c(1.1, 1.2 ,1.3 ,1.4 ,1.5), rbind(cc1,cc2,cc3,cc4,cc5))
colnames(df) = c("K", 'cor(P(A),S(T))','cor(P(A),P(E))','cor(P(A),P(G))')
knitr::kable(df, booktabs = TRUE,
             caption = "Correlation coefficients table as K increasing")
```

Table 3: Correlation coefficients table as K increasing

K	cor(P(A),S(T))	cor(P(A),P(E))	cor(P(A),P(G))
1.1	0.8016188	0.8476830	0.9968403
1.2	0.7625834	0.8402894	0.9966127
1.3	0.7244135	0.8351919	0.9952179
1.4	0.6890102	0.8278443	0.9922221
1.5	0.6045852	0.7812171	0.9915341

Thus, from the table, we know that as K increases, correlation coefficients decreases.

c

```r
set.seed(123)
optValueAppr <- function(sigma) {
  S0 <- 1; r <- 0.05; K <- 1.5; n <- 5000
  tgrid <- seq(0, 1, length = 12)[-1]
  ## payoff of call option on arithmetic average
  nt <- length(tgrid)
  TT <- tgrid[nt]
  St <- rBM(n, tgrid, sigma, r, S0)
  pAri <- pmax(rowMeans(St) - K, 0) * exp(-r * TT)
  vAri <- mean(pAri)
  ## underlying asset price
  ST <- St[, nt]
  ## value of standard option
  pStd <- pmax(ST - K, 0) * exp(-r * TT)
  ## payoff of call option on geometric average
  pGeo <- pmax(exp(rowMeans(log(St))) - K, 0) * exp(-r * TT)
  list(pAri = pAri, ST = ST, pStd = pStd, pGeo = pGeo )
}
p1 <- optValueAppr(0.2); p2 <- optValueAppr(0.3)
p3 <- optValueAppr(0.4);p4 <- optValueAppr(0.5)
cc1 <- cbind(cor(p1$pAri,p1$ST),cor(p1$pAri,p1$pStd),cor(p1$pAri,p1$pGeo))
cc2 <- cbind(cor(p2$pAri,p2$ST),cor(p2$pAri,p2$pStd),cor(p2$pAri,p2$pGeo))
cc3 <- cbind(cor(p3$pAri,p3$ST),cor(p3$pAri,p3$pStd),cor(p3$pAri,p3$pGeo))
cc4 <- cbind(cor(p4$pAri,p4$ST),cor(p4$pAri,p4$pStd),cor(p4$pAri,p4$pGeo))
df = data.frame(sigma=c(0.2, 0.3, 0.4, 0.5), rbind(cc1,cc2,cc3,cc4))
colnames(df) = c("sigma", 'cor(P(A),S(T))','cor(P(A),P(E))','cor(P(A),P(G))')
knitr::kable(df, booktabs = TRUE,
             caption = "Correlation coefficients table as sigma increasing")
```

Table 4: Correlation coefficients table as sigma increasing

sigma	cor(P(A),S(T))	cor(P(A),P(E))	cor(P(A),P(G))
0.2	0.1223955	0.4924871	0.9972307
0.3	0.3293596	0.6206308	0.9904903
0.4	0.5167319	0.7561853	0.9912850
0.5	0.6358100	0.8105785	0.9899014

Thus, from the table, we know that as σ increases, correlation coefficients increases.

d

```r
set.seed(123)
optValueAppr <- function(tgrid) {
  S0 <- 1; r <- 0.05; sigma <- 0.5; n <- 5000; K <- 1.5
  ## payoff of call option on arithmetic average
  nt <- length(tgrid)
  TT <- tgrid[nt]
  St <- rBM(n, tgrid, sigma, r, S0)
  pAri <- pmax(rowMeans(St) - K, 0) * exp(-r * TT)
  vAri <- mean(pAri)
  ## underlying asset price
  ST <- St[, nt]
  ## value of standard option
  pStd <- pmax(ST - K, 0) * exp(-r * TT)
  ## payoff of call option on geometric average
  pGeo <- pmax(exp(rowMeans(log(St))) - K, 0) * exp(-r * TT)
  list(pAri = pAri, ST = ST, pStd = pStd, pGeo = pGeo )
}

p1 <- optValueAppr(seq(0, 0.4, length = 12)[-1])
p2 <- optValueAppr(seq(0, 0.7, length = 12)[-1])
p3 <- optValueAppr(seq(0, 1, length = 12)[-1])
p4 <- optValueAppr(seq(0, 1.3, length = 12)[-1])
p5 <- optValueAppr(seq(0, 1.6, length = 12)[-1])
cc1 <- cbind(cor(p1$pAri,p1$ST),cor(p1$pAri,p1$pStd),cor(p1$pAri,p1$pGeo))
cc2 <- cbind(cor(p2$pAri,p2$ST),cor(p2$pAri,p2$pStd),cor(p2$pAri,p2$pGeo))
cc3 <- cbind(cor(p3$pAri,p3$ST),cor(p3$pAri,p3$pStd),cor(p3$pAri,p3$pGeo))
cc4 <- cbind(cor(p4$pAri,p4$ST),cor(p4$pAri,p4$pStd),cor(p4$pAri,p4$pGeo))
cc5 <- cbind(cor(p5$pAri,p5$ST),cor(p5$pAri,p5$pStd),cor(p5$pAri,p5$pGeo))
df = data.frame(T=c(0.4, 0.7, 1, 1.3, 1.6), rbind(cc1,cc2,cc3,cc4,cc5))
colnames(df) = c("T", 'cor(P(A),S(T))','cor(P(A),P(E))','cor(P(A),P(G))')
knitr::kable(df, booktabs = TRUE,
             caption = "Correlation coefficients table as T increasing")
```

Table 5: Correlation coefficients table as T increasing

T	cor(P(A),S(T))	cor(P(A),P(E))	cor(P(A),P(G))
0.4	0.3675418	0.7147200	0.9896942
0.7	0.5208184	0.7450116	0.9915409

T	cor(P(A),S(T))	cor(P(A),P(E))	cor(P(A),P(G))
1.0	0.6231265	0.8011013	0.9918954
1.3	0.6970990	0.8306583	0.9891274
1.6	0.7067006	0.8175632	0.9906789

Thus, from the table, we know that as T increases, correlation coefficients increases.

e

```
set.seed(123)
callValLognorm <- function(S0, K, mu, sigma) {
  d <- (log(S0 / K) + mu + sigma^2) / sigma
  S0 * exp(mu + 0.5 * sigma^2) * pnorm(d) - K * pnorm(d - sigma)
}
optValueAppr <- function(n, r, sigma, S0, K, tgrid) {
    ## payoff of call option on arithmetic average
    nt <- length(tgrid)
    TT <- tgrid[nt]
    St <- rBM(n, tgrid, sigma, r, S0)
    pAri <- pmax(rowMeans(St) - K, 0)
    vAri <- mean(pAri)
    ## underlying asset price
    ST <- St[, nt]
    vAs <- vAri - cov(ST, pAri) / var(ST) * (mean(ST) - exp(r * TT) * S0)
    ## value of standard option
    pStd <- pmax(ST - K, 0)
    pStdTrue <- callValLognorm(S0, K, (r - 0.5 * sigma^2) * TT,
                               sigma * sqrt(TT))
    vStd <-  vAri - cov(pStd, pAri) / var(pStd) * (mean(pStd) - pStdTrue)
    ## payoff of call option on geometric average
    pGeo <- pmax(exp(rowMeans(log(St))) - K, 0)
    tbar <- mean(tgrid)
    sBar2 <- sigma^2 / nt^2 / tbar * sum( (2 * seq(nt) - 1) * rev(tgrid) )
    pGeoTrue <- callValLognorm(S0, K, (r - 0.5 * sigma^2) * tbar,
                               sqrt(sBar2 * tbar))
    vGeo <- vAri - cov(pGeo, pAri) / var(pGeo) * (mean(pGeo) - pGeoTrue)
    ## sim <- data.frame(pAri, ST, pStd, pGeo)
    ## result
    c(vGeo, vAri) * exp(-r * TT)
}
r <- 0.05; sigma <- 0.4; S0 <- 1; K <- 1.5
tgrid <- seq(0, 1, length = 12)[-1]
sim <- replicate(1000, optValueAppr(500, r, sigma, S0, K, tgrid))
df = data.frame(rbind(apply(sim, 1, mean),apply(sim, 1, sd)))
rownames(df) <- c('Mean','SD')
colnames(df) <- c("with Control", "no Control")
knitr::kable(df, booktabs = TRUE,
             caption = "Control variate MC estimator comparison")
```

Table 6: Control variate MC estimator comparison

	with Control	no Control
Mean	0.0090529	0.0090806
SD	0.0003743	0.0025676