

# 09-monte-carlo-integration-YuanweiQiao

Yuanwei Qiao

11/13/2020

## 7.5.1 Importance Sampling Method

Importance sampling method can be used to evaluate complex integration problems or  $E(h(X))$  when  $X$  follows an unfamiliar density function  $f(x)$ . In the given problem, we are trying to evaluate an integration given in equation (1). In order to do so, we can use different envelope distributions as importance samplers. In this case we initially try using standard normal distribution as an envelope distribution. This implies that the integration can now be evaluated as an expectation of  $\frac{h(x)f(x)}{g(x)}$  where  $X$  follows density  $g(x)$ , which in this case is standard normal distribution.

$$\int_0^\infty h(x)f(x) = \int_0^\infty \frac{1}{5\sqrt{2\pi}} x^4 e^{-\frac{(x-2)^2}{2}} \quad (1)$$

### Evaluating Equation (1) using Envelope as “Standard Normal”

In order to implement the importance sampling using standard normal density as the envelope distribution, we need to initialize by defining the functions `f(x)` in R.

```
f <- function(x) {  
  if (x > 0) {(1/sqrt(50*pi))*x^2*(exp(-(1/2)*((x-2)^2))) }  
  else {0}  
}
```

The underlying technique of importance sampling is that we can evaluate any integral as an Expectation under a familiar density function  $g(x)$  that we choose, if the support of the integral is a subset of support of the envelope density. This can be observed as follows:

$$\int_0^\infty h(x)f(x) = \int_0^\infty \frac{h(x)f(x)}{g(x)}g(x) = E\left(\frac{h(X)f(X)}{g(X)}\right) \quad (2)$$

We therefore write the code that can utilize the importance sample method to evaluate the integral of  $h(x) * f(x)$  using any envelope  $g(x)$ . In order to use standard normal as  $g(x)$ , we use “envelope = dnorm”, and “renvelope = rnorm”.

```

##' @param n desired number of samples
##' @param f target density
##' @param envelope density of sampler (= dnorm in the first scenario)
##' @param renvelope random sampler function for density of envelope
##' Observe that envelope's support should be a superset of f's support (if f > 0,
##' then envelope > 0)
##' If envelope is considered normal density,  $w(x) = f(x)/g(x) = x$ ,
##' and since  $h(x) = x^2$  then  $w(x)*h(x) = x^3$ 
##' exponent function is the dominating function and hence we need to find similar
##' parameters as in the exponent.

importance_sample <- function(f, n, envelope = dnorm, renvelope = rnorm) {
  x      <- renvelope(n)
  h_x    <- x^2
  weight_x <- sapply(x,f)/envelope(x)
  estimate <- mean(h_x*weight_x)
  estimate
}

set.seed(207)
is_output <- matrix(data = c(1000, 10000, 50000,
                             importance_sample(f, 1000, dnorm, rnorm),
                             importance_sample(f, 10000, dnorm, rnorm),
                             importance_sample(f, 50000, dnorm, rnorm)),
                    nrow = 3, byrow = F)
colnames(is_output) <- c("Samples", "Estimate")
knitr::kable(is_output, caption = "Integral Value using Standard Normal Envelope")

```

Table 1: Integral Value using Standard Normal Envelope

Samples	Estimate
1000	2.859289
10000	9.300545
50000	7.953563

From the above table, we can observe that the estimate is not consistent and has a lot of variance depending upon number of simulations. In this case the actual value of the integral can be solved from the fourth raw moment of the normal distribution with mean 2, and variance 1. You can also utilize the integrate function in R to evaluate the actual value which is 7 in this case.

```

integrate(actual_integration <- function(x)
  {(1/sqrt(50*pi))*(x^4)*(exp(-(1/2)*((x-2)^2)))}, 0, Inf)

```

```
## 8.598716 with absolute error < 8.3e-05
```

In the previous table, we could observe that while there was a lot of deviation in the sample estimate, as the number of samples increased to 50000, the estimate was 7.9 which was getting closer to 8.5987 that is the actual value of the integration.

## Better Importance Sampling than using Standard Normal as Envelope

Better importance sampling is usually observed when  $f(x)$  is proportional to  $g(x)$ . We would observe that the error reduces with even few samples. We can observe that Standard Normal was not proportional to the given integral. Normal(2,1) has a density function that is proportional to  $g(x)$  and therefore will provide lesser variance estimates while applying importance sampling method.

In order to apply importance sampling method with Normal(2,1) density, we need to define Normal(2,1) density and the random number generator function from Normal(2,1).

```
d_norm_2_1 <- function(x) {  
  dnorm(x, 2, 1)  
}  
  
r_norm_2_1 <- function(n) {  
  rnorm(n, 2, 1)  
}
```

## Estimating Importance Sampling estimators and their variance using Normal(2,1) as Envelope

Earlier, we built the function “importance\_sample” so that it can take any density as envelope as long as we can construct its density function and its random generator. We have also constructed the functions for the Normal (2,1) density as “dnorm\_2\_1” and its random generator as “r\_norm\_2\_1”. Now, we use these to estimate importance sampling estimators for the unknown integral that requires to be solved.

```
set.seed(207)  
is_output <- matrix(data = c(1000, 10000, 50000,  
                             importance_sample(f, 1000, d_norm_2_1, r_norm_2_1),  
                             importance_sample(f, 10000, d_norm_2_1, r_norm_2_1),  
                             importance_sample(f, 50000, d_norm_2_1, r_norm_2_1)),  
                    nrow = 3, byrow = F)  
colnames(is_output) <- c("Samples", "Estimate")  
knitr::kable(is_output, caption = "Integral Value using Better Envelope - Normal(2,1)")
```

Table 2: Integral Value using Better Envelope - Normal(2,1)

Samples	Estimate
1000	8.153559
10000	8.695343
50000	8.607813

As it can be observed the integral value estimators using importance sampling with better envelope are obtained much closer to 8.5987 and the estimates are closer to 8 even in case of 1000 samples suggesting that Normal(2,1) is a much better envelope for this integral in comparison to Standard Normal.

## Comparison of two methods' results and Conclusion

This brings us to comparison part where we evaluate the estimate's bias and variance in order to observe and infer regarding which envelope provides better importance sampling estimates with lesser number of

samples. The estimate for variances can be evaluated replicating the estimate a few number of times, thereby constructing a distribution for the estimate. We replicate each calculation of importance sample estimate 200 times in order to evaluate the bias estimates and the variance estimates of the importance sample estimate.

```
set.seed(207)
sim1 <- replicate(200, importance_sample(f, 1000, dnorm, rnorm))
sim2 <- replicate(200, importance_sample(f, 10000, dnorm, rnorm))
sim3 <- replicate(200, importance_sample(f, 50000, dnorm, rnorm))
compare_std_norm <- matrix(c(1000, 10000, 50000,
                             abs(mean(sim1-8.5987)), abs(mean(sim2-8.5987)), abs(mean(sim3-8.5987)),
                             var(sim1), var(sim2), var(sim3)),
                             nrow = 3, byrow = F)
colnames(compare_std_norm) <- c("Samples", "Estimate's Bias", "Estimate's Variance")
knitr::kable(compare_std_norm, caption = "Bias and Variance Estimates under Standard Normal Envelope")
```

Table 3: Bias and Variance Estimates under Standard Normal Envelope

Samples	Estimate's Bias	Estimate's Variance
1000	1.2746340	45.99179
10000	0.9054780	153.76548
50000	0.2190318	2.81768

```
set.seed(207)
sim1 <- replicate(200, importance_sample(f, 1000, d_norm_2_1, r_norm_2_1))
sim2 <- replicate(200, importance_sample(f, 10000, d_norm_2_1, r_norm_2_1))
sim3 <- replicate(200, importance_sample(f, 50000, d_norm_2_1, r_norm_2_1))
compare_norm_2_1 <- matrix(c(1000, 10000, 50000, abs(mean(sim1-8.5987)), abs(mean(sim2-8.5987)),
                             abs(mean(sim3-8.5987)), var(sim1), var(sim2), var(sim3)),
                             nrow = 3, byrow = F)
colnames(compare_norm_2_1) <- c("Samples", "Estimate's Bias", "Estimate's Variance")
knitr::kable(compare_norm_2_1, caption = "Bias and Variance Estimates under Normal(2,1) Envelope")
```

Table 4: Bias and Variance Estimates under Normal(2,1) Envelope

Samples	Estimate's Bias	Estimate's Variance
1000	0.0168162	0.1939877
10000	0.0125092	0.0230246
50000	0.0009312	0.0045686

## Conclusion

From the above results we conclude that importance sampling is a good estimation process of an unknown integral or expectation and it gives better estimates with lesser bias and variance when you use envelope density function  $g(x)$  that is proportional to  $f(x)$  or the integral. In this case, therefore, we observe that  $Normal(2, 1)$  provides a better envelope and a better estimate with lesser number of samples than when using Standard Normal as envelope density function. It can be observed that  $Normal(2, 1)$  provides a lesser variance with 1000 samples than the variance obtained when using 50000 samples and standard normal

distribution envelope.

## 7.5.2 Reduction of Variance - Control Variate Method

In this question, we are evaluating the control variate method in reducing variance of an asian stock option price estimate. The monte carlo simulation produces really good result but at the expense of computational time. However, we can use payoff of correlated control variates in order to reduce the number of simulation and still obtain the same level of accuracy. We initiate with implementing an algorithm to sample the path of a geometric brownian motion for stock price.

### Algorithm to generate Geometric Brownian Process path from (0,T)

The stochastic differential for geometric brownian motion has the following closed form solution and therefore allows us to easily implement the stock price sample path. The closed form solution for the geometric brownian motion for the stock price path is given as follows:

$$\begin{aligned}\frac{dS(t)}{S(t)} &= rdt + \sigma dW(t) \\ d(\ln S(t)) &= rdt + \sigma dW(t) \\ S(t_{i+1}) &= S(t_i) \exp\left((r - \frac{1}{2}\sigma^2)(t_{i+1} - t_i) + \sigma\sqrt{(t_{i+1} - t_i)}Z\right), \\ Z &\sim N(0,1)\end{aligned}$$

```
set.seed(107)
# This is written when time grid is time values equally distant
# from each other from 0 to maturity. Else tgrid could be considered
# as input just like it is done in class.
stock_price <- function(S_0 = 1, drift_rate = 0.05, time_steps = 12, sample_no = 5000, sigma, Maturity) {
  tt <- seq(0, Maturity, 1/time_steps)
  dt <- diff(tt)
  dW <- matrix(data = rnorm(sample_no*time_steps, 0, sd = sigma*sqrt(dt)), nrow = sample_no, ncol = time_steps)
  W <- t(apply(dW, 1, cumsum))
  stock_price <- S_0*(exp((drift_rate - (1/2)*sigma^2)*matrix(tt[-1], sample_no, time_steps, byrow = T)))
  stock_price
}

a <- stock_price(sample_no = 200000, sigma = 0.5, Maturity = 1)
mean(a[,12])/1
```

```
## [1] 1.051984
```

```
exp(0.05)
```

```
## [1] 1.051271
```

We diagnosed that the stock price was correctly generated from the algorithm by evaluating the mean of terminal stock value which should be equivalent to “ $e^{rT} = e^{0.05} = 1.051$ ”

## MC Estimates of Correlation Coefficients between Asian Option Price and Control Variates with changes in K, Sigma, and Maturity Time

We now evaluate the correlation coefficients between asian option price and different control variates such as the terminal stock price, european option price and geometric option price to understand which will be a better control variate. We know that there is maximum reduction of variance or best improvement when control variate is highly correlated with the required asian option price (desired output). We evaluate the correlation values by changing the values of strike price  $K$ , maturity  $T$ , and volatility  $\sigma$  and observe how the correlation changes.

We initiate by creating a procedure that allows to evaluate the correlation between desired output (arithmetic asian option) and the possible control variates (terminal stock price, european option price and geometric option price) as a function of strike price  $K$ , standard deviation  $sd$ , and maturity  $M$ .

```
set.seed(107)
# This is written when time grid is time values equally distant
# from each other from 0 to maturity. Else tgrid could be considered
# as input just like it is done in class.

correlations <- function(K, r, M, sd) {
stock_path <- stock_price(S_0 = 1, drift_rate = 0.05, time_steps = 12, sample_no = 5000, sigma = sd, Ma
S_T <- stock_path[,dim(a)[2]]
discounted_payoff_Arithmetic <- exp(-r*M)*pmax(rowMeans(stock_path) - K, 0)
discounted_payoff_European <- exp(-r*M)*pmax(S_T - K, 0)
discounted_payoff_Geometric <- exp(-r*M)*pmax(exp(rowMeans(log(stock_path)))) - K, 0)
c(cor(discounted_payoff_Arithmetic, S_T),cor(discounted_payoff_Arithmetic, discounted_payoff_European
})
```

### Change in correlation with changes in strike price “K”

We used a set of strike prices such as  $\{1.1, 1.2, 1.3, 1.4, 1.5\}$  and observed how the correlation changed. As we increased the value of the strike price,  $K$ , we found that the correlations all decreased. We may attribute this to the reason that as  $K$  increases, more pay off values will become zero, therefore stops contributing to the correlation value. As a result it is intuitive that correlation reduces as  $K$  increases. From this, we can infer that there is highest variance reduction when  $K$  is very low.

Table 5: Changing Correlations as K Increases

K	Correlation of $P_A$ and $S(T)$	Correlation of $P_A$ and $P_E$	Correlation of $P_A$ and $P_G$
1.1	0.7966	0.8440	0.9970
1.2	0.7510	0.8284	0.9953
1.3	0.7076	0.8134	0.9950
1.4	0.6649	0.8138	0.9940
1.5	0.5909	0.7831	0.9910

### Change in correlation with changes in sigma

As we increased the value of  $\sigma$  in range  $\{0.2, 0.3, 0.4, 0.5\}$ , we found that the correlation between  $P_A$  and  $S(T)$  and the correlation between  $P_A$  and  $P_E$  increased. The correlation between  $P_A$  and  $P_G$  increased at first but then decreased for  $\sigma = 0.5$ . As  $\sigma$  increases, we expect more payoff values greater than zero and therefore contributing to the positive change in correlation. Hence we observe that correlation increases as

$\sigma$  increases. However,  $P_A$  and  $P_G$  are already highly correlated and reducing the number of zeroes may or may not contribute positively to the correlation. From this, we can infer that in most cases there is highest variance reduction when  $\sigma$  is high, specially while using  $S(T)$  or  $P_E$  as control variate

Table 6: Changing Correlations as sigma Increases

sigma	Correlation of $P_A$ and $S(T)$	Correlation of $P_A$ and $P_E$	Correlation of $P_A$ and $P_G$
0.2	0.1090	0.3556	0.9764
0.3	0.3326	0.6520	0.9866
0.4	0.5150	0.7542	0.9919
0.5	0.5921	0.7739	0.9905

### Change in correlation with changes in Maturity M

As we increased the value of the maturity time,  $M$ , we observed that correlations between  $P_A$  and  $P_E$ ,  $S(T)$  decreased with increase in  $T$ , and then as  $T$  increased from 1.3 to 1.6, correlation again increased. However the maximum correlation is obtained for  $T = 0.4$  and therefore would result in maximum variance reduction. We don't really observe a clear pattern for correlation between  $P_A$  and  $P_G$ . But the maximum correlation is observed at  $T = 0.7$  and therefore you obtain maximum variance reduction for that value of  $T$ , if you are using  $P_G$  as control variate.

Table 7: Changing Correlations as T Increases

T	Correlation of $P_A$ and $S(T)$	Correlation of $P_A$ and $P_E$	Correlation of $P_A$ and $P_G$
0.4	0.6350	0.8118	0.9894
0.7	0.6275	0.7975	0.9910
1.0	0.6258	0.7981	0.9908
1.3	0.5997	0.7676	0.9875
1.6	0.6110	0.8023	0.9902

### Control Variate Estimate and Reduction of Variance (efficiency)

We finally evaluated the control variate estimate using  $P_G$  as control variate, by using 5000 samples and obtained values of 0.0088. We verified its accuracy by evaluating the monte carlo arithmetic mean using 500000 samples which we obtained as 0.0089.

```
## [1] "control variate estimate (5000 samples) is: 0.0089"
## [1] "lesser variance monte carlo estimate is: 0.0088"
## [1] "ratio of standard deviation is: 0.1404"
## [1] "variance reduction factor is: 0.0197"
## [1] "variance reduction factor should be: 0.0196"
```

We also evaluated the ratio of Standard error of control variate estimator to the standard error of the monte carlo estimator with 5000 samples to compare and obtain the variance reduction factor. We should obtain that the square of the above ratio is equal to  $1 - \rho^2$ , where  $\rho$  stands for the correlation between the control variate and the monte carlo payoffs. We observed that the results matched as the variance reduction factor = 0.0197. Also it implies that the variance of control variate estimator is 1.9% of the total variance of the monte carlo estimator with 5000 samples.