Problem Set 6

Marc Eskew

5/20/2022

Problem 1

1

We will sample α by using variance reduction to ensure convergence quicker than naive MC sampling. The variance reduction technique we will use is conditional Monte Carlo. This will be achieved by realizing that a with iid normal Gausian RVs X_i, then $\sum_{i=1}^{k} X_i^2 \sim Chi - Squared(k)$. We can condition on the known probability of the Chi-Squared distribution and the conditional probabilities of X_i.

```
##### Part 1 ####



mean <-0
mmt <- 0
eps <- .05
i <- 0

err <- Inf
while(i < 10000 | err >= eps){

  z <- rchisq(1,4)

  gz <- pchisq(z,4)*as.numeric(z>=24.5)*dnorm(sqrt(z/2),0,sqrt(2))*2

  mean <- (mean * i + gz) / (i+1)
  mmt <- (mmt*i+gz^2)/(i+1)
  std = sqrt(mmt- mean^2)

  i <- i+1

  eps <- .05*mean
  err <- 1.96*std/sqrt(i)
}
```

Using the conditiona MC resulted in a estimate of 1.3030506×10^{-6} **+- 5% with 95% confidence**.

Problem 2

2.1

We can compute a closed form solution for the optimal allocation of n_i^* by setting up an optimization problem and solving for the Lagrangian function and taking the partial derivatives.

$$
\begin{aligned}
&min f(n_i) = \sum_{i=1}^{m} p(i)^2 \frac{Var(X(i))}{n_i} \\
&st\ n_1 + ... + n_m = n
\end{aligned}
$$

$$
\begin{aligned}
L(n_i, \lambda) &= f(n_i) + \lambda g(n_i) \\
&= \sum_{i=1}^{m} p(i)^2 \frac{Var(X(i))}{n_i} + \lambda(n1 + ... + n_i - n)
\end{aligned}
$$

By checking the first order KKT conditions we can determine the optimal value for n_i.

$$
\begin{aligned}
\frac{d}{dn_i} L(n) \\
\lambda - \frac{p(i)^2 Var(X(i))}{n^2} = 0 \\
n_i^* = \frac{p(i)\sqrt{Var(X(i))}}{\sqrt{\lambda}}
\end{aligned}
$$

2.2

Use of a burn in period to determine a sample of variances could be utilized to determine the optimal values for n_i. These simulations will be important to establish a baseline to comput the remainder of your budget if the cost is worth it.

Problem 3

3.1

X_4 contributes the most to the variance of the function because as an expontential RV, the variance increases with expected value. As such, X_4 is the most important piece of the function and conditioning should be done on the other RVs.

3.2

Using the concept that $E[\sum_i X_i] = \sum_i E[X_i]$ we can set up a an explicit expression for $g(.)$.

$$
\begin{aligned}
E[max(X_1 + X_2, X_3 + X_4, X_1 + X_4, X_3 + X_2)|X_1 = x_1, X_2 = x_2, X_3 = x_3] \\
g(X_1, X_2, X_3) = \frac{1}{n} \sum_{i=1}^{n} max(x_1 + x_2, x_3 + 4, x_1 + 4, x_3 + x_2)
\end{aligned}
$$

3.3

```
rv_x <- function(i){
  rexp(1,rate = 1/i)
}
reps <- 1000
val_total <- vector()
for(i in 1:reps) {
```

```
  x1 <- rv_x(1)
  x2 <- rv_x(2)
  x3 <- rv_x(3)

  val <- max(x1+x2,x3+4,x1+4,x3+x2)
  val_total <- append(val_total,val)
}

std_val <- sqrt(var(val_total))
err_a <- (1.96*std_val)/sqrt(reps)

err_a
```

```
## [1] 0.1940048
```

```
mean(val_total)
```

```
## [1] 7.610096
```

```
mean(val_total) + err_a
```

```
## [1] 7.804101
```

```
mean(val_total) - err_a
```

```
## [1] 7.416091
```

α is between **7.4160914 and 7.8041009** with 95% confidence after 1000 replications.