STAT 525 — Spring 2020 — Section A1

Homework 5 Solution

1. The sequential importance sampling algorithm with the state equation as the proposal can be described as following:

   - Set $w_0 = 1$
   - Sample μ_1 from the proposal distribution (state equation) $f(\mu_1) = f(mu_1|\mu_0)$, and update the weight $w_1 = w_0 \cdot g(Y_1|\mu_1)$. Here g denotes the observation equation
   - For $t \geq 2$, sample μ_t from the proposal $f(\mu_t|\mu_{t-1})$, and update the weight $w_t = w_{t-1} \cdot g(Y_t|\mu_t)$
   - Repeat the steps above for m times, and for each t, estimate $E(\mu_t|Y_{1:t})$ by

$$\hat{h}_t = \frac{\sum_{j=1}^m \mu_t^{(j)} w_t^{(j)}}{\sum_{j=1}^m w_t^{(j)}}$$

   By implementing the following code, we get the following results for the filtering problem

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\hat{h}_t$	0.0022	-0.1482	0.0469	-0.0061	0.0669	0.0299	0.1834	0.1508	0.1111	0.2583	0.6366	0.6487	0.3460	0.4621	0.5583

```
m = 10000
y = c(-0.4,-1.4,1.1,-0.3,0.7,-0.1,1.3,0.3,0.1,1.2,2.0,0.9,-0.4,0.9,1.0)
T = length(y)
x = matrix(0,m,T)
w = matrix(0,m,T)
for(i in 1:m){
        for(t in 1:T){
                if(t==1){
                        x[i,1] = ifelse(runif(1)<0.9,0,rnorm(1))
                        w[i,1] = dnorm(y[1]-x[1,t])
                }
                else{
                        x[i,t] = ifelse(runif(1)<0.9,x[i,t-1],rnorm(1))
                        w[i,t] = w[i,t-1]*dnorm(y[t]-x[i,t])
                }
        }
}
apply(w*x,2,sum)/apply(w,2,sum)
```

2. We first estimate b by using the sample version of variance and covariance, which gives $b^* = -0.5476$. Then we estimate the mean of $X(b^*)$, where

$$X(b) = \frac{1}{1+X^3} - b\left[(1+X^3) - \frac{5}{4}\right].$$

The estimate of μ and the variance of the estimate are reported in the following table.

	Control variates	Naive Monte Carlo
$\hat{\mu}$	0.8364	0.8449
variance of the estimate	7.2619e-07	2.4831e-05

R code:

```
n = 1000
x = runif(n)
mu.hat.mc = mean(1/(1+x^3)) # naive Monte Carlo
var.mc = var(1/(1+x^3))/n

b.star = cov(1/(1+x^3),1+x^3)/var(1+x^3)
mu.hat.cv = mean(1/(1+x^3) - b.star*(x^3-1/4))
var.cv = var(1/(1+x^3) - b.star*(x^3-1/4))/n
```

3. The CDF of the Pareto distribution is

$$F(x) = \int_2^x 160 s^{-6} ds = 1 - 32x^{-5}, \quad x \geq 2.$$

The inverse CDF is given by

$$F^{-1}(u) = 2(1-u)^{-1/5}, \quad 0 \leq u \leq 1.$$

Let $m = 1000$. The antithetic variates method to estimate the mean of the Pareto distribution can be described as following:

- Generate i.i.d. samples $u^{(1)}, \ldots, u^{(m/2)}$ from Uniform$(0,1)$
- Calculate $x_1^{(i)} = F^{-1}(u^{(i)}) = 2(1-u^{(i)})^{-1/5}$, and $x_2^{(i)} = F^{-1}(1-u^{(i)}) = 2(u^{(i)})^{-1/5}$
- Calculate $x^{(i)} = \left(x_1^{(i)} + x_2^{(i)}\right)/2$
- The mean of this Pareto distribution is estimated by

$$\hat{\mu} = \frac{2}{m}\sum_{i=1}^{m/2} x^{(i)}$$

- The variance of the estimate is given by $\hat{\sigma}^2/(m/2)$, where $\hat{\sigma}^2$ is the sample variance of $\{x^{(i)}\}_{i=1}^{m/2}$

	Antithetic variates	Naive inversion method
$\hat{\mu}$	2.4805	2.5057
variance of the estimate	1.5167e-04	4.2205e-04

Results are reported in the following table.

R code:

```
m = 1000
U1 = runif(m,0,1)
X1 = 2*(1-U1)^{-1/5} # naive Monte Carlo
mean(X1)
var(X1)/m

U2 = runif(m/2)
X2 = (2*(1-U2)^{-1/5} + 2*U2^{-1/5})/2 # antithetic variates
mean(X2)
var(X2)/(m/2)
```