Statistical Computing 2

숙제 3
2019년 가을학기
응용통계학과 석사과정 최석준

1. Let f~N(0,1). Calculate below Expectiation with using importance sampling with g~N(20,1)

$$\mathrm{E_f}\left(I_{\{X>20\}}\right) = \int_{20}^{\infty} \phi(x)dx \,, where\ \phi: standard\ normal\ pdf$$

```
[R]
y=rnorm(sample.num, 20, 1)
weight = exp(dnorm(y, log=TRUE) - dnorm(y, 20, 1, log=TRUE)) * (y>20)
mean(weight)
```

Result:

```
[R interpreter]
> mean(weight)
[1] 2.673156e-89
```

2. With / Without antithetic sampling, estimate of standard normal cdf value when x=1.96. Compare the variance of estimate.

```
[R]
#MC
x = 1.96
iter = 1000
n = 10000
mc.cdf = rep(NA, iter)
antithetic.cdf = rep(NA, iter)
for(i in 1:iter){
      u = runif(n)
      g = x * exp(-0.5*(u*x)^2)
      mc.cdf[i] = 0.5 + mean(g)/sqrt(2*pi)
}
mean(mc.cdf)
var(mc.cdf)

#antithetic
as.cdf = rep(NA, iter)
antithetic.cdf = rep(NA, iter)
for(i in 1:iter){
      u = runif(n/2)
      u = c(u, 1-u)
      g = x * exp(-0.5*(u*x)^2)
      as.cdf[i] = 0.5 + mean(g)/sqrt(2*pi)
}
mean(as.cdf)
var(as.cdf)

#비교
c(mean(mc.cdf), mean(as.cdf))
c(var(mc.cdf), var(as.cdf))
(var(mc.cdf)-var(as.cdf))/var(mc.cdf)
```

Result:

```
[R interpreter]
> c(mean(mc.cdf), mean(as.cdf))
[1] 0.9750090 0.9750042
> c(var(mc.cdf), var(as.cdf))
[1] 4.999333e-06 1.964741e-08
> (var(mc.cdf)-var(as.cdf))/var(mc.cdf)
[1] 0.99607
```

평균은 모두 참값인 0.975 근처에서 나온 것을 확인할 수 있다. 분산이 무려 99.6%만큼 분산이 줄었다!

3. Using control variate, compute

$$\int_0^1 \frac{e^{-x}}{1+x^2}dx$$

For pair, use $g(x) = e^{-0.5}/(1+x^2)$.

Note that $E(g(U)) = \pi e^{-0.5}/4$ when $U \sim unif(0,1)$

Find λ and calculate how much you can reduce the variance.

```
[R]
#example 2 (HW3)
m = 10000
# MC
u = runif(m)
mc.sample = (exp(-u)/(1+u^2))
mean(mc.sample)
var(mc.sample)

#control variate MC
con.var.sample = (exp(-0.5)/(1+u^2)) #같은 u 써야함
lambda = -cov(mc.sample, con.var.sample)/var(con.var.sample)
print(lambda) # -2.45가량

con.sample= mc.sample + lambda*(con.var.sample - exp(-0.5)*pi/4)
mean(con.sample)
var(con.sample)

#개선?
c(mean(mc.sample), mean(con.sample))
c(var(mc.sample), var(con.sample))
(var(mc.sample)-var(con.sample))/var(mc.sample)
```

Result:

```
[R interpreter]
> print(lambda)
[1] -2.453057
> c(mean(mc.sample), mean(con.sample))
[1] 0.5249687 0.5257966
> c(var(mc.sample), var(con.sample))
[1] 0.06101884 0.00308728
> (var(mc.sample)-var(con.sample))/var(mc.sample)
[1] 0.9494045
```

분산을 최소화하는 λ값은 -2.453057 로 계산되었다. (Analytic 하게가 아닌, Sample 값을 이용하여 계산하였다)

원하는 값에 해당하는 mean 값은 둘 다 비슷하게 나온 것을 확인할 수 있다. 하지만 분산의 경우, control variate 를 사용한 쪽이 무려 94%나 작은 것을 확인할 수 있다.

4. For given sample in norm.txt of $X|\theta \sim N(0,1)$, we assume Cauchy prior, $\theta \sim cauchy(0,1)$. Then the posterior is

$$\pi(\theta|x) \propto \frac{1}{\pi(1+\theta^2)} \frac{1}{2\pi} e^{-\frac{1}{2}(x-\theta)^2}$$

4-1. Using SIR, with Cauchy prior candidates (size 100,000), draw 5000 sample from posterior distribution and draw histogram.

4-2. Set envelope $e(x)$ as the likelihood function evaluated at MLE. Using Rejection sampling with Cauchy prior candidate, generate 5000 samples from posterior.

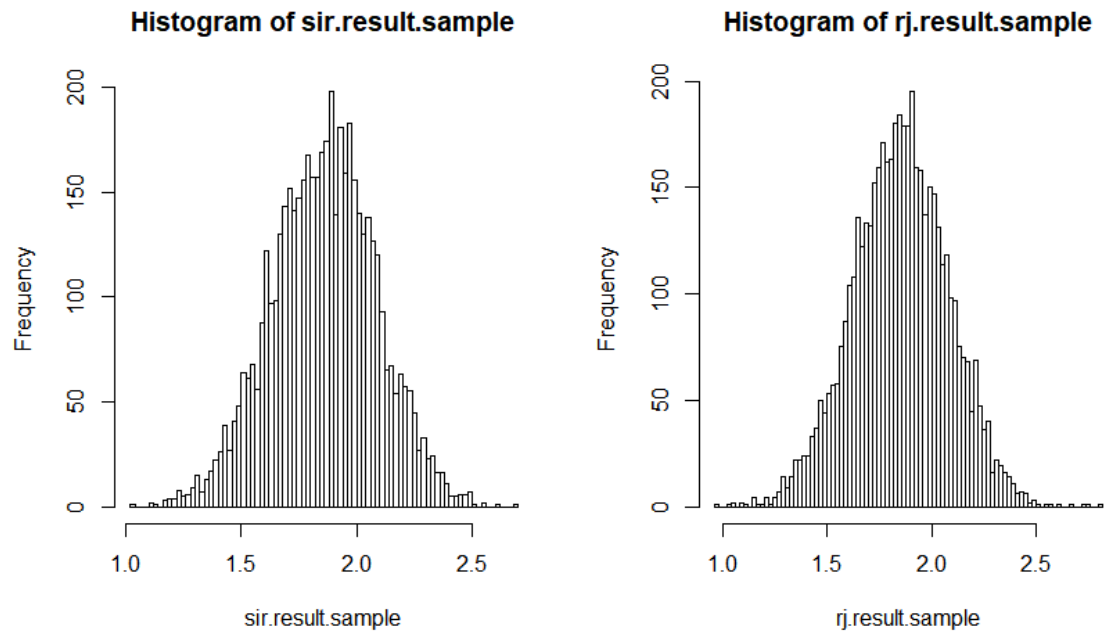
4-3. compare 4-1 and 4-2.

```
[R]
#HW3
#hw4.1 : SIR
norm.txt.sample=c(2.983, 1.309, 0.957, 2.16, 0.801, 1.747, -0.274, 1.071, 2.094, 2.215, 2.255, 3.366, 1.028, 3.572, 2.236,
4.009, 1.619, 1.354, 1.415, 1.937)
m=100000
theta.sample = rcauchy(m,0,1)
weight = rep(NA, m)
#weight
for(i in 1:m){
    weight[i] = sum(dnorm(norm.txt.sample, theta.sample[i], 1, log=TRUE))
}
weight = weight - max(weight)
weight.st = exp(weight) / sum(exp(weight))

#resampling
sir.result.sample = sample(theta.sample, 5000, replace=TRUE, prob=weight.st)
# hist(sir.result.sample, nclass=100)



#hw4.2 : Rejection Sampling
theta.mle= mean(norm.txt.sample)
log_p_cal = function(candid){
        p=0
        for(i in 1:length(norm.txt.sample)){
            p = p + dnorm(norm.txt.sample[i], candid, 1, log=TRUE)-dnorm(norm.txt.sample[i], theta.mle, 1, log=TRUE)
        }
        return(p)
}
rj.result.sample = rep(NA, 5000)
gen.sample.num=0
while(gen.sample.num<5000){
    cauchy.sample = rcauchy(1,0,1)
    unif.sample = runif(1,0,1)
    log_p = log_p_cal(cauchy.sample)
```

```
        if(log(unif.sample)<log_p){
            # print('accept')
            gen.sample.num = gen.sample.num+1
            rj.result.sample[gen.sample.num] = cauchy.sample
        } else {
            # print('reject')
        }
}
# hist(rj.result.sample, nclass=100)


#4.3. compare
par(mfrow=c(1,2))
hist(sir.result.sample, nclass=100)
hist(rj.result.sample, nclass=100)
c(mean(sir.result.sample), mean(rj.result.sample))
c(var(sir.result.sample), var(rj.result.sample))
```

Result: 변수명 sir.result.sample 이 SIR 의 결과, 거.result.sample 이 rejection sampling 의 결과이다.

```
[R interpreter]
> c(mean(sir.result.sample), mean(rj.result.sample))
[1] 1.858147 1.851118
> c(var(sir.result.sample), var(rj.result.sample))
[1] 0.05191662 0.05172896
```



히스토그램 모양도, 평균/분산도 비슷함을 확인할 수 있다. 참고로 실행시간은 SIR 쪽이 훨씬 빠르다.