HW4MuyangShi

Muyang Shi

2024-02-29

Note: the cpp source code to this document can be found on my Github, listed as `MonteCarlo.cpp`, here.

Problem 1

(a) importance sampling with standardized weights

Use $f(x)$ to denote the actual density of X which we pretend we don't know, but we do knot that $f(x) = cq(x)$ where c is a constant; use $\phi(x)$ to denote an importance sampling function.

Note that

$$\begin{aligned}
\sigma^2 &= \frac{\int x^2 \frac{f(x)}{q(x)} \frac{q(x)}{\phi(x)} \phi(x) dx}{\int \frac{f(x)}{q(x)} \frac{q(x)}{\phi(x)} \phi(x) dx} \\
&= \frac{\int x^2 \alpha \frac{q(x)}{\phi(x)} \phi(x) dx}{\int \alpha \frac{q(x)}{\phi(x)} \phi(x) dx} \\
&= \frac{\int x^2 \frac{q(x)}{\phi(x)} \phi(x) dx}{\int \frac{q(x)}{\phi(x)} \phi(x) dx}
\end{aligned}$$

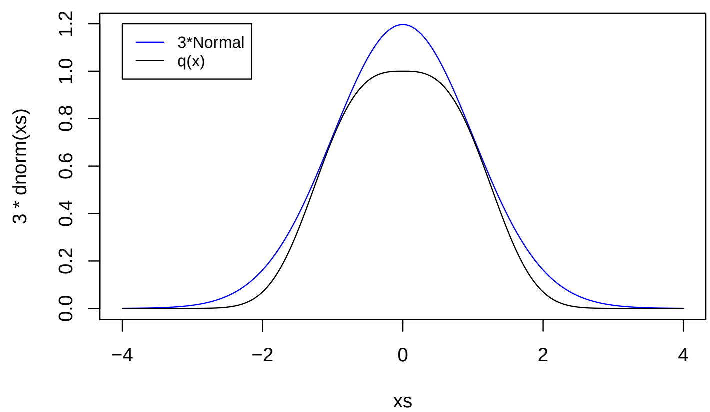
which means drawing $X_1, ..., X_n \overset{iid}{\sim} \phi(x)$ and using the estimator

$$\hat{\sigma}^2 = \sum_{i=1}^{n} X_i^2 w(X_i)$$

where

$$w(X_i) = \frac{w^*(X_i)}{\sum_{i=1}^{n} w^*(X_i)} \text{ with } w^*(X_i) = \frac{q(X_i)}{\phi(X_i)}$$

By plotting $q(x)$ and $e(x) = 3 \cdot \phi(x)$, where $\phi(x)$ is the probability density function of the standard Normal distribution, we can see that $e(x)$ can serve as an envelope (so we use $\phi(x)$ as our importance sampling function).



With $n = 10,000$ draws from the envelope $e(x)$, we estimated σ^2 to be 0.780.

```
set.seed(600)
x <- Rcpp_rnorm(10000); # draw from envelope Normal
x.q <- qx_vec(x) # corresponding q(x)
x.phi <- Rcpp_dnorm(x) # corresponding phi(x)
x.w_star <- w_star(x.q, x.phi) # unstandardized weights
sum.w_star <- sum(x.w_star)
x.w <- x.w_star/sum.w_star # standardized weights

sum((x^2)*x.w) # estimate
```

(b) rejection sampling

Note that from page 156 of the textbook, rejection sampling can still be applied when f is only known up to a proportionality constant c; we still use $e(x) = 3 \cdot \phi(x)$ as envelope and a draw $Y = y$ is rejected when $U > q(y)/e(y)$; the sampling probability remains correct because the unknown constant c cancels out.

Using rejection sampling, we got an estimation of $\hat{\sigma}^2 = 0.770$, with $n = 10,000$ (number of accepted samples).

```
set.seed(601)
rs <- rejection_sampling(10000)
mean(rs**2)
```

(c) sampling importance resampling

As with part (a), we still use $\phi(x)$ as our importance sampling function. Using sampling importance resampling, we got an estimation of $\hat{\sigma}^2 = 0.793$ with $n = 10,000$.

```
set.seed(602)
x <- Rcpp_rnorm(10000); # draw from IS Normal
x.q <- qx_vec(x) # corresponding q(x)
x.phi <- Rcpp_dnorm(x) # corresponding phi(x)
x.w_star <- w_star(x.q, x.phi) # unstandardized weights
sum.w_star <- sum(x.w_star)
```

```r
x.w <- x.w_star/sum.w_star # standardized weights

x.sir <- sample(c(x), size = 10000, replace = TRUE, prob = c(x.w))
mean(x.sir^2) # estimate
```

(d)

Using the samples from part (b) (the 10,000 rejecting sampling samples), the Philippe and Robert estimate is $\hat{\sigma}^2 = 0.777$.

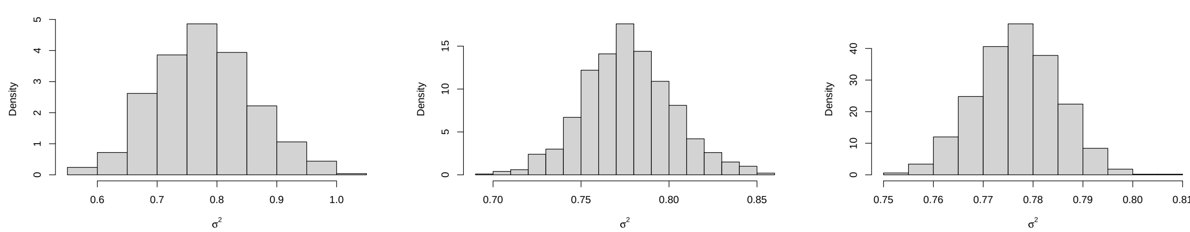
```r
x <- sort(c(rs))
PhilippeRobert(x)
```

(e)

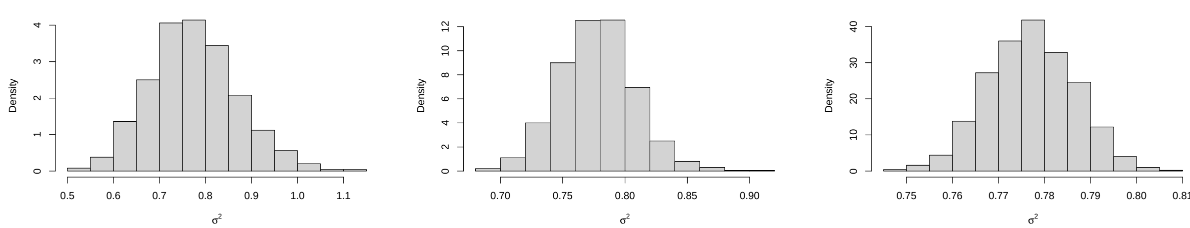
We generate 1000 estimates with each method, while also varying the number of sample taken with $n = 100, 1000, 10000$. We then compare the averages of the estimators and the variance of the estimators.

Below are the histograms showing the estimates from the four methods across the three different sample sizes:

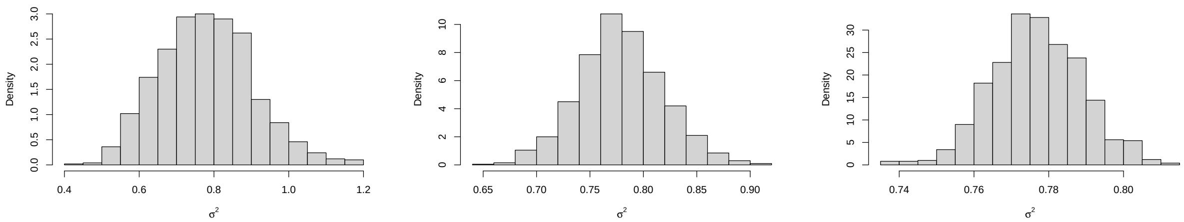
- Histograms of **Importance Sampling** estimates:



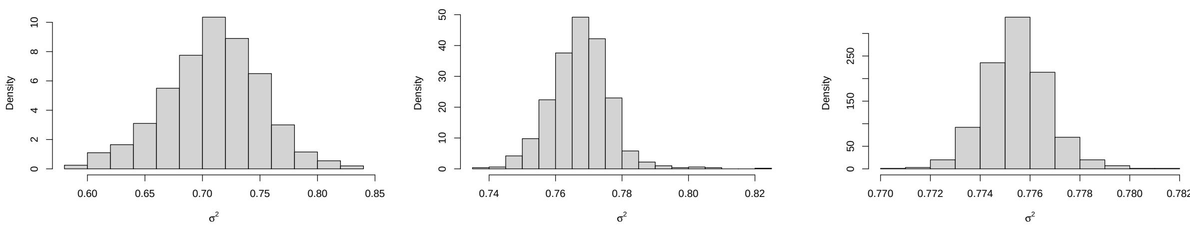
- Histograms of **Rejection Sampling** estimates:



- Histograms of **Importance Sampling Resampling** estimates:



- Histograms of **Philippe Robert** estimates:



Below the table summarizes the average estimates as well as the standard deviation of those estimates for the four methods across the three different sample sizes:

Table 1: Performance of the parameters estimation

	Mean(n=100)	SD(n=100)	Mean(n=1k)	SD(n=1k)	Mean(n=10k)	SD(n=10k)
importance sampling	0.778	0.081	0.777	0.025	0.777	0.008
rejection sampling	0.774	0.096	0.777	0.030	0.777	0.009
sampling importance resampling	0.779	0.125	0.779	0.040	0.777	0.012
Philippe and Rober	0.710	0.042	0.767	0.009	0.775	0.001