[ECO723] Assignment #1

Daeyoung Lim *Dept. of Statistics, Korea University*

Due : 2017-09-20

Question #3.1

Answer

If we let

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} N(0,1)\, I(X \geq 0)$$
$$Y_1, \ldots, Y_n \overset{\text{iid}}{\sim} N(0,1)\, I(Y < 0)$$

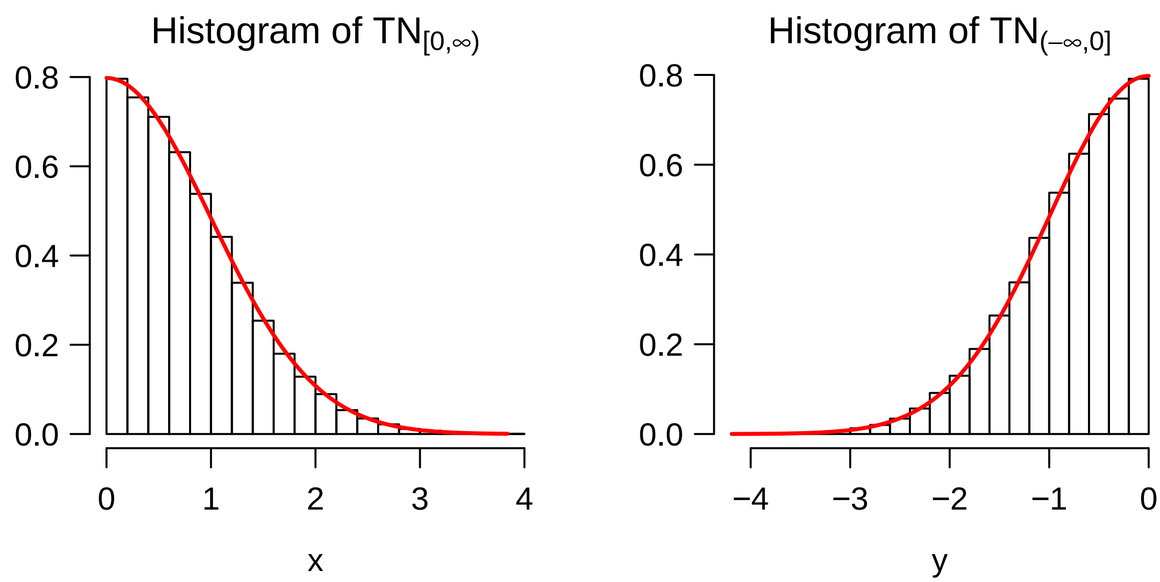
One way of sampling from a truncated normal distribution $N(\mu, \sigma^2)I(X \in [\alpha, \beta])$ is by *the probability integral transform* or the *inverse CDF method*. That is,

$$X = \Phi^{-1}(\Phi(\alpha) + U(\Phi(\beta) - \Phi(\alpha)))\sigma + \mu$$

where $U \sim \text{Unif}(0,1)$.

```r
library(latex2exp)
n = 30000L # sample size
t1 = pnorm(0, mean = 0, sd = 1)
t1[t1 > (1 - 1.0e-07)] = 1 - 1.0e-07
u1 = runif(n, min = 0, max = 1)
x = qnorm(t1 + u1 * (1 - t1), mean = 0, sd = 1)

t0 = pnorm(0, mean = 0, sd = 1)
t0[t0 < 1.0e-10] = 1.0e-10
u2 = runif(n, min = 0, max = 1)
y = qnorm(t0 * u2, mean = 0, sd = 1)
par(mar=c(3.5,3.5,1.5,1), mgp=c(2.4,0.8,0), las=1, mfrow=c(1,2))
hist(x, ylab = "", probability = TRUE, breaks = 20,
     main = TeX("Histogram of TN$_{\\[0, \\infty)}$"))
curve(2 * dnorm(x), from = 0, to = max(x), add = TRUE, col = "red", lwd = 2)
hist(y, ylab = "", probability = TRUE, breaks = 20,
     main = TeX("Histogram of TN$_{(-\\infty, 0\\]}$"))
curve(2 * dnorm(x), from = 0, to = min(y), add = TRUE, col = "red", lwd = 2)
```



Question #3.2

Answer

If $f(x) = x/2$,

$$F(x) = \int_0^x \frac{u}{2}\,\mathrm{d}u = \frac{x^2}{4}, \quad 0 \le x \le 2$$
$$F^{-1}(u) = \sqrt{4u}, \quad 0 \le u \le 1$$

Therefore, by the *probability integral transform*,

$$X = F^{-1}(U) = \sqrt{4U}, \quad U \sim \mathrm{Unif}(0,1)$$

Analytically, the expected value of X is

$$\mathrm{E}(X) = \int_0^2 \frac{x^2}{2}\,\mathrm{d}x = \left.\frac{x^3}{6}\right|_0^2 = \frac{4}{3}$$

Sampling $X_1, \ldots, X_n \overset{\text{iid}}{\sim} F$, we can use the law of large numbers to approximate $\mathrm{E}(X)$, i.e.,

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{n\to\infty} \mathrm{E}(X)$$

```r
n = 30000L
u = runif(n, min = 0, max = 1)
x = sqrt(4 * u)
Ex = mean(x)
cat("E(X) = ", 4/3, "\n",
    "Xbar = ", Ex, sep = "")
```

```
## E(X) = 1.333333
## Xbar = 1.336276
```

Question #3.3

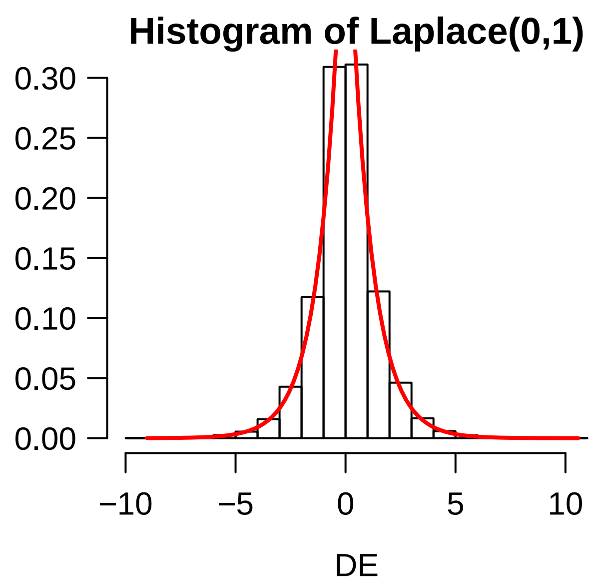
Answer

We know that $X \sim \text{Laplace}(0,1)$. The inverse CDF (or the quantile function) of $\text{Laplace}(0,1)$ is

$$F^{-1}(u) = -\,\text{sgn}\left(u - \frac{1}{2}\right) \log\left(1 - 2\left|u - \frac{1}{2}\right|\right)$$

Therefore, we can plug in $U \sim \text{Unif}(0,1)$ to generate random variables from $\text{Laplace}(0,1)$.

```
n = 20000L
u = runif(n, min = 0, max = 1)
DE = -sign(u - 0.5) * log(1 - 2 * abs(u - 0.5))
par(mfrow = c(1,1), mar=c(3.5,3.5,1,1), mgp=c(2.4,0.8,0), las=1)
hist(DE, ylab = "", probability = TRUE, breaks = 20,
     main = "Histogram of Laplace(0,1)")
curve(0.5 * exp(-abs(x)), from = min(DE), to = max(DE), add = TRUE, col = "red", lwd = 2)
```



Question #3.4

Answer

If $X \sim f$ is sought but a random variable with density g is much easier to generate, we can find β to satisfy a bound on the densities

$$\beta f(x) \leq g(x) \quad \text{for } 0 < \beta < 1$$

Since we are dealing with a mathematical optimization problem, we can do without all the constants multiplied, which reduces the objective function to

$$\hat{x} = \arg\min_{x \in \mathbb{R}} \frac{\exp\left(-(x - 1/2)^2/2\right)}{x^2(1-x)^2}$$

Moreover, the monotonically increasing function does not affect the minimizer of the function. Thus, applying the logarithm,

$$\hat{x} = \arg\min_{x\in\mathbb{R}} -\frac{1}{2}\left(x-\frac{1}{2}\right)^2 - 2\log x - 2\log(1-x)$$

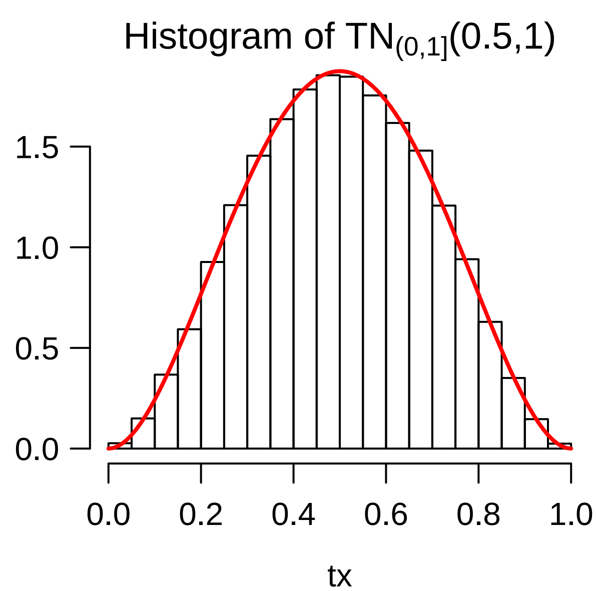
Then by the first-order condition, we find the root of the following equation

$$-\left(x-\frac{1}{2}\right) + \frac{2}{1-x} - \frac{2}{x} = 0$$

which gives $x = 1/2$ as the minimizer. Thus, $\hat{x} = 1/2$ and $c = 2$. Then, the acceptance-rejection (AR) algorithm goes by

- Generate X from g.
- Generate U from $\mathrm{Unif}(0,1)$.
- If $Ug(X) \le \beta f(X)$ then accept X; else reject and repeat from the beginning.

```
library(truncnorm)
n = 50000L
u = runif(n, min = 0, max = 1)
tx = rtruncnorm(n, a = 0, b = 1, mean = 0.5, sd = 1)
u = runif(n, min = 0, max = 1)
g = dtruncnorm(x = tx, a = 0, b = 1, mean = 0.5, sd = 1)
f = dbeta(tx, 3, 3)
M = dtruncnorm(0.5, a = 0, b = 1, mean = 0.5, sd = 1) / dbeta(0.5, 3, 3)
tx = tx[u * g < M * f]
par(mfrow = c(1,1), mar=c(3.5,3.5,1,1), mgp=c(2.4,0.8,0), las=1)
hist(tx, ylab = "", probability = TRUE, breaks = 20,
     main = TeX("Histogram of TN$_{(0,1\\]}(0.5, 1)$"))
curve(dbeta(x, 3, 3)
      , from = 0, to = 1, add = TRUE, col = "red", lwd = 2)
```



```
cat("Var(X) = ", 9 / (6^2 * 7), "\n",
    "MCvar(X) = ", var(tx), sep = "")
```

```
## Var(X) = 0.03571429
## MCvar(X) = 0.03586223
```

Question #3.6

Answer

To get the minimizer, we need to solve the following optimization problem:

$$\hat{x} = \arg\min_{x \in \mathbb{R}} \frac{x^2}{2} - |x|$$

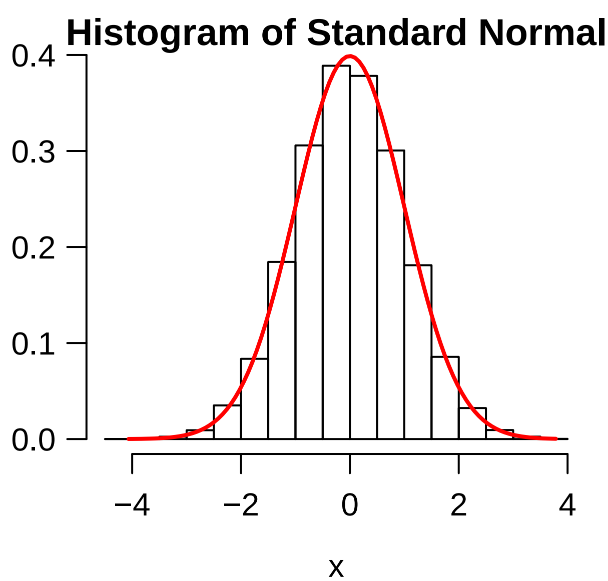
By differentiating the objective function with respect ot x,

$$x - \frac{x}{|x|} = 0$$

We get $x = -1$ or $x = 1$ as the minimizers. Thus, $\hat{x} = -1$ or 1 and $\beta = \sqrt{\pi/(2e)}$ which means $c = \sqrt{2e/\pi}$. Thus,

```
par(mfrow = c(1,1), mar=c(3.5,3.5,1,1), mgp=c(2.4,0.8,0), las=1)
n = 50000L
u = runif(n, min = 0, max = 1)
DE = -sign(u - 0.5) * log(1 - 2 * abs(u - 0.5))

u = runif(n, min = 0, max = 1)
M = 0.5 * exp(-1) / dnorm(1)
stdNormal = DE[u * (0.5 * exp(-abs(DE))) <= M * dnorm(DE)]
hist(stdNormal, probability = TRUE,
     main = "Histogram of Standard Normal", ylab = "", xlab = "x")
curve(dnorm(x), from = min(stdNormal), to = max(stdNormal),
      add = TRUE, col = "red", lwd = 2)
```



```r
cat("E(x) = ", mean(stdNormal), "\n",
    "Var(x) = ", var(stdNormal), sep = "")
```

```
## E(x) = -0.007003033
## Var(x) = 0.9921894
```