SDS 5531 Homework 1

Gifty Osei

2024-09-11

Remark: If you would like to insert images for your handwritten part into this file, please refer to this article.

Problem 1. Box-Muller transformation

The Box-Muller transformation method simulates random numbers from N(0,1) as follows.

- Step 1: Generate U_1 and U_2 i.i.d. from U(0,1).
- Step 2: Let $X_1 = \sqrt{-2 \log U_1} \cos(2\pi U_2)$ and $X_2 = \sqrt{-2 \log U_1} \sin(2\pi U_2)$.

Establish the theoretical validity of the method by proving the following results.

1. (15 points) Use the change-of-variable formula to derive that X_1 and X_2 are two independent draws from N(0,1).

Solution:

	fr. (r, 0) = fulle (4,16) . []
1. To prove that X1 and X2 are independent draws	
from M(O1)	$\begin{vmatrix} \frac{\partial \Omega^1}{\partial x^2} & \frac{\partial \Omega^2}{\partial x^3} \\ \frac{\partial \Omega^1}{\partial x^2} & \frac{\partial \Omega^2}{\partial x^3} \end{vmatrix} = \frac{\partial \Omega^1}{\partial x^1} \cdot \frac{\partial \Omega^2}{\partial x^2} - \frac{\partial \Omega^1}{\partial x^2} \cdot \frac{\partial \Omega^2}{\partial x^3} \cdot \frac{\partial \Omega^2}{\partial x^4}$
use Chara that	$\left \frac{\partial u}{\partial x}\right = \frac{\partial x}{\partial x} \cdot \frac{\partial x}{\partial x} - \frac{\partial x}{\partial x} \cdot \frac{\partial x}{\partial x}$
$f_{x_1x_2}(x_1,x_2) = f_{x_1}(x_1) \cdot f_{x_2}(x_2)$	\\ \frac{\darka}{\darka} \frac{\darka \text{X2}}{\darka \text{VL}} \frac{\darka \text{VL}}{\darka \text{VL}}{\darka \text{VL}} \frac{\darka \text{VL}}{\dark
	(80, 80)
Using transformation; we recall;	$f_{R_{\theta}}(r_{1}\theta) = f_{R}(r) \cdot f_{\theta}(\theta)$
71 = 1-2 log U, Cos (2TU2)	$=\frac{1}{2\pi}e^{-\frac{r^2}{2}}$
12 = 1-2/09 U, Sin (211 U2)	
find the	then:
$(U_1, U_2) \longrightarrow (X_1, X_2)$	$f_{x_{1}x_{2}}(z_{1}z_{0}) = \frac{1}{2\pi} e^{-(z_{1}^{2}+x_{1}^{2})}$
using thange of variables.	2π
using thange of variables. Let $R = \sqrt{-2\log U_1}$ $\theta = 2\pi U_2$	$= g(x_1) \cdot h(x_2)$
	$= g(x_1) \cdot h(x_2)$ Hence x_1 and x_2 are independent x_1 .
radius angle	
$f_{v_1,v_2}(v_1,v_2) = 1$, for $v_1 \in (o_1)$, $v_2 \in (o_1)$	
by the invese transformation;	
$U_1 = e^{\frac{-e^2}{2}}$ $U = \frac{\theta}{2}$	
27	
by the bivariate transformation)	

Figure 1: Problem 1, Question 1

2. (10 points) Show that the polar coordinates $r^2=X_1^2+X_2^2\sim\chi_2^2$, hence $e^{-\frac{r^2}{2}}\sim U(0,1)$, and $\theta=\arctan\frac{X_1}{X_2}\sim U(0,2\pi)$.

Solution:

Let $R = \sqrt{-2\log V_1}$ and $\theta = 2\pi V_2$	How given ratio x_1 , $x_1 + x_2 \sim H(x_1)$
then: X1 = RCOS & and X2 = RSin O	Xa
1, 1/2 ~ H(011); we can show that	0 is the angle Gordinate For an itel Standard normal
$\chi_1^2 + \chi_2^2 = \Gamma^2$ and	random variables so i
from 1, we show that X_1 X_2 \sim $M(o_{11})$ then recall that the sum	$f_{\theta}(\theta) = \int_{-\pi}^{\pi} Q \in [0,2\pi]$
of 2 independent Standard normal random variables follows	2.∏
a chi-Squared with 2 degrees of freedom So;	recall that X1 = Cost
$r^{2} = \chi_{1}^{2} + \chi_{2}^{2} \sim \chi_{2}^{2}$	Ya r Sin D
	χı = tanθ
New; $\mathcal{U}_1 = e^{-r^2/2}$ Then $r^2 = -2\log U$, and $U_1 \sim U(0_1) \leq e^{-r^2/2}$ $\therefore P(U_1 \leq U) = P[e^{-r^2/2} \leq U]$	to-
P(U, <u) =="" p[e2="u]</td"><td>$\therefore \theta = \operatorname{qrctan}\left(\frac{\chi_1}{\chi_2}\right)$</td></u)>	$\therefore \theta = \operatorname{qrctan}\left(\frac{\chi_1}{\chi_2}\right)$
= P [12 ≤ -2log V]	(X 2)
$= \int_{-\frac{\pi}{2}}^{\pi} \frac{1}{2} e^{\frac{\pi}{2}} dx \qquad \text{for } R + \frac{\pi}{4} + \frac{\pi}{4} + \frac{\pi}{4}$	
-2/oqU,	
$P[V_i \leq u] = \frac{-2\log V_i}{2} = V \square$	
e -	
: fu,(u) = 4 then it follows that U, & (o,1) hence	
V, ~ Uniform (0,1)	

Figure 2: Problem 1, Question 2

Problem 2. Generate Cauchy random numbers

The Cauchy distribution is Student's t-distribution with 1 degree of freedom.

1. (10 points) Derive an algorithm to simulate random numbers from the Cauchy distribution using the inverse cdf approach. (Hint: Show the Cauchy cdf is $F(x) = \tan^{-1}(x)/\pi$.)

Solution:

knitr::include graphics("D:/WashU/First Year/Sem1/SDS5071 AdvLinearModel/Homework/HW2/Q2a.pdf")

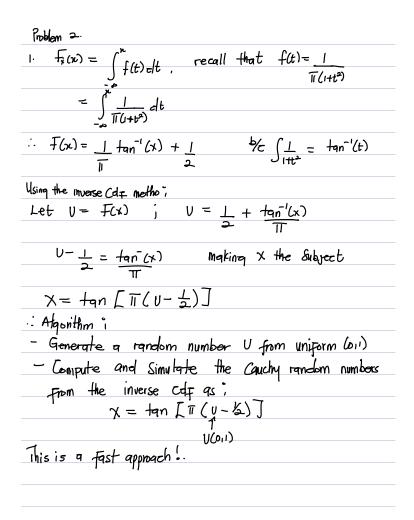


Figure 3: Problem 2, Question 1

2. (10 points) Alternatively, one can simulate from the Cauchy distribution by computing the ratio $\frac{X_1}{X_2}$, where X_1 and X_2 are two independent N(0,1) random variables. Explain why this works.

Solution:

knitr::include_graphics("D:/WashU/First Year/Sem1/SDS5071_AdvLinearModel/Homework/HW2/Q2b.pdf")

2. We know that
$$x_1, x_2 \sim N(o_1)$$
 and we have proved that x_1 and x_2 are independent c_0 . Let $Y_1 = x_1$ by $Y_2 = x_2$. Using transformation than; x_2 for $(y_1, y_2) = f_{x_1 x_2}(y_1 y_2, y_2) \cdot |J|$

$$= \underbrace{1}_{21\overline{1}} \exp\left[-\frac{y_2}{2}(y_1^2 + 1)\right] |y_2|$$

$$= \underbrace{1}_{21\overline{1}} \exp\left[-\frac{y_2}{2}(y_1^2 + 1)\right] |y_2|$$
finding the marginal of Y_1 which is the ratio;
$$f_{Y_1}(y_1) = \underbrace{\int_{21\overline{1}} y_2 \exp\left[-\frac{y_2}{2}(y_1^2 + 1)\right] dy_2}_{21\overline{1}}$$

$$= \underbrace{2}_{21\overline{1}} \underbrace{\int_{21\overline{1}} y_2 \exp\left[-\frac{y_2}{2}(y_1^2 + 1)\right] dy_2}_{21\overline{1}}$$

$$= \underbrace{2}_{1} \underbrace{\int_{21\overline{1}} 1 + y_2^2}_{21\overline{1}}$$
but hence, this is why simulating the ratio of Y_1 independent $Y_1(o_1)$ works.

Figure 4: Problem 2, Question 2

3. (20 points) Implement these two methods in R (or Python). Then compare their computing time and efficiency by simulating n Cauchy random numbers. (Choose n to be reasonably large to be able to tell the time difference in running the two methods. The exact choice of n depends on your hardware and implementation.)

Solution:

```
# 1. Implement the two methods. (Make sure your implementation is efficient. For example, avoid loops i
# Part 1
# Using the Inverse CDF to generate Cauchy random numbers
inv_cdf_cauc <- function(n) {</pre>
```

```
# Generate n uniform random numbers between 0 and 1
 u <- runif(n)
# Using inverse CDF formula
 x \leftarrow tan(pi * (u - 0.5))
 return(x)
# 2. Simulate n Cauchy random numbers and compare the execution time of the two methods. (You can find
# Part 2
# Ratio of 2 standard Normals to generate Cauchy RV.
ratio_of_norms_cauc <- function(n) {</pre>
# Generate n N(0,1) numbers as x1
 X1 <- rnorm(n)</pre>
# Generate n N(0,1) numbers as x2
 X2 <- rnorm(n)</pre>
# ratio of the two
  x <- X1 / X2
 return(x)
# Part 3
## Comparing time to compute
library(microbenchmark)
# set a reasonable n
n <- 10000
# Compare the two methods using micro benchmark
benchmark_results <- microbenchmark(</pre>
 inverse_cdf = inv_cdf_cauc(n),
 ratio_of_stan_normals = ratio_of_norms_cauc(n),
  times = 10
## Summarize
kable(summary(benchmark_results), caption = "Time to Compute",
      latex_options = c("hold_position",
                         "striped" ))
```

Table 1: Time to Compute

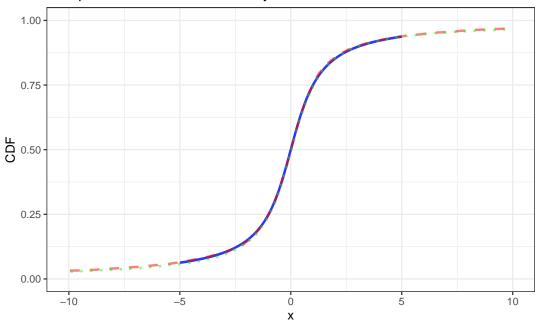
expr	min	lq	mean	median	uq	max	neval
inverse_cdf	223.6	335.4	711.95	340.7	351.2	4180.5	10
$ratio_of_stan_normals$	701.7	935.4	1297.03	1118.8	1209.5	3691.5	10

```
## We can see that inverse CDF is faster
```

Table 1: From the descriptive summary, we can see that inverse CDF is doing much better in time to compute and generate the Cauchy random variables

```
N < -2000
system.time(for (i in 1:N) inv_cdf_cauc(n))
system.time(for (i in 1:N) ratio_of_norms_cauc(n))
# 3. For both methods, draw the empirical cdf of your simulated numbers and see how close it is to the
## Plotting All 3 plots (Cauchy, Ratio, Inverse CDF)
## Empirical Values from inverse cdf and ratio
cauchy_samples <- list(</pre>
  inverse = inv cdf cauc(10000),
 ratio = ratio_of_norms_cauc(10000)
## X-values
x_values \leftarrow seq(-5, 5, length.out = 10000)
# Calculate the true Cauchy CDF values
true_cdf <- pcauchy(x_values)</pre>
## Create Data frame
empirical_cdfs <- map(cauchy_samples, function(samples) {</pre>
 data.frame(
    x = sort(samples),
    cdf = seq(1/n, 1, length.out = n)
 )
})
data_true <- data.frame(x = x_values, cdf = true_cdf)</pre>
\#data\_inverse \leftarrow data.frame(x = empirical\_cdfs\$inverse, cdf = seq(1/n, 1, length.out = n))
\#data\_ratio \leftarrow data.frame(x = empirical\_cdfs\$ratio, cdf = seq(1/n, 1, length.out = n))
ggplot() +
  geom_line(data = data_true, aes(x = x, y = cdf), color = "blue",
            size = 1, linetype = "solid", alpha = 0.8) +
```

Comparison of Simulated Cauchy Distribution CDFs



Blue: True Cauchy CDF, Red: Inverse CDF Method, Green: Ratio of Normals Method

Figure 5: Plot showing the true, inverse cdf and ratio of normals approach of simulating cauchy random number. As n gets bigger over time, both simulation approach converges to the true cdf from a cauchy distribution.

Problem 3. Accept-Reject sampling

Consider simulating from N(0,1) using the accept-reject sampling. Pretend you do not know the normalizing constant of the pdf, so $f(x) = e^{-\frac{x^2}{2}}$. First, consider using the standard Cauchy distribution as an envelope distribution. Let $g(x) = \frac{1}{1+x^2}$. (Note we have dropped the normalizing constant in the Cauchy pdf.)

1. (10 points) Show that the ratio

$$\frac{f(x)}{g(x)} = (1+x^2)e^{-\frac{x^2}{2}} \le \frac{2}{\sqrt{e}},$$

with the equality attained at $x = \pm 1$.

Solution:

knitr::include graphics("D:/WashU/First Year/Sem1/SDS5071 AdvLinearModel/Homework/HW2/Q3a.pdf")

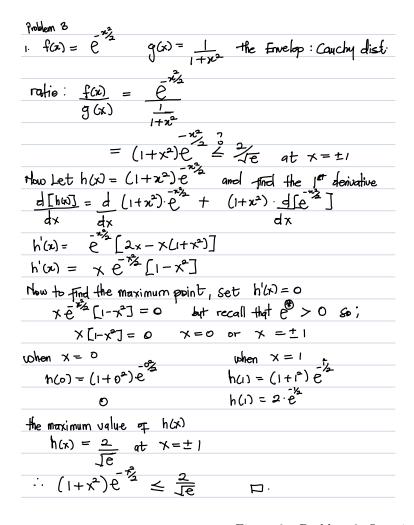


Figure 6: Problem 3, Question 1

2. (15 points) Show that the probability of acceptance is $\sqrt{\frac{e}{2\pi}} \approx 0.66$. Also run an empirical evaluation of the probability of acceptance.

Solution:

knitr::include_graphics("D:/WashU/First Year/Sem1/SDS5071_AdvLinearModel/Homework/HW2/Q3b.pdf")

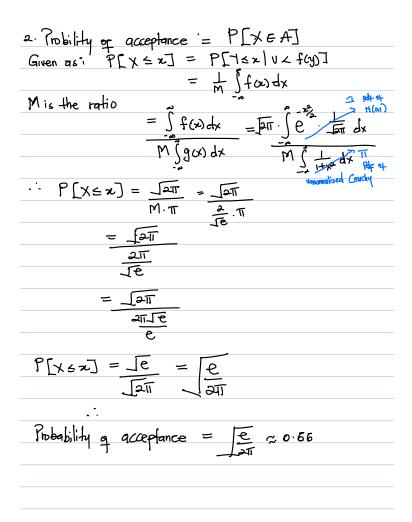


Figure 7: Problem 3 Question 2

```
# 1. implement your algorithm and record the number of acceptances or
# rejections 2. Simulate n=1000 random numbers, and find the empirical
# proportion of acceptances or rejections

set.seed(300)

# Number of simulations
n <- 10000

# Generate n samples from the standard Cauchy distribution using vectorized</pre>
```

```
# operations
x <- rcauchy(n)
# target (unnormalized standard normal)
f_x \leftarrow \exp(-x^2/2)
# envelope density (unnormalized standard Cauchy)
g_x < 1/(1 + x^2)
# ratio M
M <- 2/sqrt(exp(1))</pre>
# Acceptance probability
acceptance_prob <- f_x/g_x</pre>
# simulate uniform random numbers for comparison
u <- runif(n)
\# Accept if u is less or equal to acceptance_prob / M
accepted <- u <= (acceptance_prob/M)</pre>
# empirical proportion of acceptances
acceptance_rate <- mean(accepted)</pre>
# acceptance_rate gives the proportion of acceptance printed below
```

Based on the simulation example above, we can show that the empirical proportion of acceptance for a unnormalized Standard Normal and Cauchy is 0.66

3

. (10 points) Now, consider using a scaled Cauchy distribution as the envelope distribution, i.e. $g_{\sigma}(x) =$ $\frac{1}{\pi\sigma(1+\frac{x^2}{\sigma^2})}$. Find the upper bound for $\frac{f(x)}{g_{\sigma}(x)}$ and the value of σ that minimizes this bound.

Solution:

knitr::include_graphics(c("D:/WashU/First Year/Sem1/SDS5071_AdvLinearModel/Homework/HW2/Q3c.pdf",

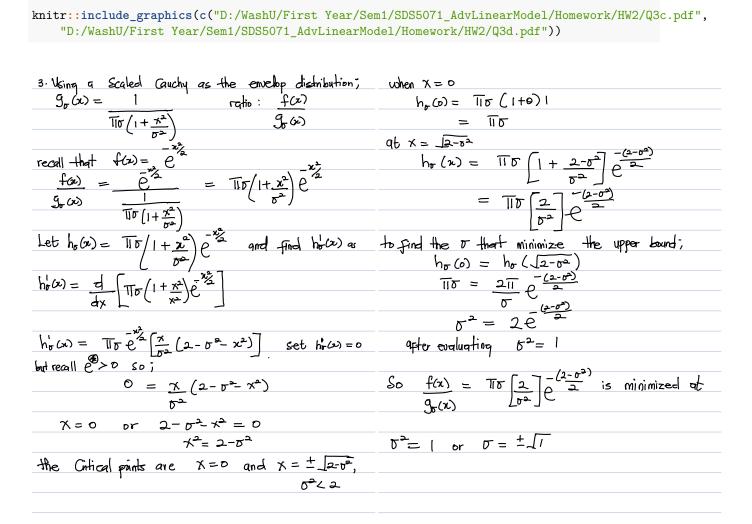


Figure 8: Problem 3, Question 3