Acceptance-Rejection method: rationale

The Acceptance-Rejection method is an algorithm used to generate random variates from a distribution for which it may be difficult to sample from. Let X and Y be two r.v. with respective PDF f and g, and let c be a constant such that, for all x, we have

$$f(x) \leq cg(x)$$

For the univariate case, the algorithm proceeds as follows:

1. Generate x (an abcissa) from a Proposal distribution, easy to sample from.
2. Generate y (an ordinate) uniformly in the interval $[0, cg(x)]$
$c \leq$ max. $f(x)$, $g(x) =$ proposal distribution
3. Accept x if $y \leq f(x)$, else repeat,
$f(x) =$target distribution, hard to sample from.

Example 1 (1/5)

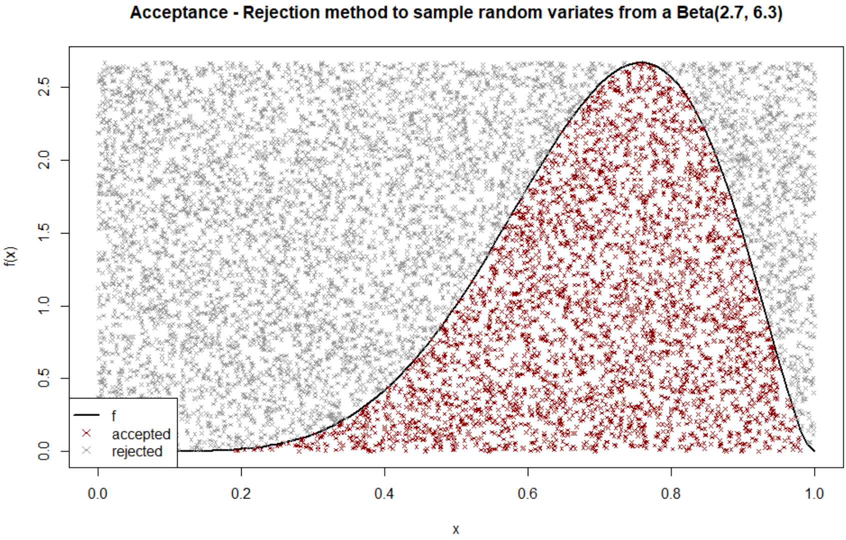
Suppose we want to sample from a Beta distribution $B(6.3, 2.7)$. We will show how to use the Acceptance-Rejection algorithm to sample from this distribution using $n = 10,000$ simulations.

To sample from the Beta distribution $B(6.3, 2.7)$, the Acceptance-Rejection algorithm proceeds as follows:

1. Generate x **(an abcissa) from a Uniform distribution** $U[0, 1]$
2. Generate y **(an ordinate) uniformly in the interval** $[0, cg(x)]$
$c = 2.67$ $\quad$ $g(x) =$ Uniform PDF
3. Accept x **if** $y \leq f(x)$**, else repeat** $\quad$ f(x) $= B(6.3, 2.7)$ PDF

Theoretical mean: $E[X] = \frac{\alpha}{\alpha+\beta} = \frac{6.3}{6.3+2.7} = 0.7$
Acceptance-Rejection approximation ≈ 0.698

Example 1 (2/5)



Acceptance - Rejection method to sample random variates from a Beta(2.7, 6.3)
f(x)
0.0
0.5
1.0
1.5
2.0
2.5
x
0.0
0.2
0.4
0.6
0.8
1.0
f
accepted
rejected

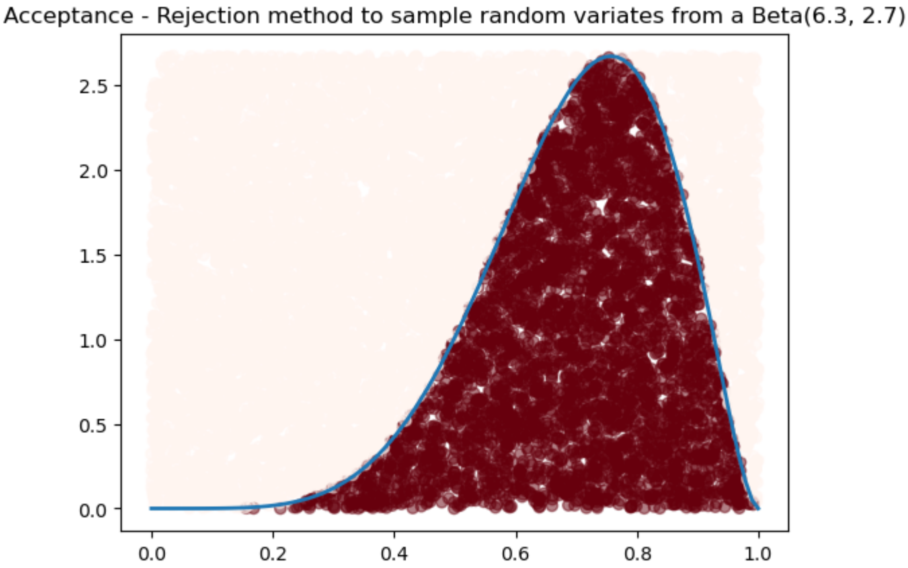
Example 1 - R code (3/5)

```
# We use Accept-Reject to estimate the mean and variance of a beta distribution
n = 10000  # number of draws
f = function(x) {dbeta(x,6.3,2.7)}  # target distribution
c = 2.67  # maximum, found using for ex:
optimize(f, interval=c(0, 1), maximum=TRUE)
g = dunif   # proposal distribution

# Accept-Reject algorithm
theta=c()
for (i in 1:n){
  x <- runif(1, 0, 1)
  y <- runif(1, 0, c)
  acceptance_rate <- f(x) / (c*g(x)) # acceptance probability
  if (y <= acceptance_rate){
    theta[i] <- x
  }
  else {next}
}

points = runif(n); uniforms = runif(n); accept = uniforms < (f(points)/(c*g(points)))
# plotting
curve(f, lwd=2, main="Acceptance - Rejection method to sample random variates from a Beta(2.7, 6.3)")
points(points, c*uniforms, pch=ifelse(accept,4,4), col=ifelse(accept,"darkred","gray60"), cex=0.5)
legend("bottomleft", c("f","accepted","rejected"),
       lwd=c(2,NA,NA), col=c("black","darkred","gray60"),
       pch=c(NA,4,4), bg="white")
```

Example 1 (4/5)


Acceptance - Rejection method to sample random variates from a Beta(6.3, 2.7)
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Example 1 - Python code (5/5)

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import beta
from scipy.stats import uniform

# Function to calculate the target distribution (beta)
def f(x):
    return beta.pdf(x, a = 6.3, b = 2.7)

n = 10000  # number of draws
c = 2.67  # maximum

# Accept-Reject algorithm
theta = []
for i in range(n):
    x = np.random.uniform(0, 1)
    y = np.random.uniform(0, c)
    acceptance_rate = f(x) / (c * np.random.uniform(0, 1))
    if y <= acceptance_rate:
        theta.append(x)

points = np.random.uniform(0, 1, n); uniforms = np.random.uniform(0, 1, n);
     accept = uniforms < (f(points) / (c * uniform.pdf(points)))

# Plotting
x_vals = np.linspace(0, 1, 100)
plt.plot(x_vals, f(x_vals), linewidth=2, label='f')
plt.scatter(points, c*uniforms, marker='o', c=accept, cmap='Reds', alpha=0.5)
plt.title("Acceptance - Rejection method to sample random variates from a Beta
     (6.3, 2.7)")
plt.show()
```

Example 2 (1/9)

Suppose that X is the survival time of some butterflies and that $X \sim Exp(\theta)$, with PDF and CDF given respectively by

$$f_\theta(x) = \theta e^{-\theta x} \qquad F_{-\theta}(x) = 1 - e^{-\theta x}$$

In addition, suppose that we observe the following survival time

number of week	0	1	2	3	4	5
number of butterflies alive	10	6	3	2	2	0
number of butterflies dead	0	4	3	1	0	2

(i) What is the posterior distribution of the data if we do not want to be informative and use a Jeffrey's prior?
(ii) What is the mean survival time and a 95 % Credible Interval for the mean survival time using Acceptance-Rejection method?

Example 2 (2/9)

The probability of surviving 'one more week' is $e^{-\theta}$ and the probability of dying is $1 - e^{-\theta}$. So, the likelihood is given by

$$p(x \mid \theta) \propto exp(-\theta)^{\sum_{i=1}^{alive}} \Big(1 - exp(-\theta)\Big)^{\sum_{i=1}^{dead}}$$

$$p(x \mid \theta) \propto exp(-13\theta)\Big(1 - exp(-\theta)\Big)^{10}$$

Since the Jeffrey's prior is $p(\theta) = 1/\theta$, we conclude that the posterior is given by

$$p(\theta \mid x) \propto \ p(x \mid \theta)\ p(\theta) \propto \frac{1}{\theta}\ exp(-13\theta)\Big(1 - exp(-\theta)\Big)^{10}$$

Example 2 (3/9)

Since we are interested in the expected survival time (EST) ($E[X] = 1/\theta$), we will use Monte Carlo integration to compute

$$MST = \int_{\mathbb{R}^+} \frac{1}{\theta}\, p(\theta \mid x)\, d\theta$$

To sample from the posterior $p(\theta \mid x)$, we proceeds as follows:

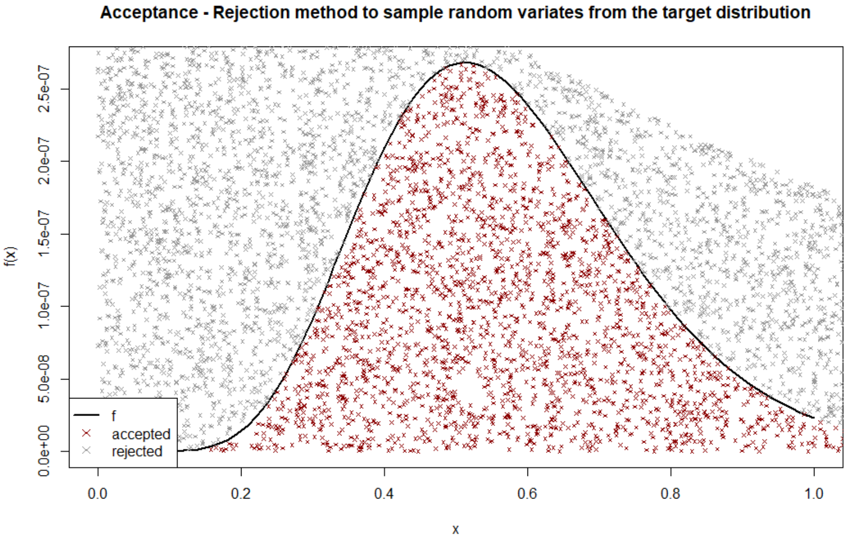
1. Generate x (an abcissa) from an Exponential distribution $E(\theta = 1)$

2. Generate y (an ordinate) uniformly in the interval $[0, cg(x)]$
$c = 5 * 10^{-7}$ $\qquad g(x) = exp(-x)$

3. Accept x if $y \leq f(x)$, else repeat the posterior
$f(x) = 1/x * exp(-13x) * (1 - exp(-x))^{10}$

Using $n = 15,000$ simulations, the Acceptance-Rejection approximation yields $EST \approx 1.9469$ $\qquad 95\%$ CI : $EST \in [1.015, 3.675]$

Example 2 (4/9)



Acceptance - Rejection method to sample random variates from the target distribution
f(x)
0.0e+00
5.0e-08
1.0e-07
1.5e-07
2.0e-07
2.5e-07
0.0
0.2
0.4
0.6
0.8
1.0
x
f
accepted
rejected

Example 2 - R code (5/9)

```
set.seed(2023)
# we want to learn about the mean survival time of butterflies using bayesian analysis
n = 10000  # number of draws
# density of exponential distribution with parameter theta = 1
g <- function(x) return( exp(-x) ) # proposal, easy to sample from

# posterior distribution of the data
# 13 = 6 + 3 + 2 + 2 : number of butterflies alive
# 10 = 4 + 3 + 1 + 0 + 2 : number of butterflies dead
# 1/x is the Jeffrey's prior
# exp(-13*x) * (1-exp(-x))^10 ) is the likelihood of the data
f <- function(x) return( 1/x *exp(-13*x) * (1-exp(-x))^10 ) # target (posterior), hard to sample from

optimize(f, interval=c(0, 100), maximum=TRUE)
c <- 5 * 10^(-7)

# Accept-Reject algorithm
theta=c()
for (i in 1:n){
  x <- runif(1, 0, 1)
  y <- runif(1, 0, c)
  acceptance_rate <- f(x) / (c*g(x)) # acceptance probability
  if (y <= acceptance_rate){
    theta[i] <- x
  }
  else {next}
}
```

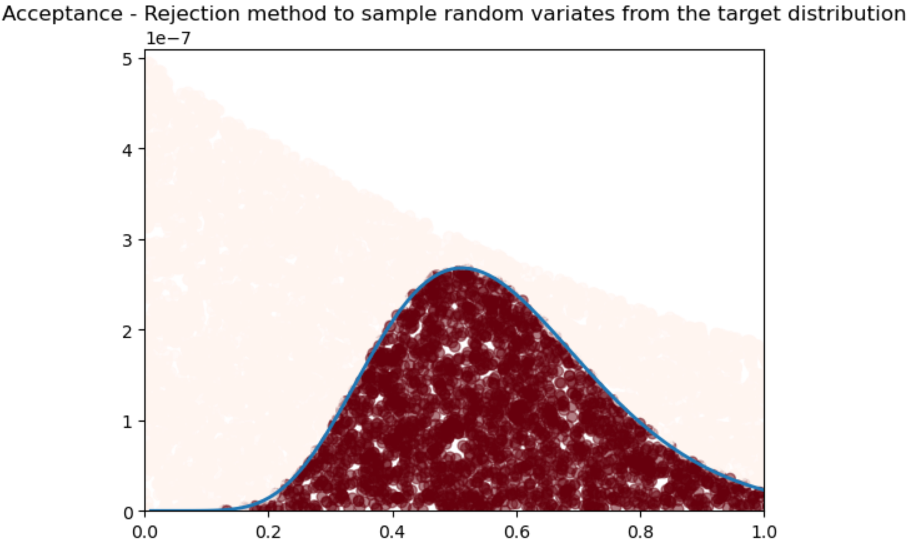
Example 2 - R code (6/9)

```
cat("Estimated EST is equal to", 1/mean(theta, na.rm=TRUE))
# Estimated EST is equal to 1.946919

points = rexp(n)
uniforms = runif(n, min = 0, max = c*g(points))
accepted = uniforms < f(points)

# plotting
curve(f, lwd=2, main="Acceptance - Rejection method to sample random variates
     from the target distribution")
points(points, uniforms, pch=ifelse(accepted,4,4), col=ifelse(accepted,"darkred"
     ,"gray60"), cex=0.5)
legend("bottomleft", c("f","accepted","rejected"),
       lwd=c(2,NA,NA), col=c("black","darkred","gray60"),
       pch=c(NA,4,4), bg="white")
```

Example 2 (7/9)


Acceptance - Rejection method to sample random variates from the target distribution
1e−7
5
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0

Example 2 - Python code (8/9)

```python
import numpy as np
from numpy import random
import matplotlib.pyplot as plt
from scipy.stats import uniform

# random seed
np.random.seed(2023)

# Function to calculate the target distribution (beta)
def f(x):
    return (1/x *np.exp(-13*x) * (1-np.exp(-x))**10 )

def g(x):
    return (np.exp(-x))

n = 10000  # number of draws
c = 5 * 10**(-7) # maximum

# Accept-Reject algorithm
theta = []
for i in range(n):
    x = np.random.uniform(0, 1)
    y = np.random.uniform(0, c)
    acceptance_rate = f(x) / (c * g(x))
    if y <= acceptance_rate:
        theta.append(x)
```

Example 2 - Python code (9/9)

```
# Calculate Expected Survival Time (EST)
mean_theta = 1/np.mean(theta)
mean_theta
1.935258913103203

points = np.random.exponential(size = n)
uniforms = np.random.uniform(0, c*g(points), n)
accepted = uniforms < f(points)

# Plotting
x_vals = np.linspace(0, 1, 100)
plt.plot(x_vals, f(x_vals), linewidth=2, label='f')
plt.scatter(points, uniforms, marker='o', c=accepted, cmap='Reds', alpha = 0.5)
plt.xlim([0, 1])
plt.ylim([0, c+0.00000001])
plt.title("Acceptance - Rejection method to sample random variates from the
     target distribution")
plt.show()
```

References

Rizzo, M.L. (2019). Statistical Computing with R, Second Edition (2nd ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9780429192760

The R Project for Statistical Computing: https://www.r-project.org/

Python: https://www.python.org/

course notes