Homework 1

Quan Nguyen

January 2026

See https://github.com/nguyencquan/Markov2026 for code used in simulations.

1 Power Law Sampling

Assume we have some distribution

$$\begin{cases} Cx^{-\gamma} & x \geq x_0, \\ 0, & x < x_0 \end{cases} \tag{1}$$

a) Where $x_0 > 0$ and $\gamma > 3$
A valid pdf must have a total probability of 1, which is calculated below:

$$1 = \int_{-\infty}^{\infty} Cx^{-\gamma} dx = C \int_{x_0}^{\infty} x^{-\gamma} dx = \frac{C}{x^{\gamma-1}(1-\gamma)} |_{x_0}^{\infty} = \frac{C}{x_0^{\gamma-1}(\gamma-1)} \tag{2}$$

$$C = x_0^{\gamma-1}(\gamma-1) \tag{3}$$

b) First calculating the cummulative distribution we take a step from equation 2 except take the integral from x_0 to x instead.

$$1 = \int_{-\infty}^{\infty} Cx^{-\gamma} dx = C \int_{x_0}^{x} x^{-\gamma} dx = \frac{C}{x^{\gamma-1}(1-\gamma)} |_{x_0}^{x} = \frac{C}{x_0^{\gamma-1}(\gamma-1)} + \frac{C}{x^{\gamma-1}(1-\gamma)} \tag{4}$$

Substituting in C

$$F(x) = \begin{cases} 1 - \frac{x_0^{\gamma-1}}{x^{\gamma-1}}, & x \geq x_0 \\ 0, & x < x_0 \end{cases} \tag{5}$$

To calculate the inverse, note we can do root math since $x_0 > 0$

$$p = 1 - \frac{x_0^{\gamma-1}}{x^{\gamma-1}} \tag{6}$$

$$\frac{x_0^{\gamma-1}}{x^{\gamma-1}} = 1 - p \tag{7}$$

$$\frac{x_0}{x} = (1-p)^{\frac{1}{\gamma-1}} \tag{8}$$

$$x = \frac{x_0}{(1-p)^{\frac{1}{\gamma-1}}} \tag{9}$$

c) Also we can note that we can use $U(0,1)$ without $1-p$ to speed it up since their distribution is exactly the same.
1. Sample from a normal distribution $U(0,1)$
2. substitue value into p for $\frac{x_0}{p^{\frac{1}{\gamma-1}}}$

d)

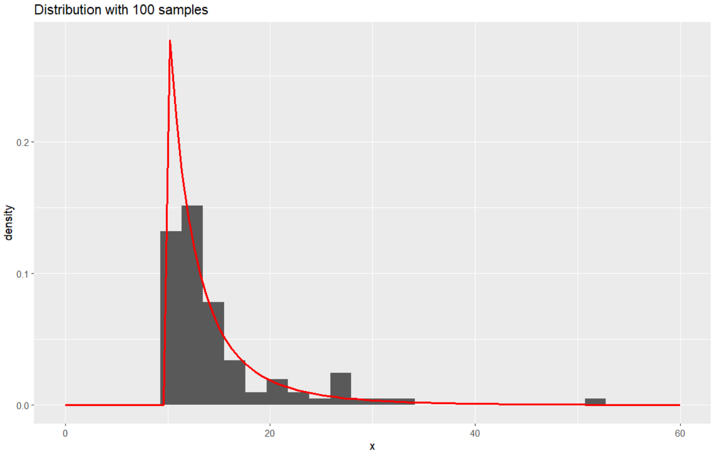

Figure 1: Graph of pdf and random sampling of 100 points, given $\gamma = 4$ and $x_0 = 10$

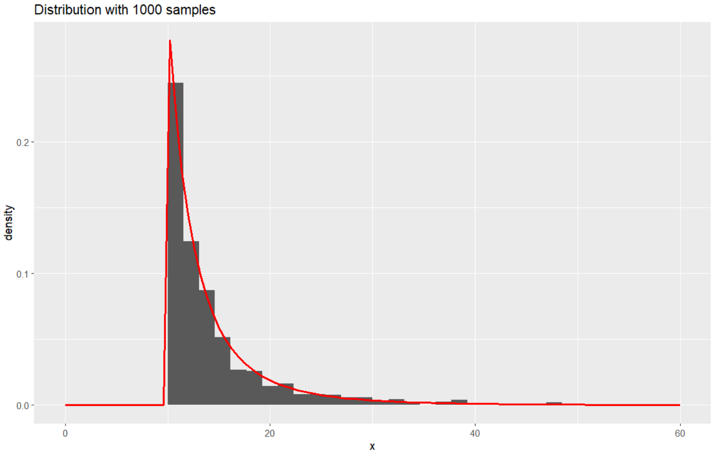

Figure 2: Graph of pdf and random sampling of 1000 points, given $\gamma = 4$ and $x_0 = 10$

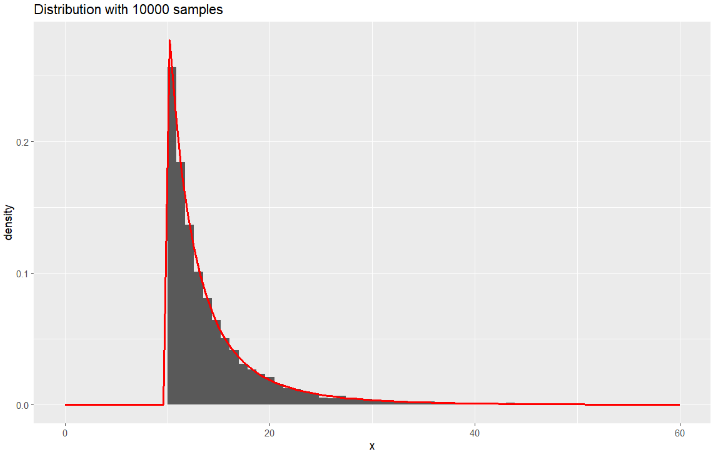

Figure 3: Graph of pdf and random sampling of 10000 points, given $\gamma = 4$ and $x_0 = 10$

2 Gamma Sampling

Suppose we have $X \sim Gamma(2,1)$ with pdf

$$f(x) = \begin{cases} xe^{-x}, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{10}$$

a) To calculate the CDF we take the integral of the pdf from 0 to x. Using integration by part with $u = x$ and $dv = e^{-x}$

$$\int_{-\infty}^{x} f(x)dx = \int_{0}^{x} xe^{-x}dx = -xe^{-x} + \int_{0}^{x} e^{-x}dx \tag{11}$$

$$= -e^{-x}(x+1)|_0^x \tag{12}$$

$$= 1 - e^{-x}(x+1) \tag{13}$$

Hence we have

$$F(x) = \begin{cases} 1 - e^{-x}(x+1), & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{14}$$

b I will be using Newtons method to find the inverse of the CDF. The method will guess a root at one and runs 5 times then repeated until the root is positive. There is a sanity check where if the root stays negative an error occurs. Fortunately the code works. The solution to the CDF will be calcualted by solving the root of $F(x) - u = 0$. The function took 5.1287 seconds to run.
c To solve for the optimal value for c, I will use the inequality:

$$C\frac{1}{2}e^{-\frac{x}{2}} \geq xe^{-x} \tag{15}$$

$$C \geq 2xe^{\frac{-x}{2}} = h(x) \tag{16}$$

$$\tag{17}$$

To solve for the critical point of the equation, take the derivative and solve for the roots

$$h'(x) = 2e^{\frac{-x}{2}} - xe^{\frac{-x}{2}} \tag{18}$$

$$= e^{\frac{-x}{2}}(2-x) = 0 \tag{19}$$

$$\tag{20}$$

Since the derivative is a strictly decreasing function and is positive when $x < 2$, the critical value is the point where $h(x)$ is the absolute maximum yielding

$$C \geq h(2) = 4/e \tag{21}$$

Since efficiency is determined by $1/C$ the smallest possible value of C to maximize efficiency is $4/e$.
Also note that the given distribution $g(x)$ is aa pdf for an exponential distribution with $\lambda = 1/2$ Using an acceptace rejection algorithm, it took 3.1873 seconds to complete.

d After running the simulation code, the experiment was completed in 0.0872 seconds.
e

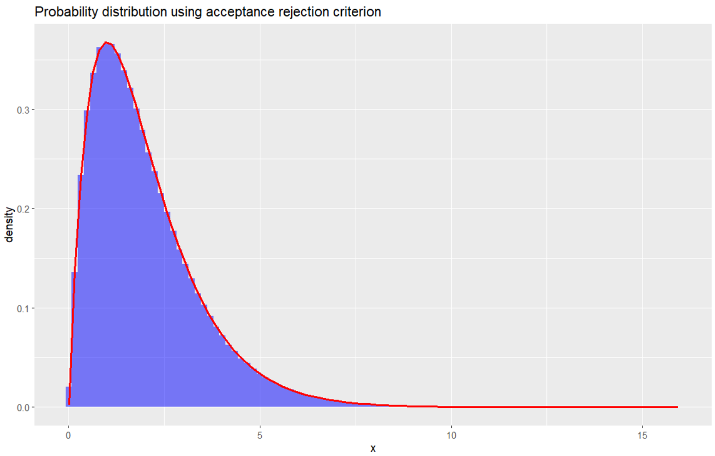

Figure 4: Graph for the pdf of xe^{-x} using acceptance rejection criterion. See HW2 Problem 2c.R for code

f The fastest algorithm is summing the exponential distributions, then it is the acceptance-rejection criterion, and the slowest is using the root solver. Note the root solver doesn't even give very accurate results and takes even more time if I want it to be more accurate.

3 Mixture Sampling to Polynomial Sampling

a To show that these two distributions are the same, we will be calculating the CDF of Y. Note the main proof follows the law of total probability:

$$\begin{aligned}
Y(y) = P(Y \leq y) &= \sum_{j=1}^{N} P(X_J \leq y | J = j) P(J = j) && (22)\\
&= \sum_{j=1}^{N} P(X_j \leq y) P(J = j) && (23)\\
&= \sum_{j=1}^{N} P(X_j \leq y) a_j && (24)\\
&= \sum_{j=1}^{N} \int_0^y f_j(x) dx a_j && (25)\\
&= \sum_{j=1}^{N} \int_0^y f_j(x) a_j dx && (26)\\
&= \int_0^y \sum_{j=1}^{N} a_j f_j(x) dx && (27)\\
&= \int_0^y f(x) dx = F(y) && (28)
\end{aligned}$$

b To solve for the CDF, I will be taking the integral of f_j from 0 to x.

$$\int_0^x f_j(x) dx = \int_0^x (j+1) x^j dx == x^{j+1}|_0^x = x^{j+1} \tag{29}$$

To sample from this distribution I will use inverse sampling. The inverse of the CDF is $x = u^{\frac{1}{j+1}}$. Hence the pseudo code will be.

```
1. Generate a random value $u \sim U(0,1)$
2. return $u^{\frac{1}{j+1}}$
```

c I will be finding a way to setup $f(z)$ Using the sum that is given above. Since the highest power of $f(z)$ is 5, $N = 5$ resulting in the following sum:

$$\sum_{j=1}^{5} a_j f_j(z) = a_1 2z + a_2 3z^2 + a_3 4z^3 + a_4 5z^4 + a_5 6z^5 \tag{30}$$

Furthermore $a_1 = \frac{1}{2}, a_5 = \frac{1}{2}$ while all other a_j equal 0. Using the fact from question a due to the law of total probability, we can do conditional sampling.
1. Sample $j \sim U(0,1)$
2. if $j \geq .5$ $j = 1$ else $j = 5$
3. generate $u \sim U(0,1)$
4. return $u^{\frac{1}{j+1}}$

4 Gaussian Sampling

The pdf of a standard normal distribution is shown below.

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \tag{31}$$

Note it is not possible to conver this integral to a CDF. Hence Acceptance-rejection will be used. I will be using the pmf $g(x)$ of an exponential distribution with $\lambda = 1$ since

$$cg(x) \geq f(x) \tag{32}$$

$$ce^{-x} \geq \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \tag{33}$$

$$c\sqrt{2\pi} e^{-x} \geq e^{\frac{-x^2}{2}} \tag{34}$$

$$c\sqrt{2\pi} e^{\frac{x^2-2x}{2}} \geq 1 \tag{35}$$

$$c\sqrt{2\pi} e^{\frac{x(x-2)}{2}} \geq 1 \tag{36}$$

$$\tag{37}$$

Note how there are two critical points for $(x(x-2))$ at $x = 0, 2$. By taking the derivative, there is one critical point at at $x = 1$ and it is the absolute minimum as this is a concave up quadratic curve. Hence the value of x that would minimize the left side of the inequality would be $x = 2$ yielding $\sqrt{2\pi} e^{-1/2} = .6577$. However we can set c to $\sqrt{\frac{e}{2\pi}} = 1.5205$ to satisfy the inequality. Also note the gaussian distribution is symmetrical hence:
1. Choose a random $x \sim Exp(1)$
2. Choose a random $u \sim U(0,1)$
3. if $u \geq \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} / exp(\frac{x^2-2x-1}{2})$ continue to 4 else go back to 1
4. Choose a random $l \sim U(0,1)$
5. If $l \geq .5$ return x else return -x

5 Discrete Random Variable

a To write the cummulative distributions we start with the probabiliy distribution function:

$$P(1) = .01 \tag{38}$$

$$P(2) = .9 \tag{39}$$

$$P(3) = .09 \tag{40}$$

$$\tag{41}$$

Hence:

$$P(X \leq 1) = .01 \tag{42}$$

$$P(X \leq 2) = .91 \tag{43}$$

$$P(X \leq 3) = 1 \tag{44}$$

$$\tag{45}$$

Note however we can invert the equations above and use the uniform distribution to generate discrete sampling
1. Sample $u = U(0,1)$
2. if $u \leq .01$ return 1 else
3. if $u \leq .91$ return 2
4. else return 3

b If we arrange them in order of the highest probabilities this would improve the efficiency of the code since the ones that more likely to be within threshold are checked first. The best order is $\{2, 3, 1\}$. Using the law of total expectation for the first algorithm, where T is the number of comparisons not including a standalone else:

$$E[T] = E[E[T|X]] = \sum_{i=1}^{3} E[T|X = i]P(X = i) = 1 * .01 + 2 * .9 + 2 * .09 = 1.99 \tag{46}$$

While the new algorithm has an expected comparison count of:

$$E[T] = E[E[T|X]] = \sum_{i=1}^{3} E[T|X = i]P(X = i) = 1 * .9 + 2 * .09 + 2 * .01 = 1.1 \tag{47}$$

Therefore, if we run the each algorithm for M samples, the first method would require $1.99M$ comparisons while the second method would only require $1.1M$ comparisons.