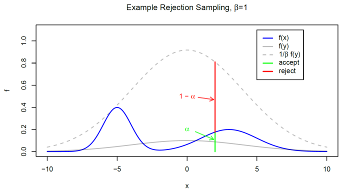
Algorithms and Data Structures

Random Numbers
(Adaptive) Rejection Sampling



Learning goals

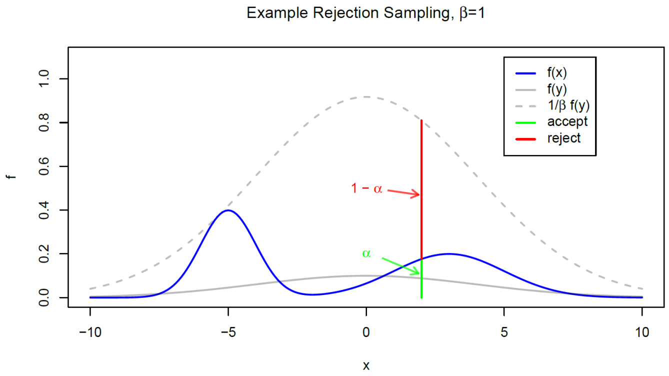
- Rejection sampling
- Adaptive rejection sampling

REJECTION SAMPLING

- Aim: Generate random numbers X from a distribution with density $f_X(x)$.
- Idea: Draw Y from distribution with density $f_Y(y)$ (Proposal density) instead;
  there must be a β with $0 < \beta < 1$ such that $\beta f_X(x) \leq f_Y(x)$ for all $x \in \text{supp}(X)$.
- Accept Y as a random number from $f_X(x)$ with probability

$$\alpha = \alpha(Y) = \beta \cdot \frac{f_X(Y)}{f_Y(Y)}$$

REJECTION SAMPLING / 2



Note: In this plot α is shown as a percentage of the "total distance", so it does not refer to the y-axis in the plot.

REJECTION SAMPLING / 3

Algorithm Rejection Sampling

```
Initialization: f_Y, f_X, β, N (number of RV needed)
i ← 0
while i ≠ N do
    Create a random number Y from f_Y (Y ~ f_Y)
    Calculate α(Y) = β f_X(Y) / f_Y(Y)
    Create a random number U ~ U(0, 1) independent of Y
    if U ≤ α(Y) then
        Accept Y
        i ← i+1
    else
        Reject Y
    end if
end while
```

PROOF: REJECTION SAMPLING

$$P(Y \le x \mid U \le \alpha(Y)) = \frac{P(Y \le x, U \le \alpha(Y))}{P(U \le \alpha(Y))}$$

Y, U independent $\Rightarrow$ common density $f(y, u) = f_Y(y) \cdot 1 = f_Y(y)$

$$P(Y \le x, U \le \alpha(Y)) = \int_{-\infty}^{x} \int_{0}^{\alpha(y)} f_Y(y)\, du\, dy = \int_{-\infty}^{x} \alpha(y) f_Y(y)\, dy$$

$$= \int_{-\infty}^{x} \beta \frac{f_X(y)}{f_Y(y)} f_Y(y)\, dy = \beta \int_{-\infty}^{x} f_X(y)\, dy$$

$$P(U \le \alpha(Y)) = P(Y \le \infty, U \le \alpha(Y)) = \beta \int_{-\infty}^{\infty} f_X(y)\, dy = \beta$$

PROOF: REJECTION SAMPLING / 2

In summary we obtain exactly what was required

$$P(Y \leq x \mid U \leq \alpha(Y)) = \int_{-\infty}^{x} f_X(y)\, dy.$$

Choice of Y:
The closer f_Y to f_X, the closer α to 1.
$\Rightarrow$ There is less rejection, hence it is faster.

However, random variables of Y should be generated as quickly as possible.

Note: β is the probability of Y being accepted. The greater β, the better.

β itself is not needed in the algorithm (only $\alpha(Y)$).
$\Rightarrow$ calculation of normalization constants not always needed.

EXAMPLE: NORMAL DISTRIBUTION

Only to illustrate Rejection Sampling! Rejection Sampling from $\mathcal{N}(0,1)$-distribution via Cauchy distribution (therefore we have Inverse transform sampling):

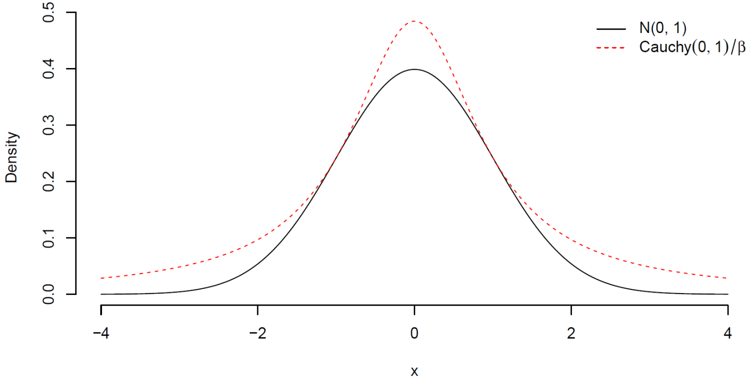
$$\begin{aligned} f_X(x) &= \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2) \\ f_Y(x) &= \frac{1}{\pi} \cdot \frac{1}{1+x^2} \end{aligned}$$

It is easy to show that $\beta = \inf_y \frac{f_Y(y)}{f_X(y)} = \sqrt{\frac{e}{2\pi}} \approx 0.657$.

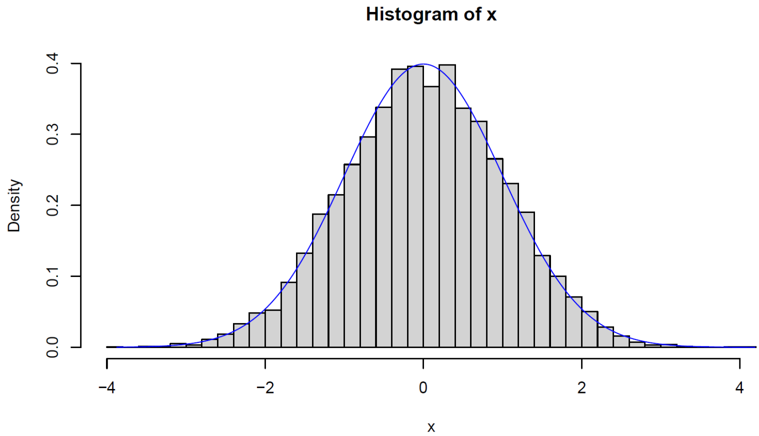
The probability of acceptance $\alpha(Y)$ is

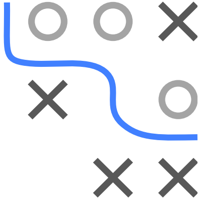
$$\begin{aligned} \alpha(Y) &= \frac{\beta f_X(Y)}{f_Y(Y)} \\ &= \frac{\sqrt{e}}{2}(1+y^2)\exp(-\frac{1}{2}y^2). \end{aligned}$$

EXAMPLE: NORMAL DISTRIBUTION / 2



EXAMPLE: NORMAL DISTRIBUTION / 3





ADAPTIVE REJECTION SAMPLING

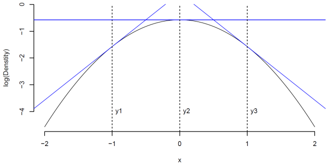
Often it is difficult to find a "good" proposal density f_Y. **Adaptive rejection sampling (ARS)** is an approach to construct adaptive proposal densities. ARS is based on the following ideas:

- Working with log densities (often algebraically simpler)
- Use piecewise linear density functions for f_Y
- Adapt f_Y as soon as a proposal Y is rejected

ADAPTIVE REJECTION SAMPLING / 2

Procedure:

1. Construction of the proposal density f_Y
   1. Start with $M := \{y_1, \ldots, y_k\}$
   2. Evaluate the log density $l_X := \log f_X(y)$ for all $y \in M$ and find the tangent lines at these points
   3. Define a piecewise linear function which is composed of the tangent lines: $l_Y \rightarrow$ upper bound for l_X

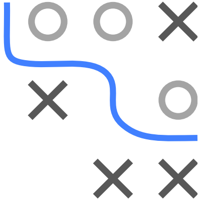


   4. Back-transform: $f_Y := \exp(l_Y)$

ADAPTIVE REJECTION SAMPLING / 3

2 Rejection sampling:

1. Create a random number $Y \sim f_Y$ $^{(*)}$
2. Calculate $\alpha(Y) = \frac{\exp(l_X(Y))}{\exp(l_Y(Y))} = \exp(l_X(Y) - l_Y(Y))$
3. Create $U \sim \mathrm{U}(0,1)$
   - If $U \leq \alpha(Y)$: Accept Y
   - Otherwise: Reject Y, add this point to $M \rightarrow M \cup Y$ and go to step 1



$^{(*)}$ A method for "sampling" f_Y and an implementation can be found here:

`https://blog.inferentialist.com/2016/09/26/adaptive-sampling.html`