Simulating continuous random variables

Florence Perronnin

Univ. Grenoble Alpes, LIG, Inria

October 18, 2018

Outline

Simulation of random variables

Summary

Transforming random numbers

1. Suppose we have a perfect uniform random number generator (PRNG) $\mathcal{U}_{01}$ or `random()`
2. Write an algorithm that uses $\mathcal{U}_{01}$ as input and produces a random variate X with distribution f as output

Prove

Correctness of the transformation algorithm

Validate

at least experimentally.

Outline

From $[0, 1]$ to $[a, b]$

Suppose you have a random generator `random()` providing uniform samples U_k in $[0, 1]$. But you need a uniform real number in $[0, b]$ where $b > 0$.

`random()` $\times b$

Proof. Let V be the output of the algorithm and U the result of `random()`. Then for any $0 < x < b$:

$$\begin{aligned}\mathbb{P}[V \leq x] &= \mathbb{P}[\texttt{random()} \times b \leq x] \\ &= \mathbb{P}[U \times b \leq x] \\ &= \mathbb{P}\left[U \leq \frac{x}{b}\right] \\ &= \frac{x}{b} \qquad \text{(uniform on } [0, b]).\end{aligned}$$

Outline

Inverse transform method

Inverse of the CDF

Suppose the CDF F is continuous and strictly monotone over the support I. Then :

- $F(\mathbb{R}) = [0, 1]$
- Any number $u \in [0, 1]$ is the probability that X exceeds some value $c \in I$. *(Intermediate value theorem)*
- That value c is unique *(F strictly increasing)*

So F is invertible and has inverse F^{-1}.

Inverse transform method

Idea

Inverse transform

Let X be a random variable with invertible CDF F. Let U be a uniform random variable over $[0,1]$. Then:

$$V = F^{-1}(U)$$

has the same distribution as X.

Proof.

$$\begin{aligned}
\mathbb{P}[V \leq x] &= \mathbb{P}\left[F^{-1}(U) \leq x\right] \\
&= \mathbb{P}\left[F \circ F^{-1}(U) \leq F(x)\right] && \text{F is increasing} \\
&= \mathbb{P}[U \leq F(x)] \\
&= F(x) && U(\text{uniform on } [0,1]).
\end{aligned}$$

Inverse transform method

Algorithm

Let X be a r.v. with invertible CDF F and let $G = F^{-1}$. Then X can be sampled with the following algorithm:

Inverse transform algorithm

```
G(random())
```

Application to uniform sampling over $[a, b]$

$$F_X(x) = \frac{x-a}{b-a} \qquad \text{for } x \in [a, b]$$

Algorithm

$a +$ `random()` $\times (b - a)$

Application to exponential distribution $\mathcal{E}(\lambda)$

$$F_X(x) = 1 - e^{-\lambda x}, \qquad \text{for } x \geq 0$$

Algorithm

$-\frac{1}{\lambda} \ln($ `random()` $)$

Density $f(x)$	$F(x)$	$X=F^{-1}(U)$	Simplified form
Exponential(λ) $\lambda e^{-\lambda x}$, $x \geq 0$	$1-e^{-\lambda x}$	$-\frac{1}{\lambda}\log(1-U)$	$-\frac{1}{\lambda}\log(U)$
Cauchy(σ) $\frac{\sigma}{\pi(x^2+\sigma^2)}$	$\frac{1}{2}+\frac{1}{\pi}\arctan(\frac{x}{\sigma})$	$\sigma\tan(\pi(U-\frac{1}{2}))$	$\sigma\tan(\pi U)$
Rayleigh(σ) $\frac{x}{\sigma}e^{-\frac{x^2}{2\sigma^2}}$, $x \geq 0$	$1-e^{-\frac{x^2}{2\sigma^2}}$	$\sigma\sqrt{-\log(1-U)}$	$\sigma\sqrt{-\log(U)}$
Triangular on$(0,a)$ $\frac{2}{a}(1-\frac{x}{a})$, $0 \leq x \leq a$	$\frac{2}{a}(x-\frac{x^2}{2a})$	$a(1-\sqrt{1-U})$	$a(1-\sqrt{U})$
Tail of Rayleigh $xe^{\frac{a^2-x^2}{2}}$, $x \geq a > 0$	$1-e^{\frac{a^2-x^2}{2}}$	$\sqrt{a^2-2\log(1-U)}$	$\sqrt{a^2-2\log U}$
Pareto(a,b) $\frac{ab^a}{x^{a+1}}$, $x \geq b > 0$	$1-(\frac{b}{x})^a$	$\frac{b}{(1-U)^{1/a}}$	$\frac{b}{U^{1/a}}$

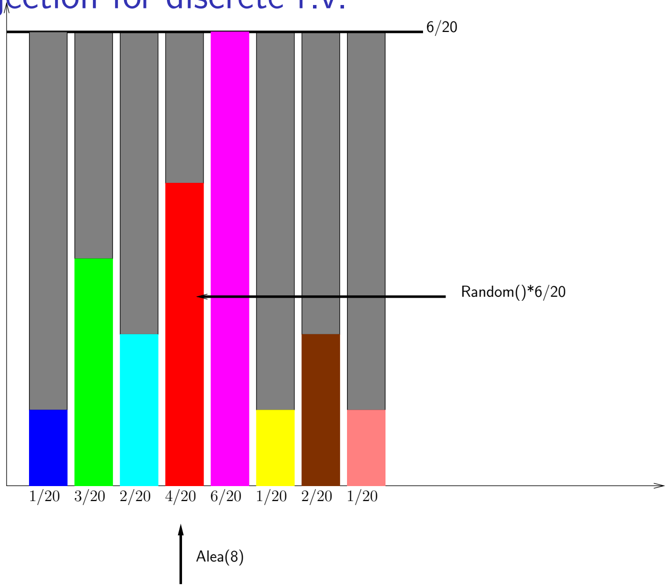
Limitations

- F may not be continuous (or strictly increasing)
- F^{-1} may require numerical computation (inexact methods)

Cf. [Devroye] for more details on numerical methods for inverse transform.

Outline

Rejection for discrete r.v.



Rejection for discrete r.v.

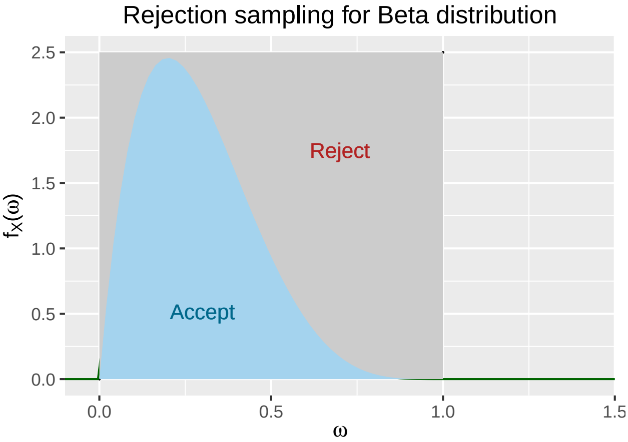
X discrete R.V with values in $1, \ldots, N$. Suppose the probabilities are bounded by *pmax*

Algorithm

```
repeat
      generate x uniformly over 1...N
      generate y uniformly over [0, pmax]
until y <= P(x)
```

- bounded support
- rejection area (large *pmax*) $\Rightarrow$ cost of sampling

Rejection sampling for a continuous r.v.



Limitations

- Bounded support : can't generate uniformly from an infinite interval!
- Bounded density (not for exponential r.v for instance)

Rejection for continuous r.v.

Algorithm

```
repeat
      generate x uniformly over I
      generate y uniformly over [0,h]
until y <= f(x)
```

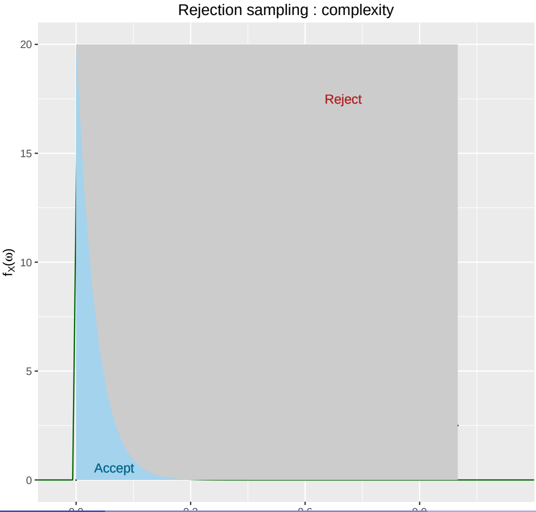
Complexity

Let N be the number of iterations (also a random variable).
$\mathbb{E}[N] = \frac{1}{p_{\text{accept}}}$

Let X be a continuous r.v. with:

- bounded support $I = [a, b]$
- bounded density $f(x) \leq h, \quad \forall x \in [a, b]$

Then $\mathbb{E}[N] = h(b - a)$ (grey area)



Rejection with a dominating density

Motivations:

- generalization for unbounded support
- increase sampling efficiency
- no constant bound on density sometimes

Idea

Let X and f_X be the r.v. to sample. Assume that for all x:

$$f_X(x) \leq c\, g(x)$$

where:

- g is a density
- random variates with density g can be easily sampled
- c is known

Outline

Outline

Methods for generating continuous random variables

Generic methods

- Inverse of CDF : can be pre-computed for finite r.v. at the extra-cost of a table
- Rejection method : complexity depends on rejection probability

Specialized methods

- exploit intrinsic structure of probability laws
- composition methods

Caveats

- Validity of the transformation
- Time complexity (number of operations)
- Memory overhead

Sources

- Jean-Marc Vincent, Random generation of discrete random variables, Notes de cours, 2010.
- Jean-Marc Vincent, Générateurs de loi uniforme et de lois discrètes, Notes de cours, 2016.
- Jean Bérard, Génération de variables pseudo-aléatoires, Notes de cours, Université Claude Bernard - Lyon 1.
- Luc Devroye, Non-uniform Random Variate Generation, http://www.eirene.de/Devroye.pdf
- Pierre L'Ecuyer, Random number generation, Handbook of Computational Statistics. Springer, Berlin, Heidelberg, pp. 35-71, 2012.
- Alastair Walker, An efficient method for generating discrete random variables with general distributions, ACM Transactions on Mathematical Software (TOMS) 3: 253-256, 1977.