Simulation

Alberto Ceselli
MSc in Computer Science
Univ. of Milan

Part 3 - Generating Random Variables

Summary

Summarizing

- Systems can be modeled, including nontrivial components in the form of *random variables*
- Statistics allow to reliably run experiments and obtain results by observations on these models
- (Pseudo-)Random numbers can be generated *algorithmically*

Summary

Summarizing

- Systems can be modeled, including nontrivial components in the form of *random variables*
- Statistics allow to reliably run experiments and obtain results by observations on these models
- (Pseudo-)Random numbers can be generated *algorithmically*
- Next step: to generate observations of *random variables* algorithmically
  - Discrete Random Variables
  - Continuous Random Variables

Main techniques

Main techniques for generating R.V.:

- inverse transform
- acceptance-rejection
- composition
- (alias)

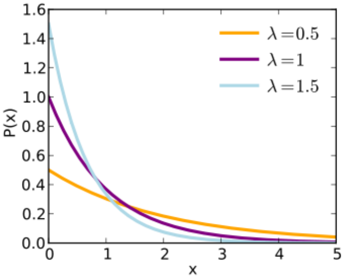
Inverse transform for Discrete R.V.

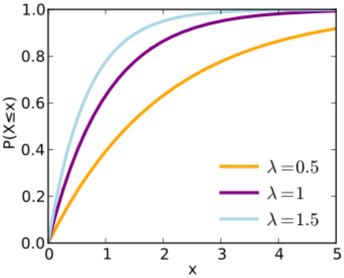
See R code.

- Example: generating a R.V. given its p.d.f.
- Example: generating a binomial R.V.
- Example: computing means through sampling
- Example: generating a geometric R.V.
- Example: generating a random permutation

Inverse transform for Continuous R.V.

Example: generating an exponential R.V. [1]





For $0 < x < \infty$

$$f(x) = \lambda \cdot e^{-\lambda \cdot x} \qquad F(x) = 1 - e^{-\lambda \cdot x}$$

[1] By Skbkekas - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=9508326

Inverse transform for Continuous R.V.

In “forward” mode: input x, obtain $u = P[X \leq x]$

- $u = F(x) \rightarrow x = F^{-1}(u)$
- $u = 1 - e^{-\lambda \cdot x} \rightarrow 1 - u = e^{-\lambda \cdot x} \rightarrow x = \dfrac{-\log(1-u)}{\lambda} = F^{-1}(u)$

In “inverse” mode: input u, obtain $x|u = P[X \leq x]$

Proposition: Let U be a uniform $(0,1)$ R.V.. For any continuous distribution function F, the R.V. X defined by

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

has distribution F. (proof on the blackboard).

Acceptance-Rejection

Still by Von Neumann. Principle:

- A R.V. X needs to be generated, with p.d.f. $f(x)$
- Another R.V. Y, with p.d.f. $g(y)$, is known and easy to generate
- You know that $f(y)/g(y) \leq c \ \ \forall y$
- Idea:
  - generate a value y for Y (from $g(y)$)
  - generate a value u for a uniformly distributed R.V. U
  - if $u \leq \dfrac{f(y)}{c \cdot g(y)}$, then output $X = y$
  - otherwise iterate

Theorem: (a) The R.V. generated in this way has p.d.f. $f(x)$, and (b) the number of iterations needed to converge is a geometric R.V. with expected value c (proof on the blackboard).

Acceptance-Rejection: examples

Example: 4f at page 58.
Example: generating a *normal* R.V. (only 1 step missing , blackboard discussion).

Composition

Principle:

- A R.V. X needs to be generated, with C.D.F. $F()$
- $F()$ can be *decomposed* as $F(x) = \sum_{i=1}^{n} \alpha_i \cdot F_i(x)$, with $\sum_{i=1}^{n} \alpha_i = 1$
- Idea:
  - generate a value j for a (discrete) R.V. whose p.d.f. is given by α_i
  - generate a value for a R.V. whose C.D.F. is $F_j()$

Theorem: the value obtained in this way is distributed according to $F()$.

Case Study: simulating normal R.V.

Generating a *normal* R.V. (blackboard discussion).

Simulating a Poisson Process

Case study: how to simulate a Poisson Process?

- Option 1: simulate a Poisson R.V. (blackboard discussion)
- Option 2: simulate interarrival times with exponential R.V. (blackboard discussion)

Copulas

Def.

a *copula* is a joint probability distribution $C(x, y)$ with both marginal distributions being uniformly distributed in $(0, 1)$

$$C(0, 0) = 0$$
$$C(x, 1) = x$$
$$C(1, y) = y$$

for $0 \leq x, y \leq 1$.
E.g. we want $H(x, y)$ for R.V. X and Y, begin $F(x)$ and $G(y)$ being their CDF.
$H(x, y) = P[X \leq x, Y \leq y] = P[F(X) \leq F(x), G(Y) \leq G(y)] = C(F(x), G(y))$

Copulas

Algorithmically

- You know $F(X)$ and $G(Y)$, but not their joint CDF $H(X, Y)$
- You know they are *not independent*
- Still, you want to generate pairs of meaningful values
- $\rightarrow$ generate values for the copula
- $\rightarrow$ apply inverse transform method to get values of the original distributions

Example: Gaussian Copula

- Generate $Z = (z_1, z_2)$ from bivariate Normal (e.g. with a certain correlation ρ)
- Obtain "random pairs of (correlated) uniform" $U = (u_1, u_2) = (\Phi(z_1), \Phi(z_2))$ (this joint distrib. is Gaussian copula)
- Obtain "random custom" $R = (F^{-1}(u_1), G^{-1}(u_2))$

Copulas

A copula should model the dependencies between X and Y.
E.g. if we think X and Y to have linear correlation ρ, the values generated by the copula should have linear correlation (approx.) ρ (not implying linear correlation ρ between $F(X)$ and $G(Y)$).
Examples (blackboard)

- Gaussian copula
- Marshall-Olkin copula

Copulas

Example: Marshall-Olkin copula

- Suppose you have three types of shocks, each occurring with exponential interarrival time (rate $\lambda_1, \lambda_2, \lambda_3$)
- Suppose you have two components: one sensitive to shocks $1, 3$, one sensitive to shocks $2, 3$
- how to generate joint inter-failure times?
- $C(x, y) = \min\{x^\alpha y, xy^\beta\}$ with $\alpha = \dfrac{\lambda_1}{\lambda_2 + \lambda_3}$ and $\beta = \dfrac{\lambda_2}{\lambda_2 + \lambda_3}$