Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Generating Continuous Random Variates

Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering

1

Prerequisite

We assume the knowledge of continuous random variables (sect.7.1).
In particular:

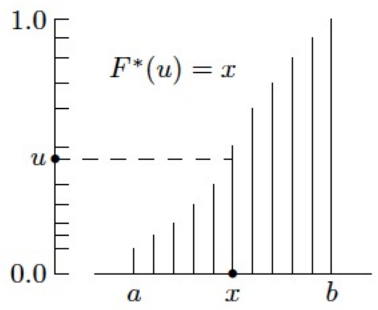
- *Uniform*(*a*,*b*)
- *Exponential*(μ)
- *Normal*(μ,σ)
- *Lognormal*(*n*,*b*)
- *Erlang*(*n*,*b*)
- *Student*(*n*)

Nel caso continuo è tutto più facile, perchè c'è corrispondenza 1-1

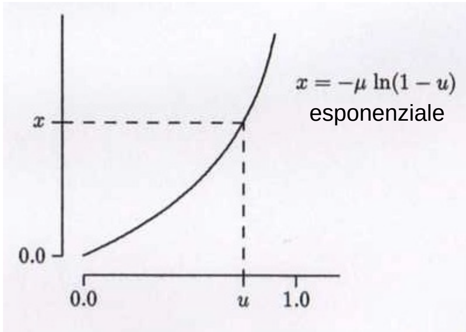
2

Discrete Random Variates

$$F^*(u) = \min_x \{x : u < F(x)\}$$



Continuous Random Variates



per un dato 'u' ho esattamente un unico e solo 'x'.
L'inversa è vera.

3

In alcuni casi inversione facile, in altri casi esistono sempre dei metodi!

Preliminary Definitions

The *inverse distribution function* (idf) of X is the function

$F^{-1} : (0, 1) \rightarrow \chi$, $\forall\, u \in (0, 1)$ as

supporto,
da cui
estraggo 'u'

$$F^{-1}(u) = x$$

stesse definizioni,
qui essendo inversa vera
posso scrivere F^(-1).

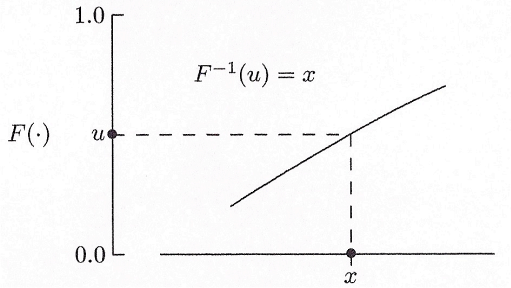
where $x \in \chi$ is the unique possible value for $F(x) = u$

There is a one-to-one correspondence between possible values $x \in \chi$ and cdf values $u = F(x) \in (0, 1)$

4

Continuous Random Variable idfs

- Unlike the a discrete random variable, the idf for a continuous random variable is a true inverse



- Can sometimes determine the idf in "closed form" by solving $F(x) = u$ for x

5

Examples di funzioni aventi "inverse specifiche".

Ricordando che F(x) = u in questi casi esplicito rispetto x, nulla di più.

- if X is *Uniform(a,b)*, $F(x) = (x\text{-}a)/(b\text{-}a)$ for $a < x < b$

$$x = F^{-1}(u) = a + (b - a)u \qquad 0 < u < 1$$

- if X is *Exponential(μ)*, $F(x) = 1\text{-exp}(\text{-}x/\mu)$ for $x > 0$

$$x = F^{-1}(u) = -\mu \ln(1 - u) \qquad 0 < u < 1$$

- if X is a continuous variable with possible value $0 < x < b$ and pdf $f(x) = 2x/b^2$, cdf $F(x) = (x/b)^2$

$$x = F^{-1}(u) = b\sqrt{u} \qquad 0 < u < 1$$

6

Random Variate Generation By Inversion

- X is a continuous random variable with idf $F^{-1}(\cdot)$
- Continuous random variable U is *Uniform*(0,1)
- Z is the continuous random variable defined by $Z = F^{-1}(U)$

Theorem
Z and X are identically distributed

Algorithm 1

```
u = Random();
return F^-1(u);
```


Inversion examples

- *Uniform*(*a*,*b*) Random Variate

```
u = Random();
return a + (b - a) * u;
```

- *Exponential*(μ) Random Variate

```
u = Random();
return - μ log(1-u);
```

Inversion algorithms

- Algorithms in the previous two examples are:
  - portable, exact, robust, efficient, clear, synchronized and monotone

una chiamata alla random == una generazione di variata

- It is not always possible to solve for a continuous random variable idf explicitly by algebraic techniques
- Two other options may be available:
  1. Use a function that accurately *approximates* $F^{-1}(\cdot)$
  2. Determine the idf by solving $u = F(x)$ *numerically*

  (see section 7.2.2)

Se inversa difficile, o approssimo funzione inversa, oppure risolvendola numericamente.

Testing for correctness

per vedere se generazione è ok!

- generate a sample of n random variates where n is large
- evaluate sample mean and standard deviation
- compare them with the theoretical values,

  they should be *reasonably* close !!

This is not enough!! Different distributions can have the same mean and standard deviation !!!

dovrei anche costruire istogramma e confrontarlo con la distribuzione che sto approssimando (non lo vediamo).

- generate a sample of n random variates and construct a k-bin continuous-data histogram with bin width δ
- f' is the histogram density and $f(x)$ is the pdf

pdf teorica

$$f' \to f(x) \quad \text{as } n \to \infty \quad \text{and} \quad \delta \to 0$$

- In practice, using a large but finite value of n and a small but non-zero value of δ, perfect agreement between f' and f will not be achieved

Discrete case: natural sampling variability !
Continuous case: variability+binning !!

Truncation

ovviamente esiste anche nel continuo!

- Let X be a continuous random variable with possible values χ and cdf $F(x)=\Pr(X \le x)$
- Suppose we wish to restrict the possible values of X to $(a, b) \subset \chi$

Sia che partendo dai due punti, che dalle due masse di probabilità che voglio eliminare, la trasformazione è esatta!

It is similar to, but simpler than truncation in the discrete-variable context

- X is $\le a$ with probability $\Pr(X \le a) = F(a)$
- X is $\ge b$ with probability $\Pr(X \ge b) = 1 - \Pr(X < b) = 1 - F(b)$
- X is between a and b with probability

$$\Pr(a < X < b) = \Pr(X < b) - \Pr(X \le a) = F(b) - F(a)$$

2 cases for truncation

Case 1 se conosco i "due punti"

if a and b are specified, the cdf of X can be used to determine the left-tail α, right-tail β truncation probabilities

$$\alpha = \Pr(X \le a) = F(a) \quad \text{and} \quad \beta = \Pr(X > b) = 1\text{-}F(b)$$

Case 2 parto dalle "masse di probabilità da escludere" e trovando i due punti associati.

if α and β are specified, the idf of X can be used to determine left and right truncation points

$$a = F^{-1}(\alpha) \quad \text{and} \quad b = F^{-1}(1\text{-}\ \beta)$$

$$F(b) = 1\text{-}\beta$$

Both transformations are exact !

Library `rvgs`

- Contains 7 continuous random variate generators
  - `double Chisquare(long n)`
  - `double Erlang(long n, double b)`
  - `double Exponential(double μ)`
  - `double Lognormal(double a, double b)`
  - `double Normal(double μ, double σ)`
  - `double Student(long n)`
  - `double Uniform(double a, double b)`