Curs 8: Simularea variabilelor aleatoare

Conf.dr. Maria Jivulescu

Departamentul de Matematică
UPT

Cuprins

Metode de simulare a variabilelor aleatoare:

- Distribuția unif. continuă;
- Distribuția unif. discretă;
- Distribuția discretă neuniformă;
- Distribuția Bernoulli;
- Distribuția Binomială;
- Distribuția geometrică;
- Ditribuții continue care au funcția de repartiție inversabilă prin metoda inversării;

A simula o variabilă aleatoare discretă X, presupune a genera independent, conform unui algoritm, un șir de numere $y_1, y_2, \dots, y_N$, care să aibă particularitățile unor valori de observație asupra variabilei.

Mai precis: dacă variabila X are distribuția de probabilitate:

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix}$$

atunci distribuția experimentală a datelor generate trebuie să fie foarte apropiată de distribuția teoretică a variabilei X:

dacă nr_k -numărul valorilor generate, egale cu x_k, $k \in \{1, n\}$, atunci $\frac{nr_k}{N}$ trebuie să fie apropiat de p_k

`https://www.randomservices.org/random/apps/SpecialSimulation.html`

Principiu de bază în simulări

Cum se simulează o v.a.?

Vom prezentăm pseudocodul pentru o funcție ce returnează **o singură valoare de observație simulată.**

- Această valoare se obține printr-o **transformare sau mai multe, aplicată unei valori `u` presupusă a fi returnată de un generator de numere pseudo–aleatoare uniform distribuite pe** $[0,1)$[1].
- Apelul acestui generator îl simbolizăm prin `urand( )`(nume generic pentru simulatorul unei variabile aleatoare $U \sim \text{Unif}[0,1)$.)

$$\texttt{u=urand( )} \xrightarrow{\textit{Transformare}} \texttt{x}$$

De ce folosim $U \sim \text{Unif}[0,1)$?

[1] detalii în Material suplimentar Cap II

Simularea distribuției uniforme continue

Reamintim: dacă $U \sim$ Unif$[0,1)$, atunci $(a,b] \subset [0,1)$,

$$P(U \in (a,b]) = b - a$$

Exemplu: Dacă $A-$ eveniment $P(A) = 0.65$ și $B = \complement A$, $P(B) = 0.35$, atunci putem simula producerea unuia din cele două evenimente astfel:

- apelăm generatorul `urand()`:
  u=urand();
- Dacă $u < 0.65$, spunem că s-a produs evenimentul $(U < 0.65)$, $P(U < 0.65) = P(U \in [0, 0.65)) = 0.65$.
  A are aceeași probabilitate de realizare, deci putem considera că s-a produs evenimentul A.
- Dacă $u \in [0.65, 1)$, atunci s-a produs evenimentul $(U \in [0.65, 1))$, $P(U \in [0.65, 1)) = 1 - 0.65 = 0.35$.
  Deci, în acest caz putem considera că s-a produs evenimentul B.

Observație: Se utilizează la algoritmii randomizați `http://en.wikipedia.org/wiki/Randomized_algorithm`.

Simularea distribuției uniforme discrete

Propoziție

Dacă U ∼Unif[0,1) v.a. uniform distribuită pe $[0,1)$, iar n este număr întreg, $n > 1$, atunci

$$X = [nU]$$

o variabilă aleatoare discretă ce are distribuția uniformă pe mulțimea $\{0, 1, 2, \ldots, n-1\}$, adică $X = \begin{pmatrix} 0 & 1 & \ldots & n-1 \\ \frac{1}{n} & \frac{1}{n} & \ldots & \frac{1}{n} \end{pmatrix}$

Demonstrație: se bazează pe

- U ia valori în $[0,1)$, variabila nU ia valori în $[0,n)$ și $[nU] \in \{0, 1, \ldots, n-1\}$.
- $$\begin{aligned} P(X = k) &= P([nU] = k) = P(k \leq nU < k+1) \\ &= P\left(\frac{k}{n} \leq U < \frac{k+1}{n}\right) = P\left(U \in \left[\frac{k}{n}, \frac{k+1}{n}\right)\right) = \frac{k+1}{n} - \frac{k}{n} = \frac{1}{n} \end{aligned}$$

Algoritm v.a. discretă uniform distribuită


```
Function SimDiscretU(n)
u=urand();
k=int(n*u);
return k;
end.
```

Observație: Inlocuind `return k;` cu return x_k; se returnează o valoare de observație asupra unei variabile aleatoare discrete, uniform distribuită pe o mulțime $\{x_0, x_1, \dots, x_{n-1}\}$, adică asupra variabilei aleatoare:

$$X = \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \\ \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} \end{pmatrix}$$

Observație: Simularea acestei variabile aleatoare este echivalentă cu simularea experimentului de extragere la întamplare a unui element (obiect) din lista $x_0, x_1, \dots, x_{n-1}$ și returnarea lui în ”recipientul” din care a fost scos. La întamplare înseamnă că fiecare element are aceeași șansă de a fi extras, adică probabilitatea $1/n$.

Generalizare

Algoritm ce extrage un număr la întâmplare din mulțimea de numere întregi $\{m, m+1, \dots, n\}$**,** $m < n$.

- Mulțimea conține $N = n - m + 1$ elemente.
- Un număr selectat la întamplare este o valoare de observație asupra variabilei aleatoare:

$$X = \begin{pmatrix} m & m+1 & \dots & n \\ \frac{1}{n-m+1} & \frac{1}{n-m+1} & \dots & \frac{1}{n-m+1} \end{pmatrix}.$$

```
Function randint(m,n)
u=urand();
k=int((n-m+1)*u));//k in {0,1,2,...,n-m}
return k+m;
end.
```

Aplicație: Generarea unei permutari aleatoare[2]

[2] a se vedea notițe curs 8.

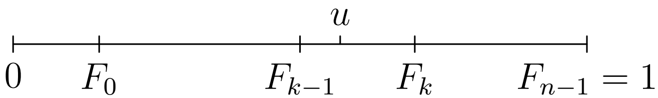
Simularea v.a. discrete cu o distribuție neuniformă

Fie X o v.a. discretă cu valorile ordonate, adică $x_0 < x_1 < \cdots < x_{n-1}$.

$$X = \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \\ p_0 & p_1 & \dots & p_{n-1} \end{pmatrix}, \quad \sum_{k=0}^{n-1} p_k = 1$$

- divizăm intervalul $[0, 1)$, prin punctele

$$F_0 = p_0, \quad F_1 = p_0 + p_1, \dots, F_k = p_0 + p_1 + \cdots + p_k, \dots,$$
$$F_{n-1} = 1$$



- Avem:

$$P(U \in (F_{k-1}, F_k]) = F_k - F_{k-1}{}_{lung.interv.} = p_k = P(X = x_k)$$

(am considerat $F_{-1} = 0$).

Simularea v.a. discrete cu o distribuție neuniformă


```
Function SimDiscret (x, p, n)//x, p sunt tablouri de n elemente
k = 0;
F = p_0;
u=urand();
while (u > F){
        k = k + 1;
        F = F + p_k;
        }
return x_k;
end.
```

Obs.: metodă recomand. pentru v.a. cu un număr redus de valori; Altfel, se recomd. calculul $F_k = p_0 + p_1 + \cdots + p_k$, $k \in \{0, n-1\}$ și căutarea binară a interv. $(F_{k-1}, F_k]$ în care cade $u \in [0, 1)$, dat de `urand()`.

Simularea distribuției Bernoulli

Distribuția Bernoulli $X = \begin{pmatrix} 1 & 0 \\ p & 1-p \end{pmatrix}$ se simulează când avem de făcut o alegere dintre două alternative, codificate cu 1 și 0.

- se generează $u \in [0, 1)$, apelând $\texttt{u} = \texttt{urand}()$;
- Dacă $u \leq p$, atunci s-a produs evenimentul $(U \leq p)$, $P(U \leq p) = P(0 \leq U \leq p) = p - 0 = p$ (aceeași probab. ca $(X = 1)$). Putem presupune că s-a produs acesta și alegem 1.
- Dacă însă $u > p$, atunci facem alegerea codificată de 0.

```
Function Bernoulli(p);
u=urand();
if (u < p) return 1;
else return 0;
end
```

Simularea distribuției binomiale

Fie $X \sim Bin(n,p)$ cu $X = \begin{pmatrix} 0 & 1 & \dots & k & \dots & n \\ Pr_0 & Pr_1 & \dots & Pr_k & \dots & Pr_n \end{pmatrix}$

Obs.: se simulează conform algoritmului prezentat mai sus, singura problemă fiind calculul probabilităților $Pr_k = P(X = k) = C_n^k p^k (1-p)^{n-k}$ și a sumelor $F_k = Pr_0 + Pr_1 + \cdots + Pr_k$, $k \in \{0, n\}$.

Folosind $C_n^{k+1} = C_n^k \frac{n-k}{k+1}$, avem

$$Pr_{k+1} = P(X = k+1) = C_n^{k+1} p^{k+1} (1-p)^{n-k-1} = \frac{n-k}{k+1} \frac{p}{1-p} P(X = k)$$

Notând cu $c = \frac{p}{1-p}$, obținem

$$Pr_{k+1} = \frac{n-k}{k+1} c\, Pr_k, \quad Pr_0 = (1-p)^n$$

Simularea distribuției binomiale

FunctionBin(n, p)
$k = 0$;
$\mathtt{c} = \mathtt{p}/(1 - \mathtt{p})$;
$pr = (1 - p)^n$;
$F = pr$;
u=urand();
while $(u > F)\{$
    $pr = (c * (n - k)/(k + 1)) * pr$
    $F = F + pr$;
    $k = k + 1$;
    $\}$
return k;
end.

Obs.: Dacă parametrul n este foarte mare, atunci căutarea liniară nu este performantă. În acest caz se recomandă precalcularea $Pr_k = P(X = k)$, $k \in \{0, n\}$ și a sumelor $F_j = \sum_{j=0}^{k} pr_j$ și folosirea căutării binare.

Simularea distribuției geometrice

Fie X o variabilă aleatoare ce are distribuția geometrică de parametru $p \in (0,1)$, $X \sim \text{Geom}(p)$, $X = \begin{pmatrix} k \\ p(1-p)^{k-1} \end{pmatrix}, \quad k \in \mathbb{N} \setminus \{0\}$

Metoda directă de simulare= variabilă geometrică este asociată unui experiment Bernoulli:

```
FunctionGeom1(p)
k=0; // k contorul pentru incercarile Bernoulli
do {
      u=urand()
      k=k+1;
      } while(u > p);
return k
end
```

Obs.:

- Se execută blocul de instrucțiuni din bucla `do-while` atâta timp cât încercările sunt un eșec.
- La primul succes returnează numărul încercării respective.
- Dacă probabilitatea succesului p este mică numărul mediu de încercări până la primul succes este mare pentru că $M(X) = 1/p$ și în acest caz algoritmul `Geom1` este ineficient.

Propoziție

Dacă $p \in (0,1)$ și $U \sim \mathrm{Unif}[0,1)$, atunci variabila aleatoare:

$$X = \left[\frac{\ln(1-U)}{\ln(1-p)}\right] + 1$$

are distribuția geometrică de parametru p.

Simularea distribuției geometrice

Deci, pentru variabila aleatoare $X \sim \text{Geom}(p)$, evenimentul $(X = k)$ are aceeași probabilitate ca evenimentul $\left(\left[\frac{\ln(1-U)}{\ln(1-p)}\right] + 1 = k\right)$,

```
FunctionGeom2(p)
u=urand();
return int(log(1 - u)/log(1 - p)) + 1;
end.
```

Binemeritată pauză

Pauză!

Dupa pauză... Simularea variabilelor aleatoare continue

Metoda inversării

Metoda inversării de simulare a variabilelor aleatoare continue se aplică pentru **variabilele ce au funcția de repartiție inversabilă.**

Funcția de repartiție a unei variabile aleatoare continue, X, având densitatea de probabilitate f_X este funcția $F_X : \mathbb{R} \to \mathbb{R}$,

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)\, dx$$

- o funcție continuă;
- nedescrecătoare, adică dacă $x_1 < x_2$, atunci $F_X(x_1) \leq F_X(x_2)$)
- $\lim\limits_{x \to -\infty} F_X(x) = 0$, iar $\lim\limits_{x \to \infty} F_X(x) = 1$.

Funcția de repartiție F_U a unei variabile aleatoare, $U \sim Unif[0,1)$:

$$F_U(x) = \begin{cases} 0 & x < 0 \\ x & x \in [0,1) \\ 1 & x \geq 1 \end{cases}$$

Obs.:

- F_U restricționată la intervalul $[0,1)$ este funct. identică: $F_U(x) = x$.
- Prin transformări inversabile, $h : \mathbb{R} \to \mathbb{R}$, adecvat alese, un șir (u_n) de numere uniform distribuite pe $[0,1)$ poate fi transformat într-un șir $(x_n = h(u_n))$ ale cărui elemente sunt valori de observație asupra unei variabile aleatoare $X = h(U)$, cu $U \sim$Unif[0,1).
- Cea mai simplă metodă de transformare a șirului (u_n) este **metoda inversării**.

Metoda inversării

- se aplică pentru a genera numere pseudo–aleatoare ca valori de observație asupra unei variabile aleatoare X, ce are funcția de repartiție inversabilă.
- dacă F_X este strict crescătoare ,atunci ea este inversabilă.
- Pentru orice $u \in (0,1)$, $F_X^{-1}(u) \in \mathbb{R}$.
- u ca valoare de observație asupra unei variabile aleatoare $U \sim$Unif[0,1), x este valoare de observație asupra variabilei $F_X{-}1(U)$.
- determinăm distribuția de probabilitate a variabilei $Y = F_X^{-1}(U)$:

Metoda inversării

Propoziție

Fie $U \sim [0,1)$ și F_X o funcție de repartiție strict crescătoare și continuă pe intervalul de lungime minimă din $\mathbb{R}$, pe care variabila aleatoare X ia valori cu probabilitatea 1.
Atunci variabila aleatoare

$$Y = F_X^{-1}(U)$$

are aceeași funcție de repartiție ca și variabila X, adică Y și X sunt identic distribuite și nu se disting din punct de vedere probabilist.

Demo: $U \sim \text{Unif}[0,1) \Rightarrow F_U(x) = x, \;\; x \in [0,1)$ și $F_U(x) = 0$, în rest.
Vom determina funcția de repartiție G_Y a variabilei $Y = F_X^{-1}(U)$:

$$G_Y(x) = P(Y \leq x) = P(F_X^{-1}(U) \leq x) = P(U \leq F_X(x)) = F_U(F_X(x)) = F_X($$

Prin urmare $G_Y(x) = F_X(x), \quad \forall x$, adică funcția de repartiție a variabilei aleatoare $Y = F_X^{-1}(U)$ este chiar F_X.

Algoritm de simulare

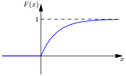
Algoritm de simulare a unei variabile aleatoare X ce are funcția de repartiție F_X inversabilă:

```
FunctionMetodaInversarii()
u=urand();
x = F_X^{-1}(u);
return x;
end.
```

Simularea distribuției exponențiale de parametru θ

Fie

- Funcția de repartiție: $F_X(x) = \begin{cases} 0 & \text{dacă } x < 0 \\ 1 - e^{-x/\theta} & \text{dacă } x \geq 0 \end{cases}$



- F_X este strict crescătoare pe intervalul $[0, \infty)$
- $\forall u \in [0,1)$, $F^{-1}(u) \in [0, \infty)$.
- O v. a. exponențial distribuită ia valori pozitive cu probabilitatea 1: $P(X \geq 0) = 1 - P(X < 0) = 1 - F_X(0) = 1 - 0 = 1$
- prin metoda inversării putem simula X.
- Din $1 - e^{-x/\theta} = u$, avem $x = F^{-1}(u) = -\theta \ln(1-u)$

Algoritm de simulare a distribuției exp.


```
Function SimulExp(theta)
u=urand();
x = -theta * log(1 - u);
return x;
end
```

Obs.: Dacă (u_n), $n = 0, 1, \ldots N$ este un șir de numere pseudo–aleatoare uniform distribuit pe $[0, 1)$, atunci șirul $(1 - u_n)$, este de asemenea uniform distribuit pe $[0, 1)$.

Astfel, în simularea unei variabile aleatoare $X \sim Exp(\theta)$ putem înlocui pe $1 - u$ cu u.