Cursul 9

Simularea distribuţiilor de probabilitate

9.1 Simularea distribuţiilor de probabilitate discrete

A simula o variabilă aleatoare discretă X presupune a genera independent, conform unui algoritm, un şir de numere $y_1, y_2, \dots, y_N$ care să aibă particularităţile unor valori de observaţie asupra variabilei. Mai precis, dacă variabila aleatoare X are distribuţia de probabilitate

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

atunci distribuţia experimentală a datelor generate trebuie să fie foarte apropiată de distribuţia teoretică a variabilei X. Şi anume, dacă notăm cu nr_k numărul valorilor generate egale cu x_k, $k = \overline{1, n}$, atunci raportul nr_k/N trebuie să fie apropiat de p_k.

În continuare prezentăm metode de simulare a variabilelor aleatoare ce au diverse distribuţii de probabilitate. Pentru fiecare distribuţie de probabilitate dăm pseudocodul pentru o funcţie ce returnează o singură valoare de observaţie simulată. O astfel de valoare se obţine printr-o transformare aplicată unei valori `u` presupusă a fi returnată de un generator de numere pseudo–aleatoare uniform distribuite pe $[0, 1)$.

Apelul acestui generator îl simbolizăm în continuare prin `urand()`. Acesta este un nume generic pentru simulatorul unei variabile aleatoare $U \sim Unif[0, 1)$.

Reamintim:

> Dacă $U \sim Unif[0, 1)$, atunci pentru orice interval $I \subset [0, 1)$, probabilitatea ca U să ia valori în acest interval este egală cu lungimea intervalului.

Dacă A este un eveniment de probabilitate $P(A) = 0.65$ şi $B = \overline{A}$, atunci putem simula producerea unuia din cele două evenimente astfel:

• apelăm generatorul `urand()`:

$$u = \texttt{urand}();$$

• Dacă $u < 0.65$, spunem că s-a produs evenimentul $(U < 0.65)$, care are probabilitatea $P(U < 0.65) = P(U \in [0, 0.65)) = 0.65$, adică lungimea intervalului. Cum şi A are aceeaşi probabilitate, putem considera că s-a produs evenimentul A.

• Dacă $u \in [0.65, 1)$, atunci s-a produs evenimentul $(U \in [0.65, 1))$ a cărui probabilitate este $P(U \in [0.65, 1)) = 1 - 0.65 = 0.35$. Deci, în acest caz, putem considera că s-a produs evenimentul B.

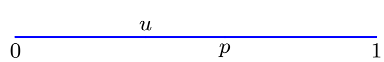
Această modalitate de a genera evenimente se foloseşte foarte mult în algoritmii randomizaţi `http://en.wikipedia.org/wiki/Randomized_algorithm`.

9.1.1 Simularea variabilei Bernoulli

Reamintim că o variabilă aleatoare Bernoulli are distribuţia

$$X = \begin{pmatrix} 1 & 0 \\ p & 1-p \end{pmatrix}, \ p \in (0, 1).$$

Ea se simulează când avem de făcut o alegere din două alternative, codificate cu 1, respectiv 0. Pentru început, se generează $u \in [0, 1)$ apelând `u` = `urand`();



Dacă $u < p$, înseamnă că s-a produs evenimentul $(U < p)$, a cărui probabilitate este

$$P(U < p) = P(0 \le U < p) = p - 0 = p.$$

Dar cum şi evenimentul $(X = 1)$ are probabilitatea p, putem presupune că s-a produs acesta, deci facem alegerea codificată cu 1. Dacă însă $u \ge p$, atunci facem alegerea codificată cu 0.

Astfel, algoritmul de simulare a unei variabile aleatoare Bernoulli este:

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

Acest algoritm se poate aplica la generarea unui şir de biţi aleatori, la căutarea aleatoare într-un arbore binar etc.

9.1.2 Simularea distribuţiei uniforme discrete

Propoziţia 9.1.1 *Dacă $U \sim$Unif[0,1) este o variabilă aleatoare uniform distribuită pe intervalul $[0, 1)$ şi $n \in \mathbb{N}^*$, atunci variabila aleatoare $X = [nU]$ ($[x]$ notează partea întreagă a lui x) este o variabilă aleatoare discretă ce are distribuţia uniformă pe mulţimea $\{0, 1, \ldots, n-1\}$, adică*

$$X = \begin{pmatrix} 0 & 1 & \ldots & n-1 \\ \dfrac{1}{n} & \dfrac{1}{n} & \ldots & \dfrac{1}{n} \end{pmatrix}.$$

Demonstraţie: Deoarece variabila aleatoare U ia valori în $[0, 1)$, variabila nU ia valori în $[0, n)$ şi $[nU] \in \{0, 1, \ldots, n-1\}$. Deci, variabila $X = [nU]$ are ca mulţime de valori $\{0, 1, \ldots, n-1\}$.

Să arătăm că probabilitatea ca X să ia valoarea k, $k = \overline{0, n-1}$, este $P(X = k) = \frac{1}{n}$:

$$\begin{aligned} P(X = k) &= P([nU] = k) = P(k \leq nU < k+1) \\ &= P\left(\frac{k}{n} \leq U < \frac{k+1}{n}\right) \\ &= \frac{k+1}{n} - \frac{k}{n} = \frac{1}{n}. \end{aligned}$$

Am demonstrat astfel că X este o variabilă aleatoare uniform distribuită pe mulţimea $\{0, 1, \ldots, n-1\}$. □

Exploatând această relaţie dintre variabila aleatoare discretă X şi variabila aleatoare U, avem următorul algoritm ce returnează o valoare de observaţie asupra variabilei X pe mulţimea $\{0, 1, \ldots, n-1\}$:

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

Înlocuind `return k;` cu `return` x_k`;` se returnează o valoare de observaţie asupra unei variabile aleatoare discrete uniform distribuită pe o mulţime $\{x_0, x_1, \ldots, x_{n-1}\}$, adică asupra variabilei aleatoare:

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

Simularea acestei variabile aleatoare este echivalentă cu simularea experimentului de extragere la întâmplare a unui element (obiect) din lista $x_0, x_1, \ldots, x_{n-1}$ şi returnarea lui în "recipientul" din care a fost scos. La întâmplare înseamnă că fiecare element are aceeaşi şansă de a fi extras, adică probabilitatea $1/n$.

Algoritmul ce extrage un număr la întâmplare din mulţimea de numere întregi $M = \{m, m+1, \ldots, n\}$, $m < n$.

Mulţimea M conţine $N = n - m + 1$ elemente. Un număr selectat la întâmplare din această mulţime este o valoare de observaţie asupra variabilei aleatoare:

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

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

9.1.3 Simularea variabilelor aleatoare discrete având o distribuţie generală neuniformă

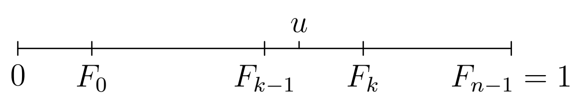
Fie X o variabilă aleatoare discretă ce are distribuţia de probabilitate

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

valorile sale sunt ordonate crescător, adică $x_0 < x_1 < \cdots < x_{n-1}$.

Ţinând seama că $\sum\limits_{k=0}^{n-1} p_k = 1$, divizăm intervalul $[0, 1)$ prin punctele

$$0,\ F_0 = p_0,\ F_1 = p_0 + p_1, \dots, F_k = p_0 + p_1 + \cdots + p_k, \dots, F_{n-1} = 1.$$



Probabilitatea ca un număr u, valoare de observaţie asupra variabilei $U \sim Unif[0, 1)$, să cadă într-un interval de forma $I_k = [F_{k-1}, F_k)$, $k = 1, 2, \dots, n-1$, este

$$P(U \in [F_{k-1}, F_k)) = \underbrace{F_k - F_{k-1}}_{lungime\,interval} = p_k = P(X = x_k),$$

respectiv $P(U \in I_0 = [0, F_0)) = F_0 - 0 = p_0$.

Rezultă că evenimentul $(X = x_k)$ se produce cu aceeaşi probabilitate ca şi evenimentul ca U să ia valori în intervalul I_k. Deci, generând u în intervalul I_k este echivalent cu producerea evenimentului $(X = x_k)$ şi în acest caz generatorul lui X returnează elementul x_k, $k = 0, 1, \dots, n-1$.

Algoritmul de generare a unui număr pseudo–aleator din legea de probabilitate a variabilei discrete X ce ia valorile x_k cu probabilităţile p_k, $k = 0, 1, \dots, n-1$, este:

```
function SimDiscret(x, p, n)//x, p sunt tablouri de n elemente
    k = 0;
    F = p0;
    u=urand();
    while (u ≥ F){
    k = k + 1;
```

$F = F + p_k$;
}
return x_k;
end function

Metoda prezentată mai sus, de căutare secvenţială a intervalului în care cade numărul u, este recomandată doar pentru simularea variabilelor aleatoare care au un număr redus de valori $x_0, x_1, \dots, x_{n-1}$.

Pentru variabilele aleatoare discrete cu număr mare de valori se recomandă calcularea prealabilă a sumelor $F_k = p_0 + p_1 + \cdots + p_k$, $k = \overline{0, n-1}$, şi căutarea binară a intervalului I_k în care cade numărul $u \in [0, 1)$, returnat de `urand()`.

9.1.4 Simularea distribuţiei geometrice

Fie X o variabilă aleatoare ce are distribuţia geometrică de parametru $p \in (0, 1)$. X dă numărul de încercări într-un experiment Bernoulli pâna la primul succes înregistrat, inclusiv. Distribuţia de probabilitate este ilustrată în tabloul:

$$X = \begin{pmatrix} k \\ p(1-p)^{k-1} \end{pmatrix}, \quad k \in \mathbb{N}^*.$$

Metoda directă de simulare, exploatând faptul că o variabilă geometrică este asociată unui experiment Bernoulli, este următoarea:

function Geom1(p)
k=0; // k contorul pentru încercările Bernoulli
do {
u=urand();
k=k+1;
}**while**($u > p$)**;**
return k;
end function

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

O modalitate mai rapidă de simulare a variabilei aleatoare X rezultă din următoarea propoziţie:

Propoziţia 9.1.2 *Dacă $p \in (0, 1)$ şi $U \sim Unif[0, 1)$, atunci variabila aleatoare*

$$X = \left[\frac{\ln(1-U)}{\ln(1-p)} \right] + 1 \tag{9.1}$$

are distribuţia geometrică de parametru p.

Demonstraţie: Notăm cu Y variabila aleatoare

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

Fie u o valoare de observaţie asupra variabilei aleatoare U, adică $u \in [0,1)$ este generat de `urand()`. Dacă $u = 0$, atunci $Y = 0$. În caz contrar, Y este un întreg pozitiv. Deci, Y este o variabilă aleatoare discretă ce ia valori în $\mathbb{N}$. În continuare arătăm că probabilitatea evenimentului $(Y = k)$ coincide cu probabiltatea unui eveniment E, asociat variabilei aleatoare U, pe care în plus îl putem şi simula. Are loc

$$P(Y=k) = P\left(\left[\frac{\ln(1-U)}{\ln(1-p)}\right] = k\right).$$

Dar cum partea întreagă $[x] = k$ implică $k \leq x < k+1$, avem

$$P(Y=k) = P\left(k \leq \frac{\ln(1-U)}{\ln(1-p)} < k+1\right).$$

Deoarece $\ln(1-p) < 0$, rezultă

$$\begin{aligned}
P(Y=k) &= P((k+1)\ln(1-p) < \ln(1-U) \leq k\ln(1-p)) \\
&= P(\ln(1-p)^{k+1} < \ln(1-U) \leq \ln(1-p)^k) \\
&= P((1-p)^{k+1} < 1-U \leq (1-p)^k) \\
&= P(-(1-p)^k \leq U-1 < -(1-p)^{k+1}) \\
&= P(1-(1-p)^k \leq U < 1-(1-p)^{k+1}).
\end{aligned}$$

Cum variabila U este uniform distribuit pe $[0,1)$ obţinem

$$P(1-(1-p)^k \leq U < 1-(1-p)^{k+1}) = 1-(1-p)^{k+1} - (1-(1-p)^k) = (1-p)^k p.$$

Pe de altă parte, $X = Y+1$, deci

$$P(X=k) = P(Y+1=k) = P(Y=k-1) = (1-p)^{k-1}p,$$

ceea ce implică faptul că variabila aleatoare X are distribuţia geometrică de parametru p. $\square$

Ţinând seama că pentru variabila aleatoare $X \sim \mathrm{Geom}(p)$, evenimentul $(X = k)$ are aceeaşi probabilitate ca evenimentul $\left(\left[\frac{\ln(1-U)}{\ln(1-p)}\right] + 1 = k\right)$, rezultă că se poate genera o valoare de observaţie x asupra lui X astfel:

function Geom2(p)
  u=urand();
  return $int(log(1-u)/log(1-p)) + 1$**;**
end function

9.2 Simularea distribuţiilor de probabilitate continue

9.2.1 Simularea distribuţiei uniforme pe intervalul $[a, b)$

Fie $a, b \in \mathbb{R}$ cu $a < b$. Simularea unei variabile aleatoare uniforme pe intervalul $[a, b)$ se bazează pe următorul rezultat:

Propoziţia 9.2.1 *Dacă* $U \sim Unif[0, 1)$, *atunci variabila aleatoare* $X = a + (b-a)U$ *este o variabilă aleatoare ce are distribuţia uniformă pe* $[a, b)$.

Demonstraţie: Vom arăta că funcţia de repartiţie F_X a variabilei X concide cu funcţia de repartiţie a distribuţiei uniforme pe $[a, b)$. Pentru $x \in [a, b)$, avem $\dfrac{x-a}{b-a} \in [0, 1)$. Reamintim că funcţia de repartiţie a lui U este $F_U(u) = u$, $\forall\ u \in [0, 1)$. Astfel, avem

$$\begin{aligned} F_X(x) = P(X \leq x) &= P(a + (b-a)U \leq x) = P\left(U \leq \frac{x-a}{b-a}\right) \\ &= F_U\left(\frac{x-a}{b-a}\right) = \frac{x-a}{b-a}. \end{aligned}$$

În mod analog se obţin şi celelalte două cazuri: $x < a$, respectiv $x \geq b$. □

Plecând de la rezultatul de mai sus, dăm următorul algoritm de simulare a unei valori corespunzătoare unei variabile aleatoare uniform distribuite pe intervalul $[a, b)$:

```
function Unif(a, b)
    u=urand();
    return a + (b − a) ∗ u;
end function
```

9.2.2 Simularea unei distribuţii de probabilitate prin metoda inversării

Metoda inversării de simulare a unei variabile aleatoare continue se aplică pentru acele variabile ce au funcţia de repartiţie inversabilă pe un anumit interval. Reamintim că funcţia de repartiţie a unei variabile aleatoare continue X, care admite densitatea de probabilitate f, este funcţia $F_X : \mathbb{R} \to [0, 1]$ definită prin

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)\, dt. \tag{9.2}$$

Funcţia de repartiţie F_X a unei variabile aleatoare continue X verifică următoarele proprietăţi:

- este continuă pe $\mathbb{R}$;

- este crecătoare, 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 $\lim\limits_{x\to\infty} F_X(x) = 1$.

De interes deosebit pentru simulare este funcţia de repartiţie F_U a unei variabile aleatoare $U \sim Unif[0,1)$,

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

Şirurile de numere pseudo–aleatoare uniform distribuite pe intervalul $[0,1)$ constituie baza oricărei simulări a unui fenomen sau proces aleator. Prin transformări inversabile $h : D \subset [0,1) \to \Delta \subset \mathbb{R}$, adecvat alese, un şir (u_n) de numere uniform distribuite pe $[0,1)$ poate fi transformat într-un şir $x_n = h(u_n)$ ale cărui elemente sunt valori de observaţie asupra unei variabile aleatoare $X = h(U)$ cu $U \sim Unif[0,1)$.

Cea mai simplă metodă de transformare a şirului (u_n) într-un şir (x_n) ca mai sus este metoda inversării. Aceasta se aplică pentru a genera numere pseudo–aleatoare ca valori de observaţie asupra unei variabile aleatoare X, ce are funcţia de repartiţie inversabilă pe un anumit interval I. Evident că dacă F_X este strict crescătoare pe I, atunci ea este inversabilă pe I. Interpretând u ca valoare de observaţie asupra unei variabile aleatoare $U \sim Unif[0,1)$, $x = h(u)$ este valoare de observaţie asupra variabilei $F_X^{-1}(U)$.

Propoziţia 9.2.2 *Fie U o variabilă aleatoare uniform distribuită pe $[0,1)$ şi F_X funcţia de repartiţie a variabilei aleatoare continue X. Dacă F_X este strict crescătoare (injectivă) pe un interval I din $\mathbb{R}$ pe care variabila aleatoare X ia valori cu probabilitatea 1, adică $P(X \in I) = 1$, atunci variabila aleatoare*

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

are aceeaşi funcţie de repartiţie ca şi variabila X, adică Y şi X sunt identic distribuite.

Demonstraţie: Cum $U \sim Unif[0,1)$, funcţia sa de repartiţie F_U este funcţia identică pe $[0,1]$, adică $F_U(x) = x$, $\forall\, x \in [0,1]$. Să determinăm funcţia de repartiţie F_Y a variabilei $Y = F_X^{-1}(U)$:

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

Prin urmare $F_Y(x) = F_X(x)$, pentru orice $x \in I$. Dar cum $P(X \in I) = 1$, rezultă că funcţia de repartiţie a variabilei aleatore $Y = F_X^{-1}(U)$ coincide din punct de vedere probabilist cu F_X. □

Bazat pe acest rezultat, avem următorul algoritm de simulare a unei variabile aleatoare X ce are funcţia de repartiţie F_X inversabilă:

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

Simularea distribuţiei exponenţiale de parametru $\theta > 0$: Funcţia de repartiţie a unei variabile aleatoare $X \sim \mathrm{Exp}(\theta)$ (Fig. 9.1) este:

$$F(x) = \begin{cases} 0, & \text{dacă } x < 0, \\ 1 - e^{-x/\theta}, & \text{dacă } x \geq 0. \end{cases} \tag{9.3}$$

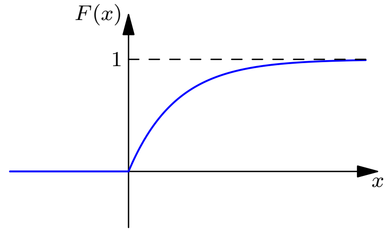


Fig.9.1: Funcţia de repartiţie a unei variabile aleatoare $X \sim Exp(\theta)$.

Observăm că funcţia F este strict crescătoare pe intervalul $[0, \infty)$, deci pentru orice $u \in [0, 1)$, $F^{-1}(u) \in [0, \infty)$. Cum o variabilă aleatoare exponenţial distribuită ia valori nenegative cu probabilitatea 1,

$$P(X \geq 0) = 1 - P(X < 0) = 1 - F(0) = 1 - 0 = 1,$$

rezultă că putem aplica metoda inversării pentru simularea lui X.

Din $1 - e^{-x/\theta} = u$, rezultă că

$$x = F^{-1}(u) = -\theta \ln(1 - u).$$

Astfel, putem simula variabila aleatoare X în modul următor:

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