Universitatea din București
Facultatea de Matematică și Informatică

CURS nr. 1
METODE MODERNE DE CALCUL ȘI SIMULARE

Recapitulare: noțiuni de probabilități

Lect. dr. Bianca Mogoș

Conținut

Experiment aleator

Definiție

Un *experiment aleator* este definit ca fiind o acțiune al cărei rezultat nu poate fi prezis cu certitudine și care se poate modifica în urma repetării experimentului.

Remarcă

Variabilitatea rezultatelor ieșite în urma unui experiment aleator poate apărea ca urmare a unor erori de măsurare, a alegerii unor obiecte diferite pentru testare, etc.

Exemplu

- Observarea pe un interval T de timp a funcționării unui calculator.
- Înregistrarea consumului de energie electrică a unui combinat.

Spațiu de selecție

Definiție

Spațiul de selecție al unui experiment aleator, notat prin Ω, reprezintă mulțimea tuturor rezultatelor posibile.

Exemplu

- Prin aruncarea banului se pot obține două rezultate. Astfel, spațiul de selecție este $\{0,1\}$.
- Prin rostogolirea unui zar cu șase fețe și numărarea punctelor de pe o față se pot obține șase rezultate posibile. Astfel, spațiul de selecție este $\{1,2,3,4,5,6\}$.

Eveniment. Eveniment sigur și eveniment imposibil

Definiție

Un *eveniment* este orice submulțime de rezultate conținute în spațiul de selecție. Un eveniment este elementar dacă el constă dintr-un singur rezultat și compus dacă constă din mai multe rezultate.

Definiție

Evenimentul sigur este evenimentul care se realizează întotdeauna ca rezultat al experimentului; va fi notat cu Ω (se asociază mulțimii totale de rezultate).

Evenimentul imposibil nu se poate realiza ca rezultat al unui experiment; va fi notat cu $\emptyset$ (corespunde mulțimii vide).

Reuniunea și intersecția evenimentelor

Definiție

Numim *reuniunea evenimentelor* $A_1, A_2, \ldots, A_k$, notată prin $\bigcup_{j=1}^{k} A_j$, evenimentul care se realizează când cel puțin unul dintre evenimentele $A_1, A_2, \ldots, A_k$ se realizează.

Numim *intersecția evenimentelor* $A_1, A_2, \ldots, A_k$, notată prin $\bigcap_{j=1}^{k} A_j$, evenimentul care se realizează când se realizează toate evenimentele $A_1, A_2, \ldots, A_k$.

Definiția empirică a probabilității

Probabilitatea unui eveniment este o măsură

- prin care modelăm *incertitudinea* producerii fenomenelor și a apariției datelor din lumea reală
- care ne permite să apreciem *gradul de încredere* și să cuantificăm lipsa de siguranță inerentă în procesul care generează datele analizate.

Definiție

(Definiția clasică a probabilității) Dacă într-un experiment cu "n" rezultate, "k" dintre ele favorizează realizarea evenimentului A, definim probabilitatea P(A) a evenimentului A prin

$$P(A) = \frac{k}{n} \tag{1}$$

Exercițiu: Aruncarea cu zarul

1. Calculați probabilitatea ca în urma aruncării unui zar să iasă un număr par.
2. Care este probabilitatea ca în urma aruncării unui zar să iasă un număr impar, strict mai mare ca 3?

σ – algebră

Fie Ω mulțimea evenimentelor elementare (rezultatelor posibile) ale unui experiment aleator.

Definiție

$\mathcal{B} \subset \mathcal{P}(\Omega)$ este o *σ – algebră* dacă

1. $\emptyset, \Omega \in \mathcal{B}$
2. Dacă $A \in \mathcal{B}$ atunci $A^C \in \mathcal{B}$
3. Dacă $(A_n)_{n \in N^*}, A_n \in \mathcal{B}$ atunci $\bigcup_{n=1}^{\infty} A_n \in \mathcal{B}$

Perechea $(\Omega, \mathcal{B})$ se numește *câmp de evenimente*.

Funcție de probabilitate

Definiție

Se numește *probabilitate* pe $\mathcal{B}$ o funcție nenegativă $P : \mathcal{B} \to [0, 1]$ cu proprietățile:

1. $P(\Omega) = 1$
2. Dacă $A, B \in \mathcal{B}$ și $A \cap B = \emptyset$ atunci $P(A \cup B) = P(A) + P(B)$.

Definiție

Tripletul $(\Omega, \mathcal{B}, P)$ se numește *câmp de probabilitate*.

Proprietăți ale funcției de probabilitate

1. $P(A^C) = 1 - P(A)$, deoarece $A \cup A^C = \Omega, A \cap A^C = \emptyset$.
2. $P(\emptyset) = P(\Omega^C) = 1 - P(\Omega) = 0$.
3. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$, dacă $A \cap B \neq \emptyset$.

Proprietatea 3. rezultă din relațiile (2) și (3).

$$\begin{aligned} A &= A \cap (B \cup B^C) = (A \cap B) \cup (A \cap B^C) \text{ cu } A \cap B \cap B^C = \emptyset \\ B &= B \cap (A \cup A^C) = (B \cap A) \cup (B \cap A^C) \text{ cu } B \cap A \cap A^C = \emptyset \\ A \cup B &= (A \cap B) \cup (A \cap B^C) \cup (B \cap A^C). \end{aligned} \tag{2}$$

Aplicăm axioma 2. din definiția probabilității și obținem

$$\begin{aligned} P(A) &= P(A \cap B) + P(A \cap B^C) \\ P(B) &= P(A \cap B) + P(B \cap A^C) \\ P(A \cup B) &= P(A \cap B) + P(A \cap B^C) + P(B \cap A^C). \end{aligned} \tag{3}$$

Exerciții

- *Problema controlului alimentelor*

  Se știe că într-o cutie sunt 550 de mere. Se verifică aleator 25 dintre ele dacă sunt stricate. Dacă 2% dintre merele din cutie sunt stricate, care este probabilitatea ca printre cele 25 testate să găsim 2 mere stricate?

- *Problema zilei de naștere*

  Care este probabilitatea ca cel puțin 2 persoane dintr-un grup de n indivizi să aibă aceeași zi de naștere?

Evenimente independente. Probabilitate condiționată

Definiție

Spunem că evenimentele A și B din $\mathcal{B}$ sunt *independente* dacă nu se influențează, adică realizarea evenimentului A nu depinde de realizarea lui B și reciproc.

Definiție

Se numește *probabilitate condiționată a evenimentului A de către evenimentul B* și se notează $P(A|B)$ sau $P_B A$ probabilitatea de realizare a evenimentului A calculată în condiția că evenimentul B s-a realizat $(P(B) > 0)$ și se definește ca:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{4}$$

Exemplu 1: (1)

Un tratament s-a aplicat la 700 de pacienți care sufereau de piatră la rinichi. S-a constatat că doar 562 s-au vindecat. De asemenea, se știe că 357 au pietre mici și 315 dintre aceștia s-au vindecat, iar 343 au pietre mari și 247 s-au vindecat. Să se calculeze probabilitatea ca un pacient să se vindece știind că acesta are piatră mică. Dar, dacă are piatră mare?

Soluție:

- Fie

  $\Omega = \{1, 2, \ldots, 700\}$ mulțimea tuturor pacienților,

  $S = \{1, 2, \ldots, 562\}$ mulțimea pacienților vindecați și

  $E = \{563, \ldots, 700\}$ mulțimea pacienților nevindecați.

Exemplu 1: (2)

- Obținem următoarele probabilități:

  $P(\omega) = \frac{1}{700}, \forall$ evenimentul elementar $\omega \in \Omega$

  $P(S) = \frac{562}{700} \approx 0,8$

  $P(E) = \frac{138}{700} = \approx 0,2$

- Notăm cu

  $\Omega_1 =$ pacienții care au pietre mici

  $\Omega_2 =$ pacienții care au pietre mari

- Rezultă $P(\Omega_1) = \frac{357}{700} = 0,51$ și $P(\Omega_2) = \frac{343}{700} = 0,49$. Astfel

$$P(S|\Omega_1) = \frac{P(S \cap \Omega_1)}{P(\Omega_1)} = \frac{315}{700} \cdot \frac{1}{0,51} = 0,88$$

Analog, se arată că $P(S|\Omega_2) = 0,72$.

Formula lui Bayes (1)

Fie $(A_i)_{i=\overline{1,k}}$ o partiție a mulțimii Ω, ceea ce înseamnă că

$$A_i \cap A_j = \emptyset, \forall i, j \in \overline{1,k}, i \neq j, P(A_j) > 0, \forall j = \overline{1,k} \text{ și } \bigcup_{j=1}^{k} A_j = \Omega. \tag{5}$$

Fie $X \in \mathcal{B}$ un eveniment oarecare. Atunci

$$\begin{aligned} P(X) = P(X \cap \Omega) = P(\bigcup_{j=1}^{k} X \cap A_j) &= \sum_{j=1}^{k} P(X \cap A_j) \\ &= \sum_{j=1}^{k} P(A_j) \cdot P(X|A_j) \end{aligned} \tag{6}$$

Obținem relația

$$P(X) = \sum_{j=1}^{k} P(A_j) \cdot P(X|A_j) \tag{7}$$

Formula lui Bayes (2)

Atunci *probabilitatea condiționată a evenimentului A_j de către evenimentul X* se poate calcula cu formula

$$P(A_j|X) = \frac{P(A_j) \cdot P(X|A_j)}{\sum_{j=1}^{k} P(A_j) \cdot P(X|A_j)}. \tag{8}$$

Relația obținută în ecuația (8) se numește *formula lui Bayes*.

Variabilă aleatoare

Fie $(\Omega, \mathcal{B}, P)$ un câmp de probabilitate.

Definiție

Numim *variabilă aleatoare (v.a.)* o funcție $X : \Omega \to \mathbb{R}$ cu proprietatea că pentru fiecare $x \in \mathbb{R}$ evenimentul

$$X^{-1}(x) = \{\omega \in \Omega | X(\omega) \leq x\} \text{ aparține mulțimii } \mathcal{B}. \tag{9}$$

Exemplu

Pentru $A \in \mathcal{B}$, funcția indicator I_A definită prin

$$I_A(\omega) = \begin{cases} 1, & \omega \in A \\ 0, & \omega \notin A \end{cases} \tag{10}$$

este o variabilă aleatoare.

Funcție de repartiție. Proprietăți ale funcției de repartiție

Fie evenimentul $A = \{\omega \in \Omega | X(\omega) \leq x\} \stackrel{not}{=} \{X \leq x\} \in \mathcal{B}$.

Definiție

Funcția $F_X : \mathbb{R} \to [0, 1]$ definită prin

$$F_X(x) = P(\{X \leq x\}) \stackrel{not}{=} P(X \leq x) \tag{11}$$

se numește *funcția de repartiție* a variabilei aleatoare X.

Proprietăți ale funcției de repartiție

1. $F_X(x) \in [0, 1]$, $\lim\limits_{x \to \infty} F_X(x) = 1$, $\lim\limits_{x \to -\infty} F_X(x) = 0$.
2. $F_X(x + h) \geq F_X(x)$, pentru $h > 0$, deoarece
$$F_X(x + h) = F_X(x) + P(\{\omega \in \Omega | x < X(\omega) \leq x + h\}).$$
3. $P(\{\omega \in \Omega | a < X(\omega) \leq b\}) = F_X(b) - F_X(a)$, pentru $a < b$.

Variabilă aleatoare discretă

- Spunem că X este o *variabilă aleatoare discretă* dacă valorile luate de variabila aleatoare X sunt în număr finit sau numărabil.
- O v.a. discretă este definită prin tabloul:
$$X : \begin{pmatrix} x_1 & x_2 & \dots & x_n & \dots \\ p_1 & p_2 & \dots & p_n & \dots \end{pmatrix} \tag{12}$$
unde $x_1, x_2, \dots, x_n, \dots$ reprezintă valorile distincte pe care le poate lua v.a. X, iar $p_n = P(\{\omega \in \Omega | X(\omega) = x_n\}), n = 1, 2, \dots$
- Tabloul din ecuația (12) se numește *repartiția variabilei aleatoare discrete X*.

Proprietăți ale variabilelor aleatoare discrete

1. $p_1 + p_2 + \ldots + p_n + \ldots = 1$, deoarece

   $$p_1 + p_2 + \ldots + p_n + \ldots = P(\{\omega \in \Omega | X(\omega) = x_1\}) + \\ + P(\{\omega \in \Omega | X(\omega) = x_2\}) + \ldots + P(\{\omega \in \Omega | X(\omega) = x_n\}) + \\ + \ldots = P(\Omega) = 1$$

   deoarece evenimentele

   $$A_i = \{\omega \in \Omega | X(\omega) = x_i\} \text{ și } A_j = \{\omega \in \Omega | X(\omega) = x_j\} \text{ cu } i \neq j$$

   sunt disjuncte (adică, $A_i \cap A_j = \emptyset$; altfel avem evenimentul $e \in A_i \cap A_j$, de unde rezultă $X(e) = x_i = x_j$, ceea ce este imposibil pentru $i \neq j$) și $\bigcup_i A_i = \Omega$

2. O variabilă aleatoare discretă realizează o partiție a spațiului de selecție.

Operații cu variabile aleatoare discrete (1)

Definiție

Dacă X și Y sunt definite pe același câmp de probabilitate $(\Omega, \mathcal{B}, P)$ atunci:

$$(X + Y)(\omega) = X(\omega) + Y(\omega)$$
$$(aX)(\omega) = aX(\omega), \text{ pentru } a \in \mathbb{R}$$
$$(XY)(\omega) = X(\omega)Y(\omega)$$
$$\frac{X}{Y}(\omega) = \frac{X(\omega)}{Y(\omega)}, \textit{ dacă } Y(\omega) \neq 0 \text{ pentru orice } \omega \in \Omega$$

Operații cu variabile aleatoare discrete (2)

Dacă

$$X : \begin{pmatrix} x_1 & x_2 & \ldots & x_n \\ p_1 & p_2 & \ldots & p_n \end{pmatrix}, Y : \begin{pmatrix} y_1 & y_2 & \ldots & y_m \\ q_1 & q_2 & \ldots & q_m \end{pmatrix}$$

atunci

$$X + Y : \begin{pmatrix} x_1 + y_1 & x_1 + y_2 & \ldots & x_i + y_j & \ldots & x_n + y_m \\ p_{11} & p_{12} & \ldots & p_{ij} & \ldots & p_{nm} \end{pmatrix}$$

unde $p_{ij} = P(\{\omega \in \Omega | X(\omega) = x_i\} \cap \{\omega \in \Omega | Y(\omega) = y_j\})$.

$$aX : \begin{pmatrix} ax_1 & ax_2 & \ldots & ax_n \\ p_1 & p_2 & \ldots & p_n \end{pmatrix}$$

$$X \cdot Y : \begin{pmatrix} x_1 y_1 & x_1 y_2 & \ldots & x_i y_j & \ldots & x_n y_m \\ p_{11} & p_{12} & \ldots & p_{ij} & \ldots & p_{nm} \end{pmatrix}$$

și

$$X/Y : \begin{pmatrix} x_1/y_1 & x_1/y_2 & \ldots & x_i/y_j & \ldots & x_n/y_m \\ p_{11} & p_{12} & \ldots & p_{ij} & \ldots & p_{nm} \end{pmatrix}$$

Caracteristici ale v.a. discrete – definiții (1)

- Fie $(\Omega, \mathcal{B}, P)$ spațiul de probabilitate pe care este definită variabila aleatoare discretă X.
- Dacă X este o variabilă aleatoare, $X : \Omega \to \mathbb{R}$ atunci numim *valoare medie* a variabilei X,

  $$E[X] = \sum_{i=1}^{\infty} x_i p_i \qquad (13)$$

  pentru cazul în care X ia un număr numărabil de valori și seria este convergentă.
- Dacă X ia un număr finit de valori, valoarea medie se definește printr-o sumă finită.

Caracteristici ale v.a. discrete – definiții (2)

- *Momentul de ordinul r* al v.a. X este $E_r[X] = E[X^r]$. Se mai folosește notația $\mu_r(X) = E[X^r]$.
- *Momentul centrat de ordin r* al v.a. X este numărul
$$\mu_r(X) = E[(X - E[X])^r] \tag{14}$$
- Pentru $r = 2$
$$\mu_2(X) = E[(X - E[X])^2] \overset{not}{=} D(X) \tag{15}$$
se numește *dispersia* variabilei X.
- Dispersia $D(X)$ dă o măsură a împrăștierii valorilor variabilei X în jurul valorii ei medii.

Proprietăți ale mediei unei v.a. discrete (1)

1. $E[aX + b] = aE[X] + b, a, b \in \mathbb{R}$
2. $E[X + Y] = E[X] + E[Y]$

Demonstrație

$$E[X+Y] = \sum_{i=1}^{n}\sum_{j=1}^{m}(x_i + y_j)p_{ij} = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} p_{ij} + \sum_{j=1}^{m} y_j \sum_{i=1}^{n} p_{ij}$$
$$= \sum_{i=1}^{n} x_i P(X = x_i) + \sum_{j=1}^{m} y_j P(Y = y_j),$$

deoarece

$$\sum_{j=1}^{m} p_{ij} = P\left((X = x_i) \cap \Omega\right) = P(X = x_i)$$
$$\sum_{i=1}^{n} p_{ij} = P\left(\Omega \cap (Y = y_j)\right) = P(Y = y_j).$$

Proprietăți ale mediei unei v.a. discrete (2)

3. Dacă X și Y sunt v.a. independente atunci
$$E[X \cdot Y] = E[X] \cdot E[Y]$$

Demonstrație

Avem $E[X \cdot Y] = \sum_{i=1}^{n}\sum_{j=1}^{m} x_i y_j p_{ij}$. Cum X și Y sunt independente
$$p_{ij} = P(X = x_i) \cdot P(Y = y_j).$$

Astfel,

$$E[X \cdot Y] = \sum_{i=1}^{n} x_i P(X = x_i) \cdot \sum_{j=1}^{m} y_j P(Y = y_j).$$

Variabilă aleatoare continuă

Definiție

Spunem că $X : \Omega \to \mathbb{R}$ este *v.a. continuă* dacă
$$P(\{\omega \in \Omega | X(\omega) = x\}) = 0, \tag{16}$$
pentru orice $x \in \mathbb{R}$.

Observații:

- Deși probabilitatea evenimentului din definiție este zero nu înseamnă că evenimentul nu se poate realiza.
- Dacă X este o variabilă aleatoare continuă atunci funcția sa de repartiție este continuă.

Densitate de repartiție. Proprietăți ale funcției de repartiție

Definiție

Dacă există

$$\lim_{h\to 0}\frac{F_X(x+h)-F_X(x)}{h}=F'_X(x) \tag{17}$$

atunci F'_X se numește *densitatea de repartiție* a variabilei X și

$$F'_X(x)=f_X(x), x\in\mathbb{R} \tag{18}$$

Proprietăți ale funcției de repartiție

1. Deoarece $F_X(x)=\int_{-\infty}^{x} f_X(t)dt$ rezultă $\int_{-\infty}^{\infty} f_X(t)dt=1$
2. Deoarece F_X este crescătoare rezultă că $f_X(x)\geq 0, x\in\mathbb{R}$.

Caracteristici ale v.a. continue – definiții (1)

- *Valoarea medie* a v.a. continue X este
$$E[X]=\int_{-\infty}^{\infty} x dF_X(x)=\int_{-\infty}^{\infty} x f_X(x)dx \tag{19}$$
- *Momentul de ordinul r* al variabilei aleatoare continue X este
$$\mu_r(X)=E[X^r]=\int_{\mathbb{R}} x^r f_X(x)dx. \tag{20}$$
- *Momentul centrat de ordinul r* al variabilei aleatoare continue X este
$$\mu_r(X)=E[(X-E[X])^r]=\int_{\mathbb{R}}(x-E[X])^r f_X(x)dx. \tag{21}$$
- Pentru $r=2$ obținem *dispersia* variabilei aleatoare X:
$$D(X)=\mu_2(X)=E[(x-E[X])^2]=\int_{\mathbb{R}}(x-E[X])^2 f_X(x)dx. \tag{22}$$

Caracteristici ale v.a. continue – definiții (2)

- $\sqrt{D(X)}$ se numește *abaterea medie pătratică* sau *deviația standard*.
- *Modul* v.a. X, notat prin M_0, este acea valoare a v.a. X pentru care densitatea de probabilitate f_X are valoare maximă.
- *Mediana* v.a. X, notată prin Me, este acea valoare pentru care
$$\int_{-\infty}^{Me} f_X(x)dx=\int_{Me}^{\infty} f_X(x)dx=\frac{1}{2}. \tag{23}$$
- *Cuantila de ordinul α* a variabile aleatoare X este acea valoare q_α care verifică relația
$$\int_{-\infty}^{q_\alpha} f_X(x)dx=\alpha. \tag{24}$$

Test

Timpul de așteptare la un ghișeu este o variabilă aleatoare T care urmează o repartiție Exp(λ) cu timpul mediu de așteptare de 15 minute. Să se afle probabilitatea ca un cetățean să aștepte mai mult de 20 de minute.

Universitatea din București
Facultatea de Matematică și Informatică

CURS nr. 2

METODE MODERNE DE CALCUL ȘI SIMULARE

1. Recapitulare: noțiuni de statistică
2. Generarea numerelor aleatoare

Lect. dr. Bianca Mogoș

Conținut

Scopul experimentului aleator

- *Experimentul aleator* se realizează pentru *colectarea de date* necesară pentru a obține informații privind un anumit fenomen de interes.
- Pe baza datelor se emit *concluzii* care, în general, ies din sfera experimentului particular.
- Cercetătorii *generalizează concluziile experimentului* pentru clasa tuturor experimentelor similare.
- Problema acestui demers este că *nu putem garanta corectitudinea concluziilor* obținute.
- Totuși, folosind tehnici statistice, putem măsura și administra *gradul de incertitudine* al rezultatelor.

Statistică inferențială

- *Statistica inferențială* este o colecție de *metode* care permit cercetătorilor să observe o submulțime a obiectelor de interes și folosind informația obținută pe baza acestor observații să facă afirmații sau inferențe privind întreaga populație.
- Câteva dintre aceste *metode* sunt:
  - estimarea parametrilor unei populații
  - verificarea ipotezelor statistice
  - estimarea densității de repartiție.

Populație țintă

- *Populația țintă* este definită ca fiind întreaga colecție de obiecte sau indivizi despre care vrem să obținem anumite informații.
- Populația țintă trebuie *bine definită* indicând
  - ce constituie *membrii* acesteia (de ex, populația unei zone geografice, o anumită firmă care construiește componente hardware, etc.)
  - *caracteristicile* populației (de ex, starea de sănătate, numărul de defecțiuni, etc.).
- In majoritatea cazurilor este imposibil sau nerealist să observăm întreaga populație; cercetătorii măsoară numai o parte a populației țintă, denumită *eșantion*.
- Pentru a face inferențe privind întreaga populație este important ca *mulțimea eșantion* să fie *reprezentativă* relativ la întreaga populație.

Model probabilist

- Fie X o v.a. cu densitatea $f(x,\theta), x \in \mathbb{R}, \theta \in \mathbb{R}$.

Definiție

Mulțimea densităților de repartiție $f(x,\theta), \theta \in \Theta \subset \mathbb{R}$, ce depind de parametrul θ se numește *model probabilist unidimensional*.

$$\{f(x,\theta) | x \in \mathbb{R}, \theta \in \Theta\}. \tag{1}$$

- Fie $X = (X_1, X_2, \ldots, X_n)$ un vector aleator cu densitatea de repartiție

$$f(x_1, x_2, \ldots, x_n; \theta), (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n, \theta \in \mathbb{R}^k.$$

Definiție

Mulțimea densităților de repartiție $f(x_1, x_2, \ldots, x_n; \theta)$ cu parametrul $\theta \in \Theta \subset \mathbb{R}^k$ se numește *model probabilist multidimensional*.

$$\{f(x_1, x_2, \ldots, x_n; \theta) | \theta \in \Theta\}. \tag{2}$$

Selecție. Repartiția selecției

Definiție

O *selecție* este o mulțime de v.a. $X_1, X_2, \ldots, X_n$ având aceeași densitate de repartiție $f(x,\theta)$.

- Deoarece selecția este o mulțime de variabile aleatoare asociate unui model probabilist, selecția trebuie să aibă o repartiție, pe care o vom numi *repartiția selecției*.

Definiție

Repartiția selecției $X_1, X_2, \ldots, X_n$ este definită ca fiind repartiția vectorului $X = (X_1, X_2, \ldots, X_n)$, notată prin $f(x_1, x_2, \ldots, x_n; \theta)$.

- Cea mai folosită formă de selecție este selecția aleatoare și este bazată pe ideea experimentului aleator.

Selecție aleatoare

Definiție

Spunem că $X_1, X_2, \ldots, X_n$ este o *selecție aleatoare* asupra v.a. X care are densitatea de repartiție $f(x;\theta)$ dacă $X_1, X_2, \ldots, X_n$ sunt v.a. independente și identic repartizate ca X.

$X_1, X_2, \ldots, X_n$ se numesc *variabile de selecție*.

- În cazul selecției aleatoare, densitatea de repartiție comună a variabilelor de selecție este

$$f(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f(x_i; \theta)$$

- O selecție aleatoare poate fi construită prin repetarea unui experiment aleator de n ori.
- Un *rezultat al selecției aleatoare* se notează prin $(x_1, x_2, \ldots, x_n)$ și mulțimea tuturor rezultatelor definesc *spațiul observațiilor* $S \equiv \mathbb{R}^n$.

Model statistic. Statistică

Definiție

Modelul probabilist

$$\{f(x;\theta), \theta \in \Theta)\}$$

împreună cu selecția $X = (X_1, X_2, \dots, X_n)$ definesc *modelul statistic*.

Definiție

Statistica este o funcție $t_n : S \to \Theta \subset \mathbb{R}^k$ care nu conține niciun parametru necunoscut.

- Cele mai utilizate statistici sunt momentele de selecție.

Momente de selecție

- *Momentul de selecție de ordinul r*

$$m'_r(x_1, x_2, \dots, x_n) = \frac{1}{n}\sum_{j=1}^{n} x_j^r \tag{3}$$

- *Momentul de selecție de ordin 1 – media de selecție*

$$m'_1(x_1, x_2, \dots, x_n) = \overline{X}_n = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{4}$$

- *Momentul centrat de selecție de ordin r*

$$m_r(x_1, x_2, \dots, x_n) = \frac{1}{n}\sum_{j=1}^{n} (x_j - \overline{X}_n)^r. \tag{5}$$

- *Momentul centrat de selecție de ordin 2 – dispersia de selecție*

$$m_2(x_1, x_2, \dots, x_n) = \frac{1}{n}\sum_{j=1}^{n} (x_j - \overline{X}_n)^2. \tag{6}$$

Convergență în probabilitate

Definiție

Șirul de v.a. $(X_n)_n$ *converge în probabilitate* la v.a. X dacă

$$\lim_{n\to\infty} P\left(\{\omega \in \Omega, |X_n(\omega) - X(\omega)| < \epsilon\}\right) = 1. \tag{7}$$

Propoziție

Avem relația

$$\overline{X}_n = \frac{1}{n}\sum_{j=1}^{n} X_j \xrightarrow[n\to\infty]{P} E[X] = \mu$$

Estimator. Estimație

Definiție

Se numește *estimator*, variabila aleatoare

$$t_n(X) : \Omega \to \Theta, \tag{8}$$

unde $X = (X_1, X_2, \dots, X_n)$, Ω este spațiul de selecție; $t_n(x), x \in S$, S spațiul observațiilor, se numește *estimație*.

Definiție

Un estimator $t_n = t_n(X)$ se numește *consistent* pentru θ dacă

$$\lim_{n\to\infty} P(|t_n - \theta| < \epsilon) = 1 \tag{9}$$

și notăm $t_n \xrightarrow{P} \theta$.

Consistența unui estimator

- Consistența unui estimator reprezintă o proprietate asimptotică a estimatorului.
- Un estimator bun pentru parametrul θ trebuie să aibă o repartiție cu o valoare centrală în vecinătatea lui θ.
- Definiția următoare cere ca estimatorul să aibă o valoare centrală în vecinătatea lui θ nu numai pentru valori mari ale lui n, ci pentru orice n.

Definiție

Estimatorul t_n se numește *nedeplasat* pentru θ dacă

$$E[t_n] = \theta. \tag{10}$$

Numere aleatoare

- Majoritatea metodelor de generare a v.a. se bazează pe generarea unor numere aleatoare uniform distribuite pe intervalul (0,1).
- Datorită calculatorului avem posibilitatea de a genera foarte ușor numere aleatoare uniforme.
- Totuși, trebuie cunoscut faptul că numerele generate de calculator sunt pseudo-aleatoare, deoarece acestea sunt generate cu un algoritm determinist.

Numere aleatoare uniform distribuite

Fie $(\Omega, \mathcal{B}, P)$ un câmp de probabilitate.

Definiție

(Variabilă aleatoare uniformă)
Spunem că *variabila aleatoare continuă* $U : \Omega \to \mathbb{R}$ este *repartizată uniform* pe intervalul $[a, b]$ și scriem $U \sim \mathcal{U}(a, b)$ dacă *densitatea sa de repartiție* este de forma:

$$f(u) = \begin{cases} \dfrac{1}{b-a}, & \text{dacă } u \in [a, b] \\ 0, & \text{altfel} \end{cases} \tag{11}$$

Funcția de repartiție $F(x)$ a v.a. $U \sim \mathcal{U}(a, b)$ este definită prin

$$F(x) = \begin{cases} 0, & \text{dacă } x \leq a \\ \dfrac{x-a}{b-a}, & \text{dacă } x \in (a, b) \\ 1, & \text{dacă } x \geq b \end{cases} \tag{12}$$

Observații referitoare la repartiția uniformă și v.a. uniforme

- Probabilitatea ca v.a. uniformă $U \sim \mathcal{U}(a, b)$ să cadă într-un interval $[u_1, u_2], a \leq u_1 < u_2 \leq b$ este proporțională cu lungimea intervalului:

$$P(u_1 < U \leq u_2) = \frac{u_2 - u_1}{b - a}. \tag{13}$$

- Pentru a genera numere aleatoare uniform distribuite pe un interval [a,b], și scriem $X \sim \mathcal{U}(a, b)$, pornind de la numere generate uniform pe intervalul [0,1] se poate folosi transformarea

$$X = (b - a) \cdot U + a, \tag{14}$$

unde $U \sim \mathcal{U}(0, 1)$.

Generarea numerelor uniforme în Matlab (1)

- În Matlab există funcția **rand** pentru generarea variabilelor aleatoare uniforme.
- **rand(**n**)**, unde n este un număr natural, returnează o matrice de dimensiune $n \times n$ având ca elemente numere aleatoare uniform distribuite între 0 și 1.
- **rand(**m, n**)**, unde m, n sunt numere naturale, returnează o matrice de dimensiune $m \times n$ având ca elemente numere aleatoare uniform distribuite între 0 și 1.

Generarea numerelor uniforme în Matlab (2)

- O secvență de numere aleatoare generată în Matlab depinde de "sămânța" sau starea generatorului. Starea este resetată la valoarea implicită în momentul pornirii Matlab-ului, astfel aceeași secvență de variabile aleatoare este generată la o nouă pornire a Matlab-ului. Acesta poate fi un avantaj în situațiile în care analistul are nevoie să reproducă rezultatele unei simulări pentru a verifica anumite concluzii.
- **rng(SD)** setează sămânța generatorului pe baza întregului nenegativ *SD*
- **rng('shuffle')** setează sămânța generatorului pe baza momentului curent de timp
- **rng('default')** resetează setările generatorului la valorile inițiale, și anume, se folosește generatorul Mersenne Twister cu sămânța 0.

Generarea numerelor uniforme în Matlab (4)

Exemplu care ilustrează utilizarea funcției **rand**.

% Generam un vector de numere aleatoare pe intervalul (0,1).
x = rand(1,1000);
% Histograma eșantionului generat in x.
[N, C] = hist(x,15);
% x: mulțimea eșantion
% 15 reprezintă numărul de dreptunghiuri ale histogramei
% N: vector conținând numărul de elemente din fiecare dintre dreptunghiuri.
% C: vector conținând centrele dreptunghiurilor
% Folosirea funcției bar pentru reprezentarea grafică a histogramei.
bar(C,N,1,'w')
title('Histograma asociata unei variabile aleatoare uniforme')
xlabel('X')
ylabel('Frecventa absoluta')

Histograma rezultată este prezentată în Figura 1.

Generarea numerelor uniforme în Matlab (5)

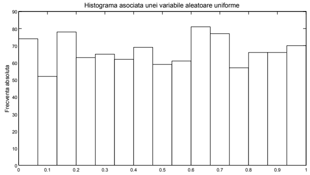


Figura : 1 – Histograma asociată unui eșantion de numere aleatoare uniform distribuite

Universitatea din București
Facultatea de Matematică și Informatică

CURS nr. 3

METODE MODERNE DE CALCUL ȘI SIMULARE

1. Algoritmi de generare a numerelor aleatoare

2. Metode generale de simulare a v.a.: metoda inversă

Lect. dr. Bianca Mogoș

Conținut

- Noțiuni ajutătoare (NA)
  - Proprietăți ale funcției de repartiție
  - Construcția câmpului de probabilitate $(\Omega, \mathcal{B}, P)$ pornind de la o funcție $F(x)$ ce îndeplinește 3 proprietăți de bază ale unei funcții de repartiție
  - Funcții de o variabilă aleatoare
- Structura generală a unui algoritm de generare/simulare a unei variabile aleatoare
- Algoritmi de generare a numerelor aleatoare
- Metode generale de simulare a variabilelor aleatoare: Metoda inversă

(NA): Proprietăți ale funcției de repartiție

Fie $(\Omega, \mathcal{B}, P)$ un câmp de probabilitate și $X : \Omega \to \mathbb{R}$ o variabilă aleatoare.

Definiție

Funcția $F_X : \mathbb{R} \to [0,1]$ definită prin

$$F_X(x) = P(\{\omega \in \Omega | X(\omega) \leq x\}) \stackrel{not}{=} P(X \leq x) \qquad (1)$$

se numește *funcția de repartiție* a variabilei aleatoare X.

Proprietăți ale funcției de repartiție

1. $F_X(x) \in [0,1]$, $\lim\limits_{x\to\infty} F_X(x) = 1$, $\lim\limits_{x\to-\infty} F_X(x) = 0$.
2. $F_X(x+h) \geq F_X(x)$, pentru $h > 0$, deoarece
$$F_X(x+h) = F_X(x) + P(\{\omega \in \Omega | x < X(\omega) \leq x+h\}).$$
3. $F_X(x)$ este o funcție continuă la dreapta, adică:
$$\lim_{\substack{h \to 0 \\ h > 0}} F_X(x+h) = F_X(x)$$

(NA): Construcția unui câmp de probabilitate $(\Omega, \mathcal{B}, P)$ și a unei variabilei aleatoare $X : \Omega \to \mathbb{R}$ pornind de la o funcție $F(x)$ ce îndeplinește 3 proprietăți de bază ale unei funcții de repartiție (1)

Ipoteză: Fie funcția $F(x)$ ce îndeplinește proprietățile 1., 2. și 3. prezentate pe slide-ul anterior.

- Fie spațiul de selecție $\Omega = (-\infty, \infty)$
- Definim $\mathcal{B}$ ca fiind σ – algebra generată de subintervalele de forma
$$(x_1, x_2], \quad (-\infty, x_2], \quad (x_1, \infty), \quad (\infty, \infty)$$
ale mulțimii $\Omega = (-\infty, \infty)$
- Definim variabila aleatoare (v.a.) $X : \Omega \to \mathbb{R}$ prin
$$X(\omega) = \omega, \forall \omega \in \Omega$$

(NA): Construcția unui câmp de probabilitate $(\Omega, \mathcal{B}, P)$ și a unei variabilei aleatoare $X : \Omega \to \mathbb{R}$ pornind de la o funcție $F(x)$ ce îndeplinește 3 proprietăți de bază ale unei funcții de repartiție (2) – continuare

- Definim funcția de probabilitate $P : \mathcal{B} \to [0, 1]$ prin

$$P(x_1 < X \leq x_2) = F(x_2) - F(x_1), \forall x_1, x_2 \in \mathbb{R}, x_1 < x_2$$
$$P(-\infty < X \leq x_2) = F(x_2)$$
$$P(x_1 < X < \infty) = 1 - F(x_1)$$
$$P(-\infty < X < \infty) = 1$$
$$P(\bigcup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P(A_n), \text{pentru } A_n \in \mathcal{B}, A_i \cap A_j = \emptyset, \forall i \neq j$$

(NA): Construcția unui câmp de probabilitate $(\Omega, \mathcal{B}, P)$ și a unei variabilei aleatoare $X : \Omega \to \mathbb{R}$ pornind de la o funcție $F(x)$ ce îndeplinește 3 proprietăți de bază ale unei funcții de repartiție (3) – continuare

Observații:

1. Se verifică că X este v.a. și că P este funcție de probabilitate binedefinită, folosind proprietățile din ipoteză ale funcției $F(x)$.
2. $F(x) = F_X(x)$, adică este funcția de repartiție a variabilei X.
3. Dacă $F(x)$ este o funcție derivabilă și $f(x)$ este densitatea de repartiție a variabilei X și este o funcție integrabilă atunci putem defini funcția de probabilitate prin

$$P(x_1 < X \leq x_2) = \int_{x_1}^{x_2} f(x)dx, \text{pentru } x_1 < x_2.$$

Funcții de o variabilă aleatoare (1)

- Fie $(\Omega, \mathcal{B}, P)$ un câmp de probabilitate.
- Fie $X : \Omega \to \mathbb{R}$ o v.a. care ia valori în $D \subset \mathbb{R}$, $\phi : D \to \mathbb{R}$, ϕ continuă și
$$\{\omega \in \Omega | X(\omega) \in \phi^{-1}(A)\} \in \mathcal{B}, \forall A \subset \phi(D).$$
- Atunci v.a. $Y = \phi(X) : \Omega \to \phi(D) \subset \mathbb{R}$ are repartiția dată de
$$P(\{\omega \in \Omega | Y(\omega) \in A\} = P(\{\omega \in \Omega | X(\omega) \in \phi^{-1}(A)\})) \quad (2)$$
pentru orice interval $A \subset \phi(D)$ și $\phi^{-1}(A) = \{z | \phi(z) \in A\}$.

Funcții de o variabilă aleatoare (2): Exemplu 1

Propoziție

Dacă $X \sim N(m, \sigma^2)$ atunci $Y = e^X$ are densitatea de repartiție:
$$f_Y(y) = \frac{1}{y\sigma\sqrt{2\pi}} e^{-\frac{(\ln y - m)^2}{2\sigma^2}}, y > 0 \quad (3)$$

Demonstrație

$$F_Y(y) = P(\{\omega \in \Omega | Y(\omega) \leq y\}) = P(e^X \leq y) = P(X \leq \ln y) =$$
$$= F_X(\ln y) =$$
$$= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\ln y} e^{-\frac{(t-m)^2}{2\sigma^2}} dt.$$

Definiție

Variabila aleatoare Y având densitatea de probabilitate (3) se numește lognormală.

Funcţii de o variabilă aleatoare (3): Exemplu 2

Propoziţie

Dacă $X \sim U(0,1)$ atunci variabila aleatoare

$$Y = \left(-\frac{1}{a}\ln(1-X)\right)^{1/b}$$

cu $a, b > 0$ are densitatea de probabilitate

$$f(y) = aby^{b-1}e^{-ay^b}, \quad 0 < y < \infty. \tag{4}$$

Definiţie

O variabilă aleatoare cu densitatea de probabilitate (0.4) se numeşte Weibull cu parametrii a şi b şi este notată $W(a, b)$.

Structura generală a unui algoritm de generare/simulare a unei variabile aleatoare

Intrare	$F(x)$: funcţia de repartiţie a variabilei X pe care ne propunem să o simulăm
Pas 1	Se generează valorile de selecţie $u_1, u_2, \ldots, u_n$ asupra v. a. $U_1, U_2, \ldots, U_n$ uniforme pe [0,1]
Pas 2	Se defineşte $X = \phi(U_1, U_2, \ldots, U_n)$ a.î. $F(x) = P(X \leq x), \forall x$
Pas 3	Se obţine valoarea de selecţie dorită $x = \phi(u_1, u_2, \ldots, u_n)$
Ieşire	Valoarea de selecţie, x, a v.a. X

Numere aleatoare

- Majoritatea metodelor de generare a v.a. se bazează pe generarea unor numere aleatoare uniform distribuite pe intervalul (0,1).
- Datorită calculatorului avem posibilitatea de a genera foarte uşor numere aleatoare uniforme.
- Totuşi, trebuie cunoscut faptul că numerele generate de calculator sunt pseudo-aleatoare, deoarece acestea sunt generate cu un algoritm determinist.

Numere aleatoare uniform distribuite

Fie $(\Omega, \mathcal{B}, P)$ un câmp de probabilitate.

Definiţie

(Variabilă aleatoare uniformă)
Spunem că *variabila aleatoare continuă* $U : \Omega \to \mathbb{R}$ este *repartizată uniform* pe intervalul $[a, b]$ şi scriem $U \sim \mathcal{U}(a, b)$ dacă *densitatea sa de repartiţie* este de forma:

$$f_U(x) = \begin{cases} \dfrac{1}{b-a}, & \text{dacă } x \in [a, b] \\ 0, & \text{altfel} \end{cases} \tag{5}$$

Funcţia de repartiţie $F_U(x)$ a v.a. $U \sim \mathcal{U}(a, b)$ este definită prin

$$F_U(x) = \begin{cases} 0, & \text{dacă } x \leq a \\ \dfrac{x-a}{b-a}, & \text{dacă } x \in (a, b) \\ 1, & \text{dacă } x \geq b \end{cases} \tag{6}$$

Observații referitoare la repartiția uniformă și v.a. uniforme

- Probabilitatea ca v.a. uniformă $U \sim \mathcal{U}(a,b)$ să cadă într-un interval $[u_1, u_2], a \leq u_1 < u_2 \leq b$ este proporțională cu lungimea intervalului:

$$P(u_1 < U \leq u_2) = \frac{u_2 - u_1}{b - a}. \tag{7}$$

- Pentru a genera numere aleatoare uniform distribuite pe un interval [a,b], și scriem $X \sim \mathcal{U}(a,b)$, pornind de la numere generate uniform pe intervalul [0,1] se poate folosi transformarea

$$X = (b-a) \cdot U + a, \tag{8}$$

unde $U \sim \mathcal{U}(0,1)$.

Generarea numerelor uniforme în Matlab (1)

- În Matlab există funcția **rand** pentru generarea variabilelor aleatoare uniforme.
- **rand(**n**)**, unde n este un număr natural, returnează o matrice de dimensiune $n \times n$ având ca elemente numere aleatoare uniform distribuite între 0 și 1.
- **rand(**m, n**)**, unde m, n sunt numere naturale, returnează o matrice de dimensiune $m \times n$ având ca elemente numere aleatoare uniform distribuite între 0 și 1.

Generarea numerelor uniforme în Matlab (2)

- O secvență de numere aleatoare generată în Matlab depinde de "sămânța" sau starea generatorului. Starea este resetată la valoarea implicită în momentul pornirii Matlab-ului, astfel aceeași secvență de variabile aleatoare este generată la o nouă pornire a Matlab-ului. Acesta poate fi un avantaj în situațiile în care analistul are nevoie să reproducă rezultatele unei simulări pentru a verifica anumite concluzii.
- **rng(SD)** setează sămânța generatorului pe baza întregului nenegativ *SD*
- **rng('shuffle')** setează sămânța generatorului pe baza momentului curent de timp
- **rng('default')** resetează setările generatorului la valorile inițiale, și anume, se folosește generatorul Mersenne Twister cu sămânța 0.

Generarea numerelor uniforme în Matlab (3)

Exemplu care ilustrează utilizarea funcției **rand**.

% Generam un vector de numere aleatoare pe intervalul (0,1).
x = rand(1,1000);
% Histograma eșantionului generat in x.
[N,X] = hist(x,15);
% x: mulțimea eșantion
% 15 reprezintă numărul de dreptunghiuri ale histogramei
% N: vector conținând numărul de elemente din fiecare dintre dreptunghiuri.
% X: vector conținând centrele dreptunghiurilor
% Folosirea funcției bar pentru reprezentarea grafică a histogramei.
bar(X,N,1,'w')
title('Histograma asociata unei variabile aleatoare uniforme')
xlabel('X')
ylabel('Frecventa absoluta')

Histograma rezultată este prezentată în Figura 1.

Generarea numerelor uniforme în Matlab (4)

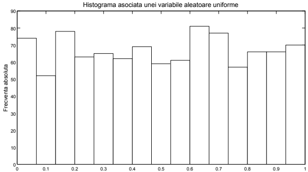


Figura : 1 – Histograma asociată unui eşantion de numere aleatoare uniform distribuite

Metoda inversă (1): Teorema lui Hincin

Propoziţie

Fie X o variabilă aleatoare (v.a.) cu funcţia de repartiţie $F(x)$ inversabilă şi de tip continuu. Atunci variabila aleatoare $Y = F(X)$ este repartizată uniform pe $[0, 1]$.

Teoremă (Teorema lui Hincin)

Fie $U \sim \mathcal{U}(0, 1)$ şi $F(x)$ o funcţie de repartiţie inversabilă, de tip continuu. Atunci variabila aleatoare

$$X = F^{-1}(U) \tag{9}$$

este o variabilă aleatoare continuă cu funcţia de repartiţie $F(x)$.

Demonstraţie: Funcţia de repartiţie a v.a. X este

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x))$$

deoarece F este monoton crescătoare.
Cum $U \sim \mathcal{U}(0, 1)$ rezultă $F_U(u) = u$ pentru $u \in [0, 1]$. Obţinem astfel $P(X \leq x) = F(x)$.

Metoda inversă (2): Descrierea metodei

- este introdusă ca o consecinţă directă a teoremei lui Hincin
- se aplică în cazul în care funcţia de repartiţie se poate inversa uşor
- dacă am putea produce valorile de selecţie $u_1, u_2, \ldots, u_n$ asupra v.a. $U \sim \mathcal{U}(0, 1)$ şi am cunoaşte funcţia de repartiţie F a variabilei X atunci am putea produce valorile de selecţie $x_1, x_2, \ldots, x_n$ asupra lui X cu formula $x_i = F^{-1}(u_i), 1 \leq i \leq n$

Metoda inversă (3): Algoritm pentru simularea unor variabile aleatoare continue

Intrare	$F(x)$: funcţia de repartiţie a variabilei X pe care ne propunem să o simulăm
Pas 1	Se generează o valoare de selecţie u uniformă pe [0,1]
Pas 2	Se determină expresia inversei funcţiei de repartiţie $F^{-1}(u)$
Pas 3	Se obţine valoarea de selecţie dorită $x = F^{-1}(u)$
Ieşire	Valoarea de selecţie, x, a v.a. X

Metoda inversă (4): Variabile aleatoare continue care pot fi simulate folosind metoda inversă

Repartiția	**Densitatea** f	**Inversa** F^{-1}
$Exp(\lambda)$, $\lambda > 0$	$f(x) = \lambda e^{-\lambda x}, x > 0$	$x = -\frac{1}{\lambda}\ln(u)$
$Weib(0,1,\nu)$, $\nu > 0$	$f(x) = \nu x^{\nu-1}e^{-x^{\nu}}$	$x = (-\ln(u))^{1/\nu}$
Cauchy	$f(x) = \frac{1}{\pi}\frac{1}{1+x^2}, x \in \mathbb{R}$	$x = \tan \pi(u - 1/2)$
Arcsin	$f(x) = \frac{1}{\pi}\frac{1}{\sqrt{1-x^2}}, x \in [-1,1]$	$x = \sin \pi(u - 1/2)$

Metoda inversă (5): Exemplu – simularea unei variabile aleatoare având repartiția $Exp(\lambda), \lambda > 0$

- Se determină funcția de repartiție a unei v.a. $X \sim Exp(\lambda)$:

$$F : \mathbb{R} \to [0,1], F(x) = \int_{-\infty}^{x} f(t)dt = \\ = \begin{cases} 1 - e^{-\lambda x}, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{10}$$

- Se determină funcția inversă a funcției de repartiție $F\big|_{(0,\infty)}$: pentru $u \in (0,1)$, determinăm $x > 0$ a.î. $F(x) = u$. Obținem inversa funcției $F\big|_{(0,\infty)}$ definită prin

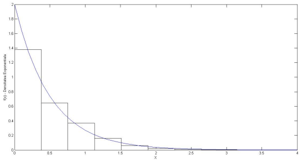
$$F^{-1} : (0,1) \to (0,\infty), F^{-1}(u) = -\frac{\ln(1-u)}{\lambda} \tag{11}$$

- Din Teorema lui Hincin rezultă că

$$X = -ln(U)/\lambda, U \sim \mathcal{U}(0,1) \tag{12}$$

este o v.a. repartizată $Exp(\lambda)$.

Metoda inversă (6): Histograma asociată variabilei aleatoare exponențiale



Metoda inversă (7): Simularea unei variabile aleatoare discrete (1)

- Fie v.a. discretă X definită prin repartiția

$$X : \begin{pmatrix} x_1 & x_2 & \dots & x_m \\ p_1 & p_2 & \dots & p_m \end{pmatrix}, \sum_{i=1}^{m} p_i = 1, x_1 < x_2 < \dots < x_m. \tag{13}$$

- Funcția de repartiție a v.a. X este dată de

$$F(x) = \begin{cases} 0 & \text{dacă} & x < x_1 \\ p_1 & \text{dacă} & x_1 \leq x < x_2 \\ p_1 + p_2 & \text{dacă} & x_2 \leq x < x_3 \\ \dots & \dots & \dots \\ p_1 + p_2 + \dots + p_k & \text{dacă} & x_k \leq x < x_{k+1} \\ \dots & \dots & \dots \\ 1 & \text{dacă} & x \geq x_m \end{cases} \tag{14}$$

Metoda inversă (8): Simularea unei variabile aleatoare discrete (2)

- Fie $(\Omega, \mathcal{B}, P)$ un câmp de probabilitate și
$$X, U : \Omega \to \mathbb{R}, U \sim \mathcal{U}(0,1).$$
- Definim v.a. $X = Y \circ U$, $X(\omega) = Y(U(\omega)), \forall \omega \in \Omega$, unde funcția $Y : [0,1] \to \{x_1, x_2, \ldots, x_m\}$ este definită prin
$$Y(u) = \begin{cases} x_1, & 0 < u \le F(x_1) \\ x_2, & F(x_1) < u \le F(x_2) \\ \ldots & \ldots \\ x_m, & F(x_{m-1}) < u \le 1 \end{cases} \quad (15)$$
- Funcția de probabilitate a v.a. X, $f(x_i) \overset{def}{=} P(X = x_i)$, devine
$$\begin{aligned} P(X = x_i) &= P(\{\omega \in \Omega \mid Y(U(\omega)) = x_i\}) = \\ &= P(\{\omega \in \Omega \mid F(x_{i-1}) < U(\omega) \le F(x_i)\}) = \\ &= F(x_i) - F(x_{i-1}) = p_i. \end{aligned} \quad (16)$$

Metoda inversă (9): Algoritm pentru simularea unor variabile aleatoare discrete

- Regula de generare a unei valori de selecție asupra v.a. X, pornind de la valoarea de selecție u a v.a. $U \sim \mathcal{U}(0,1)$:
$$X = x_i \text{ dacă } F(x_{i-1}) < u \le F(x_i) \text{ și } x_0 < x_1 \quad (17)$$
- Algoritmul pentru simularea v.a. X:

Intrare	Repartiția variabilei X $P(X = x_i) = p_i, \sum_{i=1}^{m} p_i = 1, x_1 < x_2 < \ldots < x_m.$
Pas 1	Se generează o valoare de selecție u uniformă pe [0,1]
Pas 2	Dacă $u \le p_1$ atunci $x = x_1$ Altfel dacă $u \le p_1 + p_2$ atunci $x = x_2$ Altfel dacă $u \le p_1 + p_2 + p_3$ atunci $x = x_3$ ... Altfel dacă $u \le p_1 + p_2 + \ldots + p_m$ atunci $x = x_m$
Ieșire	Valoarea de selecție, x, a v.a. X

Metoda inversă (10): Exemplu simularea unei v.a. discrete

- Vrem să generăm o v.a. discretă X cu repartiția
$$X : \begin{pmatrix} 0 & 1 & 2 \\ 0.3 & 0.2 & 0.5 \end{pmatrix} \quad (18)$$
- Funcția de repartiție este dată de
$$F(x) = \begin{cases} 0 & \text{dacă} \quad x < 0 \\ 0.3 & \text{dacă} \quad 0 \le x < 1 \\ 0.5 & \text{dacă} \quad 1 \le x < 2 \\ 1 & \text{dacă} \quad x \ge 2 \end{cases} \quad (19)$$
- Se generează valori de selecție asupra v.a. X conform regulilor
$$X = \begin{cases} 0 & U \le 0.3 \\ 1 & 0.3 < U \le 0.5 \\ 2 & 0.5 < U \le 1 \end{cases} \quad (20)$$
- Dacă v.a. $u = 0.78$ atunci obținem valoarea de selecție $x = 2$.

Bibliografie I

- M. Craiu (1998), *Statistică matematică: teorie și probleme*, Editura Matrix Rom, București
- W. L. Martinez, A. R. Martinez (2002), *Computational Statistics Handbook with MATLAB*, Chapman & Hall/CRC, Boca Raton London New York Washington, D.C.
- J. B. Thomas, *Introduction to probability*
- I. Văduva (2004), *Modele de simulare: note de curs*, Editura Universității din București, București

Universitatea din București
Facultatea de Matematică și Informatică

CURS nr. 4
METODE MODERNE DE CALCUL ȘI SIMULARE

Validarea generatorilor

Metoda amestecării – compunerii

Lect. dr. Bianca Mogoș

Conținut

Validarea generatorilor/algoritmilor de simulare a unor variabile aleatoare

1. Histograma – o validare empirică a algoritmului de simulare
2. Test bazat pe momentele de selecție
3. Testul X^2

Metoda amestecării – compunerii

1. Definiții: amestecare discretă și amestecare continuă
2. Descrierea metodei
3. Algoritm general de amestecare – compunere
4. Fundamentarea matematică a metodei
5. Exemple: Simularea v.a. Laplace și Lomax

Validarea generatorilor/algoritmilor de simulare a unor variabile aleatoare (1)

- *Algoritm de simulare*: definirea unei v.a. X având o funcție de repartiție dată $F(x) = P(X \leq x), \forall x \in \mathbb{R}$ și observarea variabilei X
- *Validarea algoritmilor de simulare* înseamnă:
  - verificarea corectitudinii formale a algoritmilor: se arată că v.a. X construită în algoritm are funcția de repartiție $F(x)$
  - analiza valorilor de selecție asupra v.a. X returnate de algoritm: pe baza unei mulțimi de selecție $X_1, X_2, \ldots, X_n$, se verifică ipoteza statistică $H_0 : X \hookrightarrow F(x)$.

Validarea generatorilor (2): Histograma – o validare empirică a algoritmului de simulare (1)

- Se verifică intuitiv dacă *repartiția empirică (de selecție)* este asemănătoare cu cea *teoretică*
- *Histograma* asociată mulțimii de valori de selecție $x_1, \ldots, x_n$ asupra variabilei aleatoare X având funcția de repartiție $F(x)$ și densitatea $f(x)$:
  - se determină $m = \min\{x_1, x_2, \ldots, x_n\}$ și $M = \max\{x_1, x_2, \ldots, x_n\}$
  - se alege k numărul de intervale/dreptunghiuri ale histogramei
  - se împarte intervalul $[m, M]$ în k intervale egale $I_i = (a_{i-1}, a_i], 2 \leq i \leq k, I_1 = [a_0, a_1]$, $a_0 = m, a_k = M$
  - se determină frecvențele relative $r_i = \frac{f_i}{n}$, unde f_i : numărul de valori de selecție ce cad în intervalul I_i, $1 \leq i \leq k$
  - se reprezintă grafic: se iau pe abscisă intervalele I_i și se construiesc dreptunghiurile având ca bază aceste intervale și ca înălțimi $h_i = r_i$.
- *Înălțimile h_i ale dreptunghiurilor se scalează a.î. $h_i \approx f(x), x \in I_i$. Definim $h_i = \frac{f_i}{nl} \approx f(x), l = a_i - a_{i-1}$.*

Validarea generatorilor (3): Histograma – o validare empirică a algoritmului de simulare (2)

- Din *Teorema lui Bernoulli – forma slabă* rezultă că pentru orice $\epsilon > 0$ avem

$$\lim_{n\to\infty} P\left(\left|\frac{f_i}{n} - p_i\right| \leq \epsilon\right) = 1, \quad (1)$$

unde $p_i = P(X \in I_i), I_i = (a_{i-1}, a_i]$

- Numărul k de dreptunghiuri ale histogramei se consideră a.î. să se minimizeze *"media pătratelor erorilor (MSE)"* estimatorului $\hat{f}(x)$ (definit în orice punct x) al densității de repartiție $f(x)$, definită prin

$$MSE(\hat{f}(x)) = E\left[\left(\hat{f}(x) - f(x)\right)^2\right]. \quad (2)$$

Astfel, avem *Regula Sturges*

$$k = [1 + \log_2 n]. \quad (3)$$

Validarea generatorilor (4): Test bazat pe momentele de selecție

- Se determină *momentele teoretice* ale v.a. X:

$$\mu = E[X] \text{ și } \sigma^2 = Var(X) \quad (4)$$

- Pe baza mulțimii de valori de selecție $\{x_1, x_2, \ldots, x_n\}$ se calculează *momentele de selecție*:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ și } s^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \overline{X}^2 \quad (5)$$

- Ca o consecință a *Legii numerelor mari*, putem considera că generatorul este bun dacă pentru n suficient de mare ($n > 1000$)

$$\overline{X} \approx \mu \text{ și } s^2 \approx \sigma^2 \quad (6)$$

Validarea generatorilor (5): Testul X^2 (1)

- Considerăm *testul de concordanță X^2* pentru verificarea ipotezei *$H_0 : X \hookrightarrow F(x)$*.
- Definim *variabila aleatoare X^2*:

$$X^2 = \sum_{i=1}^{k} \frac{(f_i - np_i)^2}{np_i} \quad (7)$$

unde $p_1 = F(a_1), p_i = F(a_i) - F(a_{i-1}), 2 \leq i \leq k-1, p_k = 1 - F(a_{k-1})$.

- *Observații*:
  - $f_i : \Omega \to \{0, 1, \ldots, n\}$ este o v.a. $Bin(n, p_i)$
  - Mulțimea tuturor valorilor posibile ale lui X^2 se obține făcând ca $f_1, f_2, \ldots, f_k$ să parcurgă toți întregii nenegativi a.î. $\sum_{i=1}^{k} f_i = n$
  - Pentru $n \to \infty$, X^2 este repartizată χ^2_{k-1} (hi pătrat cu $k-1$ grade de libertate).

Validarea generatorilor (6): Testul X^2 (2)

- Fie α *eroarea de tip I*: probabilitatea de a respinge ipoteza nulă H_0 când este adevărată. Valorile clasice pentru α sunt 0.01, 0.05, 0.1.
- Se determină *α – cuantila superioară*, notată $\chi^2_{k-1,\alpha}$ astfel încât

$$P\left(X^2 \leq \chi^2_{k-1,\alpha}\right) = 1 - \alpha \quad (8)$$

- Ipoteza *H_0 se acceptă* dacă în urma experimentului aleator s-a obținut evenimentul $\omega \in \Omega$ a.î.:

$$X^2(\omega) \leq \chi^2_{k-1,\alpha}, \quad (9)$$

în caz contrar se respinge.

Metoda compunerii (1): Definiții amestecare discretă și amestecare continuă

- Spunem că funcția de repartiție $F(x)$ este o *amestecare (compunere sau mixtură) discretă* a mulțimii de funcții de repartiție $\{F_j(x)\}_{1\leq j\leq m}$ cu repartiția discretă J dacă

$$F(x)=\sum_{j=1}^{m}p_jF_j(x) \text{ și } J:\begin{pmatrix} 1 & 2 & \dots & m \\ p_1 & p_2 & \dots & p_m\end{pmatrix}, \sum_{j=1}^{m}p_j=1 \quad (10)$$

- Spunem că funcția de repartiție $F(x)$ este o *amestecare continuă* a familiei de funcții de repartiție $\{G(x,Y)\}_{Y\in\mathbb{R}}$, cu funcția de repartiție continuă $H(y)$ a lui Y dacă este de forma

$$F(x)=\int_{\mathbb{R}}G(x,y)dH(y). \quad (11)$$

Metoda compunerii (2): Descrierea metodei

- Fie X o v.a. cu funcția de repartiție $F(x)$ definită în (10) și $X_j, 1\leq j\leq m$ v.a. având funcțiile de repartiție $F_j(x)$. Atunci

$$X=X_j \text{ cu probabilitatea } p_j \quad (12)$$

- Fie X o v.a. cu funcția de repartiție $F(x)$ definită în (11) și $Z_Y, Y\in\mathbb{R}$ v.a. având funcțiile de repartiție $G(x,Y)$, unde Y are funcția de repartiție $H(y)$. Atunci

$$X=Z_y \text{ unde } y \text{ este generat cu funcția de repartiție } h(y) \quad (13)$$

- În definiția amestecării se pot considera în loc de funcții de repartiție, densități de repartiție și formulele devin

$$f(x)=\sum_{i=1}^{m}p_if_i(x), \text{ respectiv } f(x)=\int_{\mathbb{R}}g(x,y)h(y)dy \quad (14)$$

Metoda compunerii (3): Algoritm pentru simularea unei variabile aleatoare bazat pe noțiunea de amestecare discretă

Intrare	Se cunosc algoritmi pentru simularea v.a. X_j având funcțiile de repartiție $F_j(x)$ Repartiția v.a. discrete J: $P(J=j)=p_j, \sum_{j=1}^{m}p_j=1$
Pas 1	Se generează un indice j având repartiția J
Pas 2	Se generează x_j cu funcția de repartiție $F_j(x)$
Pas 3	Se definește $x=x_j$
Ieșire	Valoarea de selecție, x, a v.a. X având funcția de repartiție $F(x)$

Metoda compunerii (4): Algoritm pentru simularea unei variabile aleatoare bazat pe noțiunea de amestecare continuă

Intrare	Se cunosc algoritmi pentru simularea v.a Z_Y având funcțiile de repartiție în familia $\{G(x,Y)\}_{Y\in\mathbb{R}}$ Se cunoaște un algoritm pentru simularea v.a. Y cu funcția de repartiție $H(y)$
Pas 1	Se generează y cu funcția de repartiție $H(y)$
Pas 2	Se generează z_y cu funcția de repartiție $G(x,y)$
Pas 3	Se definește $x=z_y$
Ieșire	Valoarea de selecție, x, a v.a. X având funcția de repartiție $F(x)$

Metoda compunerii (5): Exemplu amestecare discretă

- Fie X variabila având repartiția Laplace a cărei densitate de repartiție este

$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|}, x \in \mathbb{R}, \lambda > 0 \tag{15}$$

- Observăm că

$$f(x) = p_1 f_1(x) + p_2 f_2(x), p_1 = p_2 = \frac{1}{2}$$

unde

$$f_1(x) = \begin{cases} 0 & \text{dacă } x \leq 0 \\ \lambda e^{-\lambda x} & \text{dacă } x > 0 \end{cases}, f_2(x) = \begin{cases} \lambda e^{\lambda x} & \text{dacă } x \leq 0 \\ 0 & \text{dacă } x > 0 \end{cases}$$

Metoda compunerii (6): Algoritmul pentru simularea v.a. X având repartiția Laplace

Intrare	V.a. $X_1 \sim Exp(\lambda)$ cu densitatea de repartiție $f_1(x)$ și V.a. $X_2 = -X_1$ cu densitatea de repartiție $f_2(x)$
Pas 1	Se generează val. de selecție u a v.a. $U \sim \mathcal{U}(0,1)$
Pas 2	Dacă $u \leq 0.5$ atunci mergi la Pas 3; altfel mergi la Pas 4
Pas 3	s = 1; mergi la Pas 5
Pas 4	s = -1
Pas 5	Se generează val. de selecție x_1 a v.a. $X_1 \sim Exp(\lambda)$
Pas 6	Se definește $x = sx_1$
Ieșire	Valoarea de selecție, x, a v.a. X.

Metoda compunerii (7): Exemplu amestecare continuă

- Fie variabila $X, X > 0$, durata de funcționare a unui aparat, repartizată Exponențial$(\eta\lambda)$, unde
  - λ este un parametru determinat de producător, iar
  - η este un parametru aleator care indică influența mediului în care este studiat aparatul.
- Repartiția de probabilitate a lui η este de tip $Gamma(0, b, a)$:

$$h(\eta) = \begin{cases} 0, & \text{dacă } x < 0 \\ \frac{b^a}{\Gamma(a)} \eta^{a-1} e^{-b\eta}, & \text{altfel} \end{cases} \tag{16}$$

- Densitatea de repartiție a lui X, pentru $x \geq 0$ va fi de forma

$$\begin{aligned} f(x) &= \int_0^\infty \eta \lambda e^{-\lambda \eta x} \frac{b^a}{\Gamma(a)} \eta^{a-1} e^{-b\eta} d\eta = \\ &= \frac{a\theta}{(\theta x + 1)^{a+1}}, \text{ unde } \theta = \frac{\lambda}{b} \end{aligned} \tag{17}$$

- Repartiția având densitatea (17) se numește *repartiție Lomax*.

Bibliografie I

- M. Craiu (1998), *Statistică matematică: teorie și probleme*, Editura Matrix Rom, București
- W. L. Martinez, A. R. Martinez (2002), *Computational Statistics Handbook with MATLAB*, Chapman & Hall/CRC, Boca Raton London New York Washington, D.C.
- I. Văduva (2004), *Modele de simulare: note de curs*, Editura Universității din București, București

Universitatea din București
Facultatea de Matematică și Informatică

CURSUL nr. 7
METODE MODERNE DE CALCUL ȘI SIMULARE

Metoda acceptării – respingerii

Lect. dr. Bianca Mogoș

Conținut

Metoda acceptării – respingerii: Descrierea metodei

Presupunem că se cunosc următoarele elemente:

- Se cunoaște un procedeu de simulare a unei v.a. N care ia valori naturale pozitive
- Se cunosc metode pentru simularea unor v.a. $S_i \in S, i \geq 1$, unde S este o familie de v.a. dată
- Se cunoaște un predicat $\mathcal{P}(S_1, S_2, \dots, S_n)$ care se poate calcula $\forall S_i, n$ (acest predicat sau condiție trebuie evaluat(ă) fără calcule de mare complexitate)
- Se cunoaște funcția ψ astfel încât $X = \psi(S_1, S_2, \dots, S_n)$, $\mathcal{P}(S_1, S_2, \dots, S_n) = true$.

Observație: pentru $(\Omega, \mathcal{B}, P)$ un câmp de probabilitate astfel încât

$$\Omega_1 \overset{def}{=} \{\omega \in \Omega \mid \mathcal{P}(S_1, S_2, \dots, S_n) = true\} \in \mathcal{B} \qquad (1)$$

avem

$$X : \Omega_1 \to \mathbb{R}, X = \psi(S_1, S_2, \dots, S_n). \qquad (2)$$

Algoritm general de respingere

Intrare	V.a. N, v.a. $S_1, S_2, \dots, S_n \in S$, predicatul $\mathcal{P}(S_1, S_2, \dots, S_n)$, funcția $\psi(S_1, S_2, \dots, S_n)$
	Repetă Pas 1 – Pas 3 până când $\mathcal{P}(S_1, S_2, \dots, S_n) = true$
Pas 1	Se simulează o valoare n a lui N
Pas 2	Se simulează valorile de selecție $S_1, S_2, \dots, S_n \in S$
Pas 3	Se verifică dacă $\mathcal{P}(S_1, S_2, \dots, S_n) = true$
Pas 4	Se consideră $x = \psi(S_1, S_2, \dots, S_n)$.
Ieșire	Valoarea de selecție, x, a v.a. X

Observații

- Dacă $\mathcal{P}(S_1, S_2, \ldots, S_n) = \textit{false}$ atunci mulțimea de variabile aleatoare $\{S_1, S_2, \ldots, S_n\} \subseteq S$ *se respinge*, de unde provine și numele de *metodă de respingere*.
- Dacă $p_a = P(\mathcal{P}(S_1, S_2, \ldots, S_n) = \textit{true})$, numită *probabilitate de acceptare*, este mare (apropiată de 1) atunci algoritmul este rapid, altfel este lent.

Teorema înfășurătoarei (1)

Teoremă (Teorema înfășurătoarei)

Fie dată o variabilă aleatoare X care are densitatea de repartiție $f(x), x \in \mathbb{R}$ pe care dorim să o simulăm. Fie Y o altă variabilă aleatoare ce știm să o simulăm și a cărei densitate de repartiție este $h(x)$ astfel încât f, h au același suport S (adică iau valori diferite de zero pe aceeași mulțime $S \subseteq \mathbb{R}$). Presupunem că există o constantă $\alpha, 0 < \alpha < \infty$, astfel încât $f(x) \leq \alpha h(x), \forall x \in S$. În aceste condiții, dacă U este o variabilă uniformă pe [0,1] independentă de Y atunci densitatea de repartiție a variabilei Y condiționată de $0 \leq U \leq f(Y)/(\alpha h(Y))$ este f.

Observații (2)

- Fie $(\Omega, \mathcal{B}, P)$ un câmp de probabilitate
- Definim

$$\Omega_1 = \{\omega \in \Omega \mid 0 \leq U(\omega) \leq f(Y(\omega))/(\alpha h(Y(\omega)))\} \quad (3)$$

  și

$$X : \Omega_1 \to \mathbb{R}, X = Y|\Omega_1 \quad (4)$$

- $f(x) = F'(x), F(x) = P(X \leq x) \overset{def}{=} P(Y \leq x|\Omega_1)$
- $(\exists)\epsilon_0$ a.î. $\frac{p_A}{p_{\Omega_1}} - \epsilon_0 < \frac{k_A}{k_{\Omega_1}} < \frac{p_A}{p_{\Omega_1}} + \epsilon_0$, unde
  - $p_A = P(Y \leq x \cap \Omega_1), p_{\Omega_1} = P(\Omega_1)$
  - k_A, k_{Ω_1} reprezintă numărul de apariții ale evenimentelor A, respectiv Ω_1 pentru un număr suficient de mare de repetări ale experimentului aleator.

Demonstrația teoremei înfășurătoarei (3)

- Trebuie demonstrat că

$$P(Y \leq x|0 \leq U \leq f(Y)/(\alpha h(Y))) = F(x) = \int_{-\infty}^{x} f(v)dv \quad (5)$$

- Considerăm evenimentele

$$A = \{Y \leq x\} \text{ și } \Omega_1 = \{0 \leq U \leq f(Y)/(\alpha h(Y))\} \quad (6)$$

- Din (5) și (6) rezultă că trebuie să calculăm $P(A|\Omega_1)$ având definiția

$$P(A|\Omega_1) = \frac{P(A \cap \Omega_1)}{P(\Omega_1)} \quad (7)$$

Demonstrația teoremei înfășurătoarei (4)

- Calculăm $P(\Omega_1)$:

$$P(\Omega_1) = \int_{-\infty}^{+\infty} \left[\int_0^{f(v)/(\alpha h(v))} du \right] h(v) dv = \\ = \int_{-\infty}^{+\infty} \frac{f(v)}{\alpha h(v)} h(v) dv = \frac{1}{\alpha}, \alpha > 1 \tag{8}$$

- Astfel, $P(A|\Omega_1)$ devine:

$$P(A|\Omega_1) = \alpha \int_{-\infty}^{x} \left[\int_0^{f(v)/(\alpha h(v))} du \right] h(v) dv = \\ = \alpha \int_{-\infty}^{x} \frac{f(v)}{\alpha h(v)} h(v) dv = \int_{-\infty}^{x} f(v) dv \tag{9}$$

Algoritm pentru simularea unei v.a. bazat pe teorema înfășurătoarei (5)

Intrare	Știm să generăm v.a. Y având densitatea de repartiție $h(y)$
Pas 1	Se caută o constantă α a. î. $f(x) \leq \alpha h(x), \forall x \in S$ Repetă Pas 2 – Pas 4 până când $u \leq f(y)/\alpha h(y)$, cu u și y valori de selecție obținute în pașii indicați
Pas 2	Se simulează o valoare de selecție u a v.a. $U \sim \mathcal{U}(0,1)$
Pas 3	Se simulează o valoare de selecție y a v.a. Y cu dens. $h(y)$
Pas 4	Se verifică dacă $u \leq f(y)/\alpha h(y)$
Pas 5	Se consideră $x = y$.
Ieșire	Valoarea de selecție, x, a v.a. X având densitatea de repartiție $f(x)$

Simularea variabilei aleatoare normale (6)

- Spunem că $X \sim \mathcal{N}(\mu, \sigma)$ dacă X are densitatea de repartiție

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{10}$$

- Se va prezenta un algoritm de simulare a variabilei aleatoare $Z \sim \mathcal{N}(0,1)$ bazat pe o metodă de compunere - respingere. Se deduce ușor un algoritm de simulare a v.a. $X \sim \mathcal{N}(\mu, \sigma)$ folosind relația:

  "Dacă $Z \sim \mathcal{N}(0,1)$ atunci $X = \sigma Z + \mu$ este o v.a. $X \sim \mathcal{N}(\mu, \sigma)$"

Simularea variabilei aleatoare normale (7)

- Considerăm v.a. X_1 și $X_2 = -X_1$ având densitățile de repartiție

$$f_1(x) = \begin{cases} 0 & \text{dacă } x \leq 0 \\ \sqrt{\frac{2}{\pi}} e^{\frac{-x^2}{2}} & \text{dacă } x > 0 \end{cases} \tag{11}$$

  și respectiv,

$$f_2(x) = \begin{cases} \sqrt{\frac{2}{\pi}} e^{\frac{-x^2}{2}} & \text{dacă } x \leq 0 \\ 0 & \text{dacă } x > 0 \end{cases} \tag{12}$$

- Densitatea de repartiție a v.a. $Z \sim \mathcal{N}(0,1)$ se poate scrie sub forma

$$f(x) = \frac{1}{2} f_1(x) + \frac{1}{2} f_2(x), \tag{13}$$

  adică este o compunere discretă a densităților $f_1(x)$ și $f_2(x)$.

Simularea variabilei aleatoare X_1 (8)

Aplicăm teorema înfășurătoarei pentru a genera v.a. X_1 cu densitatea de repartiție $f_1(x)$:

- se consideră ca înfășurătoare densitatea de repartiție $h(x)$ a v.a. $Y \sim Exp(1)$
- raportul $r(x) = \frac{f_1(x)}{h(x)} = \sqrt{\frac{2}{\pi}} e^{\frac{-x^2}{2}+x}$
- ecuația $r'(x) = 0$ are soluția $x_0 = 1$ care este punct de maxim pentru funcția $r(x)$. Rezultă, de aici,

$$r(x) \leq r(x_0) = \sqrt{\frac{2e}{\pi}}. \tag{14}$$

- considerăm constanta $\alpha = \sqrt{\frac{2e}{\pi}}$. De asemenea, se observă că densitățile $f_1(x)$ și $h(x)$ au același suport $S = (0, \infty)$.

Algoritmul pentru simularea variabilei aleatoare normale $Z \sim N(0,1)$ (9)

Intrare	Pp. că știm să generăm v.a. $Y \sim Exp(1)$.
	Repetă Pas 1 – Pas 3 până când $u \leq e^{-\frac{y^2}{2}+y-0.5}$, cu u și y valori de selecție generate în pașii indicați.
Pas 1	Se simulează o valoare de selecție u a v.a. $U \sim \mathcal{U}([0,1])$
Pas 2	Se simulează o valoare de selecție y a v.a. $Y \sim Exp(1)$
Pas 3	Se verifică dacă $u \leq e^{-\frac{y^2}{2}+y-0.5}$
Pas 4	Se generează u_1 val. de selecție a v.a. $U_1 \sim \mathcal{U}([0,1])$.
Pas 5	Dacă $u_1 \leq 0.5$ atunci mergi la Pas 6; altfel mergi la Pas 7
Pas 6	Se consideră $z = y$. Mergi la Ieșire.
Pas 7	Se consideră $z = -y$.
Ieșire	Valoarea de selecție, z, a v.a. $Z \sim N(0,1)$

Teorema a doua de respingere (1)

Teoremă

Fie X o v.a. având funcția de repartiție de forma

$$F(x) = c \int_{-\infty}^{x} Q(\phi(y)) dR(y) \tag{15}$$

unde $Q(z)$ este funcția de repartiție a unei v.a. $Z, Z \in [0, M]$, $\phi(y)$ este o funcție ce ia valori în intervalul $[0, M]$ (unde M poate lua și valoarea ∞), iar $R(y)$ este funcția de repartiție a unei v.a. $Y, Y \in \mathbb{R}$, iar variabilele Z, Y sunt independente stochastic. Atunci funcția de repartiție a variabilei Y condiționată de $Z \leq \phi(Y)$ este $F(x)$.

Teorema a doua de respingere - observații (2)

- Probabilitatea de acceptare este $p_a = P(Z \leq \phi(Y)) = \frac{1}{c}$, unde c este o constantă de normare cu formula

$$c = \left[\int_{-\infty}^{+\infty} Q(\phi(x)) dR(x) \right]^{-1} \tag{16}$$

- Condiția (15) se poate scrie în termeni de densități de repartiție:

$$f(x) = cQ(\phi(x))r(x), r(x) = R'(x). \tag{17}$$

- O formă duală a teoremei se obține dacă $F(x)$ este de forma

$$F(x) = c \int_{-\infty}^{x} (1 - Q(\phi(y))) dR(y) \tag{18}$$

pentru $c = \left[\int_{-\infty}^{+\infty} (1 - Q(\phi(x))) dR(x) \right]^{-1}$; predicatul devine $\{Z \geq \phi(Y)\}$.

Teorema a doua de respingere - exemplu (3)

- Fie X o v.a. având densitatea de repartiție
$$f(x) = c\mu(1 - e^{-\lambda x})e^{-\mu x}, x \geq 0 \tag{19}$$
- Variabila X se poate simula folosind un algoritm de respingere, pentru
$$\phi(x) = x, Q(z) = 1 - e^{-\lambda z} \text{ și } r(x) = \mu e^{-\mu x} \tag{20}$$
- Se obține $c = \left[\int_0^\infty \mu(1 - e^{-\lambda x})e^{-\mu x}dx\right]^{-1} = \left[\frac{\lambda}{\lambda + \mu}\right]^{-1}$

Teorema șirului descendent (1)

Presupunem date variabilele aleatoare $Z_i, i \geq 1$ cu funcțiile de repartiție $G(x)$ și Z_0 cu funcția de repatiție $G_0(x)$, independente stochastic. Atunci sunt adevărate afirmațiile:

1. Dacă x este fixat și numărul k este fixat atunci
$$P(x \geq Z_1 \geq Z_2 \geq \ldots \geq Z_{k-1} < Z_k) = \frac{[G(x)]^{k-1}}{(k-1)!} - \frac{[G(x)]^k}{k!}. \tag{21}$$

2. Dacă x este fixat și K este indicele aleator la care se “rupe” subșirul descendent (ca la punctul 1.) atunci
$$P(K = \text{nr. impar}) = e^{-G(x)}. \tag{22}$$

Teorema șirului descendent (2)

3. Dacă subșirul descendent este $Z_0 \geq Z_1 \geq \ldots \geq Z_{K-1} < Z_K$ atunci
$$F(x) = P(Z_0 \leq x | K = \text{nr. impar}) = \frac{1}{p_a}\int_{-\infty}^{x} e^{-G(t)}dG_0(t), \tag{23}$$
unde p_a este constanta de normare dată prin
$$p_a = \left[\int_{-\infty}^{\infty} e^{-G(t)}dG_0(t)\right]. \tag{24}$$

Algoritm de simulare a unei v.a. bazat pe teorema șirului descendent (3)

Intrare	Pp. că știm să generăm v.a. $Z_i \hookrightarrow G(x), i \geq 1$ și $Z_0 \hookrightarrow G_0(x)$
	Repetă Pas 1 – Pas 2 până când K = nr. impar
Pas 1	Se generează val. de selecție z_0 a v.a. $Z_0 \hookrightarrow G_0(x)$ și z_1 a v.a. $Z_1 \hookrightarrow G(x)$. Se ia $z* = z_0$, $K = 1$;
Pas 2	Repetă Pas 3 - Pas 4 cât timp $z_0 \geq z_1$
Pas 3	$K = K + 1$; $z_0 = z_1$
Pas 4	Se generează o val. de selecție z_1 a v.a. $Z_1 \hookrightarrow G(x)$
Pas 5	Se consideră $x = z*$
Ieșire	Valoarea de selecție, x, a v.a. $X \hookrightarrow F(x)$

Teorema șirului descendent - performanța algoritmului (4)

- Cu cât p_a este mai mare cu atât mai repede va fi acceptată o val. de selecție a v.a. Z_0 (când $K =$ nr. impar)
- Performața algoritmului depinde de numărul de valori Z_i generate pentru a obține un Z_0 acceptat
- Se arată că valoarea medie a numărului de val. de selecție $Z_i, i \geq 1$ necesare până la acceptarea unui Z_0, notat N^* este

$$E[N^*] = \frac{1}{p_a}\left(1 + \int_{-\infty}^{+\infty} e^{G(x)} dG_0(x)\right) \tag{25}$$

Teorema șirului descendent - exemplu (5)

- Considerăm în Teorema șirului descendent, variabilele $Z_i, i \geq 0$ repartizate $\mathcal{U}([0,1])$ și independente stochastic.
- Avem

$$P(Z_0 \leq x | K = \text{nr. impar}) = \frac{1}{p_a}\int_0^x e^{-x} dx; \text{ unde}$$
$$p_a = \int_0^1 e^{-x} dx = 1 - e^{-1} \tag{26}$$

- Astfel, Z_0 acceptat are funcția de repartiție

$$F(x) = \begin{cases} 0, & \text{dacă } x < 0 \\ \frac{1-e^{-x}}{1-e^{-1}}, & \text{dacă } 0 \leq x \leq 1 \\ 1, & \text{dacă } x > 1 \end{cases} \tag{27}$$

adică repartiția exponențială de parametru 1, trunchiată pe intervalul [0,1].

Bibliografie I

I. Văduva (2004), *Modele de simulare: note de curs*, Editura Universității din București, București

Universitatea din București
Facultatea de Matematică și Informatică

CURSUL nr. 8
METODE MODERNE DE CALCUL ȘI SIMULARE

Modele de simulare

Lect. dr. Bianca Mogoș

Conținut

1.1 Noțiuni de bază – Definiție model (1)

- Oamenii de știință și inginerii încearcă *să înțeleagă, să dezvolte și să optimizeze sisteme*.
- O *problemă* care apare în analiza unui sistem este *complexitatea* acestuia.
- Folosirea unor *descrieri simplificate* ale sistemelor – *modelele* – reprezintă o *soluție* care permite abordarea unor sisteme complexe.
- *Definiție:* Pentru un observator B, un obiect A^* este un *model* al obiectului A dacă B poate folosi A^* pentru a răspunde întrebărilor care îl interesează referitoare la A.
- *Modelul cel mai bun* este *modelul cel mai simplu* care își îndeplinește obiectivul de a ne ajuta să înțelegem un sistem complex și de a rezolva anumite probleme.

1.2 Introducerea noțiunii de simulare

- Cuvântul *simulare* este de origine latină și înseamnă *capacitatea de a reproduce ceva*.
- Termenul de simulare a fost introdus de *John von Neumann* la începutul anilor '40.
- *Modelul de simulare* este o extindere a unui model matematic prin descrierea relațiilor dintre componentele modelului.
- *Simularea* reprezintă aplicarea unui model (numit model de simulare) pentru a descrie comportamentul și evoluția unui sistem real în timp.

1.3 Avantajele și dezavantajele utilizării simulării (1)

Avantajele obținute datorită realizării unui studiu de simulare

- Simularea furnizează o modalitate *necostisitoare* și *neriscantă* de realizare a unor teste referitoare la funcționalitatea unui sistem.
- Simularea permite *prezicerea* comportamentului unui sistem pentru diferite ipoteze/variabile de intrare ale sistemului și *reducerea riscului* de a lua o decizie greșită.
- Un model de simulare permite *analiza variabilității proceselor* dintr-un sistem, ne poate ajuta să înțelegem cum interacționează între ele mai multe componente ale sistemului și cum poate afecta interacțiunea dintre ele performanța sistemului.
- Simularea dă *posibilitatea de explorare* a mai multor căi alternative.
- Dezvoltarea unei aplicații de simulare poate determina *o mai bună înțelegere* a sistemului studiat.

1.3 Avantajele și dezavantajele utilizării simulării (2)

Dezavantajele unui studiu de simulare

- Simularea folosește un model – reprezentând o descriere simplificată a sistemului real – care nu poate surprinde întreaga complexitate a unui sistem real.

1.4 Aplicații ale simulării

- Simularea unui aeroport având ca scop minimizarea întârzierilor provocate de aglomerări
- Simularea spațiului aerian având ca scop utilizarea optimă a acestuia
- Simularea în domeniul medical a numărului de cadre medicale necesar la momente diferite de timp
- Planificarea strategică a operațiilor realizate într-o bancă și realizarea unei topologii optime a sucursalelor băncii
- Simularea unei linii de producție pentru a estima cantitatea de materie primă necesară și pentru a prezice profitul obținut

1.5 Schema de modelare și simulare

Definiții

- Definirea problemei pe care ne propunem să o rezolvăm sau a întrebării la care trebuie să răspundem
- Definirea sistemului – o colecție de obiecte/entități ale căror proprietăți vrem să le studiem

Analiza sistemului

- Identificarea părților sistemului relevante pentru problemă

Modelarea

- Dezvoltarea modelului asociat sistemului

Simularea

- Aplicarea modelului în studiul problemei
- Dezvoltarea unei strategii pentru rezolvarea problemei

Validarea

- Verificarea dacă strategia elaborată în pasul de simulare rezolvă problema unui sistem real

1.6 Construcția unui model de simulare (1) – Ceasul simulării

- *Ceasul simulării:* asigură *ordonarea* corectă în timp a *evenimentelor* create de model și uneori ajută la implementarea condiției de terminare a simulării.
- *Evenimentele* corespund unor modificări în sistem; de exemplu modificarea valorilor unor variabile care se calculează sau se generează prin instrucțiuni ale modelului
- Ceasul pornește cu valoarea zero la începutul simulării; simularea se realizează până când ceasul atinge o valoare dată inițial T_{max}
- O *clasificare* a ceasului simulării
  - *ceas cu creștere constantă:* ceasul crește cu o unitate de timp c constantă, apoi se prelucrează toate evenimentele apărute pe intervalul de timp de lungime c
  - *ceas cu creștere variabilă:* ceasul crește cu valoarea ce corespunde apariției primului eveniment următor, apoi programul prelucrează evenimentul și se reia ciclul simulării

1.6 Construcția unui model de simulare (2) – Agenda simulării

- *Agenda* conține elementele/variabilele asociate evenimentelor memorate de modelul de simulare
- Agenda A se compune din două părți:
  - *AEC – agenda evenimentelor curente* (care au timpul de apariție egal cu valoarea ceasului)
  - *AEV – agenda evenimentelor viitoare* (care au timpul de apariție ulterior valorii curente a ceasului)
- Algoritmul de simulare prelucrează numai evenimentele din AEC; prelucrarea unui eveniment înseamnă fie determinarea apariției unui nou eveniment (ce se memorează în AEV) sau modificarea unei stări, fie distrugerea unui eveniment (ștergerea) lui din agendă.

1.6 Construcția unui model de simulare (3) – Algoritmul simulării

- *Algoritmul simulării* actualizează agenda prin *interacțiunea acesteia cu ceasul*
- Într-un *ciclu al simulării* ceasul este actualizat, se selectează din agenda A evenimentele care fac parte din AEC și se prelucrează aceste evenimente până când AEC devine vidă.
- Atunci ceasul este crescut din nou și se reia *ciclul de simulare*.

1.7 Exemplu: Modelul de simulare asociat unui sistem de așteptare (1)

- *Sistemul de așteptare* este o parte a lumii reale în care se produc *aglomerări*.
- *Entitățile componente ale sistemului*:
  - *clienții* care sosesc în sistem
  - *stațiile de servire* (care servesc după anumite reguli clienții)
- Dacă sosesc mulți clienți și stațiile de servire nu pot să îi servească se formează *cozi de așteptare* în sistem.
- Administratorul sistemului trebuie să construiască sistemul a.î. nici clienții să nu *aștepte* mult până primesc serviciul, dar nici stațiile să nu *lenevească*, deoarece ar putea produce pierderi proprietarului.
- *Aplicații ale sistemelor de așteptare:* în comerț, în managementul sistemelor de comunicații și de transport, dar și în dirijarea și funcționarea în timp real a rețelelor de calculatoare.

1.7 Exemplu: Modelul de simulare asociat unui sistem de așteptare (2)

Un model de așteptare conține de obicei următoarele elemente:

- *Variabile de intrare (cunoscute, de obicei aleatoare)*:
  - *TI*: timp de intersosire al clienților
  - *TS*: timp de servire al unui client
- *Variabile de ieșire (necunoscute)*:
  - *TA*: timp de așteptare (sau *LC*: lungimea cozii)
  - *TL*: timp de lenevire (sau *NSL*: numărul stațiilor care lenevesc)

Scopul modelului: cunoscând repartițiile de probabilitate ale variabilelor *TI* și *TS* vrem să obținem informații despre *TA* și *TL* și să se stabilească cum trebuie să se realizeze *TS* a.î. să se optimizeze o anumită funcție criteriu.

2.1 Descriere generală a limbajului GPSS

- *Limbajele de simulare* sunt limbaje de programare care implementează elementele de bază ale simulării: *ceasul simulării* și *gestionarea memoriei*.
- *Utilizatorul* se ocupă de *descrierea evenimentelor* și prelucrarea lor.
- Exemplu de limbaj de simulare: *GPSS (General Purpose System Simulator)*
- *Entități ale modelului*:
  - *Blocuri:* entități care descriu activități
  - *Tranzacții:* sunt elemente create printr-un bloc GENERATE care parcurg blocurile modelului/sistemului secvențial și în funcție de acțiunea blocurilor accesate.
  - *Stații de servire* sau *facilități:* resurse care oferă un singur serviciu (adică unei singure tranzacții)
  - *Multistații de serviciu* sau *depozite:* resurse care oferă servicii mai multor tranzacții simultan
  - *Entități de calcul:* variabile, funcții
  - *Entități statistice:* cozi, histograme

2.2 Structura instrucțiunii GPSS

Etichetă	Numele blocului (Cuvânt cheie)	Parametri (separați prin virgulă)
Opțională (pt. referiri)	Verb imperativ indică acțiunea blocului	A,B,C,...

2.3 Blocuri de creare și distrugere de tranzacții

- *GENERATE* – creează tranzacții. Are forma generală:
  GENERATE A,B,C,D,E
  - A – intervalul mediu între sosiri
  - B – abaterea; lungimea intervalului de timp între generarea a două tranzacții consecutive este o valoare întreagă a unei variabile aleatoare uniform distribuite pe intervalul [A - B, A + B]
  - C – momentul creării primei tranzacții
  - D – numărul maxim de tranzacții generate
  - E – prioritatea (valoare implicită = 0)
- *TERMINATE* – determină distrugerea unei tranzacții. Are forma generală:
  TERMINATE A
  - A – decrementează parametrul dat la START (în fereastra care apare în urma rulării programului). De exemplu dacă avem "START *nr*" atunci, în momentul în care o tranzacție accesează blocul "TERMINATE A","*nr*" devine "*nr* = *nr* - A". Programul rulează cât timp "*nr* > 0".

2.4 Blocuri de acțiune (1)

- *SEIZE/RELEASE* – se folosesc împreună și determină ocuparea/eliberarea unei resurse denumită *facilitate*; de asemenea creează o facilitate, numele facilității fiind dat ca parametru. Au forma generală:

  SEIZE resursă

  ....

  RELEASE resursă

  - resursă – este numele facilității
- *ADVANCE* – determină oprirea/blocarea tranzacției care accesează acest bloc un interval de timp dat de parametri. Are forma generală:

  ADVANCE A,B

  - A – timp mediu de blocare/întârziere a tranzacției
  - B – abaterea

2.4 Blocuri de acțiune (2.1)

- *ENTER/LEAVE* – se folosesc împreună și determină ocuparea/eliberarea unei/mai multor componente dintr-o resursă denumită *punct de servire cu mai multe stații* sau *entitate de depozitare*.
- *Capacitatea* (numărul de componente care pot fi ocupate) *resursei* și *entitatea asociată* sunt definite prin blocul *STORAGE*.
- *Forma generală a blocurilor* este descrisă în slide-ul următor.

2.4 Blocuri de acțiune (2.2)

- Sintaxa blocurilor STORAGE/ENTER/LEAVE:

  resursă STORAGE capacitate

  ....

  GENERATE A1,B1

  ....

  ENTER resursă,spații_ocupate

  ....

  LEAVE resursă,spații_eliberate

  ....

  TERMINATE A2

  - resursă – este numele punctului de servire cu mai multe stații sau a entității de depozitare
  - spații_ocupate – numărul de componente ocupate din entitatea "resursă" când o tranzacție accesează blocul ENTER
  - spații_eliberate – numărul de componente eliberate din entitatea "resursă" când o tranzacție accesează blocul LEAVE
  - spații_ocupate, spații_eliberate – trebuie să ia valori mai mici sau egale decât valoarea parametrului "capacitate"

2.4 Blocuri de acțiune (3)

- *PREEMPT/RETURN* – se folosesc împreună și determină ocuparea/eliberarea unei facilități; ocuparea facilității are loc în funcție de prioritatea asociată tranzacției. Au forma generală:

  PREEMPT resursă,prioritate,blocTransfer

  ....

  RETURN resursă

  - resursă – este numele facilității
  - prioritate – poate fi PR sau poate să lipsească.
    - Dacă este PR atunci o tranzacție care accesează un bloc PREEMPT poate îndepărta tranzacția care ocupă deja facilitatea (pe care a ocupat-o printr-un bloc SEIZE sau PREEMPT) în funcție de prioritate (tranzacția având prioritatea mai mare va ocupa în continuare facilitatea).
    - Dacă PR lipsește atunci PREEMPT poate îndepărta doar o tranzacție care a ocupat facilitatea prin blocul SEIZE, în caz contrar tranzacția va fi introdusă în coada de așteptare.
  - blocTransfer – este opțional și reprezintă blocul la care este dirijată tranzacția care ocupa inițial facilitatea. Dacă lipsește atunci tranzacția va fi introdusă în coada de așteptare și după eliberarea facilității va primi în continuare serviciul.

2.5 Blocuri pentru obținerea de statistici (1)

- *QUEUE/DEPART* – se folosesc împreună și permit obținerea de statistici/informații referitoare la coada de așteptare. Au forma generală:

  QUEUE coada

  ....

  DEPART coada

  - coada – este numele cozii de așteptare
- *ATENȚIE:* blocurile *QUEUE/DEPART* NU determină crearea unei cozi de așteptare, permit doar obținerea de informații referitoare la coada de așteptare, care este creată implicit de către limbajul GPSS în momentul în care o tranzacție accesează unul dintre blocurile de acțiune *SEIZE, ENTER sau PREEMPT*

2.5 Blocuri pentru obținerea de statistici (2.1)

- *TABLE/TABULATE* – reprezintă grafic histograma asociată unei mulțimi de selecție. Au forma generală:

  etHist TABLE A,B,C,D

  ....

  GENERATE A1,B1

  ....

  TABULATE etHist

  ....

  TERMINATE A2

  - etHist – numele histogramei, parametru obligatoriu
  - A – expresie care furnizează valori de selecție (generate cf. unei repartiții de probabilitate) asupra unei v.a. X
  - B – a_1: marginea superioară a primului interval de frecvențe: $I_1 = [a_0, a_1]$ (în notațiile din Curs 7)
  - C – $l = a_i - a_{i-1}$: lungimea unui interval de frecvențe
  - D – k: numărul intervalelor de frecvențe

2.5 Blocuri pentru obținerea de statistici (2.2)

Construcția histogramei este iterativă:

- în momentul în care o tranzacție accesează blocul "TABULATE etHist", se calculează expresia definită prin parametrul "A" al histogramei cu numele "etHist";
- valoarea expresiei reprezintă o valoare de selecție, notată, x_i pentru tranzacția i, asupra unei variabile aleatoare;
- la fiecare accesare a blocului "TABULATE" de către o tranzacție se actualizează mulțimea valorilor de selecție, $S_n = S_{n-1} \cup \{x_n\}$, unde $S_{n-1} = \{x_1, x_2, \ldots, x_{n-1}\}$ este mulțimea de valori de selecție obținută pentru primele $n - 1$ tranzacții, iar x_n este valoarea expresiei date prin parametrul A calculată pentru tranzacția a n – a care accesează blocul "TABULATE";
- histograma este actualizată, fiind construită pentru mulțimea valorilor de selecție S_n.

2.5 Blocuri pentru obținerea de statistici (3)

- *QTABLE/QUEUE/DEPART* – reprezintă grafic histograma asociată timpilor de așteptare la un serviciu. Au forma generală:

  etHist QTABLE numeCoada,B,C,D

  ....

  GENERATE A1,B1

  ....

  QUEUE numeCoada

  ....

  DEPART numeCoada

  ....

  TERMINATE A2

  - etHist – numele histogramei, element obligatoriu
  - numeCoada – numele cozii de așteptare în care sunt introduse tranzacțiile pentru care se calculează timpul de așteptare pentru a primi un serviciu
  - B, C, D – parametri care definesc histograma; au aceleași semnificații ca pentru "TABLE"

2.6 Blocuri de control logic al tranzacțiilor (1)

- *TEST* – permite compararea unor valori și dirijarea tranzacțiilor în modelul de simulare. Are forma generală:

  TEST O A,B,etTransfer

  ....

  etTransfer BLOC parametri
  - O – este un operator relațional. Pentru un test cu succes trebuie să se verifice relația "A O B"; unde "O" poate lua valorile E = "=", G = ">", GE = "≥" , L = "<", LE = "≤", NE = "≠"
  - A, B – sunt valorile comparate
  - etTransfer – reprezintă eticheta blocului la care este dirijată tranzacția în cazul în care testul nu are succes
- *Observație:* dacă testul are succes (adică, condiția este adevărată) tranzacția accesează următorul bloc din program.

2.6 Blocuri de control logic al tranzacțiilor (2)

- *TRANSFER* – permite dirijarea tranzacțiilor în modelul de simulare. Două modalități de utilizare sunt prezentate în cele ce urmează:

  TRANSFER ,etTransfer

  sau

  TRANSFER prob,,etTransfer

  unde

  etTransfer BLOC parametri
  - etTransfer – este eticheta blocului la care este dirijată tranzacția
  - prob – reprezintă probabilitatea cu care o tranzacție care accesează blocul "TRANSFER" este dirijată la blocul cu eticheta "etTransfer"; cu probabilitatea "1 - prob" tranzacția este trimisă la blocul următor din program

2.7 Blocuri de modificare a caracteristicilor tranzacțiilor și a valorilor unor entități de referință (1)

- *ASSIGN* – permite atribuirea/modificarea unei valori/valorii unui parametru asociat tranzacției care accesează blocul. Are forma generală:

  ASSIGN param[±],valoare
  - param – este identificatorul asociat parametrului/caracteristicii tranzacției; valoarea parametrului poate fi obținută folosind atributul P$param (dacă "param" este șir de caractere) sau Pparam (dacă "param" este număr)
  - valoare – este valoarea atribuită sau cu care se incrementează sau decrementează parametrul tranzacției
- [±] – parantezele pătrate nu fac parte din sintaxa blocului; indică opționalitatea parametrilor "+" sau "-".
  - Dacă operatorii "+" sau "-" lipsesc atunci P$param = valoare
  - Dacă blocul ASSIGN are sintaxa: "ASSIGN param+,valoare" atunci P$param = P$param + valoare
  - Dacă blocul ASSIGN are sintaxa: "ASSIGN param−,valoare" atunci P$param = P$param - valoare

2.7 Blocuri de modificare a caracteristicilor tranzacțiilor și a valorilor unor entități de referință (2)

- *SAVEVALUE* – permite atribuirea/modificarea unei valori/valorii unei variabile (globale) atunci când o tranzacție accesează blocul. Are forma generală:

  SAVEVALUE var[±],valoare
  - var – este identificatorul asociat variabilei; valoarea variabilei poate fi obținută folosind atributul X$var (dacă "var" este șir de caractere) sau Xvar (dacă "var" este număr)
  - valoare – este valoarea atribuită sau cu care se incrementează sau decrementează variabila
- [±] – parantezele pătrate nu fac parte din sintaxa blocului; indică opționalitatea parametrilor "+" sau "-".
  - Dacă operatorii "+" sau "-" lipsesc atunci X$var = valoare
  - Dacă blocul SAVEVALUE are sintaxa: "SAVEVALUE var+,valoare" atunci X$var = X$var + valoare
  - Dacă blocul SAVEVALUE are sintaxa: "SAVEVALUE var−,valoare" atunci X$var = X$var - valoare

Universitatea din București
Facultatea de Matematică și Informatică

CURSUL nr. 10
METODE MODERNE DE CALCUL ȘI SIMULARE

Modele de simulare

Lect. dr. Bianca Mogoș

Conținut

Operatori aritmetici/relaționali/logici

- *Operatori aritmetici:*
  - "+" – adunare, "−" – scădere
  - "#" – înmulțire, "/" – împărțire
  - "\" – împărțire având ca rezultat o valoare întreagă
  - "^" – putere, "@" – restul împărțirii
- *Operatori relaționali:*
  - "G" – $>$, "GE" – $\geq$
  - "L" – $<$, "LE" – $\leq$
  - "E" – $=$, "NE" – $\neq$
- *Operatori logici:*
  - "AND" – "și" logic
  - "OR" – "sau" logic
  - "NOT" – "negație" logică

Obținerea unor caracteristici ale entităților GPSS (1)

- *SNA – System Numerical Attributes* – reprezintă "variabile de stare" ale sistemului care pot fi interogate în cadrul unui model de simulare

O clasificare a atributelor din SNA:

- Variabile furnizând *informații despre sistem*:
  - C1 – timpul de simulare de la ultimul RESET
  - AC1 – timpul de simulare de la ultimul CLEAR
- Variabile furnizând *informații despre tranzacții*:
  - M1 – lungimea intervalului de timp scurs de la crearea unei tranzacții și până în momentul în care se evaluează atributul
  - PR – prioritatea tranzacției
  - XID1 – indice asociat tranzacției la generare

Obținerea unor caracteristici ale entităților GPSS (2)

- Variabile furnizând *informații despre blocuri*:
  - W$etichetaBloc – numărul curent de tranzacții aflate în blocul având eticheta "etichetaBloc"
  - N$etichetaBloc – numărul total de tranzacții care au intrat în blocul având eticheta "etichetaBloc"
- Variabile furnizând *informații despre facilitatea cu numele "resursă"*:
  - F$resursă – indică starea curentă: 0 – liberă, 1 – ocupată
  - FC$resursă – numărul de tranzacții care au ocupat facilitatea
  - FR$resursă – procentul de timp (din timpul total de simulare) cât a fost ocupată facilitatea (ia valori între 0 și 1000)
  - FT$resursă – timpul mediu de ocupare de către o tranzacție

Obținerea unor caracteristici ale entităților GPSS (3)

- Variabile furnizând *informații despre punctul de servire cu mai multe stații sau entitatea de depozitare cu numele "resurse"*:
  - R$resurse – numărul de stații/unități libere (care pot fi ocupate de tranzacții)
  - S$resurse – numărul de stații/unități ocupate
  - SA$resurse – numărul mediu de stații/unități ocupate
  - SR$resurse – procentul de utilizare
  - ST$resurse – timpul mediu de utilizare a entității
- Variabile furnizând *informații despre coada de așteptare cu numele "coadă"*:
  - Q$coadă – numărul curent de tranzacții din coada de așteptare
  - QA$coadă – numărul mediu de tranzacții din coadă
  - QM$coadă – numărul maxim de tranzacții aflate la un anumit moment al timpului de simulare în coadă
  - QC$coadă – numărul total de tranzacții care au intrat în coada de așteptare
  - QT$coadă – timpul mediu de așteptare în coadă
  - QX$coadă – timpul mediu de așteptare în coadă calculat pentru tranzacțiile care au așteptat un timp strict pozitiv

Folosirea repartițiilor de probabilitate în GPSS (1)

- Algoritmul de generare a numerelor (pseudo)aleatoare în GPSS se bazează pe *algoritmul multiplicativ – congruențial al lui Lehmer* cu o perioadă maximă de $\lambda = 2^{31} - 2$.
- *Atributul RNj* returnează o valoare întreagă în intervalul $[0, 999]$.
- Comanda *RMULT* setează sământa generatorului pentru generatorii "RNj, j = 1,. . . , 7".
- În absența comenzii "RMULT" sământa coincide cu numărul *j* al generatorului "RNj". De exemplu, RN2 pornește cu sământa 2.

Folosirea repartițiilor de probabilitate în GPSS (2)

- *Forma generală a unei funcții de generare* de valori de selecție asupra unei variabile aleatoare având o repartiție de probabilitate dată este (de obicei):

  numeRepartiție(nrGen, locație, scală, formă)

  unde
  - *nrGen* – reprezintă numărul generatorului de numere aleatoare; trebuie să fie o valoare ≥ 1.
  - *locație, scală, formă* – sunt parametri ai repartiției de probabilitate

Folosirea repartițiilor de probabilitate în GPSS (3)

- *Exemple* de funcții de generare de valori aleatoare

Repartiția de probabilitate	*Funcția de generare*
$X \sim U(a, b)$	UNIFORM(1, a, b)
$X \sim N(\mu, \sigma^2)$	NORMAL(1, μ, σ)
$X \sim Exp(\alpha, \lambda)$	EXPONENTIAL(1, α, $1/\lambda$)
$X \sim Gamma(\alpha, \lambda, \nu)$	GAMMA(1, α, $1/\lambda$, ν)
$X \sim Bin(n, p)$	BINOMIAL(1, n, p)
$X \sim Poisson(\lambda)$	POISSON(1, λ)

Folosirea repartițiilor de probabilitate în GPSS (4)

- *FUNCTION* - este o comandă care definește o entitate funcție. Forma generală este:

  nume FUNCTION A,B
  $f_1, a_1/f_2, a_2/\ldots/f_n, a_n$

  - $f_1, f_2, \ldots, f_n$ trebuie să verifice condiția $f_1 < f_2 < \ldots < f_n$
  - nume - este o etichetă obligatorie prin care va fi accesată valoarea returnată de funcție
  - A - este argumentul funcției (poate fi o expresie) ce va fi evaluat;
  - B – este un parametru format dintr-o literă care indică tipul funcției (poate fi C – continuă sau D – discretă) și o valoare numerică reprezentând numărul n de perechi (f_i, a_i), $i = 1, 2, \ldots, n$ din definiția funcției.

- Pentru a obține valoarea returnată de funcție se folosește *atributul FN\$nume*.

Folosirea repartițiilor de probabilitate în GPSS (5)

Algoritmul care stă la baza comenzii "FUNCTION":

- Se evaluează expresia definită în argumentul funcției, notăm valoarea obținută cu A_0
- Se determină indicele i minim pentru care $A_0 \leq f_i$.
- Avem următoarele cazuri:
  - Dacă $f_n < A_0$ atunci $FN\$\text{nume} = a_n$
  - Dacă $A_0 < f_1$ atunci $FN\$\text{nume} = a_1$
  - Dacă $B = Cn$ și $f_{i-1} < A_0 \leq f_i$ atunci valoarea funcției se obține prin interpolare cu formula:

    $$FN\$\text{nume} = a_{i-1} + (a_i - a_{i-1})\#(A_0 - f_{i-1})/(f_i - f_{i-1})$$

  - Dacă $B = Dn$ și $f_{i-1} < A_0 \leq f_i$ atunci $FN\$\text{nume} = a_i$

Folosirea repartițiilor de probabilitate în GPSS (6)

Folosirea entității funcție de tip continuu pentru simularea unei v.a. continue (nu este recomandată, se recomandă folosirea funcțiilor de generare).

- Fie v.a. continuă X având funcția de repartiție $F(x)$. În acest caz,
  - argumentul funcției va fi generatorul de numere aleatoare RN1 (generează valori între 0 și 0.999999 când este folosit de comanda "FUNCTION")
  - Parametrii din sintaxa comenzii "FUNCTION":
    $a_1 < a_2 < \ldots < a_n$ și $f_1 < f_2 < \ldots < f_n$ sunt astfel încât

    $$f_i = F(a_i) = P(X \leq a_i), \forall i = 1, 2, \ldots, n$$

- *Exemplu:* aproximarea simulării v.a. $X \sim Exp(1)$

  varC FUNCTION RN1,C10
  0.05,0.0513/0.15,0.1625/0.25,0.2877/0.35,0.4308/0.45,0.5978/
  0.55,0.7985/0.65,1.0498/0.75,1.3863/0.85,1.8971/0.95,2.9957

Folosirea repartițiilor de probabilitate în GPSS (7)

- *Exemplu* de folosire a entității funcție de tip continuu pentru evaluarea unui argument și asocierea valorii corespunzătoare determinate prin interpolare

  varC FUNCTION C1,C7
  5,15/10,20/15,10/20,10/25,15/30,20/35,10

- Se evaluează argumentul funcției, C1 – timpul de simulare; de exemplu, dacă C1 = 22 atunci funcția va returna valoarea corespunzătoare lui $A_0 = C1 = 22$ prin interpolarea perechilor (20, 10) și (25, 15), și anume

$$FN\$varC = 10 + (15 - 10)\#(22 - 20)/(25 - 20) = 12$$

Folosirea repartițiilor de probabilitate în GPSS (8)

Folosirea entității funcție de tip discret pentru simularea unei v.a. discrete

- Fie v.a. discretă X definită prin repartiția

$$X : \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix}, \sum_{i=1}^{n} p_i = 1, a_1 < a_2 < \dots < a_n.$$

  - argumentul funcției va fi generatorul de numere aleatoare RN1
  - Parametrii din sintaxa comenzii "FUNCTION":
    – $\{a_1, a_2, \dots, a_n\}$ reprezintă valorile pe care le ia v.a. X
    – $\{f_1, f_2, \dots, f_n\}$ reprezintă frecvențele cumulate, adică

$$f_i = p_1 + p_2 + \dots + p_i, i = \overline{1, n}$$

- *Exemplu*

  varD FUNCTION RN1,D5
  0.1,5/0.35,10/0.7,20/0.8,25/1,30

Folosirea repartițiilor de probabilitate în GPSS (9)

- *Exemplu* de folosire a entității funcție de tip discret pentru evaluarea unui argument și asocierea valorii corespunzătoare

  varD FUNCTION C1,D7
  5,15/10,20/15,10/20,10/25,15/30,20/35,10

- Se evaluează argumentul funcției, C1 – timpul de simulare; de exemplu, dacă C1 = 22 atunci funcția va returna valoarea

$$FN\$varD = a_5 = 15$$

Predefinirea unor expresii constante sau variabile (1)

Predefinirea unor expresii constante

- *EQU* - este o comandă care definește o entitate asociată unei expresii având o valoare constantă. Forma generală este:
  nume EQU X
  - X - este o expresie obligatorie având o valoare constantă
  - nume - este o etichetă obligatorie și reprezintă numele constantei ce va lua valoarea expresiei X.

Exemplu care ilustrează folosirea constantei predefinite prin blocul "EQU"

- timpServire EQU (150#72 + 36)
  GENERATE 5
  ....
  SEIZE punctServire
  ADVANCE timpServire
  RELEASE punctServire
  ....
  TERMINATE 1

Predefinirea unor expresii constante sau variabile (2)

Predefinirea unor expresii variabile

- *BVARIABLE/VARIABLE/FVARIABLE* - este o comandă care definește o entitate asociată unei expresii a cărei valoare poate să varieze când se reevaluează expresia. Forma generală este:

  nume BVARIABLE X ; X este o expresie logică
  nume VARIABLE X ; X este o expresie ce ia o valoare întregă
  nume FVARIABLE X ; X este o expresie ce ia o valoare reală

  - nume - este o etichetă obligatorie și reprezintă numele entității ce va lua valoarea expresiei X
  - X - este o expresie obligatorie având o valoare care poate varia când se reevaluează expresia.
- *Expresia X se evaluează* folosind
  - atributul BV$nume pentru comanda "BVARIABLE"
  - atributul V$nume pentru comanda "VARIABLE"
  - atributul V$nume pentru comanda "FVARIABLE"
- BV$nume(V$nume) = valoarea expresiei X.

Predefinirea unor expresii constante sau variabile (3)

- *Exemplu – folosirea blocurilor "VARIABLE" și "FVARIABLE"*

timp VARIABLE RN1/100	timp FVARIABLE RN1/100
GENERATE 5	GENERATE 5
SEIZE servire	SEIZE servire
ADVANCE V$timp	ADVANCE V$timp
RELEASE servire	RELEASE servire
TERMINATE 1	TERMINATE 1

Predefinirea unor expresii constante sau variabile (4)

- *Exemplu – folosirea blocului "BVARIABLE"*

```
timp BVARIABLE (25<M1'AND'M1<30)
astept VARIABLE RN1@16+20
GENERATE 30
SEIZE serviciu
ADVANCE V$astept
RELEASE serviciu
SAVEVALUE 1,M1
TEST E BV$timp,1,termin
TERMINATE 1

termin TERMINATE 1
```

Definirea paramerilor tranzacțiilor și a variabilelor (1)

- *ASSIGN* – permite atribuirea/modificarea unei valori/valorii unui parametru asociat tranzacției care accesează blocul. Are forma generală:

  ASSIGN param[±],valoare

  - param – este identificatorul asociat parametrului/caracteristicii tranzacției; valoarea parametrului poate fi obținută folosind atributul P$param (dacă "param" este șir de caractere) sau Pparam (dacă "param" este număr)
  - valoare – este valoarea atribuită sau cu care se incrementează sau decrementează parametrul tranzacției
- [±] – parantezele pătrate nu fac parte din sintaxa blocului; indică opționalitatea parametrilor "+" sau "-".
  - Dacă operatorii "+" sau "-" lipsesc atunci P$param = valoare
  - Dacă blocul ASSIGN are sintaxa: "ASSIGN param+,valoare" atunci P$param = P$param + valoare
  - Dacă blocul ASSIGN are sintaxa: "ASSIGN param−,valoare" atunci P$param = P$param - valoare

Definirea paramerilor tranzacțiilor și a variabilelor (2)

- *SAVEVALUE* – permite atribuirea/modificarea unei valori/valorii unei variabile (globale) atunci când o tranzacție accesează blocul. Are forma generală:

  SAVEVALUE var[±],valoare

  - var – este identificatorul asociat variabilei; valoarea variabilei poate fi obținută folosind atributul X$var (dacă "var" este șir de caractere) sau Xvar (dacă "var" este număr)
  - valoare – este valoarea atribuită sau cu care se incrementează sau decrementează variabila
- [±] – parantezele pătrate nu fac parte din sintaxa blocului; indică opționalitatea parametrilor "+" sau "-".
  - Dacă operatorii "+" sau "-" lipsesc atunci X$var = valoare
  - Dacă blocul SAVEVALUE are sintaxa: "SAVEVALUE var+,valoare" atunci X$var = X$var + valoare
  - Dacă blocul SAVEVALUE are sintaxa: "SAVEVALUE var−,valoare" atunci X$var = X$var - valoare

Sisteme de așteptare – Exemplu 1

- *Modelarea și simularea rezolvării unor proiecte diferite* (considerate în ordinea apariției sau ordonate în funcție de priorități) într-o firmă având mai multe departamente. Câteva modalități de abordare a proiectelor având ca scop rezolvarea acestora:
  - *rezolvarea secvențială a unui proiect:* proiectul va trece prin fiecare departament în mod secvențial în sensul că Departamentul 2 depinde de partea realizată de Departamentul 1, Departamentul 3 primește proiectul de la Departamentul 2, etc.
  - *rezolvarea în paralel a părților proiectului*; se presupune că pentru a se finaliza proiectul Departamentul 2 și Departamentul 3 trebuie să discute părțile realizate separat.

Sisteme de așteptare – Exemplu 2

- *Modelarea și simularea dezvoltării unui produs* într-o fabrică având mai multe departamente
  - *construirea produsului în mod secvențial:* componenta construită în Departamentul 2 depinde de componenta contruită în Departamentul 1, etc.
  - *construirea produsului în paralel:* fiecare departament construiește o anumită componentă a produsului, în final "Departamentul asamblare" va realiza asamblarea componentelor și va construi produsul final.

Sisteme de modelare a stocurilor – Exemplu

- Presupunem că sucursala unei bănci deține 2 bancomate. Presupunem că suma maximă cu care este alimentat un bancomat este de 10000 de lei. Se realizează următoarele operațiuni:
  - la un interval repartizat exponential de medie 20 de minute se extrag sume având valori repartizate uniform între 50 și 2000 de lei. Durata unei tranzacții este de 5±2 minute.
  - La fiecare 2 ore un angajat al băncii verifică dacă bancomatul conține o sumă mai mică de 5000 de lei. În acest caz, soldul bancomatului va fi completat până la capacitatea maximă.
- Să se simuleze activitatea sistemului în primele 8 ore.

Universitatea din București
Facultatea de Matematică și Informatică

CURSUL nr. 11
METODE MODERNE DE CALCUL ȘI SIMULARE

Modele de simulare

Lect. dr. Bianca Mogoș

Conținut

1.1 Generalități privind procesele stochastice (1)

- *Definiție:* Fie $(E, \mathcal{B}, P)$ un câmp de probabilitate. Se numește *proces stochastic* o familie de variabile aleatoare
$$\{X_t : E \to \mathbb{R}\}_{t \in T \subseteq \mathbb{R}}$$
depinzând de parametrul real t (considerat adesea timpul).
- *Observații:*
  - Dacă T este o mulțime discretă ($T = \{0, 1, 2, \ldots\}$) atunci procesul se numește *lanț*.
  - Dacă $T = I \subseteq \mathbb{R}$, I un interval atunci procesul este cu *timp continuu*.
- *Notație:* fie S mulțimea tuturor valorilor pe care le pot lua v.a. X_t. Elementele mulțimii S se numesc *stări*.
- *Definiție:* Mulțimea $\{(t, X_t), t \in T\}$ se numește *traiectorie a procesului stochastic*.

1.1 Generalități privind procesele stochastice (2)

- *Procesul stochastic este cunoscut* dacă pentru $\forall n, t_1, t_2, \ldots, t_n$ este cunoscută funcția de repartiție:
$$F_{t_1,t_2,\ldots,t_n}(x_1, x_2, \ldots, x_n) = P(X_{t_1} \le x_1, X_{t_2} \le x_2, \ldots, X_{t_n} \le x_n). \quad (1)$$
- *Definiție:* Procesul stochastic $\{X_t\}_{t \in T \subseteq \mathbb{R}}$ se numește *staționar tare* dacă $\forall n, t_1, t_2, \ldots, t_n$ avem
$$F_{t_1+h,t_2+h,\ldots,t_n+h}(x_1, x_2, \ldots, x_n) = F_{t_1,t_2,\ldots,t_n}(x_1, x_2, \ldots, x_n) \quad (2)$$
- *Definiție:* Procesul stochastic $\{X_t\}_{t \in T \subseteq \mathbb{R}}$ se numește *staționar slab* dacă admite momente de ordinul doi, adică există și sunt finite
$$m_t = E[X_t], \sigma_{st} = cov[X_s, X_t], s < t \text{ și}$$
$$m_t = m = constant \text{ și } \sigma_{st} = \sigma(t-s). \quad (3)$$

1.1 Generalități privind procesele stochastice (3)

- *Simularea proceselor stochastice* înseamnă producerea de puncte ale unei traiectorii finite
- *Simularea unui proces stochastic* constă în:
  - generarea unei valori X_0 a procesului presupusă a se afla pe o traiectorie a sa la momentul $t = 0$;
  - simularea unei valori X_1 aflată pe aceeși traiectorie la momentul $t = 1$ (care poate depinde de variabila X_0);
  - simularea unei valori X_2 aflată pe aceeși traiectorie la momentul $t = 2$ (care poate depinde de variabilele X_0 și X_1), etc.
- *Problema simulării unui proces stochastic* este mult mai complicată decât a variabilelor aleatoare, deoarece v.a. X_t generate la momentul de timp t pot să nu fie independente pentru diverse valori ale lui t.

1.2 Lanțuri și procese Markov (1)

- *Definiție:* Procesul $\{X_t\}_{t\in T\subseteq\mathbb{R}}$ discret este un *proces Markov* dacă pentru orice $t_1 < t_2 < \ldots < t_n$ satisface proprietatea:

$$P(X_{t_n} = x_n \mid X_{t_{n-1}} = x_{n-1}, \ldots, X_{t_1} = x_1) = \\ = P(X_{t_n} = x_n \mid X_{t_{n-1}} = x_{n-1}) \tag{4}$$

adică procesul *nu are memorie*.

- *Definiție:* Probabilitățile

$$P(X_{t_n} = x_n \mid X_{t_{n-1}} = x_{n-1}) \tag{5}$$

se numesc *probabilități de tranziție sau probabilități de trecere*.

- Dacă $T = \{0, 1, 2, \ldots, n\}$ și $S = \{1, 2, \ldots, m\}$ atunci procesul Markov este un *lanț cu un număr finit de stări*.

1.2 Lanțuri și procese Markov (2)

- *Ipoteză:* Presupunem cunoscute probabilitățile inițiale:

$$p_i = P(X_t = i), \forall i \in S \tag{6}$$

- *Proprietate:* Dacă $\{X_n\}_{n\in\mathbb{N}}$ este un lanț Markov are loc următoarea relație:

$$P(X_n = i_n, \ldots, X_0 = i_0) = p_{i_0}\prod_{t=1}^{n} P(X_t = i_t \mid X_{t-1} = i_{t-1}) \tag{7}$$

- *Notație:*

$$p_{t;i_{t-1}i_t} = P(X_t = i_t \mid X_{t-1} = i_{t-1}) \tag{8}$$

- *Definiție:* Un lanț Markov se numește *omogen* dacă $p_{t;i_{t-1}i_t}$ nu depinde de t și notăm $p_{t;i_{t-1}i_t} = p_{i_{t-1}i_t}$. Altfel, se numește lanț *neomogen*.

1.2 Lanțuri și procese Markov (3)

- Pentru un lanț Markov omogen $\{X_n\}_{n\in\mathbb{N}}$ considerăm matricea $P = (p_{ij})_{i,j}$, unde

$$p_{ij} = P(X_n = j \mid X_{n-1} = i), \forall n \in \mathbb{N}^* \tag{9}$$

numită *matrice de trecere*.

- Matricea de trecere P a unui lanț Markov omogen satisface relațiile:

$$p_{ij} \geq 0, \forall i, j \in S \text{ și } \sum_{j\in S} p_{ij} = 1, \forall i \in S \tag{10}$$

- Fiind dat un lanț Markov omogen, *vrem să determinăm probabilitățile*

$$P(X_{n+m} = j \mid X_m = i) \tag{11}$$

numite *probabilități de trecere în n pași*.

1.2 Lanțuri și procese Markov (4)

- Definim recursiv *șirul* $p_{ij}^{(n)}$ prin

$$\begin{cases} p_{ij}^{(1)} = p_{ij} \\ p_{ij}^{(n+1)} = \sum_{k \in S} p_{ik}^{(n)} p_{kj}^{(1)}, n \in \mathbb{N}^* \end{cases} \tag{12}$$

- *Proprietate:* Pentru orice $n \in \mathbb{N}$ avem relația

$$p_{ij}^{(n)} = P(X_{n+m} = j \mid X_m = i), m \in \mathbb{N} \tag{13}$$

- Dacă într-un lanț Markov omogen se cunosc *probabilitățile inițiale* și *probabilitățile de tranziție* atunci probabilitatea ca la momentul $n \in \mathbb{N}$ sistemul să se afle în starea $i, i \in S$, pe care o notăm $p_i(n) = P(X_n = i)$ este

$$p_i(n) = \sum_{j \in S} p_j(n-1) p_{ji}. \tag{14}$$

1.3 Exemplu: modelarea unui sistem de așteptare folosind lanțuri Markov (1)

- *Ipoteze ale modelului:*
  - Sistemul conține o stație de servire care poate servi clienții la momentele $0, 1, 2, \ldots$
  - Definim v.a. X_n – numărul de clienți care sosesc în intervalul $(n, n+1)$. Presupunem că v.a. X_n sunt i.i.d.:

$$X_n : \begin{pmatrix} k \\ p_k \end{pmatrix}_{k=0}^{\infty}, \sum_{k=0}^{\infty} p_k = 1. \tag{15}$$

  - Se consideră că există un loc de așteptare pentru cel mult m clienți (inclusiv clientul care este servit); clienții care găsesc m clienți în sistem pleacă fără a fi serviți.
  - Fie Y_n – numărul de clienți din sistem la momentul n (incluzând și clientul care este servit)
- Rezultă că $\{Y_n\}_{n \in \mathbb{N}}$ este un lanț Markov cu stările $0, 1, 2, \ldots, m$, deoarece Y_{n+1} depinde doar de Y_n, iar Y_{n+1} și X_n sunt independente.

1.3 Exemplu: modelarea unui sistem de așteptare folosind lanțuri Markov (2)

- *O formulă recursivă pentru lanțul* Y_n

$$Y_{n+1} = \begin{cases} Y_n - 1 + X_n, \text{ dacă } 1 \leq Y_n \leq m, 0 \leq X_n \leq m + 1 - Y_n \\ X_n, \text{ dacă } Y_n = 0, 0 \leq X_n \leq m - 1 \\ m, \text{în rest} \end{cases} .$$

- Notăm cu $\overline{p_m} = p_m + p_{m-1} + \ldots p_0$. *Probabilitățile de tranziție* pot fi calculate cu formulele:

$$\begin{aligned} p_{0j} &= P(Y_{n+1} = j \mid Y_n = 0) = \\ &= \begin{cases} P(X_n = j) = p_j \text{ dacă } 0 \leq j \leq m-1 \\ P(X_n \geq m) = 1 - \overline{p_m} \text{ dacă } j = m \end{cases} \end{aligned}$$

$$\begin{aligned} p_{ij} &= P(Y_{n+1} = j \mid Y_n = i) = \\ &= \begin{cases} P(X_n = j + 1 - i) = p_{j-i+1} \text{ dacă } i - 1 \leq j \leq m - 1 \\ P(X_n \geq m + 1 - i) = 1 - \overline{p_{m+1-i}} \text{ dacă } j = m \\ 0, \text{ altfel} \end{cases} \end{aligned}$$

Bibliografie I

- I. Văduva (2004), *Modele de simulare: note de curs*, Editura Universității din București, București
- A. L. Pletea, L. Popa (1999), Teoria probabilităților, Disponibil la: http://math.etti.tuiasi.ro/lpopa/cursTP.pdf, Ultima accesare: ianuarie 2015