Generatori de numere aleatoare

Curs 2

December 4, 2023

Bibliografie suplimentară:

- Knuth, D. E.(1983) *Tratat de programare a calculatoarelor, Vol. 2 - Algoritmi seminumerici*, Editura Tehnică.
- Knuth D.E. (1974) *Tratat de programare a calculatoarelor, Vol 1 - Algoritmi fundamentali*, Editura Tehnică.
- Văduva, I (1977) *Modele de simulare cu calculatorul*, Ed. Tehnică.

Recapitularea unor noțiuni probabiliste

Spațiu de selecție și evenimente

- **Experiment aleator**= un experiment al cărui rezultat nu este cunoscut înainte.
- **Spațiu de selecție** al unui experiment (S) = spatiul tuturor rezultatelor posibile.
  - De exemplu un experiment aleator poate fi o cursa de cai în care aceștia sunt numerotați de la 1 la 7. Atunci:
  - $S = \{\text{toate permutările șirului } (1, 2, 3, 4, 5, 6, 7)\}$
  - Ce înseamnă rezultatul $(3, 4, 1, 7, 6, 5, 2)$?
- **Eveniment** (A)= orice submulțime a spațiului de selecție. Dacă rezultatul unui experiment aparține lui A, atunci se spune că a avut loc A.
  - Exemplu pentru S de mai sus: $A = \{\text{toate rezultatele din } S \text{ care încep cu } 5\}$

Pot fi definite:

- reuniune de doua evenimente $A \cup B$;
- intersecție de două evenimente $A \cap B$;
- reuniune de n evenimente $\bigcup_{i=1}^{n} A_i$;
- intersecție de n evenimente $\bigcap_{i=1}^{n} A_i$;
- complementarul unui eveniment A: A^c.
    - $S^c = \phi$;
    - Daca $A \cap B = \phi$, evenimentele A și B se exclud reciproc.

Axiomele probabilității

- **Probabilitate**: presupunem că pentru fiecare eveniment A asociat unui experiment cu spațiul de selecție S, există un număr, numit probabilitatea de apariție a evenimentului A, $P(A)$, care verifică următoarele trei axiome:
  - $0 \leq P(A) \leq 1$
  - $P(S) = 1$
  - $P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} P(A_i)$, $\forall n$, $A_i \cap A_j = \phi$ pentru evenimente care se exclud reciproc.

$$\Rightarrow 1 = P(S) = P(A \cup A^c) = P(A) + P(A^c)$$

Probabilitate condiționată și Independență

- **Probabilitate condiționată**$= P(A|B) = \frac{P(A \cap B)}{P\{B\}}$
- **Evenimente independente**$= P(A \cap B) = P(A)P(B)$

Variabile aleatoare

- **Variabilă aleatoare** $X : S \to \mathbb{R}$ (discrete și continue)
- **Funcție de repartiție** $F(x) = P\{X \leq x\}$
- **Funcție și densitate de probabilitate**
  - $p(x) = P\{X = x\}$; $F(a) = \sum_{x \leq a} p(x)$
  - $f(x) = F'(x)$, $P\{X \in C\} = \int_C f(x)dx$, $F(a) = \int_{-\infty}^{a} f(x)dx$
- **Două variabile aleatoare** $F(x, y) = P\{X \leq x, Y \leq y\}$,
  $p(x, y) = P\{X = x, Y = y\}$,
  $P\{X \in C, Y \in D\} = \int \int_{x \in C, y \in D} f(x, y)dxdy$

Numere aleatoare

- **Șir de numere aleatoare**: (definiție intuitivă) un sir de numere alese la întâmplare astfel încât se cunoaște probabilitatea de apariție a fiecărui număr într-o succesiune de valori dată.
- Șirurile de numere aleatoare au aplicații în: criptografie, simulare, etc.
- În general pentru șirurile de numere aleatoare probabilitatea de apariție a unei valori corespunde **repartiției uniforme**.
- **Repartiția uniformă:** (intuitiv) Toate valorile sunt egal probabile.
- În **simulare:** Șiruri de numere aleatoare $\Rightarrow$ Valori ale variabilelor aleatoare $\Rightarrow$ Model de simulare

- *Numere aleatoare:* Valori ale unor variabile aleatoare uniforme pe $[0, 1]$.
- Șirurile de numere aleatoare pot fi obținute din:
  - tabele - greu de implementat;
  - fenomene fizice (cea mai buna sursă de aleator, de exemplu: intervalele de timp dintre apăsarea unei taste și mișcarea mouse-ului) - nu se pot refolosi;
  - algoritmi.

Algoritmi de numere aleatoare

- Se bazează pe utilizarea unei **valori inițiale (sămâmnță)** și a unei **relații de recurență** cu ajutorul căreia se obțin celelalte valori ale șirului.
- Numere **pseudo-aleatoare**: numerele obținute nu sunt chiar aleatoare pentru că se bazează pe o relație de recurență. Numerele produse trebuie să aibă două proprietăți statistice importante:
  - uniformitate;
  - independență.
- Majoritatea algoritmilor generează X_n numere întregi între 0 și $m-1$ (de obicei $m-1$ este valoarea maximă a tipului întreg memorat în calculator) și apoi se iau:

$$U_n = \frac{X_n}{m}$$

uniforme pe $[0,1]$.

Algoritmi de numere aleatoare

Trebuie să aibă următoarele **proprietăți**:

- Rapiditate;
- Portabilitate de diverse calculatoare;
- Șirul de numere produs
  - să aibă o perioadă mare;
  - să fie cât mai aproape de independență și uniformitate;
  - să fie reproductibil.

Metoda părții din mijloc a pătratului

- Prima metodă propusă pentru a fi implementată pe calculatoare. A fost descrisă de John von Neumann în 1946.
- Are doar interes istoric.
- Presupunem că vrem să generăm numere aleatoare cu cel mult i cifre. Atunci:

$$X_{n+1} = \text{cele } i \text{ cifre din mijloc ale lui } X_n^2$$

Algoritmi de numere aleatoare

- Șirul tinde să se stabilizeze în scurte cicluri de elemente.
- De exemplu: 43, 84, 05, 02, 00, 00,.... (se pun 0-uri în fața numerelor care nu au patru sau două cifre).

Metoda Fibonacci

- Doar interes istoric;
- Se bazează pe relația

$$X_{n+1} = (X_n + X_{n-1}) \mod m;$$

- numerele produse nu sunt destul de aleatoare.

Alte metode:

- metoda regiștrilor de translație (shift register), metode combinate.

Metoda congruențială liniară

- Este folosit cel mai frecvent.
- Se bazează pe relația:

$$X_{n+1} = (aX_n + c) \mod m \qquad (0.1)$$

  unde
  - $m > 0 =$ modulul;
  - $a =$ multiplicatorul;
  - $c =$ incrementul;
  - $X_0 =$ termenul inițial.
- Fie $b = a - 1$. Se presupune $a \geq 2$ și $b \geq 1$, pentru că pentru $a = 0$ și $a = 1$ nu se obțin șiruri aleatoare.
- Din (1), pentru $k \geq 0$ și $n \geq 0$, rezultă că:

$$X_{n+k} = (aX_{n+k-1} + c) \mod m$$

  și prin urmare relația dintre termenii șirului aflați la distanta k este:

$$
\begin{aligned}
X_{n+k} &= [a(aX_{n+k-2} + c) \mod m + c] \mod m \\
&= [a^2 X_{n+k-2} + (a+1)c] \mod m \\
&= [a^3 X_{n+k-3} + (a^2 + a + 1)c] \mod m \\
&= .... \\
&= [a^k X_n + (a^{k-1} + a^{k-2} + ... + a + 1)c] \mod m \\
&= \left[a^k X_n + \frac{a^k - 1}{a - 1} c\right] \mod m
\end{aligned}
$$

relație utila pentru alegerea valorilor care caracterizează șirul.

Alegerea modulului

- m: trebuie să fie suficient de mare și să asigure o complexitate scăzută a calculului.
- o alegere convenabilă a lui m ar fi $w =$ dimensiunea cuvântului calculatorului.
- alte alegeri: $m = w \pm 1$ sau $m =$ cel mai mare număr prim mai mic decât w.

Alegerea multiplicatorului

Se alege a pentru un m oarecare astfel încât pentru orice valoare a lui X_0 să rezulte un generator de perioadă maximă.

Teoremă

Șirul congruențial liniar definit de m, a, c, și X_0 are perioada de lungime maximă m dacă și numai dacă:

1. *c și m sunt două numere întregi prime între ele;*
2. *$b = a - 1$ este un multiplu de p, pentru orice număr p care-l divide pe m.*
3. *b este multiplu de 4 dacă m este multiplu de 4.*

Datorită următoarei leme este suficientă demonstrarea teoremei pentru m putere a unui număr prim.

Lemă

Fie descompunerea lui m în factori primi:

$$m = p_1^{e_1}...p_t^{e_t} \tag{0.2}$$

lungimea λ a perioadei șirului congruențial liniar definit de (X_0, a, c, m) este cel mai mic multiplu comun al lungimilor λ_j ale perioadelor șirurilor congruențiale liniare $(X_0 \mod p_j^{e_j}, a \mod p_j^{e_j}, c \mod p_j^{e_j}, p_j^{e_j})$, $1 \leq j \leq t$.

De aici rezultă că:

$$p_1^{e_1}...p_t^{e_t} = \lambda = \text{c.m.m.m.c}\{\lambda_1, ..., \lambda_t\} \leq p_1^{e_1}...p_t^{e_t} \tag{0.3}$$

iar această relație poate avea loc dacă și numai dacă $\lambda_j = p_j^{e_j}$ pentru $\forall j$, $1 \leq j \leq t$. De aceea se poate presupune $m = p^e$, unde p este un număr prim iar e este un număr întreg pozitiv.

Perioada poate avea lungime m dacă și numai dacă orice număr întreg din $[0, m)$ apare în cadrul perioadei o singură dată. Dacă luăm $X_0 = 0$, atunci:

$$X_n = \left(\frac{a^n - 1}{a - 1}\right) c \mod m$$

Dacă c și m nu sunt prime între ele, atunci în acest șir nu poate exista 1. Prin urmare condiția 1. din teoremă este necesară. Demonstrarea teoremei se reduce la demonstrarea următoarei leme:

Lemă

Presupunem că $1 < a < p^e$*, cu* p *număr prim. Dacă* λ *este cel mai mic număr întreg pozitiv pentru care*

$$\frac{a^\lambda - 1}{a - 1} \equiv 0 \mod p^e$$

atunci

$$\lambda = p^e$$

dacă și numai dacă:

- *pentru* $p = 2$ $a \equiv 1 \mod 4$*;*
- *pentru* $p > 2$ $a \equiv 1 \mod p$*.*

Această lemă se demonstrează aplicând de mai multe ori următoarea lemă:

Lemă

Fie p un număr prim și fie e un număr întreg pozitiv cu $p^e > 2$. Dacă

$$x \equiv 1 \pmod{p^e}, \;\; x \not\equiv 1 \pmod{p^{e+1}} \tag{0.4}$$

atunci

$$x^p \equiv 1 \pmod{p^{e+1}}, \;\; x \not\equiv 1 \pmod{p^{e+2}} \tag{0.5}$$

Generatorul multiplicativ congruențial

Este un generator liniar congruențial cu $c = 0$:

$$X_{n+1} = aX_n \mod m \tag{0.6}$$

Observăm că X_n și m trebuie să fie prime între ele, pentru că altfel generatorul ar deveni un șir de 0. Prin urmare lungimea perioadei poate fi maxim $\varphi(m)$, numărul numerelor întregi cuprinse între 0 și m, prime cu m.

Putem să presupunem din nou că $m = p^e$ cu $p =$ nr. prim și e întreg pozitiv. Avem:

$$X_n = a^n X_0 \mod p^e$$

Dacă a este multiplu de p, atunci perioada are lungime $1 \Rightarrow a$ trebuie să fie prim cu p.

Generatorul Mersenne Twister

- http://www.math.sci.hiroshima-u.ac.jp/ m-mat/MT/emt.html
- dezvoltat în 1997 de Makoto Matsumoto și Takiji Nishimura
- există varianta pe 32 de biți și pe 64 de biți
- folosit în Python, Matlab, R
- Un generator de numere aleatoare foarte rapid
- Are perioada $2^{19937} - 1$
- este potrivit pentru simulările Monte-Carlo, nu este potrivit pentru criptografie
- folosește numerele prime Mersenne

Testarea șirurilor de numere aleatoare se face cu **teste statistice**:

- Teste de frecvență (testează repartiția uniformă pe care trebuie să o aibă numerele): testul χ^2, testul Kolmogorov-Smirnov.
- Teste de independență.