Generarea variabilelor neuniforme

Curs 7

December 9, 2020

Generarea unor variabile discrete

1. Probe Bernoulli

Fie A un eveniment aleator observabil cu probabilitatea de realizare constantă $p = P(A) > 0$. În urma unui experiment fie se realizează A, fie se realizează evenimentul contrar lui A, $\bar{A}$. Un astfel de experiment se numește *probă Bernoulli*. Presupunem că atunci când se realizează A are loc un *succes*, iar când se realizează $\bar{A}$ are loc un *eșec*.

Asociem unei probe Bernoulli variabila aleatoare Z a.î $Z = 1$ în caz de succes și $Z = 0$ în caz de eșec:

$$Z : \begin{pmatrix} 0 & 1 \\ q & p \end{pmatrix} \tag{1}$$

unde $q = 1 - p$. Observăm că

$$E[Z] = p, \quad Var[Z] = pq = p(1-p).$$

Funcția de repartiție a variabilei Z este:

$$F(x) = \begin{cases} 0 \text{ adcă } x < 0 \\ q \text{ adcă } x \in [0,1] \\ 1 \text{ adcă } x > 1 \end{cases}$$

De aici rezultă următorul algoritm de generare a variabilei aleatoare Z prin metoda inversă, cazul discret.

Algoritm Bernoulli

Intrare: p

`P1:` `Se generează` $U \sim U(0,1)$;

`P2:` `Dacă` $U \leq q$ `atunci` $Z = 0$, `altfel` $Z = 1$;

Ieșire: `Variabila aleatoare` Z.

2. Repartiția binomială

Fie X o variabilă aleatoare discretă, $n \in \mathbb{N}^*$, $p \in [0,1]$. Atunci X are o repartiție binomială $Binom(n,p)$ dacă

$$X = \text{nr. de succese în } n \text{ probe Bernoulli independente}$$

Observăm că:

$$X = \sum_{i=1}^{n} Z_i, \tag{2}$$

unde Z_i, $i = 1, ..., n$ sunt variabile Bernoulli independente.

Observăm că:

$$P(X = x) = \binom{n}{x} p^x q^{n-x}$$

Adică $P(X = x)$ este termenul general al dezvoltării binomului $(p + q)^n$ de unde derivă și denumirea de repartiție binomială.

Funcția caracteristică a variabilei binomiale este:

$$\varphi_X(t) = E[e^{itX}] = E[e^{it\sum_{j=1}^{n} Z_j}] = (q + pe^{it})^n.$$

de unde rezultă că:

$$E[X] = np, \quad Var[X] = npq$$

Din (2) rezultă următorul algoritm de generare a variabilei X:

Algoritm Binom1
Intrare: n, p
`P1:` $i = 1$, $X = 0$;
`P2: Se generează` $Z_i \sim Bern(p)$, $X := X + Z_i$;
`P3: Dacă` $i = n$ `stop, altfel` $i := i + 1$, `mergi la P2;`
Ieșire: `Variabila aleatoare` X.

O altă metodă de generare a variabilei binomiale rezultă din *teorema limită centrală*: pentru $n \to \infty$ variabila:

$$W_n = \frac{X - np}{\sqrt{npq}}$$

este repartizată normal $N(0,1)$. De aici se deduce următorul algoritm pentru n mare:

Algoritm Binom2

Intrare: n, p

`P1:` `Se generează` $W \sim N(0,1)$;

`P2:` `Se ia` $X =$ `cel mai apropiat nr. întreg de` $np + W\sqrt{npq}$;

Ieșire: `Variabila aleatoare` X.

3. Repartiția Pascal

Fie X o variabilă aleatoare discretă, $k \in \mathbb{N}^*$, $p \in [0,1]$. Atunci X are repartiția $Pascal(k,p)$ dacă:

$X =$ nr. de eșecuri până la apariția a k succese dintr-un șir oarecare de probe Bernoulli independente.

Observație

$$P(X = x) = \binom{x+k-1}{k-1} p^k q^x$$

este termenul general al dezvoltării în serie al expresiei $p^k(1-q)^{-k}$ *(de aceea repartiția Pascal se mai numește și repartiția binomială cu exponent negativ).*

Dem:

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + ... + (-1)^k x^k + ...$$

$$\left(\frac{1}{1+x}\right)^{(r)} = \sum_{k=r}^{\infty}(-1)^k k(k-1)...(k-r+1)x^{k-r} =$$

$$= \sum_{j=0}^{\infty}(-1)^{r+j}\frac{(r+j)!}{j!}x^j \tag{3}$$

Dar

$$\left(\frac{1}{1+x}\right)^{(r)} = \frac{(-1)^r r!}{(1+x)^{r+1}} \tag{4}$$

Din (3) și (4) rezultă că:

$$\frac{(-1)^r r!}{(1+x)^{r+1}} = \sum_{j=0}^{\infty}(-1)^{r+j}\frac{(r+j)!}{j!}x^j$$

Notând $r+1$ cu n avem:

$$(1+x)^{-n} = \sum_{j=0}^{\infty} (-1)^j \binom{n+j-1}{j} x^j \qquad (5)$$

Notând

$$(-1)^j \binom{n+j-1}{j} = \binom{-n}{j}$$

avem următoarea dezvoltare în serie a binomului cu exponent întreg negativ:

$$(1+x)^{-n} = \sum_{j=0}^{\infty} \binom{-n}{j} x^j.$$

Acum aplicăm (5) pentru $p^k(1-q)^{-k}$:

$$p^k(1-q)^{-k} = p^k \sum_{j=0}^{\infty} (-1)^j \binom{k+j-1}{j} (-q)^j = \sum_{j=0}^{\infty} \binom{k+j-1}{j} p^k q^j.$$

Observăm că termenul j al acestei dezvoltări în serie este tocmai funcția de probabilitate a unei variabile $Pascal(k,p)$, calculată în punctul j.

Repartiția *Pascal* este stabilă și:

$$E[X] = \frac{kq}{p}, \quad Var[X] = \frac{kq}{p^2}$$

Din definiție rezultă următorul algoritm de generare al unei variabile *Pascal*:

Algoritm Pascal

Intrare: k, p, `j=0`.

P1: Se generează $Y \sim Bernoulli(p)$;

P2: Dacă $Y = 0$ atunci $X := X + 1$, altfel $j := j + 1$;

P3: Dacă j=k stop, altfel mergi la P1.

Ieșire: Variabila aleatoare X.

4. Repartiția geometrică

Este un caz particular de repariție *Pascal*: $k = 1$. Atunci funcția de probabilitate devine pentru $k = 1$:

$$P(X = x) = pq^x, \quad x = 0, 1, 2, ... \tag{6}$$

care este termenul unei progresii geometrice (de aici provine și denumirea acestei distribuții). Observăm că:

$$F(x) = P(X \leq x) = \sum_{i=0}^{x} pq^i = 1 - q^{x+1}, x = 0, 1, 2, ...$$

și

$$E[X] = \frac{q}{p}, \quad Var[X] = \frac{q}{p^2}.$$

Simularea variabilei geometrice se poate realiza fie prin aplicarea algoritmului de generare a variabilei Pascal, fie prin metoda inversă astfel:

Algoritm Geometrică

Intrare: $p, \; q = 1 - p$;

P1: Se generează $U \sim U(0, 1)$;

P2: $X := \left[\frac{\log U}{\log q}\right] - 1$.

Ieșire: Variabila aleatoare X.

unde $[a]$ reprezintă partea întreagă a lui a.

5. Repartiția hipergeometrică

Considerăm următorul experiment aleator: fie o urnă care conține A bile albe și B bile negre, cu $A + B = N$. Presupunem că se realizează n extrageri, fără întoarcerea bilei extrase în urnă. Atunci:

$$X = \text{ numărul de bile albe extrase}$$

este o variabilă *hipergeometrică*.

Fie E_A evenimentul care constă în extragerea unei bile albe, iar E_B evenimentul care constă în extragerea unei bile negre. Atunci la prima extragere avem:

$$p = P(E_A) = \frac{A}{N}, \quad P(E_B) = \frac{B}{N}$$

Probabilitățile de extragere a unei bile albe sau negre la a doua extragere sunt condiționate de rezultatele primei extrageri:

$$P(E_A|E_A) = \frac{A-1}{N-1}, \quad P(E_A|E_B) = \frac{A}{N-1}$$

$$P(E_B|E_A) = \frac{B}{N-1}, \quad P(E_B|E_B) = \frac{B-1}{N-1}$$

La fiecare extragere compoziția urnei se modifică și probabilitatea de a extrage o bilă albă sau neagră depinde de extragerile anterioare.

Variabila hipergeometrică se notează cu $H(N, p, n)$ cu $0 < p < 1$, $n < N$. Atunci A poate fi considerat cel mai apropiat nr. întreg de Np, $B = N - A$. Probabilitatea ca în extrageri succesive fără întoarcere să se extragă x bile albe este:

$$P(X = x) = \frac{\binom{A}{x}\binom{B}{n-x}}{\binom{N}{n}}, \quad 0 \leq a \leq n, \quad n < N. \tag{7}$$

Media și dispersia variabilei hipergeometrice sunt:

$$E[X] = np, \quad Var[X] = np(1-p)\frac{N-n}{N-1}$$

Din definiție rezultă următorul algoritm de generare a variabilei hipergeometrice:

Algoritm Hipergeometrică

Intrare: $A,\ B,\ n,\ N = A + B,\ p = A/N,\ j := 0,\ X := 0;$

`P1:` `Se generează` $U \sim U(0,1)$;

`P2:` `Dacă` $U < p$ `(s-a extras o bilă albă) atunci mergi la P3, altfel (s-a extras o bilă neagră) mergi la P4;`

`P3:` $X := X + 1,\ S := 1,$ `mergi la P5;`

`P4:` $S = 0$; `P5:` $N := N - 1,\ A := A - S,\ p := \frac{A}{N}$;

`P6:` `Dacă` $j = n$ `stop, altfel mergi la P1.`

Ieșire: `Variabila aleatoare` X.

6. Repartiția Poisson

Variabila aleatoare discretă $X \in \mathbb{N}$ are repartiția $Poisson(\lambda)$, $\lambda > 0$, dacă:

$$P(X = x) = \frac{\lambda^x}{x!} e^{-\lambda} \tag{8}$$

Funcția caracteristică a variabilei $Poisson(\lambda)$ este:

$$\varphi_X(t) = e^{\lambda(e^{it} - 1)}$$

De aici rezultă că:

$$E[X] = \lambda, \quad Var[X] = \lambda.$$

Repartiția $Poisson(\lambda)$ este folosită pentru a modela numărul de evenimente care apar într-un interval de timp. Parametrul λ reprezintă numărul mediu de evenimente dintr-un anumit interval de timp.

Lemă

Dacă $E_1, E_2, \ldots$ sunt variabile aleatoare repartizate $Exp(1)$ și X este cel mai mic întreg astfel încât

$$\sum_{i=1}^{X+1} E_i > \lambda \tag{9}$$

atunci variabila X este repartizată $Poisson(\lambda)$.

Dem:

Din condiția care se pune asupra lui X rezultă că dacă $X = k$, atunci are loc $\sum_{i=1}^{k+1} E_i > \lambda$ și nu are loc $\sum_{i=1}^{k} E_i > \lambda$. Prin urmare:

$$P(X = k) = P(\sum_{i=1}^{k+1} E_i > \lambda) - P(\sum_{i=1}^{k} E_i > \lambda) = \int_{\lambda}^{\infty} f_{k+1}(y)dy - \int_{\lambda}^{\infty} f_k(y)$$

unde $f_k(y)$ este densitatea unei variabile aleatoare $Erlang(k)$.

Rezultă că:

$$P(X=k)=\int_{\lambda}^{\infty}\left[\frac{1}{k!}y^k e^{-y}-\frac{1}{(k-1)!}y^{k-1}e^{-y}\right]dy=$$

$$=\frac{1}{k!}\left[\int_{\lambda}^{\infty}y^k e^{-y}dy-k\int_{\lambda}^{\infty}y^{k-1}e^{-y}dy\right].$$

Dacă integrăm prin părți prima integrală obținem:

$$P(X=k)=\frac{1}{k!}\left[\lambda^k e^{-\lambda}+k\int_{\lambda}^{\infty}y^{k-1}e^{-y}dy-k\int_{\lambda}^{\infty}y^{k-1}e^{-y}dy\right]$$

$$=\frac{\lambda^k}{k!}e^{-\lambda}$$

Ținând cont că variabilele E_i sunt exponențiale $Exp(1)$ și ele pot fi generate prin metoda inversă cu ajutorul relației $E_i = -\log U_i$, unde $U_i \sim U(0,1)$, atunci condiția (9) se scrie:

$$\prod_{i=1}^{X+1} U_i < e^{-\lambda}$$

și rezultă următorul algoritm de generare a unei variabile $Poisson(\lambda)$:

Algoritm Poisson1

Intrare: $\lambda, \; i := 0, \; P = 1;$

`P1:` `Se generează` $U \sim U(0,1), \; i := i + 1, \; P := P * U;$

`P2:` `Dacă` $P \geq e^{-\lambda}$ `atunci mergi la P1, altfel mergi la P3;`

`P3:` $X := i - 1.$

Ieșire: `Variabila aleatoare` X.

O altă metodă de generare pentru o variabilă $Poisson(\lambda)$ se bazează pe următoarea proprietate a variabilei $Poisson(\lambda)$:

dacă $Y \sim Poisson(\lambda)$ cu $\lambda = np$ și $n \to \infty$, $p \to 0$, atunci $Y \sim Binom(n, p)$.

Acest lucru se poate demonstra folosind funcția caracteristică a celor două repartiții.

Algoritm Poisson2

Intrare: λ, se alege $p \approx 0$, de exemplu $p = 0.001$;

P1: Se determină n= cel mai apropiat întreg de λ/p;

P2: Se generează $X \sim Binom(n, p)$;

Ieșire: Variabila aleatoare X.