

Véletlenszám-generátorok

3. rész

– egy MATLAB[®] alapú megközelítés –

Róth Ágoston, Vas Orsolya

Matematika és Informatika Intézet, Babeş-Bolyai Tudományegyetem, Kolozsvár, Románia

(agoston.roth@gmail.com, vas.orsolya@yahoo.com)

3. labor / 2018. október 15–18.



Tétel (Az inverziós módszer)

- Legyen

$$F^{-1}(u) = \inf \{x \in \Omega : F(x) = u, 0 < u < 1\}$$

egy, az (Ω, \mathcal{A}, P) valószínűségi mezőn értelmezett, folytonos valószínűségi változó F eloszlásfüggvényének az inverze.

- Ha az U valószínűségi változó egyenletes eloszlású a $[0, 1]$ intervallumon, akkor az $F^{-1}(U)$ valószínűségi változó eloszlásfüggvénye éppen F .
- Fordítva, ha X eloszlásfüggvénye F , akkor az $F(X)$ valószínűségi változó egyenletes eloszlású a $[0, 1]$ intervallumon.



Bizonyítás

- A direkt állítás bizonyításához vegyük észre, hogy bármely $x \in \mathbb{R}$ esetén

$$\begin{aligned} P(F^{-1}(u) \leq x) &= P(\inf \{y \in \Omega : F(y) = U\} \leq x) \\ &= P(U \leq F(x)) \\ &= F(x)! \end{aligned}$$

- A fordított állítást minden $0 < u < 1$ esetén a

$$\begin{aligned} P(F(X) \leq u) &= P(X \leq F^{-1}(u)) \\ &= F(F^{-1}(u)) \\ &= u \end{aligned}$$

módon láthatjuk be.



- Az inverziós módszer tételét tetszőleges F eloszlásfüggvényű folytonos X valószínűségi változó értékeinek generálására használhatjuk, feltéve, hogy az F^{-1} inverz függvény explicit alakját ismerjük.
- Minél gyorsabban határozzuk meg az inverz értékeket, annál gyorsabban generálhatjuk az X valószínűségi változó értékeit egy egyenletes eloszlású U valószínűségi változó segítségével. Pszeudokóddal a következő algoritmust fogalmazhatjuk meg.

1. Algoritmus (Inverziós módszer)

Bemenet: egy folytonos X valószínűségi változó eloszlásfüggvényének F^{-1} inverze.

Kimenet: az X változónak egy n -elemű független $\{X_i\}_{i=1}^n$ mintavétele.

- 1 Minden $i = 1, 2, \dots, n$ index esetén **végezd el:**
- 2 generáld az $U \sim \mathcal{U}([0, 1])$ változót
- 3 számítsd ki az $X_i = F^{-1}(U)$ inverz értéket.



Példák

$f_X(x)$	$F_X(x)$	$X = F_X^{-1}(U)$	Egyszerűsített alak
$\lambda e^{-\lambda x}, x \geq 0, \lambda > 0$	$1 - e^{-\lambda x}$	$-\frac{1}{\lambda} \ln(1 - U)$	$-\frac{1}{\lambda} \ln(U)$
$\frac{\sigma}{\pi(x^2 + \sigma^2)}, \sigma > 0$	$\frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{\sigma}\right)$	$\sigma \tan\left(\pi\left(U - \frac{1}{2}\right)\right)$	$\sigma \tan(\pi U)$
$\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, x \geq 0, \sigma > 0$	$1 - e^{-\frac{x^2}{2\sigma^2}}$	$\sigma \sqrt{-2 \ln(1 - U)}$	$\sigma \sqrt{-2 \ln(U)}$
$\frac{2}{a} \left(1 - \frac{x}{a}\right), 0 \leq x \leq a$	$\frac{2}{a} \left(x - \frac{x^2}{2a}\right)$	$a(1 - \sqrt{1 - U})$	$a(1 - \sqrt{U})$
$x e^{\frac{a^2 - x^2}{2}}, x \geq a > 0$	$1 - e^{\frac{a^2 - x^2}{2}}$	$\sqrt{a^2 - 2 \ln(1 - U)}$	$\sqrt{a^2 - 2 \ln(U)}$
$\frac{ab^a}{x^{a+1}}, x \geq b > 0, a > 0$	$1 - \left(\frac{b}{x}\right)^a$	$\frac{b}{(1 - U)^{\frac{1}{a}}}$	$\frac{b}{U^{\frac{1}{a}}}$

táblázat. Inverziós módszer alkalmazása rendre a $\lambda > 0$ paraméterű exponenciális, a $\sigma > 0$ paraméterű Cauchy, a $\sigma > 0$ paraméterű Rayleigh, a $[0, a]$ intervallum feletti „háromszögű”, az $a > 0$ paraméterű Rayleigh-féle „vég”, illetve az $a, b > 0$ paraméterű Pareto-féle valószínűségi változók esetén. Mindegyik esetben az U valószínűségi változó egyenletes eloszlású a $(0, 1)$ intervallumon, az X pedig az f_X sűrűség- és F_X eloszlásfüggvényű, mintavételezendő valószínűségi változót jelöli.

- A bemutatott inverziós módszer az egyetlen igazán univerzális módszer nemegyenletes eloszlású folytonos változók generálására: akkor is használható, ha nem ismerjük az adott valószínűségi változó eloszlásfüggvénye inverzének explicit alakját, ekkor viszont az inverz értékek meghatározását csak **numerikus módszerekkel** végezhetjük, amelyek lényegesen lassíthatják az algoritmus futási idejét.
- Ha az inverziós tételbeli X folytonos valószínűségi változó eloszlásfüggvényének F^{-1} inverzét nem ismerjük, akkor az $F(X) = U$ egyenletet numerikusan kell megoldanunk, amely végtelen futási időt követel meg. Bármilyen leállási feltételt is szabunk, egy inegzakt algoritmust kapunk. A következő részekben három népszerű numerikus inverziós algoritmust és leállási feltételt írunk le.
- Ha a futási idő fontos szempont, akkor más módszerekhez kell folyamodnunk, de ezek az adott valószínűségi változó eloszlásfüggvényének értékei mellett egyéb információkat is megkövetelnek.



2. Algoritmus (Felező módszer)

- **Bemenet:** az F eloszlásfüggvényű folytonos X valószínűségi változó, valamint az $n \geq 1$ természetes szám és a $\delta > 0$ küszöbhatár, amelyek a kimeneti mintavétel nagyságát, illetve pontosságát határozzák meg.
- **Kimenet:** az adott folytonos valószínűségi változónak egy n -elemű független $\{X_i\}_{i=1}^n$ mintavétele.

① Minden $i = 1, 2, \dots, n$ index esetén **végezd el:**

② generáld az $U \sim \mathcal{U}([0, 1])$ valószínűségi változót

③ határozd meg az $[a, b]$ intervallumot úgy, hogy tartalmazza az ismeretlen X_i inverz értéket

④ **ismételd:**

⑤
$$X_i \leftarrow \frac{a + b}{2}$$

⑥ **ha** $F(X_i) \leq U$

⑦ **akkor** $a \leftarrow X_i$

⑧ **máskülönben** $b \leftarrow X_i$

⑨ **ameddig** $b - a \leq 2\delta$.

3. Algoritmus (Húr módszer)

- **Bemenet:** az F eloszlásfüggvényű folytonos X valószínűségi változó, valamint az $n \geq 1$ természetes szám és a $\delta > 0$ küszöbhatár, amelyek a kimeneti mintavétel elemszámát, illetve pontosságát határozzák meg.
- **Kimenet:** az adott folytonos valószínűségi változónak egy n -elemű független $\{X_i\}_{i=1}^n$ mintavétele.

1 Minden $i = 1, 2, \dots, n$ index esetén **végezd el:**

2 generáld az $U \sim \mathcal{U}([0, 1])$ valószínűségi változót

3 határozd meg az $[a, b]$ intervallumot úgy, hogy tartalmazza az X_i ismeretlen inverz értéket

4 ismételd:

5
$$X_i \leftarrow a + (b - a) \frac{U - F(a)}{F(b) - F(a)}$$

6 ha $F(X_i) \leq U$

7 akkor $a \leftarrow X_i$

8 máskülönben $b \leftarrow X_i$

9 ameddig $b - a \leq \delta$.

4. Algoritmus (Newton–Raphson módszer)

- **Bemenet:** az f sűrűség- és F eloszlásfüggvényű folytonos X valószínűségi változó, valamint az $n \geq 1$ természetes szám és a $\delta > 0$ küszöbhatár, amelyek a kimeneti mintavétel méretét, illetve pontosságát határozzák meg.
- **Kimenet:** az adott folytonos valószínűségi változónak egy n -elemű független $\{X_i\}_{i=1}^n$ mintavétele.

- 1 Minden $i = 1, 2, \dots, n$ index esetén **végezd el:**
- 2 generáld az $U \sim \mathcal{U}([0, 1])$ valószínűségi változót
- 3 adj tetszőleges kezdőértéket az ismeretlen X_i inverz értéknek
- 4 **ismételd:**
- 5
$$X_i \leftarrow X_i - \frac{F(X_i) - U}{f(X_i)}$$
- 6 **ameddig** $|F(X_i) - U| \leq \delta$.

- A 7. és a 8. algoritmusok egy $[a, b]$ intervallumot követelnek meg az inverz értékek előállításához. Ha a felhasználó ismer olyan G és H függvényeket, amelyek esetén $G(x) \geq F(x) \geq H(x)$ az $x \in \mathbb{R}$ paraméter bármely értékére, valamint a G^{-1} és H^{-1} inverz függvények könnyen kiértékelhetők, akkor adott $u \in [0, 1]$ számhoz tartozó ismeretlen $x = F^{-1}(u)$ inverz érték kiszámításához kiindulhatunk az

$$[a, b] = [G^{-1}(u), H^{-1}(u)]$$

intervallumból. Sajátosan, ha ismerjük az X folytonos valószínűségi változó F eloszlásfüggvényének az értelmezési tartományát, akkor az $[a, b]$ kiindulási intervallumot erre állíthatjuk.



- Mivel fontos, hogy minél kisebb intervallumokban keressük az $F(x) = u \in [0, 1]$ egyenlet megoldását, minden előzetesen ismert információt fel kell használnunk az $[a, b]$ intervallum beállításához. Például, ha az f sűrűségfüggvényű X valószínűségi változó szórásnégyzete σ^2 és várható értéke 0 (pl. f szimmetrikus az $x = 0$ pontra nézve), akkor a Chebyshev-egyenlőtlenség Cantelli-féle kiterjesztése folytán az

$$F(x) \geq \frac{x^2}{x^2 + \sigma^2}, \forall x > 0, \sigma > 0$$

feltételt kapjuk, ahol F az X eloszlásfüggvényét jelöli. Ez azt sugallja, hogy ha $u > \frac{1}{2}$, akkor kiindulásként az $[a, b] = \left[0, \sigma \sqrt{\frac{u}{1-u}}\right]$ intervallumot használhatjuk.

Szimmetria miatt, ha $u \leq \frac{1}{2}$, akkor az $[a, b] = \left[-\sigma \sqrt{\frac{1-u}{u}}, 0\right]$

kezdőintervallummal dolgozhatunk. Mint látható, az X valószínűségi változót jellemző bármilyen előzetes információ (pl. első- és magasabb rendű momentum, kvantilis) értékesnek bizonyulhat a kezdeti találgatásokhoz.



- A Newton–Raphson-módszerre épülő 9. algoritmus esetén az $u \in [0, 1]$ számhoz tartozó ismeretlen inverz érték kezdeti találgatásakor gyakran kiindulhatunk az $x = 0$ paraméterértékből.
- Egyedül a felezéses módszeren alapuló 7. algoritmus konvergál minden esetben. Ha az

$$F(x) = u \in [0, 1]$$

egyenletnek egyértelmű megoldása van, akkor a húmmódszert ismertető 8. algoritmus is konvergens. A 9. Newton–Raphson-féle algoritmus csak akkor konvergál, ha az X folytonos valószínűségi változó F eloszlásfüggvénye konkáv, vagy konvex. Gyakran az X valószínűségi változó f sűrűségfüggvénye egyetlen szélsőértékkel rendelkezik egy bizonyos $x = m$ értéknél, ami egyben maximumpont is. Ekkor az F eloszlásfüggvény konvex a $(-\infty, m]$ intervallumon, illetve konkáv az $[m, +\infty)$ tartományon, valamint az $x = m$ kezdőértékkel inicializált Newton–Raphson-módszer konvergál a fenti egyenlet megoldásához. Amennyiben az f sűrűségfüggvény ismeretlen, a 9. Newton–Raphson-algoritmust el kell kerülni, mert az

$$f(x) \approx \frac{1}{\delta} (F(x + \delta) - F(x)), \delta \searrow 0$$

közelítés általában nagyon pontatlanná válik.



- A folytonos valószínűségi változók esetén bemutatott inverziós módszer átalakítható diszkrét valószínűségi változók mintavételezésére is.
- Ennek érdekében tekintsük az

$$X \left(\begin{array}{c} x_i \\ p_i \end{array} \right)_{i \in I} \quad (1)$$

táblázattal leírt diszkrét valószínűségi változót, ahol az I rendezett indexhalmaz vagy véges ($I = \{0, 1, \dots, n\}$), vagy megszámlálhatóan végtelen ($I = \mathbb{N}$).

- Ekkor a $[0, 1]$ intervallumon egyenletes eloszlású U valószínűségi változó által felvett u érték esetén azt az $i \in I$ indexet kell meghatároznunk, amelyre teljesül az

$$F(x_{i-1}) = \sum_{j < i} p_j < u \leq \sum_{j \leq i} p_j = F(x_i) \quad (2)$$

egyenlőtlenség, ahol az F leképezés az X változó eloszlásfüggvényét jelöli és $F(x_{-1}) = 0$.

- Ennek érdekében a legegyszerűbb – (2)-es feltételen alapuló – inverziós módszert a következőképpen fogalmazhatjuk meg diszkrét valószínűségi változók mintavételezésére.



5. Algoritmus (Szekvenciális keresésen alapuló inverziós módszer diszkrét valószínűségi változók mintavételezésére)

- **Bemenet:** az (1)-es alakú X diszkrét valószínűségi változó, valamint az $r \geq 1$ természetes szám, amely a generált mintavétel méretét rögzíti.
- **Kimenet:** az adott diszkrét valószínűségi változónak egy r -elemű független $\{X_k\}_{k=1}^r$ mintavétele.

- 1 Minden $k = 1, 2, \dots, r$ értékre **végezd el:**
- 2 generáld az $U \sim \mathcal{U}([0, 1])$ valószínűségi változót
- 3 $i \leftarrow 0$
- 4 $S \leftarrow p_0$
- 5 **amíg** $U > S$ **végezd el:**
- 6 $i \leftarrow i + 1$
- 7 $S \leftarrow S + p_i$
- 8 $X_k \leftarrow x_i$.

- Jelölje N a szekvenciális keresésen alapuló 5. algoritmus **amíg**-ciklusa által végzett összehasonlítások számát. Ekkor

$$P(N = i) = P(X = x_{i-1}) = p_{i-1}, \quad i \geq 1,$$

amely egyszerű következményeként, várhatóan $E(N) = E(X) + 1$ összehasonlításra számíthatunk egy-egy minta előállításakor. A következő részekben az 5. algoritmus feljavítására írunk le néhány módszert.

- Sajátságos diszkrét valószínűségi változók esetén elkerülhetjük az 5. algoritmus **amíg**-ciklusa által igényelt p_i valószínűségek ismételt kiszámítását, ahogy azt az alábbi $\lambda > 0$ paraméterű Poisson-eloszlású valószínűségi változók mintavételzését végző algoritmus is mutatja.

$$x_i = i, \quad i \geq 0, \quad p_0 = e^{-\lambda}, \quad p_i = \frac{\lambda}{i} p_{i-1}, \quad i \geq 1,$$

amelyek alapján az 5. algoritmust a következőképpen gyorsíthatjuk fel.



6. Algoritmus ($\mathcal{Poisson}(\lambda)$ -eloszlású változók generálása szekvenciális kereséssel)

- **Bemenet:** a $\lambda > 0$ valós paraméter, valamint az $r \geq 1$ természetes szám, amely a generált mintavétel méretét rögzíti.
- **Kimenet:** a $\mathcal{Poisson}(\lambda)$ -eloszlású valószínűségi változónak egy r -elemű független $\{X_k\}_{k=1}^r$ mintavétele.

- 1 Minden $k = 1, 2, \dots, r$ értékre **végezd el:**
- 2 generáld az $U \sim \mathcal{U}([0, 1])$ valószínűségi változót
- 3 $i \leftarrow 0$
- 4 $p \leftarrow e^{-\lambda}$
- 5 $S \leftarrow p$
- 6 **amíg** $U > S$ **végezd el:**
- 7 $i \leftarrow i + 1$
- 8 $p \leftarrow \frac{\lambda}{i} p$
- 9 $S \leftarrow S + p$
- 10 $X_k \leftarrow i.$

- Megemlítjük, hogy a 6. algoritmus esetében egy-egy mintát várhatóan λ összehasonlítás után kapunk, mert ez az érték adja a $Poisson(\lambda)$ -eloszlás elsőrendű momentumát.
- Megszabadulhatunk az 5. algoritmusbeli S változótól is, ahogy azt az alábbi kismértékű változtatás mutatja.

7. Algoritmus (Szekvenciális keresésen alapuló inverziós módszer diszkrét valószínűségi változók mintavételezésére)

- **Bemenet:** az (1)-es alakú X diszkrét valószínűségi változó, valamint az $r \geq 1$ természetes szám, amely a generált mintavétel méretét rögzíti.
- **Kimenet:** az adott diszkrét valószínűségi változónak egy r -elemű független $\{X_k\}_{k=1}^r$ mintavétele.

- 1 Minden $k = 1, 2, \dots, r$ értékre **végezd el:**
- 2 generáld az $U \sim \mathcal{U}([0, 1])$ valószínűségi változót
- 3 $i \leftarrow 0$
- 4 amíg $U > p_i$ **végezd el:**
- 5 $U \leftarrow U - p_i$
- 6 $i \leftarrow i + 1$
- 7 $X_k \leftarrow x_i$.

- Ha az (1)-es alakú X diszkrét valószínűségi változó véges (azaz $I = \{0, 1, \dots, n\}$), akkor előre kiszámolhatjuk és eltárolhatjuk a

$$q_0 = p_0, q_1 = p_0 + p_1, \dots, q_i = \sum_{j=0}^i p_j, \dots, q_n = \sum_{j=0}^n p_n \equiv 1 \quad (3)$$

növekvő kumulatív (másképpen összegzett) gyakoriságokat, majd egy-egy $u \in [0, 1]$ értékhez tartozó minta generálásához azt az $i \in \{0, 1, \dots, n\}$ indexet keressük meg, amelyre teljesül a (2)-es egyenlőtlenséggel ekvivalens

$$q_{i-1} < u \leq q_i \quad (4)$$

feltétel, ahol $q_{-1} = 0$. Ezt az elképzelést az alábbi kódrészletben valósítottuk meg.



1. Kódrészlet. Szekvenciális keresésen alapuló inverziós módszer véges diszkrét valószínűségi változók mintavételezésére

```

1 % _____
2 % Description
3 % _____
4 % The function implements the sequential inversion method1 for finite discrete random
5 % variable sampling.
6 %
7 % _____
8 % Input
9 % _____
10 % X                – a finite discrete random variable given as a matrix in the form
11 %
12 %                
$$X \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ p_1 & p_2 & \cdots & p_n \end{pmatrix}$$

13 %
14 % uniform_rng – specifies the type of the used uniform integer random number generator,
15 % can be set as 'LEcuyer'/'MersenneTwister' or 1/2 (in case of other
16 % values the function it will use the built-in uniform random number
17 % generator of MATLAB®)
18 %
19 % count        – size of the output sample
20 %
21 % _____
22 % Output
23 % _____
24 % sample      – a random sample of the given finite discrete random variable

```



```
25 %  
26 function sample = InversionBySequentialSearch(X, uniform_rng, count)  
27  
28 % input checking  
29 [row_count, column_count] = size(X);  
30  
31 if (row_count ~= 2)  
32     error('Wrong_input_discrete_random_variable!');  
33 end  
34  
35 for j = 1:column_count  
36     if (X(2,j) < 0 || X(2,j) > 1)  
37         error('Wrong_input_discrete_random_variable!');  
38     end  
39 end  
40  
41 total_probability = sum(X(2,:));  
42 if (total_probability > 1)  
43     error('Wrong_input_discrete_random_variable!');  
44 end  
45  
46 % assume accumulated errors  
47 if (total_probability < 1)  
48     X(2, column_count) = 1 - sum(X(2, 1:column_count - 1));  
49 end  
50  
51 % calculate cumulative probabilities  
52 cumulative_probabilities = cumsum(X(2,:));  
53  
54 % allocate memory  
55 sample = zeros(1, count);
```



```
56
57 % perform sequential searching2
58 for k = 1:count
59
60     switch (uniform_rng)
61         case {'LEcuyer', 1}
62             u = ULEcuyerRNG();
63
64         case {'MersenneTwister', 2}
65             u = UMersenneTwisterRNG();
66
67         otherwise
68             u = rand();
69     end
70
71     i = 1;
72     while (u > cumulative_probabilities(i))
73         i = i + 1;
74     end
75
76     sample(k) = X(1, i);
77 end
```



- Nyilván, ha a (3)-as kumulatív valószínűségeket egy kiegyensúlyozott bináris keresőfában tároljuk el előzetesen, akkor a szekvenciális keresés helyett egy sokkal gyorsabb algoritmust ajánlhatunk véges, diszkrét valószínűségi változók mintavételezésére. Ennek megvalósítását kitűzött feladatként az olvasóra bízunk.

¹method módszer; inversion ~ inverziós módszer; rejection ~ elutasításos módszer; squeeze ~ közrefogásos módszer
²to search keresni; sequential ~ing szekvenciális keresés



- Az 1. kódrészletben bevezetett `InversionBySequentialSearch` függvényt használva, többek között szabályos dobókockával történő véletlenszerű dobássorozatot, illetve a visszatevéses binomiális modellen alapuló kísérleteket is végrehajthatunk, ahogy azt a 2. kódrészlet is mutatja.

2. Kódrészlet. Példa az `InversionBySequentialSearch` függvény használatára

```

1 %-----
2 % Dice throwing
3 %-----
4 % define a uniform discrete random variate that represents the dice
5 Dice = [1:6; ones(1, 6) / 6];
6
7 % fix the number of experiments
8 throwing_count = 1000;
9
10 % roll the dice
11 throwing_sequence = InversionBySequentialSearch(Dice, 'LEcuyer', throwing_count);
12
13 % do something nice and meaningful
14 ...
15
16 %-----
17 % Binomial probability model
18 %-----
19 % a box contains only red and blue balls
20 red = 4;

```



```
21 blue = 6;
22
23 % calculate the probability of the appearance of a red ball during a single trial
24 p = red / (red + blue);
25
26 % define a  $Bern(p)$ -distributed random variable
27 Bernoulli = [ 1, 0; p, 1 - p ];
28
29 % fix the number of experiments
30 experiment_count = 1000;
31
32 % fix the number of total draws in case of a single experiment
33 trials = 10;
34
35 % use the fact that the generic binomial distribution  $Bino(n=trials, p)$  is the sum of
36 % independent and identically distributed  $Bern(p)$ -trials
37 Binomial = zeros(1, experiment_count);
38
39 for i = 1:experiment_count
40     Binomial(i) = sum(InverseBySequentialSearch(Bernoulli, 'MersenneTwister', trials))
41 end
42
43 % do something nice and meaningful
44 ...
```



- Egyes esetekben diszkrét eloszlású valószínűségi változókat folytonos változók csonkolásával is előállíthatunk.
- Tegyük fel, hogy valamely folytonos X valószínűségi változó $F_X : [0, \infty) \rightarrow [0, 1]$ eloszlásfüggvénye az egész behelyettesítési értékekre megegyezik egy diszkrét Y valószínűségi változó $F_Y : \mathbb{N} \rightarrow [0, 1]$ eloszlásfüggvénye által felvett értékekkel úgy, hogy teljesülnek az

$$F_X(0) = 0, F_X(i+1) = F_Y(i), i \in \mathbb{N} \quad (5)$$

feltételek. Ekkor az Y diszkrét változó mintavételezésére az alábbi inverziós módszerre épülő algoritmust ajánlhatjuk.



8. Algoritmus (Folytonos valószínűségi változó csonkolásából nyert diszkrét valószínűségi változó)

- **Bemenet:** az (5)-ös feltételt kielégítő X és Y folytonos, illetve diszkrét valószínűségi változók, valamint az $n \geq 1$ természetes szám, amely a kimeneti mintavétel méretét szabja meg.
- **Kimenet:** az Y diszkrét valószínűségi változónak egy n -elemű független $\{Y_i\}_{i=1}^n$ mintavétele.

- 1 Minden $i = 1, 2, \dots, n$ értékre **végezd el:**
- 2 generáld az $U \sim \mathcal{U}([0, 1])$ valószínűségi változót
- 3 $Y_i \leftarrow \left\lfloor F_X^{-1}(U) \right\rfloor$.

- Amennyiben a 8. algoritmusbeli F_X^{-1} inverz függvény explicit alakja ismert, a módszer nagyon gyors.
- Az algoritmus helyességét a

$$P(Y \leq i) = P\left(F_X^{-1}(U) < i + 1\right) = P(U < F_X(i + 1)) = F_X(i + 1) = F_Y(i), \forall i \in \mathbb{N}$$

okfejtéssel láthatjuk be.



Példa

- A 8. algoritmus alkalmazására tekintsük az $X \sim \mathcal{Exp}(\lambda)$ és az $Y \sim \mathcal{Geo}(p)$ valószínűségi változók közötti alábbi kapcsolatot, ahol $\lambda > 0$ és $p \in (0, 1)$!
- Ekkor minden $x \geq 0$ értékre $F_X(x) = 1 - e^{-\lambda x}$, illetve, a $q = e^{-\lambda}$ jelöléssel élve, minden $i \in \mathbb{N}$ egész értékre az

$$F_X(i+1) - F_X(i) = e^{-\lambda i} - e^{-\lambda(i+1)} = e^{-\lambda i} (1 - e^{-\lambda}) = (1 - q) q^i = P(Y = i)$$

egyenlőséget kapjuk.

- Következésképpen, ha $U \sim \mathcal{U}((0, 1])$, akkor

$$\left\lceil -\frac{1}{\lambda} \ln(U) \right\rceil \sim \mathcal{Geo}(1 - q) = \mathcal{Geo}(1 - e^{-\lambda}).$$



Megjegyzés

Az eltelt évtizedek során számos jóval hatékonyabb módszert is kidolgoztak diszkrét valószínűségi változók mintavételezésére. Például beszélhetünk:

- összehasonlítás alapú inverziós módszerekről, amelyek hatékony keresési adatszerkezetekre (például bináris kereső és Huffman-fára) épülnek [Huffman, 1952, Zimmerman, 1959, Aho et al., 1983];
- vezérlő táblázatok módszeréről [Chen, Asau, 1974];
- javításos inverzióról;
- asszociatív tömbökre, vagy indextáblázatokra épülő módszerekről [Marsaglia, 1963, Norman, Cannon, 1972];
- alias táblázatok kihasználó eljárásokról [Walker, 1974, Walker, 1977], illetve azok átlalánosításairól [Peterson, Kronmal, 1982].

A segédlet nem tér ki a fenti módszerek ismertetésére, az érdeklődő hallgatóinkat vagy az előbb felsorolt forrásmunkák, vagy az ezeket átfogó [Devroye, 1986] könyv tanulmányozására kérjük.



1. feladat

- Alkalmazzátok az 1. táblázatban felsorolt folytonos valószínűségi változók mintavételezésére az 1. algoritmust! Az implementálandó függvény fejléce:

```
% X = [Xi]i=1n      – denotes the sample that has to be generated
% distribution_type – a string that identifies the distribution of the
%                   random variate X
% parameters       – an array that stores the parameters of the given
%                   distribution
% n                – denotes the sample size
function X = ExactInversion(distribution_type, parameters, n)
```

- Felhasználva, hogy tetszőleges $a, b > 0$ paraméterek esetén az

$$F(x) = \begin{cases} 0, & x \leq 0, \\ 1 - \exp\left(-\left(\frac{x}{a}\right)^b\right), & x > 0 \end{cases}$$

leképezés egy folytonos X valószínűségi változó eloszlásfüggvénye, számításátok ki az X változó sűrűségfüggvényét, végül pedig alkalmazzátok az egzakt inverziós módszert az X változó mintavételezésére (a számításaitok alapján egészítsétek ki és teszteljétek is a [ContinuousPDF](#), [ContinuousCDF](#), valamint [ExactInversion](#) függvényeiteket)! Mit tudtok mondani a kellően nagy $\{X_i \sim X\}_{i=1}^n$ mintavétel hisztogramának és az X sűrűségfüggvény alakjáról?

2. feladat

A 2. és 3. algoritmusok segítségével generáljatok tetszőleges $\mathcal{N}(\mu, \sigma)$ -eloszlású valószínűségi változókat, ahol $\mu \in \mathbb{R}$ és $\sigma > 0$!

3. feladat

Generáljatok a 4. algoritmussal $\chi^2(n=3, \sigma=1)$ -eloszlású számokat a $(0.1, 5)$ intervallumban!

4. feladat

Felhasználva a 6. és 8. algoritmust, generáljatok $\mathcal{Poisson}(\lambda)$ -, illetve $\mathcal{Geo}(p)$ -eloszlású számokat, ahol $\lambda > 0$, míg $p \in (0, 1)$!

5. feladat

A sürgősségi osztályon egy balesetet szenvedett AB vércsoportú betegnek rögtönzött véradást szerveznek. A nővérek mindaddig tesztelik az önkéntes véradók vércsoportját, amíg megfelelő vércsoportú személyt nem találnak. Tudva, hogy 200 ember közül átlagosan 50 személy A , 65 személy B , 70 személy O , a maradék pedig AB vércsoporttal rendelkezik, várhatóan hány önkéntes véradónak kell tesztelni a vércsoportját az első megfelelő vércsoportú személy megtalálásáig? A feladat megoldásához használjátok fel az 1. kódrészletben adott [InversionBySequentialSearch](#) függvényt!





D. Huffman, **1952**.

A method for the construction of minimum-redundancy codes.

In Proceedings of the IRE, **40**(9):1098–1101.



S. Zimmerman, **1959**.

An optimal search procedure,

American Mathematical Monthly, **66**(8):690–693.



A.V. Aho, J.E. Hopcroft, J.D. Ullman, **1983**.

Data Structures and Algorithms,

Addison-Wesley.



H.C. Chen, Y. Asau, **1974**.

On generating random variates from an empirical distribution,

AIIE Transactions, **6**(2):163–166.



G. Marsaglia, **1963**.

Generating discrete random variables in a computer,

Communications of the ACM, **6**(1):37–38.





J.E. Norman, L.E. Cannon, **1972.**

A computer program for the generation of random variables from any discrete distribution,

Journal of Statistical Computation and Simulation, **1**(4):331–348.



A.V. Peterson, R.A. Kronmal, **1982.**

On mixture methods for the computer generation of random variables,

The American Statistician, **36**(3):184–191.



A.J. Walker, **1974.**

New fast method for generating discrete random numbers with arbitrary frequency distributions,

Electronics Letters, **10**(8):127–128.



A.J. Walker, **1977.**

An efficient method for generating discrete random variables with general distributions,

ACM Transactions on Mathematical Software, **3**(3):253–256.



L. Devroye, **1986.**

Non-Uniform Random Variate Generation,

Springer-Verlag New York Inc.

