Generatory liczb pseudolosowych

Plan wykładu:

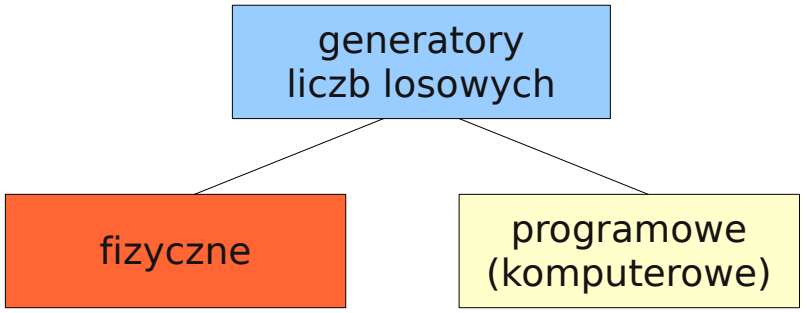
1. Generatory o rozkładzie równomiernym
   a) liniowe
   b) kombinowane – generator uniwersalny
   c) nieliniowe

2. Generatory o dowolnym rozkładzie prawdopodobieństwa
   a) metoda odwracania dystrybuanty
   b) metoda eliminacji
   c) superpozycja rozkładów
   d) rozkład dyskretny

3. Generatory o rozkładach wielowymiarowych
   a) rozkład równomierny na sferze i kuli w R^M
   b) dwuwymiarowy rozkład normalny

4. Testowanie generatorów
   a) testy zgodności z rozkładem: χ^2, OPSO
   b) testy zgodności z rozkładem statystyk: pozycyjne, test sum

Generatory o rozkładzie równomiernym



Najprostsze generatory liczb losowych to generatory fizyczne wykorzystujące:
1) szumy układów elektronicznych
2) Promieniotwórczość

Zalety: dostajemy ciągi liczb losowych (niezależne, nieskorelowane)
Wady: wymagana ciągła kalibracja (testowanie parametrów), kłopoty techniczne z obsługą, brak powtarzalności serii.

Własności generatorów komputerowych
a) łatwość obsługi
b) możliwość generowania dowolnego rozkładu
c) dowolna liczba wymiarów
d) Powtarzalność ciągów generowanych liczb

Jak pracują komputerowe generatory liczby?

1) Tworzony jest ciąg liczb nieujemnych (naturalnych)

$$X_0, X_1, X_2, \ldots, X_n$$

o **rozkładzie równomiernym** według wybranego algorytmu. Liczby ograniczone są od góry przez reprezentację, np. dla k=32 bitowej reprezentacji liczby całkowitej bez znaku, kres górny

$$m = 2^{32}$$

$$\underbrace{X_0, \ldots, X_{\nu-1}}_{T_a} \underbrace{X_\nu, X_{\nu+1}, \ldots, X_{\nu+P-1}}_{T_o}, X_{\nu+P}, \ldots$$

T_a – okres aperiodyczności ciągu
T_0 – okres ciągu

2) Aby uzyskać liczby „rzeczywiste” dokonujemy przekształcenia

$$U = \frac{X}{m} \Rightarrow U_i \in (0, 1]$$

3) Dokonujemy kolejnej transformacji ciągu aby uzyskać ciąg o zadanym rozkładzie prawdopodobieństwa (normalny, wielomianowy, etc.)

Generatory liniowe

Generatory liniowe tworzą ciąg liczb według schematu:

$$X_{n+1} = (a_1 X_n + a_2 X_{n-1} + \ldots + a_k X_{n-k+1} + c) \ mod \ m$$

gdzie:
a_1, a_2,...,a_k, c, m – parametry generatora (ustalone liczby)

Operację

$$r = (a \ mod \ n), \quad a, n, r \in Z$$

nazywamy dzieleniem modulo a jej wynikiem jest reszta z dzielenia liczb całkowitych a i n.

Lub inaczej: r **jest kongruentne do a modulo n jeśli n jest dzielnikiem a-r**.

$$a \equiv r \ mod \ n \Rightarrow r = a - \left\lfloor \frac{a}{n} \right\rfloor n$$

Generatory wykorzystujące operację dzielenia modulo to generatory **kongruentne** lub **kongruencyjne**.

Przykład

19 mod 6 =**1**
18 mod 6 =0
17 mod 6 =5
16 mod 6 =4

15 mod 6 =3
14 mod 6 =2
13 mod 6 =**1**
12 mod 6 =0

Aby wygenerować ciąg liczb pseudolosowych należy zdefiniować jego parametry.
Liczby

$$X_0, X_1, X_2, \ldots, X_k$$

nazywamy ziarnem generatora (seed). Dla bardziej rozbudowanych generatorów liczby te otrzymujemy z innego generatora lub np. używając zegara systemowego (X_0).

Najprostszy generator liniowy ma dwie odmiany
a) generator multiplikatywny gdy

$$c = 0$$

b) generator mieszany gdy

$$c \neq 0$$

Maksymalny okres generatora liniowego to (m-1)

Generator multiplikatywny

$$X_{i+1} = aX_{i-1} \ mod \ m$$

$$k_i = \left\lfloor \frac{aX_{i-1}}{m} \right\rfloor, \quad i \geq 1$$

$$X_1 = aX_0 - mk_1$$
$$X_2 = a^2 X_0 - mk_2 - mk_1 a$$
$$X_3 = a^3 X_0 - mk_3 - mk_2 a - mk_1 a^2$$
$$\ldots\ldots\ldots\ldots$$
$$X_n = a^n X_0 - m(k_n + k_{n-1} a + \ldots + k_1 a^{n-1})$$

Ostatnie równanie można zapisać w postaci

$$X_n = a^n X_0 \ mod \ m$$

skąd wynika, że wybór X_0 determinuje wszystkie liczby w generowanym ciągu (a i m są ustalone) – uzyskany ciąg liczb jest **deterministyczny**

Przykład.

Generator multiplikatywny

$$X_i = aX_{i-1} \ mod \ 11, \quad X_0 = 1$$

					X_i					
	a=1	2	3	4	5	6	7	8	9	10
i=0	1	**1**	1	1	1	**1**	**1**	**1**	1	1
1	1	**2**	3	4	5	**6**	**7**	**8**	9	10
2		**4**	9	5	3	**3**	**5**	**9**	4	1
3		**8**	5	9	4	**7**	**2**	**6**	3	
4		**5**	4	3	9	**9**	**3**	**4**	5	
5		**10**	1	1	1	**10**	**10**	**10**	1	
6		**9**				**5**	**4**	**3**		
7		**7**				**8**	**6**	**2**		
8		**3**				**4**	**9**	**5**		
9		**6**				**2**	**8**	**7**		
10		**1**				**1**	**1**	**1**		

Okres generatora multiplikatywnego

$$T = min\{i : X_i = X_0, i > 0\}$$

Maksymalny okres generatora multiplikatywnego uzyskujemy dla

$$a^{(m-1)/p} \neq 1 \ mod \ m$$

Gdy m jest liczbą pierwszą a p jest czynnikiem pierwszym liczby (m-1).

Przykład
Wykorzystujemy **liczby Marsenne'a** (które dość często są liczbami pierwszymi)

$$m = 2^p - 1$$

$$p = 31 \Rightarrow m = 2^{31} - 1, \quad a = 7^5$$

Okres generatora

$$T = 2^{31} - 2$$

Liczby

$$U = \{U_i, 1 \leq i \leq T\}$$

występują dokładnie 1 raz w pojedynczym okresie generatora.
Odległość pomiędzy najbliższymi sąsiadami

$$\frac{1}{2^{31} - 1} = 4.657 \times 10^{-10}$$

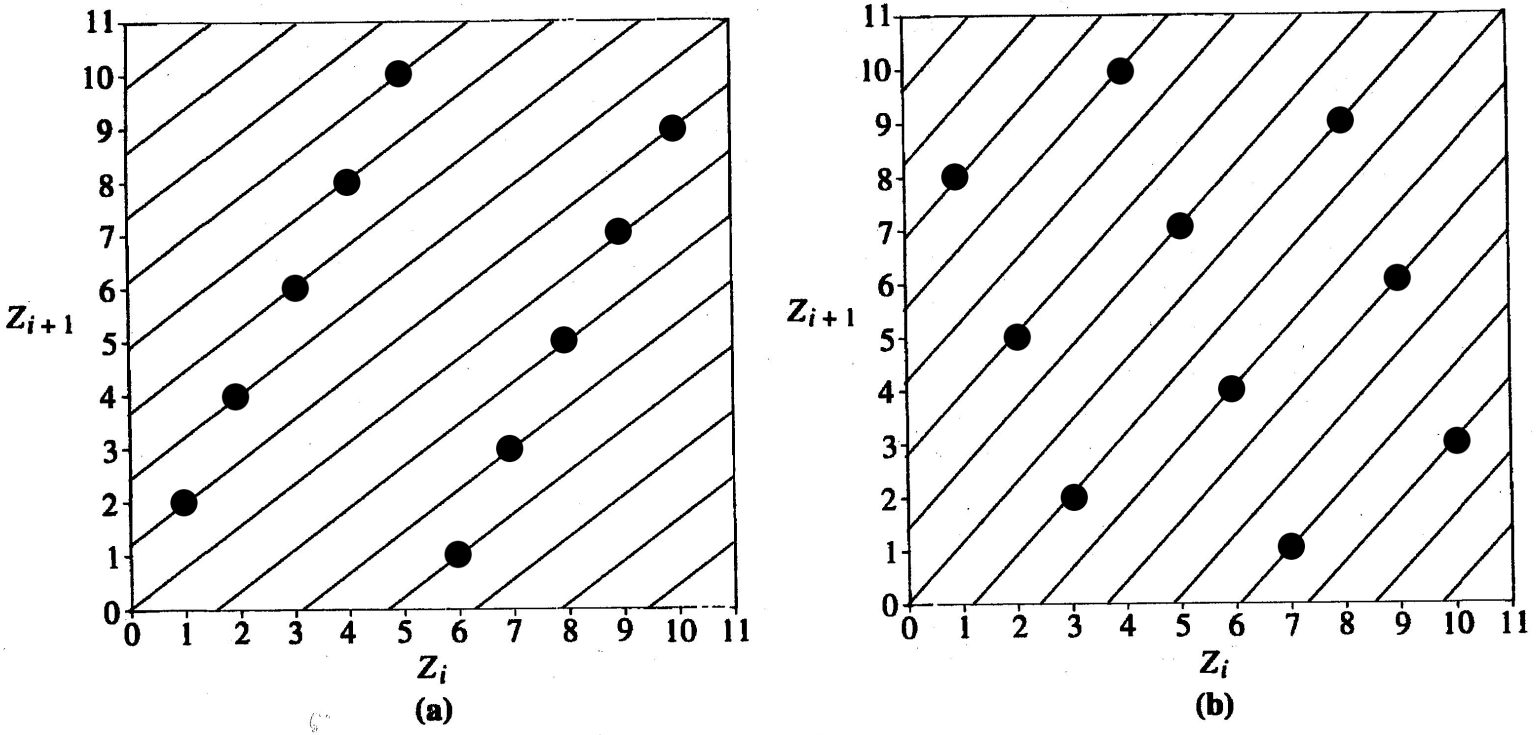
Rozkład przestrzenny ciągu

Wadą generatorów multiplikatywnych jest nierównomierne pokrycie d-wymiarowej kostki (I^d). Generowane liczby lokalizują się na hiperpłaszczyznach, których położenie uzależnione jest od parametrów generatora.

$$(U_1, U_2, \ldots, U_d), (U_2, U_3, \ldots, U_{d+1}), \ldots$$

$$(U_1, U_2, \ldots, U_d), (U_{d+1}, U_{d+2}, \ldots, U_{2d}), \ldots$$

Przykład. $X_i = aX_{i-1} \ mod \ 11$



a) X_0=1, a=2
b) X_0=1, a=8

Parametry statystyczne generatora o rozkładzie równomiernym w (0,1)-> U(0,1)

Jeśli generowany ciąg liczb jest niezależny to wartość oczekiwana (średnia) powinna wynosić

$$\mu = \int_0^1 x dx = \frac{x^2}{2}\Big|_0^1 = \frac{1}{2}$$

$$\bar{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

natomiast wariancja jest równa

$$\sigma^2 = \int_0^1 (x-\mu)^2 dx = \frac{1}{12}$$

$$\bar{\sigma} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{\mu})^2$$

Ponadto współczynniki autokorelacji elementów ciągu powinny wynosić 0.

Jeśli parametry statystyczne generatora (ciągu generowanych przez niego liczb) odbiegają od powyższych wartości to jest on nieprzydatny (lub warunkowo przydatny).

Przykład.
Przeanalizujmy parametry statystyczne L bitowego generatora

$$X_{i+1} = aX_i \ mod\ 2^L$$

Można nim wygenerować 4 ciągi liczb o okresie

$$T = 2^{L-2}$$

Jeden z nich:

$$c \equiv 3 \ mod\ 8, \quad X_0 \equiv 1,3,9,11 \ mod\ 16$$

Ciąg liczb **pseudolosowych**

$$X_0, X_1, X_2, \ldots$$

jest permutacją liczb

$$8j+1,\ 8j+3,\ j = 0,1,\ldots,2^{L-3}-1$$

Liczby znormalizowane

$$U_i = \frac{X_i}{m} = \frac{X_i}{2^L}$$

Średnia

$$\bar{\mu} = \frac{1}{2} - \frac{1}{2^{L-1}}$$

Wariancja

$$\sigma^2 = \frac{1}{12} - \frac{13}{3\cdot 2^m}$$

Funkcja autokorelacji opisuje zależność elementów ciągu od wyrazów poprzednich. Definicja

$$\tilde{R}_r = \frac{E[(X_t - \mu)(X_s - \mu)]}{\sigma^2}$$

oraz wzór dla ciągu skończonego (r=s-t)

$$R_r = \frac{1}{(N-r)\sigma^2} \sum_{i=1}^{N-r} (X_i - \mu)(X_{i+r} - \mu)$$

Inaczej: opisuje związek pomiędzy elementami dówch szeregów – danego i przesuniętego o r.

Efektywnie funkcję autokorelacji można badać przy użyciu FFT – splot dwóch wektorów.

Dla analizowanego generatora mieszanego oraz dla r=1 współczynnk autokorealcji można oszacować z poniższej relacji

$$\begin{aligned} A &= \frac{1}{a} - \frac{6c}{am}(1 - \frac{c}{m}) \\ B &= \frac{a}{m} \\ R_1 &\in [A - B, A + B] \end{aligned}$$

Przykłady generatorów liniowych

$$X_i = (1176X_{i-1} + 1476X_{i-2} + 1776X_{i-3}) \ mod \ (2^{32} - 5)$$
$$X_i = 2^{13}(X_{i-1} + X_{i-2} + X_{i-3}) \ mod \ (2^{32} - 5)$$
$$X_i = (1995X_{i-1} + 1998X_{i-2} + 2001X_{i-3}) \ mod \ (2^{35} - 849)$$
$$X_i = 2^{19}(X_{i-1} + X_{i-2} + X_{i-3}) \ mod \ (2^{32} - 1629)$$

Ich okresy są maksymalne tj. równe (m-1)

Generatory na rejestrach przesuwnych

Definiujemy ciąg bitów b_i otrzymywanych rekurencyjnie

$$b_i = (a_1 b_{i-1} + \ldots + a_k b_{i-k}) \ mod \ 2$$

$$i = k+1, k+2, \ldots$$

gdzie:

$$a_1, a_2, \ldots, a_k \in \{0, 1\}$$

są stałymi binarnymi , a stałe b_i

$$b_1, b_2, \ldots, b_k \in \{0, 1\}$$

tworzą ciąg inicjujący.

Inny sposób zapisu relacji rekurencyjnej wykorzystuje operator **xor**

a	b	a xor b
0	0	0
0	1	1
1	0	1
1	1	0

Który można zdefiniować

$$a\ xor\ b = (a+b)\ mod\ 2$$

Jeśli założymy

$$a_{j1} = a_{j2} = \ldots = a_{jk} = 1$$

to relację rekurencyjną można zapisać przy użyciu xor

$$b_i = b_{i-j_1}\ xor\ b_{i-j_2}\ xor\ \ldots\ xor\ b_{i-j_k}$$

Jakie są własności takiego ciągu bitów?

Ciąg jest okresowy o okresie nieprzekraczającym 2^k a dokładniej (2^k-1) ze względu na wyrzucenie układu bitów złożonych z samych zer.
Praktyczną realizacją tego pomysłu jest algorytm wykorzystujący tylko dwa elementy ciągu:

$$b_i = b_{i-p}\ xor\ b_{i-q}, \quad p > q, \quad p, q \in N$$

Wykorzystujemy ciąg bitów do obliczenia liczby pseudolosowej z przedziału (0,1]
(**generator Tauswortha**)

$$\begin{aligned} U_i &= \sum_{j=1}^{L} 2^{-j} b_{is+j} \\ &= 0.b_{is+1} \ldots b_{is+L} \\ i &= 0, 1, 2, \ldots \end{aligned}$$

gdzie: s jest ustaloną całkowitą liczbą nieujemną

a) jeśli s<L to do utworzenia U_i oraz U_{i+1} wykorzystywane są elementy tego samego podciągu
b) jeśli s=L to U_i oraz U_{i+1} są tworzone z rozłącznych fragmentów ciągu globalnego

Ciąg bitów łatwo generuje się przy użyciu rejestrów przesuwnych oraz bramek logicznych (xor) – łatwa implementacja w języku C.

Generator Fibonacciego

Punktem wyjścia do konstrukcji generatora są liczby Fibonacciego

$$\begin{aligned} f_0 &= f_1 = 1 \\ f_n &= f_{n-2} + f_{n-1} \end{aligned}$$

a dokładniej ciąg reszt tego ciągu

$$X_i = X_{i-2} + X_{i-1} \ mod \ m, \quad i \geq 2$$

Powyższy ciąg reszt ma rozkład równomierny ale nie spełnia testów niezależności (autokorelacja). Jego modyfikacja

$$X_i = X_{i-r} \diamond X_{i-s} \ mod \ m$$

$$i \geq r, \quad r > s \geq 1$$

$$\diamond = +, \ -, \ *, \ xor$$

nie posiada już tej wady.
Okres generatora zależy od operacji. Dla

$$m = 2^L$$

$$T = \begin{cases} (2^r - 1)2^{L-1} & F(r, s, +) \\ (2^r - 1)2^{L-1} & F(r, s, -) \\ (2^r - 1)2^{L-3} & F(r, s, *) \\ (2^r - 1) & F(r, s, xor) \end{cases}$$

Generator charakteryzuje się dużym okresem ale jest wolniejszy np. względem generatorów multiplikatywnych co obecnie nie jest już dużą wadą.

Kombinacje generatorów

Zakładamy że dysponujemy zmiennymi losowymi X i Y określonymi na zbiorze S={1,2,...,n}
z rozkładami prawdopodobieństwa

$$\begin{aligned} P\{X = i\} &= r_i \\ P\{Y = i\} &= q_i \\ i &= 1, 2, \ldots, n \end{aligned}$$

Z tych liczb możemy utworzyć wektory

$$\boldsymbol{r} = (r_1, r_2, \ldots, r_n)$$

$$\boldsymbol{q} = (q_1, q_2, \ldots, q_n)$$

Za normę wektora przyjmiemy p-normę

$$\|r\| = \left(\sum_{i=1}^{n} r_i^p \right)^{1/p}$$

Miarą „odległości” danego rozkładu od rozkładu rónomiernego będzie wówczas wyrażenie

$$\delta(X) = \|(r_1, r_2, \ldots, r_n) - (1/n, 1/n, \ldots, 1/n)\|$$

Dla określonego działania na zbiorze S tj.

$$+,\ -,\ *,\ xor$$

rozkład nowej zmiennej losowej

$$X \diamond Y$$

będzie bliższy rozkładowi równomiernemu niż rozkłady zmiennych X i Y

$$\delta(X \diamond Y) \leq min\{\delta(X), \delta(Y)\}$$

Nowy ciąg ma lepsze własności statystyczne a także większy okres. Jeżeli ciąg

$$X_1, X_2, \ldots, X_n$$

ma okres T_1, a ciąg

$$Y_1, Y_2, \ldots, Y_n$$

ma okres T_2 i są liczbami pierwszymi to wtedy nowy ciąg liczb

$$X_i \diamond Y_j$$

ma okres równy T_1T_2.

Generator uniwersalny

Daje jednakowe wyniki na dowolnym komputerze. Jego działanie oparte jest na wykorzystaniu kombinacji kilku generatorów.

$$U_n = V_n \diamond c_n$$

Pierwszy z nich jest generatorem Fibonacciego

$$F(97, 33, \diamond)$$

$$V_i = V_{i-97} \diamond V_{i-33}$$

$$V_i \in [0, 1)$$

działanie jest zdefiniowane następująco

$$x \diamond y = x - y, \quad x \geq y$$

$$x \diamond y = x - y + 1, \quad x < y$$

Inicjalizację generatora tj.wyznaczenie ciągu

$$V_1, V_2, \ldots, V_{97}$$

przeprowadzamy przy pomocy ciągu bitów (24-bitowa mantysa, 16 bitowe liczby całkowite) tj.

$$V_1 = 0.b_1b_2 \ldots b_{24}$$

$$V_2 = 0.b_{25}b_{26} \ldots b_{48}$$

Ciąg bitów generujemy przy użyciu dwóch generatorów

$$y_n = (y_{n-3}y_{n-2}y_{n-1})\ mod\ 179$$

$$z_n = (52z_{n-1} + 1) \mod 169$$

$$b_n = \begin{cases} 0, & gdy\ y_nz_n\ mod\ 64 < 32 \\ b_n = 1, & w\ pozostalych\ przypadkach \end{cases}$$

Dla określonego działania na zbiorze S tj.

$$+,\ -,\ *,\ xor$$

rozkład nowej zmiennej losowej

$$X \diamond Y$$

będzie bliższy rozkładowi równomiernemu niż rozkłady zmiennych X i Y

$$\delta(X \diamond Y) \leq min\{\delta(X), \delta(Y)\}$$

Nowy ciąg ma lepsze własności statystyczne a także większy okres. Jeżeli ciąg

$$X_1, X_2, \ldots, X_n$$

ma okres T_1, a ciąg

$$Y_1, Y_2, \ldots, Y_n$$

ma okres T_2 i są liczbami pierwszymi to wtedy nowy ciąg liczb

$$X_i \diamond Y_j$$

ma okres równy T_1T_2.

Generator uniwersalny

Daje jednakowe wyniki na dowolnym komputerze. Jego działanie oparte jest na wykorzystaniu kombinacji kilku generatorów.

$$U_n = V_n \diamond c_n$$

Pierwszy z nich jest generatorem Fibonacciego

$$F(97, 33, \diamond)$$

$$V_i = V_{i-97} \diamond V_{i-33}$$

$$V_i \in [0, 1)$$

działanie jest zdefiniowane następująco

$$x \diamond y = x - y, \quad x \geq y$$

$$x \diamond y = x - y + 1, \quad x < y$$

Inicjalizację generatora tj.wyznaczenie ciągu

$$V_1, V_2, \ldots, V_{97}$$

przeprowadzamy przy pomocy ciągu bitów (24-bitowa mantysa, 16 bitowe liczby całkowite) tj.

$$V_1 = 0.b_1b_2 \ldots b_{24}$$

$$V_2 = 0.b_{25}b_{26} \ldots b_{48}$$

Ciąg bitów generujemy przy użyciu dwóch generatorów

$$y_n = (y_{n-3}y_{n-2}y_{n-1})\ mod\ 179$$

$$z_n = (52z_{n-1} + 1) \mod 169$$

$$b_n = \begin{cases} 0, & gdy\ y_nz_n\ mod\ 64 < 32 \\ b_n = 1, & w\ pozostalych\ przypadkach \end{cases}$$

Generatory muszą być zainicjowane

$$y_1, y_2, y_3 \in \{1, 2, \ldots, 178\}$$

$$z_1 \in \{0, 1, \ldots, 168\}$$

Okres generatora kombinowanego: **2^{120}**.

Drugi generator (c_n) o rozkładzie równomiernym w (0,1) jest zdefiniowany następująco

$$\begin{aligned} c_n &= c_{n-1} \diamond \left(\frac{7654321}{16777216}\right) \\ n &\geq 2 \\ c_1 &= \frac{362436}{16777216} \end{aligned}$$

$$c \diamond d = c - d, \quad c \geq d$$

$$c \diamond d = c - d + \frac{362436}{16777216}, \quad c < d$$

$$c, d \in [0, 1)$$

Okres tego generatora to 2^{24}-3.
Okres generatora uniwersalnego (U_n): **2^{144}**

Generatory nieliniowe

Zaletą generatorów nieliniowych jest to, że ciągi generowanych przez nie liczb nie układają się na hiperpłaszczyznach tak jak w przypadku generatorów liniowych. Wykorzystuje się w nich „odwrotność modulo”:

$$x \cdot x^{-1} \ mod \ m = 1$$

Przykład:

$$5 \cdot 3 \ mod \ 7 = 1 \qquad x = 5 \qquad x^{-1} = 3$$

Do znalezienia x^{-1} wykorzystuje się algorytm podziału Euklidesa (poszukiwania największego wspólnego podzielnika dwóch liczb)

$$a, b, \in Z, \quad b \neq 0, \quad a = b \cdot q + r$$

$$q, r \in Z, \quad 0 \leq r < |b|$$

Algorytm:

$$\begin{aligned} r_0 &= a \\ r_1 &= b \qquad q_i = \left\lfloor \frac{r_{-1}}{r_i} \right\rfloor \\ &\vdots \\ r_{i+1} &= r_{i-1} - q_i r_i, \quad 0 \leq r_{i+1} < |r_i| \end{aligned}$$

Jeśli $r_{i+1} = 0, \Longrightarrow gcd(a, b) = r_i$

(gcd-greatest common divisor)

Generowanie x^{-1} algorytmem Aignera:

1) generujemy ciąg liczb:

$$\{z_{n+2}, z_{n+1}, \quad z_n, \ldots, \quad z_1\}$$

$$z_{n+2} = m, \; z_{n+1} = x, \; z_1 = 1$$

$$z_i = z_{i+2} - q_{i+1} z_{i+1}, \quad 1 \leq i \leq n$$

$$q_{i+1} = \left\lfloor \frac{z_{i+2}}{z_{i+1}} \right\rfloor$$

i wyznaczamy współczynniki q_i

$$\{q_{n+1}, q_n, \ldots, q_1\}$$

2) generujemy ciąg liczb

$$\{w_1, w_2, \ldots w_{n+2}\}$$

$$w_1 = 1, \; w_2 = 0$$

$$w_i = q_{i-1} w_{i-1} + w_{i-2}, \quad 3 \leq i \leq n+2$$

$$x^{-1} = (-1)^{n+1} w_{n+2}$$

3) jeśli x=0 to $x^{-1}=0$

Generator nielinowy Eichenauera-Lehna

$$X_{i+1} = (aX_i^{-1} + b) \; mod \; m, i = 0, 1, \ldots$$

gdzie: m jest liczbą pierwszą

Uzyskujemy w ten sposób ciąg liczb

$$X_i \in \{0, 1, \ldots, m-1\}$$

o rozkładzie równomiernym.

$$U_i = \frac{X_i}{m} \in [0, 1)$$

Generator Eichenauera-Hermanna

$$X_i = (a(i + i_0) + b)^{-1} \; mod \; m, \quad i = 0, 1, \ldots$$

W generatorze tym kolejne elementy ciągu są niezależne od poprzednich – cecha szczególnie przydatna w obliczeniach równoległych.

Dla $a \in \{1, 2, \ldots, m\}$

generator osiąga okres T=m.
Aby generator osiągał maksymalny okres równy m musi być spełniony jeszcze warunek

$$m^2 - 1$$

jest najmniejszą liczbą całkowitą dla której zachodzi

$$z^{m^2-1} \equiv 1 \; mod \; (z^2 - bz - a)$$

$$z \in Z$$

Generatory o dowolnym rozkładzie prawdopodobieństwa

Metoda odwracania dystrybuanty

Dystrybuanta określonego rozkładu prawdopodobieństwa jest funkcją

$$F : R \to R$$

niemalejącą i prawostronnie ciągłą

$$\lim_{x \to -\infty} F(x) = 0$$

$$\lim_{x \to \infty} F(x) = 1$$

Dystrybuanta jednoznacznie definiuje rozkład prawdopodobieństwa.
Związek pomiędzy dystrybuantą a **gęstością prawdopodobieństwa** f(x):

$$F(x) = \int_{-\infty}^{x} f(y)dy$$

Jeśli uda nam się znaleźć F^{-1} to:

$$U = F(x) \to x = F^{-1}(U)$$

zmienna losowa x ma rozkład o dystrybuancie F

U jest zmienną losową o rozkładzie równomiernym w przedziale (0,1).

Nową zmienną losową będzie

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

$$\begin{aligned} P\{X \le x\} &= P\{F^{-1}(U) \le x\} \\ &= P\{U \le F(x)\} \\ &= F(x) \end{aligned}$$

Generujemy więc ciąg liczb pseudolosowych

$$U_1, U_2, \ldots, U_n \in (0,1)$$

który przekształcamy w ciąg

$$X_1, X_2, \ldots, X_n \in (-\infty, \infty)$$

Liczby X_i mają rozkład prawdopodobeństwa o dystrybuancie F.
Odwracanie dystrybuanty można wykorzystać także w przypadku rozkładów dyskretnych.

Np. ciąg zmiennych

$$X_1, X_2, \ldots, X_n$$

o rozkładzie

$$p_k = P\{X = k\}, \quad k = 0, 1, 2, \ldots$$

Można wygenerować przy użyciu ciągu

$$U_1, U_2, \ldots, U_n \in (0,1)$$

korzystając ze wzoru

$$X_n = min\left\{k : U_n \le \sum_{i=0}^{k} p_i\right\}, \quad n = 1, 2, \ldots$$

Odwracanie dystrybuanty sprawia często duże trudności numeryczne.

Przykład - rozkład jednomianowy

$$f(x) = x^n \qquad x \in [0,1], \quad n = 1, 2, 3, \ldots$$

$$F(y) = \int_0^y x^n dx = \frac{y^{n+1}}{n+1} = U$$

$$y = ((n+1)U)^{\frac{1}{n+1}} \qquad y \in [0,1]$$

Przykład - rozkład wykładniczy

gęstość prawdopodobieństwa

$$f(x) = e^{-x}, \quad x \in [0, \infty)$$

Dystrybuanta

$$F(x) = \int_0^x e^{-x'} dx'$$

$$F(x) = 1 - e^{-x} = U$$

$$e^{-x} = 1 - U$$

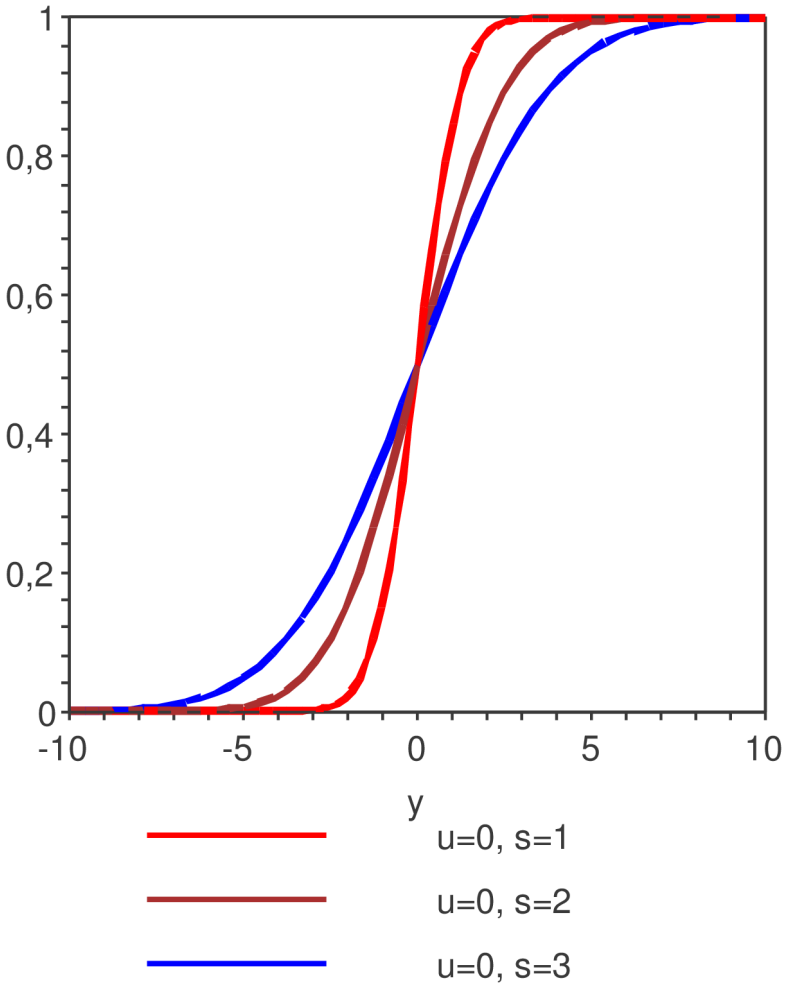
$$F^{-1}(x) = x = -ln(1 - U)$$

$$U \in (0, 1) \rightarrow x \in (0, \infty)$$

Przykład - rozkład normalny N(0,1)

$$f(x) = e^{-x^2} \qquad \left(x = \frac{y - \mu}{\sqrt{2}\sigma} \right)$$

$$F(x) = \int_{-\infty}^{x} e^{-x'^2} dx' = erf(x)$$



Szukanie funkcji odwrotnej erf(x) jest kosztowne. Częściej stosuje się metodę Boxa-Mullera:

$$f(x,y) = f(x) \cdot f(y) = e^{-\frac{x^2+y^2}{2}}$$

$$x, y \in (-\infty, \infty)$$

chcemy policzyć prawdopodobieństwo

$$p(x,y) = f(x,y)dxdy$$

Wprowadzamy nowe zmienne

$$r^2 = x^2 + y^2$$

$$x = rcos(\theta) \qquad r \in [0, \infty)$$

$$y = rsin(\theta) \qquad \theta \in [0, 2\pi]$$

$$p = f(x,y)dxdy = f(r,\theta)drd\theta$$

$$p(r,\theta) = r \cdot e^{-r^2/2}drd\theta$$

Wprowadzamy nową zmienną

$$z = \frac{r^2}{2} \rightarrow dz = rdr \qquad z \in [0, \infty)$$

$$p(z,\theta) = e^{-z}dzd\theta = f(z)dz \cdot d\theta$$

Dostajemy rozkład wykładniczy

$$f(z) = e^{-z}$$

$$z = -ln(1 - U_1), \quad U_1 \in (0,1)$$

Ponieważ

$$\theta = U_2 \cdot 2\pi, \quad U_2 \in (0,1)$$

Dla pary (U_1,U_2) dostajemy (x,y) z rozkładu N(0,1)

$$x = rcos(\theta) = \sqrt{-2ln(1-U_1)}cos(2\pi U_2)$$

$$y = rsin(\theta) = \sqrt{-2ln(1-U_1)}sin(2\pi U_2)$$

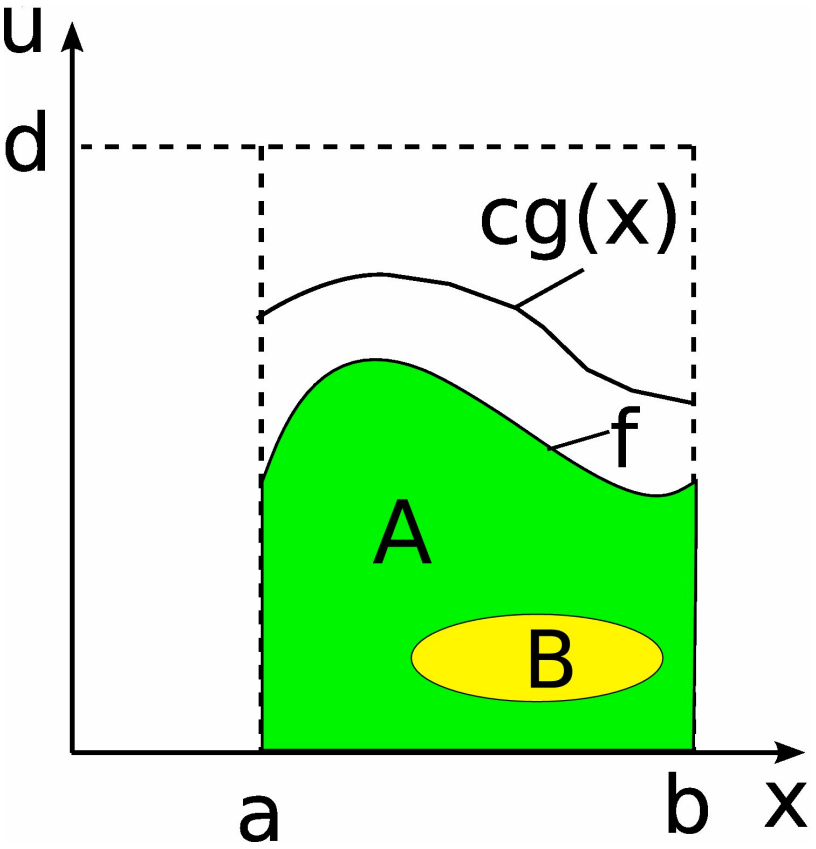
Przejście

$$x \in N(0,1) \rightarrow X \in N(\mu, \sigma)$$

$$X = x \cdot \sigma + \mu$$

Metoda eliminacji (von Neumann)

Chcemy wygenerować ciąg zmiennych losowych o gęstości prawdopodobieństwa f w przedziale [a,b].
Wartość f jest w przedziale [a,b] ograniczona od góry przez stałą d.



Sposób otrzymania ciągu zmiennych losowych o rozkładzie f(x) jest następujący:

1) Losujemy dwie zmienne o rozkładzie równomiernym

$$U_1 \in [a,b] \qquad U_2 \in [0,d]$$

2) jeżeli

$$U_2 \leq f(U_1) \Rightarrow X = U_1$$

3) gdy powyższy warunek nie jest spełniony wówczas odrzucamy parę U_1, U_2

4) wykonujemy czynności 1-3 aż do uzyskania odpowiednio licznego ciągu

Wygenerowana zmienna losowa X ma rozkład prawdopodobieństwa f.

Generowanie ciągu liczb pseudolosowych o zadanym rozkładzie algorytmem Metropolisa

Modyfikujemy metodę eliminacji. W standardowej postaci jest ona mało wydajna – bo nierzadko odrzucamy większość wyników. Dzięki algorytmowi Metropolisa nie odrzucamy żadnego.

Akceptację nowego położenia (nowej liczby w ciągu) dokonujemy zgodnie z formułą

$$h(x_{i-1}, x_i) = min(1, \frac{f(x_i)}{f(x_{i-1})})$$

czyli:

1) Jeśli $f(x_i) > f(x_{i-1})$ to nowe położenie akceptujemy zawsze.

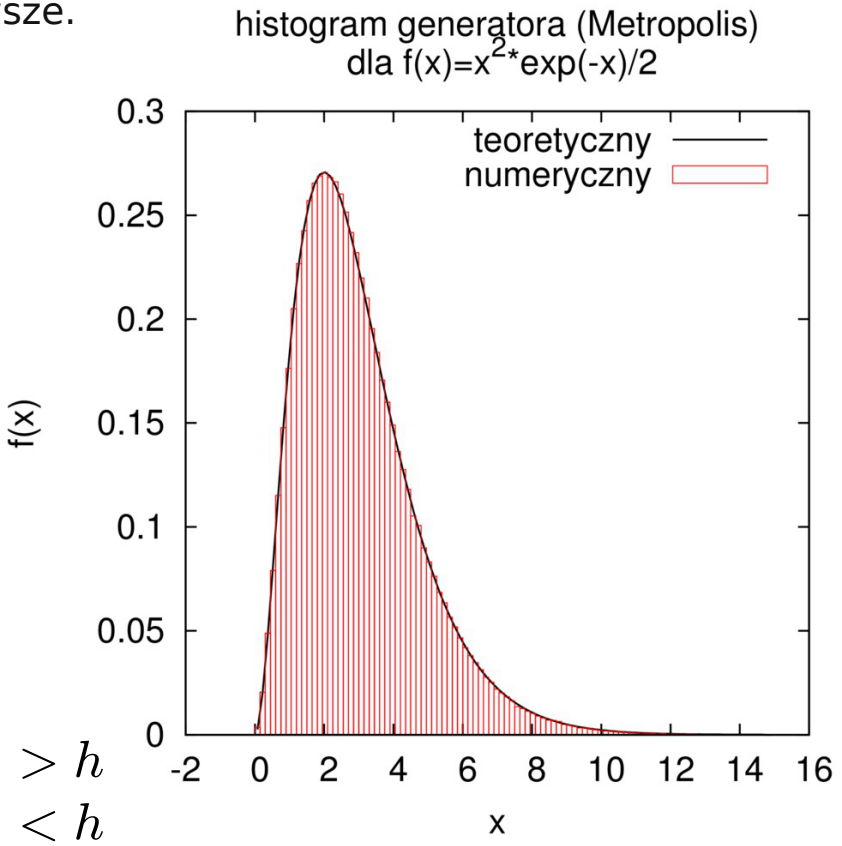
2) W przeciwnym wypadku akceptacja następuje z prawdopodobieństwem $f(x_i)/f(x_{i-1})$.
Jeśli nie akceptujemy nowego punktu to zatwierdzamy stary
(każdy krok generuje nowy element ciągu).

Uwaga: w naszym przykładzie

aby zachodził warunek $p_{ij}=p_{ji}$

$$x_{new} = x_i + (2 \cdot U(0,1) - 1)$$

$$x_{i+1} = \begin{cases} x_i & \Longleftrightarrow \quad x_{new} < 0 \\ x_i & \Longleftrightarrow \quad x_{new} > 0 \quad u \in U(0,1) > h \\ x_{new} & \Longleftrightarrow \quad x_{new} > 0 \quad u \in U(0,1) < h \end{cases}$$



Superpozycja rozkładów

Naszym zadaniem jest uzyskanie ciągu liczb pseudolosowych o gęstości f(x), którą możemy wyrazić w postaci

$$f(x) = \int_{-\infty}^{\infty} g_t(x)h(t)dt$$

gdzie: $g_t(x)$ oraz h(t) są również gęstościami prawdopodobieństwa – funkcje znane.
Rozkład prawdopodobieństwa f(x) nazywamy **rozkładem złożonym**.
Algorytm generowania liczb o rozkładzie f(x) jest następujący:

1) generujemy zmienną **T** o rozkładzie **h**
2) dla wartości **t** zmiennej losowej **T** generujemy zmienną losową **X** dla rozkładu gęstości $\mathbf{g_t(x)}$

W praktyce całkę często zastępuje się sumą

$$f(x) = \sum_{i=1}^{\infty} p_i g_i(x) = \sum_{i=1}^{K} p_i g_i(x)$$

$$p_i \geq 0, \quad \sum_{i}^{K} p_i = 1$$

Przedział [a,b] w którym generujemy ciąg zmiennych z rozkładem f(x) dzielimy na sumę K rozłącznych podprzedziałów. W każdym z nich (A_i) wyznaczamy

$$p_i = \int_{A_i} f(x)dx$$

oraz

$$g_i(x) = \frac{\mathbf{1}_{A_i} f(x)}{p_i}$$

gdzie: 1_{Ai} jest funkcją przynależności do podzbioru A_i.

Algorytm generacji ciagu zmiennej X jest wówczas następujący:

1) Losujemy zmienną losową

$$I \in \{1, 2, \ldots, K\}$$

2) Dla wygenerowanej wartości i zmiennej I generujemy X z rozkładem gęstości

$$g_i(x) = \frac{\mathbf{1}_{A_i} f(x)}{p_i}$$

Dla dostatecznie wąskich przedziałów można h(t) przybliżyć wielomianem niskiego stopnia.

Przykład – rozkłady o gęstościach wielomianowych

$$f(x) = \sum_{n=1}^{M} c_n x^n \qquad c_n \geq 0 \qquad x \in [0,1]$$

$$\int_0^1 f(x)dx = 1 = \sum_{n=1}^{M} \frac{c_n}{n+1}$$

Algorytm generowania zmiennej losowej:

1. Generujemy indeks

$$k \in \{1, 2, 3, \ldots, M\}$$

z rozkładem prawodopodobieństwa

$$P\{k = n\} = \frac{c_n}{n+1}$$

2. Wylosować zmienną losową X o rozkładzie

$$f_n(x) = (n+1)x^n$$

Zmienna X ma zadany rozkład wielomianowy.

Przykład – rozkład Beta Eulera

$$f(x;p,q) = \frac{1}{B(p,q)} x^{p-1}(1-x)^{q-1}$$

$$x \in [0,1] \qquad p,q > 0$$

$$B(p,q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$$

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$$

Jeśli dwie zmienne g_1 i g_2 mają rozkłady

$$g_1 \in \Gamma(p) \qquad g_2 \in \Gamma(q)$$

$$f_\Gamma(x,p) = x^{p-1}e^{-x}/\Gamma(p)$$

To zmienna Z:

$$Z = \frac{g_1}{g_1 + g_2}$$

ma rozkład f(x;p,q)

Rozkład dyskretny

a) metoda odwracania dystrybuanty

W rozkładzie dyskretnym mamy określone prawdopodobieństwo wylosowania danej liczby

$$P\{X = k\} = p_k, \quad k = 0, 1, 2, \ldots, M$$

wraz z warunkiem

$$\sum_{k=0}^{M} p_k = 1$$

Algorytm generowania ciągu zmiennych o rozkładzie dyskretnym:

1) X=0, S=p_0
2) Losujemy zmienną U o rozkładzie równomiernym z przedziału [0,1]
3) Sprawdzamy warunek

$$U > S$$

a)Iteracyjnie obliczamy

$$X = X + 1, \quad S = S + p_x$$

dopóki warunek jest spełniony
b) W przeciwym wypadku akceptujemy X
4) Kroki 1-3 wykonujemy aż do uzyskania odpowiednio licznego ciągu X_1,X_2,X_3,...

b) Metoda równomiernego rozbicia przedziału

Zakładamy że zmienna losowa X przyjmuje określone wartości z pewnym prawdopodobieństwem

$$P\{X = k\} = p_k, \quad k = 0, 1, 2, \ldots, K$$

Przedział (0,1) dzielimy na K+1 podprzedziałów o jednakowej długości

$$\left(\frac{i-1}{K+1}, \frac{i}{K+1}\right), \quad i = 1, 2, \ldots, K+1$$

Zmienna U wpada do przedziału

$$[(K+1)U + 1]$$

Konstruujemy dwa ciągi

$$q_i = \sum_{j=0}^{i} p_j, \quad i = 0, 1, \ldots, K \qquad q_{-1} = 0$$

$$g_i = max\left\{j : q_j < \frac{i}{K+1}\right\}, \quad i = 1, 2, \ldots, K+1$$

Algorytm

1) Generujemy zmienną U o rozkładzie równomiernym w (0,1)

2) Obliczamy

$$X = [(K+1)U + 1]$$
$$X = g_X + 1$$

3) Iteracyjnie obliczamy

$$X = X - 1$$

dopóki jest spełniony warunek

$$q_{X-1} > U$$

4) Jeśli $q_{X-1} < U$

to akceptujemy X
5) Kroki 1-4 powtarzamy aż do uzyskania odpowiednio licznego ciągu $X_1, X_2, \ldots$

Powyższy algorytm zapewnia to, że warunek

$$q_{X-1} > U$$

będzie sprawdzany conajwyżej dwukrotnie.

Generatory o rozkładach wielowymiarowych

Zadanie można sformułować następująco:

Należy wygenerować ciąg wielowymiarowych zmiennych losowych

$$\boldsymbol{X} = (X_1, X_2, \ldots, X_k)$$

których rozkład prawdopodobieństwa ma gęstość

$$f(x_1, x_2, \ldots, x_k)$$

Do generacji takiego ciągu można stosować metodę eliminacji czy superpozycji rozkładów. Przy użyciu metody elminacji w najprostszej postaci pojawiają się problemy.

Przykład.
Określić prawdopodobieństwo akceptacji wielowymiarowej zmiennej losowej o rozkładzie równomiernym na kuli jednostkowej ($K_k(0,1)$) .

Algorytm.
Losujemy m zmiennych niezależnych o rozkładzie równomiernym w (-1,1) i konstruujemy zmienną wielowymiarową

$$\boldsymbol{U} = (U_1, U_2, \ldots, U_k)$$

Zmienną akceptujemy jeśli

$$\|\boldsymbol{U}\|_2 \leq 1$$

Prawdopodobieństwo akceptacji zmiennej jest równe ilorazowi objętości kuli i opisanej na niej kostki $[-1,1]^k$

Średnia liczba wylosowanych punktów N_m potrzebnych do realizacji jednej zmiennej wynosi

$$N_m = \frac{1}{p_m}$$

m	p_k	N_k
2	7.854×10^{-1}	1.27
5	1.645×10^{-1}	6.08
10	2.490×10^{-3}	4.015×10^{2}
20	2.461×10^{-8}	4.063×10^{7}
50	1.537×10^{-28}	6.507×10^{27}

Modyfikacją usprawniającą powyższy algorytm jest podział kostki na rozłączne podobszary i przeprowadzenia losowania w każdym z nich z osobna.

Rozkład równomierny na sferze

Jeżeli k-wymiarowa zmienna losowa

$$\boldsymbol{X} = (X_1, X_2, \ldots, X_k)$$

ma rozkład równomierny na sferze to

$$S_k = \left\{ (x_1, \ldots, x_k) : \sum_{j=1}^{k} x_j^2 = 1 \right\}$$

k-wymiarowa zmienna X ma rozkład **sferycznie konturowany** jeżeli jej gęstość prawdopodobieństwa

$$g_X(x_1, x_2, \ldots, x_k) = f(\|x\|)$$

zależy tylko od

$$\|x\|^2 = \sum_{j=1}^{k} x_j^2$$

Wniosek: mając do dyspozycji odpowiedni rozkład sferycznie konturowany można go użyć do generowania zmiennej losowej o rozkładzie równomiernym na powierzchnii kuli.

Przykład.

m-wymiarowy rozkład normalny opisuje gęstość

$$g_X(x_1,\ldots,x_k) = \frac{1}{(2\pi)^{k/2}} exp\left(-\frac{1}{2}\|\vec{x}\|^2\right)$$

Algorytm generowania rozkładu równomiernego na sferze w m wymiarach:

1) Generujemy m-wymiarową zmienną losową **X** o rozkładzie normalnym
2) Obliczamy

$$\boldsymbol{X} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$$

Rozkład nowej zmiennej **X** będzie równomierny na sferze S_k.

Rozkład równomierny na sferze S_k i S_{k+1}

Tw. Jeżeli zmienna

$$\boldsymbol{X} = (x_1, x_2, \ldots, x_k)$$

ma rozkład równomierny na k-wymiarowej sferze S_k, R jest zmienną o rozkładzie

$$h(r) = \begin{cases} \frac{r^{k-1}}{\sqrt{1-r^2}}, & 0 \le r \le 1 \\ 0 & r \notin [0,1] \end{cases}$$

a s jest losowym znakiem

$$P\{s=1\} = P\{s=-1\} = \frac{1}{2}$$

to (k+1)-wymiarowa zmienna losowa

$$\boldsymbol{Z} = (Rx_1, Rx_2, \ldots, Rx_k, s\sqrt{1-R^2})$$

ma rozkład równomierny na (k+1) wymiarowej sferze.

Rozkład równomierny w kuli K_k

Długość wektora wodzącego R zmiennej **X** jest także zmienną losową i ma ona rozkład

$$h(r) = (2k-1)r^{k-1}, \quad 0 \le r \le 1$$

Wystarczy więc wygenerować punkt

$$\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_k)$$

leżący na sferze S_k a następnie obliczyć

$$\boldsymbol{X} = (RZ_1, RZ_2, \ldots, RZ_k)$$

Zmiena **X** leży wewnątrz kuli jednostkowej K_k i ma w tym obszarze rozkład równomierny.

Po co rozkład w kuli $K_k(0,1)$?

Pewne obiekty wielowymiarowe możemy przedstawić w postaci prostych transformacji liniowych współrzędnych.

Taką transformację reprezentuje macierz

$$\vec{Z} = A\vec{X}$$

Przykład – hiperelipsoidę w R^k otrzymamy transformując $K_k(0,1)$.

Testowanie generatorów liczb pseudolosowych

Ponieważ wszystkie generatory o dowolnym rozkładzie bazują na wykorzystaniu ciągów liczb losowych o rozkładzie równomiernym więc istotne jest badanie tylko generatorów liczb o takim właśnie rozkładzie.

Testowanie generatora jest procesem złożonym:

1) Dla ustalonej liczby n, generujemy n kolejny ch liczb startując od losowo wybranej liczby początkowej

2)Obliczamy wartość statystyki testowej (T)

3)Obliczamy F(T) czyli dystrybuantę statystyki T, gdy weryfikowana hipoteza jest prawdziwa

4) Kroki 1-3 powtarzamy N-krotnie obliczając statystyki: $T_1, T_2, \ldots, T_N$. Jeśli weryfikowana hipoteza jest prawdziwa to

$$F(T_1), F(T_2), F(T_3), \ldots, F(T_N)$$

jest ciągiem zmiennych niezależnych o rozkładzie równomiernym. Testowanie generatora kończy się sprawdzeniem tej hipotezy.

Testy zgodności z zadanym rozkładem

Test chi-kwadrat

Jest najczęściej stosowanym testem.
Badamy w nim hipotezę że generowana zmienna losowa X ma rozkład prawdopodobieństwa o dystrybuancie F.

Jeżeli

$$F(a) = 0 \quad F(b) = 1$$

to możemy dokonać następującego podziału zbioru wartości zmiennej X

$$a < a_1 < a_2 < \ldots < a_k = b$$

$$p_i = P\{a_{i-1} < X \leq a_i\}, \quad i = 1, 2, \ldots$$

Generujemy n liczb

$$X_1, X_2, \ldots, X_n$$

Sprawdzamy ile z nich spełnia warunek

$$a_{i-1} < X \leq a_i$$

Ich liczbę oznaczamy n_i.

Statystyką testu jest

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i}$$

Dla dużego n statystyka ta ma rozkład χ^2 o (k-1) stopniach swobody.

Możemy tak dobrać szerokości przedziałów aby otrzymać zależność

$$p_i = \frac{1}{k}$$

wówczas statystyka przyjmuje prostszą postać

$$\chi^2_{k-1} = \frac{k}{n} \sum_{i=1}^{k} n_i^2 - n$$

Test OPSO (overlapping-pairs-sparse-occupancy)

Generujemy ciąg liczb

$$X_1, X_2, \ldots, X_n$$

Jeśli z każdej weźmiemy k bitów to możemy utworzyć ciąg liczb całkowitych

$$I_1, I_2, \ldots, I_n$$

z zakresu {0,1,...,2^r-1}.

Następnie tworzymy ciąg kolejnych nakładających się par

$$(I_1, I_2), (I_2, I_3), \ldots, (I_{n-1}, I_n)$$

Jeśli przez Y oznaczymy liczbę takich par

$$\{(i, j) : i, j = 0, 1, \ldots, 2^b - 1\}$$

które nie pojawiły się w ciągu (I_i,I_{i+1}), to ta zmienna ma rozkład normalny N(μ,σ) – oczywiście dla odpowiednio dużego n.

Przykładowe parametry testu OPSO

b	n	μ	σ
10	2^{21}	141909	290.26
11	2^{22}	1542998	638.75
12	2^{23}	567639	580.80

Testy zgodności rozkładów statystyk

Jeżeli wygenerowany ciąg zmiennych losowych

$$X_1, X_2, \ldots, X_n$$

Jest ciągiem zmiennych niezależnych to możemy z elementów tego ciągu utworzyć wektory

$$(X_1, X_2, \ldots, X_m), (X_{m+1}, X_{m+2}, \ldots, X_{2m}), \ldots$$

które też będą zmiennymi losowymi ale o rozkładzie równomiernym w kostce jednostkowej $(0,1)^m$.

Możemy zatem zdefiniować pewną funkcję

$$y = h(x_1, x_2, \ldots, x_m)$$

określoną w kostce jednostkowej.
W ten sposób tworzymy ciąg nowych zmiennych losowych

$$\begin{aligned} Y_j &= h(X_{(j-1)m+1}, X_{(j-1)m+2}, \ldots, X_{jm}) \\ j &= 1, 2, \ldots \end{aligned}$$

o jednakowym rozkładzie i dystrybuancie

$$G(y) = P\{Y_j \leq y\}$$

Testowanie generatora polega na sprawdzeniu hipotezy że ciąg Y_1,Y_2,... jest próbką z populacji o dystrybuancie G.

Testy oparte na statystykach pozycyjnych

Dla wektorów losowych w kostce $(0,1)^m$ definiujemy funkcje

$$u = max\{x_1, x_2, \ldots, x_m\}$$

$$v = min\{x_1, x_2, \ldots, x_m\}$$

$$r = u - v$$

co generuje nowe zmienne

$$U,\ V,\ R$$

Nowe zmienne mają następujące rozkłady

$$P\{U_j \leq u\} = u^m, \quad 0 \leq u \leq 1$$

$$P\{V_j \leq v\} = 1 - (1 - v)^m, \quad 0 \leq v \leq 1$$

$$P\{R_j \leq r\} = mr^{m-1} - (m-1)r^m,\ 0 \leq r \leq 1$$

Testowanie generatora polega na sprawdzeniu hipotezy o zgodności rozkładów zmiennych U,V,R z powyższym. Testy przeprowadza się dla niewielkich wartości m=2,3,...,10.

Test sum

Definiujemy funkcję

$$y = x_1 + x_2 + \ldots + x_m$$

Wygenerowana zmienna losowa Y ma rozkład o gęstości

Dla m=2

$$g_2(y) = \begin{cases} y & 0 \leq y \leq 1 \\ 2 - y & 1 < y \leq 2 \end{cases}$$

Dla m=3

$$g_3(y) = \begin{cases} \frac{y^2}{2} & 0 \leq y \leq 1 \\ \frac{1}{2}(y^2 - 3(y-1)^2) & 1 < y \leq 2 \\ \frac{1}{2}(y^2 - 3((y-1)^2 + 3(y-2)^2)) & 2 < y \leq 3 \end{cases}$$

Testowanie generatora polega na weryfikacji hipotezy, że zmienna losowa Y ma rozkład zgodny z $g_m(y)$. Zazwyczaj m nie przekracza 5.