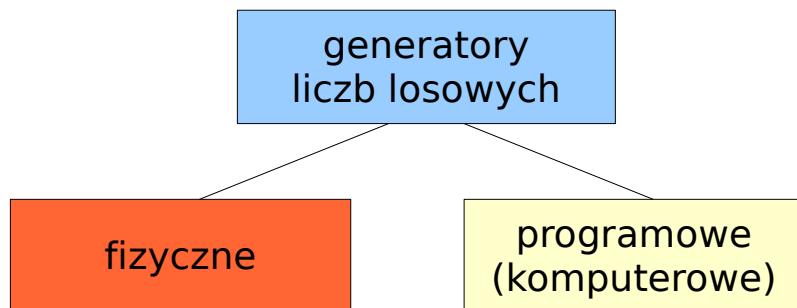


Generatory liczb pseudolosowych

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
 - c) Testy kombinatoryczne

Generatory o rozkładzie równomiernym



Najprostsze generatory liczb losowych to generatory fizyczne wykorzystujące:

- 1) szумы 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)

$$X_0, X_1, X_2, \dots, X_n$$

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

$$m = 2^{32}$$

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

T_a – okres aperiodyczności ciągu

T_o – 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:

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

gdzie:

$a_1, a_2, \dots, a_k, c, m$ - parametry generatora (ustalone liczby)

Operację

$$r = (a \bmod n), \quad a, n, r \in \mathbb{Z}$$

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

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

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

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

Przykład

$$19 \bmod 6 = \mathbf{1}$$

$$18 \bmod 6 = 0$$

$$17 \bmod 6 = 5$$

$$16 \bmod 6 = 4$$

$$15 \bmod 6 = 3$$

$$14 \bmod 6 = 2$$

$$13 \bmod 6 = \mathbf{1}$$

$$12 \bmod 6 = 0$$

Aby wygenerować ciąg liczb pseudolosowych należy zdefiniować jego parametry.

Liczby

$$X_0, X_1, X_2, \dots, X_k$$

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

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

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

$$X_1 = aX_0 - mk_1$$

$$X_2 = a^2 X_0 - mk_2 - mk_1 a$$

$$X_3 = a^3 X_0 - mk_3 - mk_2 a - mk_1 a^2$$

.....

$$X_n = a^n X_0 - m(k_n + k_{n-1}a + \dots + k_1 a^{n-1})$$

Ostatnie równanie można zapisać w postaci

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

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

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

					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 wynosi

$$a^{(m-1)/p} \not\equiv 1 \pmod{m}$$

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

Przykład

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

$$m = 2^p - 1$$

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

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

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

W przypadku **generatora mieszanego** maksymalny okres uzyskamy po spełnieniu warunków:

a) liczby c i m nie mają wspólnych dzielników

b) jeżeli 4 jest dzielnikiem m to

$$a \equiv 1 \pmod{4}$$

c) dla każdego dzielnika pierwszego p liczby m

$$a \equiv 1 \pmod{p}$$

Przykład.

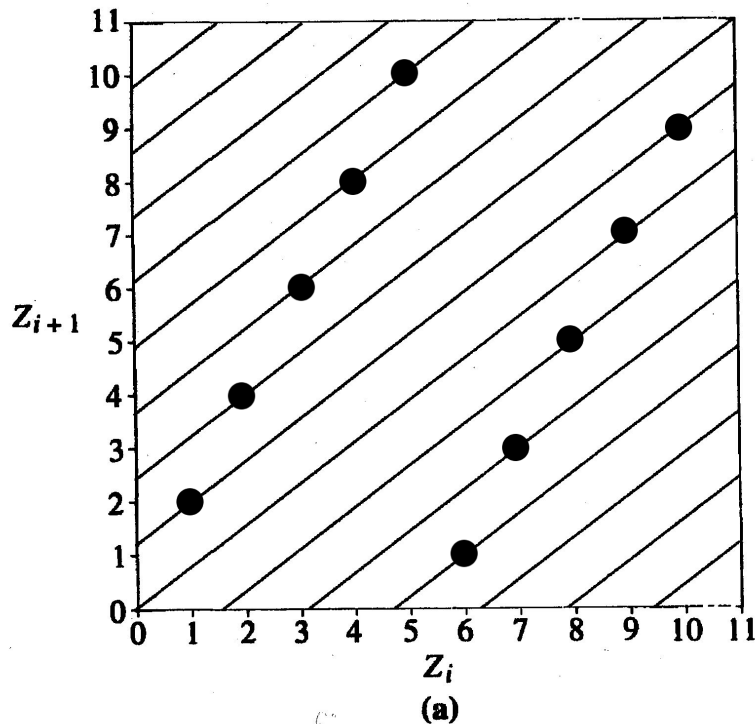
$$a = 69069 \quad c = 1 \quad m = 2^{32}$$

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.

Przykład.

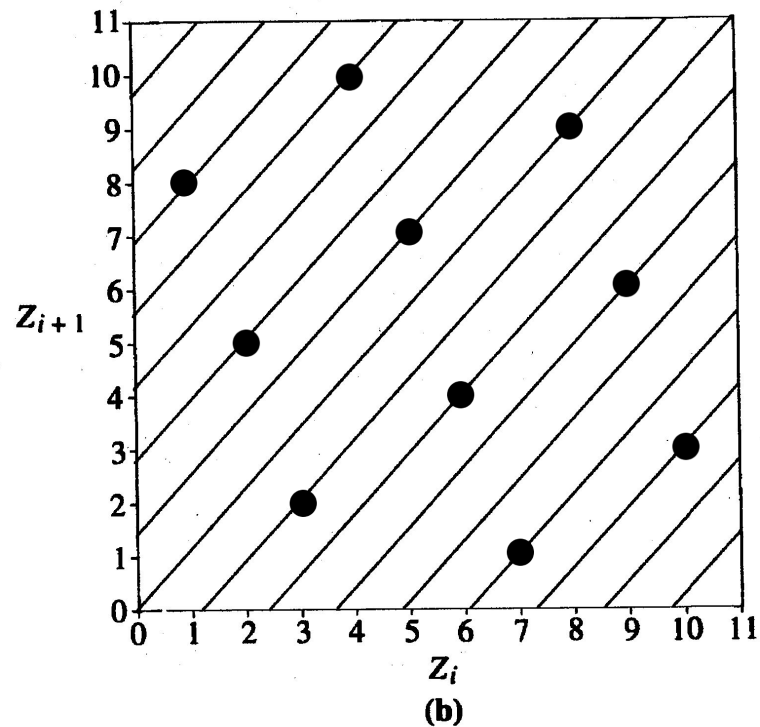
$$X_i = aX_{i-1} \bmod 11$$



a) $X_0=1, a=2$

b) $X_0=1, a=8$

$$(U_1, U_2, \dots, U_d), (U_2, U_3, \dots, U_{d+1}), \dots$$
$$(U_1, U_2, \dots, U_d), (U_{d+1}, U_{d+2}, \dots, U_{2d}), \dots$$



Parametry statystyczne generatora o rozkładzie równomiernym w (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

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

Można nim wygenerować 4 ciągi liczb o okresie

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

Jeden z nich:

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

Ciąg liczb **pseudolosowych**

$$X_0, X_1, X_2, \dots$$

jest permutacją liczb

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

Liczby znormalizowane

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

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

$$\tilde{R}_r = \lim_{n \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} X_i X_{i-r}$$

Oraz wzór dla ciągu skończonego

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

Inaczej: opisuje związek pomiędzy elementami dwóch szeregów - danego i przesuniętego o r .

Efektywnie funkcję autokorelacji można badać przy użyciu FFT - spłot dwóch wektorów.

Dla analizowanego generatora mieszanego oraz dla $r=1$ współczynnik autokorelacji można oszacować z poniższej relacji

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

Przykłady generatorów liniowych

$$X_i = (1176X_{i-1} + 1476X_{i-2} + 1776X_{i-3}) \bmod (2^{32} - 5)$$

$$X_i = 2^{13}(X_{i-1} + X_{i-2} + X_{i-3}) \bmod (2^{32} - 5)$$

$$X_i = (1995X_{i-1} + 1998X_{i-2} + 2001X_{i-3}) \bmod (2^{35} - 849)$$

$$X_i = 2^{19}(X_{i-1} + X_{i-2} + X_{i-3}) \bmod (2^{32} - 1629)$$

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

Generatory na rejestrach przesuwnych

Definiujemy ciąg bitów b_i otrzymywanych rekurencyjnie

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

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

gdzie:

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

są stałymi binarnymi, a stałe b_i

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

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

$$a \text{ xor } b = (a + b) \bmod 2$$

Jeśli założymy

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

to relację rekurencyjną można zapisać przy użyciu xor

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

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:

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

Wykorzystujemy ciąg bitów do obliczenia liczby pseudolosowej z przedziału $(0,1]$

(**generator Tauswortha**)

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

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 przesuwanych oraz bramek logicznych (xor) – łatwa implementacja w języku C.

Generator Fibonacciego

Punktem wyjścia do konstrukcji generatora są liczby Fibonacciego

$$f_0 = f_1 = 1$$

$$f_n = f_{n-2} + f_{n-1}$$

a dokładniej ciąg reszt tego ciągu

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

Powyższy ciąg reszt ma rozkład równomierny ale nie spełnia testów niezależności.

Jego modyfikacja

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

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

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

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 $S = \{1, 2, \dots, n\}$ z rozkładami prawdopodobieństwa

$$P\{X = i\} = r_i$$

$$P\{Y = i\} = q_i$$

$$i = 1, 2, \dots, n$$

Z tych liczb możemy utworzyć wektory

$$\mathbf{r} = (r_1, r_2, \dots, r_n)$$

$$\mathbf{q} = (q_1, q_2, \dots, q_n)$$

Za normę wektora przyjmujemy p-normę

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

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

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

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

$$X_1, X_2, \dots, X_n$$

ma okres T_1 , a ciąg

$$Y_1, Y_2, \dots, Y_n$$

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

$$X_i \diamond Y_j$$

ma okres równy $T_1 T_2$.

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

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

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

$$V_1 = 0.b_1 b_2 \dots b_{24}$$

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

Ciąg bitów generujemy przy użyciu dwóch generatorów

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

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

$$b_n = \begin{cases} 0, & \text{gdy } y_n z_n \bmod 64 < 32 \\ b_n = 1, & \text{w pozostałych przypadkach} \end{cases}$$

Generatory muszą być zainicjowane

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

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

Okres generatora kombinowanego: **2¹²⁰**.

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

$$c_n = c_{n-1} \diamond \left(\frac{7654321}{16777216} \right)$$

$$n \geq 2$$

$$c_1 = \frac{362436}{16777216}$$

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

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

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

Okres tego generatora to 2²⁴-3.

Okres generatora uniwersalnego (U_n): **2¹⁴⁴**

Generator ULTRA - oparty na odejmowaniu z pożyczką

Wykorzystujemy w nim operację

$$x \ominus y \bmod m$$

której wynikiem jest

$$\begin{cases} x - y - c + m & x - y - c < 0 \rightarrow c = 1 \\ x - y - c & x - y - c \geq 0 \rightarrow c = 0 \end{cases}$$

$$c \in \{0, 1\}$$

Parametr c jest tzw. **bitem przeniesienia**. Jego początkowa wartość wynosi 0.

W generatorze ULTRA przyjęto parametryzację

$$L \geq 32, r, s, \quad r > s$$

$$m = 2^L$$

Elementy ciągu wyznaczamy rekurencyjnie

$$X_i = X_{i-r} \ominus X_{i-s} \bmod m$$

Generator inicjalizujemy L-bitowymi liczbami

$$X_1 = b_L b_{L-1} \dots b_1$$

$$X_2 = b_{2L} b_{2L-1} \dots b_{L+1}$$

$$X_r = \dots$$

Ciąg bitów

$$b_1, b_2, \dots$$

tworzymy ze znaków liczb

$$w_1, w_2, \dots$$

które generujemy następująco:

1) Podajemy dwie liczby:

$$u_0, v_0 \in (0, m)$$

2) Kolejno obliczamy

$$u_i = \lambda u_{i-1} \bmod m$$

$$v_i = (v_i \gg k_1) \text{ xor } v_i$$

$$v_i = (v_i \ll k_2) \text{ xor } v_i$$

$$w_i = u_i \text{ xor } v_i$$

gdzie: k_1, k_2 są ustalonymi liczbami.

Operacja $v_i \gg k_1$

oznacza przesunięcie bitów w v_i w prawo i wypełnienie najstarszych k_1 bitów zerami.

Jeśli

$$M = m^r - m^s + 1$$

jest liczbą pierwszą to okres generatora jest równy najmniejszej liczbie naturalnej k dla której spełnione jest poniższe równanie

$$m^k \bmod M = 1$$

W generatorze ULTRA dla parametrów

$$L = 32$$

$$r = 37$$

$$s = 24$$

$$\lambda = 69069$$

Okres generatora wynosi **10³⁴⁶**.

Generator jest bardzo szybki. Pozwala generować liczby całkowite: 32, 31, 8, 7 i 1 bitowe.

Generatory nieliniowe

Zaletą generatorów nieliniowych jest to że ciągi generowanych przez nie liczb nie układają się na hiperpłaszczyznach tak jak w przypadku generatorów liniowych.

Generator nieliniowy Eichenauera-Lehna

$$X_{i+1} = (aX_i^{-1} + b) \bmod m, \quad i = 0, 1, \dots$$

gdzie: m jest liczbą pierwszą

Odwrotność modulo liczymy następująco

$$c = 0 \Rightarrow c^{-1} \bmod m = 0$$

$$c \neq 0 \Rightarrow c \cdot (c^{-1} \bmod m) = 1$$
$$c^{-1} = c^{m-2} \bmod m$$

Uzyskujemy w ten sposób ciąg liczb

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

o rozkładzie równomiernym.

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

Generator Eichenauera-Hermannna

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

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

Dla

$$a \in \{1, 2, \dots, m\}$$

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

$$z^{m^2-1} = 1 \bmod (z^2 - bz - a)$$

Generatory o dowolnym rozkładzie prawdopodobieństwa

Metoda odwracania dystrybuanty

Dystrybuanta określonego rozkładu prawdopodobieństwa jest funkcją

$$F : R \rightarrow R$$

niemalejącą i prawostronnie ciągłą

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

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

Dystrybuanta jednoznacznie definiuje rozkład prawdopodobieństwa.

Związek pomiędzy dystrybuantą a **gęstością prawdopodobieństwa** $f(x)$:

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

Przykład.

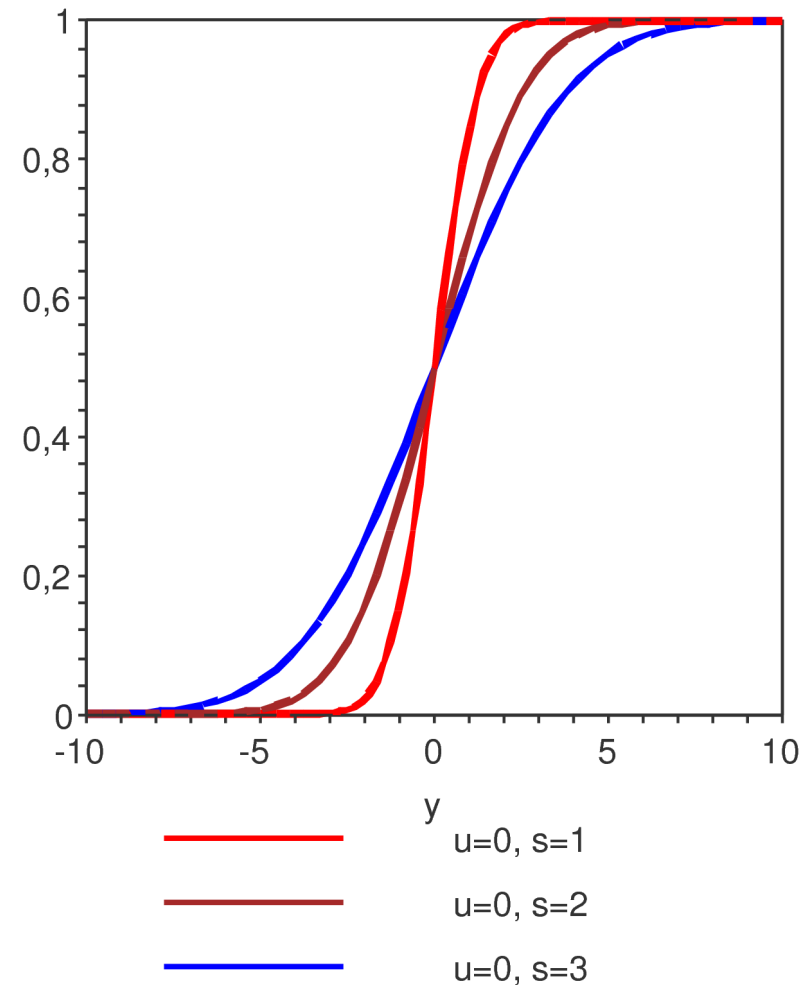
Rozkład wykładniczy

$$f(x) = e^{-\lambda x}$$

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Rozkład normalny

$$F(x) = \int_{-\infty}^x \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) dt$$



Rys. Dystrybuanta rozkładu normalnego.

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

Nową zmienną losową będzie

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

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

Generujemy więc ciąg liczb pseudolosowych

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

który przekształcamy w ciąg

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

Liczby X_i mają rozkład prawdopodobieństwa o dystrybuancie F.

Odwracanie dystrybuanty można wykorzystać także w przypadku rozkładów dyskretnych.

Np. ciąg zmiennych

$$X_1, X_2, \dots, X_n$$

o rozkładzie

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

Można wygenerować przy użyciu ciągu

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

korzystając ze wzoru

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

Odwracanie dystrybuanty sprawia duże trudności numeryczne.

Przykład.

Odwracanie dystrybuanty dla rozkładu normalnego

$$\Phi^{-1}(u) = \begin{cases} g(u) & 10^{-20} < u < 0.5 \\ -g(1-u) & 0.5 \leq u \leq 1 - 10^{-20} \end{cases}$$

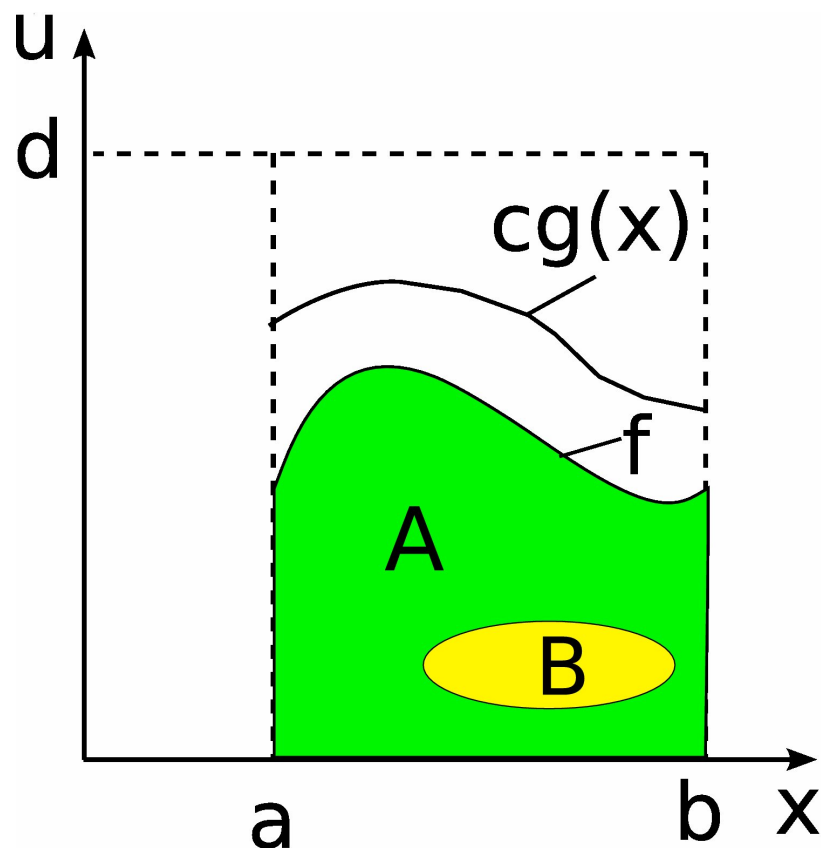
$$g(u) = t - \frac{L(t)}{M(t)}, \quad t = \sqrt{-2 \ln u}$$

gdzie: L(t) i M(t) są wielomianami stopnia 4 zmiennej t.

Zaletą tej metody jest to, że do wygenerowania jednej nowej zmiennej losowej X_n potrzeba tylko jednej zmiennej pierwotnej U_n .

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

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

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 .

Oznaczenia

- pole powierzchni zbioru C (miara Lebesgue'a): $I_2(C)$

- zbiór A

$$A = \{(x, u) : a \leq x \leq b, 0 \leq u \leq f(x)\}$$

Ponieważ $f(x)$ jest gęstością prawdopodobieństwa, więc

$$I_2(A) = 1$$

Pole powierzchni prostokąta $(a,b) \times (0,d)$

$$l_2((a, b) \times (0, d)) = (b - a)d$$

Przy tych założeniach możemy stwierdzić że dowolnie wybrany punkt ω ma rozkład równomierny na zbiorze

$$D \subset R^2$$

jeżeli dla podzbiorów C tego zbioru zachodzi

$$P\{\omega \in C\} = \frac{l_2(C)}{l_2(D)}$$

Tw. Jeśli $(X_1, U_1), (X_2, U_2), \dots$ jest ciągiem punktów losowych o rozkładzie równomiernym w prostokącie $(a,b) \times (0,d)$ oraz (X_i, U_i) będzie pierwszym punktem wpadającym do zbioru A to punkt ten ma rozkład równomierny w A .

Dowód. Rozważamy podzbiór B zbioru A

$$\begin{aligned} P\{(X, U) \in B\} &= \sum_{i=1}^{\infty} P\{(X_1, U_1) \notin A, \dots, \\ &\quad (X_{i-1}, U_{i-1}) \notin A, (X_i, U_i) \in B\} \\ &= \sum_{i=1}^{\infty} \left(1 - \frac{l_2(A)}{(b-a)d}\right)^{i-1} \frac{l_2(B)}{(b-a)d} = \frac{l_2(B)}{l_2(A)} \end{aligned}$$

Tw. Jeżeli U ma rozkład równomierny w $[0,1]$ a X ma rozkład o gęstości $f(x)$ oraz X i U są niezależne, to punkt losowy $(X, Uf(X))$ ma rozkład równomierny w zbiorze A .

Powyższe rozważania można uogólnić na większą liczbę wymiarów.

Ponadto zamiast stałej d ograniczającej $f(x)$ od góry można wprowadzić funkcję ograniczającą $cg(x)$, wówczas warunkiem akceptacji jest

$$cg(X)U \leq f(X), \quad X \in [a, b], \quad U \in [0, 1]$$

gdzie: c jest stałą, a $g(x)$ rozkładem zbliżonym do $f(x)$.

Zabieg taki opłaca się wykonać wówczas gdy wyznaczenie ciągu liczb o gęstości $g(x)$ jest łatwe i szybkie. Funkcję $g(x)$ nazywamy wtedy rozkładem dominującym.
Jak dobrać stałą c ?

$$\begin{aligned} P\{Ucg(X) \leq f(X)\} &= \\ \int_{R^k} g(x) d^k x \int_0^{f(x)/cg(x)} du &= \frac{f(x)}{cg(x)} \end{aligned}$$

$$c = \sup_x \frac{f(x)}{g(x)}$$

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 $g_t(x)$

W praktyce całkę 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: $\mathbf{1}_{A_i}$ jest funkcją przynależności do podzbioru A_i .

Algorytm generacji ciągu zmiennej X jest wówczas następujący:

- 1) Losujemy zmienną losową

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

- 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.

Rozkład dyskretny

a) metoda odwracania dystrybucyj

W rozkładzie dyskretnym mamy określone prawdopodobieństwo wylosowania danej liczby

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

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 przeciwnym wypadku akceptujemy X

4) Kroki 1-3 wykonujemy aż do uzyskania odpowiednio licznego ciągu X_1, X_2, X_3, \dots

b) Metoda równomiernego rozbicia przedziału

Zakładamy że zmienna losowa X przyjmuje określone wartości z pewnym prawdopodobieństwem

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

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

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

Zmienna U wpada do przedziału

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

Konstruujemy dwa ciągi

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

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

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, \dots

Powyższy algorytm zapewnia to, że warunek

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

będzie sprawdzany co najwyżej dwukrotnie.

Generatory o rozkładach wielowymiarowych

Zadanie można sformułować następująco:

Należy wygenerować ciąg wielowymiarowych zmiennych losowych

$$\mathbf{X} = (X_1, X_2, \dots, X_m)$$

których rozkład prawdopodobieństwa ma gęstość

$$f(x_1, x_2, \dots, x_m)$$

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

Przykład.

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

Algorytm.

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

$$\mathbf{U} = (U_1, U_2, \dots, U_m)$$

Zmienną akceptujemy jeśli

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

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

$$p_m = \frac{\pi^{m/2}}{2^m \Gamma(m/2 + 1)}$$

Średnia liczba wylosowanych punktów N_m potrzebnych do realizacji jednej zmiennej wynosi

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

m	p_m	N_m
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 m-wymiarowa zmienna losowa

$$\mathbf{X} = (X_1, X_2, \dots, X_m)$$

ma rozkład równomierny na sferze

$$S_m = \left\{ (x_1, \dots, x_m) : \sum_{j=1}^m x_j^2 = 1 \right\}$$

oraz macierz $A_{m \times m}$ jest macierzą ortonormalną tj. macierzą obrotu w przestrzeni R^m to zmienna losowa $A\mathbf{X}$ ma również rozkład równomierny na tej sferze.

m-wymiarowa zmienna

$$\mathbf{Z} = (Z_1, Z_2, \dots, Z_m)$$

ma rozkład **sferycznie konturowany** jeżeli jej gęstość prawdopodobieństwa

$$g_Z(z_1, z_2, \dots, z_m)$$

zależy tylko od

$$\|z\|^2 = \sum_{j=1}^m z_j^2$$

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

Przykład.

m-wymiarowy rozkład normalny opisuje gęstość

$$g_Z(z_1, \dots, z_m) = \frac{1}{(2\pi)^{m/2}} \exp\left(-\frac{1}{2}\|z\|^2\right)$$

Dla macierzy ortonormalnej zachodzi

$$\|AZ\|^2 = \|Z\|^2$$

więc AZ ma identyczny rozkład.

Algorytm generowania rozkładu równomiernego na sferze w m wymiarach:

1) Generujemy m-wymiarową zmienną losową Z o rozkładzie normalnym

2) Obliczamy

$$X = \frac{Z}{\|Z\|}$$

Rozkład nowej zmiennej X będzie równomierny na sferze S_m .

Rozkład równomierny na sferze S_m

Tw. Jeżeli zmienna

$$Z = (Z_1, Z_2, \dots, Z_k)$$

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

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

a s jest losowym znakiem

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

to (k+1)-wymiarowa zmienna losowa

$$X = (RZ_1, RZ_2, \dots, RZ_k, s\sqrt{1-R^2})$$

ma rozkład równomierny w (k+1) wymiarowej kuli.

Rozkład równomierny na Kuli K_m

Zauważamy że długość wektora wodzącego R zmiennej X jest także zmienną losową, i ma ona rozkład

$$h(r) = mr^{m-1}, \quad 0 \leq r \leq 1 \quad 23$$

Wystarczy więc wygenerować punkt

$$\mathbf{Z} = (Z_1, Z_2, \dots, Z_m)$$

leżący na sferze S_m a następnie obliczyć

$$\mathbf{X} = (RZ_1, RZ_2, \dots, RZ_m)$$

Zmiana \mathbf{X} leży wewnątrz kuli jednostkowej K_m i ma w tym obszarze rozkład równomierny.

Dwuwymiarowy rozkład normalny (metoda Boxa-Mullera)

Jeżeli dwie niezależne zmienne losowe U_1 i U_2 mają rozkład równomierny w przedziale $[0,1]$ to zmienne losowe

$$X_1 = \sqrt{-2\ln(U_1)}\cos(2\pi U_2)$$

$$X_2 = \sqrt{-2\ln(U_1)}\sin(2\pi U_2)$$

również są niezależne a prawdopodobieństwo ich otrzymania opisuje rozkład normalny o parametrach

$$\mu = 0$$

$$\sigma = 1$$

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 kolejnych 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, \dots, T_N . Jeśli weryfikowana hipoteza jest prawdziwa to

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

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

Testy zgodności z rozkładem równomiernym $U(0,1)$

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

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

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

Generujemy n liczb

$$X_1, X_2, \dots, X_n$$

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

$$X_1, X_2, \dots, X_n$$

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

$$I_1, I_2, \dots, I_n$$

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

Następnie tworzymy ciąg kolejnych nakładających się par

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

Jeśli przez Y oznaczmy liczbę takich par

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

które nie pojawiły się w ciągu (I_i, I_{i+1}) , to ta zmienna ma rozkład normalny $N(\mu, \sigma)$ - 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

$$X_1, X_2, \dots, X_n$$

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

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

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

Możemy zatem zdefiniować pewną funkcję

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

określoną w kostce jednostkowej.

W ten sposób tworzymy ciąg nowych zmiennych losowych

$$Y_j = h(X_{(j-1)m+1}, X_{(j-1)m+2}, \dots, X_{jm})$$

$$j = 1, 2, \dots$$

o jednakowym rozkładzie i dystrybuancie

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

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

Testy oparte na statystykach pozycyjnych

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

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

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

$$r = u - v$$

co generuje nowe zmienne

$$U, V, R$$

Nowe zmienne mają następujące rozkłady

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

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

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

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

Test sum

Definiujemy funkcję

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

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.

Testy kombinatoryczne

Pozwalają sprawdzić czy dostarczane przez generator liczby są niezależnymi zmiennymi losowymi.

W testach tych sprawdza się hipotezę że wielowymiarowa zmienna losowa utworzona z ciągu liczb

$$(X_1, X_2, \dots, X_n)$$

ma rozkład o dystrybuancie

$$H(x_1, x_2, \dots, x_n) = F(x_1)F(x_2) \dots F(x_n)$$

Do tej grupy testów zaliczają się np: test pokerowy, test kolekcjonera, test kolizji, test permutacji.