PSK zajęcia zaplanowane na 16.04

Temat: Generowanie rozkładów prawdopodobieństw,

Wracamy na chwilę do problemu generowania liczb (pseudo)losowych. Jak pamiętamy, generalnie sposoby generowania takich liczb dzielimy na generatory fizyczne i generatory programowe.

Poświęćmy chwilę uwagi tym pierwszym - od nich wszystko zaczęło się!

Problem „igły Buffona" i jego konsekwencje

W 1777 roku Georges-Louis Leclerc, Comte de Buffon pokazał (szczegóły* można przeczytać np. w [2] załączonej literatury do kursu), że jeśli igła o długości L jest rzucana przypadkowo na równoległe (na płaszczyźnie) odległe o d linie, to prawdopodobieństwo, że igła upadnie na jedną z takich linii jest równe $\frac{2L}{\pi d}$ ($d \geq L$)

W szczególności gdy $d=L$ wynosi $\frac{2}{\pi}$

Ziarnko do „dobra" zrozumiał P. Laplace, który w 1820 r. zaproponował eksperyment zadaniem, którego jest kontrola frakcji $\frac{M}{N}$, gdzie N - liczba rzutów igły

* polecam też mój esej o liczbie π.

M – liczba przecięć linii, o których mówi problem Buffona.

Laplace wykazał (był to pierwszy istotny wynik ówczesnej t. prawdopodobieństwa!), że

$$\lim_{N\to\infty} \frac{M}{N} = \frac{2L}{\pi d} \text{, a więc}$$

$$\pi = \lim_{N\to\infty} \frac{2NL}{Md} \quad \text{lub}$$

$$\pi = \lim_{N\to\infty} \frac{2N}{M} \quad (\text{gdy } d = L).$$

Znaleźli się tacy, którzy chcieli to potwierdzić eksperymentalnie – to był przykład pierwszego generatora fizycznego.

Niejaki Mario Lazzarini pochwalił się (chyba) najlepszym wynikiem (≡ wyfantazjowanym!).

W 1901 rzucał 3408 razy igłą o dł. 3 cm i uzyskał wynik $\pi = 3.1415929$ (ZDUMIEWAJĄCE!).

ZADANIE 1.

Proszę zasymulować przebieg tego generatora, powiedzmy dla $N = 100$ i przedstawić wyniki.

Generowanie dyskretnych rozkładów prawd.

Załóżmy, iż chcemy wygenerować wartości dyskretnej zmiennej losowej X, gdzie jej rozkład dp jest następujący:
$$P(\{\omega\in\Omega : X(\omega)=x_j\}) = p_j, \quad j = 1, 2, \dots, \quad \sum_j p_j = 1.$$

Procedura jest następująca:

Niech $U(\omega)$ oznacza liczbę (pseudo)losową, a więc $U^{(\omega)}\in(0,1)$ wygenerowaną np. algorytmem opisanym na poprzednim wykładzie, czyli $U\in\mathcal{U}([0,1])$.

Należy ~~[illegible]~~ podać przepis na $X(\omega)\in\{x_j, j=1,2,\dots\}$, czyli odpowiedzi na pytanie:

„Kiedy $X(\omega)=x_j$ dla danego j?”

Zasada jest następująca:

Definiujemy zmienną losową $\tilde{X}$

$$\tilde{X}(\omega) = \begin{cases} x_1, & \text{o ile } U(\omega) \leq p_1 \\ x_2, & \text{o ile } p_1 < U(\omega) \leq p_1 + p_2 \\ \vdots \\ x_j, & \text{o ile } \sum_{i=1}^{j-1} p_i < U(\omega) \leq \sum_{i=1}^{j} p_i \\ \vdots \end{cases}$$

Dla $a, b \in \mathbb{R}$: $0 < a < b < 1$

$$P(\{\omega \in \Omega : a \leq U(\omega) < b\}) = b - a,$$

więc

$$P(\{\omega \in \Omega : \tilde{X}(\omega) = x_j\}) =$$

$$= P\left(\left\{\omega \in \Omega : \sum_{i=1}^{j-1} p_i \leq U(\omega) < \sum_{i=1}^{j} p_i\right\}\right) = p_j,$$

co dowodzi, iż $d(\tilde{X}) = d(X)$, a więc procedura działa poprawnie!

Mamy zatem algorytm:

Zastadny wartości $\{p_j\}$ i $\{x_j\}$
generuj (pseudo)losową liczbę U
Jeśli $U \leq p_1$ przyjmij $X = x_1$ i STOP
Jeśli $U \leq p_1 + p_2$ przyjmij $X = x_2$ i STOP
itd.

Przedstawiona wyżej metoda nazywana jest METODĄ ODWZOROWANIA ODWROTNEGO (MOO)

Przykład 1

Chcemy MOO symulować zm. losową X:

	1	2	3	4
d_X	0,2	0,15	0,25	0,4

Wtedy mamy:

Jeśli $U \leq 0{,}20 \Rightarrow X = 1$ & STOP
$U \leq 0{,}35 \Rightarrow X = 2$ & STOP
$U \leq 0{,}6 \Rightarrow X = 0{,}25$ & STOP
w przeciwnym razie $X = 4$ & STOP

Szukajmy, ale w tym przypadku omawianej sytuacji, pi skończony rozkład dyskretny jednostajny,

cyli $P(\{\omega\in\Omega : X(\omega) = j\}) = \frac{1}{m}$

$j = 1, 2, \ldots, m$

Po wygenerowaniu (pseudo) losowej liczby $U(\omega)$, z MOO mamy:

$$X(\omega) = j, \text{ jeśli } \frac{j-1}{m} < U(\omega) \le \frac{j}{m},$$

co oznacza, że

$$j-1 < mU(\omega) \le j.$$

Zatem $j = [mU(\omega)] + 1$, gdzie (*)

$[x]$ – największa l. całkowita nie przekraczająca x: $[0,7] = 0$, $[1,5] = 1$ itd.

Przykład 2 (Generowanie PERMUTACJI)

Przypomnijmy (matematyka dyskretna!), dla ustalonego zbioru $A = \{a_1, a_2, \ldots, a_n\}$ mamy $n = |A|$

każdą funkcję (a nie relację)

$$f: A \xrightarrow[na]{1-1} A$$

wzajemnie jednoznaczną (1-1) i typu „na" nazywamy permutacją zbioru A.

Jest to nic innego jak jakieś ustawienie wszystkich ele. zb. A w ciąg długości n.

Np. $A = \{a, b, c\}$

(a,b,c), (c,b,a) itd. są przykładami permutacji.

Niech $P_n(A)$ — zbiór wszystkich permutacji zb. A.

Dobrze wiadomo, że $|P_n(A)| = n!$ $(= 1 \cdot 2 \cdot \ldots \cdot n)$.

Możemy losować na $P_n(A)$, $|A| = n$ w sposób następujący:

[f wybieramy z $P_n(A)$ z prawdop. $\frac{1}{n!}$.]

Pokażemy jak zastosować wynik (*) do generowania $f \in P_n(A)$

Pomysł na realizację tego projektu jest następujący:

Krok 1 (inicjujący)

Niech $(P_1, P_2 \ldots, P_n)$ będzie jakąś permutacją zbioru $A=\{1,2\ldots n\}$

Krok 2. W (*) bierzemy $k=n$, czyli $[nU]+1$

Krok 3 generujemy liczbę (pseudo)losową U i przyjmujemy $i=[nU]+1$

Krok 4 Zamieniamy miejscami $P_i \longleftrightarrow P_n$

Krok 5. zmniejszamy licznik $k=n \longrightarrow n-1$ i jeśli (dopóki) $k>1$ idziemy do kroku 3

Krok 6 dostajemy daną permutację.

Przykład 3 (generowanie permutacji)

Niech $n=4$, z permutacją inicjującą $(1,2,3,4)$ (K1)

Zatem, gdy U jest dane, i (K3)

$i = [4U]+1 = 3$

Wtedy (K4): $(1,2,4,3)$ i $k=3$

Jeśli teraz $i=[3U]+1 = 2$, to

$(1,4,2,3)$ i $k=2$.

Ostatni przebieg, jeśli teraz $i=[2U]+1=2$, to $i=k$ i dlatego na output mamy

$(1,4,2,3)$

ZADANIE 2

Generując $U \in (0,1)$, metodą MOO zasymulować zm. losową

X	-1	0	1
	$0{,}3$	$0{,}4$	$0{,}3$

ZADANIE 3

generując $U\in(0,1)$ metodą RNG zasymuluj zm. losową o rozkładzie jednostajnym o wartościach $\{1,2,3,4,5,6\}$.

ZADANIE 4

Wygeneruj permutacje zbioru $\{1,2,3,4,5\}$ zaczynając od $(5,4,3,2,1)$.

Inne rozkłady dyskretne

Do tej pory rozważaliśmy sytuację ogólną rozkładu dyskretnego. Teraz omówimy przypadki szczególne takich rozkładów

(I) X ma rozkład geometryczny, czyli

$$P(\{\omega\in\Omega: X(\omega)=i\})=pq^{i-1},\quad i\geq 1$$

$$q=1-p$$

Jak wiemy z wcześniejszego wykładu

$X(\omega)$ oznacza, iż dokonujemy serii powtórzeń (niezależnych) pojedynczego zjawiska w kategoriach „SUKCES" „PORAŻKA". Wtedy $X(\omega)$ to chwila (liczba powtórzenia) osiągnięcia pierwszego SUKCESU.

Zaadaptujemy dla tego przypadku MOO:

W tym celu obliczmy

$$\sum_{k=1}^{i} p_k = \sum_{k=1}^{i} P(\{\omega\in\Omega : X(\omega)=k\}) =$$

$$= P(S_{\le i}) = 1 - P(S_i^c) =$$

pierwszy sukces
pojawi się nie później
aniżeli i

$$= 1 - P(\{\omega\in\Omega : X(\omega) > i\})$$

„do chwili i same porażki”

$$= 1 - q^i, \quad i \geq 1,$$

gdzie $q = 1-p$, p – prawd. sukcesu.

Stąd warunek

$$\sum_{k=1}^{i-1} p_k < U(\omega) \le \sum_{k=1}^{i} p_k \quad \text{ma postać:}$$

$$1-q^{i-1} < U(\omega) \le 1-q^i$$

Lub równoważnie

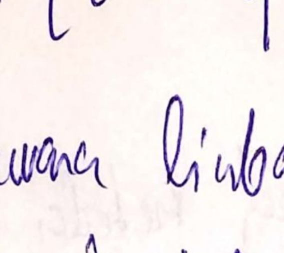
$$q^i \le 1-U(\omega) < q^{i-1}.$$

Pozwala to nam zdefiniować $X(\omega)$:

$$X(\omega) = \min\{ i : q^i \le 1-U(\omega) \}, \text{ gdzie}$$

$U(\omega)$ wygenerowana liczba (pseudo)losowa.

Okazuje się, że <u>lepiej jest</u> przejść na <u>skalę logarytmiczną</u>. Ponieważ funkcja $x \to \log x$ jest rosnąca,



Z powyższego

$$X(\omega) = \min\{ i : i\log(q) \le \log(1-U(\omega)) \}$$

$$= \min\left\{ i : i \ge \frac{\log(1-U(\omega))}{\log(q)} \right\}, \text{ bo}$$

$q \in (0,1)$.

Oznacza to, że

$$X(\omega) = \left[\frac{\log(1-U(\omega))}{\log q} \right] + 1 .$$

Ale $1-U$ też jest liczbą (pseudo)losową, więc możemy $1-U$ podmienić U i ostatecznie

$$\boxed{X(\omega) = \left[\frac{\log(U(\omega))}{\log q} \right] + 1}$$

Przykład. Pokażmy jak wykorzystać wynik z (I) do wygenerowania ciągu $(X_1, X_2 \ldots, X_n)$ zm. losowych, niezależnych o jednakowym rozkładzie $\frac{0}{q} \Big| \frac{1}{p}$ $p \in (0,1)$.

Wtedy $(X_1, X_2 \ldots, X_n)(\omega)$ jest ciągiem binarnym !!!

Zatem problem sprowadza się do generowania ciągów binarnych, gdzie wyjęcie pojawiają się w warunkach stochastycznej niezależności.

Aby przedstawić sposób generowania takiej serii dla $0 < p \leq 0{,}5$, skorzystamy z wyników (I):

Niech N oznacza zmienną losową o rozkładzie geometrycznym, gdzie $N(\omega) = \#$ pierwszego sukcesu w próbie $(X_1, X_2, \ldots, X_n)$, gdzie p - prawd. sukcesu, wygenerowanym jak w (I).

Załóżmy, że wartość symulacji $N(\omega) = j$.

Jeśli $j > n$, to przyjmujemy $X_i^{(\omega)} = 0$, $i = 1 \ldots n$, czyli same porażki.

Jeśli $j \leq n$, to $X_1(\omega) = \ldots = X_{j-1}(\omega) = 0$,

$X_j(\omega) = 1$

Dopóki (While) $j < n$, powtarzamy krok powyższy generując wartości dla pozostałych $n - j$ zmiennych.

Dlaczego $p \leq 0{,}5$? Bowiem interesuje nas czas osiągnięcia pierwszego sukcesu (w pojedynczej próbie z $p \leq 0{,}5$ jesteśmy na przegranej pozycji).

Zatem dla $p>0,5$ zmieniamy strategię i interesuje nas czas osiągnięcia pierwszej porażki.

ZADANIE 4

Niech X ma rozkład geometryczny z $p=0,3$.

Wygeneruj zmienną X

ZADANIE 5.

Bierzemy serię prób ZL: $\begin{array}{c|c} 0 & 1 \\ \hline q & p \end{array}$,

gdzie $p=0,3$, gdzie próby zachodzą niezależnie, i $n=5$, tj.

$$(X_1, X_2, X_3, X_4, X_5),$$

$$d(X_i): \begin{array}{c|c} 0 & 1 \\ \hline 0,7 & 0,3 \end{array}$$

Wygeneruj $(X_1, X_2, X_3, X_4, X_5)(\omega)$.

II Generowanie $X \in \mathcal{P}(\lambda)$

Mamy $p_i = P(\{\omega \in \Omega : X(\omega) = i\}) = e^{-\lambda} \frac{\lambda^i}{i!}$

$i = 0, 1, 2, \ldots$

Pokażemy jak w tym przypadku zaadaptować MOO.

Skorzystamy z pewnej własności tego rozkładu:

$$p_{i+1} = P(\{\omega \in \Omega : X(\omega) = i+1\}) = \frac{\lambda}{i+1} P(\{\omega \in \Omega : X(\omega) = i\}),$$

czyli $\boxed{p_{i+1} = \frac{\lambda}{i+1} p_i}$ $\quad i \geq 0$

Niech F oznacza dystrybuantę X, ale

$$F(i) = P(\{\omega \in \Omega : X(\omega) < i\}) = \sum_{j=0}^{i-1} P(\{\omega \in \Omega : X(\omega) = j\}).$$

Wtedy algorytm generowania X metodą MOO ma postać:

Krok 1: generuj (pseudo)liczbę U

Krok 2: $i = 0$, $p = e^{-\lambda}$, $F = F(0) = p$

Krok 3 Jeśli $U < F$, $X = i$ & STOP

Krok 4 $p = \frac{\lambda p}{i+1}$, $F = F + p$, $i = i + 1$

Krok 5 Idź do kroku 3.

ZADANIE 6

Zaprogramować powyższy algorytm i dokonać symulacji dla $x = 2,0$

(III) Generowanie $X \in B(n, p)$.

Mamy:

$$P(\{\omega \in \Omega : X(\omega) = i\}) = \binom{n}{i} p^i (1-p)^{n-i}$$

$$i = 0, 1, 2, \ldots, n$$

Zastosujmy jak wyżej MOO. W tym celu zauważmy, że

$$P(\{\omega \in \Omega : X(\omega) = i+1\}) = \binom{n}{i+1} p^{i+1} (1-p)^{n-(i+1)} =$$

$$= \frac{n-i}{i+1} \frac{p}{1-p} P(\{\omega \in \Omega : X(\omega) = i\})$$

Niech $F = F(i) = P(\{\omega \in \Omega : X(\omega) < i\})$

Wtedy algorytm generowania X ma postać (modyfikacja dla $P(\lambda)$!)

<u>Krok 1</u>: generuj liczbę (pseudo)losową U

<u>Krok 2</u>: $c = \frac{p}{1-p}$, $i=0$, $pr = P(\{\omega \in \Omega : X(\omega)=i\})$
$= (1-p)^n$

$F = pr$

<u>Krok 3</u>. Jeśli $U < F$, to $X = i$ & STOP

<u>Krok 4</u> $pr = \frac{c(n-i)}{i+1} pr$, $F = F + pr$,
$i = i+1$

<u>Krok 5</u> Idź do Kroku 3.

<u>ZADANIE 7</u>

Zaimplementować powyższy algorytm dla $p = 0{,}3$.