ECHANTILLONS DISTINCTS TIRES D'UNE COLLECTION DE BOULES COLOREES

par Victor A. Souline
Eugène H. Lehman
Département de Mathématiques
et Jerzy Sochanski
Centre de Calcul
Université du Québec à Trois-Rivières

Considérons un ensemble de N boules colorées dans une urne; on a N_i boules de couleur $i = 1,\ldots,c$ où c représente le nombre de couleurs. Donc $N_1 + \ldots + N_c = N$. On en prélève un échantillon de taille n (fixe) $\leq N$, dont n_i (aléatoires) boules de couleur i; $n_1 + \ldots + n_c = n$.

Soit $x_n = (N_1,\ldots,N_c)\ S\ (n,N)$ le nombre d'échantillons distincts de taille n qu'on peut tirer de l'urne (avec remise après chaque tirage de n boules); $N_i \geq N_{i+1}$.

Par exemple: supposons que $c = 5$ couleurs et $N_1 = 6$, $N_2 = 3$, $N_3 = 2$, $N_4 = 1$, $N_5 = 1$. Alors $N = 6 + 3 + 2 + 1 + 1 = 13$ et $n = 0,\ldots,13$. Soit $n = 4$: $x_4 = (6,3,2,1,1,)\ S\ (4,13)$ = le nombre d'échantillons différents de taille 4 qu'on peut distinguer.

On peut simplifier la notation en omettant les N_i qui égalent 1: $x_4 = (6,3,2)\ S\ (4,13)$. Puisque $N = 13$, il est bien sous entendu que si $N_1 + N_2 + N_3 = 11$, alors $N_4 = \ldots = N_c = 1$, où c sera tel que $N = 13$.

Soit une "partition ordonnée" notée po de n en c une façon de former n en choisissant $0 \leq n_i \leq \min(n,N_i)$ boules de couleur $i = 1,\ldots,c$. Alors (0,3,1,0,0), (3,1,0,0,0), (1,3,0,0,0) et (0,1,1,1,1) sont 4 distinctes po de 4 en 5. Noter que (0,0,2,0,2) n'est pas une po de 4 en 5 dans notre cas parce que $n_5 = 2 > N_5 = 1$.

Alors x_4= le nombre de po de 4 en 5 étant donné les N_i. Le but de cet article est de développer un programme efficace pour les ordinateurs qui déterminent x_n pour tout ensemble $(N_1,\ldots,N_c,c,n)$.

Or, étudions le produit des polynômes en t:
$(1 + t + \ldots + t^{N}) \ldots (1 + t + \ldots + t^{Nc}) = a_0 + a_1 t + \ldots + a_N t^N$. Il est évident que $a_0 = a_N = 1$ ainsi que $a_1 = a_{N-1} = c$. Considérons les vecteurs des coefficients des polynômes-facteurs: $v^1 = (1\ 1 \ldots 1)$ avec $N_1 + 1$ éléments unité, ... $v^c = (1\ 1 \ldots 1)$ avec $N_c + 1$ éléments unité. Chaque terme t^n du produit des polynômes est formé en choisissant un des éléments unité dans chacun des c vecteurs de sorte que l'exposant du t associé avec l'élément unité choisi n'excède pas n, et la somme des exposants associés aux éléments unité choisis est n. Autrement dit, chaque terme t^n est le nombre de po de n en c possible, compte tenu des N_i: soit, x_n. On observe que le total d'échantillons possibles pour $n = 0,\ldots,N$
est $\Sigma a_n = (N_1 + 1)(N_2 + 1)\ldots(N_c + 1)$

Citant encore l'exemple x_4 + (6,3,2) S (4,1,3), on a
$(1 + t + t^2 + t^3 + t^4 + t^5 + t^6)(1 + t + t^2 + t^3)(1 + t + t^2)(1 + t)(1 + t) = 1 + 5t + 13t^2 + 24t^3 + 35t^4 + 43t^5 + 47t^6 + 47t^7 + 43t^8 + 35t^9 + 24t^{10} + 13t^{11} + 5t^{12} + t^{13}$. On constate que $x_4 = 35$. La somme des x_n (ou des a_n) $= 2(1 + 5 + 13 + 24 + 35 + 43 + 47) = 7 \times 4 \times 3 \times 2 \times 2 = 168$. La symétrie peut s'expliquer en indiquant que si nous choisissons un échantillon de taille N - n c'est comme si nous avions choisi un échantillon de taille n pour élimination. Alors $x_{N-n} = x_n$.

Pour programmer la multiplication des polynômes, nous définissons le * ou "produit-étoile" de deux vecteurs comme suit: soient $u = (u_1 \ldots u_r)$, $v = (v_1 \ldots v_s)$, et $w = (w_1 \ldots w_t)$. Alors

$$u * v = \sum_{i=1}^{r} u_i(0 \ldots 0 v_1 \ldots v_s 0 \ldots 0)$$ où la première

suite de zéros contient (i - 1) zéros de la dernièrs suite en contient r - i.

Alors $u * v$ est de dimension $r + s - 1$. Par exemple: $(1\ 3\ -1\ 2) * (2\ 1\ -2)$
$= 1(2\ 1\ -2\ 0\ 0\ 0) + 3(0\ 2\ 1\ -2\ 0\ 0) - 1(0\ 0\ 2\ 1\ -2\ 0) + 2(0\ 0\ 0\ 2\ 1\ -2)$
$= (2\ 7\ -1\ -3\ 4\ -4)$ qui est le vecteur des coefficients de
$(1 + 3t - t^2 + 2t^3)(2 + t - 2t^2) = 2 + 7t - t^2 - 3t^3 + 4t^4 - 4t^3$.

On observe que $u * v = (u_1 v_1 \quad u_2 v_1 + u_1 v_2 \quad \ldots \quad u_r u_s)$ dont le k-ième élément est:

$$\sum_{a=1}^{s+r-1} u_a v_{k-a+1} = \sum_{b=1}^{s+r-1} u_{k-b+1} v_b$$

où $u_q = v_q = 0$ si $q < 1$ et $u_q = 0$ si $q > r$ et $v_q = 0$ si $q > s$.

Puisque le produit-étoile représente la multiplication ordinaire des polynômes il est évident que $u * v = v * u$ (commutatif) et que $(u * v) * w = u * (v * w)$ où la longueur du produit triple est de $(r + s - 1) + t - 1$ éléments, (associatif); c'est-à-dire le m-ième élément du produit-triple peut être écrit de deux façons:

$$\sum_{b=1}^{s+r+t-2} \left[\sum_{a=1}^{s+r-1} u_a v_{b-a+1} \right] w_{m-b+1} = \sum_{b=1}^{s+r+t-2} u_b \left[\sum_{a=1}^{r+t-1} v_a w_{(m-b+1)-a+1} \right]$$

ou

$$\sum_{b=1}^{s+r+t-2} \sum_{a=1}^{s+r-1} u_a v_{b-a+1} w_{m-b+1} = \sum_{b=1}^{s+r+t-2} \sum_{a=1}^{r+t-1} u_b v_a w_{m-b-a+2}$$

que l'on peut prouver directement.

Soit maintenant le vecteur $v^i = (1 \ldots 1)$ avec $(N_i + 1)$ éléments unité. Alors le vecteur $x = (x_0 \ldots x_N)$ sera $x = v^1 * v^2 * \ldots * v^c$, de dimension $(N_1 + 1 + N_2 + 1 - 1) + (N_3 + 1 - 1) + \ldots + (N_c + 1 - 1) = N + 1$. Le programme-Sochanski donne x en quelques secondes pour quasiment n'importe quelle combinaison de N_i et c. Par exemple, pour calculer x dans l'exemple déjà cité, avec $N_1 = 6$, $N_2 = 3$, $N_3 = 2$ et $N = 13$:

$x = (1\ 1\ 1\ 1\ 1\ 1\ 1) * (1\ 1\ 1\ 1) * (1\ 1\ 1) * (1\ 1) * (1\ 1)$
$\quad = [(1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0) + (0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0) + \ldots + (0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)]$
$\quad\quad * (1\ 1\ 1) * (1\ 1) * (1\ 1)$
$\quad = [(1\ 2\ 3\ 4\ 4\ 4\ 4\ 3\ 2) * (1\ 1\ 1)] * [(1\ 1) * (1\ 1)]$.

$= [(1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0) + 2(0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0) + \ldots$

$\quad + (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1)] * [(1\ 1\ 0) + (0\ 1\ 1)]$

$= (1\ 3\ 6\ 9\ 11\ 12\ 12\ 11\ 9\ 6\ 3\ 1) * (1\ 2\ 1)$

$= (1\ 2\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0) + 3(0\ 1\ 2\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0) + \ldots$

$\quad + (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 2\ 1)$

$= (1\ 5\ 13\ 24\ 35\ 43\ 47\ 43\ 35\ 24\ 13\ 3\ 1).$

Le programme est ci-après. Le temps pris en calculant x pour $N_1 = 6$ $N_2 = 3$ $N_3 = 2$ $N_4 = 1$ $N_5 = 1$ et $c = 5$ était de 0,080 seconde.

Voici d'autres exemples:
x pour (5 2 3) S (n 20) = (1 10 49 159 389 772 1300 1910 2490 2909 3062 2909 2490 1910 1300 772 389 159 49 10 1); temps de calcul: 0,121 sec.. Ainsi, x pour (9 7 5 3 2) S (n 30) = (1 9 41 128 311 631 ... 7787 7959 7787 ... 631 ... 9 1); temps de calcul: 0,175 sec.. Même x pour (20 13 12 7 5 3 2) S (n 100) = (1 45 997 14504 ... entiers > 10^{15} ... 997 45 1) ne prit que 1,425 sec..

PROGRAMME DE SOCHANSKY POUR DETERMINER $x = (N\ ,\ldots,N_c)\ S\ (n,N)$

```
                 PROGRAM TEST2(INPUT,OUTPUT,TAPE1=INPUT)
000003           INTEGER A(3000,C(3000),BIN(3000),NO(100),B(3000),D(2)
000003           INTEGER Q,P,S,R,SQ,RP
000003     20    CONTINUE
000003           CALL SECOND(T)
000005           PRINT 50,T
000013     50    FORMAT(/,*   TEMPS    *,F10.3,  *  SEC.*,/)
000013           D(1)=D(2)=1
000016           DO 11 J=1,3000
000017     11    A(J)=B(J)=1
000023           READ 1,K1
000031           IF(EOF,1) 30,40
000034     40    CONTINUE
000034      1    FORMAT(I2)
000034           READ 2,(NO(J),J=1,K1)
000047           READ 2,M
000055           K2=0
000056           DO A=1,K1
000060      8    K2=K2+NO(J)
000064           K2=M-K2
000065      2    FORMAT(16I5)
000065           PRINT 10,(NO(J),J=1,K1)
```

```
000100          PRINT 10,M
000106   10     FORMAT(3X,10I10)
000106          S=NO(1)+1
000110          DO 3 J=2,K1
000112          Q=NO(J)=1
000114          SQ=S+Q-1
000116          CALL MULT(A,Q,1,B,S,1,C,SQ,1,BIN)
000127          PRINT 12
000133          PRINT 9,(C(K),K=1,SQ)
000146          DO 4 K=1,SQ
000150    4     B(K)=C(K)
000154          S=SQ
000154    3     CONTINUE
000157          DO 5 J=1,K2
000161          SQ=S+1
000163          CALL MULT(D,2,1,B,S,1,C,SQ,1,BIN)
000175          DO 6 K=1,SQ
000177    6     B(K)=C(K)
000203          S=SQ
000203    5     CONTINUE
000206          PRINT 12
000212    7     FORMAT(1H1)
000212          PRINT 9,(B(K),K=1,SQ)
000225          CALL SECOND(T)
000227          PRINT 50,T
000235          GO TO 20
000236    9     FORMAT(5X,5I20)
000236   12     FORMAT(//)
000236   30     STOP
000240          END
```