NOTE 15 – UN EXERCICE DE PROBABILITÉ ORIGINAL

www.daniel-saada.eu

PAUL BARBAROUX a inventé l'exercice de probabilité suivant:

On tire à pile ou face une infinité de fois avec une pièce équilibrée. Quelle est la probabilité que pour tout $n \geq 1$ on obtienne au moins un pile parmi les tirages $n+1$ à $2n$?

On appelle $(X_n)_{n\geq 1}$ la suite des lancers, en convenant que $X_n = 1$ si pile sort au n[ième] lancer et $X_n = 0$ si c'est face ; A_n est l'événement $\sum_{n+1}^{2n} X_k \geq 1$, sa probabilité est $P(A_n) = 1 - 0,5^n$.

Soit $E_N = \bigcap_{n\geq N} A_n$ et $p_N = P(E_N)$: on cherche p_1 .

La suite p_n est évidemment croissante, sa limite est 1 car

$$1 - p_N = P\left(\bigcup_{n\geq N} \overline{A_n}\right) \leq \sum_{n\geq N} P(\overline{A_n}) = \sum_{n\geq N} 0,5^n = 2/2^N \to 0 .$$

La suite (p_n) vérifie la relation de récurrence $p_n = \sum_{k=1}^{n} p_{n+k}/2^k$ pour tout $n \geq 1$

En effet , E_n est la réunion *disjointe* des n intersections :

– $X_{n+1} = 1$ et E_{n+1} ,

– $X_{n+1} = 0$ et $X_{n+2} = 1$ et E_{n+2} ,

– .... ,

– $X_{n+1} = X_{n+2} = ... = X_{2n-1} = 0$ et $X_{2n} = 1$ et E_{2n} ,

dont les probabilités respectives sont $\frac{p_{n+1}}{2}$, $\frac{p_{n+2}}{2^2}$,..., $\frac{p_{n+n}}{2^n}$ en raison de l'indépendance des lancers.

Premier calcul numérique[1] de p_1

On se donne n et on appelle $(q_i)_{1\leq i\leq 2n}$ la suite *finie* vérifiant la relation de récurrence et définie par $q_i = 1$ pour tout $n+1 \leq i \leq 2n$: $q_i = \sum_{k=1}^{i} q_{i+k}/2^k$ pour i allant de n à 1 .

Comme $p_i \leq q_i$ pour tout $n+1 \leq i \leq 2n$, on aura $p_1 \leq q_1$; comme $p_i \geq 1 - 2/2^i$ pour tout i , il vient $p_i \geq (1 - 2/2^n)q_i$ pour $n+1 \leq i \leq 2n$ et donc $p_1 \geq (1 - 2/2^n)q_1$. En conclusion :

$$(1 - 2/2^n)q_1 \leq p_1 \leq q_1 .$$

Il reste à évaluer q_1 .

Algorithme de calcul de q_1 :

0. se donner n et réserver les mémoires $q(1), q(2), ..., q(2n)$

1. pour i allant de $n+1$ à $2n$ faire $q(i) \leftarrow 1$

2. pour i allant de n à 1

3. $q = 0$

4. pour j allant de 1 à i faire $q \leftarrow q + q(i+j)/2^j$

[1] Du à un intervenant qui souhaite rester anonyme sur le forum http://www.les-mathematiques.net/.

5. $q(i) \leftarrow q$

6. boucle sur i et test (next i)

7. afficher $q(1)$.

Pour $n = 20$, on lit $q(1) = 0{,}316684328$, d'où $0{,}3166837 \leq p_1 \leq 0{,}3166844$.

Deuxième calcul numérique de p_1 (Robert CHARDARD)

Robert Chardard raisonne sur l'événement contraire : $1 - p_1 = P\left(\bigcup_{n\geq 1} \overline{A_n}\right)$.

De ce fait, il introduit $r_n = P\left(\bigcup_1^n \overline{A_k}\right)$ et on a similairement $r_n \leq 1 - p_1 \leq r_n + 1/2^n$.

Un calcul direct donne $r_1 \leq 1/2$ et $r_2 \leq 5/8$; pour $n \geq 2$, on va prouver que

$$r_{n+1} = r_n + \frac{1 - r_m}{2^{n+2}}, \text{ où } m = \operatorname{int}(n/2).$$

Cette formule, facile à programmer, donne encore plus rapidement les décimales de p_1.

Démonstration. La différence $r_{n+1} - r_n$ est la probabilité que lors des $2n+2$ premiers lancers tous les lancers de n+2 à 2n+2 ont eu pour résultat "face" et parmi tous les tirages de k+1 à 2k on obtient au moins un pile, ceci pour tous les k de 1 à n. Il en résulte que le lancer n+1 a été pile, sinon la deuxième condition est violée pour $k = n$. Puisque le lancer n+1 a été pile, on obtient au moins un pile pour toutes les séquences de k+1 à 2k quand $m + 1 \leq k \leq n$. La seule condition à imposer est donc que A_k se réalise pour k allant de 1 à m. D'où $r_{n+1} - r_n = (1 - r_m) \times \frac{1}{2} \times \left(\frac{1}{2}\right)^{n+1}$.

Calcul probabiliste de p_1

On utilise $P\left(\bigcup_1^n \overline{A_i}\right) \leq 1 - p_1 \leq P\left(\bigcup_1^n \overline{A_i}\right) + 2/2^n$, d'où $P\left(\bigcap_1^n A_i\right) - 2/2^n \leq p_1 \leq P\left(\bigcap_1^n A_i\right)$.

Une fois $P\left(\bigcap_1^n A_i\right)$ ou $P\left(\bigcup_1^n \overline{A_i}\right)$ évaluée, pour $n = 11$, soit 22 lancers, p_1 sera connu à 10^{-3} près par excès, pour $n = 21$, soit 42 lancers, p_1 sera connu à 10^{-6} près par excès.

Nous donnons trois méthodes de calcul, pour $n = 11$ ou $n = 22$.

1) Par dénombrement exhaustif sur 22 lancers ($n = 11$) : calcul de $P\left(\bigcap_1^n A_i\right)$.

Stéphane Gonnord a testé la présence d'un 1 parmi les sous-listes [1..2],[2..4],..., [n/2-1..n-1]. Une fois les mémoires v[i] remplies de 0 et de 1, on teste la présence d'un 1 dans [k+1..2k] en calculant s(k)= v[k+1]+..+v[2k] ; le calcul est accéléré par la formule s(k+1)=s(k) + v[2k+1]+v[2k+2]-v[k+1].
Le remplissage exhaustif des mémoires v[i] par des 0 ou des 1 se fait en décomposant en base 2 les entiers de 0 à 2^{22} -1. Voici son programme en Python :

```
def decomp(n,p):
  res=[0]*p
```

```
    x=n
    i=0
    while (x>0) and i<p:
        res[i]=x % 2
        x//=2
        i+=1
    return res
def test(v):
    s=v[0]
    if s==0:return 0
    for k in range(len(v)//2 -1): #len paire
        s+=v[2*k+1]+v[2*k+2]-v[k]
        if s==0:return 0
    return 1
-------------------------------------------------

s=sum(test(decomp(i,22)) for i in range(2**22))

print(float(s)/2**22)
```

#0.316762447357 en 42 secondes.

Conformément à la théorie, la valeur obtenue est par excès et l'erreur est (largement) $\leq 10^{-3}$.

2) Par simulation sur 42 **lancers (** $n=21$)

Algorithme de 100 simulations sur 42 lancers :

1. $P \leftarrow 0$

2. pour i allant de 1 à 100

3. pour j allant de 1 à 42 faire $X(j) \leftarrow \text{int}(2 * random)$

4. si $X(2)=0$ aller en 10

5. pour k allant de 2 à 21

6. $S=0$ et pour L allant de $K+1$ à $2*K$ faire $S \leftarrow S+X(L)$

7. si $S=0$ aller en 10

8. boucle et test sur k (next k)

9. $P \leftarrow P+1$

10. boucle et test sur i (next i)

11. afficher P .

J'ai obtenu : $40, 29, 33, 33, 32, 29, 31$ et 34, soit $P=32,6$ % sur 800 simulations, au lieu de $31,7$ %.

Plus rapide, comme cela a été signalé :

4. $S=X(2)$ et si $X(2)=0$ aller en 10

5. pour k allant de 1 à 21

6. pour L allant de $K+1$ à $2*K$ faire $S \leftarrow S+X(2L+1)+X(2L)-X(L+1)$

3) Par la formule du crible : $P\left(\bigcup_1^n \overline{A_i}\right) \leq 1-p_1 \leq P\left(\bigcup_1^n \overline{A_i}\right)+2/2^n$

On sait que $P\left(\bigcup_1^n \overline{A_i}\right)$ est donné par la somme alternée :

$$\sum_1^n P\left(\overline{A_i}\right)-\sum_{i<j} P\left(\overline{A_i}\cap\overline{A_j}\right)+\ldots+(-1)^{k-1}\sum_{i_1<\ldots<i_k} P\left(\overline{A_{i_1}}\cap\ldots\cap\overline{A_{i_k}}\right)+\ldots+(-1)^{n-1}P\left(\overline{A_1}\cap\ldots\cap\overline{A_k}\right)$$

Calcul de $S(k)=\sum_{i_1<\ldots<i_k} P\left(\overline{A_{i_1}}\cap\ldots\cap\overline{A_{i_k}}\right)$: par indépendance des lancers,

$P\left(\overline{A_{i_1}}\cap\ldots\cap\overline{A_{i_k}}\right)=(1/2)^C$, où C est le cardinal de $[i_1+1,2i_1]\cup\ldots\cup[i_k+1,2i_k]$.

Il nous faut donc franchir deux étapes :

– obtenir tous les $k-$uplets $(i_1,\ldots,i_k)$ avec $1\leq i_1<\ldots<i_k\leq n$;

– calculer le cardinal de la réunion d'entiers $[i_1+1,2i_1]\cup\ldots\cup[i_k+1,2i_k]$.

a) *obtention des sous-ensembles* $(x_1,x_2,\ldots,x_k)$ *de* $\{1,2,\ldots,n\}^k$, $1\leq k\leq n$, *tels que* $x_1<x_2<\ldots<x_k$.

0. on se donne n et k et on réserve les mémoires $X(0),X(1),\ldots,X(k)$.

1. pour i allant de 1 à k faire $X(i)\leftarrow i$

2. afficher, quand $n\leq 9$, $\sum_{i=1}^k 10^{k-i}X(i)$ pour lire les résultats, sinon afficher les $X(i)$

3. $i\leftarrow k$

4. $X(i)\leftarrow 1+X(i)$

5. si $X(i)\leq n-k+i$ aller en **2**

6. $i\leftarrow i-1$ et $X(i)\leftarrow 1+X(i)$

7. si $i=0$ alors FIN.

8. si $X(i)>n-k+i$ aller en **6**

9. pour j allant de $i+1$ à k faire $X(j)\leftarrow 1+X(j-1)$

10. aller en **2**.

b) calcul du cardinal C de $[i_1+1,2i_1]\cup\ldots\cup[i_k+1,2i_k]$

0. k est donné ainsi que $X(p)=i_p$ pour p allant de 1 à k.

1. $C\leftarrow 0$

2. pour p allant de 1 à k

3. pour t allant de $1+X(p)$ à $2X(p)$

4. $D\leftarrow 0$

5. pour l allant de 1 à k

6. si $t\in[1+X(l),2X(l)]$ alors $D\leftarrow D+1$

7. next l

8. $C\leftarrow C+1/D$

9. next t

10. next p

11. $C\leftarrow \mathrm{int}(C+0{,}1)$ et afficher C.

Explications. Au début, la cardinal C est mis à 0, puis les éléments sont examinés un par un. Si t ne figure qu'une fois dans la réunion, C est incrémenté de 1. Si t figure D fois, alors $C \leftarrow C+1/D$ pour que sur les D passages t ne figure qu'une fois.

c) Calcul de $S(k)$ pour $n=22$, $1 \leq k \leq 11$

On réunit les deux algorithmes élaborés en **a)** et **b)**.

10. se donner k et $S(k) \leftarrow 0$
20. pour i allant de 1 à k faire $X(i) \leftarrow i$
101. $C \leftarrow 0$
102. pour p allant de 1 à k
103. pour t allant de $1+X(p)$ à $2X(p)$
104. $D \leftarrow 0$
105. pour l allant de 1 à k
106. si $t \in [1+X(l), 2X(l)]$ alors $D \leftarrow D+1$
107. next l
108. $C \leftarrow C+1/D$ [les erreurs d'arrondi font que C ne sera pas entier]
109. next t
110. next p
111. $C \leftarrow \text{int}(C+0{,}1)$ [pour corriger les erreurs d'arrondi]
200. $S(k) \leftarrow S(k)+1/2^C$
203. $i \leftarrow k$
204. $X(i) \leftarrow 1+X(i)$
205. si $X(i) \leq 11-k+i$ aller en **101**
206. $i \leftarrow i-1$ et $X(i) \leftarrow 1+X(i)$
207. si $i=0$ afficher $S(k)$ et FIN.
208. si $X(i) > n-k+i$ aller en **206**
209. pour j allant de $i+1$ à k faire $X(j) \leftarrow 1+X(j-1)$
210. aller en **101**

Pour k allant de 1 à 11, on relève :

$S(1)=0{,}99951171875$,on vérifie que $S(1)=\sum_1^{11}(1/2)^k = 1-(1/2)^{11}$
$S(2)=0{,}427612$
$S(3)=0{,}148489$
$S(4)=0{,}0492296$
$S(5)=0{,}0158491$
$S(6)=4{,}85706\times10^{-3}$
$S(7)=1{,}35803\times10^{-3}$
$S(8)=3{,}24726\times10^{-4}$
$S(9)=1{,}21117\times10^{-4}$

$S(10) = 7,62939 \times 10^{-6}$

$S(11) = 4,76837 \times 10^{-7}$.

Enfin, $P\left(\bigcup_1^{11} \overline{A_i}\right) = \sum_1^{11} (-1)^{k+1} S(k) = 0,68329811$, et

$0,3157 \le P\left(\bigcap_1^{11} A_i\right) - 2/2^{11} \le p_1 \le P\left(\bigcap_1^{11} A_i\right) \le 0,316702$.