M1–Master Informatique ML – 2020/2021

ACA - Devoir surveillé

21 mai 2021 - durée 1h30

Document autorisé : feuille de papier A4 manuscrite.

Toutes vos réponses doivent être justifiées avec rigueur.

Exercice 1 : On suppose que nous avons deux variables aléatoires indépendantes X_1 et X_2. Ces variables prennent leurs valeurs dans $\{1, \ldots, k\}$ avec probabilité uniforme.
Question 1.1 : Calculer $E(\max(X_1, X_2))$.

Solution. Pour calculer $E(\max(X_1, X_2))$, il y a deux cas à considérer : celui où X_1 prend une valeur supérieure à celle de X_2 et celui, symmétrique, où X_2 prend une valeur supérieure à celle de X_1.

Nous commençons par évaluer $\Pr(X_1 = i \wedge X_2 < i)$. Comme les deux variables aléatoires sont indépendantes, nous avons :

$$\begin{array}{rcl} \Pr(X_1 = i \wedge X_2 < i) & = & \Pr(X_1 = i) \times \Pr(X_2 < i) \\ & = & \frac{1}{k} \times \frac{i-1}{k} \\ & = & \frac{i-1}{k^2} \end{array}$$

Symmétriquement, nous avons que $\Pr(X_2 = i \wedge X_1 < i) = \frac{i-1}{k^2}$. Nous avons également que $\Pr(X_1 = i \wedge X_2 = i) = \frac{1}{k^2}$. Finalement, $\Pr(\max(X_1, X_2) = i) = \Pr(X_1 = i \wedge X_2 < i) + \Pr(X_1 = i \wedge X_2 = i) + \Pr(X_2 = i \wedge X_1 < i)$ car les trois évènements $X_i = i \wedge X_2 < i$, $X_1 = i \wedge X_2 = i$ et $X_2 = i \wedge X_1 < i$ sont disjoints. Nous pouvons déduire que :

$$\Pr(\max(X_1, X_2) = i) = 2\frac{i-1}{k^2} + \frac{1}{k^2} = \frac{2i-1}{k^2}$$

Nous avons les éléments pour calculer $E(\max(X_1, X_2))$:

$$\begin{array}{rcl} E(\max(X_1, X_2)) & = & \sum_{i=1}^{k} i \Pr(\max(X_1, X_2) = i) \\ & = & \sum_{i=1}^{k} i \frac{2i-1}{k^2} \end{array}$$

□

Question 1.2 : Calculer $E(\min(X_1, X_2))$

Solution. Nous allons procéder de façon similaire à la question précédente :

$$\begin{array}{rcl} \Pr(X_i = i \wedge X_2 > i) & = & \Pr(X_i = i) \times \Pr(X_2 > i) \\ & = & \frac{1}{k} \times \frac{k-i}{k} \\ & = & \frac{k-i}{k^2} \end{array}$$

De la même manière, $\Pr(X_2 = i \wedge X_1 > i) = \frac{k-i}{k^2}$.

Comme pour la question précédente, nous avons :

$$\begin{array}{rcl}\Pr(\min(X_1,X_2)=i) & = & \Pr(X_1=i\wedge X_2>i)+\Pr(X_1=i\wedge X_2=i)+\Pr(X_2=i\wedge X_1>i)\\ & = & 2\frac{k-i}{k^2}+\frac{1}{k^2}\\ & = & \frac{2k-2i+1}{k^2}\end{array}$$

On en déduit que :

$$\begin{array}{rcl}E(\min(X_1,X_2)) & = & \sum_{i=1}^{k} i\Pr(\min(X_1,X_2)=i)\\ & = & \sum_{i=1}^{k} i\frac{2k-2i+1}{k^2}\end{array}$$

□

Question 1.3 : À partir des calculs précédents, montrer que

$$E(\max(X_1,X_2))+E(\min(X_1,X_2))=E(X_1)+E(X_2)$$

Solution.

$$\begin{array}{rcl}E(\max(X_1,X_2))+E(\min(X_1,X_2)) & = & \sum_{i=1}^{k} i\frac{2i-1}{k^2}+\sum_{i=1}^{k} i\frac{2k-2i+1}{k^2}\\ & = & \sum_{i=1}^{k} i\left(\frac{2i-1}{k^2}+\frac{2k-2i+1}{k^2}\right)\\ & = & \sum_{i=1}^{k} i\left(\frac{2i-1+2k-2i+1}{k^2}\right)\\ & = & \sum_{i=1}^{k} i\left(\frac{2k}{k^2}\right)\\ & = & \sum_{i=1}^{k} i\left(\frac{2}{k}\right)\\ & = & \sum_{i=1}^{k} i\frac{1}{k}+\sum_{i=1}^{k} i\frac{1}{k}\\ & = & E(X_1)+E(X_2)\end{array}$$

□

Question 1.4 : Redémontrer l'identité de la question précédente en utilisant uniquement la linéarité de l'espérance.

Solution. Pour tout x et y, nous avons que $\max(x,y)+\min(x,y)=x+y$. Ainsi :

$$\begin{array}{rcl}E(X_1)+E(X_2) & = & E(X_1+X_2)\\ & = & E(\max(X_1,X_2)+\min(X_1,X_2))\\ & = & E(\max(X_1,X_2))+E(\min(X_1,X_2))\end{array}$$

□

Exercice 2 : Vous jouez à la roulette dans un casino. Vous vous contentez de jouer les couleurs : rouge ou noir. Chaque couleur sort avec probabilité 1/2. Vous décidez de jouer de la manière suivante :

— vous commencez par jouer 1 euro,
— tant que vous perdez vous jouez 2 fois votre mise précédente,
— vous vous arrêtez lorsque vous gagnez.

Une victoire vous permet de récupérer deux fois votre mise.

Question 2.1 : Combien de tour en moyenne allez-vous jouer ?

Solution. La variable aléatoire qui représente le nombre de tours joués est une variable géométrique de paramètre 1/2. Son espérance est ainsi :

$$\frac{1}{\frac{1}{2}} = 2 \ .$$

Ainsi nous allons jouer en moyenne 2 tours. □

Question 2.2 : Si vous gagnez au bout de n tours, quels seront vos gains ?

Solution. Si on gagne au bout de n tours, la somme des mises successives est :

$$\sum_{i=0}^{n-1} 2^i = 2^n - 1 \ .$$

Les gains seront le double de la dernière mise soit 2^n euros. Au final, le total de nos gains s'élève à 1 euro. □

Exercice 3 : Nous supposons que nous avons accès à une fonction `rand()` qui renvoie `True` avec une probabilité non nulle `p` et `False` avec un probabilité non nulle $1 - $ `p`. On suppose que les appels successifs à la fonction `rand()` renvoient des résultats indépendants les uns des autres.

Nous allons étudier la fonction de tire aléatoire de Von Neumann définie en `Python` par :

```
def tir_von_neumann():
    x1 = rand()
    x2 = rand()
    return x1 and not x2 or (x1 or not x2) and tir_von_neumann()
```

Question 3.1 : Pour quelles valeurs de `x1` et `x2`, la fonction renvoie-t-elle une valeur sans faire d'appel récursif ?

NB : rappelez-vous que les opérateurs booléens sont séquentiels à gauche.

Solution. Pour qu'il n'y ait pas d'appel récursif, il suffit que l'expression :

```
x1 and not x2
```

prenne la valeur `True` (dans ce cas l'expression s'évalue à `True` car le premier argument du `or` principal est `True`), ou que l'expression :

```
x1 or not x2
```

prenne la valeur `False` (dans ce cas, le premier argument du `or` principal s'évalue à `False` et le premier argument du second `and` s'évalue à `False`. Il n'y a alors pas besoin d'évaluer son second argument pour renvoyer `False`). Cela se produit lorsque `x1` est égal à `True` et `x2` à `False`, ou lorsque `x1` est égal à `False` et `x2` à `True`. En résumé, il n'y a pas d'appel récursif lorsque `x1` et `x2` prennent des valeurs différentes. □

On considère des appels récursifs successifs de la fonction `tir_von_neumann()` et on appelle X_k la variable aléatoire qui pour le k-ème appel vaut 0 si il y a un $k+1$-ème appel et 1 sinon.
Question 3.2 : Quelle est la probabilité que $X_k = 0$.

Solution. Comme nous l'avons vu dans la question précédente, il n'y a pas d'appel récursif lorsque `x1` et `x2` prennent des valeurs différentes, X_k prend pour valeur 0 lorsque `x1` et `x2` prennent la même valeur. Ainsi, nous avons :

$$\begin{aligned}\Pr(X_k = 0) &= \Pr(\texttt{x1} = \texttt{True} \wedge \texttt{x2} = \texttt{True}) + \Pr(\texttt{x1} = \texttt{False} \wedge \texttt{x2} = \texttt{False}) \\ &= \Pr(\texttt{x1} = \texttt{True}) \times \Pr(\texttt{x2} = \texttt{True}) + \Pr(\texttt{x1} = \texttt{False}) \times \Pr(\texttt{x2} = \texttt{False}) \\ &= p^2 + (1-p)^2 \\ &= 2p^2 - 2p + 1\end{aligned}$$

□

On appelle X la variable aléatoire qui représente le nombre d'appels récursifs successifs de la fonction `tir_von_neumann()`.
Question 3.3 : Quelle est la valeur de $E(X)$?

Solution. La variable X est une distribution géométrique de paramètre $1 - 2p^2 + 2p - 1 = 2p(1-p)$. Ainsi $E(X) = \frac{1}{2p(1-p)}$. □

Question 3.4 : Quelle est la probabilité que `tir_von_neumann()` revoie `True` ?
Quelle est la probabilité que `tir_von_neumann()` revoie `False` ?

Solution. La probabilité que `tir_von_neumann()` revoie `True` est celle que les dernières valeurs prises par `x1` et `x2` soient respectivement `True` et `False`. Comme à chaque tir les valeurs prises par `x1` et `x2` sont indépendantes des précédentes et qu'elles sont indépendantes entre elles, nous avons :

$$\begin{aligned}\Pr(\texttt{tir\_von\_neumann}() = \texttt{True}) &= \Pr(\texttt{x1} = \texttt{True} \wedge \texttt{x2} = \texttt{False} \mid \texttt{x1} \neq \texttt{x2}) \\ &= \frac{\Pr(\texttt{x1}=\texttt{True} \wedge \texttt{x2}=\texttt{False} \wedge \texttt{x1} \neq \texttt{x2})}{\Pr(\texttt{x1} \neq \texttt{x2})} \\ &= \frac{\Pr(\texttt{x1}=\texttt{True} \wedge \texttt{x2}=\texttt{False})}{\Pr(\texttt{x1} \neq \texttt{x2})} \\ &= \frac{\Pr(\texttt{x1}=\texttt{True}) \times \Pr(\texttt{x2}=\texttt{False})}{\Pr(\texttt{x1} \neq \texttt{x2})} \\ &= \frac{p(1-p)}{2p(1-p)} \\ &= \frac{1}{2}\end{aligned}$$

De façon symmétrique, nous obtenons que

$$\Pr(\texttt{tir\_von\_neumann}() = \texttt{False}) = \frac{1}{2}$$

□

Exercice 4 : Deux bases de données de très grandes tailles, contenant 2^N bits d'information (e.g. $N = 16$ représente 10 tera-octets) sont censées contenir exactement les mêmes données. Les ordinateurs qui les contiennent sont très éloignés l'un de l'autre et nous souhaiterions vérifier que les contenus sont bien identiques. Le seul moyen est de passer par le réseau. Il est bien entendu hors de question de transférer tout le contenu de l'une des bases sur le réseau pour faire la vérification. L'une des ingénieurs, versée dans les arcanes de la programmation probabiliste propose de considérer le contenu de chaque base comme un nombre écrit en base 2 de 2^N bits respectivement n_1 pour la première base de données et n_2 pour la seconde base de données. Elle décrit ensuite l'algorithme suivant :

- tirons un entier premier p inférieur à N^2 uniformément au hasard,
- calculons $k_1 = n_1 \bmod p$,
- les représentations binaires de p et k_1 ont une taille de l'ordre de $\log_2(N^2)$ (pour $N = 16$, cela représente 6 bits), et peuvent facilement être envoyés sur le deuxième cite par le réseau,
- calculons alors $k_2 = n_2 \bmod p$,
- et comparons k_1 et k_2 pour savoir si n_1 et n_2 sont égaux.

Le but de cet exercice est d'évaluer la qualité de cet algorithme.

Question 4.1 : Si $k_1 \neq k_2$ que peut-on conclure ?

Si $k_1 = k_2$ est-on sûr que $n_1 = n_2$?

Solution. Lorsque $k_1 \neq k_2$, nous sommes sûrs que $n_1 \neq n_2$. En revanche, il est tout à fait possible que $n_1 \neq n_2$ et que cependant $k_1 = k_2$. □

Question 4.2 : Montrer que k_1 est égal à k_2 ssi p divise $|n_1 - n_2|$.

Solution. $k_1 = k_2$ ssi $n_1 \bmod p = n_2 \bmod p$ ssi $n_1 - n_2 \bmod p = 0$ ssi $|n_1 - n_2 \,| \bmod p = 0$ ssi p divise $|n_1 - n_2|$. □

Afin d'évaluer la probabilité que p divise $m = |n_1 - n_2|$, nous allons montrer qu'il y a au plus N nombres premiers qui divisent m. Tout d'abord, il est clair que $m \leq 2^N$, maintenant supposons que m s'écrive $p_1^{i_1} \cdots p_k^{i_k}$ pour des nombres premiers p_1, …, p_k et des entiers strictement positifs i_1, …, i_k.

Question 4.3 : Montrer que $k \leq N$.

NB : pensez que pour tout i, $p_i \geq 2$.

Solution. Nous raisonnons par l'absurde. Si $k > N$, alors nous avons :

$$p_1^{i_1} \cdots p_k^{i_k} > p_1 \cdots p_N \geq 2^N$$

car pour chaque i, $p_i \geq 2$. Or nous savons que $m \leq 2^N$, ce qui nous amène à conclure que $k \leq N$. □

Si on écrit prim(m) pour le nombre de nombres premiers inférieurs à m, le *théorème des nombres premiers* nous indique que :

$$\text{prim}(m) > \frac{m}{\ln(m)}$$

Question 4.4 : Montrer que, lorsque $n_1 \neq n_2$, la probabilité que $k_1 = k_2$ est inférieure à $\frac{\ln(N^2)}{N}$.

Solution. Comme nous choisissons uniformément aléatoirement un nombre premier entre 2 et N^2 et que dans le pire des cas, m est multiple d'au plus N nombres premiers distincts, nous avons :

$$\Pr(k_1 = k_2 \wedge n_1 \neq n_2) = \Pr(p \text{ divise } m) \leq \frac{N}{\text{prim}(N^2)}$$

Or, le théorème des nombres premiers nous indique que :

$$\text{prim}(N^2) > \frac{N^2}{\ln(N^2)}$$

Ainsi, nous avons :

$$\Pr(k_1 = k_2 \wedge n_1 \neq n_2) \leq \frac{N}{\frac{N^2}{\ln(N^2)}} = \frac{\ln(N^2)}{N}$$

□

Pour $N = 16$, notre borne indique seulement que la probabilité est ainsi inférieure à 0.35. Tout le monde est très déçu que la probabilité de se tromper soit si élevée. L'ingénieur qui a proposé l'algorithme prétend que ce n'est pas un problème et que l'on peut très facilement améliorer la confiance de l'algorithme.

Question 4.5 : Avez-vous une idée simple pour améliorer la confiance en l'algorithme dans cette situation ?

Solution. Pour améliorer la probabilité de ne pas se tromper, il suffit de renouveler l'expérience suffisamment de fois.

Ici, comme $N = 16$ est un nombre suffisamment petit, on peut également faire un test avec 17 nombres premiers différents (typiquement les 17 premiers) pour être certain du résultat.

□

Exercice 5 : On prend uniformément au hasard un carte dans un jeu de n cartes, puis on l'y replace

Question 5.1 : Combien de fois faut-il en moyenne réaliser cette expérience pour avoir vu toutes les cartes ?

Solution. Lorsque l'on a vu k cartes distinctes, la probabilité de voir une carte l'on n'a pas vue est $\frac{n-k}{n}$. Ainsi, la variable aléatoire X_k, qui représente le nombre d'expériences réalisées pour voir une nouvelle carte après en avoir vu k différentes est une variable aléatoire géométrique et nous avons $E(X_k) = \frac{n}{n-k}$.

Si nous appelons X la variable aléatoire qui donne le nombre d'expériences réalisées pour avoir vu toutes les cartes, nous avons :

$$X = \sum_{k=0}^{n-1} X_k$$

et ainsi :

$$\begin{aligned} E(X) &= \sum_{k=0}^{n-1} E(X_k) \\ &= \sum_{k=0}^{n-1} \frac{n}{n-k} \\ &= n \sum_{j=1}^{n} \frac{1}{j} \\ &= nH_n \\ &= n\ln(n) + \Theta(n) \ . \end{aligned}$$

où H_n est le nombre harmonique de rang n. □

Question 5.2 : Si on réalise cette expérience $2n$ fois, combien de cartes en moyenne n'ont pas été vues ?

Solution. On appelle X_i la variable aléatoire de Bernoulli qui indique si la i-ème carte n'a pas été tirée au cours des $2n$ expériences. Comme chaque expérience est indépendante des précédentes, nous avons que :

$$\Pr(X_i = 1) = \left(\frac{n-1}{n}\right)^{2n} \ .$$

Remarquons que, du fait que X_i est une variable de Bernoulli, son espérance est :

$$E(X_i) = \left(\frac{n-1}{n}\right)^{2n} \ .$$

Si X est la variable aléatoire qui correspond au nombre de cartes qui n'ont pas été tirées pendant les $2n$ expériences, nous avons :

$$X = \sum_{i=1}^{n} X_i \ ,$$

ainsi, par linéarité de l'espérance :

$$E(X) = E\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} E(X_i) = n\left(\frac{n-1}{n}\right)^{2n} \ .$$

□

Question 5.3 : Si on réalise cette expérience $2n$ fois, combien de cartes en moyenne n'ont été vues qu'une seule fois ?

Solution. On appelle $Y_{i,j}$ la variable aléatoire de Bernoulli qui indique si la i-ème carte a été tirée seulement lors de la j-ème expérience après les $2n$ expériences. Comme chaque expérience est indépendante des précédentes, nous avons que :

$$\Pr(Y_{i,j} = 1) = \frac{1}{n}\left(\frac{n-1}{n}\right)^{2n-1} .$$

Si Y_i est la variable aléatoire de Bernoulli qui vaut 1 ssi la i-ème carte a été tirée une fois au cours des $2n$ expériences, comme les évènements $Y_{i,j_1} = 1$ et Y_{i,j_2} sont disjoints dès lors que $j_1 \neq j_2$, nous avons :

$$\Pr(Y_i = 1) = \Pr\left(\bigcup_{j=1}^{2n} Y_{i,j} = 1\right) = \sum_{j=1}^{2n} \Pr(Y_{i,j} = 1) = 2n\frac{1}{n}\left(\frac{n-1}{n}\right)^{2n-1} = 2\left(\frac{n-1}{n}\right)^{2n-1} .$$

Comme Y_i est une variable de Bernoulli, son espérance est :

$$E(Y_i) = 2\left(\frac{n-1}{n}\right)^{2n-1} .$$

Pour finir, soit Y la variable aléatoire qui correspond au nombre de cartes qui ont été vues exactement une fois lors des $2n$ expériences, alors :

$$Y = \sum_{i=1}^{n} Y_i \ ,$$

ainsi, par linéarité de l'espérance :

$$E(Y) = E\left(\sum_{i=1}^{n} Y_i\right) = \sum_{i=1}^{n} E(Y_i) = 2n\left(\frac{n-1}{n}\right)^{2n-1} .$$

□