Exercices sur les algorithmes probabilités
Nom du rédacteur : Johanne Cohen
`https://www.lri.fr/~jcohen/pages/enseignement.html`

Probabilité : Dénombrement et génération

Génération d'une loi de probabilité

Question 1. Soit $\mathcal{E}(i)$ l'événement suivant : la fonction $random(p)$ retourne la valeur i. Que vaut la probabilité $Pr(\mathcal{E}(i))$?

Comme la fonction $random(p)$ retourne un entier entre 1 et p, l'ensemble des événements Ω est le suivant

$$\Omega = \{\mathcal{E}(i) : 1 \leq i \leq p\}.$$

Soit i et ℓ deux distincts entiers entre 1 et p. Comme la fonction $random(p)$ tire un entier entre 1 et p de façon uniforme, cela signifie que la probabilité que l'évènement $\mathcal{E}(i)$ se réalise est la même que celle que l'événement $\mathcal{E}(\ell)$ se réalise.
Donc $Pr(\mathcal{E}(i)) = Pr(\mathcal{E}(\ell))$ pour tous événements $\mathcal{E}(i)$ et $\mathcal{E}(\ell)$ de Ω.
Calculons maintenant $Pr(\mathcal{E}(i))$.

$$\begin{aligned} Pr(\Omega) &= 1 \\ \sum_{\ell=1}^{p} Pr(\mathcal{E}(\ell)) &= 1 \\ \sum_{\ell=1}^{p} Pr(\mathcal{E}(i)) &= 1 \\ Pr(\mathcal{E}(i)) &= \frac{1}{p} \end{aligned}$$

Question 2. Soit X une variable aléatoire qui peut prendre la valeur entière soit 0 à 1. Soit α et p deux entiers tels que $\alpha \leq p$. Fixons la loi de probabilité suivante : $Pr(X = 0) = \frac{\alpha}{p}$.

— Que vaut la probabilité $Pr(X = 1)$?

Comme la variable aléatoire X peut prendre la valeur soit 0 à 1, l'ensemble des événements Ω est $\{0, 1\}$. Comme $Pr(\Omega) = Pr(X \in \{0, 1\}) = 1$, on a $Pr(\Omega) = Pr(X = 1) + Pr(X = 0)$. Par conséquent,

$$Pr(X = 1) = 1 - \frac{\alpha}{p}$$

— Soit ℓ est un entier tel que $0 < \ell$ et $\ell + \alpha - 1 \leq p$. Calculer la probabilité de l'évènement $\bigcup_{j=\ell}^{\ell+\alpha-1} \mathcal{E}(j)$.

Rappellons que $\mathcal{E}(i)$ est l'événement suivant : la fonction $random(p)$ retourne la valeur i. D'après la question précédente, on a $Pr(\mathcal{E}(i)) = \frac{1}{p}$.
Soit deux événements $\mathcal{E}(i)$ et $\mathcal{E}(\ell)$ de Ω tel que $i \neq \ell$. Comme ils sont disjoints, on a

$$Pr(\mathcal{E}(i) \cup \mathcal{E}(\ell)) = Pr(\mathcal{E}(i)) + Pr(\mathcal{E}(\ell)) = \frac{2}{p}$$

On peut appliquer les mêmes arguments pour α événements disjoints de Ω. En supposant que ℓ est un entier strictement positif tel que $\ell + \alpha - 1 \leq p$, on peut calculer la probabilité de l'évènement $\bigcup_{j=\ell}^{\ell+\alpha-1} \mathcal{E}(j)$.

$$Pr(\bigcup_{j=\ell}^{\ell+\alpha-1} \mathcal{E}(j)) = \sum_{j=\ell}^{\ell+\alpha-1} Pr(\mathcal{E}(j)) = \frac{\alpha}{p}$$

— Concevoir un algorithme qui permet de tirer une valeur aléatoire en fonction de cette loi de probabilité.

D'après la question précédente, on peut en déduire que

$$Pr(\bigcup_{j=1}^{\alpha} \mathcal{E}(j)) = Pr(\text{ la fct } random(p) \text{ retourne un entier entre 1 et } \alpha) = \frac{\alpha}{p}$$

Algorithme Loi_binomiale
Entrée : p, α deux entiers avec $\alpha \leq p$.
Sortie : un entier b dans$\{0, 1\}$.

1. $i \leftarrow random(p)$
2. Si $i \leq \alpha$ alors retourner 0 sinon retourner 1

Question 3. Soit X une variable aléatoire qui peut prendre la valeur entière de 0 à n. Soit p un entier. Soit $n+1$ entiers tel $\alpha_0, \ldots, \alpha_n$ tel que $\sum_{i=0}^{n} \alpha_i = p$. Fixons la loi de probabilité suivante : $Pr(X = i) = \frac{\alpha_i}{p}$.

Concevoir un algorithme qui permet de tirer une valeur aléatoire en fonction de cette loi de probabilité.

D'après la question précédente : $Pr(X = i) = \frac{\alpha_i}{p}$ correspond à la probabilité $Pr(\bigcup_{j=\ell}^{\ell+\alpha_i-1} \mathcal{E}(j))$ en supposant que ℓ est en entier tel que $\ell + \alpha_i - 1 \leq p$. En d'autre termes,

$$Pr(X = i) = Pr(\text{ la fct } random(p) \text{ retourne un entier entre } \ell \text{ et } \ell+\alpha_i-1) = \frac{\alpha_i}{p}$$

Maintenant il suffit de partitionner l'ensemble des p premiers entiers en $n+1$ sous-ensembles : le i-ème sous-ensemble contient α_i entiers distincts. Considérons le sous-ensemble des p premiers entiers $A_i = \{\ell : \sum_{j=1}^{i-1} \alpha_j < \ell \leq \sum_{j=1}^{i} \alpha_j\}$. Observons que $\sum_{j=1}^{i} \alpha_j - \sum_{j=1}^{i-1} \alpha_j = \alpha_i$. Donc

1. $|A_i| = \alpha_i$ pour $i = 0, \cdots, n$
2. $A_i \cap A_\ell = \emptyset$ pour $0 \leq i < \ell \leq n$
3. $\bigcup_{j=0}^{n} A_0 = \{1, \cdots, p\}$

Donc remarquons que $Pr(\bigcup_{j \in A_i} \mathcal{E}(j)) = \frac{\alpha_i}{p}$. Nous allons concevoir un algorithme qui tourne i si l'entier k tiré par la fonction aléatoire $random(p)$ est dans l'ensemble A_i ou k vérifie la condition suivante $\sum_{j=1}^{i-1} \alpha_j < k \leq \sum_{j=1}^{i} \alpha_j$.

Algorithme Génération_des_lois

Entrée : p un entier et $n+1$ entiers $\alpha_0, \ldots, \alpha_n$ tel que $\sum_{i=0}^{n} \alpha_i = p$.

Sortie : un entier b compris entre 1 et n.

1. $k \leftarrow random(p)$
2. $b \leftarrow 0$; $somme \leftarrow 0$
3. tant que $somme + \alpha_b < k$ et $b \leq n$ faire
   (a) $b \leftarrow b + 1$
   (b) $somme \leftarrow somme + \alpha_b$
4. retourner b ;

Parties dans un ensemble S de n éléments

Nous allons considérer que S correspond aux n premiers entiers positifs : $S = \{1, \cdots, n\}$. Nous rappellons qu'une ***partie*** de S est un sous-ensemble de S.

Question 4. Enumérer toutes les parties de l'ensemble $\{1, 2, 3\}$.

$\emptyset$, $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$, $\{1, 2, 3\}$,

Question 5. Notons par $\mathcal{P}(S)$ l'ensemble de toutes les parties de S. Nous allons considérons une fonction $f : P(S) \to \mathbb{N}$ tel que

$$f(C) = \mathbf{x} \text{ avec } \mathbf{x} = \sum_{\ell \in C} 2^{\ell - 1}$$

1. Calculer $f(C)$ quand $C = \emptyset$. $C = \{3\}$, et $C = \{1, 3\}$.

   $f(\emptyset) = 0,\ f(\{3\}) = 4,\ f(\{3\}) = 4,\ f(\{1\}) = 1,\ f(\{1, 3\}) = 5$

2. Soit C une partie de S. Montrer que il existe un entier $\mathbf{x}$ compris entre 0 et $2^n - 1$ tel que $f(C) = \mathbf{x}$. En déduire que $|\mathcal{P}(S)| \leq 2^n$.

Soit C une partie de S. L'entier $f(C) = \mathbf{x}$ peut se représenter sous forme de vecteur $(x_{n-1}, \cdots, x_0)$ avec $x_{i-1} = \begin{cases} 1 & si \quad i \in C \\ 0 & sinon \end{cases}$. Cette représentation correspond à la représentation binaire sur n bits de l'entier $\mathbf{x}$, car

$$\mathbf{x} = \sum_{\ell \in C} 2^{\ell-1} = \sum_{\ell=0}^{n-1} x_\ell 2^\ell$$

Donc le nombre de partie de S est majoré par le nombre d'entiers que l'on peut coder avec n bits.

$$|\mathcal{P}(S)| \leq 2^n$$

.

3. Montrer que pour un entier $\mathbf{x}$ compris entre 0 et $2^n - 1$, il existe une partie C de S telle que $f(C) = x$.

   Maintenant, on va associer une partie C de S à un entier $\mathbf{x}$ que l'on peut coder avec n bits c'est-à-dire $0 \leq \mathbf{x} \leq 2^n - 1$. Notons $(x_{n-1}, \cdots, x_0)$ représentation binaire sur n bits de l'entier $\mathbf{x}$.

   Nous allons construire la partie C associé à $\mathbf{x}$ de la façon suivante

   — $i \in C$ si $x_{i-1} = 1$,

   — $i \notin C$ sinon

   Donc une partie C de S ainsi construite peut être associé à un entier entre 0 et $2^n - 1$. Donc le nombre d'entiers que l'on peut coder avec n bits est majoré par le nombre de partie de S.

$$|\mathcal{P}(S)| \geq 2^n$$

   .

4. En déduire que le nombre de parties de S est 2^n.

   Donc on peut en déduire que $|\mathcal{P}(S)| = 2^n$ car on a prouvé que $|\mathcal{P}(S)| \leq 2^n$ et $|\mathcal{P}(S)| \geq 2^n$

Question 6. Quelle est la probabilité de générer une partie C si elle est construite de façon uniforme.

Soit X la variable aléatoire correspondant à la partie générée de S de façon uniforme.

L'ensemble des événements Ω est le suivant $\Omega = \{X = C : C \in \mathcal{P}(S)\}$.

Comme la variable aléatoire X correspond à la partie générée de S de façon uniforme, on a $Pr(X = C) = Pr(X = C')$ pour toutes parties C et C' de S. Calculons maintenant $Pr(X = C)$.

$$\begin{aligned} Pr(\Omega) &= 1 \\ \sum_{C \in \mathcal{P}(S)} Pr(X = C) &= 1 \\ 2^n Pr(X = C) &= 1 \\ Pr(X = C) &= \frac{1}{2^n} \end{aligned}$$

Question 7. Concevoir un algorithme qui construit une partie de S générée de façon uniforme.

Il suffit de générer un entier x entre 0 et $2^n - 1$.

Algorithme Génération_partie(S)

Entrée : un ensemble S des n premiers

Sortie : une partie C

1. générer un entier x entre 0 et $2^n - 1$ de façon uniforme
2. convertir l'entier en codage binaire : $x = (x_{n-1}, \cdots, x_0)$
3. $C \leftarrow \emptyset$
4. pour i allant de 0 à n
   (a) si $x_i = 1$ alors $C \leftarrow C \cup \{i + 1\}$
5. Retourner C