L'algorithme de Fisher-Yates

Marc Lorenzi

12 mai 2022

1 Introduction

On représente en Python une permutation $\sigma \in \mathfrak{S}_n$ par la liste $[\sigma(0), \ldots, \sigma(n-1)]$, c'est à dire par une liste d'entiers distincts de $[\![0, n-1]\!]$.

Dans toute la suite, nous identifions l'ensemble $\mathfrak{S}_n$ avec l'ensemble des listes Python de longueur n dont les éléments sont les entiers entre 0 et $n-1$.

Si s est une liste de longueur n et $0 \leq a \leq b \leq n$, $s[a : b]$ est la sous-liste des éléments de s d'indices $a, a+1, \ldots, b-1$. En particulier, $s[a : a]$ est la liste vide.

Si a et b sont deux entiers tels que $a \leq b$, l'appel `randint(a,b)` renvoie un élément aléatoire de $[\![a, b]\!]$ avec une probabilité uniforme. Nous y revenons un peu plus loin.

2 L'algorithme

L'algorithme de Fisher-Yates est un algorithme qui renvoie une permutation « aléatoire » de $\mathfrak{S}_n$ en temps $O(n)$.

Voici une implémentation en Python de l'algorithme de Fisher-Yates. La fonction `fisher_yates` prend en paramètre un entier n et renvoie une liste.

```
def fisher_yates(n):
    s = list(range(n))
    for i in range(n):
        j = randint(i, n - 1)
        s[i], s[j] = s[j], s[i]
    return s
```

Une version légèrement modifiée de cet algorithme permet de mélanger une liste en place, c'est à dire avec une complexité en espace en $O(1)$. La voici.

```
def mélanger(xs):
    n = len(xs)
    for i in range(n):
        j = randint(i, n - 1)
        xs[i], xs[j] = xs[j], xs[i]
```

Nous allons dans la suite de l'article analyser la fonction `fisher_yates`.

3 Propriétés faciles

Commençons par deux propriétés évidentes.

Proposition 1. *L'algorithme effectue n échanges d'éléments de listes, et termine.*

Proposition 2. *La liste renvoyée est de longueur n.*

Maintenant, un tout petit peu moins évident.

Proposition 3. *Pour tout $i \in [\![0, n]\!]$, la valeur de s avant la i^e itération est une permutation de $\mathfrak{S}_n$.*

Démonstration. Pour $i = n$, il faut comprendre que l'on parle de la valeur de s à la sortie de la boucle `for`.

Les modifications de la liste s effectuées par l'algorithme sont des transpositions (et donc des permutations) d'éléments de s. La liste s est initialement égale à id, c'est donc au départ un élément de $\mathfrak{S}_n$. Comme toute composée de permutations en est encore une, la proposition en découle par une récurrence sur i. □

Bien entendu, en prenant $i = n$ dans la proposition précédente, on obtient que la fonction renvoie un élément de $\mathfrak{S}_n$.

Ce qui est beaucoup moins évident, c'est que l'algorithme de Fisher-Yates peut renvoyer *n'importe quelle* permutation de $\mathfrak{S}_n$, avec une probabilité uniforme. C'est ce que nous allons maintenant prouver.

4 Analyse probabiliste

Définition 1. Pour tout $i \in [\![0, n]\!]$, une *i-permutation* est une liste de i entiers distincts de $[\![0, n-1]\!]$.

On crée une i-permutation en choisissant son élément 0 (n possibilités), son élément 1 ($n-1$ possibilités),..., son élément $i-1$ ($n-i+1$ possibilités). Le cardinal de l'ensemble des i permutations est donc

$$n(n-1)\dots(n-i+1) = \frac{n!}{(n-i)!}$$

Pour parler de probabilités, donnons-nous un espace probabilisé $(\Omega, \mathcal{T}, \mathbb{P})$. Nous supposerons cet espace assez riche pour qu'il existe n variables aléatoires indépendantes $X_i : \Omega \longrightarrow \mathbb{R}$, $i \in [\![0, n-1]\!]$, de sorte que X_i suit une loi uniforme sur $[\![i, n-1]\!]$. L'appel `randint(i, n - 1)` renvoie $X_i(\omega)$ où ω est un élément de Ω.

Notation. Pour tout $i \in [\![0, n]\!]$, notons s_i la valeur de s avant la i^e itération.

Proposition 4. *Soit $\sigma \in \mathfrak{S}_n$. Alors*

$$\mathbb{P}(s_n = \sigma) = \frac{1}{n!}$$

Appelons $\mathcal{P}(i)$ la propriété suivante :

Pour toute i-permutation σ,

$$\mathbb{P}(s_i[0:i] = \sigma) = \frac{(n-i)!}{n!}$$

Lemme 5. $\forall i \in [\![0, n]\!], \mathcal{P}(i)$.

Démonstration. Faisons une récurrence sur i.

Pour $i = 0$, il y a une unique 0-permutation σ, qui est la liste vide. La sous-liste $s_0[0:0]$ est aussi la liste vide. On a donc

$$\mathbb{P}([s_0[0:0] = \sigma]) = \mathbb{P}(\Omega) = 1 = \frac{(n-0)!}{n!}$$

Soit $i \in [\![0, n-1]\!]$. Supposons $\mathcal{P}(i)$. Soit $\sigma = [x_0, \ldots, x_i]$ une $(i+1)$-permutation.

Soit $\sigma' = \sigma[0:i] = [x_0, \ldots, x_{i-1}]$. Considérons les événements

$$\begin{aligned} E_1 &= [s_i[0:i] = \sigma'] \\ E_2 &= [s_{i+1}[0:i+1] = \sigma] \end{aligned}$$

Par l'hypothèse de récurrence, on a

$$\mathbb{P}(E_1) = \frac{(n-i)!}{n!}$$

Maintenant, remarquons que

$$\mathbb{P}(E_1 \cap E_2) = \mathbb{P}(E_2 \mid E_1)\mathbb{P}(E_1)$$

À la i^{e} itération, l'algorithme crée s_{i+1} en échangeant $s_i[i]$ et $s_i[X_i(\omega)]$. Comme $X_i(\omega) \in [\![i, n-1]\!]$, on a

$$s_i[0:i] = s_{i+1}[0:i]$$

Calculons $\mathbb{P}(E_2 \mid E_1)$. On a

$$\begin{aligned} \mathbb{P}(E_2 \mid E_1) &= \mathbb{P}(s_{i+1}[0:i+1] = \sigma \mid s_i[0:i] = \sigma') \\ &= \mathbb{P}(s_{i+1}[i] = x_i, s_{i+1}[0:i] = \sigma' \mid s_i[0:i] = \sigma') \\ &= \mathbb{P}(s_{i+1}[i] = x_i, s_i[0:i] = \sigma' \mid s_i[0:i] = \sigma') \\ &= \mathbb{P}(s_{i+1}[i] = x_i \mid s_i[0:i] = \sigma') \\ &= \frac{1}{n-i} \end{aligned}$$

Précisons un peu la dernière égalité. Sous l'hypothèse que l'événement $[s_i[0 : i] = \sigma']$ est réalisé, x_i n'est aucun des éléments $s_i[0], \ldots, s_i[i-1]$. Il existe donc $k \in [\![i, n-1]\!]$ tel que $s_i[k] = x_i$. De plus,

$$s_{i+1}[i] = s_i[X_i(\omega)]$$

On a donc

$$s_{i+1}[i] = x_i \iff X_i(\omega) = k$$

Ainsi, puisque la variable aléatoire X_i suit une loi uniforme sur $[\![i, n-1]\!]$,

$$\mathbb{P}(s_{i+1}[i] = x_i \mid s_i[0 : i] = \sigma') = \mathbb{P}([X_i = k]) = \frac{1}{n-i}$$

De là,

$$\begin{aligned} \mathbb{P}(E_1 \cap E_2) &= \mathbb{P}(E_2 \mid E_1)\mathbb{P}(E_1) \\ &= \frac{(n-i)!}{n!}\frac{1}{n-i} \\ &= \frac{(n-(i+1))!}{n!} \end{aligned}$$

□

Nous pouvons maintenant prouver la proposition.

Démonstration. Appliquons le lemme à $i = n$. Soit σ une n-permutation, c'est à dire un élément de $\mathfrak{S}_n$. On a

$$\mathbb{P}(s_n[0 : n] = \sigma) = \frac{(n-n)!}{n!} = \frac{1}{n!}$$

□