

Méthode de Monte-Carlo

La méthode de Monte-Carlo consiste à estimer une quantité difficile à calculer directement, et ce en l'interprétant comme l'espérance d'une variable aléatoire ou comme la probabilité d'un événement lié à une variable aléatoire, cette dernière pouvant être simulée en Python.

1 Calcul d'une espérance

Considérons une variable aléatoire X et une fonction $g : \mathbb{R} \rightarrow \mathbb{R}$. On suppose que $g(X)$ est une variable aléatoire admettant une espérance m et une variance σ^2 . Considérons une suite $(X_n)_{n \in \mathbb{N}^*}$ de variables aléatoires indépendantes et de même loi que X . Comme les variables aléatoires $g(X_1), \dots, g(X_n), \dots$ sont indépendantes d'après le lemme des coalitions, qu'elles suivent toutes la même loi et qu'elles admettent une espérance et une variance, la loi faible des grands nombres entraîne que la suite $(\overline{g(X_n)})_{n \in \mathbb{N}^*}$ de leurs moyennes empiriques, définie pour tout $n \in \mathbb{N}^*$ par :

$$\overline{g(X_n)} = \frac{g(X_1) + \dots + g(X_n)}{n},$$

converge en probabilité vers $m = E(g(X))$. La méthode de Monte-Carlo consiste alors à approcher la valeur de $E(g(X))$, difficile à calculer, par une réalisation de la variable aléatoire $\overline{g(X_n)}$ pour n assez grand. Elle permet notamment de donner des valeurs approchées de sommes de séries et d'intégrales.

Exemple 1.1 *Supposons que l'on veuille calculer la somme $S = \sum_{k=0}^{+\infty} \frac{1}{k!(1+k^2)}$ à l'aide de la méthode de Monte-Carlo. Pour ce faire, on commence par écrire que $S = E(\frac{e}{1+X^2})$, où X suit la loi de Poisson de paramètre 1. Ensuite, on simule la variable aléatoire $g(X) = \frac{e}{1+X^2}$ un grand nombre n de fois, puis on calcule la moyenne des résultats obtenus, ce qui correspond à la moyenne empirique $\overline{g(X_n)}$. Pour n assez grand, le résultat obtenu sera une bonne approximation de la somme S . Dès lors, pour calculer S , on pourra utiliser la liste de commandes suivante :*

```
import numpy as np
import numpy.random as rd

a=rd.poisson(1,10000)
b=np.exp(1)/(1+(a**2))
s=np.mean(b)
return s
```

Exemple 1.2 *Supposons que l'on veuille calculer l'intégrale $I = \int_0^1 \frac{1}{1+t^4} dt$ à l'aide de la méthode de Monte-Carlo. Pour ce faire, on commence par écrire que $I = E(\frac{1}{1+X^4})$, où X suit la loi uniforme sur $[0, 1]$. Ensuite, on simule la variable aléatoire $g(X) = \frac{1}{1+X^4}$ un grand nombre n de fois, puis on calcule la moyenne des résultats obtenus, ce qui correspond à la moyenne empirique $\overline{g(X_n)}$. Pour n assez grand, le résultat obtenu sera une bonne approximation de l'intégrale I . Dès lors, pour calculer I , on pourra utiliser la liste de commandes suivante :*

```
import numpy as np
import numpy.random as rd

a=rd.random(10000)
b=1/(1+(a**4))
s=np.mean(b)
return s
```

2 Calcul d'une probabilité

Considérons un événement A d'un espace probabilisé (Ω, \mathcal{A}, P) . Pour calculer la valeur de $P(A)$, on peut simuler un grand nombre de fois l'expérience au cours de laquelle A se produit, et on considère la suite de variables aléatoires $(X_n)_{n \in \mathbb{N}^*}$, telle que, pour tout $n \in \mathbb{N}^*$, $X_n = 1$ si A se produit et $X_n = 0$ sinon. Par construction, $(X_n)_{n \in \mathbb{N}^*}$ est une suite de variables de Bernoulli indépendantes de même paramètre $p = P(A)$. D'après la loi faible des grands nombres (ou le théorème d'or de Bernoulli), la suite des moyennes empiriques $(\overline{X_n})_{n \in \mathbb{N}^*}$, définie pour tout $n \in \mathbb{N}^*$ par :

$$\overline{X_n} = \frac{X_1 + \dots + X_n}{n},$$

converge en probabilité vers p . La méthode de Monte-Carlo consiste alors à approcher la valeur de $p = P(A)$, difficile à calculer, par une réalisation de la variable aléatoire \overline{X}_n pour n assez grand. A noter qu'en général, l'événement A est construit à l'aide d'une ou de plusieurs variables aléatoires que l'on sait simuler.

Exemple 2.1 *Considérons deux variables aléatoires U et V indépendantes de même loi uniforme sur $[0, 1]$. Supposons que l'on veuille calculer la probabilité $p = P([U + V \leq 1])$ à l'aide de la méthode de Monte-Carlo. Pour ce faire, on simule d'abord la variable aléatoire $Z = U + V$ un grand nombre n de fois, puis on compte le nombre de fois que le résultat obtenu est ≤ 1 . Si l'on divise ce nombre par n , on obtient une estimation de la moyenne empirique \overline{X}_n citée plus haut, ce qui nous donne une bonne approximation de p pour n assez grand. Dès lors, pour calculer p , on utilisera la liste de commandes suivante :*

```
import numpy as np
import numpy.random as rd

a=rd.random(10000)
b=rd.random(10000)
c=a+b
s=np.mean(c<=1)
return s
```

3 Intervalles de confiance et méthode de Monte-Carlo

On notera que la méthode de Monte-Carlo repose sur des estimations des moyennes empiriques $\overline{g(X_n)}$, lesquelles sont obtenues à partir de simulations. Or, on sait d'après la loi faible des grands nombres que cette suite de moyennes empiriques converge en probabilité (mais pas tout le temps) vers l'espérance m recherchée. En d'autres termes, la méthode de Monte-Carlo ne nous fournit pas une estimation de m fiable à 100%. En général, on cherchera à limiter la probabilité que le résultat obtenu ne soit pas satisfaisant, ce qui revient à déterminer un intervalle de confiance. Plus précisément, étant donné un réel $\alpha \in]0, 1[$, un intervalle de confiance de m au niveau de confiance $1 - \alpha$ est un intervalle I dont on sait que la valeur de m s'y trouve avec une probabilité $\geq 1 - \alpha$. Pour déterminer un tel intervalle, on dispose de deux méthodes :

1. Une première méthode consiste à utiliser l'inégalité de Bienaymé-Tchebychev. Plus précisément, supposons que toutes les variables aléatoires $g(X_n)$ soient indépendantes et admettent une espérance commune m et une variance commune σ^2 . Alors on peut écrire que, pour tout $\varepsilon > 0$:

$$P(|\overline{g(X_n)} - m| \geq \varepsilon) \leq \frac{\sigma^2}{n\varepsilon^2}.$$

En d'autres termes, une estimation de $\overline{g(X_n)}$ nous donne une valeur approchée de m à la précision ε et au niveau de confiance $1 - \frac{\sigma^2}{n\varepsilon^2}$. A titre d'exemple, si $(X_n)_{n \geq 1}$ est une suite de variables de Bernoulli indépendantes et de même paramètre p inconnu, on sait que $\sigma^2 = p(1-p) \leq \frac{1}{4}$, et donc :

$$P(|\overline{X}_n - p| \geq \varepsilon) \leq \frac{1}{4n\varepsilon^2}.$$

Si l'on veut avoir un niveau de confiance $\geq 1 - \alpha$, on doit choisir n et ε de telle sorte que $\frac{1}{4n\varepsilon^2} \leq \alpha$, ce qui revient à écrire que $\varepsilon \geq \frac{1}{2\sqrt{n\alpha}}$. Dès lors, un intervalle de confiance de p au niveau de confiance $1 - \alpha$ est donné par :

$$I = \left[\overline{X}_n - \frac{1}{2\sqrt{n\alpha}}, \overline{X}_n + \frac{1}{2\sqrt{n\alpha}} \right].$$

En particulier, on voit que, si n est choisi assez grand, on obtient une estimation de m avec une grande précision et à un niveau de confiance proche de 100%. A noter qu'avec cette méthode, la demi-longueur L de l'intervalle de confiance est donnée par :

$$L = \frac{1}{2\sqrt{n\alpha}}.$$

2. Une deuxième méthode consiste à utiliser le théorème limite central. Plus précisément, supposons que toutes les variables aléatoires $g(X_n)$ soient indépendantes et admettent une espérance commune m et une variance commune σ^2 . D'après le théorème limite central, on sait que la suite $(\overline{g(X_n)})^*$ converge en loi vers une variable normale centrée réduite. Si n est choisi assez grand (en général à

partir de $n \geq 30$), on considère que la loi de $\overline{g(X_n)}^*$ est bien approchée par la loi normale centrée réduite. Si l'on désigne par Φ la fonction de répartition de la loi normale centrée réduite, on obtient avec cette approximation que, pour tout $\gamma > 0$:

$$P\left(\left[\left|\overline{g(X_n)} - m\right| \geq \gamma \frac{\sigma}{\sqrt{n}}\right]\right) = P\left(\left[\left|\overline{g(X_n)}^* \right| \geq \gamma\right]\right) = \Phi(\gamma) - \Phi(-\gamma) = 2\Phi(\gamma) - 1.$$

A noter que, comme la fonction Φ est continue et strictement croissante, qu'elle tend vers 0 en $-\infty$ et vers 1 en $+\infty$, elle définit une bijection de \mathbb{R} dans $]0, 1[$ d'après le théorème de la bijection. De plus, sa bijection réciproque Φ^{-1} est aussi continue et strictement croissante. En particulier, on trouve avec la relation ci-dessus que :

$$P\left(\left[\left|\overline{g(X_n)} - m\right| \geq \gamma \frac{\sigma}{\sqrt{n}}\right]\right) \leq \alpha \quad \text{si} \quad \gamma \geq \Phi^{-1}\left(1 - \frac{\alpha}{2}\right).$$

Si l'écart-type σ est connu ou s'il est majoré par un réel $K > 0$ et si l'on pose $t_\alpha = \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)$, un intervalle de confiance (asymptotique) de m au niveau de confiance $1 - \alpha$ est donné par :

$$I = \left[\overline{g(X_n)} - \frac{t_\alpha K}{\sqrt{n}}, \overline{g(X_n)} + \frac{t_\alpha K}{\sqrt{n}} \right].$$

En particulier, on voit que, si n est choisi assez grand, on obtient une estimation de m avec une grande précision et à un niveau de confiance proche de 100%. A noter qu'avec cette méthode, la demi-longueur L de l'intervalle de confiance est donnée par :

$$L = \frac{t_\alpha K}{\sqrt{n}}.$$

En général, on préférera la méthode issue du théorème limite central que celle issue de l'inégalité de Bienaymé-Tchebychev, puisque la première méthode permet de fournir un meilleur intervalle de confiance (c'est-à-dire, de longueur plus petite) que la deuxième. A titre d'exemple, supposons que (X_n) soit une suite de variables de Bernoulli indépendantes de même paramètre p inconnu, que l'on cherche à estimer. Avec la méthode issue de l'inégalité de Bienaymé-Tchebychev et pour un niveau de confiance de 95%, on obtient une demi-longueur d'intervalle égale à :

$$L \simeq \frac{2,24}{\sqrt{n}}.$$

Avec la méthode issue du théorème limite central et pour un niveau de confiance de 95%, on obtient une demi-longueur d'intervalle égale à :

$$L \simeq \frac{0,98}{\sqrt{n}},$$

vu que $\sigma^2 = p(1-p) \leq \frac{1}{4}$ et que $t_\alpha \simeq 1,96$ si $\alpha = 0,05$.

A noter que la fonction de répartition Φ de la loi normale centrée réduite et sa bijection réciproque Φ^{-1} sont implémentées en Python. Pour obtenir la première (resp. la deuxième), il suffit d'utiliser la commande `sp.ndtr()` (resp. `sp.ndtri()`) après avoir importé au préalable la librairie `scipy.special`, via la commande `import scipy.special as sp`.