Méthodes de Monte-Carlo et algorithmes stochastiques – TD/TP 4

Réduction de variance II : stratification, importance sampling et événements rares

M2 Finance – Université Gustave Eiffel

Objectifs

Cette séance met en pratique les deux dernières grandes idées du chapitre 2 : *stratifier* pour imposer une exploration équilibrée de l'espace, puis *changer la loi de simulation* lorsque les régions importantes sont trop rarement visitées.

Exercice 1 – Stratification : allocation proportionnelle ou allocation de Neyman ?

Soit $U \sim \mathcal{U}([0,1])$. On veut estimer

$$I = \mathbb{E}(U^2) = \int_0^1 x^2 \, \mathrm{d}x = \frac{1}{3}.$$

On découpe $[0,1]$ en deux strates

$$A_1 = \left[0, \frac{1}{2}\right], \qquad A_2 = \left(\frac{1}{2}, 1\right].$$

On note $p_k = \mathbb{P}(U \in A_k)$,

$$I_k = \mathbb{E}(U^2 \mid U \in A_k), \qquad \sigma_k^2 = \mathrm{Var}(U^2 \mid U \in A_k).$$

1) Calculer la variance de l'estimateur Monte-Carlo naïf basé sur n tirages de U.
2) Calculer p_1, p_2, puis I_1 et I_2. Vérifier que
$$I = p_1 I_1 + p_2 I_2.$$
3) Calculer exactement σ_1^2 et σ_2^2. Quelle strate est la plus variable ?
4) On suppose n pair et on utilise l'allocation proportionnelle $n_1 = n_2 = n/2$. Calculer la variance de l'estimateur stratifié et la comparer à celle de l'estimateur naïf.
5) Déterminer l'allocation optimale de Neyman. Donner le rapport n_2/n_1, puis les valeurs approximatives de n_1, n_2 lorsque $n = 120$.
6) Calculer la variance minimale obtenue avec l'allocation de Neyman. Comparer les trois méthodes : naïve, stratification proportionnelle et stratification optimale.
7) En pratique, σ_1 et σ_2 seraient inconnus. Décrire une procédure en deux étapes permettant de choisir les n_k sans connaître ces variances à l'avance.

Idée à retenir

La stratification ne consiste pas seulement à « couper l'espace ». Une fois les strates choisies, il reste une seconde question numérique : *où dépenser les simulations ?* L'allocation de Neyman met davantage de tirages dans les strates qui contribuent le plus à la variance.

Exercice 2 – Importance sampling : choisir le paramètre d'une loi exponentielle

Soit $X \sim \mathcal{E}(1)$. Pour $\ell > 0$, on veut estimer la probabilité de queue

$$p_\ell = \mathbb{P}(X > \ell) = \mathrm{e}^{-\ell}.$$

Au lieu de simuler sous la loi naturelle, on propose de tirer

$$Y \sim \mathcal{E}(\lambda), \qquad \lambda > 0.$$

On note $p(x) = \mathrm{e}^{-x}\mathbf{1}_{\{x\geq 0\}}$ la densité naturelle et

$$q_\lambda(x) = \lambda \mathrm{e}^{-\lambda x}\mathbf{1}_{\{x\geq 0\}}$$

la densité de simulation.

1) Construire une variable de réplication H_λ sans biais pour p_ℓ.
2) Montrer que, pour $0 < \lambda < 2$,
$$\mathbb{E}_\lambda(H_\lambda^2) = \frac{\mathrm{e}^{-(2-\lambda)\ell}}{\lambda(2-\lambda)}.$$
Que se passe-t-il lorsque $\lambda \geq 2$?
3) En déduire la variance relative d'une réplication,
$$R_\ell(\lambda) = \frac{\mathrm{Var}_\lambda(H_\lambda)}{p_\ell^2}.$$
Vérifier que $\lambda = 1$ redonne la méthode naïve.
4) Minimiser $R_\ell(\lambda)$ sur $(0, 2)$. Montrer que le minimiseur est
$$\lambda_\ell^\star = \frac{\ell + 1 - \sqrt{\ell^2+1}}{\ell}.$$
Pourquoi est-il naturel que $\lambda_\ell^\star < 1$?
5) Pour $\ell = 5$, comparer la variance relative de la méthode naïve et celle obtenue avec $\lambda_5^\star$.
6) Montrer que, lorsque $\ell \to +\infty$,
$$\lambda_\ell^\star \sim \frac{1}{\ell},$$
et que la variance relative optimale ne croît plus exponentiellement mais seulement linéairement en ℓ. La méthode possède-t-elle pour autant une erreur relative bornée ?

Exercice 3 – Un événement rare à deux régions : pourquoi une seule translation peut échouer

Soit $Z \sim \mathcal{N}(0, 1)$ et, pour $a > 0$,

$$p_a = \mathbb{P}(|Z| > a) = 2\,\overline{\Phi}(a).$$

L'événement rare possède maintenant deux régions importantes : une queue à droite et une queue à gauche.
On considère la loi de simulation symétrique

$$q_a(y) = \frac{1}{2}\varphi(y-a) + \frac{1}{2}\varphi(y+a),$$

qui correspond à : choisir un signe $S \in \{-1, +1\}$ avec probabilité $1/2$, puis simuler $Y \sim \mathcal{N}(Sa, 1)$.

1) Montrer que
$$q_a(y) = \varphi(y)\mathrm{e}^{-a^2/2}\cosh(ay).$$
En déduire le poids de vraisemblance
$$w_a(y) = \frac{\varphi(y)}{q_a(y)}.$$

2) Construire une variable de réplication sans biais pour p_a sous la loi q_a.

3) Sur l'événement $\{|y| > a\}$, montrer que
$$w_a(y) \leq 2\mathrm{e}^{-a^2/2}.$$
En déduire une borne sur le second moment de la variable pondérée.

4) On aurait pu utiliser seulement $Y \sim \mathcal{N}(a, 1)$. Montrer que, pour cette proposition unilatérale, le second moment de l'estimateur de p_a contient le terme
$$\frac{1}{2}\mathrm{e}^{a^2},$$
qui provient de la queue gauche. Pourquoi est-ce un très mauvais signe lorsque a est grand ?

5) Expliquer, sans nouveau calcul, pourquoi le mélange symétrique est beaucoup mieux adapté au problème que la translation unique $\mathcal{N}(a, 1)$.

Idée à retenir

Une bonne loi d'importance doit couvrir *toutes* les régions qui contribuent de façon importante à la quantité cherchée. Déplacer les tirages vers une seule région peut rendre une autre région presque invisible et produire des poids énormes lorsqu'elle est enfin visitée.

Exercice 4 – TP Python : queue gaussienne bilatérale et faux sentiment de convergence

On reprend
$$p_a = \mathbb{P}(|Z| > a), \qquad Z \sim \mathcal{N}(0, 1),$$
avec $a = 4$. On comparera :

— l'estimateur naïf ;

— une translation unilatérale $\mathcal{N}(a, 1)$;

— le mélange symétrique $\frac{1}{2}\mathcal{N}(a, 1) + \frac{1}{2}\mathcal{N}(-a, 1)$.

À programmer vous-même

Les fonctions doivent renvoyer les réplications individuelles. On comparera non seulement les moyennes, mais aussi les variances empiriques, les intervalles de confiance et la stabilité lorsque l'expérience est répétée avec plusieurs graines aléatoires.

```python
import numpy as np
from scipy.stats import norm


def summary(samples, exact):
    n = len(samples)
    mean = np.mean(samples)
    var = np.var(samples, ddof=1)
    half_width = 1.96 * np.sqrt(var / n)
    rel_var = var / exact**2
    return mean, var, rel_var, mean-half_width, mean+half_width


def naive_samples(rng, n, a):
    # A compléter.
    pass


def one_sided_samples(rng, n, a):
    # Proposition : Y ~ N(a,1).
    # A compléter : poids phi(Y)/phi_a(Y).
    pass


def mixture_samples(rng, n, a):
    # 1. tirer S dans {-1,+1} avec probabilités 1/2,1/2 ;
    # 2. tirer Y ~ N(S*a,1) ;
    # 3. utiliser w(Y)=exp(a^2/2)/cosh(aY).
    pass


a = 4.0
n = 200_000
exact = 2 * norm.sf(a)
print("valeur exacte", exact)

for seed in [1, 2, 3, 4, 5]:
    rng = np.random.default_rng(seed)
    print("seed", seed)
    for name, fun in [
        ("naif", naive_samples),
        ("translation +a", one_sided_samples),
        ("melange symetrique", mixture_samples),
    ]:
        x = fun(rng, n, a)
        print(name, summary(x, exact))
    print()
```

1) Compléter les trois fonctions.
2) Comparer les trois méthodes pour les cinq graines. Quelle méthode donne des résultats stables autour de la valeur exacte ?
3) La translation unilatérale peut afficher un intervalle de confiance très court alors que son estimation est loin de p_a. Expliquer ce phénomène à partir de l'exercice 3.
4) Refaire l'expérience pour $a = 2$ puis $a = 5$. Comment la difficulté évolue-t-elle pour chaque méthode ?
5) Pour le mélange symétrique, calculer aussi les poids
$$w_i = \frac{\varphi(Y_i)}{q_a(Y_i)}$$
et l'ESS associée. Une ESS élevée est-elle nécessaire pour que l'estimateur de p_a soit efficace ?

Justifier à partir de ce que vous observez.

Décodage du code Python

Le point important du TP est le suivant : un estimateur peut être exactement sans biais en théorie et pourtant donner, pour une taille de simulation réaliste, une impression trompeuse de convergence si les contributions qui corrigeraient la moyenne sont extraordinairement rares. Il faut donc regarder la structure des poids et répéter les expériences, pas seulement lire un IC calculé sur une seule simulation.