ECOLE SUPÉRIEURE DE LA STATISTIQUE
ET DE L'ANALYSE DE L'INFORMATION

Méthodes de simulation
2023-2024(P4)
2ème année

Chapitre 4: Échantillonnage préférentiel (Importance Sampling)

Tasnime Hamdeni

April 18, 2024

Sommaire

1. Introduction
2. Estimateur d'échantillonnage préférentiel
3. Efficacité relative d'un estimateur
4. Choix de la densité instrumentale
5. Application de la méthode

1- Introduction

1- Introduction

Nous avons vu au chapitre 1 que si nous calculons une quantité $E(X)$ par la méthode de Monte-Carlo, c'est à dire si nous approchons $E(X)$ par une moyenne empirique $\frac{X_1+X_2+...+X_n}{n}$, $X_1, X_2, \dots$ i.i.d. de même loi que X, alors l'erreur

$$E(X) - \frac{X_1 + X_2 + \ldots + X_n}{n}$$

est d'ordre $\frac{\sigma}{\sqrt{n}}$, où $\sigma^2 = V(X)$.
y Nous allons prés7enter ici des méthodes qui permettent d'écrire $E(X) = E(Y)$ avec $V(Y) \leq V(X)$. Ces méthodes sont dites **de réduction de variance**.

2- Estimateur d'échantillonnage préférentiel

2- Estimateur d'échantillonnage préférentiel

Définition

Nous cherchons à calculer $\mathbb{E}[h(X)]$ avec X variable à valeurs dans $\mathbb{R}^d$, de densité f. Pour toute densité $g > 0$, nous pouvons écrire pour $Y \sim g$

$$\delta = \mathbb{E}_f[h(X)] \approx \mathbb{E}_g\left[\frac{h(Y)f(Y)}{g(Y)}\right].$$

Étant donnée $(Y_n)_{n\geq 1}$ une suite de variables aléatoires i.i.d. suivant la densité g, on définit l'estimateur d'échantillonnage préférentiel par:

$$\hat{\delta}_n(g) = \frac{1}{n}\sum_{k=1}^{n}\frac{f(Y_k)}{g(Y_k)}h(Y_k).$$

La densité g est appelée loi instrumentale (ou loi d'importance).

Définition (suite)

Le rapport

$$w_k = \frac{f(Y_k)}{g(Y_k)}$$

est appelé poids d'importance.

L'estimateur d'échantillonnage préférentiel est donc donnée par:

$$\hat{\delta}_n(g) = \frac{1}{n} \sum_{k=1}^{n} \frac{f(Y_k)}{g(Y_k)} h(Y_k) = \frac{1}{n} \sum_{k=1}^{n} w_k h(Y_k).$$

2- Estimateur d'échantillonnage préférentiel

Remarque:

Le coût de calcul de $\hat{\delta}_n(g)$ peut être différent de celui de l'estimateur classique : le coût de simulation suivant g pouvant être différent du coût de simulation suivant f.

3- Efficacité relative d'un estimateur

3- Efficacité relative d'un estimateur

3- Efficacité relative d'un estimateur

Limiter la comparaison des méthodes à la simple comparaison des variances σ^2 et σ_1^2 n'est néanmoins pas nécessairement judicieux si la méthode alternative n'est pas de complexité comparable (par exemple, coût de calcul et utilisation de la mémoire plus importants) à la méthode classique.

3- Efficacité relative d'un estimateur

Limiter la comparaison des méthodes à la simple comparaison des variances σ^2 et σ_1^2 n'est néanmoins pas nécessairement judicieux si la méthode alternative n'est pas de complexité comparable (par exemple, coût de calcul et utilisation de la mémoire plus importants) à la méthode classique. Quel est le critère de comparaison ?

3- Efficacité relative d'un estimateur

Définition

Supposons que le coût de la méthode classique pour simuler un n-échantillon $X_1, \ldots, X_n$ et évaluer n fois h est

$$C \times n.$$

Si le coût de calcul de $\hat{\delta}_n$ est

$$C_1 \times n,$$

alors pour obtenir la même précision ϵ^2, l'efficacité relative (relative efficiency) de $\hat{\delta}_n$ par rapport $\overline{h_n}$:

$$R(\overline{h_n}, \hat{\delta}_n) = \frac{C \times \sigma^2}{C_1 \times \sigma_1^2}.$$

3- Efficacité relative d'un estimateur

Remarque 1

Pour obtenir une erreur quadratique moyenne donnée, une méthode de réduction de variance est d'autant plus efficace que $R(\overline{h_n}, \hat{\delta}_n)$ est grand devant 1.

3- Efficacité relative d'un estimateur

Remarque 1

Pour obtenir une erreur quadratique moyenne donnée, une méthode de réduction de variance est d'autant plus efficace que $R(\overline{h_n}, \hat{\delta}_n)$ est grand devant 1.

Remarque 2

Il n'est pas toujours facile de quantifier l'effet du rapport des coûts C/C_1, celui-ci pouvant dépendre du langage, de la façon de programmer les méthodes, des algorithmes de simulation utilisés, ... Dans le cours, on se concentrera sur la comparaison des variances.

4- Choix de la densité instrumentale

4- Choix de la densité instrumentale

La méthode d'échantillonnage préférentiel repose sur le fait qu'il n'y a pas unicité de la loi par rapport à laquelle on intègre pour calculer δ. Cela laisse beaucoup de liberté quant au choix de g parmi les lois faciles à simuler.

4- Choix de la densité instrumentale

La méthode d'échantillonnage préférentiel repose sur le fait qu'il n'y a pas unicité de la loi par rapport à laquelle on intègre pour calculer δ. Cela laisse beaucoup de liberté quant au choix de g parmi les lois faciles à simuler.

Proposition

L'estimateur d'échantillonnage préférentiel de variance minimale, $\hat{\delta}_n(g^*)$, est obtenu pour la densité instrumentale

$$g^*(y) = \frac{|h(y)| \cdot f(y)}{\int_{\text{support}(f)} |h(x)| \cdot f(x)\, dx}, \quad y \in \mathbb{R}^d.$$

La méthode d'échantillonnage préférentiel repose sur le fait qu'il n'y a pas unicité de la loi par rapport à laquelle on intègre pour calculer δ. Cela laisse beaucoup de liberté quant au choix de g parmi les lois faciles à simuler.

Proposition

L'estimateur d'échantillonnage préférentiel de variance minimale, $\hat{\delta}_n(g^*)$, est obtenu pour la densité instrumentale

$$g^*(y) = \frac{|h(y)| \cdot f(y)}{\int_{\text{support}(f)} |h(x)| \cdot f(x)\, dx}, \quad y \in \mathbb{R}^d.$$

Remarque

Ce résultat n'a qu'un intérêt limité en pratique.

4- Choix de la densité instrumentale

Lorsque $h > 0$, la densité instrumentale optimale est

$$g^*(y) = \frac{h \cdot f}{\mathbb{E}_f[h(X)]},$$

1. Vérifier la validité de $g*$.
2. Vérifier que $\mathrm{Var}_{g^*}[\hat{\delta}_n(g^*)] = 0$

4- Choix de la densité instrumentale

Nous avons donc ici une méthode de Monte-Carlo de variance nulle, ce qui semble n'avoir aucun sens.

4- Choix de la densité instrumentale

Nous avons donc ici une méthode de Monte-Carlo de variance nulle, ce qui semble n'avoir aucun sens.
En fait, la variance de cette méthode n'a pas d'intérêt car cette méthode **n'est pas implémentable**.

4- Choix de la densité instrumentale

Nous avons donc ici une méthode de Monte-Carlo de variance nulle, ce qui semble n'avoir aucun sens.
En fait, la variance de cette méthode n'a pas d'intérêt car cette méthode **n'est pas implémentable**.
En effet, il faudrait pour cela savoir simuler suivant la densité g, mais l'expression de g contient la constante $E_f(h(X))$, que nous ne connaissons pas.

4- Choix de la densité instrumentale

Nous avons donc ici une méthode de Monte-Carlo de variance nulle, ce qui semble n'avoir aucun sens.
En fait, la variance de cette méthode n'a pas d'intérêt car cette méthode **n'est pas implémentable**.
En effet, il faudrait pour cela savoir simuler suivant la densité g, mais l'expression de g contient la constante $E_f(h(X))$, que nous ne connaissons pas.

un candidat pertinent est tel que $\frac{|h|f}{g}$ soit quasi constant et de variance finie.

5- Application de la méthode

5- Application de la méthode

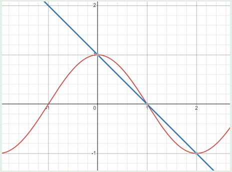
Exercice:

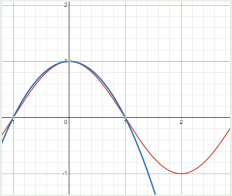
1- Estimer la valeur de $I = \int_0^1 \cos(\pi x/2)\, dx$ par la méthode de Monte-Carlo et appliquer une amélioration de cette valeur à l'aide de la méthode de l'échantillonnage préférentiel.

5- Application de la méthode

Exercice:

1- Estimer la valeur de $I = \int_0^1 \cos(\pi x/2)\, dx$ par la méthode de Monte-Carlo et appliquer une amélioration de cette valeur à l'aide de la méthode de l'échantillonnage préférentiel.

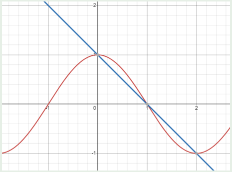


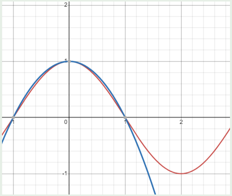


5- Application de la méthode

Exercice:

1- Estimer la valeur de $I = \int_0^1 \cos(\pi x/2)\, dx$ par la méthode de Monte-Carlo et appliquer une amélioration de cette valeur à l'aide de la méthode de l'échantillonnage préférentiel.





2- Donner une implémenter avec Python en précisant les estimations de l'intégrale par Monte Carlo et par échantillonnage préférentiel ainsi que les variances de l'estimation.

```python
import numpy as np

# Fonction à intégrer
def h(x):
    return np.cos(np.pi * x / 2)

# Instrumentale linéaire
def g(x):
    return 2 * (1 - x)

# Instrumentale quadratique
def g_2(x):
    return 3/2 * (1 - x**2)

# méthode 1: Estimation de l'intégrale par la méthode de Monte-Carlo
def monte_carlo(n):
    x_samples = np.random.uniform(0, 1, n)
    integral_estimate = np.mean(h(x_samples))
    return integral_estimate

# méthode 2: échantillonnage préférentiel
def importance_sampling(n):
    # Générer des échantillons aléatoires selon la distribution de densité g(x)
    x_samples = 1 - np.sqrt(np.random.uniform(0, 1, n))
    # Calculer la moyenne de la fonction évaluée en ces échantillons, ajustée par la densité g(x)
    integral_estimate = np.mean(h(x_samples) / g(x_samples))
    return integral_estimate
```

```python
n = 100000

# Estimation de l'intégrale par la méthode de Monte-Carlo
monte_carlo_estimate = monte_carlo(n)
print("Estimation de l'intégrale par Monte-Carlo:", monte_carlo_estimate)

# Estimation de la variance par la méthode de Monte-Carlo
monte_carlo_samples = [monte_carlo(n) for _ in range(10)]  # On effectue plusieurs estimations pour calculer la variance
monte_carlo_variance = np.var(monte_carlo_samples)
print("Variance de l'estimation par Monte-Carlo:", monte_carlo_variance)

# Estimation de l'intégrale par la méthode de l'échantillonnage préférentiel
importance_sampling_estimate = importance_sampling(n)
print("Estimation de l'intégrale par échantillonnage préférentiel:", importance_sampling_estimate)

# Estimation de la variance par la méthode de l'échantillonnage préférentiel
importance_sampling_samples = [importance_sampling(n) for _ in range(10)]
importance_sampling_variance = np.var(importance_sampling_samples)
print("Variance de l'estimation par échantillonnage préférentiel:", importance_sampling_variance)
```

```python
n = 100000

# Estimation de l'intégrale par la méthode de Monte-Carlo
monte_carlo_estimate = monte_carlo(n)
print("Estimation de l'intégrale par Monte-Carlo:", monte_carlo_estimate)

# Estimation de la variance par la méthode de Monte-Carlo
monte_carlo_samples = [monte_carlo(n) for _ in range(10)]  # On effectue plusieurs estimations pour calculer la variance
monte_carlo_variance = np.var(monte_carlo_samples)
print("Variance de l'estimation par Monte-Carlo:", monte_carlo_variance)

# Estimation de l'intégrale par la méthode de l'échantillonnage préférentiel
importance_sampling_estimate = importance_sampling(n)
print("Estimation de l'intégrale par échantillonnage préférentiel:", importance_sampling_estimate)

# Estimation de la variance par la méthode de l'échantillonnage préférentiel
importance_sampling_samples = [importance_sampling(n) for _ in range(10)]
importance_sampling_variance = np.var(importance_sampling_samples)
print("Variance de l'estimation par échantillonnage préférentiel:", importance_sampling_variance)
```

```
Estimation de l'intégrale par Monte-Carlo: 0.6370235728672002
Variance de l'estimation par Monte-Carlo: 1.445408233255845e-06
Estimation de l'intégrale par échantillonnage préférentiel: 0.6362394990834955
Variance de l'estimation par échantillonnage préférentiel: 6.092557968093681e-08
```