
DEVOIR MAISON : numéro 1

Vous devez charger votre **ipynb** sur eCampus, avant le dimanche 18/10/2020 23h59. Il vous est possible de charger un screenshot additionnel pour les questions de maths. Votre nom ne doit apparaître nulle part y compris dans le nom du fichier lui-même.

Rappel : aucun travail par mail accepté !

Intégration de Monte Carlo par moindres carrées

On considère un **problème d'intégration** d'une fonction $f : \mathcal{X} \rightarrow \mathbb{R}$, avec $\mathcal{X} \subset \mathbb{R}$ par rapport à une mesure de probabilité μ . En pratique on regardera des problèmes numériques avec une mesure de probabilité qui sera uniforme (resp. exponentielle ou gaussienne) et on aura donc $\mathcal{X} = [0, 1]$ (resp. $\mathcal{X} = \mathbb{R}_+$ ou \mathbb{R}). Le but est de calculer l'intégrale

$$I(f) = \int_{\mathcal{X}} f(x) d\mu(x) = \mathbb{E}_{X \sim \mu}[f(X)]$$

Lorsqu'il n'y a pas d'expression exacte de cette intégrale, ou bien que la primitive est lourde à calculer, on a recours à une procédure aléatoire de type Monte Carlo. On a accès à un simulateur suivant la loi μ et on peut évaluer la fonction f de manière exacte. Une méthode de Monte Carlo consiste à :

- Tirer des points aléatoires, appelés “particles”, $X_1, \dots, X_n \sim \mu$ suivant la loi μ .
- Evaluer la fonction objectif f aux particles : $(f(X_1), \dots, f(X_n))$
- Construire un estimateur de Monte Carlo $\hat{I}_n(f)$ à l'aide de la statistique $(X_1, f(X_1), \dots, X_n, f(X_n))$

Pour les estimateurs de $I(f)$, les 2 propriétés suivantes sont désirables : être sans biais, i.e., $\mathbb{E}[\hat{I}_n(f)] = I(f)$ et avoir une faible variance (pour garantir une bonne stabilité et précision du résultat). Soit $n \geq 1$, on tire des particles X_1, \dots, X_n selon la loi μ . L'estimateur de Monte-Carlo de $I(f)$ est donnée par

$$\hat{I}_n(f) = n^{-1} \sum_{i=1}^n f(X_i)$$

- 1) Calculer le biais et la variance de $\hat{I}_n(f)$.
- 2) Etant donné $n \geq 1$ fixé, on souhaite générer les particles $X_1, \dots, X_n \sim \mu$. Ecrire une fonction **draw_sample** qui prend en entrée un entier n et une chaîne de caractère “law”, qui renvoie n échantillons tirés suivant la loi “law”. En pratique on pourra avoir law='uniform' (la loi uniforme sur $[0, 1]$), 'exponential' (la loi exponentielle de paramètre 1), 'normal' (la loi normale centrée réduite).
- 3) Ecrire une fonction **naive_MC** qui prend en entrée une fonction f ainsi qu'un tableau de particles $X = (X_1, \dots, X_n)$ et qui renvoie l'estimateur $\hat{I}_n(f)$. On testera cette fonction avec $f = 1_{[0, 1/2]}$ et μ la loi uniforme sur $[0, 1]$. On rappelle que pour tout ensemble A , 1_A est défini comme suit $1_A(x)$ vaut 1 si $x \in A$ et 0 sinon. En répétant l'expérience 100 fois, et en faisant varier $n = 20, 50, 100, 200$ illustrer votre réponse à la question 1 (notamment la décroissance de la variance par rapport à n). Indication : pour chaque valeur de n , on pourra calculer 100 fois l'estimateur $\hat{I}_n(f)$ et à l'aide de ces 100 valeurs évaluer la variance et le biais.

- 4) En utilisant le théorème central limite, construire un interval de confiance IC à 95% pour $\hat{I}_n(f)$. On justifiera clairement sa réponse. Ensuite, on vérifiera en repetant l'expérience 100 fois, que $\mathbb{P}(I(f) \in IC)$ est proche de 0.95.

Afin de réduire la variance de cet estimateur, deux options s'offrent à nous : (i) augmenter la taille n de l'échantillon tiré (mais cela peut être prohibitif en pratique car on a par exemple un budget limité pour le simulateur), (ii) réduire la variance, ce qui est possible via une technique appelée "variables de contrôle". On s'intéresse aux variables de contrôle dans la suite. On suppose que l'on dispose de $m \geq 1$ fonctions h_1, \dots, h_m dont on connaît l'intégrale contre la mesure μ . Sans perte de généralité, on suppose que ces fonctions, dites "variables de contrôle" sont d'intégrale nulle, i.e.

$$\forall k = 1, \dots, m \quad \int_{\mathcal{X}} h_k(x) d\mu(x) = 0$$

On obtient ainsi une classe d'estimateurs $\hat{I}_n(f, \beta)$ paramétrée par le coefficient β :

$$\forall \beta \in \mathbb{R}^m, \quad \hat{I}_n(f, \beta) = \frac{1}{n} \sum_{i=1}^n \left\{ f(X_i) - \sum_{j=1}^m \beta_j h_j(X_i) \right\}$$

Notons qu'en prenant $\beta = 0$, on retrouve l'estimateur simple $\hat{I}_n(f, 0) = \hat{I}_n(f)$ donc cette forme est plus générale. En vue d'opérer une réduction de la variance, il est naturel de chercher le β^* optimal qui permet de réduire la variance de l'estimateur :

$$\beta^* \in \arg \min_{\beta \in \mathbb{R}^m} \mathbb{E} \left[(f - I(f) - \beta^T h)^2 \right]$$

avec $h = (h_1, \dots, h_m)^T$. L'estimateur associé est alors $\hat{I}_n(f, \beta^*)$ mais β^* est inconnu. En pratique on l'estime par moindres carrés :

$$\beta_n^{ols} \in \arg \min_{\beta \in \mathbb{R}^m} \|f^{(n,c)} - H^{(c)}\beta\|^2,$$

où $f^{(n)} = (f(X_1), \dots, f(X_n))^T$ est le vecteur des évaluations de f , $H = (h_j(X_i)), i = 1, \dots, n, j = 1, \dots, m$ est la matrice des régresseurs. L'indice (c) signifie que ces quantités ont été centrées empiriquement, i.e., $H^{(c)}$ est tel que $H_c^T 1_n = 0$ (ou encore que chaque colonne de la matrice $H^{(c)}$ est centrée). L'estimateur obtenu est

$$\hat{I}_n^{ols}(f) = I_n(f, \beta_n^{ols}).$$

Pour les simulations numériques, on utilisera comme fonctions de contrôle h_1, \dots, h_m des polynômes orthogonaux de différents degré. Plus précisément, on utilisera les polynômes de Laguerre et Hermite. Les polynômes de Laguerre Lag_k sont orthogonaux par rapport à la mesure exponentielle https://fr.wikipedia.org/wiki/Polyn%C3%B4me_de_Laguerre, i.e., $\int_0^{+\infty} Lag_k(x) e^{-x} dx = 0$. Les polynômes de Hermite Her_k sont orthogonaux par rapport à la mesure Gaussienne https://fr.wikipedia.org/wiki/Polyn%C3%B4me_d%27Hermite. Ces familles forment des variables de contrôle au sens où $\int h_k d\mu = 0$ pour les mesures décrites précédemment. De plus elles sont orthogonales au sens où, $\forall i \neq j$

$$\int_0^{+\infty} Lag_i(x) Lag_j(x) e^{-x} dx = \int_{-\infty}^{+\infty} Her_i(x) Her_j(x) e^{-x^2/2} dx = 0.$$

- 5) Soit $\beta \in \mathbb{R}^m$ et $\bar{h} = n^{-1} \sum_{i=1}^n h(X_i)$, justifier que $\hat{I}_n(f, \beta) = \hat{I}_n(f) - \beta^T \bar{h}$ est un estimateur sans biais de $I(f)$.
- 6) En utilisant les fonctions `lagval`, `hermeval` de Numpy, écrire des fonctions `laguerre(k,t)`, `hermite(k,t)` qui prenne un entier k (le degré du polynôme) ainsi qu'un réel t et qui renvoient la valeur du polynôme considéré de degré k au point t . Afficher les courbes pour les premiers degrés $k = 1, 2, 3, 4$. Notons que le degré 0, qui correspond dans chacun des cas à la fonction constante, n'est pas intéressant pour nous car la propriété de biais nul évoquée lors de la question précédente serait perdue.

- 7) En utilisant les fonctions de la question précédente, écrire une fonction `get_H(X,m,basis)` qui prend en entrée un tableau de particules $X = (X_1, \dots, X_n)$, le nombre m de variables de contrôle et la base "basis" (string) de la famille considérée (basis \in `laguerre`, `hermite`) et qui construit la matrice $H = (h_j(X_i))$ des régresseurs (dans la matrice H le coefficient d'indice (i, j) correspond à l'évaluation de la fonction h_j au point X_i). Pour remplir cette matrice, on peut voir cela en colonnes : il y a m colonnes dans la matrice H , chaque colonne correspond à un h_j évalué sur l'ensemble des particules (X_1, \dots, X_n) (c'est un bien une colonne de taille $n \times 1$, et on a m telles colonnes au total). Ici, on fixe $m = 30$. Tracer les valeurs propres de la matrice $H^T H$ pour chacune des bases en question et pour des tirages de $n = 25, 50$ et 100 particules.

Dans la suite, on veut intégrer les fonctions suivantes :

$$f_2(x) = 2\sqrt{x}, \quad f_t(x) = 1_{\{x < t\}},$$

avec $t = 1.645$, selon les mesures suivantes

$$\begin{aligned} \text{Exponentielle pour } f_2, \quad & \int_0^{+\infty} f_2(x) e^{-x} dx = \sqrt{\pi} \\ \text{Normale pour } f_t, \quad & \mathbb{P}(X < t) = \int_{-\infty}^t \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = 0.95. \end{aligned}$$

- 8) Pour des tirages de $n = 25, 50$ et 100 particules, tracer le vecteur des coefficients β_n^{ols} pour f_2 et f_t .
- 9) Écrire une fonction `ols_MC(f,X,m,basis)` qui estime l'intégrale $I(f)$ par la méthode des moindres carrés. Dans chacun des cas décrits plus haut, vérifier que votre méthode d'intégration donne des résultats attendus (du moins lorsque m est bien choisi). On pourra par exemple, tracer des courbes d'erreur en fonction de $n = 25, 50, 100$ et essayer différentes valeurs de m parmi $2, 4, 8, 16$.
- 10) Afin de comparer la variance des estimateurs, on répète la simulation 100 fois. A chaque étape on enregistre l'erreur. Puis, on affiche les boxplots des erreurs pour visualiser leur dispersion. Tracer les boxplot représentent l'évolution des erreurs en fonction de $m = 2, 4, 8, 16$ et pour $n = 200$.
- 11) Pour s'affranchir de la limite $m < n$, on va effectuer de la sélection de variables. En s'inspirant de l'estimateur OLS, écrire un estimateur de Monte Carlo par régression Lasso. Comparer cet estimateur avec les précédents.