
Travail pratique

Simulation de Monte-Carlo

Principaux objectifs

- ▷ Réaliser une simulation de Monte-Carlo afin d'estimer la valeur d'une intégrale multiple.
- ▷ Calculer un intervalle de confiance et confirmer empiriquement son seuil de couverture.
- ▷ Présenter proprement des résultats scientifiques et les discuter.

Description du problème

Le problème qui nous intéresse dans ce travail pratique est celui consistant à calculer la distance moyenne entre deux points du carré unité. Plus précisément on désire calculer l'espérance de la distance euclidienne séparant deux points indépendants et identiquement distribués dans le carré unité $[0; 1] \times [0; 1]$.

Si on note $P(x_1; y_1)$ et $Q(x_2; y_2)$ les deux points et D la distance euclidienne qui les sépare, l'espérance de cette dernière est

$$E(D) = \int_0^1 \int_0^1 \int_0^1 \int_0^1 \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} dy_2 dy_1 dx_2 dx_1.$$

Estimation de $E(D)$ à l'aide d'une simulation de Monte-Carlo

Afin de réaliser un programme permettant de calculer un estimateur \hat{D} de l'espérance $E(D)$, ainsi qu'un intervalle de confiance (au seuil de 95%) pour cette espérance, vous partirez des sources contenues dans l'archive `MonteCarlo.zip` disponible sur le serveur de fichiers.

Votre première tâche consistera à programmer la méthode `simulateTillGivenCIHalfWidth` de la classe `MonteCarloSimulation`. Cette méthode doit permettre de simuler une expérience aléatoire donnée jusqu'à ce que la demi-largeur de l'intervalle de confiance (au seuil de 95%) pour la mesure de performance associée passe en dessous d'une valeur maximale fixée.

Plus précisément cette méthode utilise trois paramètres spécifiques :

- ▷ la demi-largeur maximale Δ_{\max} (`maxHalfWidth` dans les sources) de l'intervalle de confiance (au seuil de 95%) produit ;
- ▷ le nombre N_{init} (`initialNumberOfRuns` dans les sources) de réalisations à générer initialement ;
- ▷ le nombre N_{add} (`additionalNumberOfRuns` dans les sources) de réalisations supplémentaires à générer tant que la borne Δ_{\max} pour la demi-largeur de l'intervalle de confiance n'est pas atteinte.

Le fonctionnement de la méthode est alors le suivant.

- 1) Dans une première phase N_{init} simulations de l'expérience sont effectuées.
- 2) À partir des données récoltées une estimation du nombre N de réalisations à générer afin d'obtenir un intervalle de confiance dont la demi-largeur ne dépasse pas Δ_{\max} est effectuée. Cette valeur de N est ensuite arrondie, vers le haut, au plus proche multiple de N_{add} .

- 3) La simulation est poursuivie jusqu'à atteindre N réalisations de l'expérience.
- 4) Si la demi-largeur de l'intervalle de confiance, calculé sur la base de ces N réalisations, est inférieure ou égale à Δ_{\max} le processus s'arrête. Sinon N_{add} simulations supplémentaires sont effectuées avant de recalculer un nouvel intervalle de confiance et de retester la condition d'arrêt. Ce processus est répété jusqu'à ce que la condition d'arrêt soit satisfaite.

Expériences et études à réaliser

Une fois votre programme de simulation fonctionnel vous l'exécuterez avec les valeurs suivantes des paramètres :

$$\Delta_{\max} = 5 \cdot 10^{-5}, \quad N_{\text{init}} = 10^6 \quad \text{et} \quad N_{\text{add}} = 10^5.$$

Vous confirmerez expérimentalement le bon comportement de votre programme en étudiant également la distribution des estimateurs ponctuels \hat{D} obtenus ainsi que la couverture empirique des intervalles de confiance calculés. Pour ce dernier point vous utiliserez le fait que la valeur exacte de l'espérance de D est

$$E(D) = \frac{2 + \sqrt{2} + 5 \ln(\sqrt{2} + 1)}{15} \simeq 0,5214054.$$

Documents à rendre

En plus du code source de votre projet, intelligemment documenté, vous rendrez un exécutable permettant de résoudre la première question de la section précédente (calcul d'un intervalle de confiance pour $E(D)$ d'une largeur ne dépassant pas 10^{-4}) ainsi qu'un court document, au format PDF, contenant une présentation, précise et soignée, et une discussion des résultats de vos expériences.

Modalités et délais

- ▷ Le travail de programmation est à effectuer en Java, version 8 ou 11.
- ▷ L'analyse statistique et les graphiques peuvent être réalisés avec le logiciel de votre choix (**R**, Excel, ...).
- ▷ Au début de chaque étude, vous initialiserez votre générateur de nombres pseudo-aléatoires avec la graine 20 190 528.
- ▷ Vos documents sont à envoyer à l'adresse jean-francois.heche@heig-vd.ch dans une archive zip dont le nom respectera impérativement le format

SIO_TP2_Nom_Prenom.zip

- ▷ Cette archive contiendra un répertoire du même nom (sans l'extension évidemment) qui contiendra votre rapport au format PDF (dont le nom respectera le même format que l'archive mais avec une extension **.pdf**), le code source de votre projet et l'exécutable demandé.
- ▷ Le délai officiel pour l'envoi de votre travail est fixé au **mardi 11 juin 2019 avant minuit**. Vos contributions seront néanmoins acceptées, sans pénalités, jusqu'au lundi 17 juin 2019 à 9 heures.