Sup Galilée
MACS1

février 2022

Initiation au calcul numérique – TP Proba. 4

Simulation de lois conditionelle. Simulation par rejet

Les exercices marqués d'un astérisque sont à réaliser en priorité.

Avant de commencer ce TP, créer un sous-répertoire `TP_Proba_4`, et créer un script (vide) `TP.m` dans ce sous-répertoire.

L'objectif de ce TP est d'étudier comment simuler une expérience sur laquelle on dispose d'une information : comment simuler l'expérience *sachant* qu'un certain événement est réalisé ? Cette méthode, appelée **méthode de rejet**, est ensuite exploitée pour simuler des lois à densité.

1 Simulation de loi conditionnelle

Supposons que l'on souhaite simuler des variables aléatoires $X, Y, \ldots$ sachant qu'un événement $A = \{(X, Y, \ldots) \in B\}$ est réalisé. On suivra le schéma suivant :

```
Faire
        simuler X, Y, ...;
Jusqu'à ce que l'événement A soit réalisé par X,Y,...,
Renvoyer X, Y, ...;
```

On dit que l'on **rejette** les tirages tant qu'ils ne satisfont pas la condition souhaitée.

Dans Octave, cela correspond à une boucle `do ... until(condition)`. Dans les langages (comme Matlab) qui ne disposent que des boucles `while(condition) ... end`, on écrira plutôt :

```
simuler X,Y,...;
Tant_que l'événement A n'est pas réalisé par X,Y,...,
        simuler X, Y,...;
Fin
Renvoyer X, Y,...;
```

Exercice 1* : Justification de la méthode

On rappelle que chaque fois que l'on "simule X", on considère en fait une nouvelle variable aléatoire indépendante des précédentes, et ayant même loi que X. Cela revient à faire à nouveau appel à `rand`. Le programme ci-dessus calcule donc X_1, puis X_2, etc., jusqu'à ce qu'on ait $X_N \in B$, autrement dit on s'arrête au plus petit N pour lequel $X_N \in B$.

Formalisons donc le fonctionnement du schéma précédent : (pour simplifier, on considère une seule variable X, à valeurs dans $\mathbb{R}^d$, et un événement $A = \{X \in B\}$)

Proposition 1.1. *Soit $X_1, X_2, \ldots$ une suite de variables aléatoires indépendantes et de même loi qu'une variable aléatoire X, à valeurs dans $\mathbb{R}^d$. Soit $B \in \mathcal{B}(\mathbb{R}^d)$ tel que $P(X \in B) > 0$. On note*

$$N = \inf\{n \geq 1 \,|\, X_n \in B\}.$$

Alors X_N suit la loi de X sachant $\{X \in B\}$, N suit la loi géométrique de paramètre $P(X \in B)$, et N et X_N sont indépendantes. En particulier,

$$E[N] = \frac{1}{P(X \in B)}.$$

1. Notons $p = P(X \in B)$. Démontrer que, pour tout $A \in \mathcal{B}(\mathbb{R}^d)$ et tout $n \in \mathbb{N}^*$,

$$P(N = n,\ X_N \in A) = (1-p)^{n-1} p P(X \in A \,|\, X \in B).$$

Expliciter ce que signifie "$N = n$" en fonction de $X_1, X_2, \ldots, X_N$.

2. Conclure.

Indication : a) Prendre $A = \mathbb{R}^d$ pour en déduire la loi de N. b) Toujours en partant de la formule ci-dessus, sommer sur $n \in \mathbb{N}^*$ pour en déduire la loi de X_N. c) Enfin, en déduire que N et X_N sont indépendantes.

Exercice 2* : Mise en œuvre

1. Écrire une fonction `x=simul_cond(a)` qui renvoie une réalisation d'une variable aléatoire de loi exponentielle de paramètre 1, conditionnée à appartenir à $[0, a]$.

2. (Dans `ex2.m`) Estimer, par la méthode de Monte-Carlo, $E\big[X^2 \,\big|\, X < 1\big]$, où X suit la loi exponentielle de paramètre 1.

3. Écrire une fonction `[x y]=simul_disque(R)` qui renvoie une réalisation de deux variables aléatoires X et Y indépendantes, de loi uniforme sur $[-R, R]$, conditionnées à ce que $X^2 + Y^2 < R^2$. (Dans `ex2.m`) Représenter graphiquement dans le plan 1000 tirages renvoyés par cette fonction pour $R = 1$.

2 Simuler une loi à densité par rejet

On souhaite simuler une variable aléatoire de densité f. On suppose que

$$\text{pour tout } x \in \mathbb{R}, \qquad f(x) \leq Cg(x),$$

où g est une densité (pour laquelle on *sait déjà* simuler des variables aléatoires), et C est une constante.

Le schéma de la méthode est le suivant :

```
Répéter
        simuler U de loi uniforme sur [0,1];
        simuler Y de densité g;
Jusqu'à avoir C*g(Y)*U<=f(Y);
Renvoyer Y;
```

Ou, avec une boucle `while`,

```
U=2;Y=0;
Tant que C*g(Y)*U>f(Y),
        simuler U de loi uniforme sur [0,1];
        simuler Y de densité g;
Fin
Renvoyer Y;
```

NB. La valeur initiale de `U` sert uniquement à garantir que le code dans la boucle est exécuté au moins une fois. Le code avec "Répéter... Tant que" est donc plus naturel.

Exercice 3* : Justification de la méthode

1. Justifier qu'il suffit de prouver la propriété suivante :

Proposition 2.1. *On suppose que Y et U sont des variables aléatoires indépendantes, telles que Y a pour densité g, et que U suit la loi uniforme sur $[0, 1]$. Pour tout borélien A de $\mathbb{R}$,*

$$P(Y \in A \,|\, Cg(Y)U \leq f(Y)) = \int_A f(x)dx.$$

Autrement dit, la loi de Y sachant $\{Cg(Y)U \leq f(Y)\}$ est la loi de densité f.

2. Écrire $P(Y \in A,\ Cg(Y)U \leq f(Y))$ sous la forme d'une intégrale double.

3. En intégrant une fois, en déduire la valeur de cette probabilité.

4. En prenant $A = \mathbb{R}$, en déduire $\mathbb{P}(Cg(Y)U \leq f(Y))$, et conclure la preuve.

5. Quel est le nombre moyen d'itérations de l'algorithme ? (penser à la partie 1)

Exercice 4* : Mise en œuvre sur des exemples

On veillera impérativement à **tester** la validité des programmes.

1. Écrire une fonction `simul_demicercle()` pour simuler la loi de densité $f : x \mapsto \frac{2}{\pi}\sqrt{1 - x^2}\mathbf{1}_{[-1,1]}(x)$ à partir de la loi uniforme sur $[-1, 1]$.

2. Écrire une fonction `simul_absnormale()` pour simuler la loi de densité $f : x \mapsto \frac{2}{\sqrt{2\pi}}e^{-x^2/2}\mathbf{1}_{[0,+\infty[}(x)$ à partir de la loi exponentielle de paramètre 1 (cf. TP3).

3. Si X a la densité f de la question précédente, et si Z a pour loi $P(Z = +1) = \frac{1}{2} = P(Z = -1)$ et est indépendante de X, alors XZ suit la loi $\mathcal{N}(0, 1)$. En admettant ce résultat, écrire une fonction `simul_normale()` pour simuler la loi normale $\mathcal{N}(0, 1)$.

Exercice 5 : Un autre exemple

1. Déterminer un réel $C > 0$ tel que, pour tout x, $\frac{1}{2}\sin(x)\mathbf{1}_{[0,\pi]}(x) \leq \frac{C}{\pi}\mathbf{1}_{[0,\pi]}(x)$.

2. En déduire une fonction `simul_sinus()` qui simule la loi de densité $f : x \mapsto \frac{1}{2}\sin(x)\mathbf{1}_{[0,\pi]}(x)$ par la méthode du rejet, à partir de la loi uniforme sur $[0, \pi]$ (que l'on sait simuler : si U suit la loi uniforme sur $[0, 1]$, alors πU suit la loi uniforme sur $[0, \pi]$).

3. Valider cette fonction en comparant un histogramme normalisé de $N = 10000$ tirages fournis par cette fonction avec la densité attendue.