MÉTHODE DE MONTE-CARLO

EXERCICES DU 23 FÉVRIER 2015

par Rémi Peyre

EXERCICE 1 — Grandes déviations d'un tirage à pile ou face

Dans cet exercice, on considère 100 lancers successifs d'une pièce de monnaie équilibrée; la probabilité correspondant à cette situation aléatoire est notée $\mathbb{P}$. On note X la variable aléatoire correspondant au nombre de « pile » obtenus; et $f := (X - 60)_+^{[*]}$. Le but de cet exercice est d'estimer $\mathbb{E}(f)$ aussi précisément que possible.

1. Écrire un algorithme « naïf » de Monte-Carlo pour estimer $\mathbb{E}(f)$, appelé PF60_naif. Cette fonction prendra en argument le nombre de simulations demandées; affichera la valeur estimée ainsi que sa marge d'erreur à 95 % de confiance; et ne renverra rien.

On peut imaginer la même expérience, mais où la pièce de monnaie serait piéée et aurait 60 % de chances de tomber sur « pile » à chaque fois. La probabilité correspondant à cette situation est notée $\mathbb{P}'$.

2. Justifier qu'il est légitime de considérer que les probabilités $\mathbb{P}$ et $\mathbb{P}'$ portent sur un même espace mesurable Ω ; proposer ce que pourrait être cet Ω ainsi la façon d'interpréter ses éventualités^[†].

Indication : Pour la suite de l'exercice, le compositeur de la question a en tête qu'on propose un choix de Ω qui soit naturel, et qui constitue en un ensemble fini de grand cardinal.

3. Quelle est la densité de $\mathbb{P}$ par rapport à $\mathbb{P}'$?

Indication : La réponse est $(5/6)^X (5/4)^{100-X}$; encore faut-il le justifier...

4. En déduire un algorithme d'échantillonnage préférentiel PF60_pref pour le calcul de $\mathbb{E}(f)$. (Cet algorithme utilisera la même interface que PF60_naif; seule la structure interne sera différente).

EXERCICE 2 — La grande suite (I)

Nous considérons dans cet exercice la situation d'un joueur de yahtzee^[‡] qui cherche à obtenir une grande suite, c.à.d. la séquence des chiffres de '2' à '6'^[§]. Notant A cet évènement et p sa probabilité, p peut être estimée par la méthode de Monte-Carlo à l'aide du programme « naïf » suivant :

```
% "Nsim" est le nombre de simulations demandées.
function GS_naif (Nsim)
succes = 0;
% Boucle de Monte-Carlo.
for i = 1: Nsim
```

[*]. Pour $a \in \mathbb{R}$, a_+ désigne la partie positive de a , c.à.d. $a_+ := \max(a, 0)$.

[†]. C.à.d. de dire à quels phénomène correspondrait chaque $\omega \in \Omega$.

[‡]. Le yahtzee est un jeu très connu, qui se joue avec 5 dés. Après avoir lancés les dés une première fois, le joueur choisit les dés qu'il souhaite relancer (cela peut éventuellement être aucun, ou tous) et les relance; puis recommence une fois cette étape de relancer partiel (en changeant, s'il le veut, le choix des dés à relancer). Le but est d'obtenir une certaine combinaison sur les chiffres affichés par les dés à l'issue de ce protocole. (Les dés sont interchangeable: savoir quel dé est tombé sur quel chiffre n'importe pas pour la combinaison).

[§]. À noter que cette définition de la « grande suite » n'est pas celle de la variante standard du yahtzee (pour laquelle la séquence des chiffres de '1' à '5' compte aussi).

```

% "lesdes" est ce qu'affichent les dés.
lesdes = ceil (6 * rand (1, 5));
% Première étape de relancer.
dejavu = zeros (1, 5);
for j = 1: 5
    if lesdes(j) == 1 || dejavu(lesdes(j) - 1)
        lesdes(j) = ceil (6 * rand ());
    else
        dejavu(lesdes(j) - 1) = true;
    end
end
% Seconde étape de relancer.
dejavu = zeros (1, 5);
for j = 1: 5
    if lesdes(j) == 1 || dejavu(lesdes(j) - 1)
        lesdes(j) = ceil (6 * rand ());
    else
        dejavu(lesdes(j) - 1) = true;
    end
end
% L'objectif est-il atteint ?
dejavu = zeros (1, 6);
for j = 1:5
    dejavu(lesdes(j)) = true;
end
if (all (dejavu(2: 6)))
    succes = succes + 1;
end
end
p = succes / Nsim;
fprintf ('Probabilité de succès: %f ± %f.\n', p, ...
        1.96 * sqrt (p * (1 - p) / Nsim));
return
end

```

1. Expliquer à quelle stratégie du joueur correspond le code ci-dessus. (Nous admettrons que cette stratégie est effectivement la meilleure possible).

Indication : La stratégie en question n'a rien que de très intuitif; la question est plutôt de comprendre comment fonctionne le programme et en quoi il suit bien la stratégie naturelle...

2. Notons $n_i^{(k)}$ le nombre de fois que le chiffre ' i ' apparaît parmi les dés à l'issue de k étapes de relancers [¶]. Montrer que, conditionnellement à tout ce qui s'est passé jusqu'à l'issue de la première relance, la probabilité que le joueur atteigne finalement son objectif est égale à

$$\frac{(5 - Z_1)!}{6^{5-Z_1}}, \quad (1)$$

où

$$Z_1 := \sum_{i=2}^6 \mathbf{1}_{\{n_i^{(1)} \geq 1\}}. \quad (2)$$

[¶]. Avec cette notation, le joueur atteint son objectif lorsque $n_i^{(2)} = \mathbf{1}_{\{2 \leq i \leq 6\}} \quad \forall i \in \{1, \dots, 6\}$.

3. En déduire un algorithme `GS_cond` (ayant la même interface que `GS_naif`) évaluant la probabilité de succès à l'aide d'une technique de conditionnement.

EXERCICE 3 — Histoire grecque

On considère un actif financier dont le cours vaut 0 € au jour 0. Chaque jour, le cours de l'actif financier augmente ou diminue indépendamment, d'une quantité aléatoire $\mathcal{N}(-0,2; 1)$ (exprimée en €). On note $\mathbb{P}_0$ la loi de probabilité correspondant à cette situation. On appelle p_0 la probabilité, sous $\mathbb{P}_0$, qu'il existe au moins un jour $j \in \{1, \dots, 5\}$ pour lequel le cours de l'actif dépasse 2 €.

1. Écrire un programme `grecque_base` pour évaluer p_0 par la méthode de Monte-Carlo [||].

La question qu'on se pose est maintenant de savoir comment changerait la valeur p_0 si on s'était trompé dans notre modélisation, et qu'en fait, la loi que suit la variation quotidienne de notre actif financier était $\mathcal{N}(-0,18; 1,1)$ [**]. On note $\mathbb{P}_1$ la loi associée à cette nouvelle situation aléatoire, et p_1 la probabilité correspondante que l'actif dépasse 2 €.

2. Comment réaliser un couplage naturel entre $\mathbb{P}_0$ et $\mathbb{P}_1$ [††] ?

3. À l'aide du couplage trouvé à la question précédente, écrire un algorithme `grecque_coup1` [††] qui évalue la différence $p_1 - p_0$ par la méthode de Monte-Carlo avec technique de couplage.

4. Comparer l'efficacité à celle qu'on aurait obtenue en l'absence de couplage.

[||]. La fonction `grecque_base` prendra en argument le nombre de simulations demandées, affichera la valeur estimée de p_0 ainsi que la marge d'erreur (au risque 5 %), et ne renverra rien.

[**]. Attention : le second paramètre d'une loi normale exprime sa *variance*, pas son écart-type...!

[††]. Rappelons qu'un tel « couplage » consiste à construire une situation aléatoire dans laquelle chaque éventualité fournit *deux* trajectoires $(X_i^{(0)})_{i \in \{1, \dots, 5\}}$ et $(X_i^{(1)})_{i \in \{1, \dots, 5\}}$, qui soient telles que $(X_i^{(0)})_i$ suive la loi $\mathbb{P}_0$ et que $(X_i^{(1)})_i$ suive la loi $\mathbb{P}_1$, et que les trajectoires de $X^{(0)}$ et $X^{(1)}$ «se ressemblent» dans la mesure du possible.

[††]. Dont l'interface sera essentiellement la même que pour `grecque_base`.

EXERCICE 4 — La grande suite (II)

On reprend le paradigme de la grande suite vu dans l'exercice 2; mais cette fois-ci, c'est une technique d'échantillonnage préférentiel qu'on va essayer d'utiliser pour améliorer l'efficacité. Notre nouvelle loi d'échantillonnage est définie en considérant que les dés sont pipés, et ce de la façon suivante : le premier dé a (à chaque lancer) $1/3$ chance de tomber sur '2' et $2/15$ chance de tomber sur n'importe quel autre chiffre; le deuxième dé a $2/3$ chance de tomber sur '3' et $2/15$ chance de tomber sur n'importe quel autre chiffre; etc. On appelle $\mathbb{P}'$ la probabilité correspondant à cette situation de dés pipés.

1. Modifier le code présenté à l'exercice 2 pour que :
 - Le tirage des dés ait lieu selon la loi $\mathbb{P}'$;
 - Outre l'obtention ou non d'une grande suite, on calcule aussi la densité de probabilité $d\mathbb{P}/d\mathbb{P}'$;
 - Mais c'est toujours $p := \mathbb{P}(A)$ qu'on évalue (en utilisant donc la technique de couplage).

Indication : Attention ! La stratégie du joueur dans ce nouveau code doit-elle tenir compte du fait que les dés sont pipés, et s'adapter en conséquence; ou le joueur doit-il continuer à jouer comme s'il ignorait que les dés sont pipés?...

2. (★) Exécutez le code obtenu à la question précédente. Normalement vous constaterez que le résultat est mauvais, voire incompatible avec le précédent... Les soupçons se portent naturellement un problème de sous-échantillonnage, et c'est effectivement le cas. D'où vient ce sous-échantillonnage, à votre avis...?

Pour éviter le problème de sous-échantillonnage, on considère un nouvel événement de succès, noté B , défini par « Le premier dé finit sur '2' (conformément à son pipage), le seconde dé finit sur '3', etc. ». On note q la probabilité de cet événement sous $\mathbb{P}$.

3. Justifier sommairement qu'on devrait avoir $q = p / 120$.

4. En déduire un algorithme d'échantillonnage préférentiel pour évaluer p , mais qui ne présente pas le problème de sous-échantillonnage.

Indication : Notre algorithme calculera de prime abord un estimateur de q ; mais puisqu'on sait que $p = 120q$, on peut en déduire un estimateur de p . Attention ! Il faudra penser à tenir compte de la multiplication par 120 pour transporter la marge d'erreur de l'estimateur de q vers l'estimateur de p ...

Indication : Attention ! Le joueur doit-il tenir compte du fait que le succès consistera en fait à obtenir l'évènement B , et donc adapter sa stratégie en conséquence; ou devra-t-il continuer à jouer comme s'il cherchait à obtenir l'évènement A ?...

5. (★) Exécuter cet algorithme. Normalement ça ne marche toujours pas; et même, c'est pire encore : l'algorithme converge clairement vers une valeur erronée...! Sauriez-vous expliquer ce qui s'est passé; et comment y remédier?

Indication : En fait, le problème est qu'avec la stratégie suivie par notre joueur, on n'a pas $q = p / 120$...! Un élément essentiel manque en effet à la stratégie pour que cette identité soit valide : lequel...? Rajoutez cet élément pour réparer l'erreur!

EXERCICE 5 — Le retour de la finance

On revient dans cet exercice sur le contexte de l'exercice 3. Notre objectif est d'utiliser la technique de conditionnement pour améliorer l'estimation de p_0 . Pour ce faire, nous considérons la sous-tribu engendrée par le couple (X_2, X_4) , notée $\mathcal{A}$.

1. Justifier que, conditionnellement à la tribu $\mathcal{A}$, les v.a. X_1, X_3, X_5 sont indépendantes, et suivent respectivement les lois

$$\mathcal{N}(X_2 / 2, 1/2), \quad \mathcal{N}((X_2 + X_4) / 2, 1/2), \quad \mathcal{N}(X_4, 1). \quad (3)$$

2. Calculer la probabilité que, conditionnellement à la tribu $\mathcal{A}$, le cours de l'actif dépasse 2 €.

Indication : Dans cette question, on considère que nous avons à notre disposition la fonction d'erreur erf^[*].

3. En déduire un algorithme de Monte-Carlo utilisant la technique de conditionnement pour évaluer p_0 plus finement.

EXERCICE 6 — Épidémie

On considère une population de 625 personnes, représentées par un carré de dimensions 25×25 (qu'on représentera dans Matlab sous la forme d'une matrice), parmi lesquelles une épidémie est susceptible de se propager. Les règles de propagation sont les suivantes^[†] :

- Initialement, seule la personne centrale de coordonnées (13, 13) (que nous appellerons patient zéro) est exposée (on représentera cet état par le nombre 1); les autres sont pures (état représenté par le nombre 0).
- Lorsqu'une personne est exposée, on tire une variable aléatoire de Bernoulli de paramètre 0,55 pour savoir ce qu'elle devient. Si la variable aléatoire tirée est positive, la personne devient infectée (état représenté par le nombre 2); si elle est nulle, la personne devient immunisée (état représenté par le nombre -1).
- Lorsqu'une personne est infectée, chacune de ses quatre voisines (ou trois, resp. deux, si on est sur le bord du carré, resp. dans un coin), si elle était pure, devient exposée.

1. (★) Simuler la propagation de l'épidémie.

Indication : Une façon de simuler rapidement l'épidémie est de maintenir à jour la liste des personnes exposées, et de traiter^[‡] les personnes *exposées* dans l'ordre de cette liste. Techniquement, on utilisera trois variables : une matrice `exposees` de taille 2×625 destinée à contenir les coordonnées des personnes exposées (initialisée de sorte que sa première colonne soit $(13, 13)^T$); et deux indices indiquant quelle sont les colonnes de `exposees` correspondant à la première, resp. à la dernière personne exposée encore à traiter (initialisés tous les deux à 1). Chaque fois qu'on tire la variable de Bernoulli d'une personne exposée, on incrémente le premier compteur, puis si la personne s'avère infectée, on rend exposés ses voisins encore pures; et chaque fois qu'une nouvelle personne devient exposée, on incrémente le second compteur et on met à jour la colonne correspondante de la matrice `exposees`.

2. Déterminer par la méthode de Monte-Carlo le nombre moyen de personnes qui seront infectées par l'épidémie (en incluant le patient zéro le cas échéant).

[*]. Qui est implémentée (sous ce nom) dans Matlab.

[†]. On peut démontrer que l'ordre dans lequel on effectue les différentes opérations importe peu, pourvu qu'on aille jusqu'au bout de la simulation en s'assurant qu'il ne reste plus aucune personne *exposée* à la fin.

[‡]. Par « traiter » une personne exposée, je n'entends pas la *soigner* (cela n'aurait d'ailleurs pas de sens ici...), mais déterminer si elle finit par basculer dans l'état infectée (rendant alors exposées ses voisines encore pures) ou dans l'état immunisée.

On s'intéresse maintenant à la possibilité de vacciner une personne prise au hasard. Cela signifie que cette personne démarrera la simulation avec le statut immunisée au lieu de pure (ou, si cette personne se trouve être le patient zéro, que l'épidémie ne démarrera jamais). La question est de savoir « combien d'infections cela évite-t-il, en moyenne ? ».

3. Utiliser une technique de couplage pour répondre à la question posée.

4. Comparer le nombre moyen d'infections évitées avec la probabilité (dans la situation initiale) qu'une personne prise au hasard finisse infectée. Expliquer pourquoi les épidémiologistes affirment que le choix d'être vacciné ou non n'est pas une question purement individuelle, mais engage aussi autrui.