TP Python M1 MEEF - Probabilités

October 7, 2020

1 Aiguilles de Buffon

L'aiguille de Buffon est une expérience de probabilité proposée en 1733 par Georges-Louis Leclerc de Buffon, un scientifique français du XVIIIe siècle. Cette expérience fournit une approximation du nombre π.

Le principe est de lancer un grand nombre d'aiguilles sur un parquet dont les rainures sont espacées d'une longueur d'aiguille. On comptabilise le nombre de lancers où l'aiguille tombe à cheval sur une rainure du parquet (cas "favorable") par rapport au nombre total de lancers. Au fur et à mesure que le nombre de lancers augmente, cette proportion se rapproche d'un certain nombre permettant de retrouver π.

On modélisera ici l'aiguille par un segment du plan de longueur 1 dont la position et l'inclinaison seront tirées au hasard uniformément. Les rainures du parquet seront modélisées par des droites parallèles verticales espacées d'une longueur d'aiguille (i.e. d'équations $x = k$, $k \in \mathbb{N}$).

Question 1.1 *Implémenter une fonction* `tirage_aiguille(xmin, xmax, ymin, ymax)` *qui prend en entrée une fenêtre* $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$*, et qui renvoie les coordonnées des deux extrémités de l'aiguille, dont la position sera tirée comme suit :*

- *La position de la première extrémité est tirée uniformément à l'intérieur de la fenêtre,*
- *La deuxième extrémité est à distance 1 de la première, et l'inclinaison de l'aiguille est tirée uniformément dans* $[0, 2\pi]$.

Question 1.2 *Implémenter une fonction* `check_aiguille(coord)` *qui prend en entrée les coordonnées d'une aiguille et qui renvoie* `True` *si l'aiguille chevauche une rainure du parquet,* `False` *sinon.*

Pour tout $t \in \mathbb{N}$, soit B_t la variable de Bernoulli valant 1 si la t-ième aiguille chevauche une rainure du parquet, 0 sinon. On fait l'hypothèse que x_{min} et x_{max} sont entiers.

Question 1.3 *Montrer que* $\mathbb{P}(B_0 = 1) = \frac{2}{\pi}$ *; on pourra commencer par calculer pour tout* $\theta \in [0, 2\pi]$ *la probabilité* $\mathbb{P}(B_0 = 1 | \Theta = \theta)$*, où* Θ *est l'inclinaison de l'aiguille.*

En déduire que $\frac{1}{T} \sum_{t \leq T} B_t \to \frac{2}{\pi}$ *presque sûrement.*

Question 1.4 *En prenant par exemple* $[x_{min}, x_{max}] \times [y_{min}, y_{max}] = [0, 10]^2$*, tracer l'évolution de* $\frac{1}{T} \sum_{t \leq T} B_t - \frac{2}{\pi}$ *en fonction de* T.

◇ **Bonus 1 :** Vous pouvez dessiner les rainures et les aiguilles avec `matplotlib`.

◇ **Bonus 2 :** La méthode proposée étant aléatoire, avec par exemple des inégalités de concentration on peut obtenir des estimations de π qui seront vraies avec grande probabilité. Quelle sera la vitesse de convergence de $\frac{1}{T} \sum_{t \leq T} B_t$ vers $\frac{2}{\pi}$ (avec grande probabilité) ? Pour des calculs (beaucoup!) plus rapides des décimales de π en base 2 ou 16, vous pourrez regarder la formule de Plouffe.

2 Variable aléatoire sans espérance

Le but de l'exercice est de présenter un contre-exemple à la loi forte des grands nombres quand on enlève l'hypothèse d'espérance finie.

Premier jeu

On considère le jeu suivant :

- Payez la somme initiale de s euros,
- On lance une pièce équilibrée jusqu'à avoir rempli l'une de ces conditions : on a obtenu Face, ou 3 lancers ont été effectués.
- Recevez 2^{N_t} euros, où N_t est le nombre de fois où on a obtenu Pile.

Pour tout $t \in \mathbb{N}$ soit $X_t \doteq 2^{N_t} - s$ la somme obtenue au t-ième essai du jeu.

Question 2.1 *Pour quelle valeur de s le jeu est-il équilibré (i.e. X_0 est d'espérance nulle) ?*

Question 2.2 *Implémenter une fonction* `essai(s)` *qui prend en entrée la somme initiale à payer et qui renvoie la somme obtenue sur un essai du jeu.*

Question 2.3 *Pour tout $t \in \mathbb{N}$, soit S_t la somme dont le joueur dispose après t essais du jeu. Montrer que le gain moyen par tour $\frac{1}{t}S_t$ converge presque sûrement vers $\mathbb{E}[X_0]$.*

Question 2.4 *Implémenter une fonction* `essai_itere(s,T)` *qui renvoie la liste des $\frac{1}{t}S_t$ pour tout $t < T$. Afficher ensuite $\mathbb{E}[X_0]$ ainsi que $\frac{1}{t}S_t$ en fonction de t pour un grand nombre d'essais (par exemple $t < 10000$ et s laissé à votre choix).*

Variante

On considère à présent la variante suivante :

- Payez la somme initiale de s euros,
- On lance une pièce équilibrée jusqu'à obtenir Face.
- Recevez $2^{N'_t}$ euros, où N'_t est le nombre de fois où on a obtenu Pile.

Pour tout $t \in \mathbb{N}$ on note $X'_t = 2^{N'_t} - s$ le gain obtenu au t-ième essai de cette variante.

Question 2.5 *Montrer que $\mathbb{E}[X'_0] = +\infty$ quelle que soit la valeur de s.*

Question 2.6 *Implémenter une fonction* `essai_variante(s)` *qui prend en entrée la somme initiale à payer et qui renvoie la somme obtenue sur un essai de la variante.*

Question 2.7 *Pour tout $t \in \mathbb{N}$, soit S'_t la somme dont le joueur dispose après t essais de la variante. Implémenter une fonction* `essai_variante_itere(s,T)` *qui renvoie la liste des $\frac{1}{t}S'_t$ pour tout $t < T$. Afficher ensuite $\frac{1}{t}S'_t$ en fonction de t pour un grand nombre d'essais (par exemple $t < 10000$ et s laissé à votre choix), est-ce que la suite $(\frac{1}{t}S'_t)_{t\in\mathbb{N}}$ semble converger?*

◇ **Bonus :** La probabilité d'obtenir un gain d'au moins 2^n euros (perte initiale non comptée) est de 2^{-n}. Même avec un coût initial de 1024 euros, le jeu favorise toujours autant le joueur, même s'il a une probabilité de $1 - 2^9 \simeq 0.9980$ de payer au moins 512 euros par essai ; les richesses immenses qu'il pourrait gagner après son 10ème Pile (1 chance sur 1000) compensent infiniment cela, pour le calcul de l'espérance.

3 Codes à longueur variable

On considère ici des ensembles de mots résultant de la concaténation de "mots de base" de longueur variable. Le but de l'exercice sera de déterminer la croissance asymptotique de l'ensemble des mots concaténés ainsi que les distributions empiriques sur les caractères les composant. Cette approche trouve des applications en théorie du codage de source et de canal pour déterminer les types et les débits moyens des codes à longueur variable.

Préliminaires

Soit $\mathcal{X}$ un ensemble fini que nous appellerons *alphabet*. On définit $\mathcal{X}^* \doteq \bigcup_{n\in\mathbb{N}} \mathcal{X}^n$ l'ensemble des mots sur $\mathcal{X}$, avec la convention $\mathcal{X}^0 = \epsilon$, où ϵ est le mot vide. Par exemple, $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, ...\}$.

Soient $w = x_1...x_{|w|}, w' = x'_1...x'_{|w'|}$ deux mots de $\mathcal{X}^*$, on note $|w|$ et $|w'|$ leur longueur respective, et ww' leur concaténation : $ww' = x_1...x_{|w|}x'_1...x'_{|w'|}$, le mot vide ϵ étant l'élément neutre pour la concaténation. Plus généralement, pour tout ensemble de mots $\mathcal{S} \subseteq \mathcal{X}^*$, on note $\mathcal{S}^*$ l'ensemble des mots résultant de la concaténation des mots de $\mathcal{S}$.

Question 3.1 *Prenons comme alphabet $\mathcal{X} = [\![0,2]\!]$, énumérer tous les mots $w \in \{0, 11, 211\}^*$ tels que $|w| \le 4$.*

Question 3.2 *Soit $\mathcal{S} = \{01, 10, 101\}$. Est-ce qu'un mot $w \in \mathcal{S}^*$ se factorise toujours de façon unique en concaténations de mots de $\mathcal{S}$? Illustrer avec un exemple (ou contre-exemple).*

Calcul du débit

Un code à longueur variable est un sous-ensemble fini de mots $\mathcal{C} \subseteq \mathcal{X}^*$. Pour tout $n \in \mathbb{N}$ on définit $\mathcal{C}^*_{[n]} \doteq \{w \in \mathcal{C}^* \mid |w| = n\}$. Le but de cette section sera d'évaluer le débit du code, c'est-à-dire $\lim_{n\to\infty} \sqrt[n]{\#\mathcal{C}^*_{[n]}}$. Le débit correspond au nombre moyen de suites générées par concatenation, par exemple le débit de $[\![1,k]\!]$ est de k, puisque $\#[\![1,k]\!]^*_{[n]} = k^n$.

Question 3.3 *On donne ici le lemme de Fekete : pour toute suite u telle que $u_{n+m} \ge u_n + u_m$ pour tout m, n, $\lim\limits_{n\to\infty} \frac{u_n}{n}$ existe et vaut $\sup\limits_{n} \frac{u_n}{n}$. Montrer que sous l'hypothèse $pgcd_{c\in\mathcal{C}}|c| = 1$, le débit est bien défini. On pourra considérer la suite $(\log \#\mathcal{C}^*_{[n]})_{n\in\mathbb{N}}$.*

Soient w, w' deux mots de $\mathcal{X}^*$, on dit que w est préfixe de w' si w' s'écrit sous la forme $w' = wx_1, ..., x_{|w'|-|w|}$. On dit qu'un code $\mathcal{C}$ est *prefix-free* si pour tout $w, w' \in \mathcal{C}$ tels que $|w| \le |w'|$, w n'est pas préfixe de w'.

Question 3.4 *Le code $\mathcal{C} = \{011, 023, 11, 23, 35, 42, 54\}$ est-il prefix-free ?*

Question 3.5 *Montrer que si $\mathcal{C}$ est prefix-free, alors tout mot de $\mathcal{C}^*$ se factorise de façon unique comme concaténation de mots de $\mathcal{C}$.*

On fera dorénavant l'hypothèse que $\mathcal{C}$ est prefix-free.

Question 3.6 *Soit $\bar{\ell} \doteq \max_{c\in\mathcal{C}} |c|$. Exprimer $\#\mathcal{C}^*_{[n+\bar{\ell}]}$ en fonction de $(\#\mathcal{C}^*_{[n+\ell]})_{\ell<\bar{\ell}}$.*

Question 3.7 *Implémenter une fonction* `sqrt_card_C_star(C,n)` *qui calcule $\sqrt[n]{\#\mathcal{C}^*_{[n]}}$ pour tout $n < N$. En prenant par exemple $\mathcal{C} = \{011, 023, 11, 23, 35, 42, 54\}$, afficher l'évolution de $\sqrt[n]{\#\mathcal{C}^*_{[n]}}$ en fonction de n. La suite $\sqrt[n]{\#\mathcal{C}^*_{[n]}}$ semble-t-elle converger?*

Question 3.8 *On rappelle la formule de Gelfand : pour toute matrice A et toute norme matricielle $\|\cdot\|$, on a $\sqrt[n]{\|A^n\|} \underset{n\to\infty}{\to} \rho(A)$, avec $\rho(A)$ le rayon spectral de A. Déterminer le débit du code $\lim_{n\to\infty} \sqrt[n]{\#\mathcal{C}^*_{[n]}}$ en fonction du rayon spectral d'une matrice bien choisie ne dépendant que de $\mathcal{C}$.*

Question 3.9 *En utilisant par exemple la fonction de décomposition en valeurs singulières de* `numpy.linalg`*, implémenter une fonction* `debit(C)` *qui renvoie le débit du code* `C` *en entrée. Comparer avec la limite trouvée à la Question 3.7.*

$\diamond$ **Bonus :** On peut observer des oscillations dans la question 3.7, leur amplitude détermine la vitesse de convergence de $\sqrt[n]{\#\mathcal{C}^*_{[n]}}$ vers sa limite. Quelle est leur vitesse de décroissance, et comment peut-on la calculer?

Marche aléatoire

Le but de cette section est de retrouver, à l'aide d'arguments probabilistes, une formule un peu plus explicite pour le débit d'un code à longueur variable.

Question 3.10 *Implémenter une fonction* `mot_aleatoire(C,p)` *qui prend en entrée un ensemble de mots* `C` *et une distribution de probabilités* `p` *sur* `C`*, et qui renvoie un mot aléatoire de* `C` *tiré selon* `p`.

Question 3.11 *Implémenter une fonction* `mot_concatene_aleatoire(C,p,n)` *qui prend en entrée un ensemble de mots* `C`*, une distribution de probabilités* `p` *sur* `C` *et une longueur maximale* `n`*. Cette fonction doit renvoyer un mot aléatoire de* `C*` *résultant de la concaténation de mots donnés par* `mot_aleatoire(C,p)`*, avec la condition d'arrêt suivante : la longueur maximale* `n` *a été atteinte ou dépassée.*

On modélise les différents états de la concaténation aléatoire par une marche aléatoire sur le graphe G représenté sur la figure 1, avec n'importe quel sommet de l'ensemble $\mathcal{F}$ comme sommet initial. Pour tout $c \in \mathcal{C}$, le sommet (c, i) représente l'ajout du i-ème caractère de c. Les probabilités de transition sur les arêtes de chaque branche $(c, 1) - (c, 2) - ... - (c, |c|)$ sont toutes de 1.

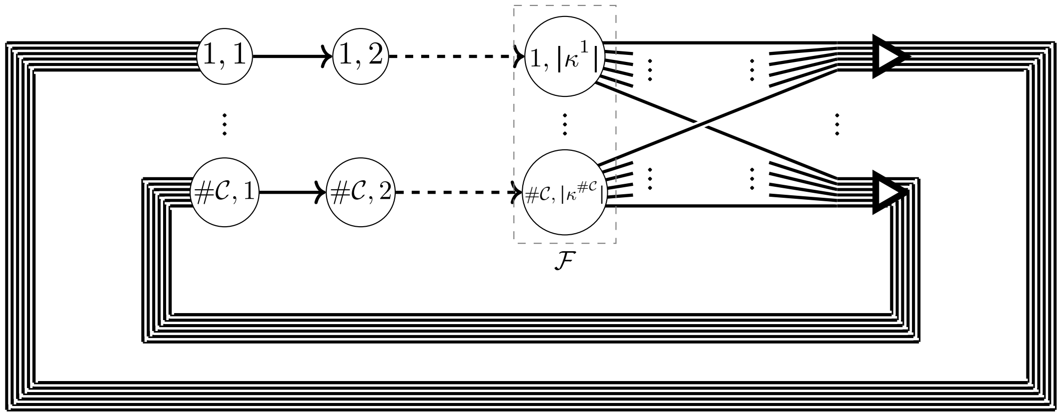

Figure 1: Le graph G correspondant au code $\mathcal{C}$

Question 3.12 *Soient* $\underline{\ell} \doteq \min_{c\in\mathcal{C}} |c|$ *et* $\overline{\ell} \doteq \max_{c\in\mathcal{C}} |c|$*. Montrer qu'il existe un unique réel* $\delta > 0$ *tel que* $\sum_{\ell=\underline{\ell}}^{\overline{\ell}} \#\mathcal{C}_{[\ell]}\delta^{-\ell} = 1$.

Question 3.13 *Déterminer la loi de probabilité* p *de transition depuis chaque sommet* $(c, |c|)$ *telle que pour tout* n*, les probabilités induites sur chaque chemin de longueur* n *terminant dans* $\mathcal{F}$ *soient égales.*

Question 3.14 *En prenant pour* `p` *la distribution trouvée à la question précédente, afficher la fréquence d'apparition des mots générés par* `mot_concatene_aleatoire(C,p,n)` *avec par exemple* $\mathcal{C} = \{011, 023, 11, 23, 35, 42, 54\}$*. NB : pas besoin de prendre un* `n` *grand ici, mais générez un grand nombre de mots ; ceux-ci ne seront pas tous de longueur* `n`*.*

Dans toute la suite, pour tout $c, c' \in \mathcal{C}$, on prendra $\delta^{-|c'|}$ comme probabilité de se rendre de $(c, |c|)$ à $(c', 1)$.

Question 3.15 *En déduire que le débit d'un code $\mathcal{C}$ est la seule solution positive de l'équation polynomiale* $X^{\overline{\ell}} = \sum_{\ell=\underline{\ell}}^{\overline{\ell}} \#\mathcal{C}_{[\ell]} X^{\overline{\ell}-\ell}$.

⋄ **Bonus :** On peut aussi retrouver cette équation polynomiale en cherchant le polynôme carctéristique de la matrice de transition de la question 3.6.