Examen Mag 372 – Modélisation en probabilités 23-04-2024

1. Le sujet comporte 2 exercices indépendants.
2. Vous ne pouvez pas vous connecter à votre session personnelle, vous vous connectez à la session `mathens` avec le mot de passe `mathens`. Tous vos fichiers devront être enregistrés dans le dossier “`remise_copie`” présent sur le bureau. Il faut impérativement inclure votre nom dans le nom des fichiers.
3. L'idéal est de transmettre l'exportation en pdf de votre feuille Jupyter.
4. La réponse aux questions théoriques doit être sur les copies papier.
5. Les graphes doivent avoir un titre et être légendés (les graphes doivent être présents, le code qui les dessine ne suffit pas).
6. Pour toutes les questions qui demandent d'écrire un programme, on demande en plus du programme écrit une exécution du dit programme avec des paramètres convenablement choisis. On rappelle par ailleurs les commandes d'import des bibliothèques adéquates :

   ```python
   import math
   import numpy as np
   import scipy.stats as scs
   import matplotlib.pyplot as plt
   import numpy.random as random
   ```

   Si besoin est, on pourra s'assurer que la dernière version de `matplotlib` (incluant la commande `ecdf` pour la fonction de répartition empirique) est bien installée en tapant dans un terminal

   ```
   pip install matplotlib --upgrade
   ```

Exercice 1. Lois beta discrètes et continues. Soit n, a et b trois entiers strictement positifs. La *loi beta discrète* de paramètres (n, a, b) est la mesure de probabilités sur $\mathbb{N}$ donnée par :

$$\mathbb{P}_{n,a,b}[k] = \begin{cases} \frac{n!\,(k+a-1)!\,(n-k+b-1)!\,(a+b-1)!}{k!\,(n-k)!\,(n+a+b-1)!\,(a-1)!\,(b-1)!} & \text{si } k \leq n, \\ 0 & \text{si } k > n. \end{cases}$$

1. Écrire un programme `verifier(n, a, b)` qui vérifie que la formule ci-dessus donne bien une mesure de probabilités, c'est-à-dire que $\sum_{k=0}^{n} \mathbb{P}_{n,a,b}[k] = 1$. Exécuter ce programme avec $n = 10$ et $a = b = 5$.
2. En utilisant la formule du binôme de Newton $\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$, montrer que

   $$\mathbb{P}_{n+1,a,b}[k] = \frac{n+b-k}{n+a+b}\,\mathbb{P}_{n,a,b}[k] + \frac{k+a-1}{n+a+b}\,\mathbb{P}_{n,a,b}[k-1].$$

   On demande une preuve formelle (pas un programme).
3. On considère une urne contenant initialement $R_0 = a$ boules rouges et $B_0 = b$ boules blanches. À chaque instant $n \geq 0$, on tire au hasard uniformément l'une des $n + a + b = R_n + B_n$ boules de l'urne, et on la remplace par deux boules de la même couleur pour obtenir

une urne au temps $n+1$ avec $n+a+b+1 = R_{n+1}+B_{n+1}$ boules rouges ou blanches. Montrer que la distribution du nombre R_n de boules rouges dans l'urne au temps n est :

$$\mathbb{P}[R_n = a + k] = \mathbb{P}_{n,a,b}[k].$$

En utilisant cette description, écrire un programme `tirer_au_hasard(n, a, b)` qui tire au hasard une variable entière de distribution $\mathbb{P}_{n,a,b}$. Exécuter ce programme 20 fois avec $n = 10$ et $a = b = 5$.

4. Sur un même graphique, dessiner l'histogramme théorique de la distribution $\mathbb{P}_{10,5,5}$, et l'histogramme empirique de $N = 10000$ variables aléatoires $A_1, \ldots, A_{10000}$ indépendantes toutes de loi $\mathbb{P}_{10,5,5}$. Commenter ce dessin.
5. On revient à la suite de variables aléatoires $(R_n)_{n\in\mathbb{N}}$ de la question 3. Posons $X_n = \frac{R_n}{n+a+b}$: cette variable aléatoire est la proportion de boules rouges dans l'urne au temps n. Écrire un programme `proportion(N, a, b)` qui dessine $n \mapsto X_n$ sur l'intervalle $[0, N]$. Avec $a = b = 5$, dessiner plusieurs trajectoires des proportions jusqu'au temps $N = 10000$.
6. Quel type de convergence peut-on conjecturer pour la suite $(X_n)_{n\in\mathbb{N}}$? Sa limite X est-elle déterministe ou aléatoire ? Justifier votre réponse en dessinant la fonction de répartition empirique de X_n pour $n = 1000$, par exemple avec $a = b = 5$.
7. La *loi beta continue* de paramètres $a, b \in \mathbb{N}^*$ est la loi sur $[0, 1]$ de densité

$$f_{a,b}(x)\,\mathrm{d}x = 1_{0<x<1}\, x^{a-1}(1-x)^{b-1}\,\frac{(a+b-1)!}{(a-1)!\,(b-1)!}\,\mathrm{d}x;$$

on ne demande pas de vérifier que $\int_0^1 f_{a,b}(x)\,\mathrm{d}x = 1$. La fonction de répartition de cette loi est obtenue par la commande `scs.beta.cdf(x, a, b)`. Comparer cette fonction de répartition au dessin de la question précédente, et en déduire une conjecture plus précise concernant la limite de la suite de variables aléatoires $(X_n)_{n\in\mathbb{N}}$.

Exercice 2. Loi du demi-cercle. La *loi du demi-cercle* sur l'intervalle $[-2, 2]$ est la loi de densité

$$f(x) = \begin{cases} c\sqrt{4-x^2} & \text{si } x \in [-2, 2]; \\ 0 & \text{si } x \notin [-2, 2]. \end{cases}$$

On souhaite simuler une variable aléatoire X ayant cette loi.

1. Que vaut c pour que f soit une densité de probabilité ? On pourra faire le changement de variables $x = 2\sin\theta$ dans le calcul de l'intégrale. Calculer avec cette valeur de c la quantité $K = \sup(\{f(x) \mid x \in \mathbb{R}\})$.
2. Écrire un programme `pointdemicercle()` qui engendre une suite de points (X_i, Y_i) aléatoires indépendants et uniformément répartis dans le rectangle $[-2; 2] \times [0; K]$ jusqu'à ce que l'un d'entre eux vérifie $Y_i < f(X_i)$, et renvoie alors ce point sous la forme de ses deux coordonnées `[u,v]` .
3. Illustrer votre programme en traçant sur le même graphique la courbe de f et $N = 200$ points engendrés par `pointdemicercle()`.
4. Quelle est la loi de l'abscisse `u` d'un point donné par `pointdemicercle()` ?
5. Écrire un programme `repartition_empirique_demicercle(N)` qui récupère les abscisses de N points engendrés par `pointdemicercle()` et qui dessine la fonction de répartition empirique de ces variables. On testera ce programme avec $N = 100$.

6. Superposer sur un même dessin la fonction de répartition de la loi semi-circulaire (on ne cherchera pas à la calculer on utilisera directement `scs.semicircular.cdf(t, scale=2)` qui donne ses valeurs). Commenter ces dessins.
7. Une *matrice de Wigner* gaussienne de dimension N est une matrice X symétrique de dimension $N \times N$ tel que les entrées $\{X_{ij} : 1 \leqslant i \leqslant j \leqslant N\}$ dont des gaussiennes indépendantes centrées de variance $\frac{1}{N}$. Il y a donc $\frac{N(N+1)}{2}$ variables gaussiennes indépendantes, car les $\frac{N(N-1)}{2}$ variables X_{ij} sous la diagonale ($j < i$) sont égales aux X_{ji} au-dessus de la diagonale.
   — Écrire un programme `spectre(N)` qui construit X une matrice de Wigner gaussienne de dimension N et renvoie les valeurs propres $\lambda_1, \ldots, \lambda_N$ de X. On pourra utiliser `scs.norm.rvs(size=(a,b))` pour engendrer une matrice $a \times b$ de gaussiennes centrées de variance 1, et la commande `w, v = np.linalg.eigh(x)` qui met dans w les valeurs propres de la matrice symétrique x.
   — Tracer la fonction de répartition empirique de $\lambda_1, \ldots, \lambda_N$ obtenu par `spectre(N)` pour $N = 100$, et superposer lui la fonction de répartition de la loi du demi-cercle. Qu'observez-vous ? Wigner a prouvé ce phénomène dans les années 50, et obtenu le prix Nobel de physique en 1963.