DM4 : Markov travaille à la chaîne !

Préliminaires sur les matrices

1. Soient $A \in \mathscr{M}_{n,p}(\mathbb{K})$ et $B \in \mathscr{M}_{p,q}(\mathbb{K})$, démontrer que $(AB)^\top = B^\top A^\top$.
2. Soit $A \in \mathscr{M}_n(\mathbb{K})$. Déduire de la question précédente que si A est inversible, alors $A^\top$ est inversible.

En utilisant ce résultat à $A^\top$, on obtient que si $A^\top$ est inversible, alors $(A^\top)^\top = A$ est inversible. On peut en conclure que A est inversible ssi $A^\top$ est inversible.

3. Soit $A \in \mathscr{M}_n(\mathbb{K})$, utiliser ce qui précède, pour montrer que $\mathrm{Sp}(A) = \mathrm{Sp}(A^\top)$.
4. On note F l'ensemble des matrices M de $\mathscr{M}_3(\mathbb{R})$ tel que $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ soit vecteur propre de M. Démontrer que F est un sous-espace vectoriel de $\mathscr{M}_3(\mathbb{R})$.
5. (question optionnelle) Déterminer une base et la dimension de F.

Préliminaires Python

6. Soit (a, b, c) trois nombres positifs vérifiant que $a+b+c = 1$. Écrire une fonction Python `SimulVA(a,b,c)` qui renvoie 1 avec probabilité a, 0 avec probabilité b et -1 avec probabilité c. Pour cela, on tirera un nombre au hasard dans $[\,0\,;1\,]$ et on remarque que $[\,0\,;1\,]$ est égal à l'union disjointe de $[\,0\,;a\,[$, de $[\,a\,;a+b\,[$ et de $[\,a+b\,;1\,]$, et que la longueur de chaque intervalle vaut respectivement a, b et c.
7. On considère deux matrices A et B modélisées en Python comme des listes de listes ou comme des array de numpy, écrire une fonction `Produit(A,B)` qui renvoie le produit AB (sous l'hypothèse que ce produit est possible).
8. Écrire une fonction `Identité(n)` qui renvoie la matrice I_n.
On n'utilisera pas une fonction ad-hoc comme `np.eye`.
9. Écrire une fonction `Puissance(A,p)` qui, à une matrice $A \in \mathscr{M}_n(\mathbb{R})$ et $p \in \mathbb{N}$, renvoie A^p de façon récursive en utilisant la relation $A^p = A^{p-1} \times A$.
10. Écrire `PuissanceBis(A,p)` qui renvoie A^p de façon récursive en utilisant le fait que si p est pair, alors il existe $q \in \mathbb{N}$ tel que $p = 2q$ et $A^p = (A^q) \times (A^q)$. et si p est impair, alors il existe $q \in \mathbb{N}$ tel que $p = 2q + 1$ et $A^p = A^q \times A^q \times A$.
11. Quel est l'intérêt de la fonction `PuissanceBis` par rapport à la fonction `Puissance` ? Compter le nombre de multiplications nécessaires pour calculer A^{64} dans les deux fonctions.
12. En important la librairie `time`, on peut mesurer le temps de calcul avec les commandes suivantes :

```python
tic = time.time()
# Un calcul
tac = time.time()
temps = tac - tic # temps mis pour le calcul
```

Écrire un script représentant le temps nécessaire pour calculer B^n (la matrice B est définie plus loin) en fonction de $n \in [\![0\,;950]\!]$ [1]. On représentera le temps mis pour les deux fonctions, le graphe obtenu est-il cohérent avec votre réponse à la question 11 ?

Une équipe qui a du temps pour jouer !

Tous les jours, l'équipe de 2Bio2 joue un match d'ultimate (ils et elles n'ont que ça à faire car leur prof de maths leur donne peu de travail). Au jour $n \in \mathbb{N}$, on note X_n le résultat de l'équipe, X_n vaut 1 en cas de victoire, 0 en cas de match nul et -1 en cas de défaite. Le premier jour, l'équipe gagne et donc $X_0 = 1$. Le moral jouant beaucoup la probabilité de gagner/perdre/match nul au jour $n+1$ dépend du résultat de la veille (le jour n) :

- Si $X_n = 1$, alors l'équipe a un moral au beau fixe et la probabilité que l'équipe gagne au jour $n+1$ vaut $7/9$, la probabilité de perdre ou de faire match nul valent toutes les deux $1/9$.

1. On rappelle que Python, par défaut, à une limite de 1000 appels récursifs. Cependant, il est possible de changer cette valeur.

- Si $X_n = 0$, alors le moral est neutre et les probabilités de gagner/perdre/match nul au jour $n+1$ valent toutes les trois $1/3$.
- Si $X_n = -1$, alors le moral est un peu bas, mais il peut avoir une volonté de revanche, ainsi la probabilité de gagner au jour $n+1$ est de $1/3$ mais la probabilité de perdre est de $2/3$ (le cas où l'équipe se démotive).

13. Donner la loi X_1.
14. Donner la loi du couple (X_1, X_2), on pourra donner le résultat sous forme d'un tableau.
15. En déduire la loi de X_2.
16. Donner l'espérance et la variance de X_0, X_1 et X_2.
17. Calculer la covariance de X_1 et de X_2, que peut-on en déduire sur l'indépendance de X_1 et X_2 ?
18. Votre professeur de maths va voir exactement un match au hasard et uniformément parmi les trois premiers matchs (ceux numéros 0, 1 et 2). Quelle est la probabilité qu'il assiste à une victoire ?
19. Sachant que l'équipe a gagné au jour 2, quel était le résultat le plus probable au jour 1 ?

Gagner le plus possible avec des simulations (pas la plus benne manière de gagner !)

20. Écrire une fonction `Transition(Xn)` qui simule X_{n+1} en fonction de X_n c'est-à-dire qui va renvoyer aléatoirement -1, 0 ou 1 mais cet aléa dépend de la valeur de X_n.
21. Écrire une fonction `SimulerXn(n)` qui simule X_0, X_1 jusqu'à X_n sous forme d'une liste.
22. Soit $n \in \mathbb{N}^*$. Calculer la probabilité que la classe gagne entre les jours 0 jusqu'à $n-1$ mais ne gagne pas au jour n (défaite ou match nul).
23. Démontrer que la probabilité que la classe gagne tous les jours $n \in \mathbb{N}$ est nul (dommage !).
24. On note D la variable aléatoire indiquant le premier jour de la défaite ou du premier match nul (cette variable est bien définie d'après la question précédente), déterminer la loi de D ainsi que $\mathbb{E}(D)$.
25. Écrire une fonction qui simule la variable aléatoire D.
26. On souhaite vérifier expérimentalement la valeur de $\mathbb{E}(D)$: soit $D_1, \ldots, D_n$ n variables aléatoires indépendantes et suivant la même loi que D. On pose $S = \dfrac{1}{n}\sum_{k=1}^{n} D_k$, à l'aide de l'inégalité de Bienaymé-Tchebychef, déterminer $n \in \mathbb{N}^*$ tel qu'on soit sûr avec une probabilité supérieur à 99% que S est une approximation de $\mathbb{E}(D)$ à 0.01 près.
27. Pour ce n-là, simuler S, le résultat est-il cohérent ?

Relation de récurrence entre la loi de X_n et celle de X_{n+1}

28. Soit $n \in \mathbb{N}$, démontrer que :

$$\mathbb{P}(X_{n+1} = 1) = \frac{7}{9}\mathbb{P}(X_n = 1) + \frac{1}{9}\mathbb{P}(X_n = 0) + \frac{1}{9}\mathbb{P}(X_n = -1)$$

29. De même, donner, sans preuve, une expression de $\mathbb{P}(X_{n+1} = 0)$ et une expression de $\mathbb{P}(X_{n+1} = -1)$ en fonction de $\mathbb{P}(X_n = 1)$, $\mathbb{P}(X_n = 0)$ et $\mathbb{P}(X_n = -1)$.
30. On pose $U_n = \begin{pmatrix} \mathbb{P}(X_n = 1) \\ \mathbb{P}(X_n = 0) \\ \mathbb{P}(X_n = -1) \end{pmatrix} \in \mathscr{M}_{3,1}(\mathbb{R})$, déterminer une matrice $A \in \mathscr{M}_3(\mathbb{R})$ tel que pour tout $n \in \mathbb{N}$, $U_{n+1} = AU_n$.
31. Déterminer une expression de U_n en fonction seulement de A.
32. Utiliser les fonctions Python écrites, pour calculer $\mathbb{P}(X_{100} = 1)$.
33. En écrivant un script permettant de simuler 1000 tirages, donner une approximation de $\mathbb{P}(X_{100} = 1)$, le résultat de cette simulation est-il cohérent avec le résultat de la question 32.

Un calcul de puissance de A

Comme, on l'a vu à la question précédente, le calcul de A^n permet le calcul de la loi de X_n. On va donc calculer A^n. Pour éviter les fractions, on pose $B = \begin{pmatrix} 7 & 3 & 3 \\ 1 & 3 & 0 \\ 1 & 3 & 6 \end{pmatrix}$

34. Exprimer A en fonction de B puis A^n en fonction de B^n.
35. Démontrer que $B^\top \in F$ (F a été défini à la question 4). En déduire une valeur propre de B.
36. Calculer B^2 et B^3 à l'aide des fonctions Python écrites avant.
37. Montrer que $B^3 \in \text{vect}(I_n, B, B^2)$, en déterminant $(\alpha, \beta, \gamma) \in \mathbb{R}^3$ tel que $B^3 = \alpha I_n + \beta B + \gamma B^2$.
Indication : on pourra trouver γ en regardant le coefficient à la deuxième ligne troisième colonne.
38. En déduire que B est inversible et que $B^{-1} \in \text{vect}(I_n, B, B^2)$.
39. Soit λ une valeur propre de B, démontrer que λ est racine de $Q = X^3 - 16X^2 + 73X - 108$.
40. Factoriser Q. On notera les trois racines de Q : λ_1, λ_2, λ_3 avec $\lambda_1 > \lambda_2 > \lambda_3$.
Indication : on rappelle que l'on connaît déjà une valeur propre de B.
41. Déterminer, pour tout $i \in [\![1\,;3]\!]$, une base de $\text{Ker}(B - \lambda_i I_3)$. En conclure que B est diagonalisable, et écrire B sous la forme $B = PDP^{-1}$ avec D une matrice diagonale contenant dans cette ordre $\lambda_1, \lambda_2, \lambda_3$ et P une matrice inversible dont la seconde ligne n'a que des 1.
Il est conseillé au brouillon de vérifier la cohérence des résultats en calculant BP et PD.
42. Retrouver à nouveau les valeurs propres de B en calculant le rang de $B - \lambda I_3$ en fonction de $\lambda \in \mathbb{K}$.
43. Exprimer B^n en fonction des matrices P et D et P^{-1}, on démontrera proprement le résultat.
44. Calculer P^{-1}. *On vérifiera la cohérence des résultats au brouillon en effectuant PP^{-1}.*
45. Calculer explicitement B^n sous la forme $B^n = 9^n C + 4^n E + 3^n F$ avec $(C, E, F) \in \mathscr{M}_3(\mathbb{R})^3$.
Il est conseillé de décomposer D^n dans la base canonique, d'écrire P^{-1} sous la forme un scalaire multiplié par une matrice à coefficients entiers, puis de calculer les produits de matrices (à coefficients entiers seulement) à l'aide de la fonction python `Produit`. Il est aussi conseillé de vérifier la cohérence des résultats pour $n = 0$, $n = 1$ et $n = 2$.
46. En déduire une valeur de A^n.

Application aux probabilités de victoires !

47. Déduire de la question précédente la loi de X_n.
48. Déterminer $\lim\limits_{n\to+\infty} \mathbb{P}(X_n = 1)$, $\lim\limits_{n\to+\infty} \mathbb{P}(X_n = 0)$ et $\lim\limits_{n\to+\infty} \mathbb{P}(X_n = -1)$.
49. Déterminer l'espérance de X_n et la variance de X_n.
Il est conseillé de vérifier les résultats avec ceux de la question 16.
50. On note $S_n = \dfrac{1}{n+1} \sum\limits_{k=0}^{n} X_k$, S_n représente le score moyen par match (1 point pour la victoire, 0 pour le match nul et -1 pour la défaite). Calculer $\mathbb{E}(S_n)$ puis en déduire un équivalent.