

Colles de Mathématiques en E2A

Graphes et chaînes de Markov

Semaine 21 : 4 - 8 mars

Toutes les définitions et tous les énoncés de théorèmes/propositions du cours sont exigibles des élèves. Les démonstrations des théorèmes du cours ne sont pas exigibles, sauf si elles apparaissent en question de cours.

On pourra à tout moment demander à un·e élève de donner la nature (réel, suite, fonction, ensemble, proposition, etc) d'une expression manipulée dans un exercice, pour vérifier sa bonne compréhension. On pourra aussi demander de préciser quelles sont les variables libres et quelles sont les variables liées (muettes).

On portera une attention toute particulière à ce que les objets soient correctement introduits avant d'être utilisés, et ne soient pas introduits pour rien.

1 Chapitre XV : Graphes et chaînes de Markov

1.1 Définitions

- Graphe (fini), sommets (ou noeuds), arêtes.
- Notion de graphe orienté ou non-orienté.
- Voisin d'un sommet, voisinage d'un sommet, degré d'un sommet.
- Matrice d'adjacence d'un graphe.
- Liste d'adjacence d'un sommet, liste des listes d'adjacence.
- Chemin de longueur k allant d'un sommet à un autre. Existence d'un chemin allant d'un sommet à un autre. Distance entre deux sommets.
- Notion de graphe connexe.
- Notion de graphe pondéré. Poids associé à une arête.
- Notion de graphe probabiliste.
- Chaîne de Markov.
- Probabilités de transition (ou probabilités transitionnelles). Chaîne de Markov homogène.
- Le n^{e} état probabiliste de la chaîne de Markov est une **matrice ligne**. Etat initial.
- Matrice de transition associée à une chaîne de Markov. Graphe probabiliste associé à une matrice de transition.
- Etat stable (ou loi stationnaire).

1.2 Résultats

- Un graphe G est non orienté ssi sa matrice d'adjacence M est symétrique.
- Si on note M la matrice d'adjacence du graphe G (dont les sommets sont s_1, \dots, s_n), alors $[M^k]_{i,j}$ est le nombre de chemins de longueur k allant de s_i à s_j .
- G est connexe si et seulement si tous les coefficients de la matrice $\sum_{k=0}^{n-1} M^k$ sont strictement positifs.

- L'algorithme de Dijkstra permet de trouver le plus court chemin entre deux sommets dans un graphe pondéré.
- Soit M une matrice de transition d'une chaîne de Markov. Alors les coefficients de la matrice M sont tous positifs ou nuls et la somme des coefficients de chaque ligne est égale à 1.
- Caractérisation des états stables d'une chaîne de Markov (où M est la matrice de transition)

$$\pi \text{ est un état stable de la chaîne de Markov} \iff \begin{cases} \pi \text{ est un vecteur propre de } {}^tM \text{ associé à la valeur propre } 1 \\ \pi \text{ définit une loi de probabilité} \end{cases}$$

$$\iff \begin{cases} \pi \text{ est un vecteur propre de } {}^tM \text{ associé à la valeur propre } 1 \\ \forall j \in \llbracket 1, r \rrbracket, \pi_j \geq 0 \\ \sum_{j=1}^r \pi_j = 1 \end{cases}$$

- Soit (X_n) une chaîne de Markov homogène à r états notés $1, 2, \dots, r$. Si la suite (X_n) converge en loi, alors la loi limite est nécessairement une loi stationnaire de la chaîne de Markov. Autrement dit, si, pour tout $j \in \llbracket 1, r \rrbracket$, $\pi_j = \lim_{n \rightarrow +\infty} \mathbb{P}([X_n = j])$ existe, alors le vecteur $\pi = (\pi_1, \dots, \pi_r)$ est un état stable de la chaîne de Markov.

1.3 Méthodes

Il faut savoir redémontrer le théorème suivant, qui est une question classique des concours.

Théorème 1.1. Soit (X_n) une chaîne de Markov homogène à r états notés $1, 2, \dots, r$. On note $M = (p_{i,j})_{\substack{1 \leq i \leq r \\ 1 \leq j \leq r}}$ sa matrice de transition. Alors, pour tout $n \in \mathbb{N}$,

$$V_n = V_0 M^n$$

La preuve se fait en deux étapes.

1. En utilisant la formule des probabilités totales, on montre la formule de récurrence : $V_{n+1} = V_n M$.
2. On démontre ensuite que $V_n = V_0 M^n$ par récurrence. Même si la récurrence est très simple, il faut la détailler si c'est le coeur de la question.

2 Questions de cours

1. Ecrire une fonction **Python**, nommée `List_deg`, qui :
 - prend en argument la liste `L` des listes d'adjacence d'un graphe
 - renvoie la liste des degrés des sommets de ce graphe

On expliquera rapidement à l'oral les étapes clés de l'algorithme.

2. Compléter la fonction **Python** suivante qui
 - prend en argument la liste `L` des listes d'adjacence d'un graphe
 - renvoie la matrice d'adjacence de ce graphe

On expliquera à l'oral chaque ligne complétée et à quoi servent les variables créées.

```

1 import numpy as np
2 def Ad_matrix_from_list(L):
3     nbS = _____
4     M = np.zeros([nbS,nbS])
5     for i in range(nbS):
6         nbV = _____
7         for j in range(nbV):
8             M[i,L[i][j]] = _____
9     return M

```

3. Ecrire une fonction **Python**, nommée `graph_connexe`, qui
- prend en argument la matrice d'adjacence `M` d'un graphe
 - renvoie le booléen `True` si ce graphe est connexe et `False` sinon

On expliquera rapidement à l'oral les étapes clés de l'algorithme et les fonctions **Python** utilisées.