

Programme de colle n°11

Semaine du 16/12

Réduction des matrices

Application aux chaînes de Markov

Pour cette colle, ce qui tient lieu d'« exercice étoilé » est de savoir, à partir d'une situation concrète, tracer un graphe probabiliste, donner la matrice de transition associée et déterminer un état stable

Reprise du programme de réduction des matrices.

Chaînes de Markov

- Rappels d'ECG1 sur les graphes : sommets, arêtes, graphes orientés, graphes pondérés.
- Matrice d'adjacence d'un graphe (orienté ou non).
- Graphe probabiliste : c'est un graphe
 - orienté et pondéré ;
 - pour tous $(i, j) \in \llbracket 1, r \rrbracket^2$, on a au plus une arête $i \rightarrow j$;
 - pour tout $i \in \llbracket 1, r \rrbracket$, la somme des poids des arêtes sortant du sommet i est 1.

On notera r le nombre de sommets du graphe probabiliste considéré.

- Matrice de transition d'un graphe probabiliste.
- Définition : matrice stochastique. Une matrice est stochastique ssi c'est la matrice de transition d'un graphe probabiliste. Si M est stochastique, $1 \in \text{Sp}(M)$ et $\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in E_1(M)$.
- Chaîne de Markov associée à un graphe probabiliste de matrice de transition M : c'est une suite de variables aléatoires $(X_n)_{n \in \mathbb{N}}$, à valeurs dans $\llbracket 1, r \rrbracket$, telles que

$$\forall (i, j) \in \llbracket 1, r \rrbracket^2, \mathbb{P}_{(X_n=i)}(X_{n+1} = j) = m_{i,j}$$

où $m_{i,j}$ est le coefficient (i, j) de M (c'est donc le poids de l'arête $i \rightarrow j$).

Interprétation : position au temps n d'un système sans mémoire « se déplaçant aléatoirement » sur le graphe en temps discret.

- On range la loi de X_n dans un vecteur de $\mathcal{M}_{1,r}^r$:

$$V_n = (\mathbb{P}(X_n = 1) \quad \dots \quad \mathbb{P}(X_n = r))$$

On a alors la relation $V_{n+1} = V_n M$. V_n est appelé n -ième état probabiliste de la chaîne.

- $\forall n \in \mathbb{N}, V_n = V_0 M^n$.
- État probabiliste stable : c'est un état probabiliste $V = (p_1 \quad \dots \quad p_r)$ (les p_i sont donc positifs et de somme 1) tel que $VM = V$.

Un état $V = (p_1 \quad \dots \quad p_r)$ est stable ssi $\begin{pmatrix} p_1 \\ \vdots \\ p_r \end{pmatrix}$ est vecteur propre de ${}^t M$ pour la valeur propre 1.

Python

- Simulation d'expériences aléatoires : pas de commandes spécifiques aux chaînes de Markov mais on peut demander de simuler des expériences aléatoires simples avec `rd.random`, `rd.randint` par exemple.
- Algèbre linéaire : on doit savoir utiliser les outils de `numpy.linalg`, notamment sur la matrice de transition. Ci-dessous les principales fonctions qui peuvent être utilisées :
 - `al.eig` : recherche d'éléments propres.
 - `al.rank` : rang d'une matrice.
 - `al.matrix_power` : mise à la puissance n d'une matrice.
 - `np.dot` : produit matriciel.
 - `np.eye` : matrice identité.
 - `np.transpose` : transposition