Université Paris Diderot Année 2018-2019
L1 Info MI2

Quelques applications de l'algèbre linéaire

Juan Pablo Vigneaux

Exemple 1 : Page Rank. PageRank est un algorithme inventé par Larry Page, cofondateur de Google, en 1998. Il donne une mesure quantitative de la popularité d'une page web et contribue ainsi au système de classement de sites web utilisé par le moteur de recherche Google. L'algorithme se base sur le principe suivant :

PageRank fonctionne en comptant le nombre *et la* qualité *des liens vers une page pour déterminer une estimation approximative de l'importance du site web. L'hypothèse sous-jacente est que les sites web les plus importants sont susceptibles de recevoir davantage de liens d'autres sites web.*

La connectivité entre les sites web est représentée par un graphe orienté. Un sommet correspond à un site web et une flèche du sommet A au sommet B indique que A contient un hyperlien vers B. Nos considérons ici un exemple très simple :

$$\begin{tikzcd}
1 \arrow[r] \arrow[d, shift left] \arrow[dr, shift left] & 2 \arrow[d] \\
3 \arrow[u, shift left] \arrow[r, shift left] & 4 \arrow[ul, shift left] \arrow[l, shift left]
\end{tikzcd}$$

Soit $L(p)$ les nombre de flèches qui sortent d'un sommet p. Dans l'exemple $L(1) = 3$, $L(2) = 1$ et $L(3) = L(4) = 2$.

Soit N le nombre de sommets. On définit une matrice $A = (a_{ij})$ de taille $N \times N$ par la formule

$$a_{ij} = \begin{cases} 1/L(j) & \text{s'il existe une flèche de } j \text{ vers } i, \\ 0 & \text{sinon.} \end{cases}$$

(i) Déterminer la matrice A dans l'exemple.

L'algorithme commence par assigner à chaque site web la même importance. Par exemple, $1/N$. On représente cela par un vecteur $\mathbf{x}^0 = (1/N, ..., 1/N)$.

Si $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{R}^n$ est un vecteur qui vérifie $x_1 + ... + x_n = 1$, on l'appellera une *distribution de probabilité.* Le vecteur de popularité $\mathbf{x}^0$ est une distribution de probabilité. En fait, chaque composante $\mathbf{x}_i^0$ peut s'interpréter comme la probabilité de tomber sur le site i au hasard, si l'on ne fait pas de différence entre eux (pas très réaliste pour le moment).

(ii) Montrer que les colonnes de la matrice A sont des distributions de probabilité.

(iii) Montrer que si $\mathbf{x}$ est une distribution de probabilité, alors $A\mathbf{x}$ aussi.

On peut alors imaginer que chaque site répartit sa popularité/importance en parties égales entre ses voisins et que l'on obtient un nouveau vecteur de popularité $\mathbf{x}^1 := A\mathbf{x}^0$. La répétition de cette procédure donne une suite $\mathbf{x}^{n+1} := A\mathbf{x}^n$, pour $n \geq 0$. La composante $\mathbf{x}_i^{n+1}$ peut s'interpréter comme la probabilité de visiter le site i à l'instant $n+1$, après avoir suivie un hyperlien quelconque (tous avec la même probabilité), si au temps n la probabilité d'être sur chaque site j était $\mathbf{x}_j^n$.

On peut itérer cette procédure, jusqu'à attendre un certain équilibre (i.e. $\mathbf{x}^{i+1} \approx A\mathbf{x}^n$), qu'on appelle l'état stationnaire.

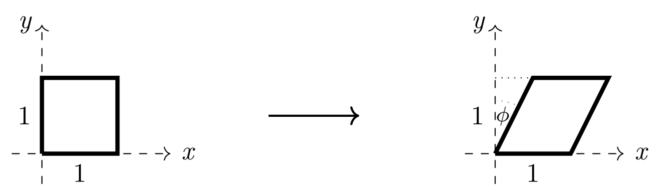
(iv) Un point fixe de l'algorithme est une distribution de probabilité $\mathbf{y}$ qui satisfait $\mathbf{y} = A\mathbf{y}$. Calculer le point fixe dans l'exemple.

On peut montrer que $\mathbf{x}^n \to \mathbf{y}$ quand $n \to \infty$. Le vecteur $\mathbf{y}$ donne le classement de sites qu'on cherchait.

Exemple 2 : Graphisme informatique. Une image peut être vue comme un sous-ensemble de $\mathbb{R}^2$. Les endomorphismes de $\mathbb{R}^2$ correspondent alors aux transformations possibles des images.

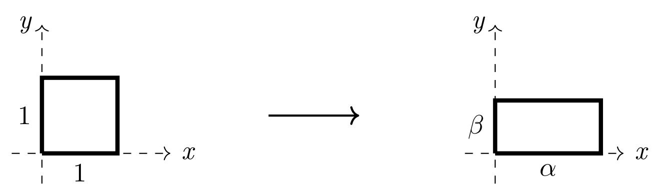
(i) Soit $R_\theta : \mathbb{R}^2 \to \mathbb{R}^2$ la rotation d'angle θ dans le sens antihoraire. Sachant que $\mathbb{R}_\theta$ est une application linéaire, trouver la matrice qui représente cette application.

(ii) Soit $T_\phi : \mathbb{R}^2 \to \mathbb{R}^2$ l'application linéaire qui réalise la transformation suivante :



Trouver la matrice qui représente cette fonction.

(iii) Soit $D_{\alpha,\beta} : \mathbb{R}_2 \to \mathbb{R}^2$ l'application linéaire qui réalise la transformation suivante :



Trouver la matrice qui représente cette fonction.

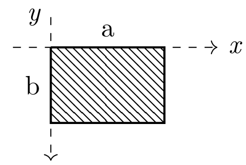
Néanmoins, les translations non-triviales ne sont pas linéaires (pourquoi ?). Elles peuvent être représentées par une matrice si l'on "élargit" l'espace et on représente donc une position (x, y) par un vecteur $(x, y, 1)$. La translation par (a, b) correspond alors à

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ 1 \end{bmatrix}.$$

(On parle d'une *transformation affine*.)

(iv) Quelles matrices de taille 3×3 représentent les rotations et déformations des questions précédentes ?

(v) Trouver la matrice de taille 3×3 qui représente la rotation d'angle θ de la figure suivante autour de son centre.



Exemple 3 : Codes correcteurs d'erreurs. On dénote par $\mathbb{F}_2$ l'ensemble $\{0, 1\}$ muni de deux opérations

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Donc, la somme correspond à l'opérateur logique XOR ("ou exclusif") et la multiplication est pareille à celle de $\mathbb{N}$. Notez que $-1 = 1$.

Normalement on ne parle pas de ce corps fini dans les cours introductifs d'algèbre. C'est dommage, parce que dans un ordinateur tout est codifié comme une suite de 1s et 0s (physiquement, les deux états possibles d'un transistor) et l'arithmétique sur $\mathbb{F}_2$ a naturellement beaucoup d'applications en informatique.

On peut définir un "espace vectoriel sur $\mathbb{F}_2$" comme un ensemble muni d'une somme et d'une multiplication externe par les éléments de $\mathbb{F}_2$ (qui est forcément trivial !). Notamment, $\mathbb{F}_2^n$ est un espace vectoriel—la somme se fait composante par composante, comme d'habitude—et les vecteurs de cet espace sont *de mots binaires de longueur n*.

Dans les applications, on veut envoyer des mots binaires (un film, une chanson, un courriel...) sur un canal bruyant (le réseaux 4G, un DVD...). L'idée de base c'est alors d'ajouter de la redondance pour corriger les erreurs. Par exemple, on pourrait envoyer le message trois fois et après comparer ce qu'on a reçu à chaque fois... mais bien sûr cela devient très inefficace.

On étudie ici un système de codage introduit par Richard Hamming en 1950. Supposons que l'on veut envoyer des mots de longueur 4. On va associer à chaque mot de longueur 4 un autre mot de longueur 7 par une application linéaire $f : \mathbb{F}_2^4 \to \mathbb{F}_2^7$, puis transmettre ce mot de longueur 7 sur le canal bruyant. De l'autre coté du canal, on va recevoir un mot possiblement altéré sur certaines positions.

On pose $\mathcal{C} := \operatorname{im} f$. Chaque vecteur appartenant à $\mathcal{C}$ s'appelle un mot-codé. Il existe toujours une matrice H tel que $\mathcal{C} = \ker H$.

Supposons maintenant qu'un mot-codé $\mathbf{c}$ est envoyé et un autre mot $\tilde{\mathbf{c}}$ est reçu à l'autre coté ($\tilde{\mathbf{c}}$ n'est pas nécessairement un mot-codé). Écrivons,

$$\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{e}.$$

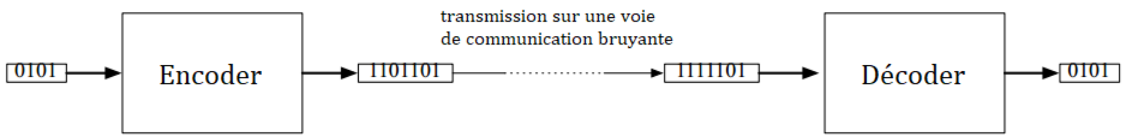


FIGURE 1 – Schéma de transmission.

Évidement,

$$H\tilde{\mathbf{c}} = H\mathbf{e}.$$

Donc, si le récepteur calcule $H\tilde{\mathbf{c}}$ et obtient un résultat différent de zéro, il peut être sûr de la présence d'une erreur.

Dans le code de Hamming, la matrice H est

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

(i) Supposons que le canal introduit au plus un bit d'erreur. Soit $\tilde{\mathbf{c}} = (1, 1, 1, 0, 0, 1, 0)$. Quelle est la valeur de $H\tilde{\mathbf{c}}$? et de $\mathbf{e}$? Déterminer $\mathbf{c}$.

(ii) Montrer que tout vecteur $\mathbf{e}$ qui contient *exactement* deux 1s satisfait $H\mathbf{e} \neq 0$, mais aussi que l'on ne peut pas *corriger* ce type d'erreurs avec le code de Hamming. Hint : trouver $\mathbf{e}_1$ et $\mathbf{e}_2$ tels que $H\mathbf{e}_1 = H\mathbf{e}_2$.

(iii) Trouver une base de $\ker H$. En déduire une matrice qui représente une function f tel que $\operatorname{im} f = \ker H = \mathcal{C}$.