

chaines_Markov

April 2, 2020

1 Chaînes de Markov : exemple d'une épidémie

```
[1]: import numpy as np # bibliothèques chargées : numpy

import numpy.linalg as alg # utile en algèbre linéaire

import matplotlib.pyplot as plt # pour les graphiques
```

1.1 Le problème à étudier

Evolution d'une pandémie.

Vecteur état mélangé le mois n : $X(n) = (x_1(n), x_2(n), x_3(n)) \in \mathbb{R}^3$.

$x_1(n)$ proportion population malade, $x_2(n)$ proportion immunisée, $x_3(n)$ proportion saine mais non immunisée.

Relation de récurrence linéaire $X(n+1) = MX(n)$ avec la matrice de transition

$$M = \begin{pmatrix} 0.2 & 0 & 0.5 \\ 0.8 & 0.9 & 0 \\ 0 & 0.1 & 0.5 \end{pmatrix}.$$

1.1.1 Initialisation des matrices en Python

```
[21]: M=np.array([[0.2,0,0.5],[0.8,0.9,0],[0,0.1,0.5]])
M
```

```
[21]: array([[0.2, 0. , 0.5],
          [0.8, 0.9, 0. ],
          [0. , 0.1, 0.5]])
```

Exemple avec $X(0) = (0, 0, 1)$. 100% sains mais non immunisés le mois $n = 0$.

```
[13]: X0=np.array([0,0,1])
X1= M.dot(X0)
X1
```

```
[13]: array([0.5, 0. , 0.5])
```

1.1.2 On effectue une boucle pour calculer l'évolution de l'épidémie.

On définit une fonction qui renvoie la suite des $X(n)$ (on dit l'orbite ou la trajectoire de $X(0)$).

```
[5]: def orbite(M,V,n): # V est la donnée initiale et n le nombre de vecteurs
    ↪ dessinés.
    X=V # on va emplir les résultats dans X
    for i in np.arange(1,n):
        V =M.dot(V) # la récurrence
        X=np.vstack((X,V))# on empile verticalement
    return X
```

```
[9]: X0= np.array([0,0,1])
X0
```

```
[9]: array([0, 0, 1])
```

```
[7]: orbite(M,X0,10)
```

```
[7]: array([[0.          , 0.          , 1.          ],
          [0.5         , 0.          , 0.5         ],
          [0.35        , 0.4         , 0.25        ],
          [0.195       , 0.64        , 0.165       ],
          [0.1215      , 0.732       , 0.1465      ],
          [0.09755     , 0.756       , 0.14645     ],
          [0.092735    , 0.75844    , 0.148825    ],
          [0.0929595   , 0.756784   , 0.1502565   ],
          [0.09372015  , 0.7554732  , 0.15080665  ],
          [0.09414736  , 0.754902   , 0.15095064  ]])
```

Calcul des puissances de M .

```
[22]: N=M
print(N)
for i in np.arange(10):
    N=M.dot(N)
    print(N)
```

```
[[0.2 0. 0.5]
 [0.8 0.9 0. ]
 [0. 0.1 0.5]]
[[0.04 0.05 0.35]
 [0.88 0.81 0.4 ]
 [0.08 0.14 0.25]]
[[0.048 0.08 0.195]
 [0.824 0.769 0.64 ]
 [0.128 0.151 0.165]]
[[0.0736 0.0915 0.1215]
 [0.78 0.7561 0.732 ]
```

```

[0.1464 0.1524 0.1465]]
[[0.08792 0.0945 0.09755]
 [0.76088 0.75369 0.756 ]
 [0.1512 0.15181 0.14645]]
[[0.093184 0.094805 0.092735]
 [0.755128 0.753921 0.75844 ]
 [0.151688 0.151274 0.148825]]
[[0.0944808 0.094598 0.0929595]
 [0.7541624 0.7543729 0.756784 ]
 [0.1513568 0.1510291 0.1502565]]
[[0.09457456 0.09443415 0.09372015]
 [0.7543308 0.75461401 0.7554732 ]
 [0.15109464 0.15095184 0.15080665]]
[[0.09446223 0.09436275 0.09414736]
 [0.75455737 0.75469993 0.754902 ]
 [0.1509804 0.15093732 0.15095064]]
[[0.09438265 0.09434121 0.09430479]
 [0.75467142 0.75472014 0.75472968]
 [0.15094594 0.15093865 0.15096552]]
[[0.0943495 0.09433757 0.09434372]
 [0.75471039 0.75472109 0.75470055]
 [0.15094011 0.15094134 0.15095573]]

```

1.1.3 Calcul de l'état mélangé stationnaire : distribution limite de l'épidémie.

Les effectifs limites doivent satisfaire $MX = X$ c'est-à-dire $(Id - M)X = 0$.

```
[23]: N= np.eye(3) - M # np.eye(2) fabrique la matrice identité de taille 2x2
N
```

```
[23]: array([[ 0.8,  0. , -0.5],
          [-0.8,  0.1,  0. ],
          [ 0. , -0.1,  0.5]])
```

Attention, la matrice $N = Id - M$ n'est pas inversible!

```
[24]: alg.matrix_rank(N) # on calcule son rang.
```

```
[24]: 2
```

On cherche X dans le noyau de N .

```
[25]: from scipy.linalg import null_space
```

```
[27]: Y=null_space(N)
Y
```

```
[27]: array([[0.12166206],
           [0.97329651],
           [0.1946593 ]])
```

Y est un vecteur unitaire (de norme euclidienne 1).

On normalise Y pour avoir un état mélangé (somme des coordonnées = 1).

```
[29]: X=Y/np.sum(Y)
      X
```

```
[29]: array([[0.09433962],
           [0.75471698],
           [0.1509434 ]])
```