Cours du 20/04/2020.

Application 3 : Évolution de systèmes.

Félix le chat confiné.

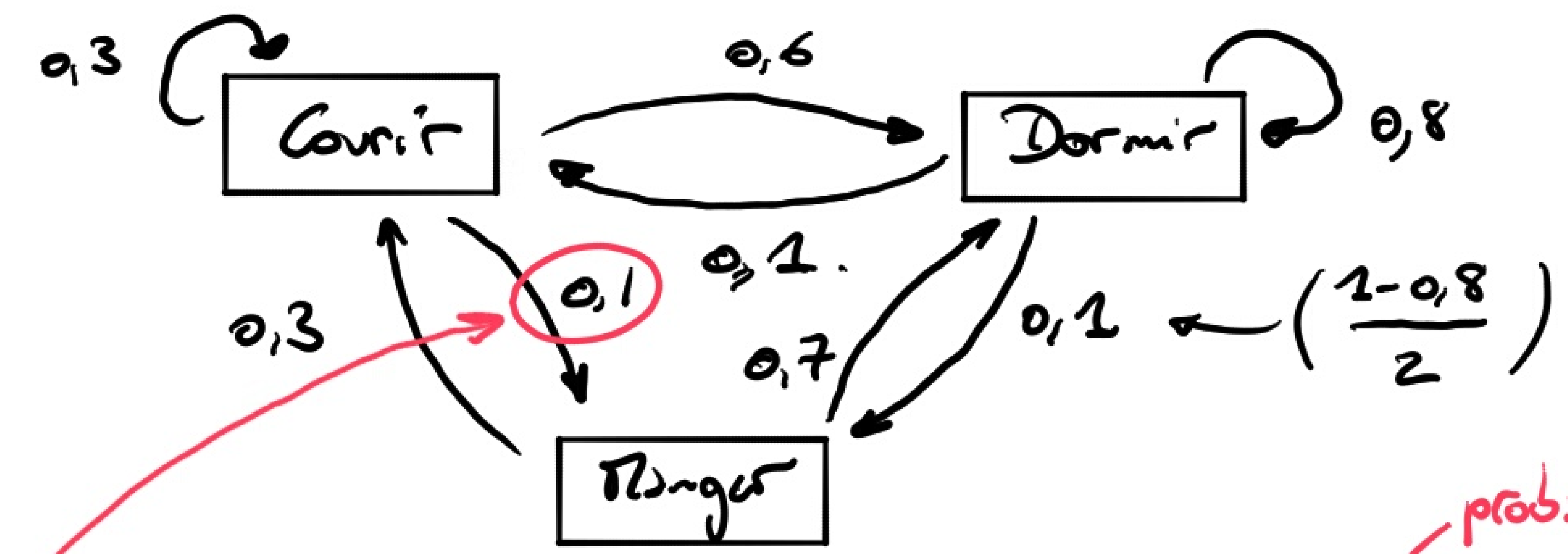
Félix : dort / court / mange. Toutes les 5 minutes, Félix change d'activité (ou pas).

Le processus est sans mémoire (comme Félix !) i.e le comportement à l'instant T (1 instant = 5 minutes) ne dépend que du comportement à l'instant $T-1$ (et pas des comportements aux instants $t < T-1$).

Déf. Un tel processus est appelé processus de Markov (alors qu'un tel chat est appelé un chat).

Le processus suit les règles suivantes :

- Félix dort : Félix a 8/10 chances de continuer à dormir et s'il se réveille il a 1/2 chance d'aller manger et 1/2 de se mettre à courir.
- Les repas de Félix durent 5 minutes.
- Après manger : Félix a 7/10 chances d'aller dormir, 3/10 chances de partir en courant.
- Après 5 min de course : 6/10 d'aller dormir, 3/10 de continuer à courir, 1/10 d'aller manger.



quand Félix ... dort mange court

proba qu'il aille

$$M = \begin{array}{c} \\ \\ \end{array} \begin{pmatrix} 0,8 & 0,7 & 0,6 \\ 0,1 & 0 & 0,1 \\ 0,1 & 0,3 & 0,3 \end{pmatrix} \begin{array}{l} \text{dormir} \\ \text{manger} \\ \text{courir.} \end{array}$$

1 chance /10 qu'il aille manger après course i.e proba (course $\to$ manger)
E_3 E_2

Déf M est la matrice de transition du processus.

$M = (a_{ij})_{i,j}$, a_{ij} = proba de passer à l'état E_i depuis l'état E_j.

Rmq. M matrice de transition d'un processus de Markov : la somme des éléments de chaque colonne vaut 1. En effet : "Il y a 100% d'être dans un des états E_i", ou "la liste des états est complète".

Ici, Félix ne fait rien d'autre que dormir, manger, courir.

On modélise le comportement de Félix à l'instant t par

$$F(t) = \begin{pmatrix} d(t) \\ m(t) \\ c(t) \end{pmatrix}$$

proba dormir

proba manger

proba courir.

("un instant" = 5 minutes).

Les règles donnent : $F(t+1)$ ne dpd que de $F(t)$ et

- $d(t+1) = \underbrace{0{,}8}\, d(t) + \underbrace{0{,}7}\, m(t) + \underbrace{0{,}6}_{(\dots)}\, c(t)$

proba qu'il dorme après dormir

proba qu'il dorme après repas

ie les flèches qui arrivent sur "dormir" pondérées par la proba d'y arriver depuis les autres états.

- $m(t+1) = 0{,}1\, d(t) + 0{,}1\, c(t)$, et
- $c(t+1) = 0{,}1\, d(t) + 0{,}3\, m(t) + 0{,}3\, c(t)$.

Prop $F(t+1) = M \cdot F(t)$ et donc $F(t) = M^t \cdot F(0)$.

Déf. • X est un état du système si $X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ dont les composantes sont ≥ 0 et dont la somme vaut 1 : $x_1 + x_2 + \dots x_n = 1$.

• L'état est stationnaire si $MX = X$ (ou $X(n+1) = X(n) = \dots X$).

ici : un état est un vecteur de $\mathbb{R}^3$: $F = \begin{pmatrix} d \\ m \\ c \end{pmatrix}$ avec $d, m, c \geq 0$ et $d + m + c = 1$.

Déf. M est irréductible s'il existe une puissance de M, M^n $(n \geq 1)$, dont tous les coefficients sont strictement positifs.

ici : M^2 a tous ses coeffs > 0 donc M est irréductible.

Thm (Perron-Frobenius).

Soit M une matrice de transition d'un processus de Markov. Si M est irréductible, alors il existe un unique état stationnaire X.

De plus, quel que soit l'état initial du système $X(0)$, $X(n) = M^n \cdot X(0) \xrightarrow[n \to +\infty]{} X$.

ici / en pratique :

- M donnée, on calcule $M^2, M^3 \ldots \to M$ irréductible ?
  (python : np.linalg.matrix_power(M, p) donne M^p).
- On cherche $\tilde{X}$ tq $M\tilde{X} = \tilde{X}$ i.e. $(M - I)\tilde{X} = 0$.
  (scipy.linalg.null_space(M − I) donne une base de $\ker(M - I)$ ...)

  ici $\tilde{F} = \begin{pmatrix} 0{,}975\ldots \\ 0{,}116\ldots \\ 0{,}189\ldots \end{pmatrix}$ tq $M\tilde{F} = \tilde{F}$.
- On normalise $\tilde{F}$ pour obtenir l'état du système, qui est stationnaire.

ici : $F = (1/\mathrm{sum}(\tilde{F})) * \tilde{F} = \begin{pmatrix} 0{,}761\ldots \\ 0{,}090\ldots \\ 0{,}147\ldots \end{pmatrix}$.

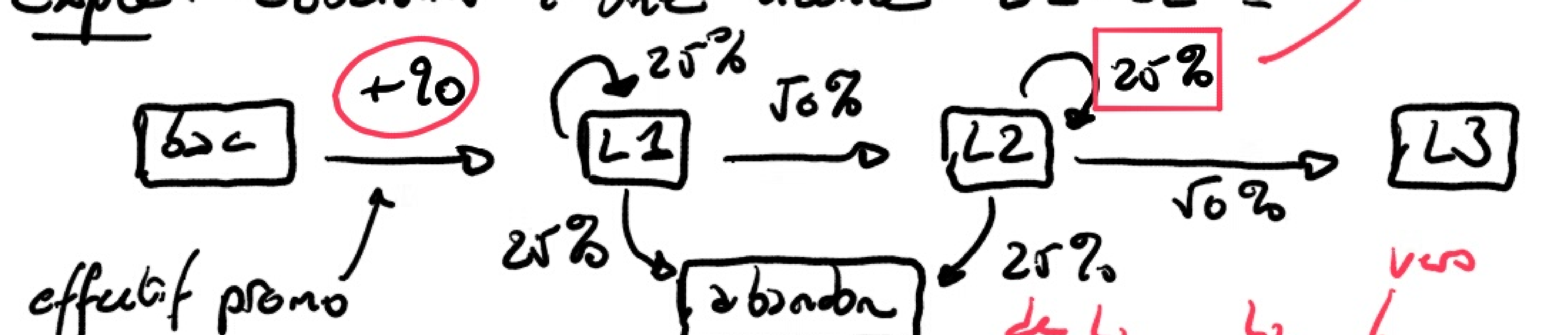
Félix passe
- 76% du tps à dormir,
- 9% du tps à manger,
- 15% du tps à courir.

Conclusion : c'est dur la vie des chats.

Rmq : Le thm ne fonctionne que pour matrices de transition des processus de Markov.

Par contre, le reste est utile dans un cadre beaucoup plus général.

Exple. Étudiants d'une licence L1-L2.



$$X(n) = \begin{pmatrix} x_1(n) \\ x_2(n) \end{pmatrix} : \quad X(n+1) = \begin{array}{c} \text{de} \\ \text{vers} \end{array} \begin{array}{c c} L_1 & L_2 \\ \underbrace{\begin{pmatrix} 1/4 & 0 \\ 1/2 & \boxed{1/4} \end{pmatrix}}_{M} \end{array} \begin{array}{c} \\ L_1 \\ L_2 \end{array} X(n) + \underbrace{\begin{pmatrix} 90 \\ 0 \end{pmatrix}}_{E}$$

$$X(n+1) = M X(n) + E.$$

On a la somme des éléments des colonnes de M est $\neq 1$ (< 1, à cause des abandons et des diplômés).

La suite des états $(L1, L2)$ est incomplète.

Par contre, il existe des effectifs stables : $X = MX + E$
(i.e. des états stationnaires).

<u>Ici</u> il est unique $\begin{pmatrix} 120 \\ 80 \end{pmatrix} = X$.

<u>Si</u> les effectifs sont stables, il y a 120 étudiants en L1 et 80 en L2.

En étudiant $X(n) - X$, on peut prouver que pour toute condition initiale, $X(n) \xrightarrow[n \to +\infty]{} X$.

(cf. TP3 pour d'autres exemples !).