

Propagation bi-dimensionnelle : épidémie

Guillaume Collet

1 Introduction

Dans ce TP, en ce basant sur l'automate cellulaire du feu de forêt, vous allez modéliser la propagation d'une épidémie.

2 Modélisation

Valeurs des cellules : Chaque case de la matrice prendra une valeur entière en fonction de son état :

- 0 pour les cases "vides" (pas d'individu) ;
- 1 pour les cases contenant un "individu sain non vacciné" ;
- 2 pour les cases contenant un "individu sain vacciné" ;
- 3 pour les cases contenant un "individu atteint" ;
- 4 pour les cases contenant un "individu rétabli" ;
- 5 pour les cases contenant un "individu mort".

Vous pouvez proposer d'autres catégories si elles vous semblent pertinentes.

Règles de propagation de l'épidémie : Nous commençons avec des règles très dures pour ensuite ajouter de la variabilité avec des probabilités d'attraper la maladie et de s'en sortir.

1. un "individu sain vacciné" ne peut pas être malade ;
2. un "individu sain non vacciné" dont un des voisins est un "individu atteint" à l'état actuel devient un "individu atteint" à l'étape suivante, sinon il ne change pas d'état ;
3. un "individu atteint" à l'état actuel devient un "individu mort" à l'étape suivante ;
4. un "individu rétabli" ne peut plus être malade ;
5. les cellules vides restent vides.

Dans ce modèle, comme tout malade meurt, vous voyez qu'aucun individu ne peut être "rétabli". Vous changerez ces règles plus tard pour introduire un pourcentage de risque et ainsi pouvoir générer des individus "rétablis"

Dans un fichier nommé `epidemie.py`, vous devez tout d'abord reprendre le code du feu de forêt pour l'adapter à cette nouvelle modélisation. Vous pouvez utiliser `numpy` ou d'autres modules pour faire cet exercice.

3 Observations

Maintenant que vous avez de quoi modéliser la propagation d'une épidémie avec des règles simple, vous pouvez étudier différents facteurs.

Effets de la densité : Je vous propose d'observer le pourcentage de mort en fonction de la densité de population. Écrivez un programme qui crée des automates avec 10%, 20%, 30%, etc de densité de population, qui lance la propagation jusqu'à la stabilité et qui relève la quantité d'individus morts dans une liste.

Je vous invite à visualiser la courbe du pourcentage de morts avec `matplotlib` ou `seaborn`. Observez-vous le fameux seuil de 60% de vaccinés (immunité collective) ?

Effets de la vaccination : Je vous propose maintenant d'observer l'influence de la vaccination sur le pourcentage de morts en fonction de la densité. À vous de modifier votre programme (ou d'en écrire un autre) pour calculer et visualiser ces données.

4 Améliorations

Je vous propose d'intégrer des paramètres plus fin dans la modélisation :

- la vaccination ne protège pas complètement mais utilise un taux de protection
- on peut imaginer faire varier l'infection en fonction d'un taux de propagation et du nombre de voisins infectés.
- on peut ajouter un taux de survie pour les malades
- on peut faire bouger ce taux de survie en fonction du nombre de jours de maladie (avec un taux moyen de durée de la maladie)
- etc

N'hésitez pas à proposer des idées, vous aurez l'occasion d'expliquer vos propositions aux autres étudiant-e-s !