

Algorithme et Programmation 3 - TP 4.5

Propagation d'un virus

13/11/2023

1 Propagation de virus

On veut étudier la propagation d'un virus dans une population. Pour cela, on suppose que la population est représentée par une grille remplie de 0 et de 1. On dit qu'une personne est saine si son état est 0 et qu'elle est infectée si son état est 1. Un exemple de grille est la suivante :

1	0	0	1
0	0	1	0
0	0	0	0
1	0	0	0

Chaque jour, de nouvelles personnes sont infectées : une personne devient infectée si au moins deux de ses quatre voisins directs (horizontaux ou verticaux) sont infectés. Ainsi, la nouvelle personne infectée passe à l'état 1 au jour suivant $j + 1$. Une personne infectée reste infectée pour toujours. Lorsque le tableau ne change pas entre le jour j et le jour $j + 1$, alors on dit que la propagation s'est stabilisée.

1. Avec la population représentée par la grille ci-dessus, calculez à la main au bout de combien de jours la propagation va se stabiliser. Combien de personnes seront alors contaminées ?

Nous souhaitons utiliser python pour étudier la propagation de ce virus. Pour cela on peut utiliser des listes de listes pour représenter des grilles :

```
1 grille = [[1,0,0,1], [0,0,1,0], [0,0,0,0], [1,0,0,0]]
```

2. Implémentez une fonction `updateEtat(grille,i,j)` qui étant donné une grille et la position i, j d'une personne, détermine si la personne est malade au temps suivant (et retourne 0 ou 1 le nouvel état de la personne). Attention au cas particulier si on se trouve sur les bords de la grille.
3. En vous aidant de la fonction `updateEtat` de la question précédente, faites une fonction `updateGrille(grille)` qui renvoie la grille de l'état au jour

suivant. **Attention:** vous ne pouvez pas modifier directement la grille passée en paramètre sinon cela créerait un effet de bord, il faut donc initialiser une nouvelle grille au début de votre fonction.

4. En vous aidant de la fonction précédente, faites une fonction `propagation(grille)` qui, étant donnée une grille de départ, renvoi le nombre de jours nécessaires pour la propagation, ainsi que la grille finale. Pour cela, il faut vérifier à chaque étape si la grille est identique à celle de l'étape précédente.
5. Pour tester votre fonction, lancez la sur la grille donnée en exemple et vérifiez que vous obtenez les mêmes résultats qu'à la question 1.
6. Faire une fonction `nombreInfectes(grille)` qui renvoi le nombre d'infectés dans une population.
7. Faire une fonction `createRandomGrille(n, p)` qui retourne une grille de taille $n \times n$ dans laquelle chaque personne est infectée avec une probabilité $p \in [0,1]$. Pour cela, on fera `import random` et on regardera par exemple la fonction `random.random()`.
8. Recopier et exécuter le code suivant. Que fait ce code ? Quelle est sa complexité ? N'hésitez pas à essayer d'autres paramètres pour `taille_grille` ou `n_tentatives`. Vous pouvez aussi améliorer le figure en regardant la documentation de `plt.plot` dans `matplotlib`.

```
import matplotlib.pyplot as plt

liste_proba = [0.1*i for i in range(10)]
n_tentatives = 100
taille_grille = 10
liste_infectes_init = []
liste_infectes_fin = []
for i in range(len(liste_proba)):
    p = liste_proba[i]
    n_infectes_init = 0
    n_infectes_fin = 0

    for j in range(n_tentatives):
        grille_initiale = createRandomGrille(taille_grille, p)
        n_infectes_init += nombreInfectes(grille_initiale)
        nombre_jours, grille_finale = propagation(grille_initiale)
        n_infectes_fin += nombreInfectes(grille_finale)

    n_infectes_init = n_infectes_init/n_tentatives
    n_infectes_fin = n_infectes_fin/n_tentatives
    liste_infectes_init.append(n_infectes_init)
    liste_infectes_fin.append(n_infectes_fin)

plt.plot(liste_proba, liste_infectes_init, label="Infectes Init")
plt.plot(liste_proba, liste_infectes_fin, label="Infectes Fin")

plt.legend()
plt.show()
```

9. A votre avis, que se passerait-il s'il ne fallait qu'un seul voisin contaminé au lieu de deux pour être contaminé à son tour ? Combien de contaminé seraient suffisants au jour 0 pour contaminer toute la grille ?
10. Le code suivant présente une animation de la propagation. Dans la fonction `FuncAnimation`, le paramètre `frames` décide sur combien de jour on regarde la propagation, le paramètre `interval` indique combien de temps séparent deux étapes (en millisecondes). Faites plusieurs tests avec une proba d'infection de 0.1 puis de 0.2.

```
import matplotlib.animation as animation

grille_animated = grille
def animatePropagation(grille):

    fig = plt.figure()
    im = plt.imshow(grille, animated=True)

    def animate(i):
        global grille_animated
        if i == 0:
            grille_animated = grille
            im.set_array(grille_animated)
            grille_animated = updateGrille(grille_animated)
            return im,

    ani = animation.FuncAnimation(fig, animate, frames=50,
                                  interval=200, blit=True, repeat=True)

    plt.show()

taille_grille = 20
proba_infection = 0.2
random_grille = createRandomGrille(taille_grille,proba_infection)
animatePropagation(random_grille)
```

On a finalement réussi à trouver un vaccin contre ce virus étrange. Les personnes vaccinées ne peuvent pas être contaminées par le virus. Dans la grille, on représentera ces personnes par un 2.

11. Considérez la grille de l'exemple, avec une personne vaccinée en plus. Combien de contaminations cette vaccination a-t-elle empêché ?

1	2	0	1
0	0	1	0
0	0	0	0
1	0	0	0

12. Réécrire les fonctions `updateEtat` et `nombreInfectes` si nécessaire pour qu'elle prenne en compte les vacciner.

13. Réécrire la fonction `createRandomGrille(n, pinfection, pvaccin)` avec deux probabilités : celle d'être contaminée et celle d'être vacciné. On suppose que $p_{infection} + p_{vaccin} \leq 1$.
 14. Relancer le code de la question 8 avec une probabilité de vaccination de 0.1 (c'est à dire 10% de vaccinés) puis 0.2, 0.3 (etc.), et comparer les résultats.
 15. Relancer le code de la question 10 avec différentes probabilités de vaccination, et comparer les résultats.
- Bonus. Si vous avez fini, inspirez vous de cet exercice pour implémenter le Jeu de la Vie de Conway.