



Programación del Algoritmo de propagación por contagio en Python

Conceptos Avanzados de Internet

*César Francisco San Nicolás Martínez
Jorge San Emeterio Villalaín*

1. El algoritmo

Contagio:

- Modelo de adopción de una nueva idea, tendencia o producto
- Cada nodo puede estar en dos estados:
 - 1: ha adoptado el nuevo servicio, tendencia o producto
 - 0: no ha adoptado el nuevo servicio, tendencia o producto
- Un nodo pasa del estado 0 a 1 cuando la proporción de sus vecinos en estado 1 supera un determinado umbral p (%)
- En el modelo más sencillo todos los nodos tienen el mismo umbral p .

2. Formulación del problema

Tenemos los siguientes datos de entrada del problema:

- N nodos.
- A = matriz de adyacencia de nodos ($N \times N$):
 - $A(i,j) = 1$ si Nodo i y Nodo j están conectados.
 - $A(i,j) = 0$ si Nodo i y Nodo j no están conectados.
 - Autoenlaces no contemplados, por lo que $A(i,i) = 0$ si $i = j$.
- Dos nodos contagiados inicialmente.
- Umbral de contagio p .

El dato de salida del problema será un vector de N posiciones que tendrá dos posibles valores: 0 ó 1.

- La posición n de ese vector valdrá 1 si el Nodo n está contagiado* al final del algoritmo.
- La posición n de ese vector valdrá 0 si el Nodo n no está contagiado* al final del algoritmo.

**Estar contagiado = adoptar el nuevo servicio, tendencia o producto.

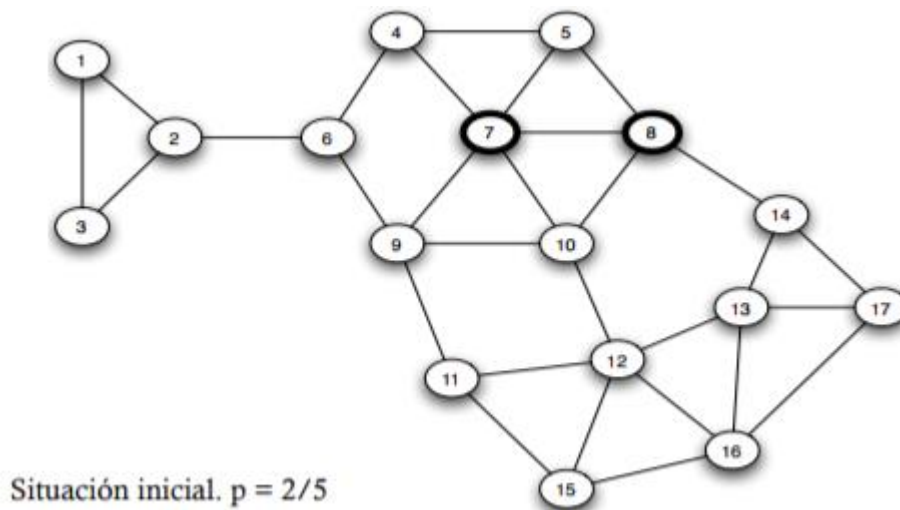
3. Ejecución del algoritmo

En primer lugar, se puede ver una explicación del algoritmo programado en pseudocódigo:

```
While true
  Foreach Nodo n
    If Nodos_contagiados.contains(Nodo n)
      Continue
    Nodos_ = nodos conectados al Nodo n
    Nodos_contagiados = nodos conectados al Nodo n y contagiados
    Umbral_n = Nodos_contagiados / Nodos_
    If p >= Umbral_n
      Nuevos_nodos_contagiados.add(Nodo n)
  end

  If Nuevos_nodos_contagiados.length = 0
    Break
  Nodos_contagiados = nodos_contagiados U Nuevos_nodos_contagiados
end
```

Para comprobar la correcta implementación, hemos utilizado el ejemplo visto en teoría:

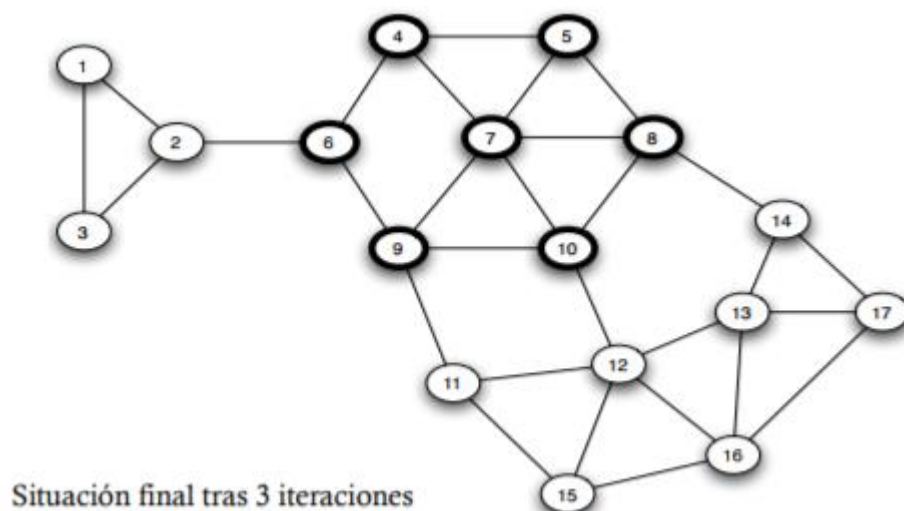


A continuación, podemos ver una imagen en la que se ve la ejecución del algoritmo con los parámetros de entrada del caso de teoría:

```

Windows PowerShell
PS C:\Users\Cesar\Desktop\Contagio Python> python Contagio.py --archivomatriz=matriz.txt ejecutarAlgoritmo
La matriz de adyacencia A es
[[0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]]
Los nodos contagiados inicialmente son 6 y 7
El umbral de contagio es 0.4
*****
EMPIEZA EL ALGORITMO
*****
Iteración nº1
*****
El nodo 4 ha sido contagiado
El nodo 9 ha sido contagiado
*****
Iteración nº2
*****
El nodo 3 ha sido contagiado
El nodo 8 ha sido contagiado
*****
Iteración nº3
*****
El nodo 5 ha sido contagiado
*****
Iteración nº4
*****
No hay nuevos nodos contagiados en la iteración nº4
El algoritmo ha terminado...

```



Vemos que el resultado del algoritmo programado y del algoritmo de teoría coinciden en su resultado. Tras 3 iteraciones, hay 7 nodos contagiados.

** En teoría, los índices van de 1 a 17, mientras que en el algoritmo programado van de 0 a 16.

4. Bibliografía y recursos

- Teoría de la asignatura Conceptos Avanzados de Internet.
- <https://www.pybonacci.org>
- Paquetes Numpy y Click de Python.

ANEXO. Código del algoritmo en Python

```
import click
import numpy as np

@click.group()
@click.option('--archivomatriz',
              prompt=True,
              help='Ruta del archivo donde se encuentra definida la matriz de
adyacencia')
@click.pass_context
def cli(ctx, archivomatriz):
    ctx.obj = archivomatriz

@cli.command(name='ejecutarAlgoritmo')
@click.option('--semillauno',
              default=6,
              help='Primer nodo contagiado inicialmente')
@click.option('--semillados',
              default=7,
              help='Segundo nodo contagiado inicialmente')
@click.option('--umbral',
              default=0.4,
              help='Umbral para establecer contagio')
@click.pass_context
def ejec_algoritmo(ctx, semillauno, semillados, umbral):

    A = extraer_matriz(ctx.obj)
    nodos_contagiados = algoritmo(A, semillauno, semillados, umbral)

def algoritmo(A, semillauno, semillados, umbral):
    numero_nodos = len(A)
    print('La matriz de adyacencia A es ')
    print(A)
    print('Los nodos contagiados inicialmente son ' + str(semillauno) + ' y ' +
str(semillados))
    print('El umbral de contagio es ' + str(umbral))

    nodos_contagiados = np.zeros((1, numero_nodos))
    nodos_contagiados[0][semillauno] = 1
    nodos_contagiados[0][semillados] = 1

    print('*****')
    print('EMPIEZA EL ALGORITMO')

    iter = 1
    while (True):
        nuevos_nodos_contagiados = []
        print('*****')
        print('*****')
        print('Iteración n°' + str(iter))
        print('*****')
        for nodo in range(numero_nodos):
            if (nodos_contagiados[0][nodo] == 1):
                continue
            fila = A[nodo, :]
            nodos_totales_nodo = sum(fila)
            nodos_contagiados_nodo = sum(np.multiply(fila, nodos_contagiados[0, :]))
            umbral_nodo = nodos_contagiados_nodo / nodos_totales_nodo
            if (umbral_nodo >= umbral):
                nuevos_nodos_contagiados.append(nodo)
                print('El nodo ' + str(nodo) + ' ha sido contagiado')

        if (len(nuevos_nodos_contagiados) == 0):
            print('No hay nuevos nodos contagiados en la iteración n°' + str(iter))
```

```

        print('El algoritmo ha terminado...')
        print('*****')
        break

    iter = iter + 1
    nodos_contagiados = actualizar_nodos_contagiados(nodos_contagiados,
nuevos_nodos_contagiados)

    return nodos_contagiados

def extraer_matriz(path):

    file = open(path, 'r')
    filas = file.readlines()
    num_nodos = len(filas)
    matriz = np.zeros((num_nodos, num_nodos))
    cont_fila = 0
    for fila in filas:
        elementos = fila.split()
        cont_col = 0
        for elem in elementos:
            matriz[cont_fila][cont_col] = elem
            cont_col = cont_col + 1
        cont_fila = cont_fila + 1

    return matriz

def actualizar_nodos_contagiados(nodos_contagiados_actuales,
nuevos_nodos_contagiados):

    nodos_contagiados = nodos_contagiados_actuales

    for index in nuevos_nodos_contagiados:
        nodos_contagiados[0][index] = 1

    return nodos_contagiados

if __name__ == '__main__':
    cli()

```

