

Experimentos con el algoritmo de Floyd-Warshall en Python

Gabriela Sánchez Y.

Introducción

En esta práctica se trabaja con un tipo específico de grafos no dirigidos, con el objetivo de analizar ciertas propiedades de los mismos, tales como la distancia promedio y el coeficiente de agrupamiento.

Contrucción de los grafos

Los grafos con los que se trabaja tienen sus nodos distribuidos uniformemente en un círculo unitario. Las aristas se construyen en base a el parámetro k , éste parámetro indica con cuántos nodos se conecta cada nodo i del grafo, tanto en sentido horario como en sentido antihorario, finalmente se agregan aristas al azar con cierta probabilidad p . En las figuras 1 y 2 se muestran unos ejemplos.

```
n = 8 # nodos
k = 2
i = 0
G = Grafo()
ang = math.radians(360/n) # angulo en radianes de separacion entre nodos
r1 = 0.5
xc = 0.5
yc = 0.5
for i in range(n):
    x = r1*math.cos(i*ang) + xc
    y = r1*math.sin(i*ang) + yc
    r = 0.02
    G.nodo(i,x,y,r)
    i += 1
p = 1
for nodo in G.V:
    for j in range(k):
```

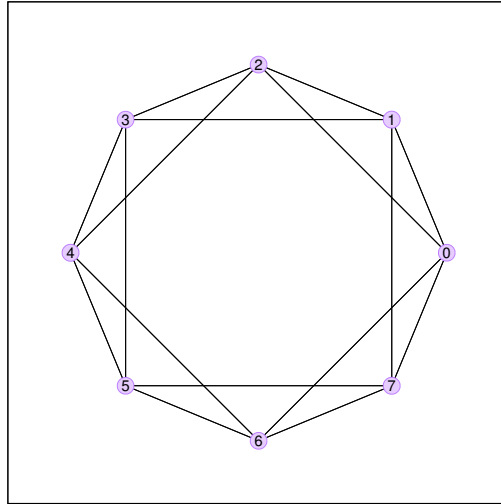


Figura 1: Grafo con $n = 8$ nodos, $k = 2$ y $p = 0$.

```

if (nodo + j + 1) >= n:
    a = (nodo + j + 1) - n
else:
    a = (nodo + j + 1)
G.arista(nodo, a, False, p)

```

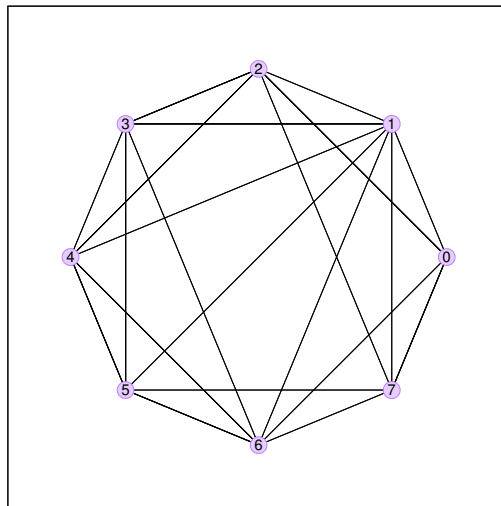


Figura 2: Grafo con $n = 8$ nodos, $k = 2$ y $p = 0.2$.

Coeficiente de agrupamiento

Para cada nodo i en el grafo, el coeficiente de agrupamiento se calcula en base a la ecuación (1), donde m representa la cantidad de aristas entre la vecindad del nodo i y n la cantidad de nodos en la vecindad del nodo i ,

$$\delta = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)}. \quad (1)$$

Por ejemplo, para el grafo de la figura 2, la vecindad del nodo $i = 0$ es $\{1, 2, 6, 7\}$ y las aristas entre su vecindad son $\{(1, 2), (1, 6), (1, 7), (2, 7), (6, 7)\}$, por lo tanto $n = 4$ y $m = 5$. De manera que $\delta = \frac{10}{12} = 0.83333$.

Finalmente, el coeficiente de agrupamiento de un grafo es el promedio de los coeficientes de cada uno de sus nodos. El cálculo del mismo se realiza mediante la función `clust_coef()` del archivo `grafo.py`.

Distancia promedio

En la práctica anterior [1] se programó el algoritmo de Floyd-Warshall que determina la distancia mínima entre cualquier par de nodos en un grafo. Para calcular la distancia promedio del grafo simplemente se suman los resultados del algoritmo y se divide entre n^2 , donde n es el número de nodos en el grafo.

Nos interesa que el valor de la distancia promedio d sea normalizada, por lo que se propone una cota superior s para la distancia máxima, de manera que

$$0 \leq \frac{d}{s} \leq 1.$$

Si $k = 1$ y $p = 0$, la distancia máxima se tiene en el lado opuesto, esto es $d_m = \frac{n}{2}$, en la medida en que aumente el valor de k , se crean “saltos”, en consecuencia la distancia máxima se reduce al menos en una unidad. De acuerdo a ésto, se propone la cota descrita en la ecuación (2)

$$s = \frac{n}{2} - (k - 1). \quad (2)$$

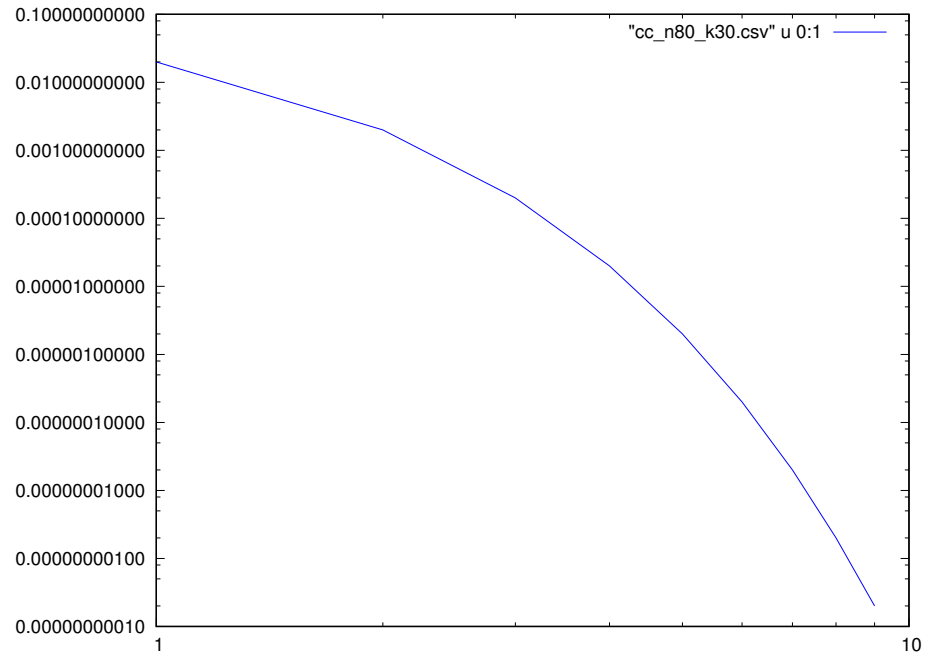


Figura 3: Comportamiento del coeficiente de agrupamiento para $n = 80$ y $k = 30$.

Resultados

Para poder observar el comportamiento del coeficiente de agrupamiento se realizó un experimento en el que se fijan valores para n y k pero se varía la probabilidad $p = 2^{-i}$ con $i = \{1, 2, \dots, 10\}$. Los resultados se muestran en la figura 3.

Referencias

- [1] Gabriela Sánchez Y. Medición experimental de la complejidad asintótica.
https://github.com/Saphira3000/Flujo_Redes/tree/master/tarea3.