

# CAIM Lab, Session 7: Introduction to igraph

Moreno Oya, Daniel  
Cidraque Sala, Carles

## TASK 1

En la primera tarea debíamos reproducir un gráfico de un grafo, el cual es un plot del coeficiente de clustering y el camino más corto medio en función del parámetro  $p$  (probabilidad de rewiring) según el modelo de Watts-Strogatz.

En el gráfico dado podemos ver que son 14 puntos y los valores de  $p$  están entre 0,0001 y 1. También podemos ver que el eje x está en escala logarítmica y que los valores del coeficiente de clustering y del camino más corto medio están normalizados, es decir, están divididos por el valor cuando  $p = 0,0001$ .

Para generar la gráfica hemos hecho un script en Python, en el cual hemos creado una lista de 14 valores de probabilidad entre 0 y 1 para generar los diferentes grafos según el modelo Watts-Strogatz. Después para cada grafo creado hemos calculado el coeficiente de clustering y el camino más corto medio, ambos normalizados, y hemos creado un plot. Cada grafo del modelo Watts-Strogatz lo hemos creado con 2000 nodos, 20 vecinos por nodo y el valor correspondiente de la probabilidad de rewiring.

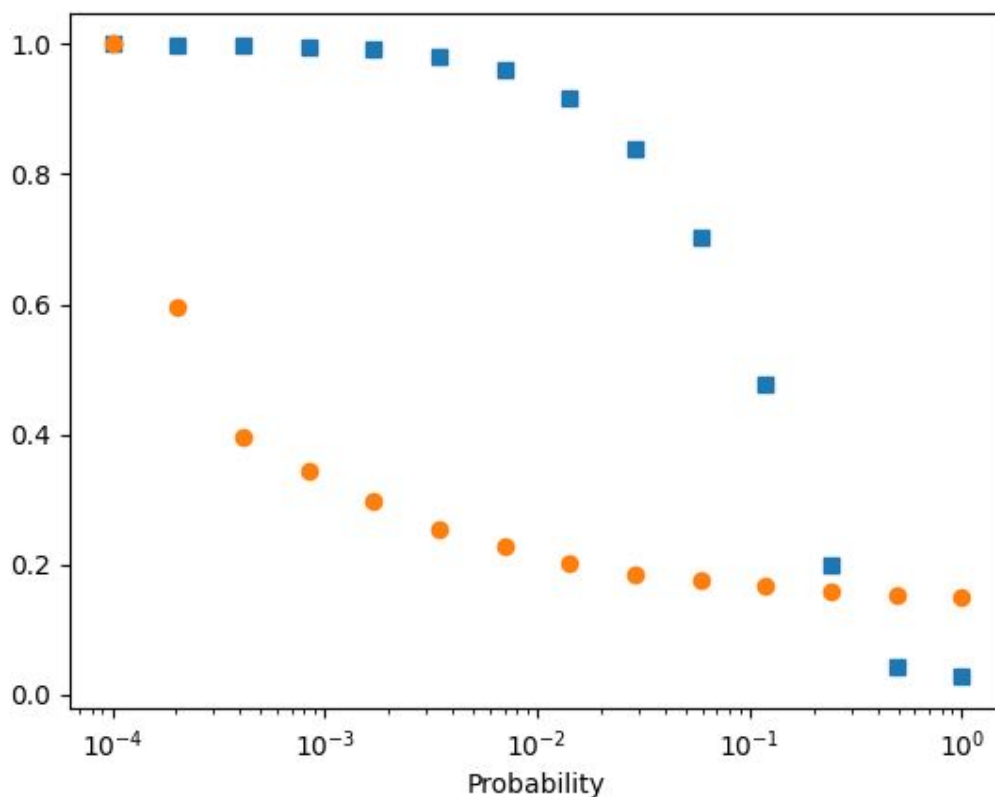


Fig1. Plot del coeficiente de clustering(azul) y del camino más corto medio(naranja) según la probabilidad de de rewiring

Como podemos observar, a medida que incrementamos la probabilidad de rewiring, tenemos un decremento rápido del camino más corto medio(naranja) y un decremento más lento del coeficiente de clustering(azul).

## TASK 2

En la segunda tarea debíamos cargar el fichero edges.txt, el cual era un grafo no dirigido. Seguidamente debíamos calcular algunas propiedades y características del grafo cargado:

```
dani@dani-HP-Pavilion-Notebook:~/Escritorio/caim/session/networks$
dani@dani-HP-Pavilion-Notebook:~/Escritorio/caim/session/networks$ python3 igraph_networks2.py
Número de arcos: 602
Número de vértices: 62
Diámetro del grafo: 2
Transitivity del grafo: 0.5227690047741461
```

Fig2. Número de arcos y nodos, diámetro y transitivity del grafo del archivo edges.txt

```
Transitivity del grafo: 0.5227690047741461
Degree distribution: N = 62, mean +- sd: 19.4194 +- 12.5388
[ 3, 4): ** (2)
[ 4, 5): (0)
[ 5, 6): (0)
[ 6, 7): (0)
[ 7, 8): ***** (8)
[ 8, 9): (0)
[ 9, 10): ***** (5)
[10, 11): (0)
[11, 12): **** (4)
[12, 13): (0)
[13, 14): ***** (7)
[14, 15): (0)
[15, 16): ***** (7)
[16, 17): (0)
[17, 18): *** (3)
[18, 19): (0)
[19, 20): ** (2)
[20, 21): (0)
[21, 22): *** (3)
[22, 23): (0)
[23, 24): **** (4)
[24, 25): (0)
[25, 26): ** (2)
[26, 27): (0)
[27, 28): * (1)
[28, 29): (0)
[29, 30): * (1)
[30, 31): (0)
[31, 32): ** (2)
[32, 33): (0)
[33, 34): *** (3)
[34, 35): (0)
[35, 36): *** (3)
[36, 37): (0)
[37, 38): * (1)
[38, 39): (0)
[39, 40): * (1)
[40, 41): (0)
[41, 42): (0)
[42, 43): (0)
[43, 44): * (1)
[44, 45): (0)
[45, 46): (0)
[46, 47): (0)
[47, 48): (0)
[48, 49): (0)
[49, 50): (0)
[50, 51): (0)
[51, 52): (0)
[52, 53): (0)
[53, 54): (0)
[54, 55): (0)
[55, 56): (0)
[56, 57): (0)
[57, 58): (0)
[58, 59): (0)
[59, 60): * (1)
[60, 61): (0)
[61, 62): (0)
[62, 63): (0)
[63, 64): * (1)
```

Fig3. Degree distribution del grafo del archivo edges.txt

Después debíamos calcular el pagerank para cada nodo del grafo y visualizarlo con el tamaño de los nodos proporcional a su pagerank:

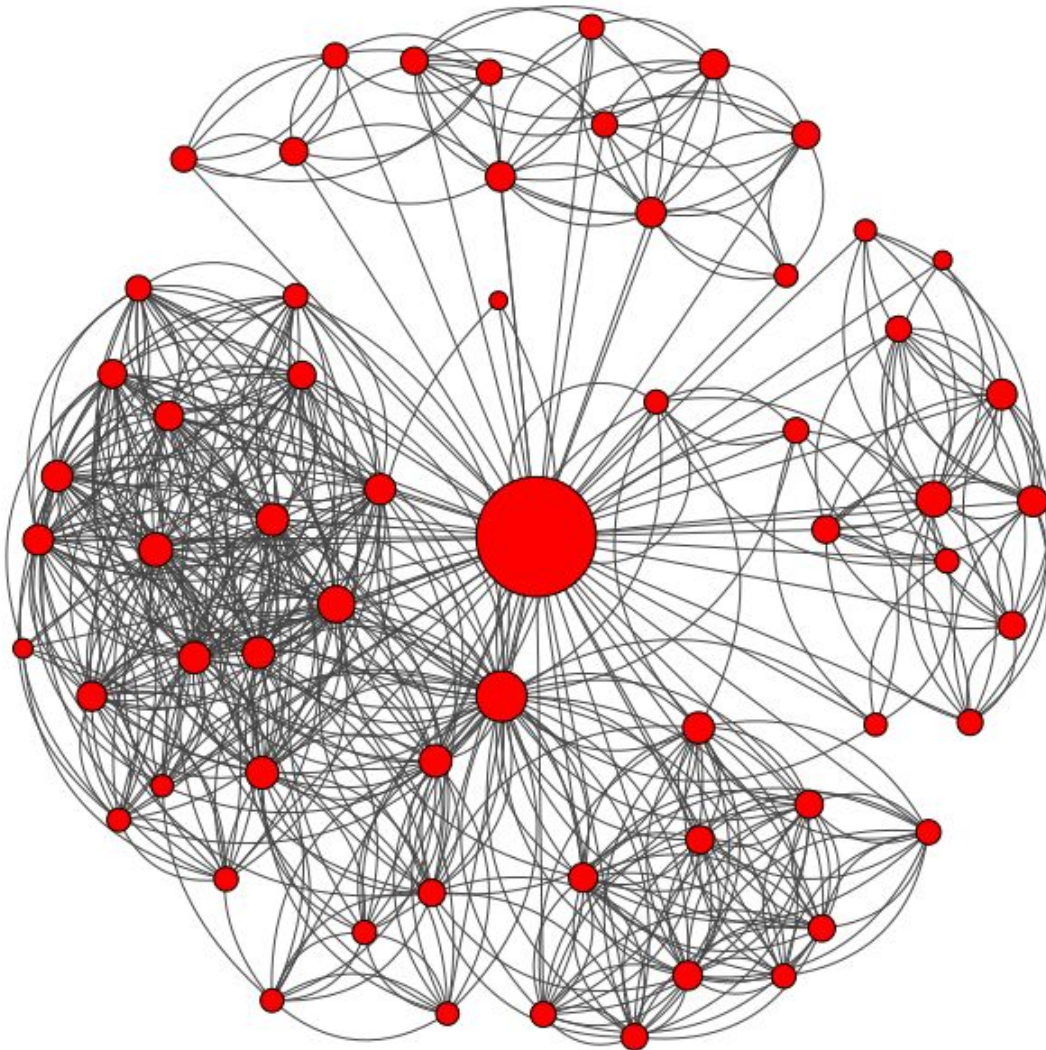


Fig4. Plot del grafo del archivo edges.txt con el tamaño de los nodos proporcional a su pagerank

También debíamos indicar si la red tenía aspecto de red aleatoria. Para analizarlo debemos observar la distribución de los grados de los nodos, que nos indica cómo está repartida la red :

[63, 13, 9, 39, 35, 7, 9, 59, 13, 25, 13, 11, 33, 31, 19, 21, 27, 7, 23, 33, 7, 9, 13, 35, 37, 17, 13, 13, 17, 9, 35, 23, 25, 33, 31, 3, 21, 11, 11, 13, 21, 11, 23, 29, 43, 15, 15, 23, 15, 7, 15, 15, 7, 19, 7, 15, 7, 9, 15, 17, 3, 7]

Podemos observar que, excepto unos pocos nodos, los grados están organizados de manera uniforme. También podemos ver que existe un nodo hub (nodo con un grado muy superior a los demás), lo cual es propiedad de los modelos no aleatorios (modelo Barabási-Albert).

Además podemos observar en la figura 4 que , excepto el nodo central, la mayoría de nodos tienen un tamaño similar (proporcional a su pagerank). También nos indica que no se trata de una red aleatoria.

Para finalizar nos pedían utilizar un algoritmo de detección de comunidades. Nosotros escogimos el edge betweenness. El algoritmo nos detecta 7 clusters y el cluster más grande contiene 26 nodos. También debíamos hacer un histograma de los tamaños de los clusters.

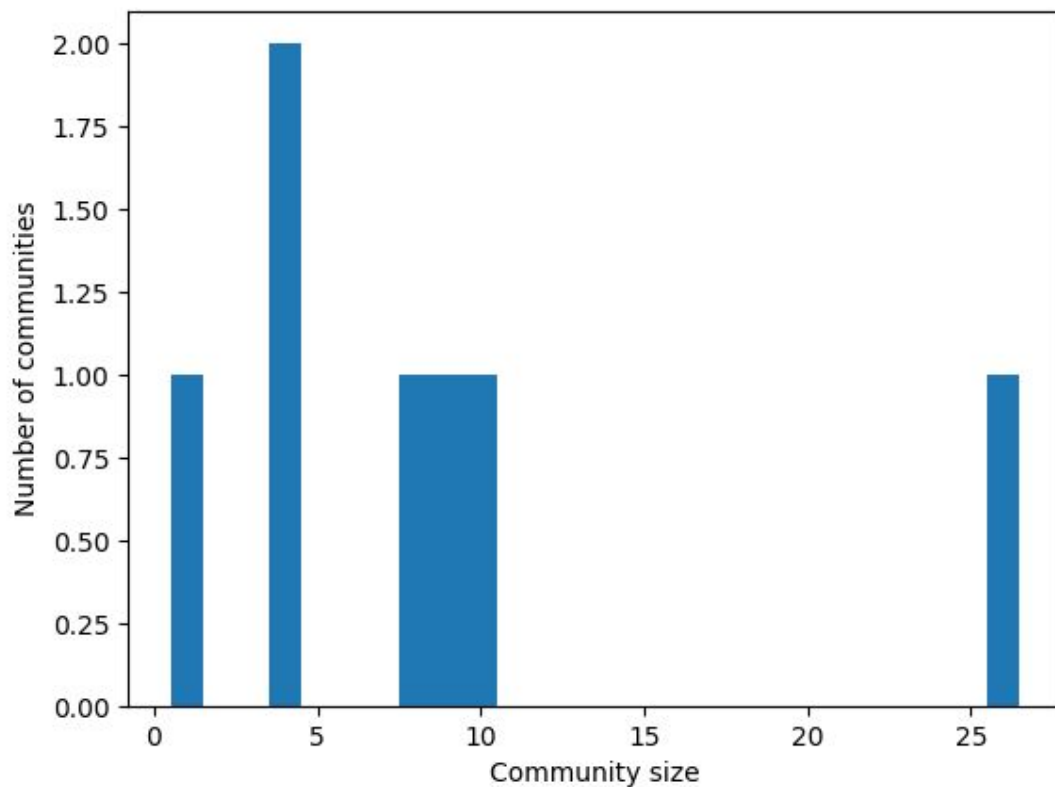


Fig5. Histograma de los tamaños de los clusters detectados por el algoritmo

Como podemos observar en la figura 5, excepto el cluster de tamaño 26, la mayoría de los clusters tienen tamaño entre 0-10 nodos.

También debíamos producir un plot del grafo en el cual se observen las diferentes comunidades:



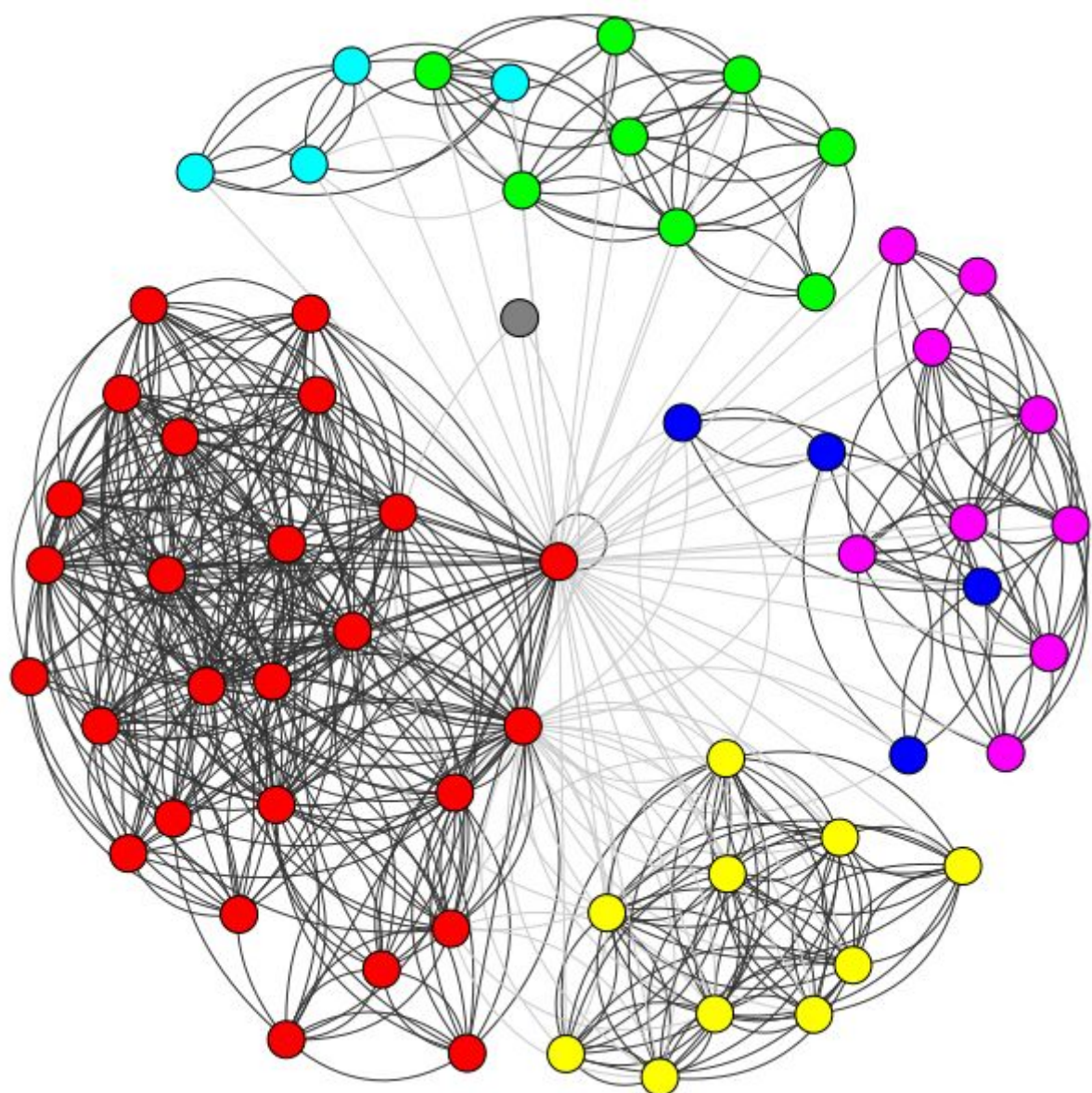


Fig6. Plot del grafo del archivo edges.txt, en el cual cada comunidad está representada por un color