Experimentos con el algoritmo de Floyd-Wharshall en Python

Astrid González

Optimización de flujo en redes

23 de abril de 2018

Introducción

El algoritmo de Floyd-Wharshall [3] es de utilidad para obtener las distancias entre cada par de nodos presentes en un grafo $G(V,E)$, tal que V es el conjunto de vértices o nodos que pertenecen a G, es decir $\{i \in V\} \in G$, mientras que E es el conjunto de aristas o conexiones presentes en G, tal que $\{(i,j) \in E\} \in G$.

Dicho algoritmo es responsable de computar una matriz de distancia $D_i j$ tal que i,$j \in V$ con complejidad asintótica O(V^3) [3] de la cual es posible obtener la distancia promedio para G.

El coeficiente de agrupamiento de un grafo corresponde al promedio de los coeficientes de agrupamiento de cada uno de sus nodos. Éstos últimos permiten determinar el índice de la cantidad de conexiones entre los vecinos para cada nodo en el grafo.

Metodología

Se utilizó la versión 3.6.4 del lenguaje de programación Python [4] acorde al procedimiento obtenido previamente para la generación de grafos simples, dirigidos y ponderados [1, 2]. Así mismo, fue utilizada la versión 4.6 de Gnuplot [5] para visualizar la relación entre los parámetros obtenidos.

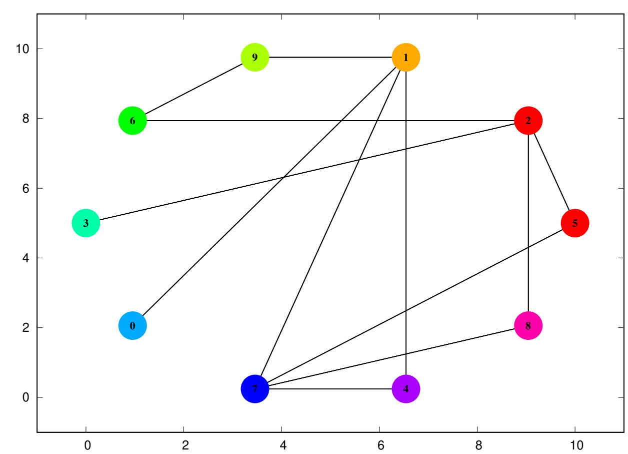


Figura 1: Grafo de 10 nodos, $k = 5$, $p = 0.5$

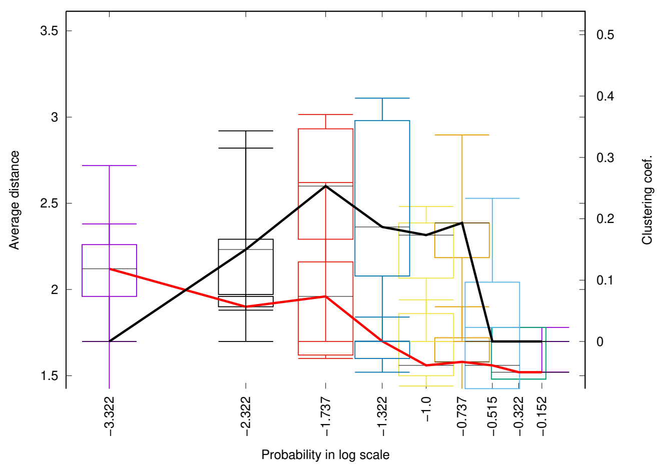
Para cada tamaño n, se llevaron a cabo cinco repeticiones para la generación de diferentes grafos con distinta probabilidad de unión p. Dicha probabilidad se encuentra dentro del rango $[0, 1]$ y corresponde a la probabilidad de conexión entre cada par de nodos en el grafo, es decir de que el nodo i y el nodo j se encuentren en $G(E)$ tal que $(i, j) \in E$ siempre que cualquiera de los nodos en la arista no tenga más de k vecinos, donde $k = \frac{n}{2}$. Para la ilustración de dicho proceso, en la figura 1 se puede apreciar un ejemplo de un grafo con dichas características.

Posteriormente se utiliza el algoritmo Floyd-Warshall para calcular las distancias entre cada par de nodos, y se obtiene la distancia promedio para el grafo dividiento la suma de las distancias de todos los pares de nodos entre el cuadrado de la cantidad de nodos presentes en el grafo. Ésto es debido a que para cada $(i, j) \in E$ se tiene que $(i, j) = (j, i)$.

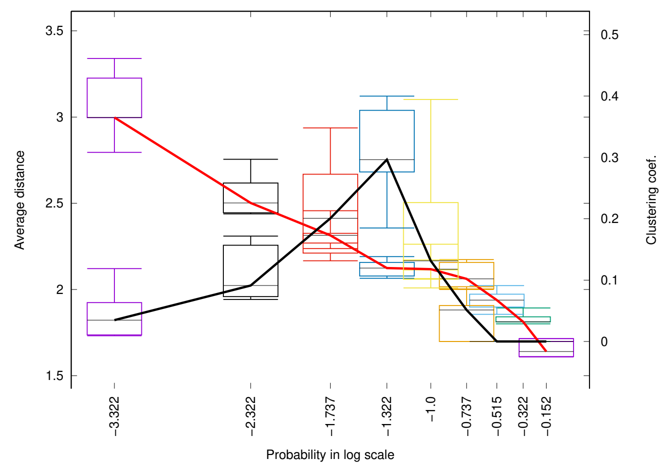
Se procede a calcular el coeficiente de agrupamiento de cada nodo i en el grafo G, el cual es obtenido mediante el cociente de la cantidad de aristas presentes entre los vecinos del nodo i, k_i y la cantidad máxima de posibles aristas que se pueden generar entre los j vecinos del nodo i.

$$coefC_i = \frac{k_i}{\frac{n!}{2!(n-2)!}}$$

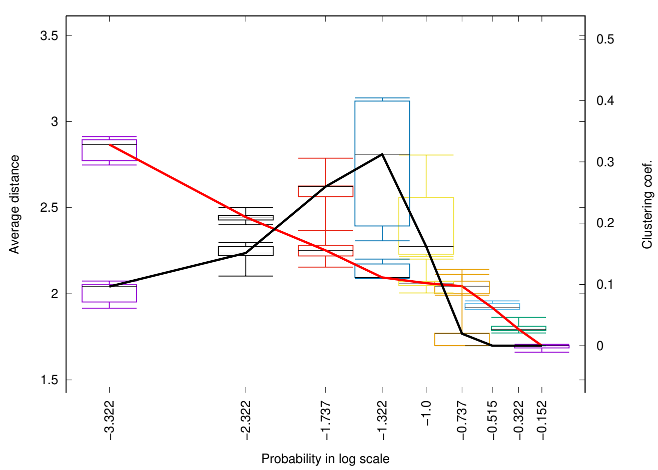
Una vez determinado el coeficiente de cada nodo, se genera el coeficiente de agrupamiento del grafo mediante el cociente de la suma de los coeficientes individuales y la cantidad de nodos presentes en G.



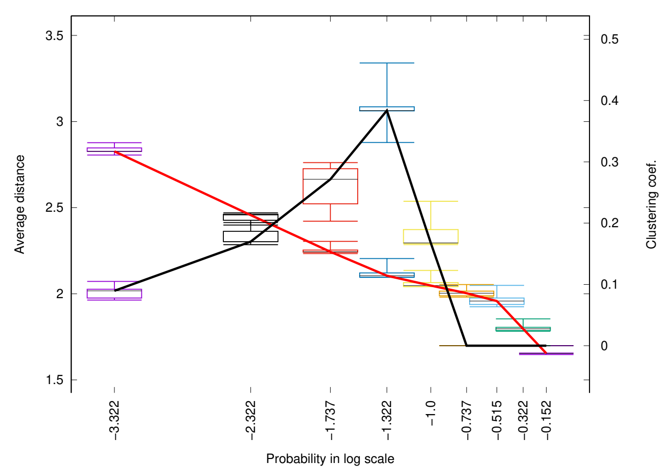
(a) Grafo de 10 nodos.



(b) Grafo de 50 nodos.



(c) Grafo de 100 nodos.



(d) Grafo de 200 nodos.

Figura 2: Visualización del comportamiento de los valores de la distancia promedio y el coeficiente de agrupamiento, en base al logaritmo de la probabilidad de unión del Grafo.

Resultados

En la figura 2 se puede observar el comportamiento de los valores obtenidos de la distancia promedio obtenida y el coeficiente de agrupamiento de cada grafo con probabilidad p, donde $p = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$.

El eje X de los diagramas corresponde al logaritmo de base 2 de la probabilidad p, mientras que el eje Y del lado izquierdo pertenece al rango de la distancia promedio, y el eje Y del lado derecho al rango del coeficiente de agrupamiento.

La línea de color rojo corresponde a la distancia promedio, mientras que la línea de color negro es el comportamiento del coeficiente de agrupamiento.

Se puede observar como para valores de n suficientemente grandes, es decir, para $n >= 50$, el comportamiento es similar entre los diversos valores de n. Ésto significa que el valor de la distancia promedio del grafo sera casi inversamente proporcional a la probabilidad p. Mientras que el coeficiente de agrupamiento será casi proporcional a p siempre que $p < 0.5$, por otro lado, si $p >= 0.5$ entonces se podría decir que el comportamiento que se aprecia para el coeficiente de agrupamiento es casi inversamente proporcional a p.

Referencias

[1] González, A. (2018a). Medición experimental de la complejidad asintótica con python y gnuplot. *Universidad Autónoma de Nuevo León*, Optimización de flujo en redes.

[2] González, A. (2018b). Visualización de grafos simples, ponderados y dirigidos con gnuplot. *Universidad Autónoma de Nuevo León*, Optimización de flujo en redes.

[3] Jungnickel, D. (1999). *Graphs, networks and algorithms*, volume 5. Springer. ISBN: 9783540727798.

[4] Van Rossum, G. and the Python Development Team (2018). The python language reference. release 3.6.4. `https://docs.python.org/3.6/`. Consultado en Febrero 2018.

[5] Williams, T. and Kelley, C. (2013). Gnuplot 4.6: an interactive plotting program. `http://gnuplot.info/`. Consultado en Febrero 2018.