Representación de redes a través de la teoría de grafos en Python.

Alumno: Astrid González, Matrícula: 1423921.

Materia: Optimización de flujo en redes.

18 de febrero de 2018

Introducción

Una red puede ser considerada un catálogo de los componentes de un sistema y las interacciones entre ellos. A los componentes se les denomina nodos o vértices, mientras que a las interacciones se les conoce como aristas o, simplemente, conexiones[2].

Podemos observar ejemplos de redes en la vida diaria[1] tales como: las redes eléctricas que abastecen una zona, las redes telefónicas necesarias para comunicarnos, e incluso, los sistemas viales empleados para transitar zonas.

Si bien a lo largo de la literatura científica en algunos casos los términos *"red"* y *"grafo"* son considerados sinónimos, formalmente un grafo es la representación matemática de una red. La teoría de grafos es una rama de las matemáticas que consiste en representar una red en forma de grafo con la finalidad de simplificar un problema matemático[2].

En éste trabajo se empleará Python para el procesamiento de datos y Gnuplot para la visualización de los mismos. Python es un lenguaje de programación de alto nivel orientado a objetos[3]. Es clasificado como un lenguaje interpretado, lo que quiere decir que el código no es directamente ejecutado, sino que es leído y ejecutado por otro programa. Gnuplot es una utilidad gráfica que permite visualizar datos y funciones matemáticas de manera interactiva.

Metodología

Para la generación de los algoritmos se utilizó la versión 3.6 de Python. Se importaron las librerías: *random*, para la obtención de números aleatorios, y *pandas*, para integrar los datos en un marco de información.

El número de nodos presentes en el grafo se determina por la variable 'n', la cual se definió en un valor entero de 100 ($n = 100$).

Para la obtención de datos se generó un bucle para cada i, en el rango $[1 : n]$, con la finalidad de obtener valores aleatorios entre 0 y 1 que fueron almacenados en las variables 'x', 'y', 'w', y 'z'. Éstas variables corresponden a las coordenadas X_i, coordenadas Y_i, el tamaño del nodo i, y el color del nodo i, respectivamente. Para poder procesar la información, los valores obtenidos para cada variable de cada i son posteriormente almacenados en listas correspondientes.

Se procede a diseñar un marco de información que contendrá las listas de las variables. Una vez teniendo integrados los datos, es posible ordenarlos en base a una de las variables manteniendo la congruencia de los datos. Se decidió que dicha variable fuera el color, y se realizó el ordenamiento de los datos del marco de información de forma ascendente por el método de quicksort.

Posteriormente, los datos ordenados son grabados en un archivo con extensión '.dat', en donde se tendrán cuatro columnas y cien filas. Cada columna j representa una de las variables 'x', 'y', 'w', o 'z', y cada fila i corresponde a la información del nodo i. Las coordenadas reordenadas (X_i,Y_i) para cada nodo i se reingresaron a sus listas de origen sustituyendo los valores previos.

Se diseñó un script con extensión '.dat' que generará un grafo a partir de los datos, cuyos ejes X y Y se establecen dentro del rango $[0 : 1]$. Se utilizaron

las coordenadas (X_i,Y_i) presentes en las columnas 1 y 2 para representar la posición de cada nodo i, mientras que su tamaño y color se encuentran representados en la columna 3 y 4, respectivamente. Las coordenadas de los nodos son entonces utilizadas para indicar las posiciones del punto de inicio y punto final de las aristas que conectarán los nodos.

Finalmente, para la obtención del grafo se utilizó la versión 5.2 de Gnuplot. En donde mediante el uso del comando *load* es posible leer y ejecutar los comandos presentes en el script (*ver pseudocódigo 1*). Cabe mencionar que el grafo de salida del script se produce en formato '.png' para facilitar su visualización.

Pseudocódigo 1: Visualización en Gnuplot de la representación de los datos obtenidos en Python.

Entrada: Script "aristas.dat", archivo "nodos.dat".
Salida: Grafo de los datos de entrada.
inicio
| load "aristas.dat"
fin

Resultados

Los valores generados en Python por medio de la función '*random()*' son de tipo flotante, pertenecen al conjunto de números reales, y se encuentran entre 0 y 1. Dado que se utilizó ésta función para generar los valores de color, es posible correlacionar un gradiente de color tomando como correspondencia los valores dentro del rango permitido, lo que nos permite designar un color a cada nodo.

Por consecuente, dentro del gradiente, los nodos con un valor en la columna de color cercano a cero serán de color verde, mientras que los nodos con valor cercano a uno serán de color rosa (*ver figura 1*).

En el grafo, se puede observar que cada nodo está representado por un punto dado por las coordenadas (X_i,Y_i), cuyo diámetro está definido por su respectivo valor de tamaño.

Gracias al ordenamiento de los datos, las conexiones entre los nodos se realizaron de forma congruente con los valores designados de color. Lo que quiere decir que cada nodo i estará conectado con aquel nodo k que tenga el valor de color más cercano al valor del nodo i.

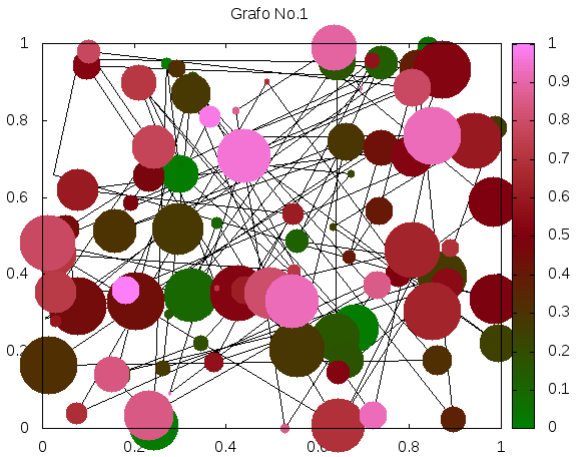

Figura 1: Grafo No.1 obtenido en Gnuplot a partir de los datos generados en Python.

Discusión

Tras haber generado la información de los datos y haberlos agrupado en sus respectivas listas, se procedió a integrarlos dentro de un marco de información con la finalidad de habilitar su manipulación mientras se conserva la congruencia interna de los datos; lo que quiere decir que, los datos generados para cada nodo i se mantuvieron en un solo bloque de información para poder realizar el ordenamiento.

Dado que la información fue creada de manera aleatoria, para proporcionarle un contexto hipotético a los datos, se supone la siguiente interpretación:

- Cada nodo representará una unidad formadora de colonia de una variedad de distintas cepas de la bacteria *Escherichia coli*.
- Se sabe que las coordenadas representan la posición de los nodos dentro de un plano, lo que puede ser interpretado como la posición de las colonias bacterianas en un cuadrante de un medio de crecimiento sólido.

- Si bien cada nodo tiene su respectivo tamaño, éste representará el tamaño o la densidad de dicha colonia bacteriana.
- Se supondrá que en el medio de crecimiento se encuentra una mezcla de tres sustratos cromogénicos que tienen influencia directa en la coloración de la colonia dependiendo las características de la cepa bacteriana. Por lo que entonces que el valor de color corresponderá al color observado de la colonia tras un crecimiento de 24 horas.

Teniendo establecido el previo escenario hipotético, se procedió a realizar la conexión entre nodos o colonias bacterianas en base a la interacción que presentaron con respecto a los sustratos. Debido a la probabilidad de mutaciones aleatorias, se supone que en éste escenario existe la posibilidad de obtener aleatoriamente cepas que procesen en diferente grado dos de los sustratos, generando de ésta manera un gradiente de color observable en las colonias bacterianas presentes en la muestra.

Por lo que dentro del grafo, cada nodo o colonia i se unirá con la colonia k que tenga la interacción de sustratos más cercana a la interacción de la colonia i. Esto se logró ordenando los datos de forma ascendente en base a la columna con los valores de color. De ésta manera sería posible obtener una distribuci'on del comportamiento de las distintas cepas dada la presencia del sustrato cromogénico.

Por último, es importante mencionar que para facilitar su visualización y comparación, se escalaron los valores del tamaño de cada nodo. Obteniendo así, un grafo que permite apreciar sin cansar la vista o ser abrumante, las interacciones de los componentes del sistema generado aleatoriamente.

Referencias

[1] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications.* Prentice Hall. ISBN: 013617549X.

[2] Barabási, A.-L. (2016). *Network science.* Cambridge University press. ISBN: 9781107076266.

[3] Lutz, M. (2008). *Learning Python.* O'Reilly, third edition. ISBN: 9780596513986.