Proyecto integrador de las materias de Inteligencia Artificial y Sistemas Expertos (Junio de 2020)

Calos Andrade
candradev3@est.ups.edu.ec
Jessica Rocano
jrocano@est.ups.edu.ec

Resumen - Dentro de este documento trataremos lo relacionado a bases de daos orientadas a grafos así como su implementación mediante el uso de neo4j implementando algoritmos con datos reales de ciudades del Ecuador, pudiendo comprender mejor el proceso y resultados de cada uno de ellos.

Índice de Términos – neo4j, Python, algoritmos,

I. INTRODUCCION

Este contiene la implementación de los diferentes algoritmos orientados a grafos mediante el uso de aplicación neo4j y jupyter notebook para poder comprender mejor el alcance de cada uno de ellos, así como su implementación, mediante el uso de datos reales.

II. DESAROLLO

A. Algoritmo de Centralidad

Los algoritmos de centralidad se utilizan para determinar la importancia de distintos nodos en una red de datos.

Page Rank (Rango de página)

INTRODUCCION:

El algoritmo de Page Rank mide la importancia de cada nodo dentro del grafo, en función del número de relaciones entrantes y la importancia de los nodos de origen correspondientes. Una página es tan importante como las páginas que enlazan con ella.

DESCRIPCION:

Page Rank se define en el documento original de Google como una función que resuelve la siguiente ecuación: PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

PROCESO:

Sintaxis:

Modo de escritura:

PageRank en modo de escritura en un gráfico almacenado en el catálogo.

```
CALL gds.pageRank.write(
graphName: String,
configuration: Map
 )
 YIELD
// general write return columns
ranIterations: Integer,
didConverge: Boolean
```

donde:

graphName: representa el nombre de un gráfico almacenado en el catálogo.

configuration: reprenta la configuración para algoritmos específicos y / o filtrado de gráficos.

Configuración específica:

se debe especificar los siguientes parámetros

dampingFactor: el valor de 0.85 por defecto,

factor de amortiguación del cálculo del rango de página.

maxIterations: el valor de 20 por defecto,

represnta número máximo de iteraciones de Page Rank para ejecutar.

tolerance: el valor de 0.0000001 por defecto,

representa el cambio mínimo en las puntuaciones entre iteraciones. Si todas las puntuaciones cambian menos que el valor de tolerancia, el resultado se considera estable y el algoritmo regresa.

ejemplo

Internet que consiste solo de tres páginas web y que están conectadas como muestra la figura, es decir, en la página 1

existen una liga para que se acceda a la página 2, de manera similar en la página 2 existe una liga para poder acceder a la página 1. Ocurre lo mismo para las páginas 2 y 3. Para las páginas 1 y 3, solo existe un botón en 3 para ir a 1.

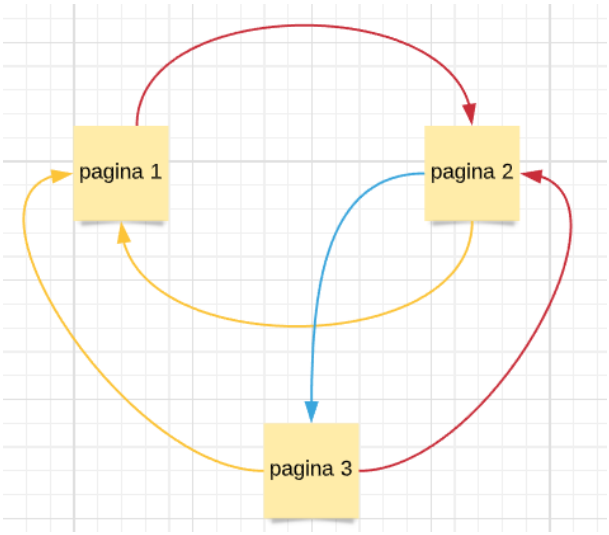


Fig. 1. Grafo de ejemplo 1

relationshipWeightProperty. el valor por defecto es null, de tipo opcional , permite especificar el nombre de la propiedad que contiene peso. Si es nulo, trata el gráfico como no ponderado. Debe ser numérico.

Contaremos los enlaces entrantes para cada una de las paginas
(Página 1) $x_1 = 2$
(Página 2) $x_2 = 2$
(Página 3) $x_3 = 1$

Pasamos los datos a una table donde llenaremos los enlaces de entrada que existe para X1, X2, X3 que son las paginas 1 2 3 corepondiente mente

pagina	entrada (página 1)	entrada (página 2)	entrada (página 3)
X1	0	X2	X3
X2	X1	0	X3
X3	0	X2	0

Dividimos para el numero de entradas en cada pagina.

pagina	entrada (página 1)	entrada (página 2)	entrada (página 3)
X1	0/2	1/2	1/2
X2	1/2	0/2	1/2
X3	0/1	1/1	0/1

Creamos nuestra matriz

$$\begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 1/2 \\ 0 & 1/2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Cálculo del PageRank

Para la página 1 Veamos que ocurre con el pagerank de la página 1,

De la página 2 Recibe 1/2 del pagerank de la 2, debido a que la página 2 tiene 2 salidas.

De la página 3 Recibe también 1/2 de la página 3 ya que tiene dos salidas.

Por lo tanto la página 1 tiene un pagerank de (1/2)(1/3) de 3 y (1/2)(1/3) de 2, después de la

primera transición, es decir un total de (1/2)(1/3) + (1/2)(1/3) = 1/3.

Para la página 2 Veamos que ocurre con el pagerank de la página 2,

De la página 1 Recibe 1 de esta página, debido a que está página solo tiene una salida.

De la página 3 Recibe 1/2 de esta página, debido a que está página tiene 2 salidas.

Por lo tanto la página 2 tiene un pagerank de (1/2)(1/3) de 1 y (1)(1/3) de 3, después de la primera

transición. Es decir un total de (1)(1/3) + (1/2)(1/3) = 1/2.

Para la página 3 Veamos que ocurre con el pagerank de la página 3,

De la página 1 No recibe nada de esta página.

De la página 2 Recibe (1/2), debido a que la página 2 tiene dos ligas de salida.

Por lo tanto la página 3 tiene un pagerank de (0)(1/3) de 1 y (1/2)(1/3) de 3, después de la primera

transición. Es decir un total de (1/2)(1/3) = 1/6.

Donde tenemos que el pague rank para cada uno de ellos es

$$PR = \begin{pmatrix} 1/3 \\ 1/2 \\ 1/6 \end{pmatrix}$$

B. Community detection algorithms

Los algoritmos de detección de comunidad se utilizan para evaluar cómo se agrupan o particionan los grupos de nodos, así como su tendencia a fortalecerse o separarse.

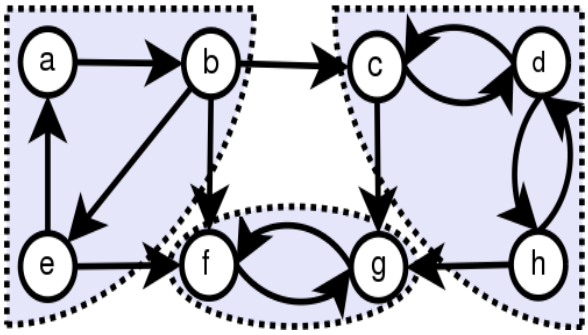


Fig2. Ejemplo de componentes fuerte mente conexos.

Componentes fuertemente conectados

los algoritmos de componentes fuertemente conectados permiten entender el comportamiento de los nodos que conforman un grafo dirigido y poseen un tipo específico de conectividad. Para que se considere un componente fuertemente conectado o conexo este grafo debe poseer nodos que estén totalmente conectados entre sí. Dentro de un grafo dirigido es posible que no exista una conexión fuerte, aunque es posible que exista una o varias vinculaciones. Se busca con estos algoritmos determinar los componentes fuertemente conectados de mayor tamaño.

Casos de uso

Entonces podríamos usar la similitud calculada como parte de una consulta de recomendación. Por ejemplo, puede usar el algoritmo de similitud Jaccard para mostrar los productos que fueron comprados por clientes similares, en términos de productos anteriores comprados.

Ejemplo

Primero explicaremos que es un grafo fuertemente conexos y débilmente conexos.

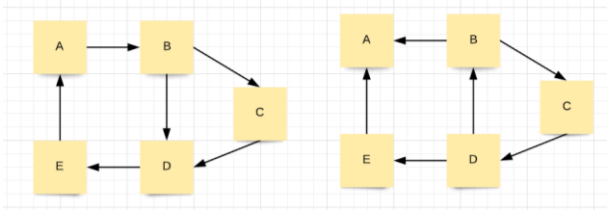


Fig3. Grafo fuertemente conexo y Debilmente Conexo.

Un grafo es fuertemente conexo si hay un camino entre cada par de vértices del mismo, caso contrario es un grafo débilmente conexo.

Los grafos fuertemente conexos nos permiten llegar a cada uno de los nodos y viceversa

ejemplo podemos observar que partiendo del nodo A podemos llegar a nodo C pasando por el nodo B.

cambiando el sentido ahora partiremos del nodo ce para llegar al nodo a pasando primero por el nodo D y luego por el nodo E llegando al nodo A.

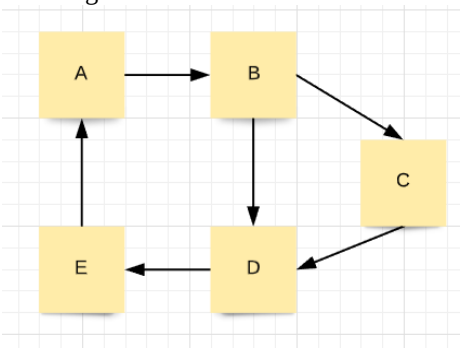


Fig4. Grafo fuertemente Conexo.

El siguiente grafo es débilmente conexo porque al intentar llegar al nodo A partiendo del nodo C nos permite llegar pasando por el nodo D y E.

Pero al intentar llegar al nodo C partiendo del nodo A no es posible por lo tanto es un grafo débilmente conexo

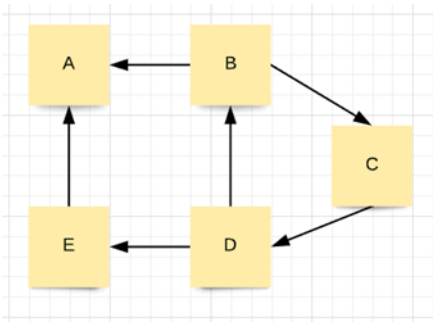


Fig5. Grafo debilmente Conexo

Componentes fuerte mente conexos

El algoritmo de componentes fuerte mente conexos nos permite dividir un grafo débilmente conexo en subgrafos los cuales serán fuerte mente conexos.

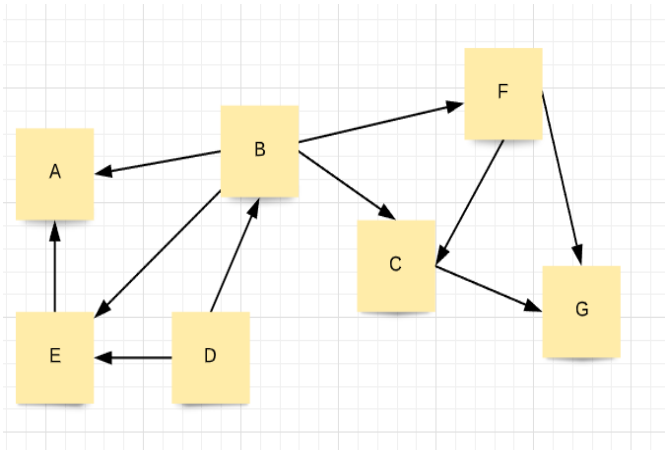


Fig6. Grafo debilmente conexo.

En este caso podemos identificar los componentes fuertemente conexos los cueles tiene los siguientes vértices

- {A, B, D, E}
- {C, F, G}

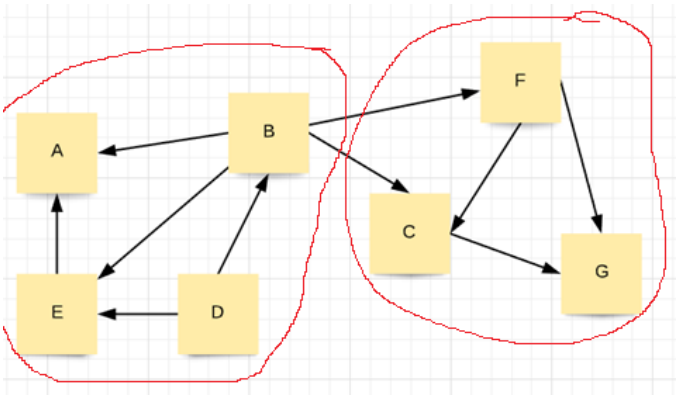


Fig7. Aplicando el algoritmo de Componentes fuerte mente conexos.

C. Similarity algorithms

Los algoritmos de similitud calculan la similitud de pares de nodos utilizando diferentes métricas basadas en vectores.

Los algoritmos de similaridad o algoritmos de similitud, son aquellos que ejecutan cálculos matemáticos para determinar los niveles de semejanzas o compatibilidades que existen entre conjuntos de datos.

Jaccard Similitud

Jaccard Similarity (coeficiente), un término acuñado por Paul Jaccard, mide las similitudes entre conjuntos. Se define como el tamaño de la intersección dividido por el tamaño de la unión de dos conjuntos.

Este se define como el tamaño de la intersección dividida por el tamaño de la unión de dos conjuntos. Este se rige bajo la siguiente formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Fig.8 formula para el calculo de la similitude de Jaccard

Casos de uso

- Puede usarse para encontrar el conjunto de empresas en las que cada miembro posee acciones directas y / o indirectas en todos los demás miembros.

- puede utilizar para calcular la conectividad de diferentes configuraciones de red al medir el rendimiento de enrutamiento en redes inalámbricas de múltiples tiendas.

- Pueden usar para encontrar dichos grupos y sugerir las páginas o juegos que comúnmente les gustan a las personas del grupo que aún no les han gustado esas páginas o juegos.

Ejemplo

Tenemos 5 personas con preferencias personales de comida

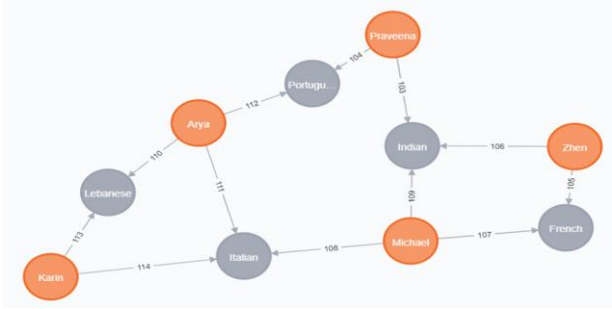


Fig8. Grafo ejemplo similitud de Jaccard

Siguiendo la siguiente formula vamos a obtener la similitud de Jaccard entre Karin y Arya

A Karin le gusta la comida Lebanese e Italian y Arya le gusta la comida Portugal, Italiana y lebanese

Karin={Lebanese, Italian}
Arya= {Portugual, Italiana,lebanese}

Implementado la siguiente formula tenemos que
J(A,B) = |A ∩ B| / |A| + |B| - |A ∩ B|

J(Karin, Aria)= |2|/|5|-|2|

J(Karin, Aria)= 2/3
J(Karin, Aria)= 0.6666
La similitud de Jaccard entre Karin y Aria es de 0.6666.

D. Link Prediction algorithms.

Los algoritmos de predicción de enlaces ayudan a determinar la cercanía de un par de nodos.
Parten en su mayoría de la hipótesis de que si un nodo posee varías conexiones, puede ser mayor la probabilidad de que este reciba enlaces adicionales. Estos algoritmos son de uso fundamental en el análisis de redes sociales y otros ámbitos tecnológicos.

Asignación de recursos

¿Cómo funciona?
Este algoritmo de asignación de recursos nos ofrece una medida con la que podemos estimar con precisión la cercanía entre vértices pertenecientes a un grafo, tomando como principal referencia la función de sus vecinos compartidos.

Fig9. Asignacion de recursos.

Su funcionamiento esta regido por la siguiente formula:

$$RA(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{|N(u)|}$$

Donde N(u) es el conjunto de nodos adyacentes a u.

Dentro de los resultados que podemos obtener con este algoritmo se tiene que los valores 0 indican que los nodos que son objeto de estudio no están cerca, mientras que en el caso de valores altos indican que los nodos están más cerca.

E. Path finding algorithms

Los algoritmos de similitud calculan la similitud de pares de nodos utilizando diferentes métricas basadas en vectores.
Los algoritmos de similaridad o algoritmos de similitud, son aquellos que ejecutan cálculos matemáticos para determinar los niveles de semejanzas o compatibilidades que existen entre conjuntos de datos.

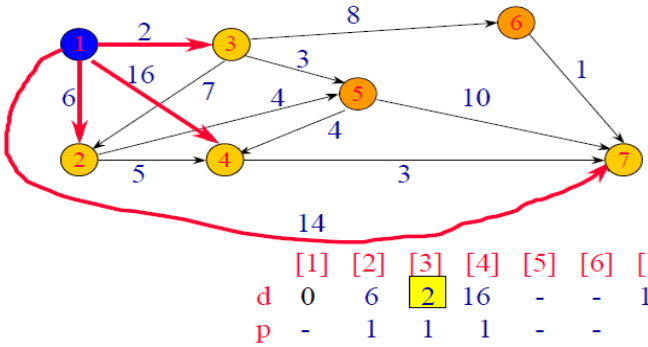
Historia y explicacion
SSSP se hizo prominente al mismo tiempo que el algoritmo de ruta más corta y el algoritmo de Dijkstra puede actuar como una implementación para ambos problemas.

Implementamos un algoritmo delta-stepping que se ha demostrado que supera al de Dijkstra.

Casos de uso
cuándo usar el algoritmo de ruta más corta de fuente única
Open Shortest Path First es un protocolo de enrutamiento para redes IP. Utiliza el algoritmo de Dijkstra para ayudar a detectar cambios en la topología, como fallas de enlaces, y crear una nueva estructura de enrutamiento en segundos .

Ejemplo

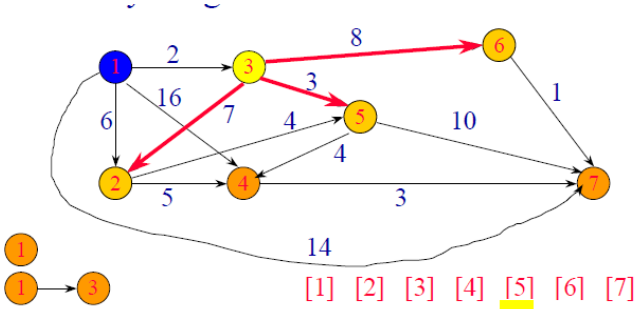
Necesitamos llegar del nodo 1 al nodo 7



	[1]	[2]	[3]	[4]	[5]	[6]	[7]
d	0	6	2	16	-	-	14
p	-	1	1	1	-	-	1

Fig10. Ruta mas corta de una sola fuente

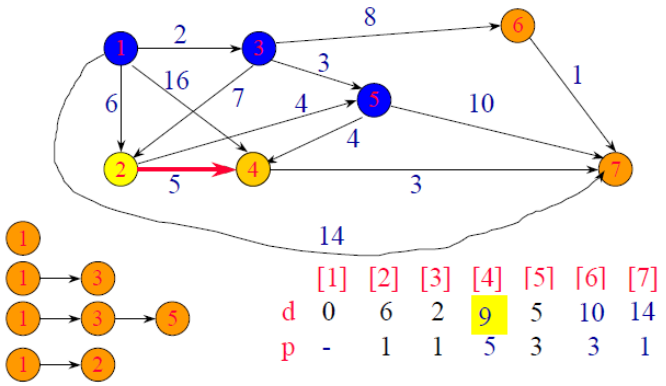
Segiremos la ruta de menos coste en este caso 2 que va al nodo 3



	[1]	[2]	[3]	[4]	[5]	[6]	[7]
d	0	6	2	16	5	10	14
p	-	1	1	1	3	3	1

Pig.11 Ruta mas corta de una sola fuente paso 2

Desde el nodo 3 podemos segir al nodo 5 que es el que tiene menor costo



Pig.13 Ruta mas corta de una sola fuente paso 3

Dado que el costo de ir del nodo 5 al nodo 4 es mayo regresamos para toma la ruta que pasa por el nodo 2.

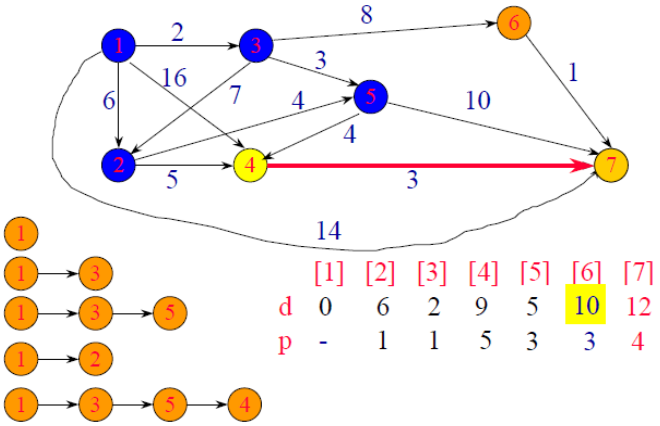


Fig.14 Ruta mas corta de una sola fuente paso 4

el costo de ir por el nodo dos al pasar por el nodo 4 aumenta asi que retomaremos el camino por el nodo 3

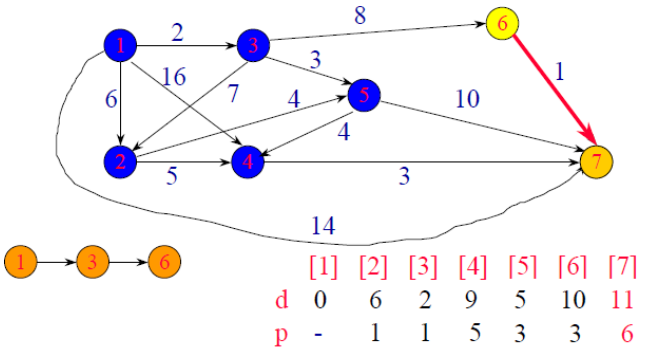


Fig.15 Ruta mas corta de una sola fuente paso 5

De nodo 6 es el de menor coste y por medio de el llegamos al nodo 7.

Descripción detallada del problema y su propuesta de solucion

III. CONCLUSIÓN

Mediante el uso de los grafos pondemos tener una major idea de como estan conformados los datos y mediante los cuales poder tomar decisiones que mejoraran del rendimiento y aprovechar major los recursos que disponemos.

REFERENCES

[1] G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.

[5] E. H. Miller, “A note on reflector arrays (Periodical style—Accepted for publication),” *IEEE Trans. Antennas Propagat.*, to be published.

[6] J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” *IEEE J. Quantum Electron.*, submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style),” *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982,