Memoria Mineria 3

Daniel Santo-Tomás y Juan Velasco

Mayo 2021

1. Ejercicios realizados

En esta última práctica hemos decidido realizar todos los ejercicios de ambos bloques. Cabe concretar que en el Ejercicio 3 hemos implementado kNN orientado a items y la similitud de Pearson para kNN orientados a usuarios, en el Ejercicio 4 hemos realizado el método de evaluación RMSE y en el Ejercicio 7 hemos implementado las métricas adicionales closeness y distancia promedio y se ha implementado una clase que crea un grafo del tipo *Amigos de Amigos*.

Vamos a desarrollar, si procede, qué algoritmos se han utilizado en los distintos ejercicios.

- **Ejercicio 1:** destacar que tanto en *MajorityRecommender* como en *AverageRecommender* hemos añadido un diccionario que guarda el valor del recomendador para cada item, ya que no depende del usuario, y así resulta más rápido para datasets más grandes.
- **Ejercicio 2:** en este ejercicio se han seguido las recomendaciones de los vecindarios offline y de el uso de la clase Ranking.
- **Ejercicio 3:** en este ejercicio no hay nada que destacar, ya que se ha aclarado anteriormente los elementos realizados.
- **Ejercicio 4:** en este ejercicio queremos realizar un comentario acerca de cómo se realiza la media de las medidas de evaluación individuales. Nos hemos fijado que estas se realizan teniendo en cuenta el número de elementos totales en el dataset original. Sin embargo puede no cumplirse que, al realizar las particiones de training y test, aparezcan representados todos los usuarios. Por tanto, para acercarnos lo máximo posible, tomamos el valor de usuario máximo de los que hay en la partición de test.
- **Ejercicio 5:** este ejercicio será desarrollado y explicado más adelante en su propio apartado.
- **Ejercicio 6:** en este ejercicio no hay nada que destacar.
- **Ejercicio 7:** dado que ya se ha aclarado anteriormente los elementos realizados, solo tenemos que destacar como hemos interpretado *Amigos*

de amigos. Nuestro algoritmo funciona de la siguiente manera: creamos todos los nodos al principio; para el primer nodo realizamos Erdös-Rényi simplemente y a partir del segundo nodo iteramos sobre todos los nodos, vemos si es vecino si lo es, lo tomamos con probabilidad q y realizamos el enlace con probabilidad p mientras que si no lo es, lo tomamos con probabilidad (1-q) y realizamos el enlace con probabilidad p.

2. Bloque 1. Recomendación

2.1. Ejercicio 4. Evaluación

Vamos a mostrar la tabla con las métricas de evaluación para cada uno de los algoritmos de predicción.

Métodos de Evaluación			
Método de predicción	*P@10*	*R@10*	*Tiempo de ejecución*
kNN usuario	0.16590	0.089339	15 segundos
kNN usuario normalizado (coseno)	0.103442	0.05624	16 segundos
kNN usuario normalizado (pearson)	0.09606	0.05225	13 segundos

No hemos incluido en la tabla *kNN Item* ya que nuestra versión tarda demasiado para el dataset **ml-latest-small**, aunque si que funciona como se muestra en el student test.

3. Bloque 2. Redes Sociales

3.1. Ejercicio 5. Preliminares

Hemos generado dos redes sociales siguiendo los modelos indicados. El grafo de Barabási-Albert tiene 50 nodos y el número de enlaces que se añade a cada paso es de 5.

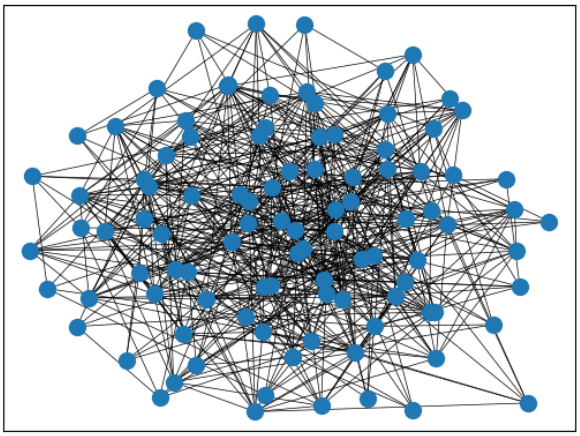

Figura 1: Red siguiendo el modelo de Barabási-Albert

El grafo de Erdös-Renyi tiene 50 nodos y la probabilidad de unión a cada paso es de 0.29

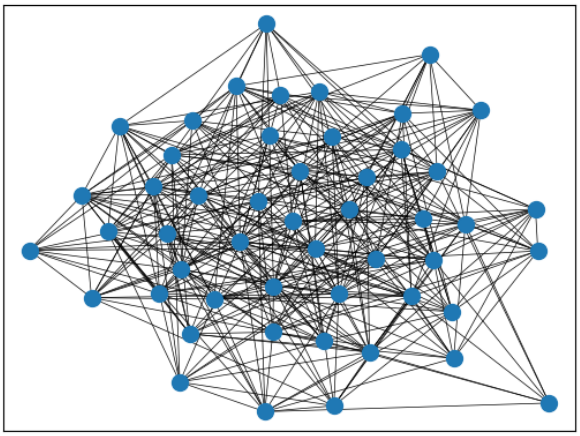

Figura 2: Red siguiendo el modelo de Erdös-Rényi

Tanto de estas dos redes como del resto indicadas por el enunciado (small x3 y Facebook), se han obtenido una serie de datos relativos a la distribución de

grado. Esta información se ve reflejada en las siguientes gráficas. Cabe destacar que la escala del eje y es logarítmica en todas las gráficas

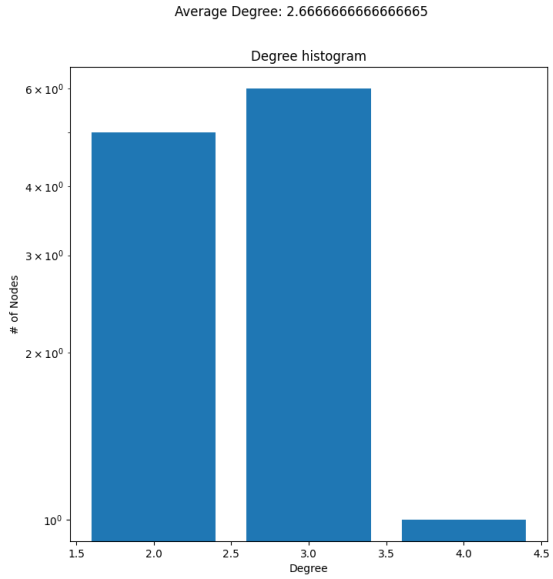

Figura 3: distribución de grado de small1

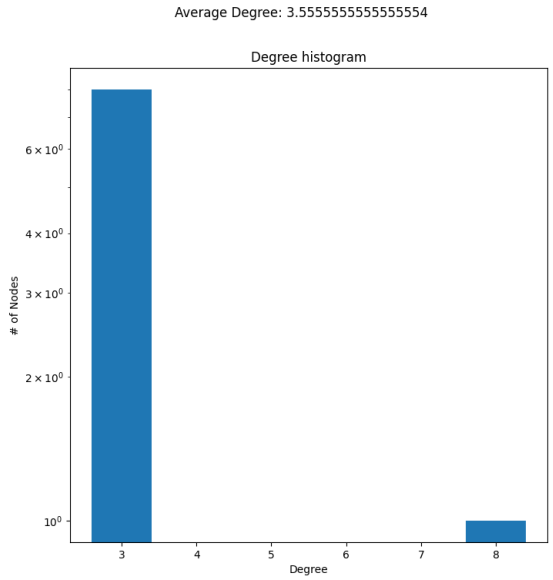

Figura 4: distribución de grado de small2

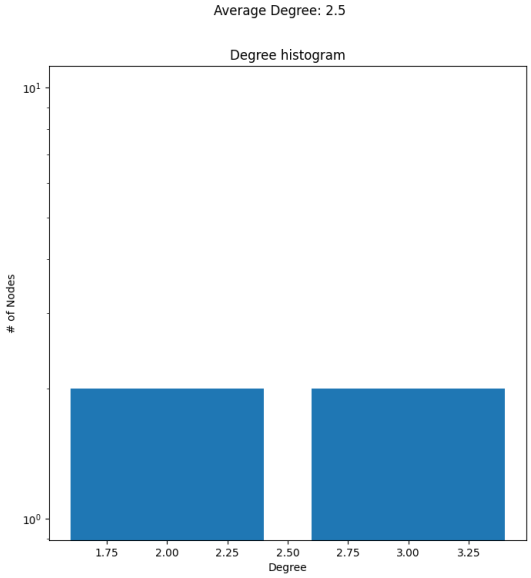

Figura 5: distribución de grado de small3

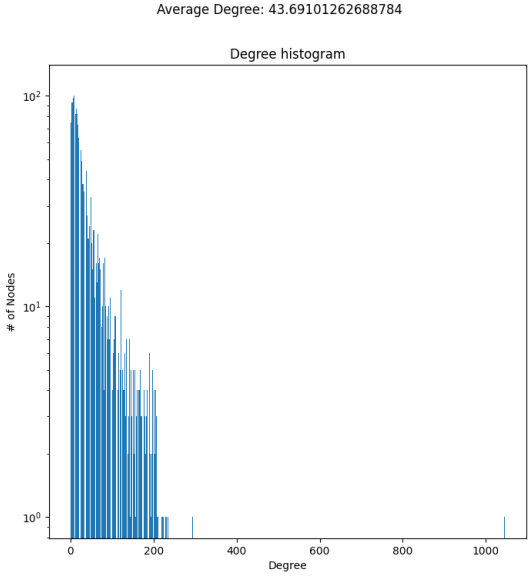

Figura 6: distribución de grado de Facebook

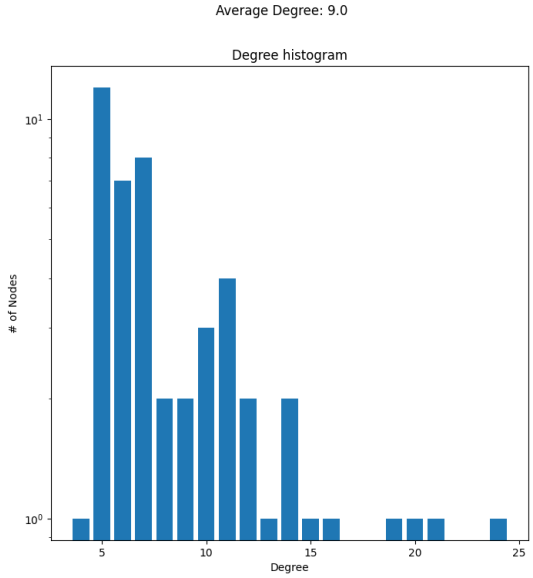

Figura 7: distribución de grado de Barabási-Albert

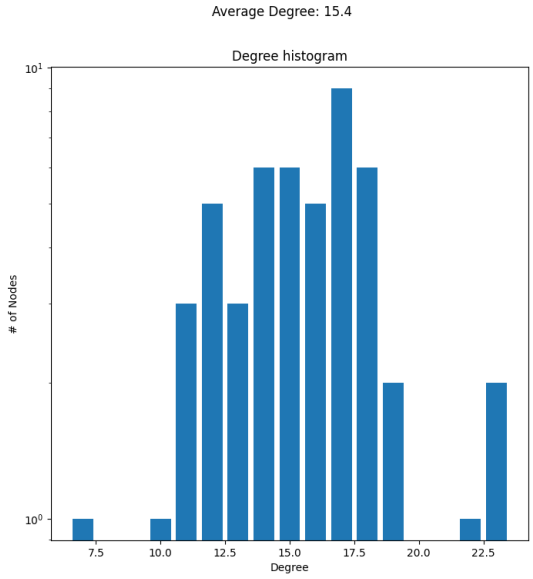

Figura 8: distribución de grado de Erdös-Rényi

Los gráficos de los grafos small no revelan nada interesante, puesto que el número de nodos es muy pequeño. Sin embargo, en redes más grandes como

Barabási-Albert y Facebook se puede observar que emerge una distribución power law: abundan los nodos con grados bajos frente a los nodos con grados altos. Sin embargo, en Erdös-Rényi , los grados toman sobre todo valores medio-altos.

Pasamos ahora a analizar si la paradoja de la amistad se cumple. En particular calcularemos el promedio del nº de amigos de los amigos de un nodo, y promediaremos este valor sobre todos los nodos. Esto debería ser siempre mayor que el grado medio. Además, comprobaremos si se cumple que la mediana de los grados es menor que la media.Esto último ocure si la distribución de grado es monótona decreciente, por lo tanto y con la información vista, deberia cumplirse para (minimo) Facebook y Barabási-Albert.

$$avg_u g(u) \leq avg_u avg_{v:u \to v} g(v)$$

$$median_u g(u) \leq avg_u g(u)$$

Mediante un sencillo programa, hemos obtenido los datos necesarios para las comparaciones expuestas. Estos se ven reflejados en la siguiente tabla:

Paradoja de la Amistad					
Grafos	$avg_u g(u)$	$avg_u avg_{v:u \to v} g(v)$	$median_u g(u)$	$avg_u g(u) \leq avg_u avg_{v:u \to v} g(v)$	$median_u g(u) \leq avg_u g(u)$
Small1	2.66666665	2.84722222	3	Verdadero	Falso
Small2	3.55555555	4.48148148	3	Verdadero	Verdadero
Small3	2.5	2.66666667	2.5	Verdadero	Verdadero
Facebook	43.69101263	105.55179301	25	Verdadero	Verdadero
Barabási-Albert	9	11.63979459	7	Verdadero	Verdadero
Erdös-Rényi	15.4	16.05864511	15.5	Verdadero	Falso

Como habiamos predicho, la fórmula de la mediana se cumple en Facebook y Barabási-Albert, además de en small2 y small3. En Erdös-Rényi no se cumple debido a que, como se indicó previamente, los grados no siguen una distribució monótona decreciente. En cuanto a la primera fórmula, tal y como sabíamos ya, se cumple para todos los grafos.

3.2. Ejercicio 6. Métricas

A continuación mostramos la tabla con los tiempos para los grafos de Twitter y Facebook.

Tiempos de ejecución de las métricas		
Métrica	*Facebook*	*Twitter*
Coef. clustering usuario	0:00:02	0:00:20
Embedeness	0:01:19	0:15:26
Coef. clustering global	0:00:04	0:01:34
Asortatividad	menor que 0:00:01	menor que 0:00:01

Como podemos ver asortatividad es la métrica más rápida de calcular seguido de los coeficientes de clústering de usuario y global y por último embedeness. Además vemos que el crecimiento de tiempo para cada métrica es exponencial sobre todo para embedeness, ya que aumenta mucho el tiempo de facebook a twitter.

3.3. Ejercicio 7. Ejercicio libre

Las pruebas que hemos desarrollado para las nuevas métricas consiste simplemente en utilizar las funciones de *sna-main.py* con los grafos small con las mismas y comprobar a mano (ya que son grafos pequeños) que son correctos sus valores.

Respecto al generador de grafos de Amigos de amigos las prueba que hemos realizado ha sido generar el grafo y usar el código de generación de gráficas del Ejercicio 5 para visualizarlo. Para terminar vamos a verlas a continuación.

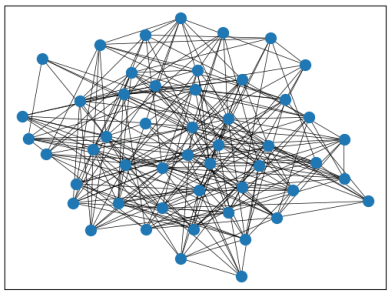

Figura 9: Red siguiendo el modelo de Amigos de Amigos

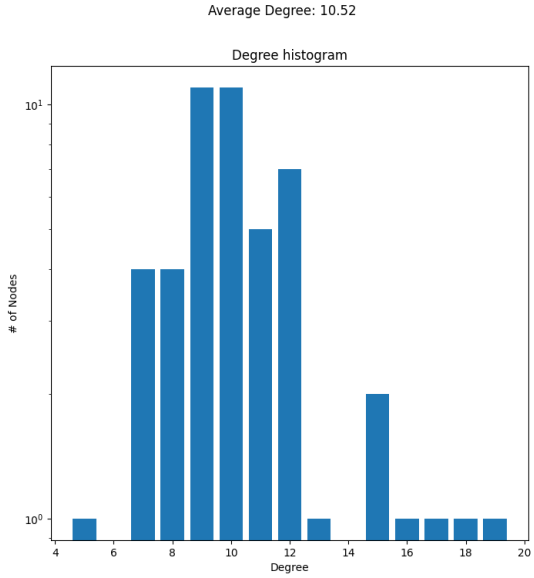

Figura 10: distribución de grado de Amigos de Amigos

4. Observaciones

Se ha incluido en la carpeta data el archivo **degree-distribution.py** en donde se encuentra la implementación para obtener las diferentes gráficas mostradas en la memoria.