



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE INFORMÁTICA  
*1er Semestre 2017*

---

TAREA II Y III - REDES COMPLEJAS  
*Prof. Andres Moreira*

---

*Autor:*  
Francisco Mena Toro  
201373504-5  
*San Joaquín*

*Autor:*  
Francisco Perez Castro  
201373516-9  
*San Joaquín*

## Desarrollo

Para el desarrollo de la tarea se trabajó con el lenguaje de programación Python, en conjunto con librerías que apoyaron el cálculo de ciertas funciones, **iGraph** para el manejo de grafos y una gran cantidad de operaciones sobre estos, **numpy** para el manejo de arreglos y **pandas** para lectura de algunos grafos.

1. Magister

2. Para un grafo completo no orientado de 400 nodos, se calcula cuántas aristas deben borrarse para que deje de existir una componente gigante.

Para el caso en que se tiene  $n$  nodos todos conectados con todos, es decir, con número de aristas igual a  $\frac{n(n-1)}{2}$ , una componente gigante existe cuando hay más de  $n/2$  nodos conectados entre sí. La componente gigante deja de existir cuando se reduce a la mitad de su tamaño, menor o igual a  $n/2$ . Con esto en mente se calcula cuántas aristas deben ser removidas escogidas al azar para que deje de existir la componente gigante, es decir, la componente gigante (que en comienzo es de tamaño  $n$ ) pase a la mitad de su tamaño.

Se toma el mejor caso en que aleatoriamente se escojan y se borren las aristas específicas para destruir la componente gigante, dejando dos componentes de tamaño  $n/2$  cada una. Para esto es necesario borrar todas las aristas entre esas 2 componentes.

Aristas a borrar = por cada nodo en componente 1 borrar todas sus aristas hacia componente 2

$$\text{Aristas a borrar} = \sum_i^{n/2} \frac{n}{2} = \frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4}$$

Por otro lado se tiene el peor caso, el cual es borrar todas las aristas y que la última sea la que hace que deje de existir la componente gigante, por lo que es necesario borrar todas las aristas dejando una componente gigante conectada con lo mínimo, cantidad de nodos  $n/2 + 1$  y aristas  $n/2$ , luego que la última arista que se borre sea la que hace que la componente ya no sea gigante.

Aristas a borrar = Aristas iniciales - aristas componente gigante + remover 1 arista

$$\text{Aristas a borrar} = \frac{n(n-1)}{2} - \frac{n}{2} + 1 = \frac{n(n-2)}{2} + 1$$

Para el caso dado en que la cantidad de nodos ( $n$ ) 400, la cantidad de aristas a borrar en el mejor caso es 40000 y para el peor caso es 79601, por lo que el caso promedio es de **59801** aristas a borrar. A razón de borrar una arista por segundo, el borrar las 59801 aristas tomaría 16 horas, 36 minutos y 41 segundos aproximadamente, por lo que si se comienzan a borrar aristas a las 00:00 horas, en promedio a las **16:37** horas dejaría de existir la componente gigante, siendo el mejor caso a las 11:07 horas y el peor caso a las 22:07 horas.

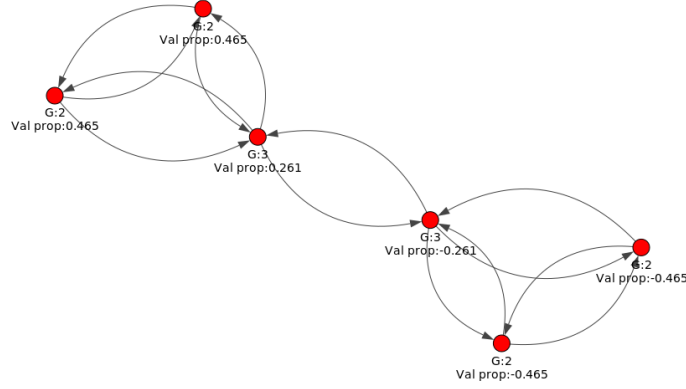
3. Para el grafo de 6 nodos entregado se calcula la matriz simétrica Laplaciana de la forma  $L = D - A$ :

$$L = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

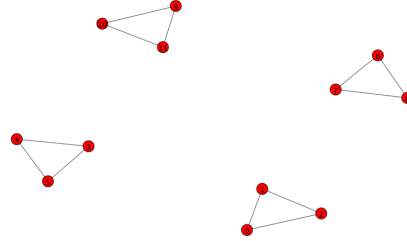
En base a esta matriz, se calculan los valores propios  $\lambda_i$  en conjunto con su vector propio asociado  $w_i$ . Se calcula el valor de Fiedler (segundo valor propio más pequeño)  $\lambda_2$  y con este un vector propio  $w_2$  con cada componente asociada a cada nodo:

$$\lambda_2 = 0,438 \quad \wedge \quad w_2 = (0,465, 0,465, 0,261, -0,261, -0,465, -0,465)$$

Se adjunta una imagen con la red en donde se puede ver a cada nodo asociado con su valor dentro del vector propio asociado al valor de Fiedler, en donde se puede ver la clara partición de la red en dos comunidades de 3 nodos cada una, donde la comunidad de la izquierda tiene asociado un valor en el vector propio positivo y la comunidad de la derecha tiene asociado un valor de vector propio negativo, donde resulta ser simétrico únicamente por la simetría de la red, además de ser bidireccional con arcos en cada una de las direcciones.



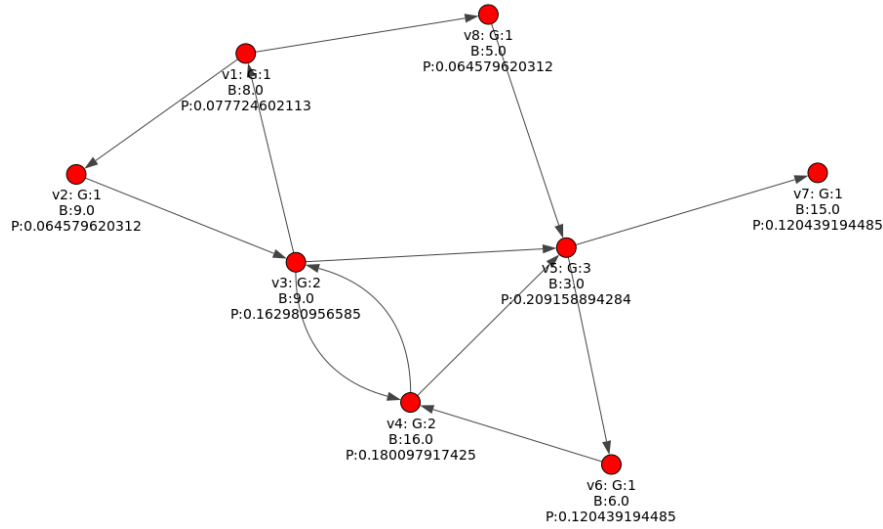
4. Dentro de un grafo o red, existe los conceptos de  $k$ -componente y  $k$ -core, los cuales hacen referencia a configuraciones totalmente distintas de la red. Por una parte una  $k$ -componente hace referencia a un conjunto de  $k$  nodos dentro del grafo que están conectados, existiendo un camino desde cualquier nodo de la componente a cualquier otro nodo de la componente. Por otro lado una  $k$ -core hace referencia a un conjunto de nodos en un grafo los cuales tienen grado como mínimo  $k$  dentro de la red. Esto se diferencia más claramente en un ejemplo de una red que tiene dos 2-componentes y una 2-core:



En esta red se puede ver que existen dos 2-componentes, los cuales son las 2 componentes superiores de 3 nodos cada una y las 2 componentes inferiores de 3 nodos cada una. Esta red posee un único 2-core ya que todos los nodos presentes tienen grado mínimo 2. Por lo que se puede ver que son conceptos totalmente distintos donde uno es sobre la cantidad de componentes (subredes conectadas) que posea la red principal y otro en la subred donde cada nodo posee a lo menos un grado mínimo.

5. Magister

6. Para esta pregunta, se utiliza el grafo dirigido "redchica.gdf", el cual se muestra a continuación:



Como se puede observar en la imagen, se muestra a los 8 nodos de la red con su respectiva etiqueta ( $V_x$ ), su grado de entrada ( $G$ ), su betweenness ( $B$ ) y su PageRank ( $P$ ).

A continuación, se confecciona un ranking de los nodos en función de cada uno de esos tres índices:

En base a grado entrada		En base a betweenness		En base a pagerank	
Nodo	Grado entrada	Nodo	Betweenness	Nodo	PageRank
v5	3	v4	16	v5	0.2091590
v3	2	v7	15	v4	0.1800980
v4	2	v2	9	v3	0.1629810
v1	1	v3	9	v6	0.1204390
v2	1	v1	8	v7	0.1204390
v6	1	v6	6	v1	0.0777246
v7	1	v8	5	v2	0.0645796
v8	1	v5	3	v8	0.0645796

Al analizar los ranking para cada índice se puede observar que existen correlaciones y divergencias entre estos.

Por ejemplo, una divergencia es el nodo  $V_5$  que posee el mayor grado de entrada con un valor de 3 y el mayor *Pagerank* con valor 0.21, esto en primera instancia hace pensar que es un nodo al que deben pasar varios caminos dentro de la red para llegar de un nodo a otro, ya que posee 3 arcos entrantes y se considera parte de un grupo bien cohesivo por su valor de *Pagerank*, pero esto no es así, puesto que su valor de *Betweenness* es el más bajo con un valor de 3. Lo que significa que al tomar dos nodos aleatorios de la red  $V_5$  no actúa con frecuencia como un puente a lo largo del camino más corto entre estos nodos.

Esto se puede explicar visualizando que  $V_5$  posee 2 enlaces de salida y 3 de entrada, pero los enlaces de salida llegan a nodos bastante "aislados" con respecto a la red general, por ejemplo, un enlace llega al nodo  $V_7$ , el cual solo posee un enlace de entrada y ninguno de salida, lo que quiere decir que llegando a ese nodo no se puede seguir recorriendo la red, y por lo tanto hay menos posibilidades para el nodo  $V_5$  para actuar como puente, ya que solo puede ser intermediario para pasar al nodo  $V_6$ .

Otra divergencia puede ser el nodo  $V_7$ , el cual posee un grado de entrada bajo con valor 1 y también un *Pagerank* bajo con un valor de 0.12, pero sorpresivamente posee el segundo valor más alto de *Betweenness* con un valor de 15, lo que quiere decir que en 15 ocasiones este nodo sirve como puente a lo largo del camino más corto entre dos nodos aleatorios. Esto es poco usual, ya que  $V_7$  no posee enlaces

de salida, lo que debería ser un impedimento para actuar como intermediario entre nodos, pero este valor puede explicarse al considerar el caso en que  $V_7$  sea su propio intermediario para llegar a él, en otras palabras, como solo posee un grado de entrada, el único intermediario a parte de  $V_5$  es el mismo nodo, lo que aumenta el número de *Betweenness*.

Los demás nodos poseen correlaciones esperables en los valores de los índices.

7. Para esta pregunta, se utiliza el grafo "*gnutella.gdf*", "*dolphins.gml*" y dos grafos *Erdos Renyi* generados con la misma cantidad de nodos y misma densidad de aristas.

Ambas redes originales son no orientadas; la primera es un fragmento de la red p2p Gnutella, mientras que la segunda corresponde a la red de interacciones sociales de una comunidad de delfines en Nueva Zelania.

La idea es realizar diferentes ataques a estas redes, ya sea atacando nodos al azar, atacando nodos en orden de grado decreciente o atacando nodos en orden de betweenness decreciente.

Para cada red y para cada modo de ataque, se determina el porcentaje de nodos que hace falta eliminar para que la componente gigante de su red caiga a la mitad de su tamaño inicial.

A continuación se presentan imágenes de cada grafo antes y después de cada uno de los ataques:

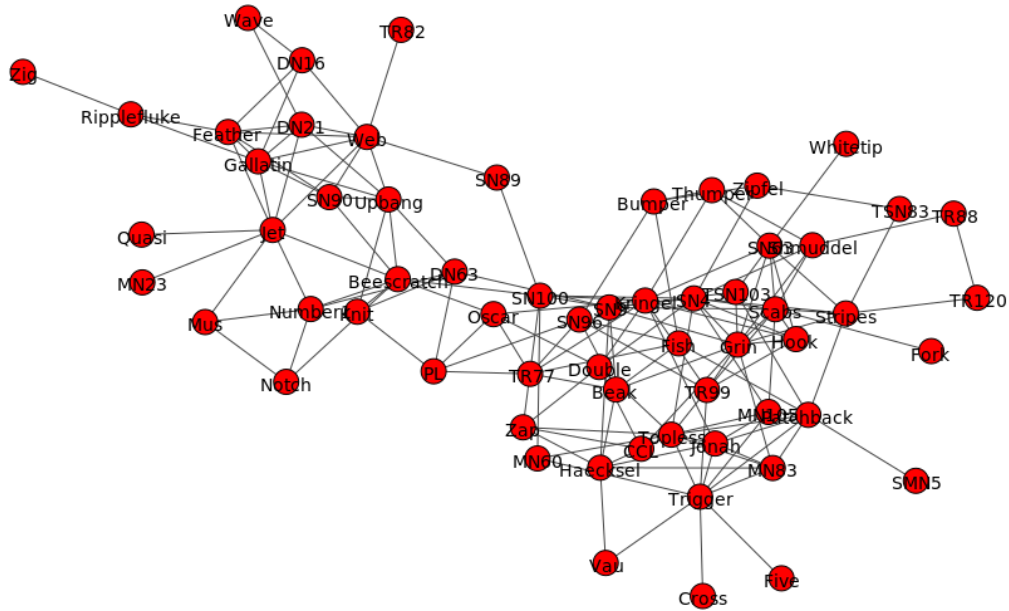
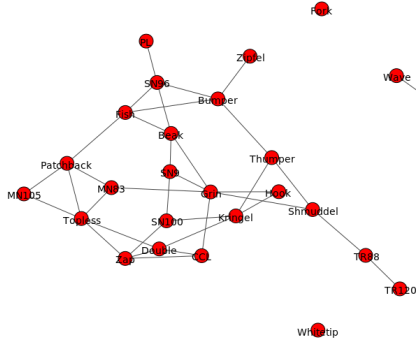
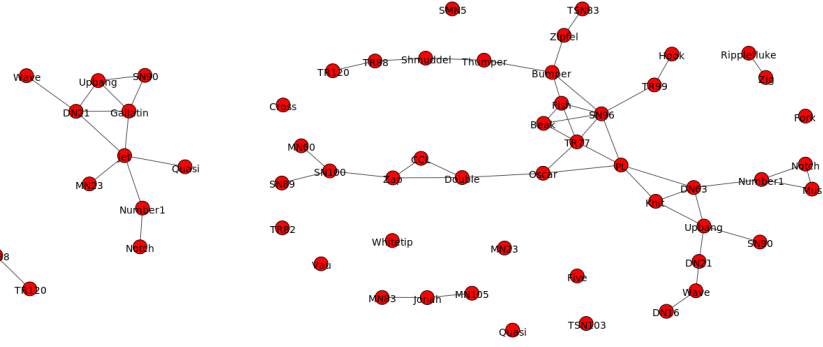


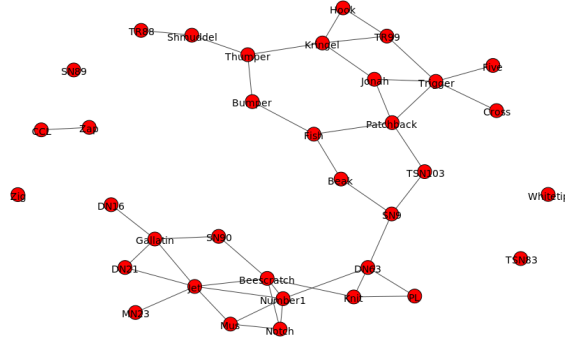
Figura 1: Grafo Dolphins (sin ataques)



(a) Ataque Random



(b) Ataque grados decrecientes



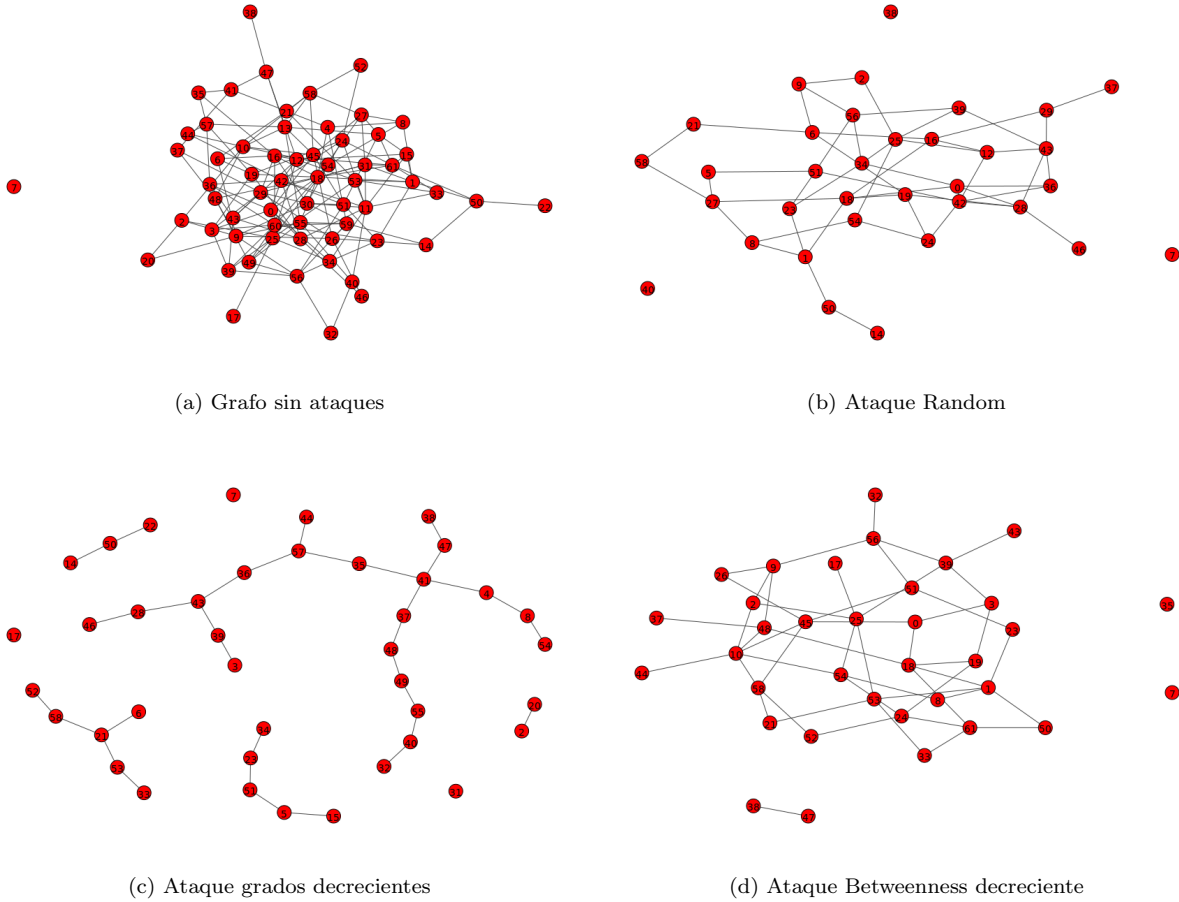
(c) Ataque Betweenness decreciente

Es fácil visualizar que cada tipo de ataque produce diferentes tipos de desconexiones, lo que implica diferentes tipos de componentes gigantes reducidos a su correspondiente mitad. Pero la gran diferencia se produce en la cantidad de eliminaciones que requiere hacer cada tipo de ataque para lograr esto. El número de eliminaciones y el porcentaje de nodos eliminados para lograr esto se resume en la tabla a continuación:

Ataque	Num. Eliminaciones	% nodos eliminados
RANDOM	27	43 %
GRADO DECRE.	16	25 %
BETW. DECRE.	27	43 %

Al analizar la tabla se puede observar que en este grafo los ataques *Random* y *Betweenness* decreciente realizan un número de eliminaciones idéntico, produciendo un porcentaje de nodos eliminados idéntico, con un valor de 27 eliminaciones y 43 % respectivamente. Esto puede traducirse en que un ataque guiado por el *Betweenness* no produce ninguna mejoría con respecto a atacar cualquier nodo arbitrariamente. El ataque guiado por grados decrecientes, en este caso, posee el número de eliminaciones más bajo, con un valor de 16, produciendo un porcentaje del 25 % de nodos eliminados. Esto quiere decir que un ataque guiado hacia los nodos con mayor grado produce una mayor impacto de desconexión en la red, dañando con más fuerza la componente gigante, y por lo tanto se requiere aproximadamente la mitad de eliminaciones con este ataque que con un ataque *Random*.

Para el grafo Erdos Renyi Dolphins:



Nuevamente se puede apreciar que cada tipo de ataque produce diferentes tipos de desconexiones, lo que implica diferentes tipos de componentes gigantes reducidos a su correspondiente mitad.

Pero la gran diferencia se produce en la cantidad de eliminaciones que requiere hacer cada tipo de ataque para lograr esto. El número de eliminaciones y el porcentaje de nodos eliminados para lograr esto se resume en la tabla a continuación:

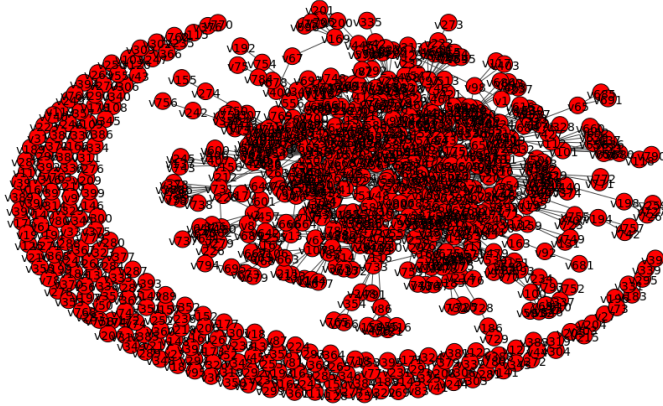
Ataque	Num. Eliminaciones	% nodos eliminados
RANDOM	28	45 %
GRADO DECRE.	24	38 %
BETW. DECRE.	27	43 %

Al analizar la tabla se puede observar que en este grafo los ataques *Random* y *Betweenness* decreciente realizan un número de eliminaciones muy similar entre sí, al igual que en el grafo real analizado anteriormente, produciendo un porcentaje de nodos eliminados similar, se obtiene un valor de 28 eliminaciones con un 45 % en el ataque *Random* y 27 eliminaciones con un 43 % respectivamente en el ataque con *Betweenness* decreciente. Esto nuevamente se puede traducir en que un ataque guiado por el *Betweenness* no produce ninguna mejoría con respecto a atacar cualquier nodo arbitrariamente.

El ataque guiado por grados decrecientes, en este caso particular, posee el número de eliminaciones más bajo, con un valor de 24, produciendo un porcentaje del 38 % de nodos eliminados, lo que no es muy diferente de los demás valores obtenidos por los demás índices, pero aún así es menor. Esto quiere

decir que un ataque guiado hacia los nodos con mayor grado produce una mayor desconexión en la red, al igual que el grafo real, dañando con más fuerza la componente gigante, y por lo tanto se requieren menos eliminaciones con este ataque que con un ataque arbitrario.

Para el grafo de Nutella, las conexiones de la componente gigante a medida que se van produciendo los ataques no son muy visibles en imágenes, debido a la gran cantidad de nodos en la red, la librería iGraph no logra situar visualmente correcta los nodos, por lo que a continuación solo se mostrará la red inicial antes de los ataques:



Como se puede observar en la imagen, este grafo es bastante denso, con muchos nodos muy conectados entre sí en el centro y muchos nodos medianamente conectados rodeando los anteriores.

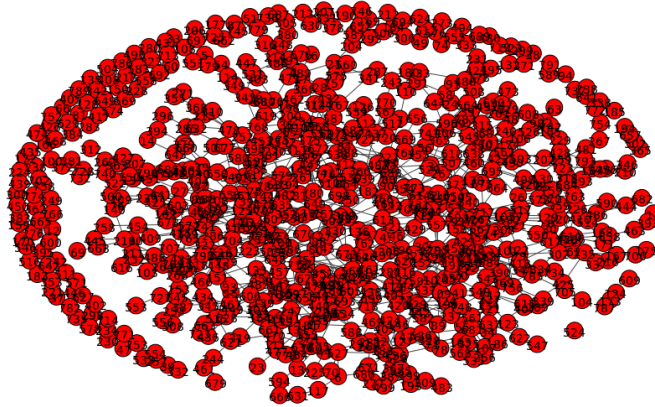
Ataque	Num. Eliminaciones	% nodos eliminados
RANDOM	136	17 %
GRADO DECRE.	14	1 %
BETW. DECRE.	189	23 %

Al analizar la tabla se puede observar que en este grafo los ataques *Random* y *Betweenness* decreciente nuevamente poseen el número de eliminaciones más altas para reducir a la mitad la componente gigante. En este caso se obtienen valores de 136 eliminaciones con un 17 % en el ataque *Random* y 189 eliminaciones con un 23 % respectivamente en el ataque con *Betweenness* decreciente. A diferencia de los demás grafos, el ataque guiado por *Betweenness* produce una mejoría significativa en comparación con un ataque arbitrario a cualquier nodo.

El ataque guiado por grados decrecientes, en este caso, es extremadamente eficiente en cuanto a la eliminación de la mitad de la componente gigante, ya que se realizan 14 eliminaciones, produciéndose un porcentaje del 1 % de nodos eliminados, es decir, solo se requieren eliminar 1 % de los nodos para reducir la componente gigante a la mitad. Esto quiere decir que un ataque guiado hacia los nodos con mayor grado produce un mayor impacto desconexión en la red, dañando con más fuerza la componente gigante, debido a que este grafo está fuertemente conectado a través de los nodos con alto grado y por lo tanto se requieren menos eliminaciones con este ataque que con un ataque arbitrario.



Para el grafo Erdos Renyi Nutella:



Como se puede observar en la imagen, este grafo también es bastante denso, muy parecido al original, ya que también posee muchos nodos muy conectados entre sí en el centro y muchos nodos medianamente conectados rodeando los anteriores.

Ataque	Num. Eliminaciones	% nodos eliminados
RANDOM	187	23 %
GRADO DECRE.	73	9 %
BETW. DECRE.	191	24 %

Al analizar la tabla se puede observar que en este grafo, al igual que los anteriores, los ataques *Random* y *Betweenness* decreciente nuevamente poseen el número de eliminaciones más altas para reducir a la mitad la componente gigante. En este caso se obtienen valores de 187 eliminaciones con un 23 % en el ataque *Random* y 191 eliminaciones con un 24 % respectivamente en el ataque con *Betweenness* decreciente. Se puede observar que el ataque guiado por *Betweenness* no produce una mejoría significativa como en el grafo real, por lo que se cree que la generación aleatoria de Erdos Renyi contrarresta el beneficio que se produce al realizar ataque en *Betweenness* decreciente.

El ataque guiado por grados decrecientes, en este caso, es eficiente en cuanto a la eliminación de la mitad de la componente gigante, ya que se realizan 74 eliminaciones, produciéndose un porcentaje del 9 % de nodos eliminados, si bien no es tan efectiva como el grafo real, igual es un porcentaje de eliminación bajo para la reducción de la componente gigante. Esto nuevamente implica que un ataque guiado hacia los nodos con mayor grado produce una mayor desconexión en la red, dañando con más fuerza la componente gigante, y por lo tanto se requieren menos eliminaciones con este ataque que con un ataque arbitrario.

En conclusión, se comprueba que el ataque más efectivo a realizar en un grafo cualquiera es un ataque por grado decreciente, ya que se genera un mayor impacto en la red eliminando una menor cantidad de nodos. Por otro lado, se visualiza que la generación aleatoria de Erdos Renyi disminuye la efectividad del ataque por *Betweenness* decreciente.

8. Para esta pregunta se trabaja con la red estudiada en la tarea 1, la cual corresponde a la red de transporte de carreteras en Chicago USA, la cual tiene 1467 nodos y 1298 aristas no dirigidas, con una densidad de 0.0012. Además de esta red se generan 3 nuevas red para trabajar y comparar resultados, una de ellas es a través de Erdos-Renyi, una Barabasi-Albert, las cuales mantienen la cantidad de nodos y la densidad de la red original. La tercera red que se agrega es una generada aleatorizando las conexiones (preservando la distribución de grados de la red).

- a) Lo primero en comparar son las estructuras  $k$ -cores de las 4 redes, resumiéndolas a continuación:

Red	0-core	1-core	2-core	3-core
Original	0	1467	0	0
Erdos Renyi	257	674	536	0
Barabasi Albert	0	1467	0	0
Original Aleatorizada	0	1340	33	94

En esta tabla se puede ver que en la red original todos los nodos pertenecen a la estructura 1-core, no existiendo nodos 2-core o superior. La única red que preserva esta estructura es la red generada Barabasi Albert y similar la red aleatorizada (con *rewire*) que tiene la gran mayoría de nodos en la estructura 1-core.

En la red con Erdos Renyi se puede ver que existen 257 nodos en la estructura 0-core, esto es debido a que la probabilidad que modela la red hizo que ese subconjunto de nodos no tengan conexiones.

- b) Otra métrica que se compara es la modularidad de las distintas redes, calculando la partición de la red a través del algoritmo *fast greedy* y luego se calcula la modularidad, valores resumidos en la siguiente tabla:

Red	Modularidad
Original	0.940
Erdos Renyi	0.847
Barabasi Albert	0.938
Original Aleatorizada	0.854

En estos valores se puede ver la clara partición de la red original, con un valor de modularidad cercano a 1, indicando que la partición generada por el algoritmo glotón es la correcta para la red, por lo que la red trabajada contiene comunidades. La modularidad de las otras redes generadas son muy parecidas a ésta, por lo que también existe la partición en comunidades generadas por el algoritmo glotón. El valor de modularidad que más se acerca al de la red original es el del modelo generador Barabasi Albert.

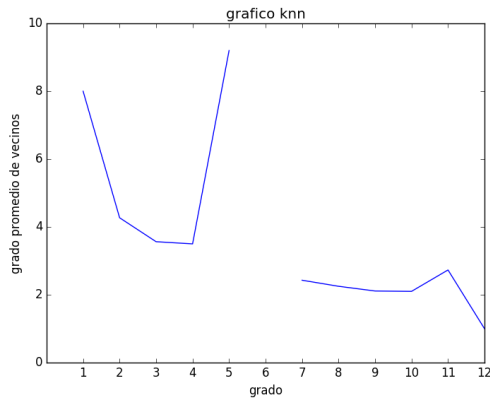
- c) Otra métrica a calcular es la asortatividad, la cual es para medir un sesgo a favor de las conexiones entre nodos similares, indicando una correlación lineal entre los grados de los nodos, mostrando a continuación:

Red	Asortatividad
Original	-0.505
Erdos Renyi	-0.022
Barabasi Albert	-0.089
Original Aleatorizada	-0.028

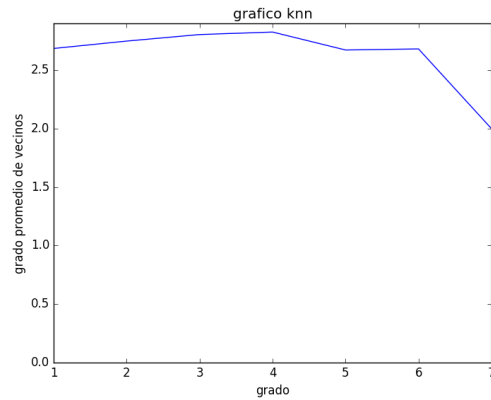
En donde se puede ver que todos los valores al ser negativos, indica que los nodos no son selectivos, por lo que no se forman los *club de ricos*. Para la red original es un valor negativo bastante alto negativamente por lo que se puede ver que son anti selectivos, es decir, un nodo de grado alto prefiere un nodo de grado bajo, esto es lo lógico en la red trabajada ya que deben existir pocos puntos que sean las conexiones en la red de transporte, por otro lado las redes generadas tienen una asortatividad cercana a cero por lo que no hay una tendencia.

Otra forma de visualizar la asortatividad es el gráfico *knn*, donde para cada grado de nodos se calcula el promedio de grado a la cual los nodos están conectados, mostrando un gráfico para cada red generada y la original.

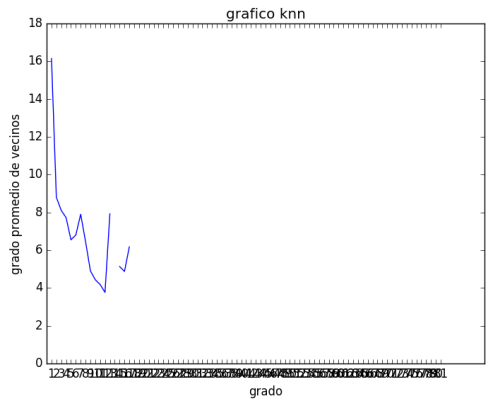
Estos gráficos soportan el análisis anterior de que no existe una tendencia entre los grados de los nodos en las redes generadas. Por otro lado para la red original se puede ver (exceptuando el grado



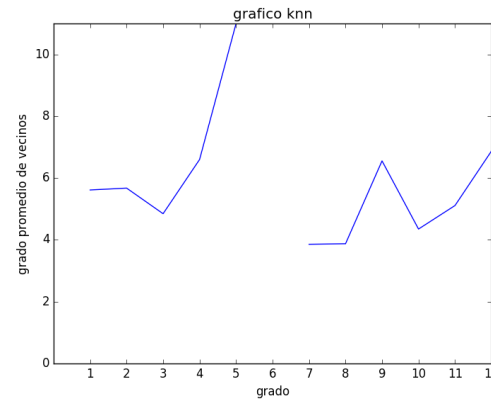
(a) Red original



(b) Red Erdos Renyi



(c) Red Barabasi Albert



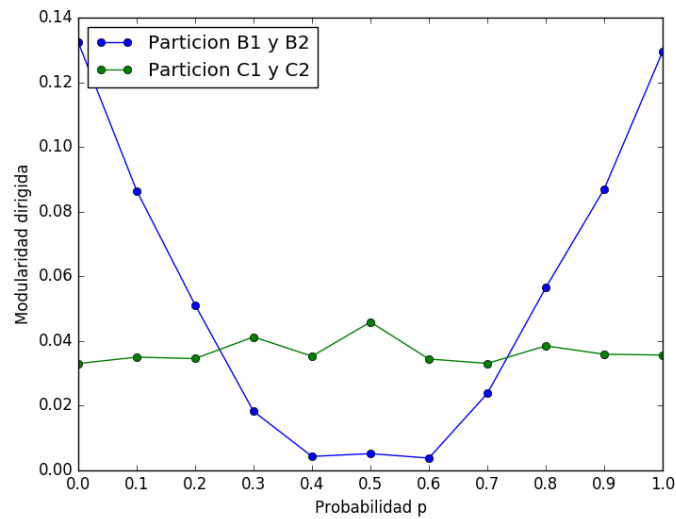
(d) Red aleatorizada

5 y 6) la tendencia descendiente en el grafo, donde los nodos de grado bajo se conectan (tienen vecinos) a nodos de grado mayor, o que los nodos de grado alto tienen vecinos a nodos de grado bajo, apoyando que los nodos son anti selectivos, prefiriendo nodos de grado distinto al que posean.

## 9. Magister

- Se genera un grafo Erdos Renyi con 80 nodos y probabilidad de conexión 0.2, denotando dos candidatos a particiones (B1,B2) y (C1,C2). Se transforma en un grafo dirigido asignando aleatoriamente cual punta es cual y para probar la calidad de la partición B sobre la C se asigna una probabilidad distinta para orientar la punta desde la partición B1 hacia la B2, probando distintos valores de probabilidad, calculando la modularidad dirigida, repitiendo y promediando para presentar los resultados a continuación.

Se presenta un gráfico resumen de la modularidad vs la probabilidad:



En esto se puede ver que para cada probabilidad de asignar un arco desde B1 a B2, la partición (C1,C2) tiene un valor de modularidad dirigida sin variar mucho y bastante bajo, por lo que esto muestra que esta partición no es la correcta para comunidades en la red y que al variar la probabilidad no afecta el valor de particionar en base a esto. Por otro lado al variar la probabilidad afecta la modularidad de la partición (B1,B2) disminuyendo hasta con probabilidad 0.4 y aumentando desde probabilidad 0.6. Esto indica que con valores de probabilidad chicos va a generar la mayor cantidad de arcos desde B2 a B1 y con probabilidad altos va a generar arcos desde B1 a B2, obteniendo valores más altos de modularidad que todo el resto, mostrando la clara partición en comunidades con estas particiones candidatas. Para la partición (B1,B2) con probabilidad 0, 0.1,0.2, 0.8,0.9 y 1 se obtiene una modularidad mayor que la partición (C1,C2), por lo que con esto se puede ver que los nodos en el conjunto B1 son similares en base a las conexiones entre sí, lo mismo para el conjunto B2, comparados con C1 Y C2 que podrían ser considerados como una partición aleatoria.

11. En esta pregunta se trabaja con el grafo "*pescado.gdf*" la cual es una red dirigida y con pesos que representa las exportaciones de pescado de un país a otro en 1998. Este grafo posee  $m = 2744$  arcos y  $n = 151$  nodos, lo que obviamente demuestra que posee pocos nodos que están fuertemente conectados entre sí.

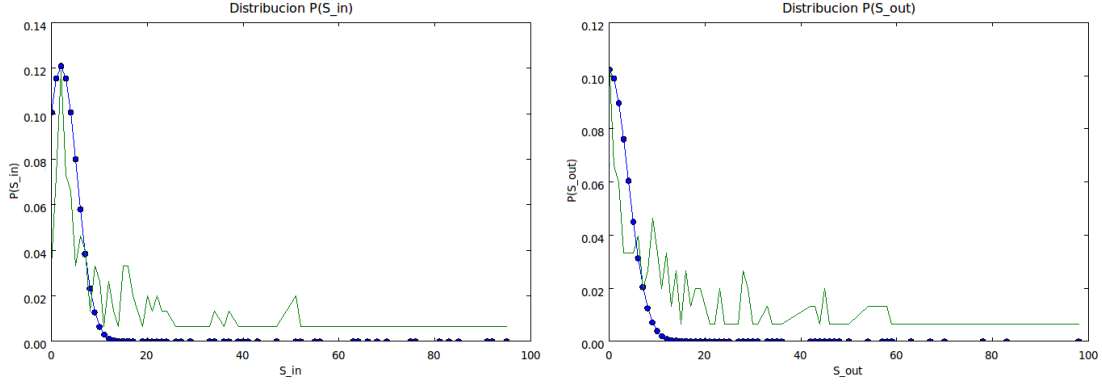
a) A continuación, se obtiene la matriz de adyacencia de la red y se utiliza para obtener la reciprocidad corregida de esta, a través de la formulación vista en clases, obteniendo los siguientes valores:

- Número de vértices: 151
- Número de arcos: 2744
- Reciprocidad: 0.467201166
- Reciprocidad Corregida: 0.393755974

Al obtenerse un valor mayor a cero en el valor de la reciprocidad corregida, se concluye que **existe reciprocidad**, es decir, en general, si existe un enlace de un nodo hacia otro, existe también un enlace en el sentido contrario.

b) Magister

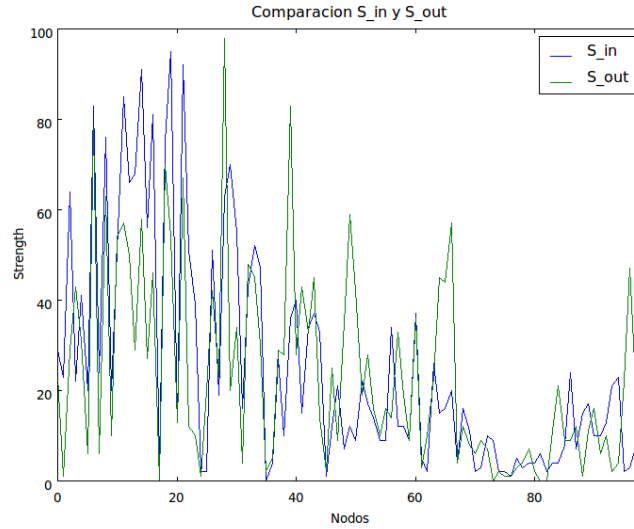
- c) Siendo  $S$  la suma total de pesos de los arcos que llegan y salen de los nodos, se grafican las distribuciones  $P(S_{in})$  y  $P(S_{out})$ :



Los gráficos presentados poseen una distribución normal conocida, por lo que se estiman los siguientes parámetros aproximados:

- $\mu_{in}$ : 2.0
- $\mu_{out}$ : 0.0
- $\sigma_{in}$ : 27.8327661103
- $\sigma_{out}$ : 23.0510291874

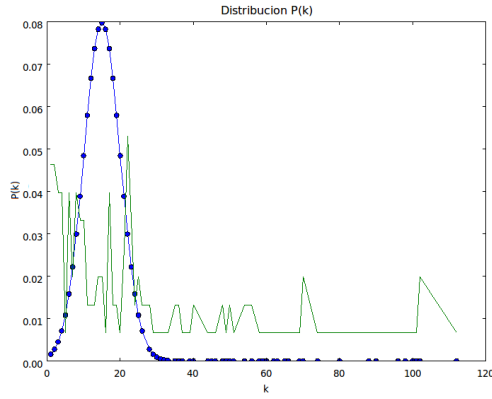
- d) En esta parte se grafican los *Strength* de entrada y salida con respecto a los nodos del grafo, es decir, se compara  $S_{in}$  con  $S_{out}$ :



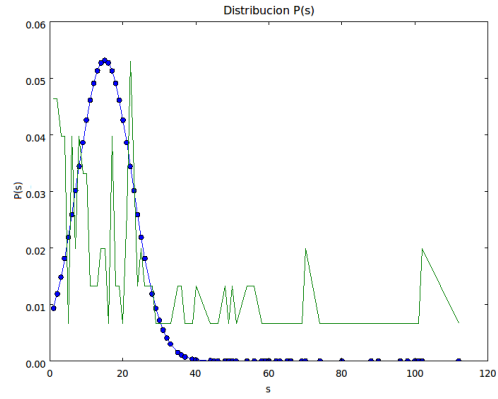
Claramente entre estas dos variables existe una relación muy fuerte, y es que son opuestas, es decir, cuando se produce un valor de  $S_{in}$  alto se produce un valor de  $S_{out}$  bajo, lo cual es lógico, puesto que estamos hablando de sumas de los pesos de cada nodo y ya se ha comprobado que existe una gran tendencia a la reciprocidad en la red.

Ahora se convierte la red en una red no dirigida, posicionando una arista entre cada par de nodos que estaban conectados por uno o dos arcos, y se le asocia como peso la suma de los pesos de esos arcos. De ahora en adelante, se trabaja con esta nueva versión de la red.

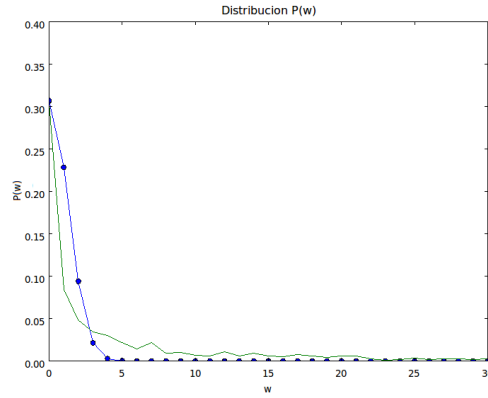
- e) Se evalúa  $P(k)$ ,  $P(s)$  y  $P(w)$  para esta nueva versión de la red, y se grafican junto a la distribución normal con parámetros estimados respectiva:



(a) Distribución  $P(k)$



(b) Distribución  $P(s)$



(c) Distribución  $P(w)$

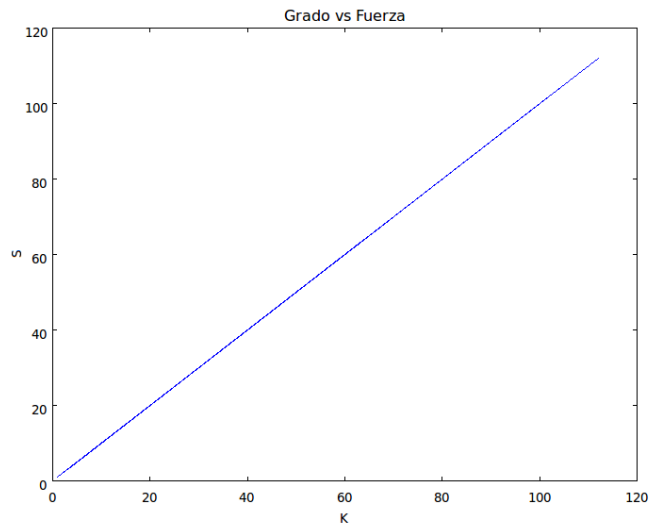
Para las distintas funciones de probabilidades para *degree* ( $k$ ), *strength* ( $s$ ) y *weight* ( $w$ ), se presentan sus parámetros estimados para la distribución normal:

- $\mu_k$ : 15.0
- $\mu_s$ : 15.0
- $\mu_w$ : 0.0
- $\sigma_k$ : 5.0
- $\sigma_s$ : 7.5
- $\sigma_w$ : 1.3

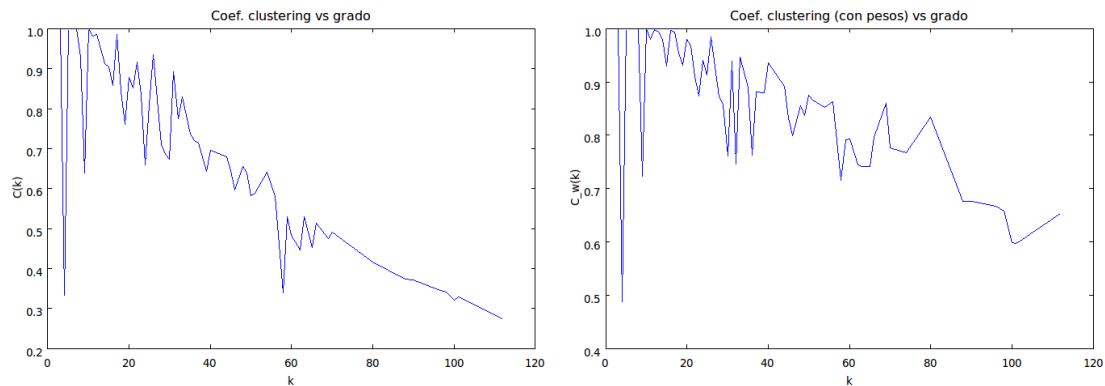
En estos valores se puede ver que la probabilidad de grados y la fuerza tienen la misma media, lo cual también se aprecia en el gráfico, variando en cuanto varían los datos respecto a su media (desviación estándar). Por otro lado la probabilidad sobre los pesos tiene media 0, como se puede ver en el gráfico.

- f) En esta pregunta se grafica el grado versus la fuerza de los nodos:

Como se puede observar, la relación es perfectamente lineal, es decir, el grado de cada nodo influye directamente en la fuerza de estos. Esto es esperable, ya que por definición la fuerza se plantea como la suma de los pesos de las aristas de cada nodo, es decir, para una mayor cantidad de enlaces en un nodo, se produce una mayor cantidad de fuerza.



- g) En esta pregunta se evalúa el coeficiente de clustering de la red, con y sin pesos, además de estudiarse las dependencias de este coeficiente con respecto al grado  $k$ :



Como se puede observar en la comparación anterior, en ambas versiones el coeficiente de clustering tiende a disminuir su valor a medida que se tiene un mayor grado ( $k$ ), lo cual es esperado ya que tener un mayor grado implica tener más conexiones con otros nodos, lo que tiende a disminuir el grado de agrupación con sus vecinos, y se requerirían muchas más conexiones para mantener un grado de agrupación (Coeficiente de clustering) constante.

También, se puede observar, que el gráfico que posee pesos tiene valores del coeficiente de clustering más altos, lo que hace pensar que los pesos actúan como una forma de interconexión más fuerte entre los vecinos de cada nodo.

12. Para esta sección se trabaja con la red "*epinion.net*", la cual consta de 131827 nodos, representando a personas, y 711782 arcos los cuales tienen pesos de la forma +1 y -1, simbolizando relaciones de confianza, donde +1 simboliza que esas dos personas confían entre sí, por otro lado -1 simboliza que dos personas desconfían entre sí, mostrando un extracto de la red (sample) aleatorio.

Mostrando visualmente el tipo de grafo que es, donde las aristas verdes representa a las relaciones de confianza y las aristas rojas representan relaciones de desconfianza, pudiendo ver que existen personas (nodos) que se desconfían entre sí y otros subgrupos que confían entre sí.

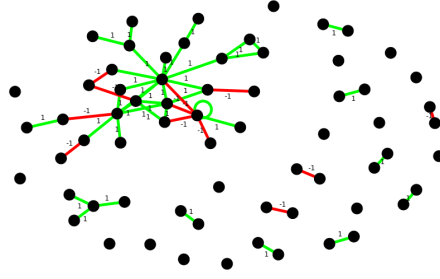


Figura 2: Subgrafo aleatorio de red epinion

- a) Se determinan la cantidad de triángulos en la red, de la forma  $t_0, t_1, t_2, t_3$ , simbolizando en el subíndice  $i$  de  $t_i$  la cantidad de  $+1$  en el triángulo. Pudiendo ver en el subgrafo anterior algunos de estos tipos de triángulos.

- Triángulos tipo  $t_0$  son 58732
- Triángulos tipo  $t_1$  son 396548
- Triángulos tipo  $t_2$  son 451711
- Triángulos tipo  $t_3$  son 4003085

Se puede ver que existe una gran cantidad de triángulos donde los 3 nodos tienen relación de confianza entre ellos, siendo el número mucho mayor que la cantidad de nodos y de aristas en la red. Se puede ver una tendencia de que existe una mayor cantidad de triángulos con relaciones más positivas, siendo mayoritaria las relaciones positivas en la red.

- b) La cantidad total de aristas  $m$  es de 711782, la cantidad total de aristas positivas  $m_+$  es 592591, y la tasa entre estas, es decir, la probabilidad  $p$  de escoger una arista  $+1$  dentro de todas las aristas del grafo es de 0.83, pudiendo ver que la cantidad de relaciones de confianza en el grafo es bastante alta, por lo que existe una tendencia de que la mayor cantidad de personas que tienen relación, confían entre sí.
- c) La cantidad de aristas positivas  $m_+$  es distinto a la la suma de los triángulos, según la fórmula:

$$m_+ \neq 0 \cdot t_0 + 1 \cdot t_1 + 2 \cdot t_2 + 3 \cdot t_3$$

Esto es ya que la parte derecha de la ecuación es aproximadamente 5 millones y la parte derecha son alrededor de 600 mil. Esta desigualdad es ya que dentro de los triángulos se puede dar el caso en que se cuente 2 o más veces una arista positiva (con etiqueta  $+1$ ), ya que existen triángulos "compartidos", es decir, podrían estar sobrepuestos compartiendo alguna arista con una etiqueta  $+1$ .

- d) Con la probabilidad de seleccionar una arista positiva ( $p = 0.83$ ) se calcula qué tan probable es el asignar pesos al azar a un triángulo basado en estas probabilidades, con probabilidad  $p$  asignarle una arista  $+1$  y con probabilidad  $1 - p$  asignarle una arista  $-1$ , definido a continuación:

$$t_0 = (1 - p) \cdot (1 - p) \cdot (1 - p) = 0,005$$

$$t_1 = (1 - p) \cdot (1 - p) \cdot p = 0,024$$

$$t_2 = (1 - p) \cdot p \cdot p = 0,117$$

$$t_3 = p \cdot p \cdot p = 0,572$$

Donde  $t_0$  es la probabilidad de que las 3 aristas en el triángulo se asignen  $-1$ ,  $t_1$  es la probabilidad de que a 1 arista únicamente se le asigne  $+1$ , y así hasta  $t_3$ , donde es la probabilidad de que a las 3 aristas en el triángulo se asignen  $+1$  en base a la probabilidad  $p$ . Con estos valores se ve que



en la red trabajada es más probable que existen triángulos de tipo  $t_3$  y lo menos probable es que ocurran triángulos de tipo  $t_0$  esto es ya que la probabilidad de asignar una arista  $+1$  es muy alta debido a que la red tiene muchas aristas positivas  $+1$ . Dicho de otra forma, se observa que en esta red es más probable que un grupo de 3 personas que se relacionan entre sí se tengan confianza entre ellos, y que lo menos probable es que 3 personas que se relacionan entre sí es porque se desconfíen entre ellos, viendo en base a las probabilidades que en un grupo de 3 personas lo más común es que exista por lo menos dos personas que confíen entre ellas.

- e) El resultado obtenido en a) se concuerda con las probabilidades obtenidas en b), ya que estas probabilidades dependen de la cantidad de aristas positivas al igual que la cantidad de triángulos positivos. Esto muestra que lo observado concuerda con un modelo teórico basado en probabilidades, donde ambos dependen del número de aristas positivas y negativas, siendo mayoritaria las positivas (relaciones de confianza en la red), pudiendo observar este comportamiento.

### 13. Magister