DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP 2

Redes Sociales

27 de octubre de 2022 — Métodos Numéricos

12

Integrante	LU	Correo electrónico
Damburiarena, Gabriel	889/19	gabriel.damburiarena@gmail.com
Guastella, Mariano	888/19	marianoguastella@gmail.com
Silva, Ignacio Tomas	410/19	ignaciotomas.silva622@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

1. Introducción

Las redes sociales son estructuras formadas en la web y son una gran herramienta para la comunicación y el intercambio de ideas entre usuarios, empresas, etc. En este informe vamos a analizar 2 redes particulares con el fin de caracterizarlas y entender sus estructuras.

Figura 1: Esquema de una Red

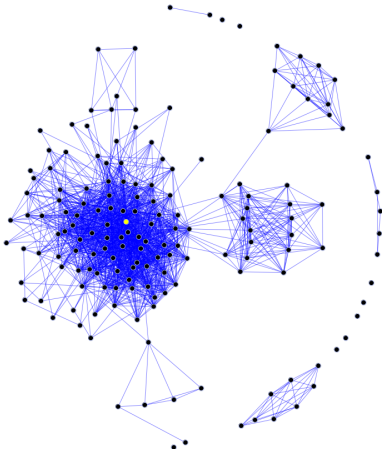

Figura 2: Grafo asociado a una red

La primer red es la red del Club de Karate la cual es una red social de un club universitario. Esta red fue estudiada por 3 años y tiene 34 nodos, que representan a los miembros del club, y aristas, las cuales son las interacciones entre los miembros fuera del club. Durante el período del estudio hubo un conflicto entre el instructor y el administrador, el cual llevó a que el grupo se divida en 2. Sabiendo en que parte de la división quedó cada uno vamos a estudiar la influencia de los miembros y ver si podemos predecir el resultado de la separación.

La segunda red es la del EgoFacebook. Esta red esta constituida por nodos que representan las amistades de un usuario, sin incluir a este último. Las aristas representan la amistad entre nodos. También contamos con una matriz de atributos donde hay un dato para cada nodo y una serie de atributos relacionados a categorías como trabajo, instituto educativo, ciudad natal, etc. Esperamos ver si los nodos están conectados a partir de sus atributos compartidos. Para esto vamos a computar la matriz de correlación o covarianza de los datos y establecer que dos nodos se conectan si superan un cierto valor.

Para que se pueda seguir el desarrollo del mismo, vamos a enunciar ciertas definiciones.

1.1. Definiciones

Vamos a modelar las redes con matrices de adyacencia, en la que la conexión entre nodos se representará con un 1 y con 0 la ausencia de la misma. Vamos a llamar autovalores y autovectores asociados a los grafos de cada red a los que obtengamos de sus respectivas matrices de adyacencia. La definición de autovectores y autovalores esta dada por la ecuación

$$\mathbf{A}v = \lambda v \tag{1}$$

donde $\mathbf{A}$ es una matriz cuadrada $\in \mathbb{R}^{n\times n}$, λ siendo un escalar (autovalor) y v un vector (autovector). Vamos a usar estos últimos para descomponer las matrices y así poder analizar las redes.

También utilizaremos la matriz Laplaciana la cual esta dada por la ecuación

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{2}$$

donde D es una matriz diagonal cuyos valores indican la cantidad de conexiones que tiene cada nodo. Para calcular esto, sumaremos las filas de la matriz de adyacencia previamente enunciada. Descomponer esta matriz en autovectores y autovalores nos brinda información importante ya que, para redes no dirigidas, L es simétrica semidefinida positiva, por lo cual sus autovalores son iguales o mayores que cero. En particular nos interesa el autovalor más chico distinto de cero, y su correspondiente autovector, permite establecer un criterio aproximado para cortar la red en dos minimizando la cantidad de aristas que se cortan. Esto nos da una noción de centralidad de un nodo, la cual es una medida que nos permite denotar la importancia de un nodo por ejemplo, a partir de las conexiones que este tiene.

Como anticipamos previamente, descomponer en autovectores y autovalores a la matriz de adyacencia nos brinda información sobre la estructura de la red, específicamente sobre los nodos más centrales ya que podemos medir la centralidad de un nodo en base a la importancia que tienen los nodos con los que está conectado

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{n} \mathbf{A}_{ij} x_j \tag{3}$$

Esta medida de centralidad se conoce como centralidad del autovector ya que la expresión es equivalente a la ecuación $\lambda x = \mathbf{A}x$. Si además pedimos que la medida de centralidad tome exclusivamente valores positivos, entonces λ tiene que ser el mayor autovalor de la matriz de adyacencia.

Usaremos los siguientes dos métodos para calcular los autovalores y autovectores:

Método de la Potencia: Sea $\mathbf{A} \in \mathbb{R}^{n \times n}$. λ^1, ... , λ^n sus n autovalores con v^1, ... , v^n los autovectores asociados que conforman una base. Supongamos que $|\lambda^1| > |\lambda^2|$, ... , $\geq |\lambda^n|$.

Dado $q^0 \in \mathbb{R}^n$, $||q^0||_2 = 1$, la sucesión q^k definida como

$$\text{Para } k = 1, ...$$
$$z^k = \mathbf{A}q^{k-1}$$
$$q^k = \frac{z^k}{||z^k||_2}$$

converge al autovector v^1. Además $\lambda_k = (q^k)^t \mathbf{A} q^k$ converge λ^1.

El algoritmo implementado para la potencia es el siguiente:

PowerIteration(matriz, niter, ϵ)

```
v = []
for i = 0 hasta A.columnas  do
    v[i] = numRealAleatorioEntre(0, 1)
end for
vAnt = [0, ...,0]
AxV = [0]
for j = 0 hasta niter do
    AxV = A · v
    AxVnorm = ||AxV||
    v = AxV · 1/AxVnorm
    if ||v - vAnt|| < ϵ then
        break
    end if
    vAnt = v
end for
a = v · AxV/||v||
res = (a, v)
return res
```

Método de la deflación: Sea $\mathbf{A} \in \mathbb{R}^{n \times n}$. λ^1 autovalor con v^1 autovector asociado, $||v^1||_2 = 1$.

Sea $\mathbf{H} \in \mathbb{R}^{n \times n}$ matriz ortogonal tal que $\mathbf{H}v^1 = e_1$

$$\mathbf{HAH^t} = \begin{bmatrix} \lambda^1 & a^t \\ 0 & \mathbf{B} \end{bmatrix}$$

Como $\mathbf{A}$ y $\mathbf{HAH^t}$ tienen los mismos autovalores, los otros autovalores de $\mathbf{A}$ corresponden a los autovalores de $\mathbf{B}$.

El algoritmo implementado para la deflación es el siguiente:

Eigen(matriz, num, niter, ϵ)

```
eigenvalues = []
eigenvectors = [[0, ..., 0], ..., [0, ..., 0]]
maxEigenValue = 0
eigenvector = []
for i = 0 hasta num do
    maxEigenValue, eigenvector = PowerIteration(A)
    A = A - (maxEigenValue · eigenvector · eigenvector^t)
    eigenvalue.pushback(maxEigenValue)
    eigenvectors[i] = eigenvector
end for
return (eigenvalues, eigenvectors)
```

Covarianza y correlación: Definimos la correlación como una covarianza normalizada para que su rango sea entre -1 y 1 como

$$\mathbf{Corr(x,\ y)} = \frac{(\mathbf{x} - u_x) \cdot (\mathbf{y} - u_y)}{\sqrt{(\mathbf{x} - u_x) \cdot (\mathbf{x} - u_x)(\mathbf{y} - u_y) \cdot (\mathbf{y} - u_y)}} \tag{4}$$

donde $\mathbf{x}$ e $\mathbf{y}$ son vectores. Si se consideran los vectores columna x_i - u_{x_i} formando la matriz $\mathbf{X}$, la covarianza entre todas las columnas se llama matriz de covarianza y se expresa como:

$$\mathbf{C} = \frac{\mathbf{X^t X}}{n-1} \tag{5}$$

Esta matriz es simétrica y semidefinida positiva por lo cual sus autovalores son no negativos.

Análisis de componentes principales (PCA): Es una técnica utilizada para describir un conjunto de datos. El PCA construye una transformación lineal que busca ordenar las dimensiones según su varianza, de mayor a menor. Para lograr esto descompone en autovectores y autovalores a la matriz de covarianza

$$\mathbf{C = VDV^t} \tag{6}$$

donde $\mathbf{V}$ es la matriz de autovectores en las columnas y $\mathbf{D}$ es una matriz diagonal con los autovalores. Estos autovalores son la varianza que captura cada nueva dirección dada por los autovectores.

Matriz de similaridad: Dado un conjunto de datos $\mathbf{X} \in \mathbb{R}^{m \times n}$ con m datos y n atributos, se define una matriz de similaridad a una matriz de $\mathbf{D} \in \mathbb{R}^{m \times m}$ que computa una función sobre cada par de datos ij

$$\mathbf{D}_{ij} = f(\mathbf{X_i}, \mathbf{X_j}) \tag{7}$$

Un caso sencillo de realizar es el del producto interno

$$\mathbf{D = XX^t} \tag{8}$$

2. Experimentación

2.1. Tests algoritmos

Vamos a computar una matriz cualquiera teniendo los autovalores y autovectores de antemano para compararlos con los arrojados por nuestros algoritmos y así evaluar su correctitud. La matriz A la definimos como

$$\mathbf{A} = \begin{bmatrix} 17/5 & (-6)/5 & (-4)/5 & (-2)/5 & 0 \\ (-6)/5 & 16/5 & (-2)/5 & 0 & 2/5 \\ (-4)/5 & (-2)/5 & 3 & 2/5 & 4/5 \\ (-2)/5 & 0 & 2/5 & 14/5 & 6/5 \\ 0 & 2/5 & 4/5 & 6/5 & 13/5 \end{bmatrix}$$

Sus autovalores y autovectores en columnas asociados conocidos son

$$\text{eigenvalues} = \begin{bmatrix} 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

$$\text{eigenvectors} = \begin{bmatrix} -0{,}6 & 0{,}4 & 0{,}4 & 0{,}4 & 0{,}4 \\ 0{,}4 & -0{,}6 & 0{,}4 & 0{,}4 & 0{,}4 \\ 0{,}4 & 0{,}4 & -0{,}6 & 0{,}4 & 0{,}4 \\ 0{,}4 & 0{,}4 & 0{,}4 & -0{,}6 & 0{,}4 \\ 0{,}4 & 0{,}4 & 0{,}4 & 0{,}4 & -0{,}6 \end{bmatrix}$$

Luego, los arrojados por nuestros algoritmos con un máximo de 10000 iteraciones y un epsilon para el criterio de parada de $1 * 10^6$ son:

$$\text{eigenvalues} = \begin{bmatrix} 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

$$\text{eigenvectors} = \begin{bmatrix} -0{,}599999 & 0{,}399998 & 0{,}400001 & 0{,}400001 & 0{,}400001 \\ 0{,}400004 & -0{,}600001 & 0{,}399997 & 0{,}399999 & 0{,}399999 \\ 0{,}399999 & 0{,}400003 & -0{,}600001 & 0{,}399998 & 0{,}399999 \\ 0{,}399999 & 0{,}399999 & 0{,}400002 & -0{,}600001 & 0{,}399999 \\ 0{,}399999 & 0{,}399999 & 0{,}399999 & 0{,}400001 & -0{,}600001 \end{bmatrix}$$

Como se puede ver, los autovectores dan casi exactamente lo mismo, mientras que para los autovalores no hay diferencia. Para analizar el error de los autovectores, calculamos el error cuadrático medio entre todos los resultados, el cual es $2{,}6 * 10^{-12}$, un valor muy reducido en relación a la magnitud de los autovectores. Algo interesante para observar es que los autovectores tienen un error de la magnitud del epsilon elegido. En particular, para este ejemplo tomamos un epsilon de $1 * 10^6$ y podemos ver que la diferencia con los autovectores reales se encuentra del sexto decimal en adelante. Si tomásemos un epsilon de $1 * 10^{10}$, nos encontraríamos con errores a partir del décimo decimal. Decidimos usar un epsilon de $1 * 10^6$ en los siguientes experimentos ya que lo consideramos lo suficientemente preciso para obtener buenos resultados, sin afectar los tiempos del algoritmo. Para la cantidad de iteraciones consideramos que 10000 es un buen limite, ya que, por lo menos para los casos que analizamos, observamos que si el algoritmo llega a la última iteración, significa que este no converge. Luego, en caso de que converja, el criterio de parada del epsilon siempre corta la ejecución antes de llegar a la última iteración.

Pasemos al estudio de las redes.

2.2. Red de Karate

Como anticipamos en la introducción, esta red tiene 2 nodos que destacan para nosotros, el instructor y el administrador, ya que basaremos el estudio en analizar sus respectivos lazos e impactos en la dinámica del grupo luego del conflicto. Esta red cuenta con un grafo que nos indica las relaciones entre cada persona del grupo y que bando escoje cada uno posteriormente al conflicto.

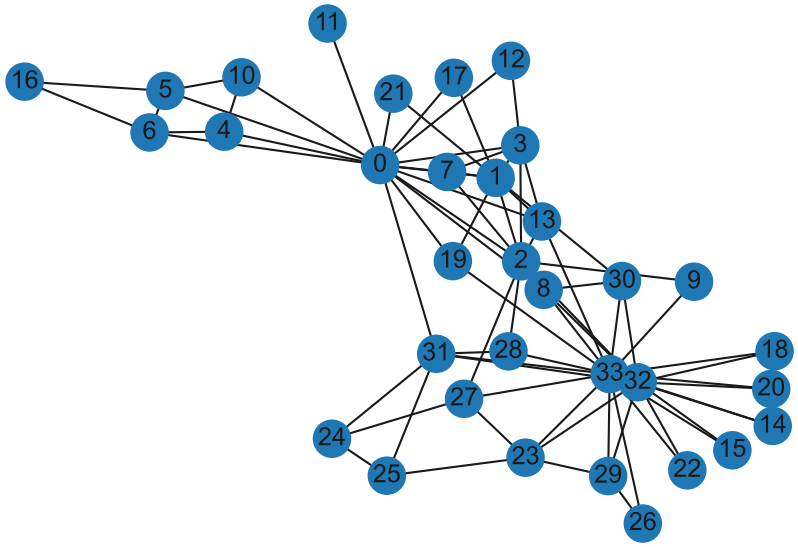

Figura 3: Grafo asociado a la red de Karate

Se puede intuir por el grafo asociado que los nodos más importantes son el 0 y el 33, los cuales son nuestros dos sujetos en cuestión, pero vamos a estudiar su centralidad a través del cómputo de centralidad.

Efectivamente se cumple lo esperado y los nodos más centrales son el 0 y el 33.

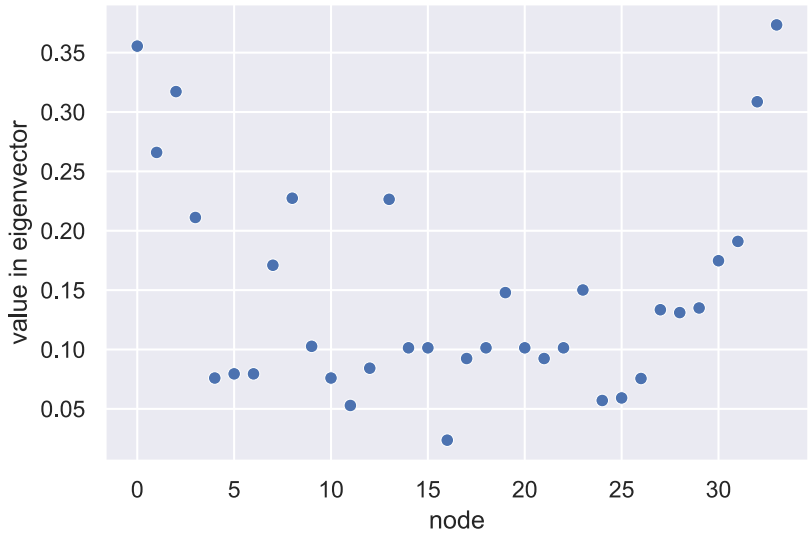

Figura 4: Valuación de los nodos según su centralidad

El vector de la centralidad normalizado lo deja ver claramente :

v = [**0.355**, 0.265, 0.317, 0.211, 0.075, 0.079, 0.079, 0.170, 0.227, 0.102, 0.075, 0.052, 0.084, 0.226, 0.101, 0.101, 0.023, 0.092, 0.101, 0.147, 0.101, 0.092, 0.101, 0.150, 0.057, 0.059, 0.075, 0.133, 0.131, 0.134, 0.174, 0.191, 0.308, **0.373**]

Vamos a buscar un autovector que prediga como terminaran los grupos luego del conflicto, para eso vamos a computar la matriz Laplaciana de la red Karate y sus autovectores para así calcular la correlación de nuestros autovectores contra el resultado de la separación karate. Si la correlación es alta significa que se asemejan mucho.

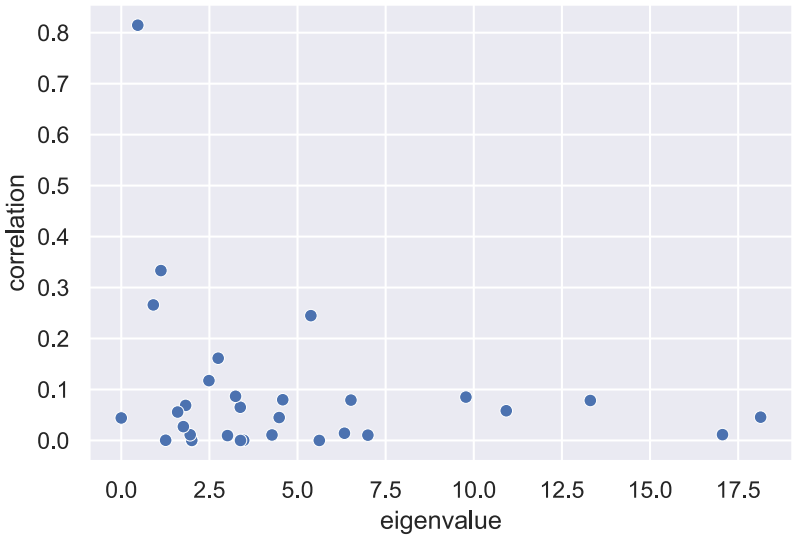

Figura 5: Correlación entre los autovectores de la Laplaciana y el resultado final de la red de Karate

Hay un autovector que nos da una correlación deseada y es el siguiente

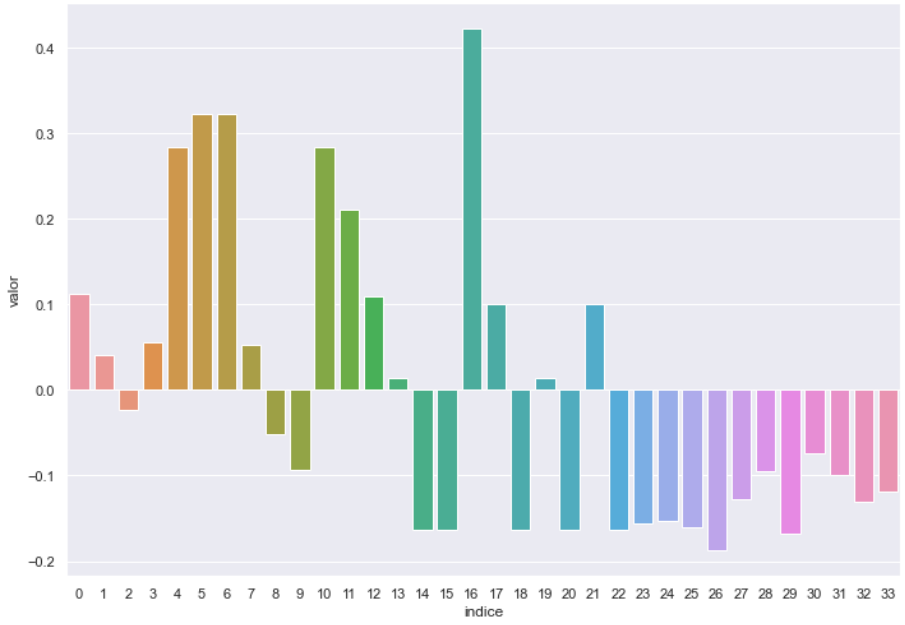

Figura 6: Vector de correlación más preciso

Entendiendo que los positivos pertenecen a un grupo y los negativos pertenecen al otro grupo, solo se equivoca en el nodo 3 y en el 9, ambos dan negativo cuando deberían ser positivos. Es correcta la alta correlación ya que solo predice mal 2 nodos de 34.

2.3. Red de Ego-Facebook

Esta red representa las amistades entre usuarios de Facebook, donde los nodos son los usuarios y las aristas definen amistad entre usuarios. Además de las amistades, también contamos con una matriz que contiene las características de cada usuario, que llamaremos matriz de atributos. A partir de estos dos conjuntos de datos, vamos a analizar cuan precisa puede ser la matriz de atributos para predecir si dos usuarios son amigos, a juzgar por sus atributos en común.

Luego de computar la matriz de similaridad con producto interno de la matriz de atributos, propusimos varios umbrales (del 0 al 14) y una matriz de adyacencia a partir de la matriz de similaridad para cada umbral, dejando las aristas que lo superaban.

Como podemos ver, el umbral 6 presento la mayor correlación, es decir la mayor similaridad.

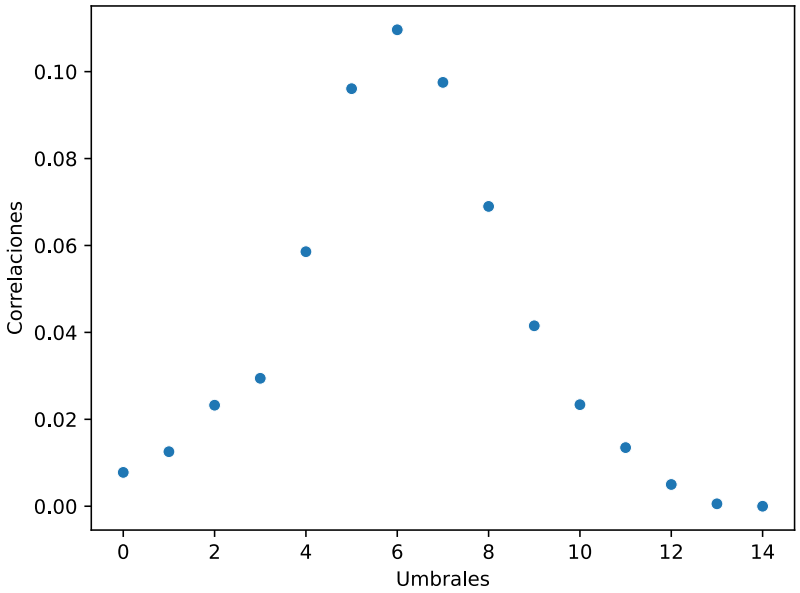

Figura 7: Correlación con 14 umbrales

Construimos 14 grafos variando los umbrales y luego computamos su correlación con la original. Teniendo en cuenta que la correlación más grande esta entre 0.1 y 0.2, el uso de los atributos para predecir los vínculos de las redes no parece ser una buena herramienta. Hagamos referencia a la predicción que hicimos sobre la red de karate, donde logramos un 0.8 de correlación y el resto de valores rondaban el 0.1 . Los valores que conseguimos con este método no resultan igual de efectivos, por lo menos en comparación (aunque se trate de distintas redes y métodos). Vale la aclaración de que en la red de karate solo intentábamos predecir como resultan los 2 bandos mientras que este caso es mas complejo porque se intenta predecir una red completa.

2.4. PCA

En el siguiente paso llevaremos a cabo un análisis de la matriz de atributos de la red Ego-Facebook mediante un análisis de las componentes principales. La idea es poder reducir la dimensionalidad de esta matriz, conservando al máximo la información que contiene. Vamos a hacer la prueba con 2 y 3 dimensiones y luego vamos a graficar los resultados para poder obtener una intuición visual sobre el dataset.

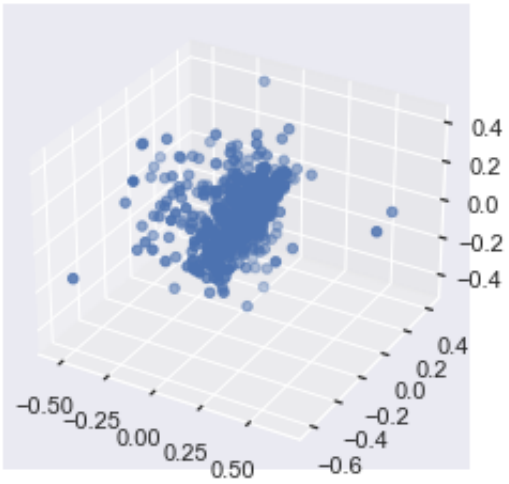

Figura 8: PCA en 3 dimensiones

De la primer figura observamos fundamentalmente una tendencia del eje x (1er componente principal) hacia los números negativos, con 2 excepciones fuertes como outliers. Y una distribución mas uniforme en el segundo y tercer componente principal alrededor del 0.

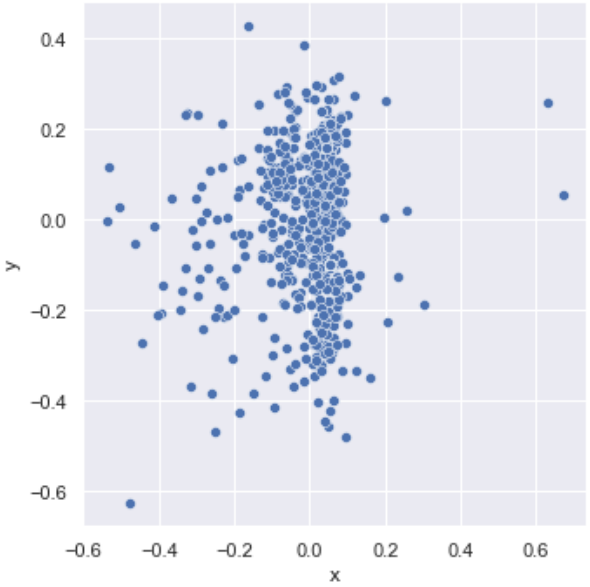

Figura 9: PCA en 2 dimensiones

Al rehacer el análisis con 2 dimensiones, refuerza la idea vista en el gráfico anterior.

Finalmente vamos comparar a través de la correlación la matriz de adyacencia que obtenemos con los atributos de PCA contra la de adyacencia que obtenemos con todos los atributos y un umbral 6, que daba el mejor resultado en cuanto a correlación con la matriz de adyacencia original. Queremos evaluar que valores funcionan mejor para este caso particular. Para esto vamos a tomar distintos valores de k componentes principales y distintos u umbrales.

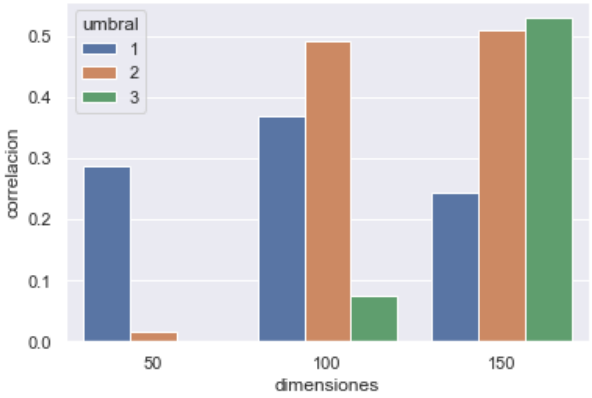

Figura 10: PCA vs Matriz de adyacencia

Como podemos ver, el resultado donde más se parecen es con $k = 150$ y tomando $u = 2$ o 3. Igualmente se puede obtener un resultado similar con menos computo si tomamos $k = 100$ y $u = 2$ o 1 en una última instancia. Intentamos utilizar una proporción de umbrales similar a la que se uso con la matriz original, usando umbrales mayores a 4 los resultados no eran buenos con estos k.

3. Conclusiones

En el dataset de la red de karate, podemos decir que los resultados acompañaron las hipótesis. Tanto la idea de centralidad de nodos como el resultado de la partición de la red se condicen con los resultados reales del experimento llevado a cabo. Por otro lado, y al contrario de nuestras hipótesis, en el caso de la red de facebook nos encontramos con que no parece haber un vinculo muy claro entre los atributos de los nodos de la red y sus vínculos en la misma. Por lo menos en lo que respecta los experimentos que llevamos a cabo en este trabajo. En cuanto a PCA, mostró ser una forma practica de poder visualizar la matriz de atributos y una buena técnica para reducir dimensionalidad conservando información. Luego de realizar los estudios, concluimos que los autovalores y autovectores son muy buenas herramientas a la hora de desarmar y analizar la red. También provee información critica a la hora de poder predecir resultados.

Esperamos en el futuro volver a utilizar estas herramientas y ahondar más en profundidad en las posibilidades que las mismas nos presenta.