UNIVERSIDAD DE LA REPÚBLICA

FACULTAD DE INGENIERÍA

APRENDIZAJE AUTOMÁTICO PARA DATOS EN GRAFOS

Laboratorio 1 | Introducción al procesamiento de grafos

Autores:
Federico BELLO
Gonzalo CHIARLONE

28 de septiembre de 2025

1. Ejercicio 1

1.1.

Número de aristas dirigidas: 3007.

```
G.number_of_edges()
```

1.2.

Número de aristas no-dirigidas: 2097.

```
G.to_undirected().number_of_edges()
```

1.3.

Número de arcos mutuos: 910.

```
G.to_undirected(reciprocal=True).number_of_edges()
```

El parámetro *reciprocal=True* deja en el grafo solamente aristas que van en ambas direcciones.

1.4.

Número de nodos con in-degree igual a 0 (o sea, empleados que no recibieron mails): 3.

```
len([node for node, in_degree in G.in_degree() if in_degree == 0])
```

Corresponde a las personas [71, 117, 135], las únicas personas que no **recibieron** mails en todo el periodo de tiempo. Esto se puede ver al eliminar la función *len()* de la linea de código provista.

1.5.

Número de nodos con out-degree igual a 0 (o sea, empleados que no enviaron mails): 9.

```
len([node for node, out_degree in G.out_degree() if out_degree == 0])
```

Corresponde a las personas [42, 52, 71, 87, 111, 117, 122, 150, 164], las únicas personas que no **enviaron** mails en todo el periodo de tiempo. Esto se puede ver al eliminar la función *len()* de la linea de código provista.

1.6.

El número de empleados que fueron contactados por más de 30 empleados.

```
len([node for node, in_degree in G.in_degree() if in_degree > 30])
```

El grafo resultante de remarcar estos nodos se ve en la figura 1.

1.7.

El número de empleados que contactaron a más de 30 empleados.

```
len([node for node, out_degree in G.out_degree() if out_degree > 30])
```

El grafo resultante de remarcar estos nodos se ve en la figura 2.

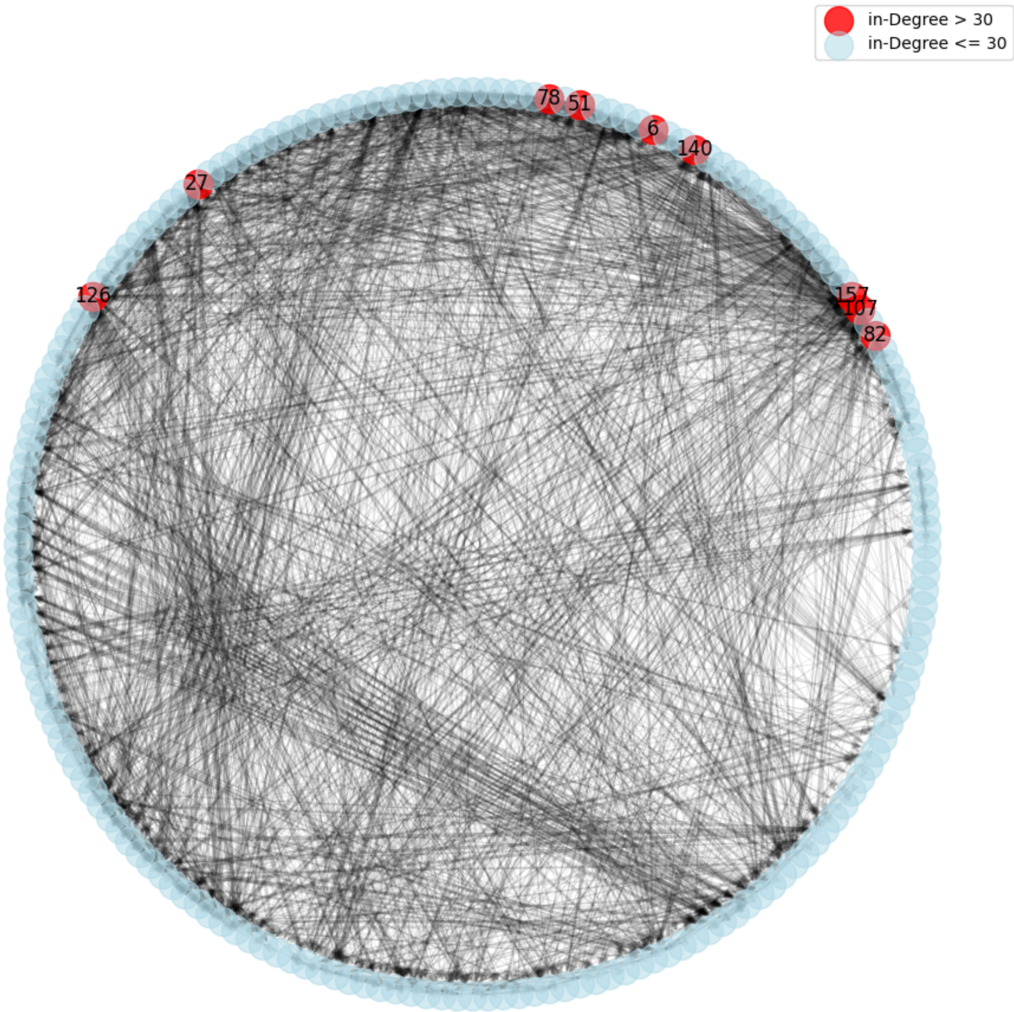


Figura 1: Grafo remarcando los nodos con grado entrante mayor a 30

1.8.

En la figura 3 se observa histograma del grado de los nodos (in y out), donde se puede verificar fácilmente el patrón de muchos nodos con un grado bajo y pocos nodos con un grado alto, tanto para los grados salientes como para los entrantes.

1.9.

Como medidas de indicadores de nodos se tomaron en consideración la centralidad por grado y el *Page Rank*. La elección de la primera fue debido a que es probable que si una persona envía y/o recibe una gran cantidad de mails es porque esa persona es importante en el contexto del problema. Por ejemplo, es esperable que el CEO de la empresa reciba mas mails que un empleado cualquiera. Por otro lado, también se tuvo en cuenta el *Page Rank* para complementar la medida anterior, de esta forma, al recibir un mail se “hereda"la importancia de quien lo envió y se evita asignar importancia a nodos que reciben *spam*.

La tabla 1 muestra las 5 personas mas importantes según cada ranking, tanto al tomar el grafo global o por instante de tiempo. Al tomarlo por instante se ordenaron de forma tal que el mas importante es el que

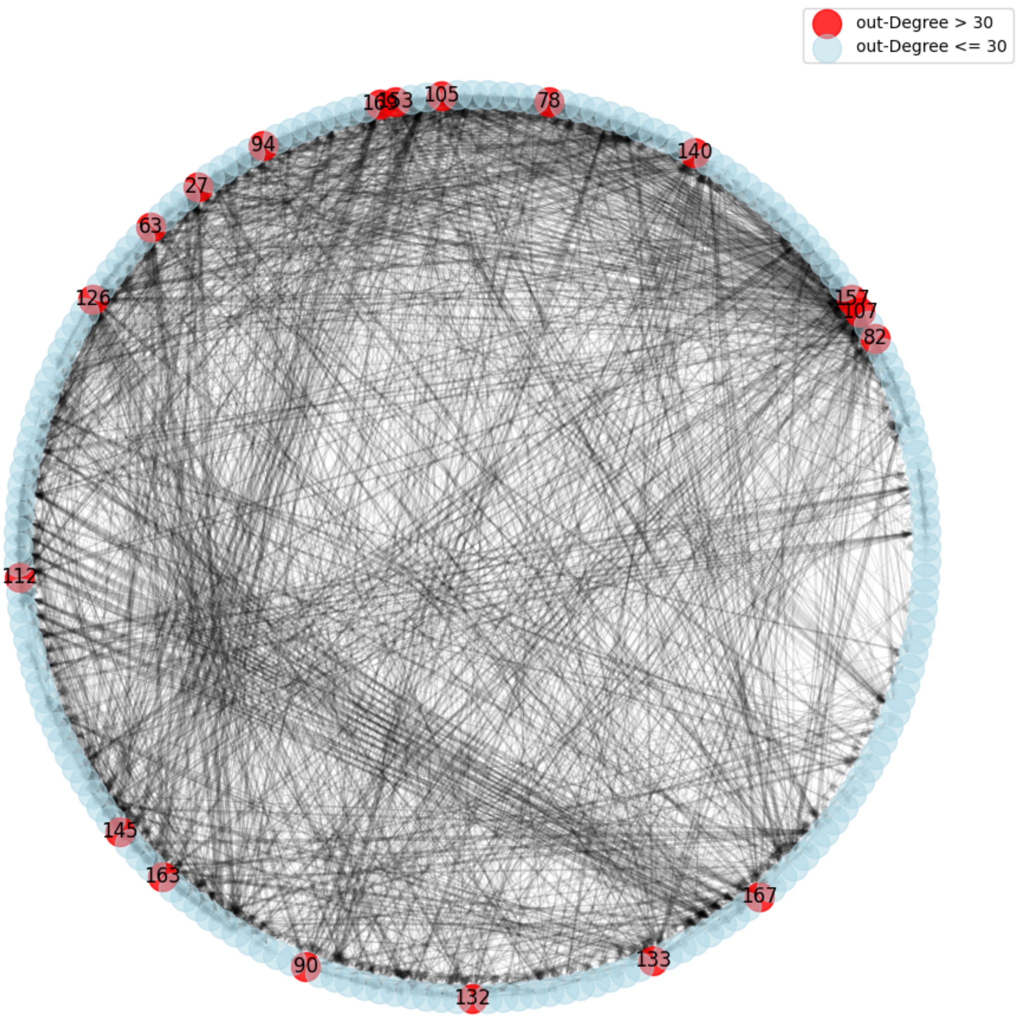


Figura 2: Grafo remarcando los nodos con grado saliente mayor a 30

“gano"la mayor cantidad de instantes temporales. Se observa entonces como si bien el resultado obtenido para cada metodología y métrica es distinto se observan claros patrones. Por ejemplo, el nodo 107 aparece en las 4 columnas, indicando que esta persona es realmente importante.

Ranking \ Métrica	Por Grado Por instante	Page Rank Por instante	Por Grado Global	Page Rank Global
1	82	82	169	114
2	107	126	114	169
3	153	107	155	82
4	105	114	50	107
5	126	169	107	65

Cuadro 1: Importancia de personas para distintas métricas

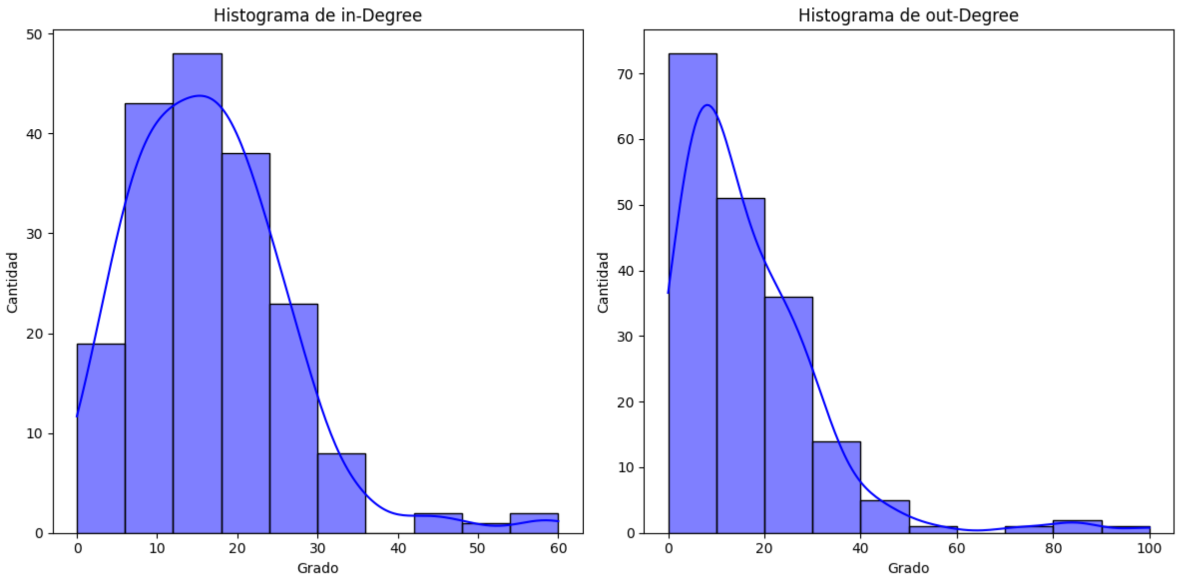


Figura 3: Histogramas del grado de los nodos

1.10.

Como medida indicadora a nivel grafo, se tomaron el coeficiente de clustering y el coeficiente de asortatividad por grado. El primero, como se vio en el curso, da una idea de que tan agrupado (o "clusterizado") se encuentra el grafo. En este caso, esta métrica resulta de interés ya que seria esperable ver mas grupos de mails ante un evento o noticia. Por ejemplo, además de los grupos que se forman por área de trabajo, se podrían formar grupos mas sociales que equivalgan a conversaciones mas casuales.

Por otro lado, el coeficiente de asortatividad por grado también resulta interesante. Este nos dice que tanto intercambio hay entre nodos de distinto grado. En este caso, indica que tanto se hablan personas que mandan pocos mails con personas que mandan muchos. Podría darse que este coeficiente aumente al ocurrir un evento importante, ya que personas que normalmente no se comunican pueden necesitar hacerlo.

Como se aclara en el *paper* publicado en la letra, se pueden ver grandes variaciones de ambas métricas que coinciden con eventos importantes. Por ejemplo, en Diciembre de 1999 con lanzamiento de Enron online se observan variaciones marcadas en los gráficos de abajo de la figura 4.

2. Ejercicio 2

2.1.

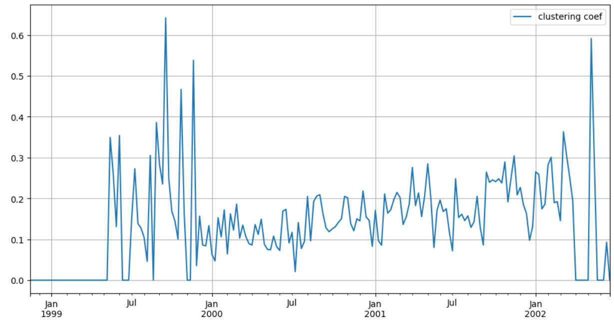
Para calcular el Laplaciano del grafo alcanza con ejecutar:

```
L = get_laplacian(data.edge_index)
L = to_torch_csc_tensor(edge_index = L[0], edge_attr= L[1])
```

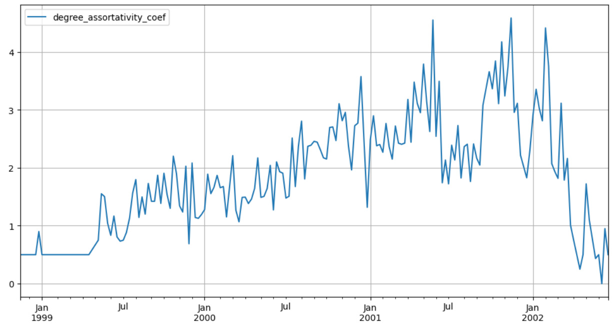
donde la segunda linea simplemente convierte L de una tupla a un tensor.

2.2.

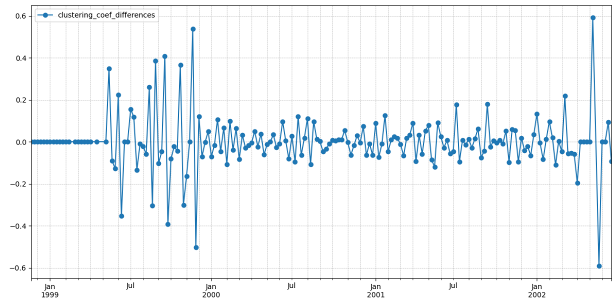
Para verificar que el vector $\mathbf{1} = [1, 1, \ldots, 1]^\top$ es vector propio del Laplaciano con valor propio asociado 0 es necesario primero calcular $\mathbf{L} \cdot \mathbf{1}$ y verificar que de como resultado el vector de ceros. Para esto, se ejecutaron las siguientes lineas y se verifico que dieran el resultado esperado.



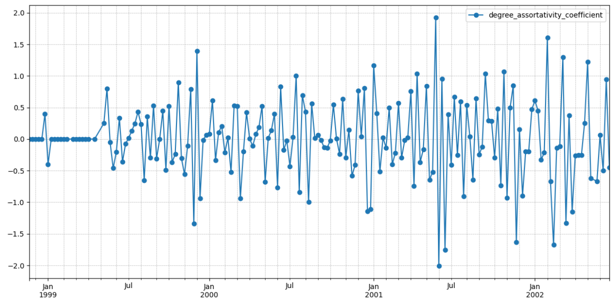
(a) Coeficiente de clustering



(b) Coeficiente de asortatividad por grafo



(c) Variación del coeficiente de clustering



(d) Variación del coeficiente de asortatividad por grafo

Figura 4: Variación semanal de los indicadores a nivel de grafo.

```
ones = torch.ones(L.shape[0])
torch.linalg.multi_dot([L, ones.t()])
```

2.3.

Con el objetivo de verificar que **L** sea semi definida positiva, alcanza con calcular los valores propios de la matriz:

```
vaps = torch.linalg.eigvals(L.to_dense())
```

Al realizar el calculo, si bien uno de los valores da negativo, no es mas que aproximaciones de punto flotante, siendo este el valor propio 0.

2.4.

Por último, para armar una matriz $\tilde{\mathbf{B}}$ tal que $\mathbf{L} = \tilde{\mathbf{B}}\tilde{\mathbf{B}}^{\mathbf{T}}$:

```
B = torch.Tensor(nx.incidence_matrix(G, oriented=True).toarray())
```

Para verificar que se cumpla la igualdad alcanza con realizar la cuenta y ver que ambos lados sean iguales.

3. Ejercicio 3

Para este próximo ejercicio se siguieron las ideas planteadas en el capitulo 6 de [1].

3.1. Parte 1

Recordando la definición de valor y vector propio, se tiene que una matriz **X** tiene de vector propio a **v** con valor propio asociado λ sii se cumple que $\mathbf{X}\,\mathbf{v} = \lambda\mathbf{v}$

De la ecuación $\mathbf{L} = \mathbf{D} - \mathbf{A}$ se desprende el elemento en la i-ésima fila y j-ésima columna de la matriz $\mathbf{L}$ es:

$$\mathbf{L} = \begin{cases} k_i, & \text{si } i = j \\ -1, & \text{si } i \neq j \text{ y existe el vértice } (i,j) \\ 0, & \text{en otro caso} \end{cases}$$

dónde k_i es el grado del nodo i.

Por lo tanto, al multiplicar el vector $\mathbf{1} = (1, 1, .., 1)$ se obtiene que el resultado del i-ésimo elemento es:

$$\sum_j L_{ij} \times 1 \overbrace{=}^{(1)} \sum_j (\delta_{ij} k_i - A_{ij}) \overbrace{=}^{(2)} k_i - \sum_j A_{ij} = k_i - k_i = 0$$

dónde (1) sale de escribir el Laplaciano como se mencionó anteriormente $\delta_{ij} = 0$ sii $i \neq j$ y (2) de observar que la sumatoria de aristas adyacentes es el grado del vértice. De lo anterior surge entonces que el 0 es valor propio asociado a $\mathbf{1}$:

$$\mathbf{L} \cdot \mathbf{1} = \mathbf{0} = 0\,\mathbf{1}$$

3.2. Parte 2

Considerar la suma $\sum_e \tilde{B}_{ie} \tilde{B} B_{je}$. Si $i \neq j$ pueden pasar dos cosas. Si la arista e conecta el vértice i con el vértice j entonces el valor es -1, ya que hay como máximo una arista entre dos vértices. Si no los conecta el valor es 0.

Si $i = j$ se tiene que $\sum_e \tilde{B}_{ie} \tilde{B}_{je} = \sum_k \tilde{B}_{je}^2$, por lo que cada arista conectada al vértice j aportara un termino de 1, es decir el grado del vértice.

De esta forma, se obtiene que $\sum_k \tilde{B}_{ie} \tilde{B}_{ej} = \mathbf{L}$, de lo que se desprende fácilmente que

$$\mathbf{L} = \tilde{\mathbf{B}} \tilde{\mathbf{B}}^T$$

3.3. Parte 3

Observar que $\mathbf{x^T L x} = \mathbf{x^T \tilde{B} \tilde{B}^T x}$ no es mas que el producto de $\mathbf{\tilde{B}^T x}$ consigo mismo. Por lo tanto, el resultado es la suma de los elementos del vector resultante al cuadrado. Como $\tilde{\mathbf{B}}$ tiene exactamente un $+1$ y un -1 en cada fila, se obtiene el resultado deseado. Notar además que el resultado es independiente de la orientación "virtual.$^{\text{el}}$egida.

Siguiendo la misma linea de razonamiento, se ve que $\mathbf{x^T L X} \geq 0 \;\; \forall\, \mathbf{x}$, en particular, se cumple también si $\mathbf{x}$ es un vector propio. Recordando que el 0 es siempre valor propio de la matriz, se llega a que la matriz es semi definida positiva.

La simetría se deduce de forma simple de que la matriz $\mathbf{D}$ es diagonal (y por lo tanto simétrica), la matriz $-\mathbf{A}$ es simétrica (ya que el grafo no es dirigido) y la suma preserva la simetría.

3.4. Parte 4

Si el grafo no es conexo implica que no hay ningún camino entre dos pares de vértices cualesquiera de cada componente, por lo que la matriz $\mathbf{A}$puede escribirse de la forma:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A_1} & 0 & \cdots \\ 0 & \mathbf{A_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{1}$$

donde $\mathbf{A_1}, \mathbf{A_2}$... son las matrices de adyacencia de cada componente, las cuales cumplen por separado las mismas propiedades que $\mathbf{A}$. Por lo tanto, $\mathbf{L}$ puede escribirse de la misma forma:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L_1} & 0 & \cdots \\ 0 & \mathbf{L_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{2}$$

Considerar que la matriz $\mathbf{L_1}$ consiste de n_1 vértices y la matriz L_2 de n_2. Luego, multiplicar $\mathbf{L}$ por $\mathbf{v} = (\underbrace{1, 1, .., 1}_{n_1}, 0, 0, .., 0)$ da como resultado 0, así como $\mathbf{v} = (\underbrace{0, 0, ..., 0}_{n_1}, \underbrace{1, 1, .., 1}_{n_2}, 0, 0, .., 0)$, donde los vectores son linealmente independientes entre sí.

Referencias

[1] Mark Newman. *Networks.* Oxford University Press, March 2010.