Álgebra Lineal Computacional

1er Cuatrimestre 2025

Trabajo Práctico N° 2: Detección de comunidades con métodos espectrales.

Introducción

En este segundo TP continuaremos trabajando con los museos que usaron en el TP1. Nuestro objetivo será poner a prueba dos métodos iterativos para *detectar* comunidades (es decir, grupos o segmentos) en redes. Ambos métodos son *no-supervisados*, es decir que no emplean ejemplos de clasificación: directamente proveen una segmentación de los nodos de la red. Para esto, hacen uso de la estructura de la misma. La idea intuitiva que hay detrás es simple: vértices que están más conectados entre sí que con el resto deberían pertenecer al mismo grupo. De esta forma, optimizar métricas que cuantifiquen la conectividad entre los grupos elegidos suena razonable. Sin embargo, no es obvio cuál es la métrica que debe optimizarse, ni como (si les da curiosidad, miren [1] para un *review* de distintas estrategias para identificar grupos en redes, mientras que los métodos presentados aquí se encuentran en [2]).

Lo esencial en cada problema es pensar críticamente qué magnitud representa de mejor manera lo que queremos capturar. En este TP utilizaremos dos de las magnitudes más clásicas: el *corte mínimo*, y la *modularidad*. El corte mínimo representa el conjunto mínimo de conexiones en una red que se debe remover para partirla en dos grupos. Intuitivamente, podríamos esperar que si una red está representada por dos grupos debilmente conectados, podría partirse en dos grupos con solo remover unas pocas conexiones. La modularidad, por otra, parte mide (para una asignación en grupos) cuantas más conexiones hay entre ellos de las que esperaríamos por azar. Obviamente, esto conlleva definir *azar* más especificamente. En la Figura 1, podemos identificar visualmente dos grupos, representados por los nodos 1 a 4 y 5 a 8. Sin embargo hay algunas conexiones entre los dos grupos. Vamos a ver que en esta red ambos criterios coiden, aunque este no sea el caso para todas las redes.

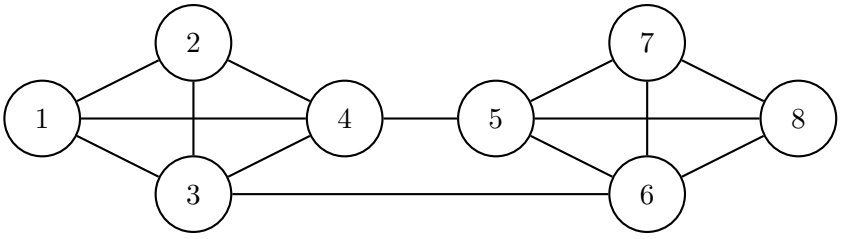

Figura 1: Ejemplo de red con dos grupos.

Avancemos en la formalización matemática. A lo largo del TP siempre estaremos pensando en partir las redes en dos grupos, por simplicidad. Por esto, vamos a poder representar la asignación en comunidades mediante un vector $\mathbf{s}$ tal que $\mathbf{s}_i = 1$ si $i \in G_1$ (grupo 1) y $\mathbf{s}_i = -1$ si $i \in G_2$ (grupo 2). En estos términos, nuestro objetivo va a ser encontrar un vector $\mathbf{s}$ optimo para un dado criterio. Por ejemplo, el que encontramos a *ojimetro* en el grafo anterior sería

$$\mathbf{s} = (1, 1, 1, 1, -1, -1, -1, -1)^t$$

(notar que $-s$ sería equivalente).

Retomando sobre el TP anterior, consideremos la matriz de adyacencia A para el grafo de la Figura 1, con $A_{ij} = 1$ si i y j están conectados y $A_{ij} = 0$ si no. En este TP, consideraremos por simplicidad redes *sin dirigir*, y por lo tanto supondremos que la matriz A es simétrica [1]. Podemos contar la cantidad de conexiones entre ambos grupos como

$$\Lambda = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} \frac{(1 - s_i s_j)}{2} \tag{1}$$

Noten como $(1 - s_i s_j)/2$ vale 1 si i y j están en grupos distintos y 0 si no. El factor 2 adelante sirve para descontar conexiones que se cuentan dos veces por ser la red no dirigida. Haciendo algunas manipulaciones algebráicas se puede mostrar que

$$\begin{aligned} \Lambda &= \frac{1}{4} \mathbf{s}^t L \mathbf{s} \\ L &= K - A \end{aligned} \tag{2}$$

donde K es la matriz (diagonal) de grado que definimos en el TP anterior, con $K_{ii} = k_i = \sum_{j=1}^{N} A_{ij}$, y L se denomina *matriz laplaciana* del grafo. Una estrategia posible para encontrar el vector $\mathbf{s}^\Lambda$ optimo que minimice Λ es buscar los autovectores de autovalor más pequeño de la matriz. Dado que nada garantiza que estos vectores estén compuestos de -1s y 1 (como corresponde a los vectores $\mathbf{s}$, la heuristica es tomar para s_i el signo de cada uno de los elementos del vector. De esta forma, el problema de minimizar Λ se convierte en el de encontrar sus autovectores de autovalor más bajo; problema que podemos resolver con las herramientas de la materia. **Se puede mostrar que la matriz L es semidefinida positiva**, y por lo tanto **todos sus autovalores son mayores o iguales a cero**. No mostraremos aquí este resultado, pero será importante para el resto del TP.

A través de su uso en distintos contextos, se ha observado que la estrategia de buscar los cortes mínimos no siempre devuelve los resultados esperados intuitivamente. Es por esto que surge la noción de modularidad. La modularidad es un concepto más refinado, donde buscamos comparar la estructura de la red que observamos con respecto a la que podría haber tenido. Esto agrega un nivel de referencia contra el cuál comparar, y es útil cuando tenemos un modelo sobre cómo podría ser la estructura de una red en otras realizaciones. Si definimos a P_{ij} como el número esperado de conexiones entre i y j (en nuestro caso, un número entre 0 y 1), entonces la modularidad se calcula como:

$$Q = \frac{1}{2E} \sum_{i=1}^{N} \sum_{j=1}^{N} (A_{ij} - P_{ij}) \frac{(1 + s_i s_j)}{2} \tag{3}$$

donde $2E = \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij}$ es dos veces el número total de conexiones en la red. La matriz P se construye de forma tal que $\sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} = 2E$. Fijada esta característica, muchos modelos son posibles. El más común es el denominado *configuration model* donde se imagina que los nodos son *recableados*, manteniendo su grado. Eso da lugar a

[1] Para esto, podemos simetrizar la matriz A del TP anterior haciendo $\lceil \frac{1}{2}(A + A^t) \rceil$.

$$P_{ij} = \frac{k_i k_j}{2E} \tag{4}$$

Es decir que la cantidad esperada de conexiones entre dos nodos es proporcional a sus grados. Definiendo $R = A - P$, se puede mostrar que

$$Q = \frac{1}{4E}\mathbf{s}^t R \mathbf{s} \tag{5}$$

En este caso, sin embargo, queremos encontrar los valores de $\mathbf{s}$ que *maximizen* el valor de Q. Nuevamente, una vez encontrado el autovector que tiene (en este caso) mayor autovalor, tomamos como $\mathbf{s}_i$ al signo del elemento i de dicho vector. De esta forma, tenemos dos heurísticas para encontrar comunidades en grafos. Mientras que tanto L como R son simétricas [2], la matriz R no tiene un signo definido y por lo tanto sus autovalores pueden ser cualesquiera.

Heurística de bisección

En base a lo discutido en la sección anterior, nuestras dos heurísticas se enfocarán

- en minimizar Λ, buscando los autovalores más bajos de L. Debido a las características de L, esto corresponderá a encontrar su segundo autovalor más chico λ_{N-1} [3]. El autovalor más chico, λ_N es igual a cero, y está asociado a una división poco interesante de la red (a ser probado en el primer punto del TP).
- en maximizar Q, buscando los autovalores positivos más grandes de R. En este caso, la lógica de búsqueda es más sencilla, y particionaremos en base a λ_1.

Partición iterativa

En todo lo que discutimos anteriormente, la metodología sirve para encontrar particiones de la red en dos grupos. Sin embargo, en muchos casos podría ocurrir que la red tuviese más de dos grupos en su interior. Una heurística para circunvalar ese problema se basa en partir la red en dos grupos, y luego repetir el algoritmo en cada uno de los subgrupos encontrados. En el caso del laplaciano esto puede continuar indefinidamente, y es necesario proveer externamente un número máximo de subdivisiones hasta el cuál llegar. En cambio, la modularidad nos da un criterio de parada: sólo continuaremos subdividiendo si la nueva partición aumenta la modularidad global de la red.

Enunciado

En este segundo TP vamos a tener que definir varias funciones para calcular los autovectores que necesitan cada método. Apoyense en cada paso con el grafo de la Figura 1. El mismo tiene matriz de adyacencia

[2] Y por lo tanto admiten bases ortonormales de autovectores. Tomamos este resultado como válido, a ser probado en la teórica.

[3] Al escribir esto, asumimos que los autovalores están ordenados de forma tal que $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$.

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{6}$$

y pueden encontrar una versión ya escrita en `numpy` en el template que acompaña el TP. **Recuerden en todos los puntos que trabajamos con matrices de adyacencia simétricas**, es decir $A = A^t$

1. **Autovectores y autovalores de L y R**

   a. Muestren que el vector de unos $\mathbf{1}$ es autovector de las matrices R y L. ¿Qué autovalor tiene? ¿Y qué agrupación de la red representa?

   b. Muestren que si L (R) tienen dos autovectores $\mathbf{v}_1$ y $\mathbf{v}_2$ asociados a autovalores $\lambda_1 \neq \lambda_2$, entonces $\mathbf{v}_1^t \mathbf{v}_2 = 0$. *Tip: Consideren una matriz M simétrica con dos autovectores $\boldsymbol{v}_1$ y $\boldsymbol{v}_2$ con autovalores distintos λ_1 y λ_2. Comparen el resultado de hacer $\boldsymbol{v}_1^t M \boldsymbol{v}_2$ y $\boldsymbol{v}_2^t M \boldsymbol{v}_1$.*

   c. Muestren si $\mathbf{v}$ es un autovector de autovalor $\lambda \neq 0$ de R o L, entonces $\sum_i \mathbf{v}_i = 0$.

2. **Extensiones del método de la potencia** Consideren una matriz $M \in \mathbb{R}^{n \times n}$ diagonalizable con autovalores $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, y autovector $\mathbf{v}_i$ asociado a λ_i

   a. **Shifting de autovalores:** Muestre que los autovalores de $M + \mu I$ son $\gamma_i = \lambda_i + \mu$, y que el autovector asociado a γ_i es $\mathbf{v}_i$. Concluya que si $\mu + \lambda_i \neq 0 \ \forall i$, entonces $M + \mu I$ es inversible.

   b. **Método de la potencia inverso:** Considerando $\mu > 0$, muestren que $L + \mu I$ es inversible, con L el laplaciano definido en la ecuación 2. Muestren que aplicar el método de la potencia a $(L + \mu I)^{-1}$ converge a su autovector de autovalor más chico si se parte de una semilla adecuada. Indique, en el caso de que hay sólo un autovector con dicho autovalor, cuál es dicho autovector y cuánto vale su autovalor.

   c. **Deflación de Hotelling.** Suponiendo que M es simétrica (y por lo tanto admite una base ortogonal de autovectores), muestre que la matriz $\tilde{M} = M - \lambda_1 \frac{\mathbf{v}_1 \mathbf{v}_1^t}{\mathbf{v}_1^t \mathbf{v}_1}$ tiene los mismos autovectores que M, pero el autovalor asociado a $\mathbf{v}_1$ es igual a cero.

3. **Implementación computacional**:

   a. Construyan las matrices del problema:

      i. `calcula_L(A)` y `calcula_R(A)` que reciban la matriz de adyacencia A y retornen las matrices L y R respectivamente.

      ii. `calcula_lambda(L,v)` y `calcula_Q(R,v)` que reciban las matrices L y R, y un autovector $\mathbf{v}$, y retornen el corte Λ y la modularidad Q asociados a $\mathbf{v}$. Calculen con ellas el corte y la modularidad de las particiones *esperadas* en la red de ejemplo.

Nota 1: Para el optimizar, pueden calcular $\boldsymbol{s}^t R \boldsymbol{s}$ en lugar de Q.
Nota 2: Recuerden que tanto para la modularidad como para el corte mínimo, tomamos $\boldsymbol{s}_i = signo(\boldsymbol{v}_i)$, el signo de cada elemento de $\boldsymbol{v}$

b. Construya funciones que encuentren los autovectores a través del método de la potencia:

   i. Una función `metpot1(M)` que reciba una matriz M y use el método de la potencia para retornar el autovalor de mayor módulo y su correspondiente autovector.

   ii. Una función `deflaciona(M)` que reciba una matriz M, calcule su primer autovector y autovalor, y calcule su versión deflacionada. *Tip: Aproveche la función `numpy.linalg.outer`*

   iii. Una función `metpotI(M,mu)` que reciba una matriz M y un coeficiente μ y calcule el autovalor más chico de $M + \mu I$ junto a su autovector asociado, usando el método de la potencia inversa. *Para invertir la matriz $M + \mu I$, utilice la descomposición LU programada en el TP anterior.*

   iv. Una función `metpotI2(M,mu)` que reciba una matriz M y un coeficiente μ y calcule el segundo autovalor más chico, con su autovector asociado, de la matriz $M + \mu I$, y bajo la suposición de que todos los autovectores de M son no-negativos, y sólo uno de ellos es igual a cero.

   Aplique las funciones obtenidas para calcular el autovector asociado al segundo autovalor más chico de la matriz L, y el autovector asociado al autovalor más grande de R. Observe los valores obtenidos en términos del grafo de ejemplo e interprete. Calcule los vectores $\mathbf{s}$ asociados a estos autovectores, y comparen con la partición esperada para el grafo de ejemplo.

c. Construya dos funciones que realicen particiones iterativas de los grafos, *apoyandose en los ejemplos provistos en el template*:

   i. `laplaciano_iterativo(A,niveles)` que reciba la matriz de adyacencia A y el número de niveles que se debe alcanzar realizando particiones iterativamente (para `niveles=k` se obtienen 2^k particiones). La función debe calcular el laplaciano L, y recursivamente partir la red hasta llegar a n niveles de partición. El resultado debe ser una lista de listas, donde cada sub-lista contiene los indices de los nodos correspondientes a una misma comunidad.

   ii. `modularidad_iterativo(A)` que reciba la matriz de adyacencia A y compute la matriz de modularidad R. Luego debe realizar iterativamente particiones en las comunidades del grafo en mitades. El algoritmo debe detenerse cuando las siguientes divisiones no aumentan la modularidad total.

   Aplique estas funciones al grafo de ejemplo de la Figura 1. ¿Cuál es la partición óptima en 4 grupos para el método basado en el laplaciano? ¿Cuál es la partición óptima basada en la modularidad?

4. ***Vuelta a la red de los museos...*** Usando la red de museos definida en el TP anterior, calcule las particiones óptimas usando el método basado en el laplaciano y el método basado en la modularidad. Utilice la matriz de adyacencia A construida usando $m = 3, 5, 10, 50$, luego de haberla simetrizado haciendo

$$A' = \lceil \frac{1}{2}(A + A^t) \rceil$$

con A' la versión simetrizada de A y $\lceil x \rceil$ el resultado de aplicar la funcion *ceiling* a x. Exploren cómo cambia la estructura de comunidades obtenida usando la modularidad (en términos de número de comunidades, de su tamaño y de las regiones del mapa que ocupan, así como la estabilidad ante realizaciones) encontrada para distintos valores de m. Comparen visualmente las comunidades obtenidas mediante el laplaciano y la modularidad, buscando un número de niveles que dé una cantidad de comunidades comparable en ambos métodos. **Discuta los resultados obtenidos**.

5. **Síntesis final.** Habiendo realizado las correcciones solicitadas al primer TP, escriba una conclusión general de lo observado al analizar la red de museos considerando tanto el TP1 como el TP2. Escriba un texto de 400 palabras discutiendo lo observando en la red de museos, vinculando los resultados obtenidos en ambos TPs. El texto debe detallar sus conclusiones y aprendizajes sobre los métodos y datasets empleados.

Entrega y lineamientos

La entrega se realizará a través del campus virtual de la materia con las siguientes fechas y formato:

- Fecha de entrega: hasta el Martes **17 de Junio** a las 23:59 hs.
- Formato: Jupyter Notebook del template-alumnos. Archivo template-funciones.py completo.

Prestar especial atención a las siguientes indicaciones:

- El TP2 se realizará en grupos de tres personas. Deberán inscribir su grupo en el foro 'Foro de Grupos de TP' destinado para tal fin, dentro de la sección Laboratorio/TP2 del campus de la materia.

  Importante: es indispensable realizar la inscripción previa del grupo para poder hacer el envío a través del campus. Los grupos o personas no inscriptas en grupos no estarán habilitadas en el formulario de carga del TP. No se aceptarán envíos por email.

- Leer el enunciado completo antes de comenzar a generar código y sacarse todas las dudas de cada ítem antes de implementar. Para obtener un código más legible y organizado, pensar de antemano qué funciones deberán implementarse y cuáles podrían reutilizarse.

- El código debe estar correctamente comentado. Cada función definida debe contener un encabezado donde se explique los parámetros que recibe y qué se espera que retorne. Además las secciones de código dentro de la función deben estar debidamente comentados. Los nombre de las variables deben ser explicativos.

- Las conclusiones y razonamientos que respondan los ejercicios, o cualquier experimentación agregada, debe estar debidamente explicada en bloques de texto de las notebooks (markdown cells), separado de los bloques de código. Aprovechen a utilizar código LaTeX si necesitan incluir fórmulas.

- Gráficos: deben contener título, etiquetas en cada eje y leyendas indicando qué es lo que se muestra.

Referencias

[1] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics reports*, 659:1–44, 2016.

[2] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 74(3):036104, 2006.