

AGRUPAMIENTO ESPECTRAL

ALAN REYES-FIGUEROA

INTRODUCCIÓN A LA CIENCIA DE DATOS

(AULA 25) 25.ABRIL.2022

Agrupamiento espectral

Dada una matriz de datos $\mathbb{X} \in \mathbb{R}^{n \times d}$, se constuye una estructura de grafo G , que marca las relaciones de afinidad o semejanza entre los datos.

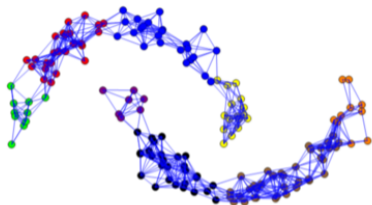
Para ello, construimos una matriz de afinidad $W = (w_{ij})$, que mide la similitud entre los datos $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$. Esta matriz no solo codifica la estructura del grafo G , sino el grado de semejanza entre pares de datos.

El agrupamiento espectral (*spectral clustering*) basa su idea de clasificación en estudiar el grafo G , mediante el espectro de su matriz Laplaciana L o Laplaciana normalizada \mathcal{L} : los autovalores de \mathcal{L} inducen relaciones de agrupamiento entre los datos.

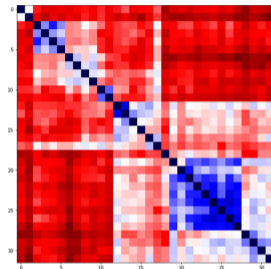
Obs. Las ideas son similares al *spectral embedding*.

Agrupamiento espectral

Datos
 $\mathbb{X} = (\mathbf{x}_i)$



Embedding + Grafo



Matriz de afinidad o
similitud W

- $W = (w_{ij})$, donde w_{ij} = similitud entre el v3rtice \mathbf{x}_i y \mathbf{x}_j .
- W es sim3trica, no-negativa ($w_{ij} \geq 0$).

Agrupamiento espectral

Observaciones:

- El grafo G puede construirse usando ideas de conectar vecinos cercanos. Por ejemplo, se pre-establece una métrica $d(\cdot, \cdot)$ en \mathbb{R}^d , y para cada \mathbf{x}_i se define una vecindad

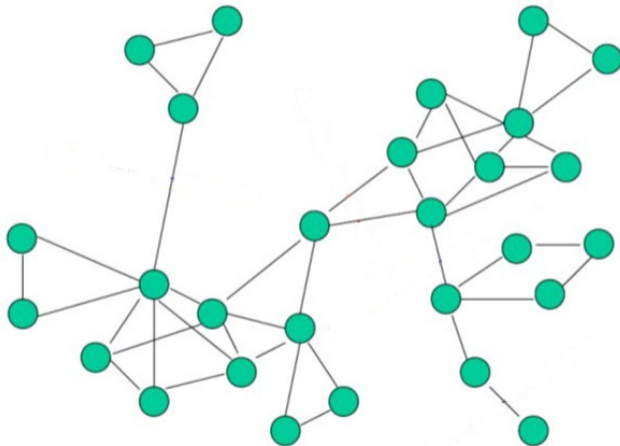
$$\mathcal{N}_r(i) = \{j : d(\mathbf{x}_j, \mathbf{x}_i) \leq r\}.$$

- Otra estrategia consiste en conectar aquellos vértices que están entre los k -vecinos más cercanos: Se predefine k , y se construyen las vecindades

$$\mathcal{N}_k(i) = \{j : \mathbf{x}_j \text{ es } k\text{-vecino más cercano a } \mathbf{x}_i\} = k\text{- argmin}_j d(\mathbf{x}_j, \mathbf{x}_i).$$

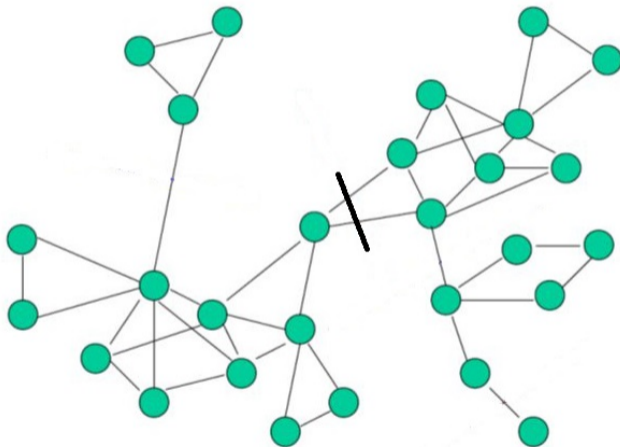
Agrupamiento espectral

Idea: Hallar dónde hacer el corte óptimo.



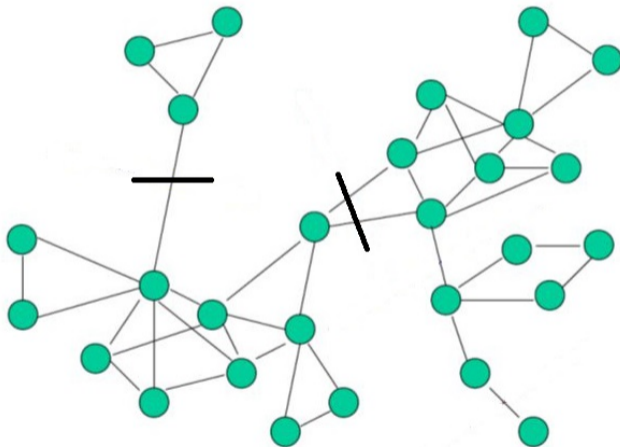
Agrupamiento espectral

Idea: Hallar dónde hacer el corte óptimo.



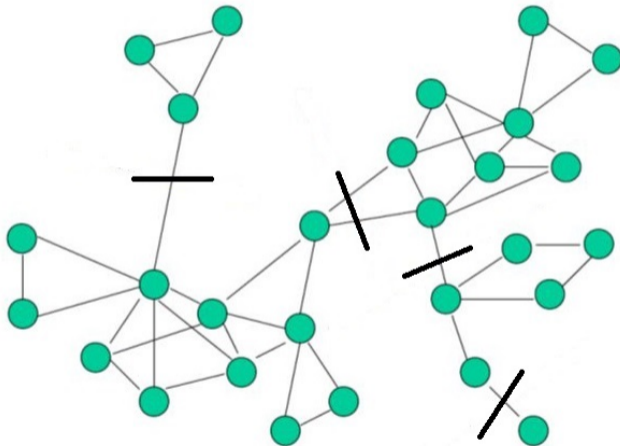
Agrupamiento espectral

Idea: Hallar dónde hacer el corte óptimo.



Agrupamiento espectral

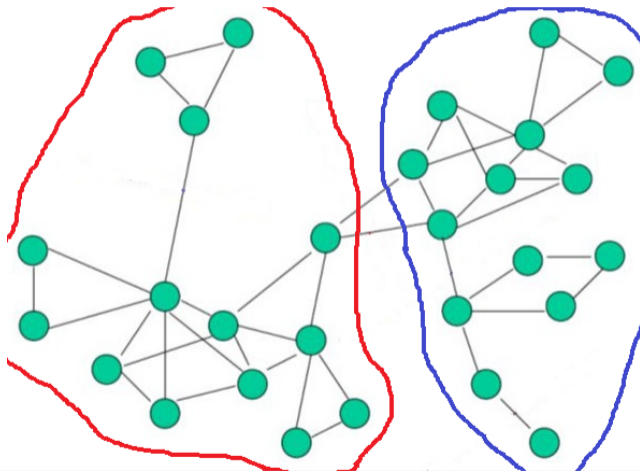
Idea: Hallar dónde hacer el corte óptimo.



- (Ford-Fulkerson, 1956). Algoritmo para calcular flujo máximo.
- (Edmonds-Karp, 1972). Implementación eficiente de Ford-Fulkerson.
- (Fiedler, 1973). Conectividad algebraica en grafos.
Método del vector de Fiedler.
- (Shi-Malik, 2000). Algoritmo *Normalized-Cuts*.
Exportan la idea de clustering espectral a visión computacional.
- (Boykov-Kolmogorov, 2001). Versión extendida de *Normalized-Cuts*.

Graph-Cuts

Caso más simple: segmentación binaria (dos componentes).



Sea G un grafo conexo, W su matriz de afinidad.

Queremos encontrar una partición óptima del grafo (B, B') en dos componentes conexas $B \neq \emptyset$ y $B' \neq \emptyset$.

Este corte óptimo (B, B') puede encontrarse como la solución del problema de optimización

$$\min_{B, B'} \text{Cut}(B, B') = \min_{B, B'} \sum_{i \in B, j \in B'} w_{ij} - \sum_{i, j \in B} w_{ij} - \sum_{i, j \in B'} w_{ij}. \quad (1)$$

$$\begin{aligned}\min_{B, B'} \text{Cut}(B, B') &= \min_{B, B'} \sum_{i \in B, j \in B'} w_{ij} - \sum_{i, j \in B} w_{ij} - \sum_{i, j \in B'} w_{ij} \\ &= \min_{B, B'} - \sum_{i \in B, j \in B'} (1)(-1)w_{ij} - \sum_{i, j \in B} (1)^2 w_{ij} - \sum_{i, j \in B'} (-1)^2 w_{ij} \\ &= \min_{\mathbf{u} \in \{-1, 1\}^n} -\mathbf{u}^T \mathbf{W} \mathbf{u}\end{aligned}$$

con $\mathbf{u} \in \{-1, 1\}^n$ un vector indicador ($u_i = 1$ si $\mathbf{x}_i \in B$, y $u_i = -1$ si $\mathbf{x}_i \in B'$).
El problema (1) puede resolverse hallando el corte óptimo

$$\min_{\mathbf{u} \in \{-1, 1\}^n} -\mathbf{u}^T \mathbf{W} \mathbf{u} \tag{2}$$

Sea $d_i = \sum_{j=1}^n w_{ij}$ el grado del vértice \mathbf{x}_i . El vector de grados $\mathbf{d} = (d_1, \dots, d_n)^T \in \mathbb{R}^n$ está dado por $\mathbf{d} = W\mathbf{1}$, donde $\mathbf{1} \in \mathbb{R}^n$ es un vector con entradas 1.

Definimos la matriz $D = \text{diag}(\mathbf{d})$, cuya diagonal contiene los grados de cada vértice.

Definición

El **Laplaciano** del grafo G se define como

$$L = D - W = \text{diag}(\mathbf{d}) - W. \quad (3)$$

Teorema (Segmentación Binaria)

La solución de

$$\min_{\mathbf{u} \in \{-1,1\}^n} \mathbf{u}^T \mathbf{L} \mathbf{u} \quad (4)$$

es un vector indicador de la segmentación binaria de G .

Malas noticias:

El problema (4) es un problema NP-completo.

Propiedades de L :

- L es simétrico y positivo-semidefinido.
- el menor auto-valor de L es 0, con auto-vector asociado $\mathbf{1}$.
- la multiplicidad de 0 como auto-valor de L corresponde al número de componentes conexas de G .

Vamos a considerar una relajación del problema anterior. En lugar de tomar $\mathbf{u} \in \{-1, 1\}^n$, tomamos $\mathbf{u} \in \mathbb{R}^n$:

$$\min_{\mathbf{u} \in \mathbb{R}^n} \mathbf{u}^T L \mathbf{u}.$$

Teorema (Fiedler, 1973)

Sea \mathbf{u}^* la solución de

$$\min_{\mathbf{u} \in \mathbb{R}^n} \mathbf{u}^T L \mathbf{u}. \quad (5)$$

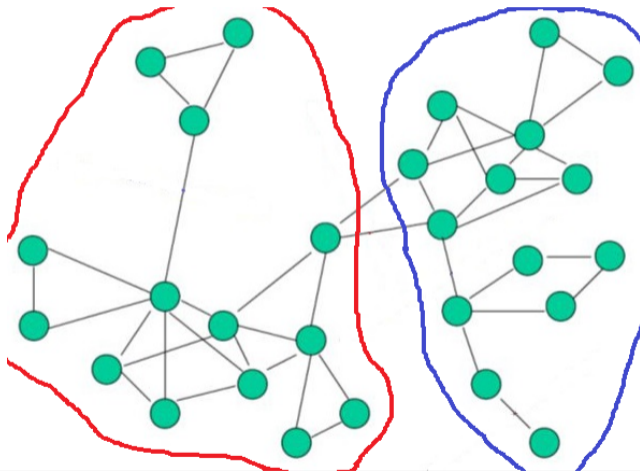
Entonces, el signo $\text{sign}(\mathbf{u}^*)$ es un indicador de la segmentación binaria de G .

Más aún, esta solución está dada por el vector de Fiedler $\mathbf{u}^* = \mathbf{v}_{n-1}$, el auto-vector asociado al segundo menor auto-valor λ_{n-1} de L .

(Obs: consecuencia del Teorema de Eckhart-Young.)

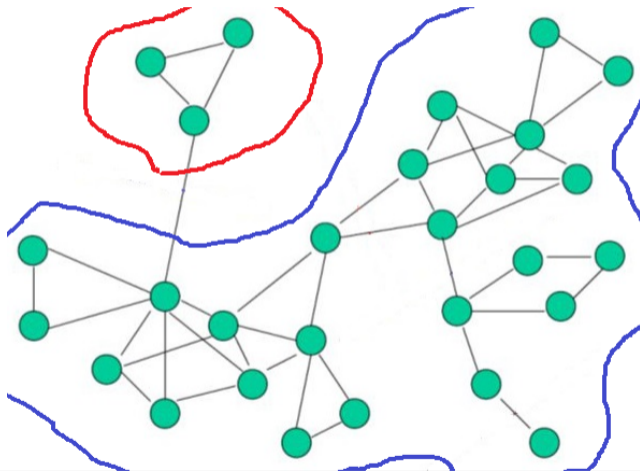
Graph-Cuts

El método de Fiedler funciona bien cuando la partición es balanceada.



Graph-Cuts

Presenta problemas cuando la partición no es balanceada.



Normalized Graph-Cuts

J Shi y J. Malik (2000) introducen el método de *normalized graph-cuts* o NCuts.

La idea es considerar un factor que normalice el tamaño de los clúster, para que sean ambos comparables.

En lugar de (1), se propone la función de costo *NCut*

$$\min_{B, B'} nCut(B, B') = \min_{B, B'} \left(\frac{\sum_{i \in B, j \in B'} w_{ij}}{\sum_{i \in B, j \in V} w_{ij}} + \frac{\sum_{i \in B', j \in B} w_{ij}}{\sum_{i \in B', j \in V} w_{ij}} \right) \quad (6)$$

Teorema (Shi-Malik, 2000)

Dada la función de costo $NCut$ (6) Entonces

i) La solución $\mathbf{u} \in \mathbb{R}^n$ del problema de auto-valores generalizado

$$(D - W)\mathbf{u} = \lambda D\mathbf{u}, \quad (7)$$

es una solución relajada del problema de segmentación binaria (6).

ii) La solución de (7) se reduce al problema de auto-valores

$$D^{-1/2}(D - W)D^{-1/2}\mathbf{u} = \lambda\mathbf{u}, \quad (8)$$

Definición

El **Laplaciano normalizado** del grafo G está dado por

$$\mathcal{L} = D^{-1/2}LD^{-1/2} = D^{-1/2}(D - W)D^{-1/2}. \quad (9)$$

Propiedades de \mathcal{L} :

- \mathcal{L} es simétrico y positivo-semidefinido.
- el menor auto-valor de \mathcal{L} es 0, con auto-vector asociado $D^{1/2}\mathbf{1}$.
- la multiplicidad de 0 como auto-valor de \mathcal{L} corresponde al número de componentes conexas de G .

Teorema (Shi-Malik + Fiedler, 2000)

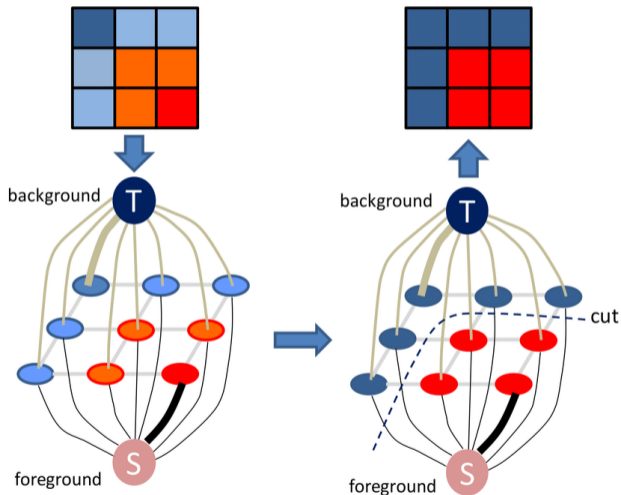
Sea \mathbf{u}^* la solución del problema de auto-valores $n\text{Cut}$

$$\mathcal{L}\mathbf{u} = \lambda\mathbf{u},$$

que minimiza el funcional $N\text{Cuts}$. Entonces, el signo $\text{sign}(\mathbf{u}^*)$ es un indicador de la segmentación binaria de G .

Más aún, esta solución está dada por el vector de Fiedler $\mathbf{u}^* = \mathbf{v}_{n-1}$, el auto-vector asociado al segundo menor auto-valor λ_{n-1} de \mathcal{L} .

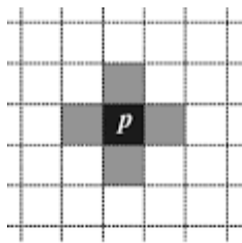
NCuts en imágenes



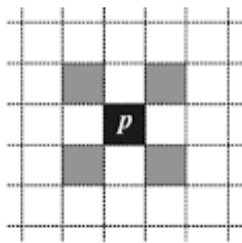
Píxeles vecinos

Vamos a aprovechar esta estructura "natural" de píxeles en una imagen I .
El grafo G asociado será el de píxeles vecinos

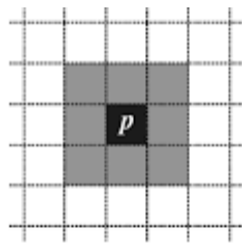
$$\mathcal{N}_c^p(\mathbf{x}) = \{\text{píxeles } \mathbf{z} \in I : \|\mathbf{z} - \mathbf{x}\|_p \leq c\}$$



(a) $\mathcal{N}_1^1(\mathbf{p})$



(b)

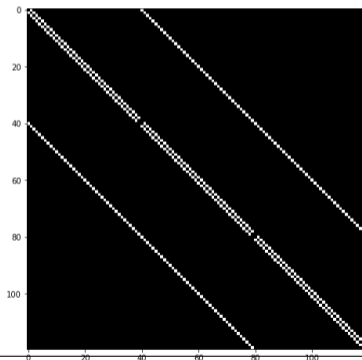


(c) $\mathcal{N}_1^\infty(\mathbf{p})$

Matriz de adyacencia

Definimos la matriz de adyacencia $A = (a_{ij})$ de G como

$$a_{ij} = \begin{cases} 1, & \mathbf{x}_j \in \mathcal{N}_c(\mathbf{x}_i); \\ 0, & \text{caso contrario.} \end{cases}$$



Matrices ralas

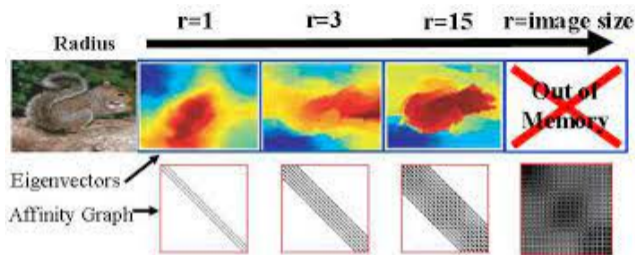


Imagen pequeña: $360 \times 480 = 153,600$ píxeles.

Las matrices A , W y \mathcal{L} son de tamaño $153,600 \times 153,600$.

Aproximadamente 87.9 GB de memoria.

Imagen de $5,000 \times 5,000 = 25,000,000$ píxeles.

Aproximadamente 2.3 millones GB de memoria.

Similaridad

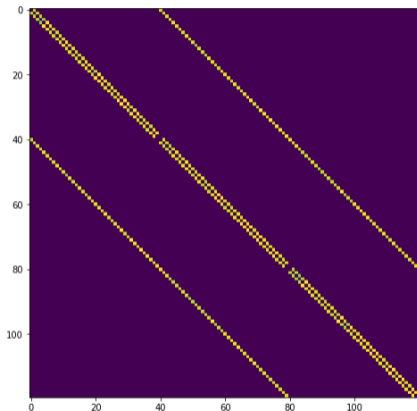
Existen muchas formas de definir la similaridad. Una de las más comunes es usar un *kernel* gaussiano.

$$W_{ij} = \exp \left[- (\alpha \|\mathbf{x}_i - \mathbf{x}_j\|_{color}^2 + \beta \|\mathbf{x}_i - \mathbf{x}_j\|_{espacial}^2 + \gamma \|\mathbf{x}_i - \mathbf{x}_j\|_{textura}^2) / 2\sigma^2 \right],$$

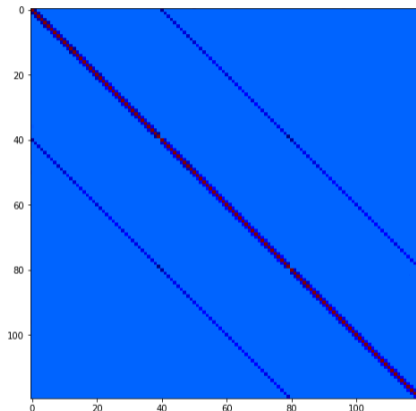
donde

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|_{color} &= \|\text{RGB}(\mathbf{x}_i) - \text{RGB}(\mathbf{x}_j)\|_1, \\ \|\mathbf{x}_i - \mathbf{x}_j\|_{espacial} &= \|(x_i, y_i) - (x_j, y_j)\|_1, \\ \|\mathbf{x}_i - \mathbf{x}_j\|_{textura} &= \|\text{Tex}(\mathbf{x}_i) - \text{Tex}(\mathbf{x}_j)\|, \end{aligned}$$

y $\alpha, \beta, \gamma \geq 0$ son los pesos de cada componente; σ^2 parámetro de escala. W se calcula sólo en aquellas entradas indicadas por A : $W = W_{\text{aff}} \odot A$.



Matriz de afinidad W



Laplaciano normalizado \mathcal{L}

Algorithm 1: Multiclass segmentation NCuts

Result: Multiclass segmentation of I

Given an image I , and its associated graph G ;

while *Stop criterion doesn't hold* **do**

 Construct A , W and \mathcal{L} ;

 Compute second-smallest eigenvalue λ_{n-1} of \mathcal{L} ;

 Compute Fiedler vector \mathbf{v}_{n-1} ;

 Split G in two subgraphs G_1 and G_2 as indicated by $\text{sign}(\mathbf{v}_{n-1})$;

 Repeat algorithm for each subgraph G_j .

end

Segmentación múltiple

¿Cómo separar en más de dos componentes? Tenemos dos alternativas

- Utilizar el método del vector de Fiedler, secuencialmente, para construir un árbol de segmentación binaria. Se aplica nuevamente a cada subgrafo obtenido por cada componente conexa de G .
- Al método anterior se le puede aplicar un criterio de paro en función de la variabilidad de las componentes (e.g. varianza, entropía, ...)
- Si queremos k grupos, se utilizan los k menores autovalores de \mathcal{L} (diferentes de 0), y sus respectivos autovectores $\mathbf{v}_{n-1}, \mathbf{v}_{n-2}, \dots, \mathbf{v}_{n-k}$ se colocan en la matriz reducida $\tilde{\mathbf{X}} = [\mathbf{v}_{n-1}, \dots, \mathbf{v}_{n-k}] \in \mathbb{R}^{n \times k}$.
Luego se aplica cualquier otro método de agrupamiento a la matriz $\tilde{\mathbf{X}}$ (e.g. k -medias).