spectral clustering

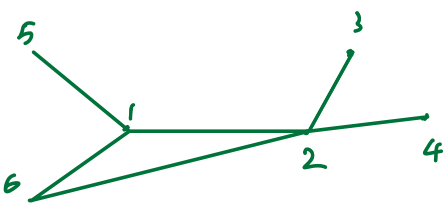
## Spectral Graph theory

- Consider a graph $G=(V,E)$ with $|V|=n$ vertices
  - **Adjacency matrix** $A_{ij}=1$ if node $i$ and node $j$ are connected
    $A_{ij}=0$ otherwise
  - **Degree matrix** (Diagonal matrix) $D_{ii}$ = number of nodes connected to node $i$
    $D_{ii} = \sum_{j=1}^{n} A_{ij}$
  - **Laplacian matrix** The laplacian matrix associated with a graph $G$ is defined as $L_G = D-A$

$$L_G(i,j) = \begin{cases} \deg(V_i) & i=j \\ -1 & (i,j)\in E \\ 0 & \text{otherwise} \end{cases}$$

(i, j) is connected

Example.




adjacency matrix:

$$A=\begin{bmatrix} 0&1&0&0&1&1\\ 1&0&1&1&0&1\\ 0&1&0&0&0&0\\ 0&1&0&0&0&0\\ 1&0&0&0&0&0\\ 1&1&0&0&0&0 \end{bmatrix}$$

Degree matrix:

$$D=\begin{bmatrix} 3&&&&&\\ &4&&&&\\ &&1&&&\\ &&&1&&\\ &&&&1&\\ &&&&&2 \end{bmatrix}$$

Laplacian matrix

$$\begin{bmatrix} 3&1&0&0&1&1\\ 1&4&1&1&0&1\\ 0&1&1&0&0&0\\ 0&1&0&1&0&0\\ 1&0&0&0&1&0\\ 1&1&0&0&0&2 \end{bmatrix}$$

properties of Laplacian matrix

① $L$ is symmetric

② $L$ is positive semi_definite

③ $L\mathbb{1} = 0 \quad L^T\mathbb{1} = 0 \qquad \mathbb{1} = \begin{bmatrix} \vdots \end{bmatrix}$

④ $Lv = \sum_{(i,j)\in E} v(i) - v(j)$

⑤ $v^T L v = \sum_{\substack{(i,j)\in E \\ i<j}} [v(i) - v(j)]^2$

Property ④

$x \in \mathbb{R}^n \quad L \in \mathbb{R}^{n\times n}$

$y = Lx \qquad \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \end{pmatrix} = \underset{L}{\begin{pmatrix} \quad \end{pmatrix}} \underset{x}{\begin{pmatrix} \quad \end{pmatrix}}$

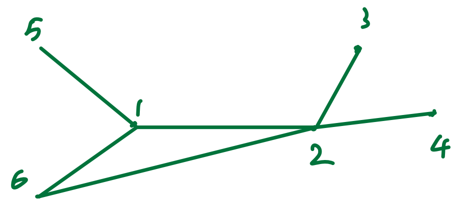
$$L_G(i,j) = \begin{cases} \deg(v_i) & i=j \\ -1 & (i,j)\in E \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \deg(v_i)\cdot x_i - \sum_{\substack{j \\ (i,j)\in E}} x_j$$

$$= \sum_{\substack{j \\ (i,j)\in E}} x_i - \sum_{\substack{j \\ (i,j)\in E}} x_j \qquad ?$$

$$= \sum_{\substack{j \\ (i,j)\in E}} (x_i - x_j)$$

Example




$y = Lx$

$y_1 = (x_1 - x_5) + (x_1 - x_6) + (x_1 - x_2)$

$y_2 = (x_2 - x_6) + (x_2 - x_1) + (x_2 - x_3) + (x_2 - x_4)$

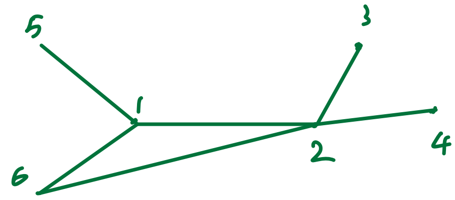
Property ⑤

$$x^T L x = x^T (Lx)$$
$$= \sum_i x_i^T (Lx)_i$$
$$= \sum_i x_i \Big[ \sum_{j \,(i,j)\in E} (x_i - x_j) \Big]$$

$(i,j) \in E \longleftrightarrow i, j$ is connected

$$= \sum_{(i,j)\in E} x_i (x_i - x_j)$$
$$= \sum_{i<j:(i,j)\in E} x_i (x_i - x_j) + x_j (v_j - v_i)$$
$$= \sum_{i<j:(i,j)\in E} (x_i - x_j)^2$$

Example




$$v^T L v = (v_1 - v_6)^2 + (v_1 - v_5)^2 + (v_1 - v_2)^2 + (v_2 - v_3)^2 + (v_2 - v_4)^2 + (v_2 - v_6)^2$$

- Quadratic Form

For any $x \in \mathbb{R}^n$, $x^T L x \geq 0 \Rightarrow L$ is positive semi-definite

$\Rightarrow L^T = (D-A)^T = D^T - A^T = D - A = L$ (D/A is symmetric)

$\Rightarrow$ all $\lambda$ are non-negative

- eigenvalues and eigenvectors of the laplacian matrix

Theorem: The dimension of $E_0$ (eigenspace corresponding to $\lambda = 0$) equals number of
☆ connected components of a graph (a way to find how many connection in the graph)

<u>Prove</u>: The graph Laplacian has at least one zero eigenvalue.

$\Rightarrow$ Assume we have k connected components

$G = [G_1, G_2, \cdots, G_K]$. G has N vertex

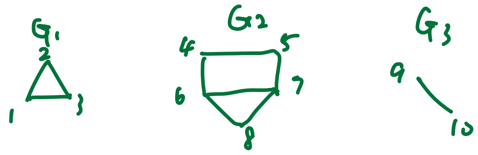
$$V_1 = \begin{bmatrix} x \\ x \\ x \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad V_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ x \\ x \\ x \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad V_3 \cdots$$

$$V_i^T V_j = 0$$

$$V_1^T L V_1 = \sum_{\substack{i<j \\ (i,j)\in E}} [(V_1)_i - (V_1)_j]^2$$

Example:

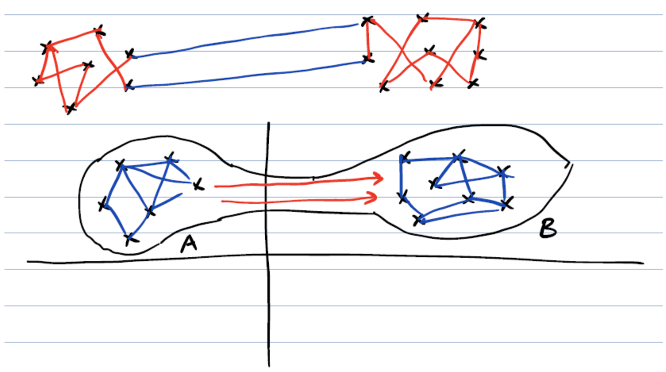
$G =$




Eigenvectors corresponding to zero eigenvalue

$$V_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad V_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad V_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

## General setup




Indicator function: $f$

$f_i = 1$ if $i \in A$

$f_i = -1$ if $i \in B$

Question:

If $i \in A$ and $j \in A$. $\quad f_i - f_j = 0$

$i \in B$ and $j \in B$ $\quad f_i - f_j = 0$

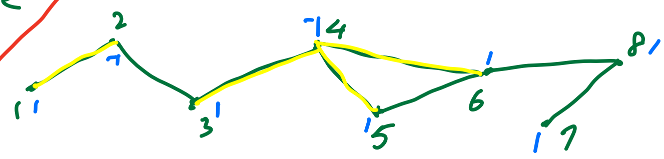
$i \in A$ and $j \in B$
or
$i \in B$ and $j \in A$ $\quad |f_i - f_j| = 2$

Minimization Problem:

$$\min_{f \in [-1,1]^n} \frac{1}{n} \sum_{(i,j)\in E} (f_i - f_j)^2$$

continuous relaxation

Example




$$\underset{f \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{n} f^T L f$$

Assumption:

* Balanced graph

The clusters have the same size

\# of nodes labels 1 = # of nodes labeled -1

## Graph cuts and spectral clustering

- Given graph $G$, the goal is to partition the nodes into two cluster
- You form two clusters A and B as follows:

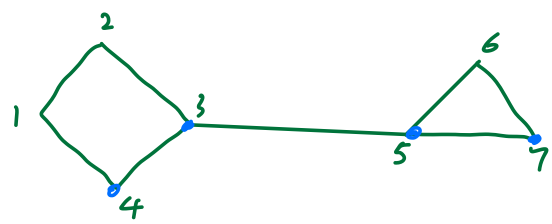
$$f = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

$f_i \equiv$ Cluster of label node $i$

$f_i = 1$ if $i \in A$

$f_i = -1$ if $i \in B$

Example.




one example of $f$ $\quad f = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$

$2^7$ ways to partition the graph

# Graph Cut

Using $f$, graph is clustered into two disjoint sets $A$ and $B$

$$\boxed{\text{cut}(A,B) = \sum_{\substack{i \in A \\ j \in B}} A_{ij}}$$

When is cut(A, B) large ? → There are many edges from A to B

small ? → few edges

**Definition of Graph cut problem**: Given a graph $G = (V, E)$, find a partition of $V$ into two disjoint subsets $A$ and $B$ such that $\text{cut}(A, B)$ is minimized

Question: what is the 'expensive' way to do this ?

A : ① Find all possible partitions

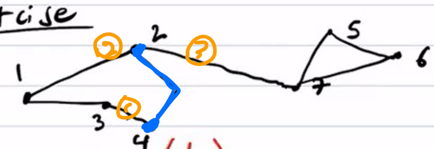
② compute cut for each partition

③ choose the minimum cut

Continuous relaxation    Discrete problem → continuous Problem

Exercise

(i)




(i) $f = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$  What is the cut of the graph corresponding to this partition?

(ii) Find an optimal (minimal) cut to the graph above

A : ① cut = 3

② $A = \{1, 2, 3, 4\}$, $B = \{5, 6, 7\}$ ⇒ Cut = 1

Balanced partition

$$\min_{f \in [-1,1]} \sum_{(i,j) \in E} (f_i - f_j)^2$$

$\downarrow$ Continuous relaxation

$$\min_{f \in \mathbb{R}^n} \sum_{(i,j) \in E} (f_i - f_j)^2 = f^T L f$$

Second assumption:

Number of nodes labeled 1 = number of nodes labeled -1

$$\Rightarrow \sum_{i=1}^{n} f_i = 0 \Rightarrow (f_1\ f_2\ f_3 \cdots f_n) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = 0$$

$$\boxed{\|f\|_2^2 = N}$$

First Step

$$\underset{f}{\operatorname{argmin}}\ f^T L f$$
$$\|f\|_2 = n$$
$$f^T \mathbb{1} = 0$$

Courant-Fischer

$f$ = Second eigenvector of the Laplacian corresponding to the second smallest eigenvalue

= Fiedler vector of a graph.

Second Step

Map it back to [-1,1]

$f \rightarrow \operatorname{sign}(f)$

$f_i \geq 0 \rightarrow 1$

$f_i < 0 \rightarrow -1$

Fiedler vector

$$\operatorname{sign}(f) = \begin{pmatrix} + \\ - \\ + \\ + \\ - \\ - \\ + \end{pmatrix} \begin{matrix} \rightarrow 1 \\ \rightarrow -1 \\ \rightarrow 1 \\ \rightarrow 1 \\ \rightarrow -1 \\ \rightarrow -1 \\ \rightarrow 1 \end{matrix}$$

Spectral embedding

Spectral embedding of node 1

## Apply spectral clustering to real data

$x_1, x_2, \cdots, x_n$ n data points in $\mathbb{R}^d$

Goal: Cluster the points (Different similarity graph)

① $\varepsilon$ - neighbourhood graph

$\varepsilon$ = parameter

$x_i$ and $x_j$ are connected if $\|x_i - x_j\| < \varepsilon$

② $k$-nearest neighbour graph

connect $x_i$ to $x_j$ if $x_j$ is among the $k$-nearest neighbours of $x_i$

③ Fully connected graph

$$\boxed{s(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma^2}\right)}$$

similarity function

(i) What happens when $x_i$ and $x_j$ are far away? $s(x_i, x_j) \approx 0$

(ii) What happen when $x_i$ and $x_j$ are close? $s(x_i, x_j) \approx 1$

Redefine Laplacian as $L = D - W$

All observations we had so far about $L = D - A$ also works in this setting.

$$x^T L x = \sum_{(i,j) \in E} W_{ij} (x_i - x_j)^2$$

### Spectral clustering Algorithm

Input: weighted graph

① compute $L$

$L = D - W$

② compute first $k$ eigenvectors of $L$

$v_1, v_2, \cdots, v_k$ of $L$

$$V = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_k \\ | & | & & | \end{pmatrix} \quad V \in \mathbb{R}^{n \times k}$$

③ Let $y_i$ be the $i^{th}$ row of $V$

$(y_1, y_2, \cdots, y_n)$

④ cluster $y_1, y_2, \cdots, y_n$.

Label of $y_i$ = Label of $x_i$

**Unnormalized spectral clustering**

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number $k$ of clusters to construct

- Construct a similarity graph by one of the ways described in Section 2. Let $W$ be its weighted adjacency matrix.
- Compute the unnormalized Laplacian $L$.
- **Compute the first $k$ eigenvectors $v_1, \dots, v_k$ of $L$.**
- Let $V \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $v_1, \dots, v_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the $i$-th row of $V$.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the $k$-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j |\ y_j \in C_i\}$.