SPECTRAL CLUSTERING

RECALL:

- Weighted graph - (G, W) $|V| = n$.
- Incidence matrix - M.
- Laplacian matrix $(n \times n)$: $L = M^T M$.

$$L_{ij} = \begin{cases} \overset{\sum_{k \neq i} W(v_i, v_k)}{\overset{/\!/}{\deg(v_i)}} & i = j \\ -w(v_i, v_j) & i \neq j \end{cases}$$

all ones

- $L \cdot \underline{1} = \underline{0}$ ($\underline{1}$ = eignevector with eig. value $\lambda = 0$)

BINARY CLUSTERING:

Denote: $\underbrace{\overset{0}{\overset{\shortparallel}{\lambda_1}} \leq \lambda_2 \leq \cdots \leq \lambda_n}_{\text{all eig. values of } L}$

Find $\underline{u}^{(2)}$ = eig. vector for λ_2.

Then: $u_i^{(2)} > 0 \Rightarrow v_i$ cluster 0

$u_i^{(2)} < 0 \Rightarrow v_i$ cluster 1

QUESTIONS:

(1) Why does it work?

(2) What happes when we have more than two clusters?

THE LAPLACIAN MATRIX

$$(G, w) \rightarrow M \rightarrow L = M^T M.$$

ALTERNATIVE DEFINITION:

- W = weight matrix $(n \times n)$

$$W_{i,j} = w(v_i, v_j)$$

- $d_i = \deg(v_i) = \sum_{j \neq i} W_{i,j}$

$$D = \operatorname{diag}(d_1, \_\_\_, d_n) = \begin{pmatrix} d_1 & & & \\ & d_2 & & 0 \\ & & \ddots & \\ 0 & & & d_n \end{pmatrix}$$

- Then: $L = D - W$

$L =$ real values
$+$ symmetric
$+$ positive semi-definite (PSD) $\Rightarrow$ diagonalisable.

$\downarrow$ all eig-vals are ≥ 0

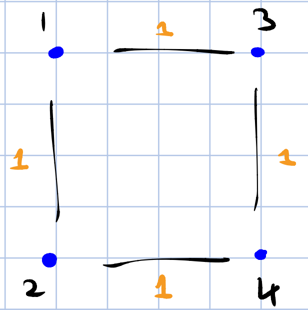
Eigenvalues:

$$\underset{\overset{\shortparallel}{0}}{\lambda_1} \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$$

Q: What is $\operatorname{Ker}(L)$? $(L \cdot \underline{v} = \underline{0})$

(we know $\underline{1} \in \operatorname{Ker}(L)$).

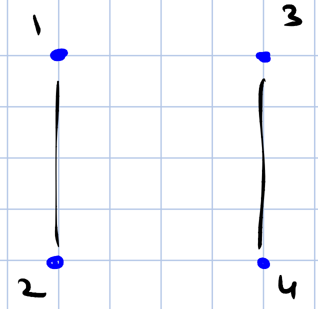
Ex. $G_1 =$



$$\rightarrow L_1 = \begin{array}{cccc} & (1,2)\downarrow & (1,3)\downarrow & (1,4)\downarrow \\ 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{array}$$

$\operatorname{Ker}(L_1) = \operatorname{Span}(\underline{1})$

$G_2 =$



$$\rightarrow \quad L_2 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

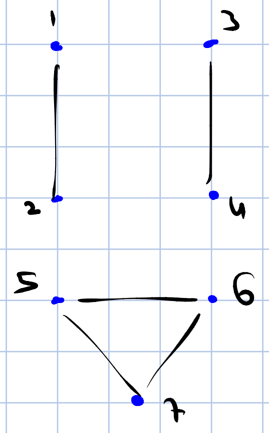
$$L_2 \cdot \overset{v_1}{\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}} = \underline{0} \qquad L_2 \cdot \overset{v_2}{\begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}} = \underline{0}$$

$$\hookrightarrow \quad L_2 \cdot (\alpha v_1 + \beta v_2) = \underline{0}$$

$$\Downarrow$$

$$\mathrm{Ker}(L_2) = \mathrm{Span}(v_1, v_2)$$

$G_3 =$



$$\rightarrow \quad L_3 = \begin{pmatrix} 1 & -1 & & & & & \\ -1 & 1 & & & & 0 & \\ & & 1 & -1 & & & \\ & & -1 & 1 & & & \\ & & & & 2 & -1 & -1 \\ & 0 & & & -1 & 2 & -1 \\ & & & & -1 & -1 & 2 \end{pmatrix}$$

$$v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad v_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$L_3 \cdot v_1 = L_3 \cdot v_2 = L_3 \cdot v_3 = \underline{0}$$

$$\mathrm{Ker}(L_3) = \mathrm{Span}(v_1, v_2, v_3)$$

<u>Conclusion:</u>

$\dim(\mathrm{Ker}(L)) =$ # connected components in G.

THEOREM:

(G, W) - weighted graph

$C_1, C_2, \_\_, C_k \subset V \to$ the connected components of G.

(C_i is connected, and no edges between C_i and any other C_j)

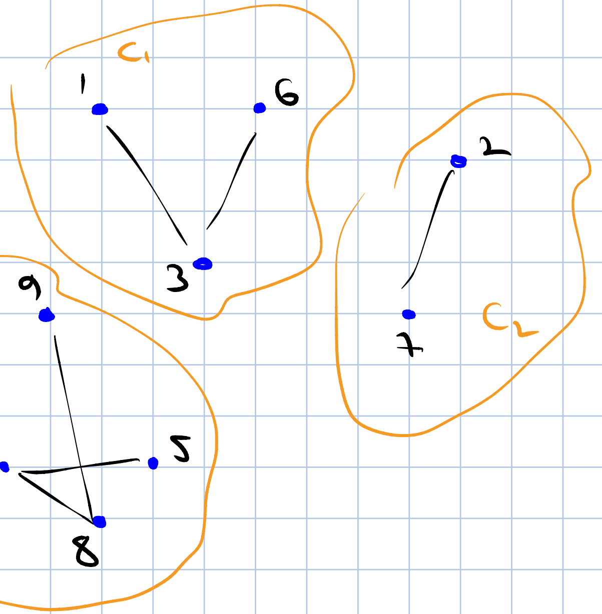
Define: $u^{(1)}, u^{(2)}, \_\_, u^{(k)} \in \mathbb{R}^n$

$$u_i^{(j)} = \begin{cases} 1 & v_i \in C_j \\ 0 & \text{otherwise.} \end{cases}$$

Then: $\mathrm{Ker}(L) = \mathrm{Span}(u^{(1)}, \_\_, u^{(k)})$

$(\dim \mathrm{Ker}(L) = k)$

Ex.



$$(u^{(1)}, u^{(2)}, u^{(3)}) = \begin{array}{cc} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} & \begin{matrix} \leftarrow v_1 \\ \leftarrow v_2 \\ \leftarrow v_3 \\ \vdots \\ \\ \\ \\ \\ \leftarrow v_9 \end{matrix} \end{array} \quad \textcircled{\#}$$

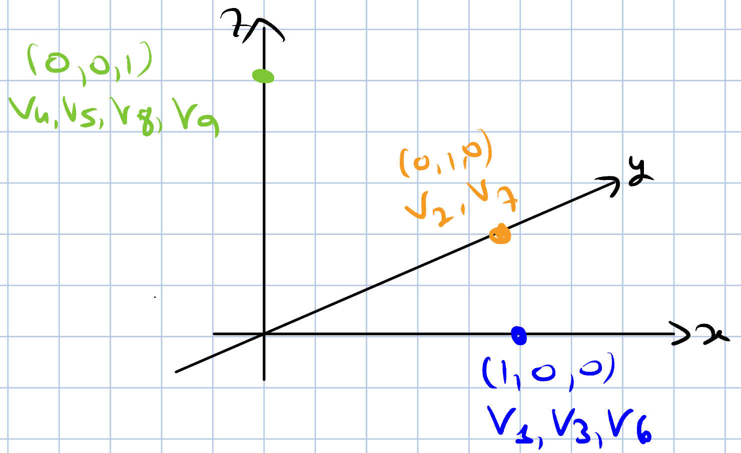
Define: $P : V \to \mathbb{R}^3$

$P(v_i) = R_i = i$-th row of $\textcircled{\#}$

$P(V_1) = P(V_3) = P(V_6) = (1,0,0)$

$P(V_2) = P(V_7) = (0,1,0)$

$P(V_4) = P(V_5) = P(V_8) = P(V_9) = (0,0,1)$



→ apply a clustering algorithm (k-means)

↓

find the components / clusters of G.

Q1:

$Ker(L) = Span(u^{(1)}, \_\_, u^{(k)})$

what if we chose a different basis?

A1:

$U = (u^{(1)}, u^{(2)}, u^{(3)})$

$A \in \mathbb{R}^{3\times3}$ - invertible

New basis: $\tilde{U} = U \cdot A$.

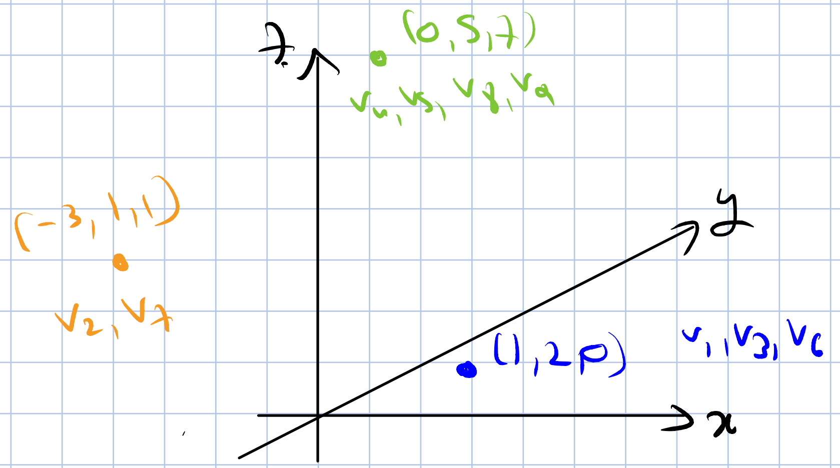
example: $A = \begin{pmatrix} 1 & 2 & 0 \\ -3 & 1 & 1 \\ 0 & 5 & 7 \end{pmatrix}$

$$U = (u^{(1)}, u^{(2)}, u^{(3)}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

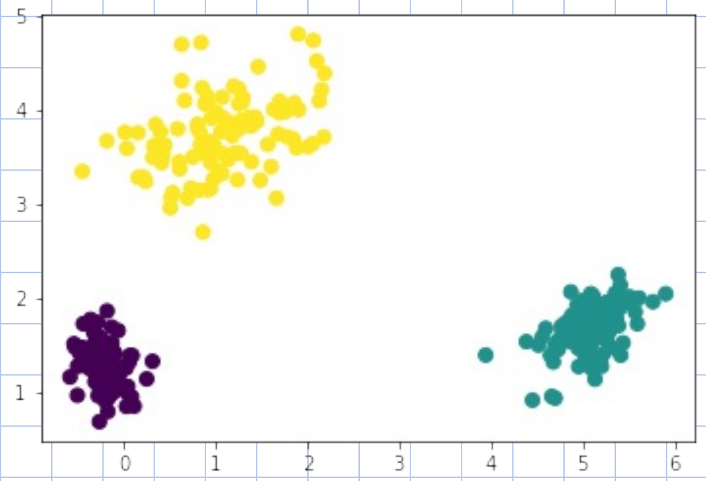
$$A = \begin{pmatrix} 1 & 2 & 0 \\ -3 & 1 & 1 \\ 0 & 5 & 7 \end{pmatrix}$$

$$\tilde{U} = U \cdot A = \begin{pmatrix} 1 & 2 & 0 \\ -3 & 1 & 1 \\ 1 & 2 & 0 \\ 0 & 5 & 7 \\ 0 & 5 & 7 \\ 1 & 2 & 0 \\ -3 & 1 & 1 \\ 0 & 5 & 7 \\ 0 & 5 & 7 \end{pmatrix}$$

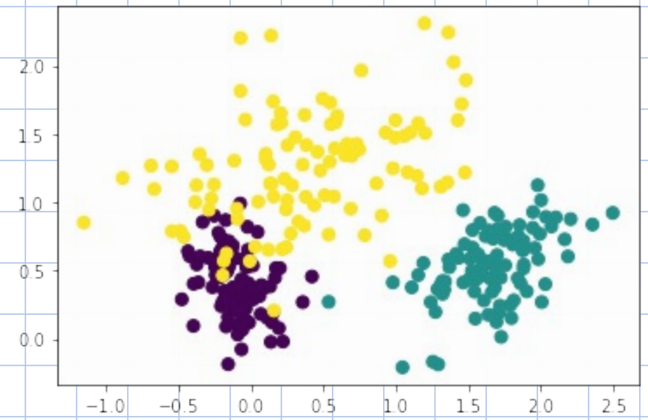
Apply $\mathcal{P}$ the same way:



Q2: This method works well when clusters are well-sparated



What happens if this isn't the case?



→ graph will be connected

$\ker(L) = Sp(\mathbb{1})$

A2:

Assume G is connected.

$$\underset{\substack{\shortparallel\\ 0}}{\lambda_1} < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$$

(G disconnected $\Rightarrow$ $\lambda_1 = \lambda_2 = \dots = \lambda_k = 0$
k = num. components)

Before (G disconnected):

$$\lambda_1 = \lambda_2 = \dots = \lambda_k = 0$$

P = projection on eigen-space of $\lambda_1, \_\_, \lambda_k$ (all 0)

Now: (G connected $\lambda_2 > 0$)

$u^{(j)}$ = eigen-vector for λ_j

$\overset{\mathbb{1}}{\shortparallel}$ no information

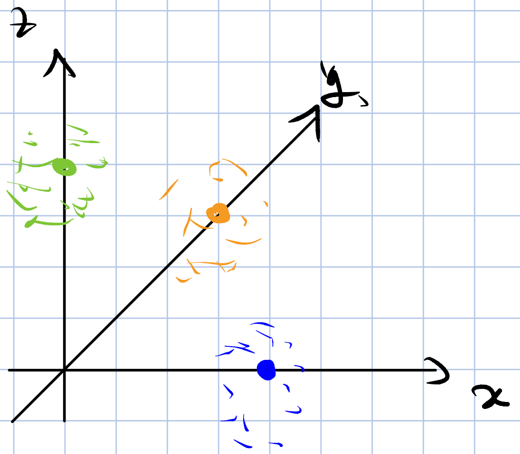
$$U = (\xcancel{u^{(1)}}, u^{(2)}, \_\_, u^{(k)})$$

$$\begin{pmatrix} 1 & 7 & 2 & -1 \\ 1 & 7 & 6 & 7 \\ 1 & -3 & 2 & 17 \\ 1 & & \vdots & \\ \vdots & & & \\ 1 & & & \end{pmatrix}$$

Define: $P : V \to \mathbb{R}^{k-1}$

$P(v_i) = R_i$ = i-th row of U

Expected result:



If we have "nice enough" clusters vertices in the same cluster will lie next to each other.

↓

Run clustering algorithm (k-means)

ALGRORITHM (SPECTRAL CLUSTERING)

Input: $W \in \mathbb{R}^{n \times n}$ - weights.
k = # clusters.

- Construct: $L = D - W$
- Find $\cancel{u^{(1)}}, u^{(2)}, \_\_, u^{(k)} \in \mathbb{R}^n$ - eigen-vecs. corresponding to $\cancel{\lambda_1}, \lambda_2, \_\_, \lambda_k$ (lowest)
- Construct: $U = \left(\underset{\downarrow}{u^{(2)}}, \underset{\downarrow}{\_}, \underset{\downarrow}{u^{(k)}} \right)$
- Map: $v_i \to R_i$ (i-th row) $\in \mathbb{R}^{k-1}$
- Run any clustering algorithm on $\underset{\displaystyle R_1}{\underset{\shortparallel}{R(v_1)}}, \_\_, R(v_n)$ $\_\_, R_n \in \mathbb{R}^{k-1}$