L10: Spectral Clustering

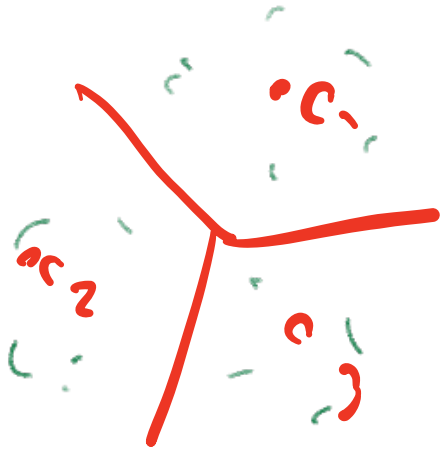
September 22, 2025

Clusterings

Input: Data set $X = \{x_1, \ldots, x_n\}$ and metric/distance $d : X \times X \longrightarrow R_{\geq 0}$.

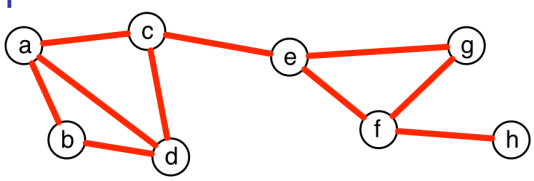
- **Hierarchical Clustering:** Bottom-up - Merge the two closest clusters - (Extended: Density Based)

- **Assignment-based Clustering:** Find centers $C = \{c_1, \ldots, c_k\}$ and update – Voronoi Diagrams



- **Spectral Clustering**:
  1. Top-Down
  2. Input: Graphs (Similarities)

Graphs



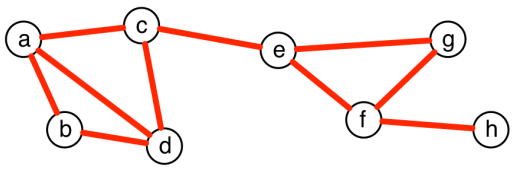
set of nodes/vertices

set of edges

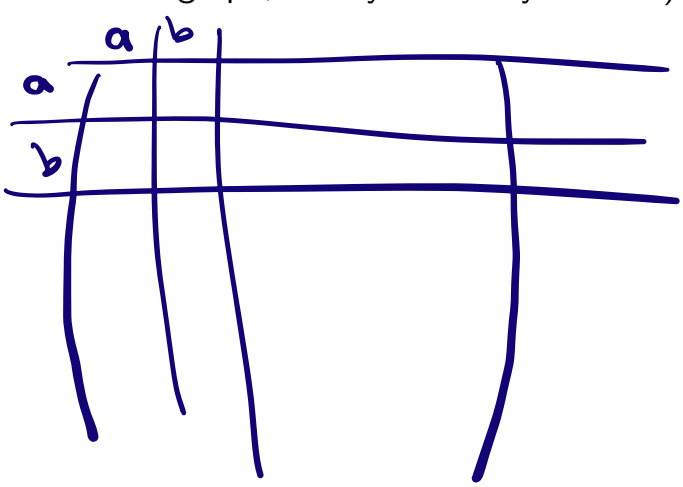
Vertex/Edge Set-style: $G = (V, E)$ where

$V = \{a, b, c, d, e, f, g\}$ and

$E = \Big\{\{a, b\}, \{a, c\}, \{a, d\}, \{b, d\}, \{c, d\}, \{c, e\}, \{e, f\}, \{e, g\}, \{f, g\}, \{f, h\}\Big\}$.



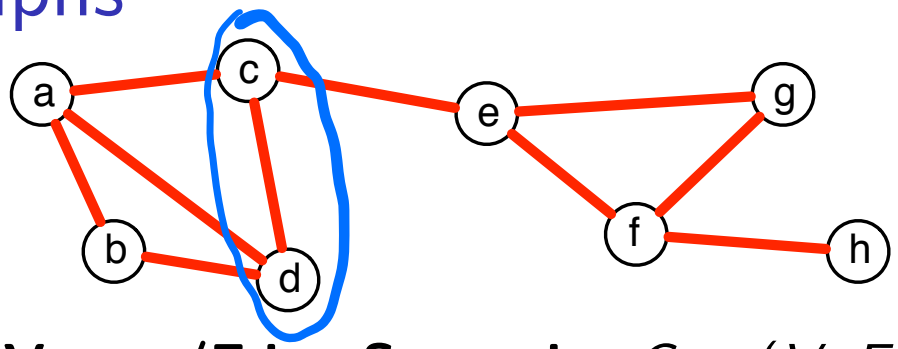
Matrix-Style: As a matrix with 1 if there is an edge, and 0 otherwise. (For a directed graph, it may not be symmetric).



0 - 1

$$a_{ij} = \begin{cases} 1 & ij \in E \\ 0 & o.w \end{cases}$$

Graphs



Vertex/Edge Set-style: $G = (V, E)$ where

$V = \{a, b, c, d, e, f, g\}$ and

$$E = \Big\{\{a,b\},\{a,c\},\{a,d\},\{b,d\},\{c,d\},\{c,e\},\{e,f\},\{e,g\},\{f,g\},\{f,h\}\Big\}.$$

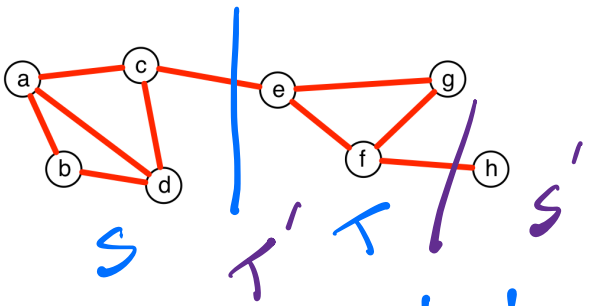
Matrix-Style: As a matrix with 1 if there is an edge, and 0 otherwise.
(For a directed graph, it may not be symmetric).

$G =$

	a	b	c	d	e	f	g	h
a	0	1	1	1	0	0	0	0
b	1	0	0	1	0	0	0	0
c	1	0	0	1	1	0	0	0
d	1	1	1	0	0	0	0	0
e	0	0	1	0	0	1	1	0
f	0	0	0	0	1	0	1	1
g	0	0	0	0	1	1	0	0
h	0	0	0	0	0	1	0	0

$$= \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/3 & 1/4 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Clustering on Graphs



Cluster: A subset of vertices $\phi \neq S \subsetneq V$

Cut(S,T): # edges from S to T = $E(S,T)$

$cut(S,T) = 1$ $cut(S',T') = 1$

Vol(S):

edges with one end in S

NCut(S,T): $Vol(S) = 6$ $Vol(T) = 5$

$Vol(S') = 1$ $Vol(T') = 10$

$$Ncut(S,T) = \frac{cut(S,T)}{Vol(S)} + \frac{cut(S,T)}{vol(T)}$$

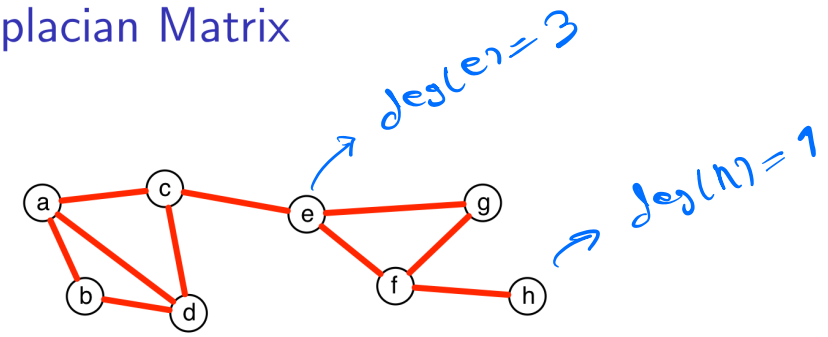
$$Ncut(S,T) = \frac{1}{6} + \frac{1}{5} = 0.367$$

$$Ncut(S',T') = \frac{1}{1} + \frac{1}{10} = 1.1$$

NP-hard problem

Goal: Find the cut (S,T) minimizing th Ncut-number

Laplacian Matrix

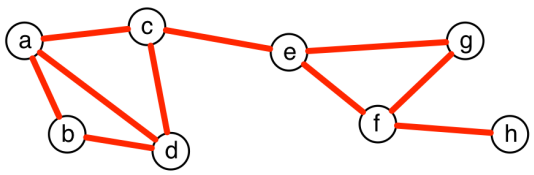


Adjacency

Symmetric

$$A = \underbrace{\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}}_{\text{adjacency}} \quad D = \underbrace{\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{degree}}.$$

Laplacian Matrix



adjacency

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

degree

$$D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Unnormalized Laplacian Matrix

symmetric p.s.d → every eigenvalue is ≥ 0

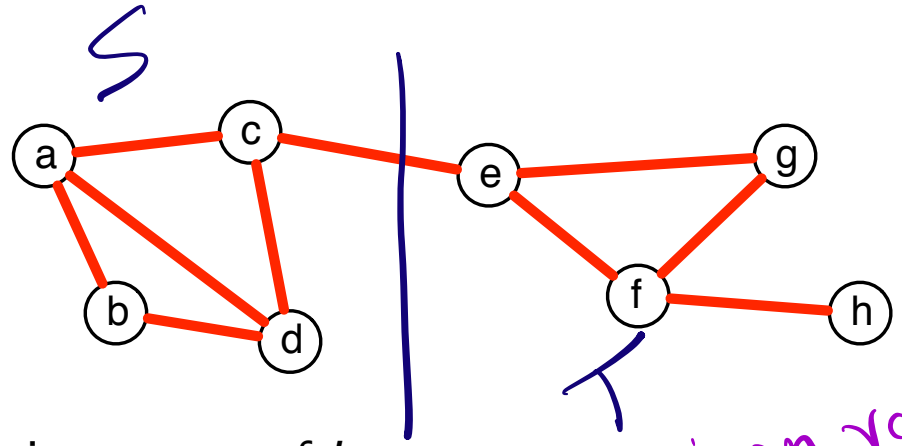
$L_0 = U^T \Lambda U$ → diagonal $\begin{pmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$

$U^T U = U U^T = I$

$$L_0 = D - A = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 3 & -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 3 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}.$$

$\lambda_0 = 0$

Unnormalized Laplacian Matrix



eigenvectors of L_0

$L_0 = U^T \Lambda U$

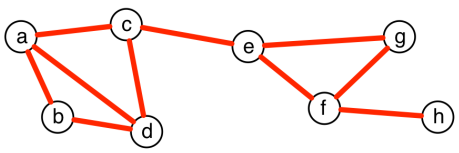
each row of U is an eigen vector of L_0

eigen values

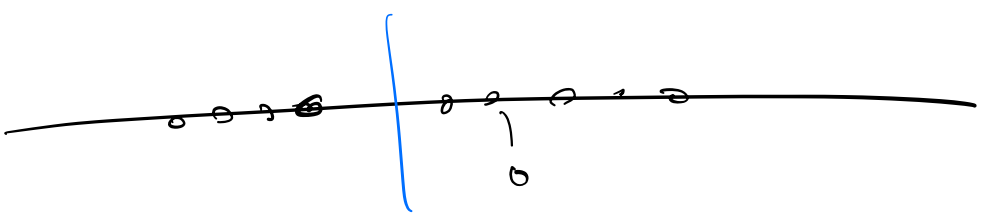
λ	0	**0.278**	1.11	2.31	3.46	4	4.82	
V	$1/\sqrt{8}$	$-.36$	0.08	0.10	0.28	0.25	$1/\sqrt{2}$	a
	$1/\sqrt{8}$	$-.42$	0.18	0.64	$-.38$	0.25	0	b
	$1/\sqrt{8}$	$-.20$	$-.11$	0.61	0.03	$-.25$	0	c
	$1/\sqrt{8}$	$-.36$	0.08	0.10	0.28	0.25	$-1/\sqrt{2}$	d
	$1/\sqrt{8}$	0.17	$-.37$	0.21	$-.54$	$-.25$	0	e
	$1/\sqrt{8}$	0.36	$-.08$	$-.10$	$-.28$	0.75	0	f
	$1/\sqrt{8}$	0.31	$-.51$	$-.36$	$-.56$	0.56	0	g
	$1/\sqrt{8}$	0.50	0.73	0.08	0.11	0.11	0	h

eigen vectors

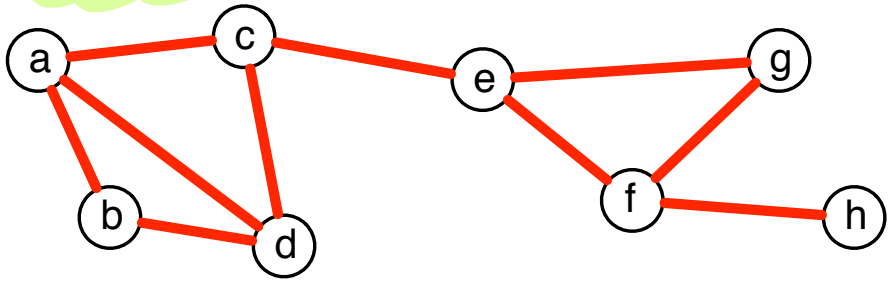
v_2 is the eigen vector for $\lambda_2 = 0.278$



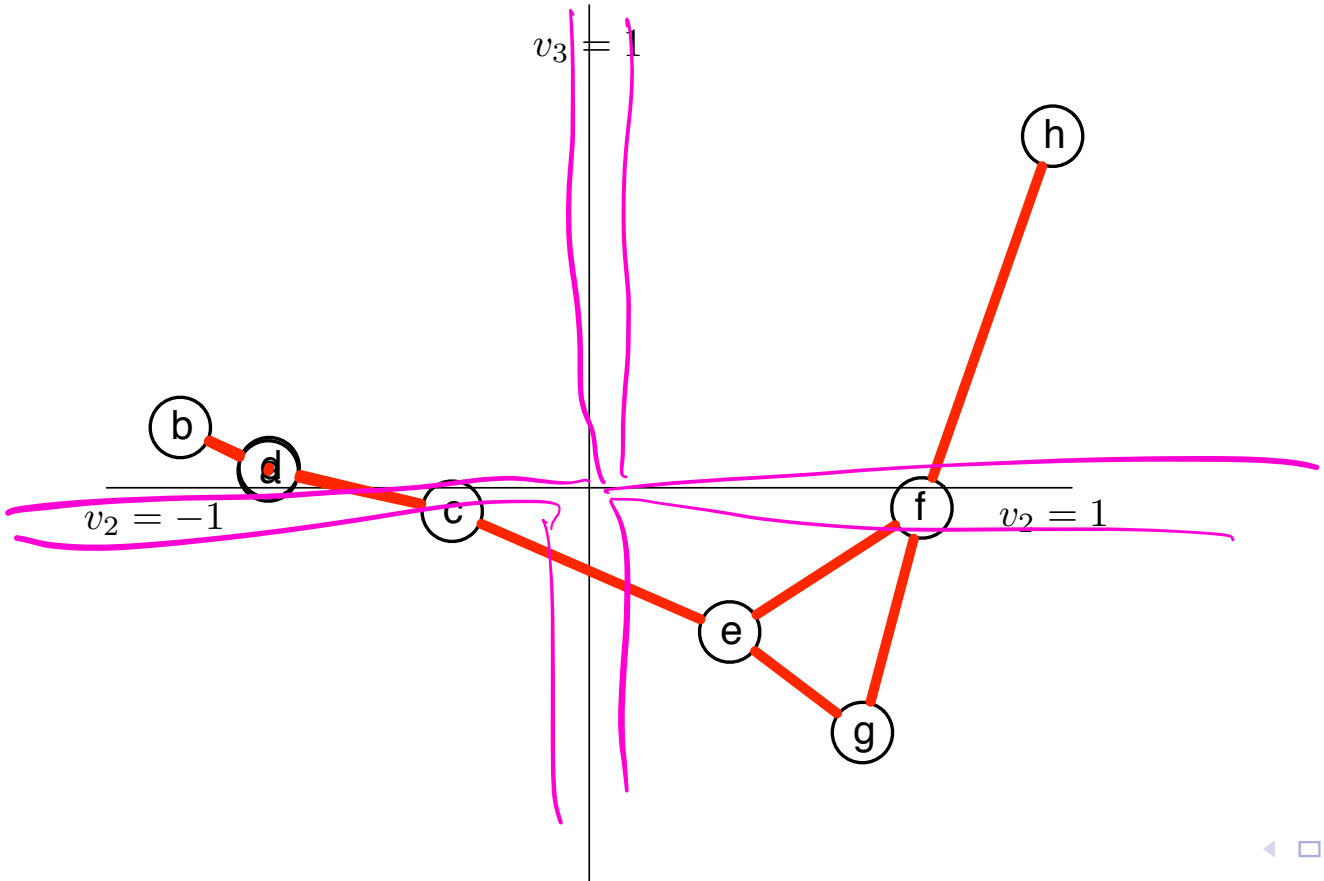
	0.278
a	$-.36$
b	$-.42$
c	$-.20$
d	$-.36$
e	0.17
f	0.36
g	0.31
h	0.50



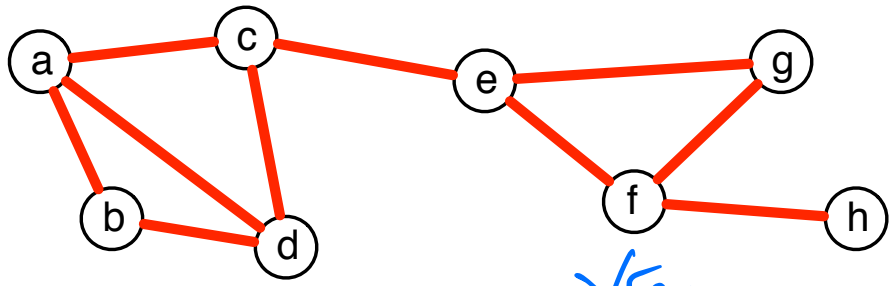
Unnormalized Laplacian Matrix



λ	**0.278**	1.11	
V	$-.36$	0.08	a
	$-.42$	0.18	b
	$-.20$	$-.11$	c
	$-.36$	0.08	d
	0.17	$-.37$	e
	0.36	$-.08$	f
	0.31	$-.51$	g
	0.50	0.73	h
	v_2	v_3	

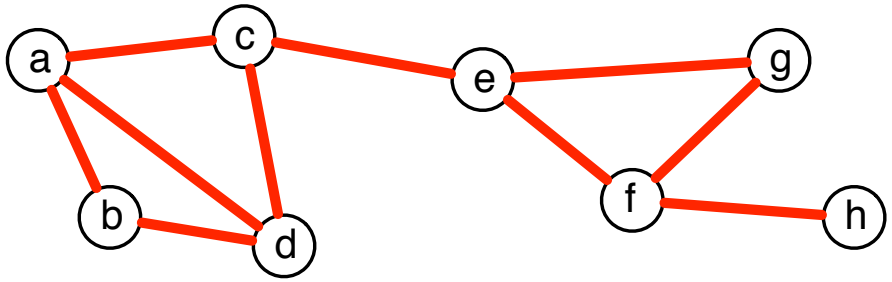


Laplacian Matrix



$$D^{-1/2} = \begin{pmatrix} 0.577 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.707 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.577 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.577 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.577 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.577 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.707 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Laplacian Matrix

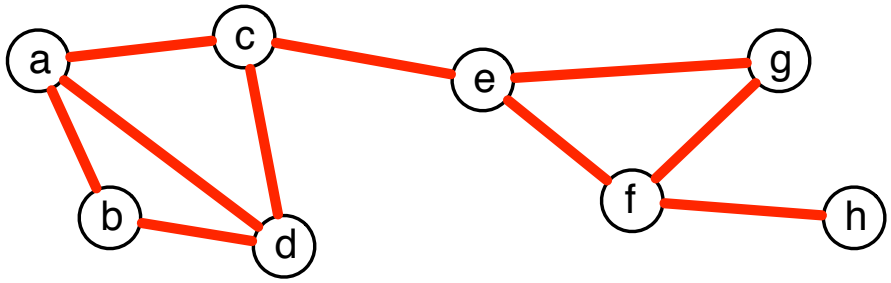


normalized Laplacian

$$L = I - D^{-1/2} A D^{-1/2} =$$

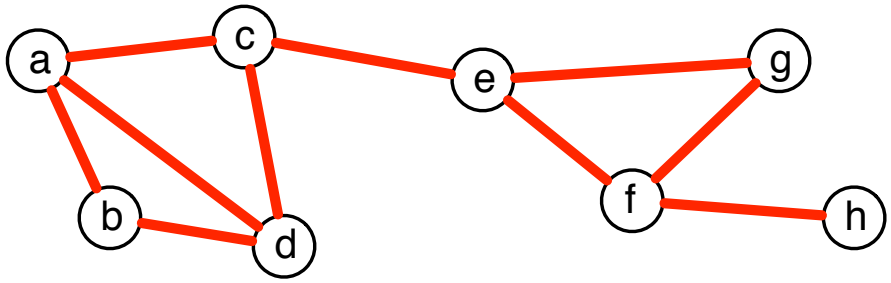
$$\begin{pmatrix}
1 & -0.408 & -0.333 & -0.333 & 0 & 0 & 0 & 0 \\
-0.408 & 1 & 0 & -0.408 & 0 & 0 & 0 & 0 \\
-0.333 & 0 & 1 & -0.333 & -0.333 & 0 & 0 & 0 \\
-0.333 & -0.408 & -0.333 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.333 & 0 & 1 & -0.333 & -0.408 & 0 \\
0 & 0 & 0 & 0 & -0.333 & 1 & -0.408 & -0.577 \\
0 & 0 & 0 & 0 & -0.408 & -0.408 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.577 & 0 & 1
\end{pmatrix}.$$

Laplacian Matrix



eigenvectors of L

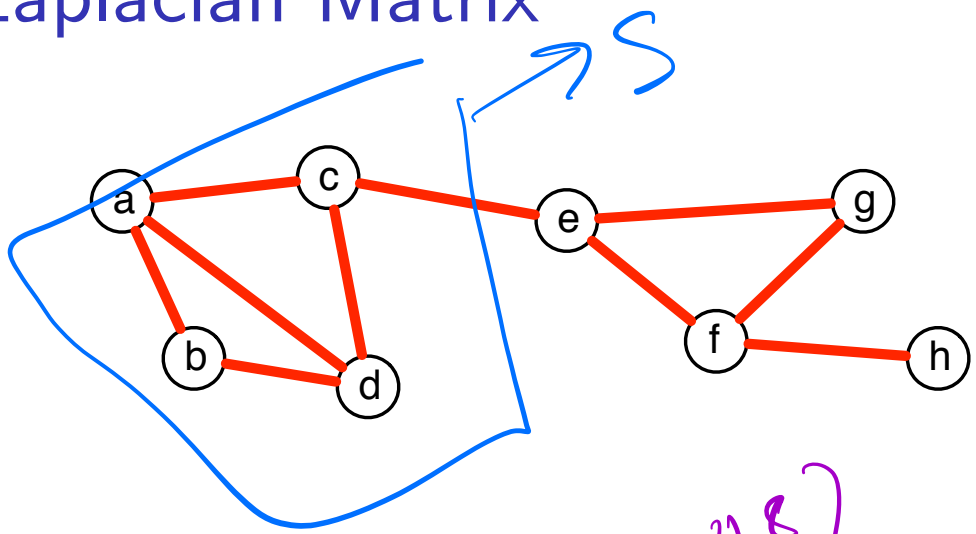
Laplacian Matrix



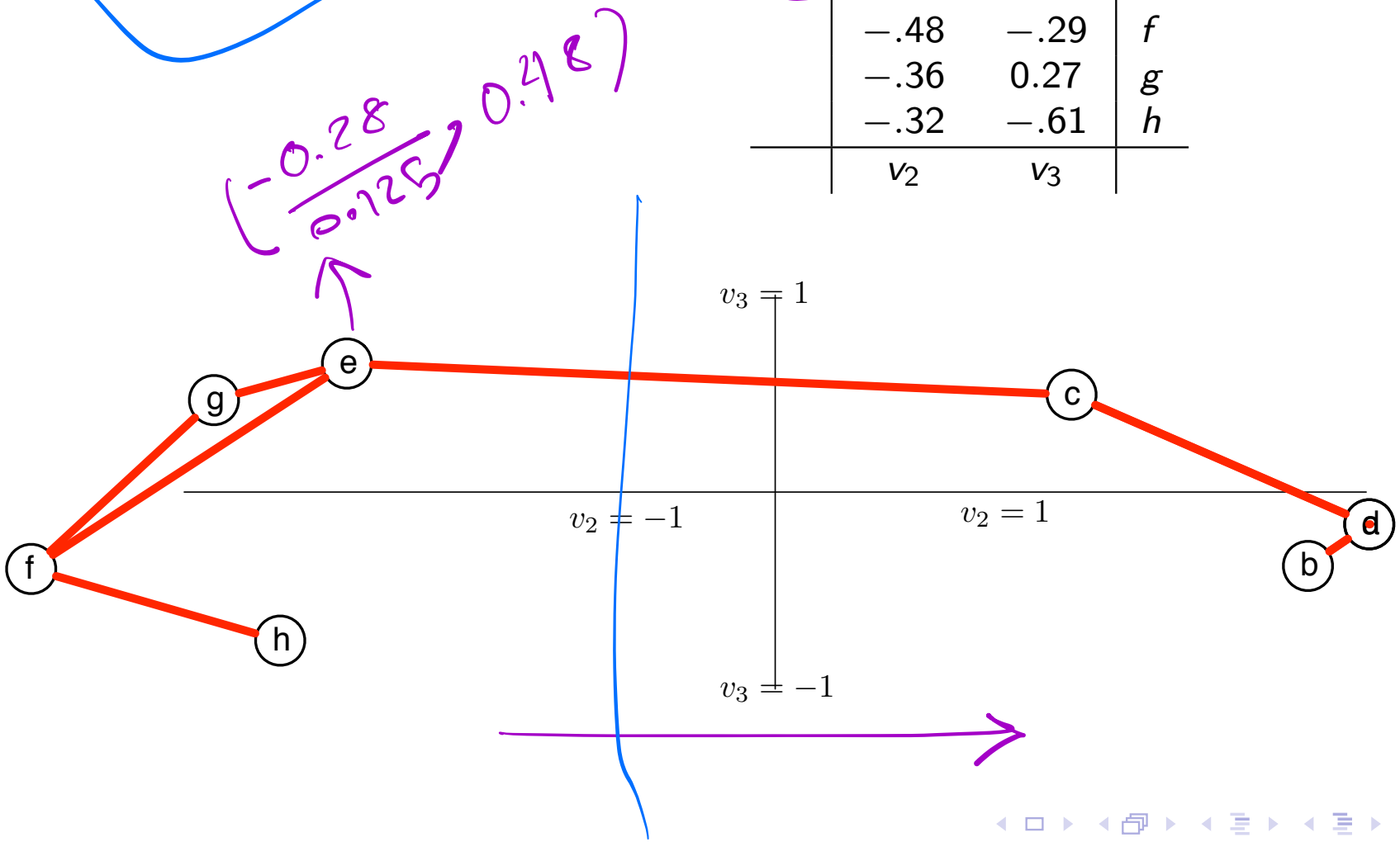
eigenvectors of L

λ	0	**0.125**	0.724	1.00	1.33	1.42	1.66	1.73
V	−.39	0.38	−.09	0.00	0.71	0.26	−.32	0.16
	−.32	0.36	−.27	0.50	0.00	−.51	0.38	−.18
	−.39	0.18	0.36	−.61	0.00	0.03	0.47	−.29
	−.39	0.38	−.09	0.00	−.71	0.26	−.32	0.16
	−.39	−.28	0.48	0.00	0.00	−.57	0.31	0.33
	−.39	−.48	−.29	0.00	0.00	0.05	−.31	−.65
	−.31	−.36	0.27	0.50	0.00	0.51	0.38	−.18
	−.22	−.32	−.61	−.35	0.00	−.07	0.27	0.51

Laplacian Matrix



λ	**0.125**	0.724	
V	0.38	−.09	a
	0.36	−.27	b
	0.18	0.36	c
	0.38	−.09	d
e	−.28	0.48	e
	−.48	−.29	f
	−.36	0.27	g
	−.32	−.61	h
	v_2	v_3	



Affinity Matrix

- Adjacency matrix A is a $0-1$-matrix determining hard similarities.
- For a **Similarity:** $s : X \times X \longrightarrow [0,1]$, define

$$A = (a_{ij})_{n \times n} \qquad a_{ij} = s(x_i, x_j) \in [0,1].$$

- Define

$$D = diag(d_1, \ldots, d_n) \qquad \text{where} \qquad d_i = \sum_{j=1}^{n} a_{ij}$$

and Laplacian Matrix

$$L_0 = D - A$$

Spectral Clustering and the Fiedler Vector

Set-up

Vol(S) = sum of degrees of nodes incident to vertices of S

Build a weighted graph on your data with adjacency matrix A and degree matrix D ($D_{ii} = \sum_j A_{ij}$). The (combinatorial) Laplacian is

$$L_0 = D - A.$$

The (symmetric, normalized) Laplacian is

$$L \;=\; I - D^{-1/2} A D^{-1/2} \;=\; D^{-1/2} L_0 D^{-1/2}.$$

The clustering quality measure is the *normalized cut* (NCut) of a bipartition S, $\bar{S}$:

$$\mathrm{NCut}(S, \bar{S}) \;=\; \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(S)} \;+\; \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(\bar{S})}.$$

Here, $\mathrm{cut}(S, \bar{S})$ counts edge weights crossing the split, and $\mathrm{vol}(S)$ is the sum of degrees in S.

Step 1: Quadratic form for cuts

For any vector $x \in \mathbb{R}^n$,

$$x^\top L_0 x \;=\; \tfrac{1}{2} \sum_{i,j} A_{ij} (x_i - x_j)^2.$$

If x is *piecewise constant* on a cut—say $x_i = a$ for $i \in S$ and $x_i = b$ for $i \in \bar{S}$—then only crossing edges contribute, and

$$x^\top L_0 x \;=\; (a - b)^2 \, \mathrm{cut}(S, \bar{S}).$$

Also,

$$x^\top D x \;=\; a^2 \, \mathrm{vol}(S) \;+\; b^2 \, \mathrm{vol}(\bar{S}).$$

Step 2: Encode NCut as a Rayleigh quotient

Choose constants

$$a = \frac{1}{\mathrm{vol}(S)}, \qquad b = -\frac{1}{\mathrm{vol}(\bar{S})}.$$

Then

$$x^\top D\mathbf{1} \;=\; a\,\mathrm{vol}(S) + b\,\mathrm{vol}(\bar{S}) = 0,$$

so x is "balanced", and

$$\frac{x^\top L_0 x}{x^\top D x} = \frac{\left(\frac{1}{\mathrm{vol}(S)} + \frac{1}{\mathrm{vol}(\bar{S})}\right)^2 \mathrm{cut}(S,\bar{S})}{\frac{1}{\mathrm{vol}(S)} + \frac{1}{\mathrm{vol}(\bar{S})}} = \mathrm{NCut}(S,\bar{S}).$$

Thus minimizing NCut is equivalent to minimizing the Rayleigh quotient

$$R(x) = \frac{x^\top L_0 x}{x^\top D x} \quad \text{over vectors } x \text{ taking two values and satisfying } x^\top D\mathbf{1} = 0.$$

Step 3: Relax the discrete constraint

The two-valued constraint makes the problem combinatorial. Relax it: allow any real $x \neq 0$ with $x^\top D\mathbf{1} = 0$. By the *Rayleigh–Ritz theorem*, the minimizer of $R(x)$ is the eigenvector of the generalized problem

$$L_0 x = \lambda D x$$

corresponding to the second-smallest eigenvalue λ_2 (the smallest, $\lambda_1 = 0$, has eigenvector $\mathbf{1}$).

Equivalently, with $y = D^{1/2}x$, the problem becomes

$$Ly = \lambda y,$$

so the minimizing vector is the second eigenvector y_2—the *Fiedler vector*.

Step 4: From relaxed solution to a cut

Sort the coordinates of the Fiedler vector and try all threshold splits ("sweep cut"); select the split with smallest NCut. This operationalizes the statement: *use v_2 to best cut the graph.*

Summary

1. Encode a cut with a vector constant on each side.
2. NCut becomes a Rayleigh quotient under a balancing constraint.
3. Relax the discrete problem ⇒ generalized eigenproblem $L_0 x = \lambda D x$.
4. Use the second eigenvector (Fiedler vector), and threshold it to recover a cut.