Clustering in Graphs

ACMS 80770: Deep Learning with Graphs

Instructor: Navid Shervani-Tabar

Department of Applied and Comp Math and Stats

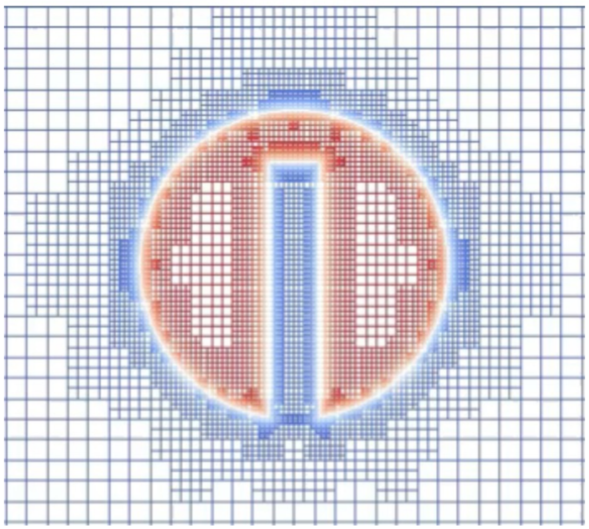
UNIVERSITY OF NOTRE DAME

Graph Partitioning

- A graph partitioning problem divides the graph into K clusters $\mathcal{A}_i$ and $\mathcal{A}_j$ such that $\mathcal{A}_i \cup \mathcal{A}_j = V$ and $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ if $i \neq$ j.

Graph Partitioning

- A graph partitioning problem divides the graph into K clusters $\mathcal{A}_i$ and $\mathcal{A}_j$ such that $\mathcal{A}_i \cup \mathcal{A}_j = V$ and $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ if $i \neq$ j.
- Zoltan, by Sandia National Lab, is a library for load balancing in parallel computing for unstructured and adaptive mesh.



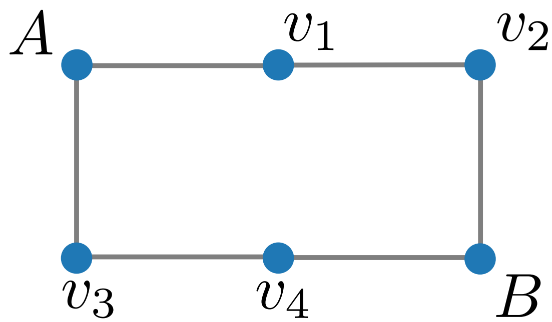
Adaptive Wavelet Collocation Method

Graph Partitioning

- A graph partitioning problem divides the graph into K clusters $\mathcal{A}_i$ and $\mathcal{A}_j$ such that $\mathcal{A}_i \cup \mathcal{A}_j = V$ and $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ if $i \neq$ j.
- Zoltan, by Sandia National Lab, is a library for load balancing in parallel computing for unstructured and adaptive mesh.
- Zoltan performs graph and hypergraph partitioning to distribute computational load.
  - Graph: data objects are nodes and pairwise data dependency are edges.
  - Hypergraph: data objects are nodes and dependencies among set of data objects are hyper-edges.

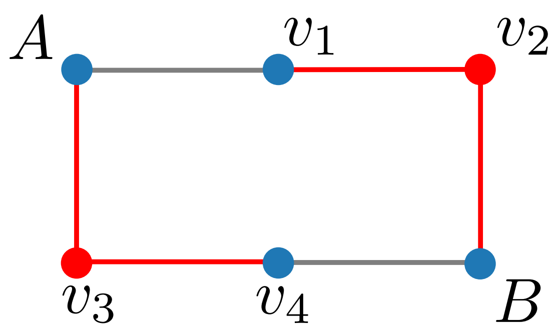
Cut Set

- **Node cut set:** For any pair of nodes A and B in the graph, a node cut set is a set of nodes that if removed (along with their incident edges) will disconnect A and B.



Cut Set

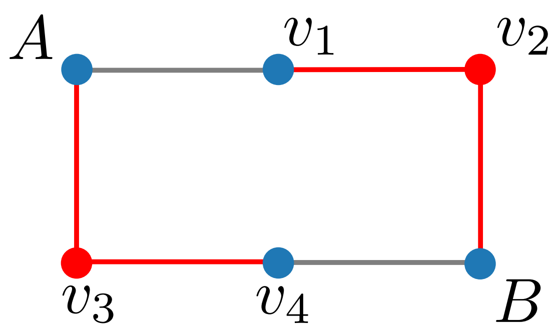
- **Node cut set:** For any pair of nodes A and B in the graph, a node cut set is a set of nodes that if removed (along with their incident edges) will disconnect A and B.



- $\{v_2, v_3\}$

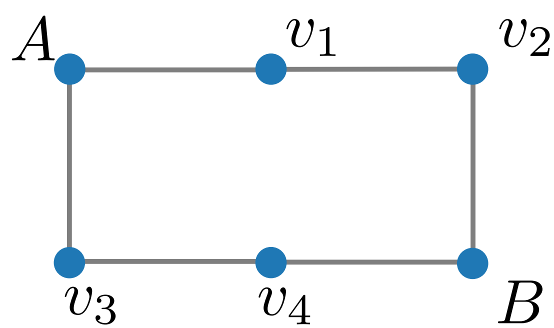
Cut Set

- **Node cut set:** For any pair of nodes A and B in the graph, a node cut set is a set of nodes that if removed (along with their incident edges) will disconnect A and B.



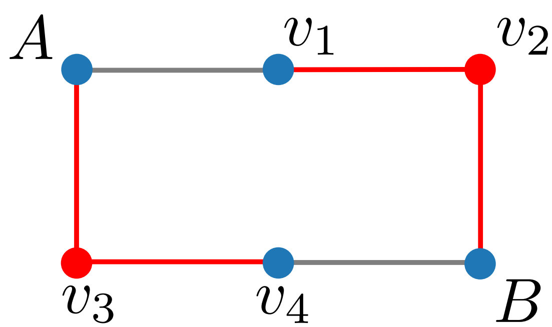
  - $\{v_2, v_3\}$

- **Edge cut set:** The set of edges that if removed, it will disconnect specified nodes A and B.



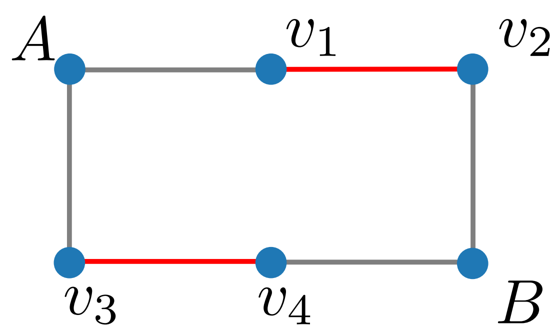
Cut Set

- **Node cut set:** For any pair of nodes A and B in the graph, a node cut set is a set of nodes that if removed (along with their incident edges) will disconnect A and B.



  - $\{v_2, v_3\}$

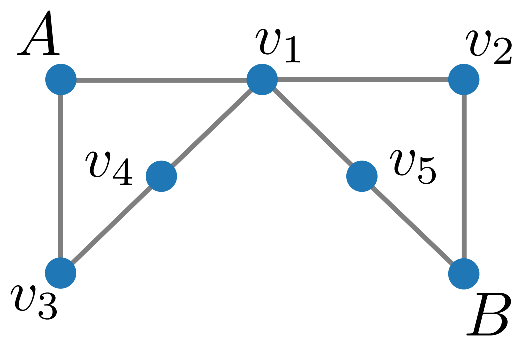
- **Edge cut set:** The set of edges that if removed, it will disconnect specified nodes A and B.



  - $\{(v_1, v_2), (v_3, v_4)\}$

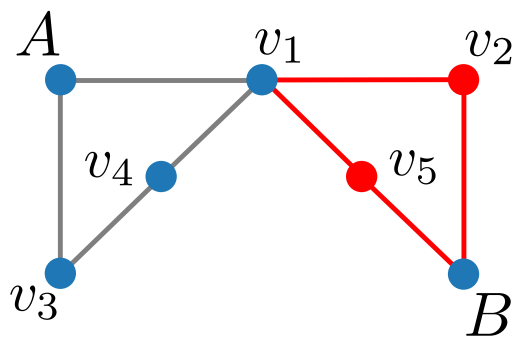
Cut Set

- Cut sets are not unique.



Cut Set

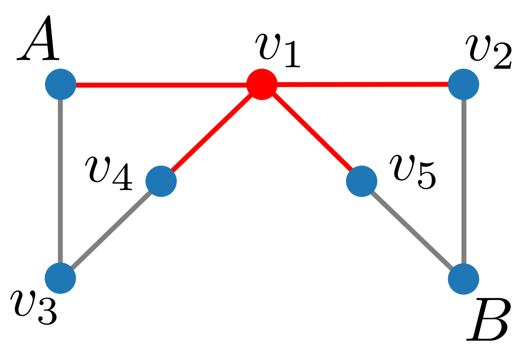
- Cut sets are not unique.



- $\{v_2, v_5\}$

Cut Set

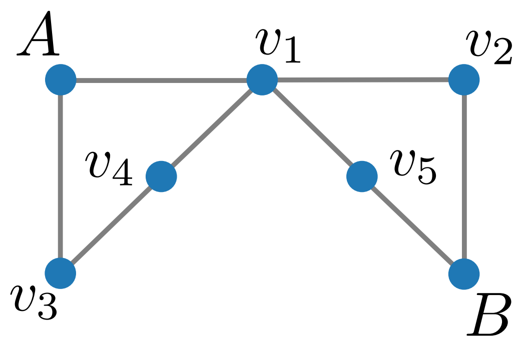
❖ Cut sets are not unique.



➢ $\{v_2, v_5\}$, $\{v_1\}$

Minimum Cut Set

- Cut sets are not unique.



- $\{v_2, v_5\}$, $\{v_1\}$

- **Minimum cut set:** is the minimal set of nodes whose removal will disconnect a pair of nodes A and B in the graph.

- $\{v_1\}$

Cut Size

- One graph partitioning strategy is to divide the graph into K clusters such that the edge cut size is minimized.
- Minimize data passing across 2 processing units.
- We define a cut size as

$$R(\mathcal{A}_1, \ldots, \mathcal{A}_K) = \frac{1}{2} \sum_{k=1}^{K} \left| \left\{ (v_i, v_j) \in E \mid v_i \in \mathcal{A}_k, v_j \in \bar{\mathcal{A}}_k \right\} \right|$$
$$= \frac{1}{2} \sum_{k=1}^{K} \sum_{v_i \in \mathcal{A}_k} \sum_{v_j \in \bar{\mathcal{A}}_k} A_{ij}$$

- Consider partitioning a graph into 2 clusters, such that

$$s_i = \begin{cases} 1 & v_i \in \mathcal{A}_1 \\ -1 & v_i \in \mathcal{A}_2 \end{cases}$$

Cut Size

- Then the indicator function

$$\frac{1}{2}(1 - s_i s_j) = \begin{cases} 0 & v_i, v_j \in \mathcal{A}_1 \text{ or } v_i, v_j \in \mathcal{A}_2 \\ 1 & v_i \in \mathcal{A}_1, v_j \in \mathcal{A}_2 \text{ or } v_j \in \mathcal{A}_1, v_i \in \mathcal{A}_2 \end{cases}$$

  shows if 2 nodes belong to different clusters

- We rewrite the cut size as

$$\begin{aligned} R(\mathcal{A}_1, \mathcal{A}_2) &= \frac{1}{2} \sum_{k=1}^{2} \sum_{v_i \in \mathcal{A}_k} \sum_{v_j \in \overline{\mathcal{A}}_k} A_{ij} \\ &= \frac{1}{4} \sum_{v_i \in V} \sum_{v_j \in V} A_{ij} (1 - s_i s_j) \end{aligned}$$

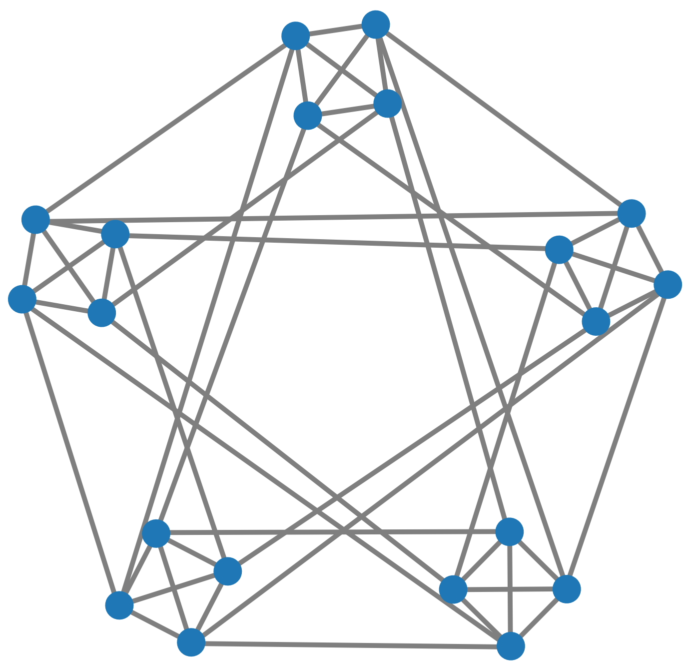
- The clustering problem boils down to finding vector $\boldsymbol{s}$.

Graph Partitioning

- A graph partitioning problem can be formulated as minimizing the edge cut set, AKA mincut problem

$$\min_{\mathcal{A}_k \subset V} R(\mathcal{A}_1, \ldots, \mathcal{A}_K)$$

- This formulation often does not yield optimal clusters

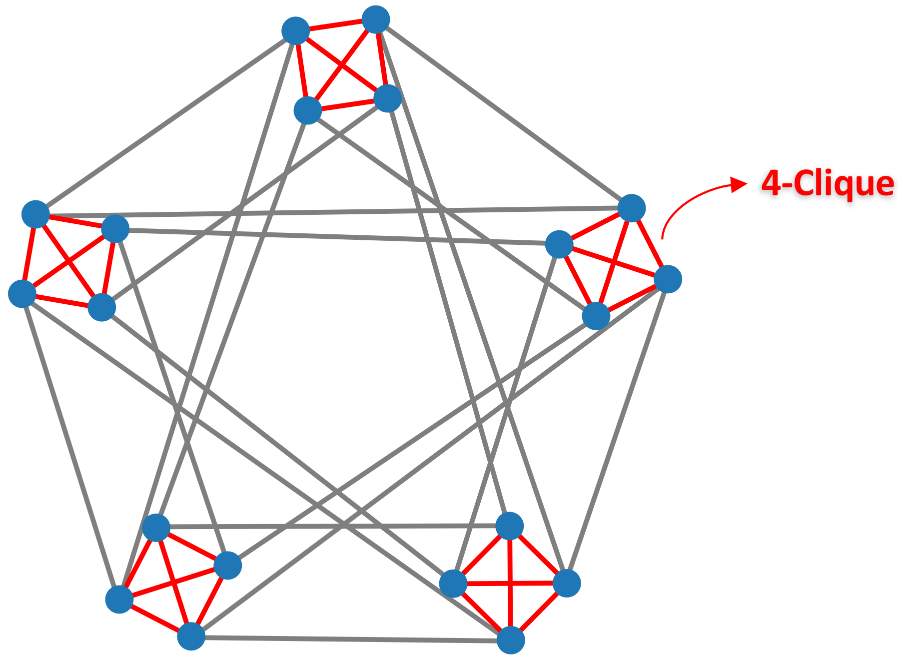


Graph Partitioning

- A graph partitioning problem can be formulated as minimizing the edge cut set, AKA mincut problem

$$\min_{\mathcal{A}_k \subset V} R(\mathcal{A}_1, \ldots, \mathcal{A}_K)$$

- This formulation often does not yield optimal clusters

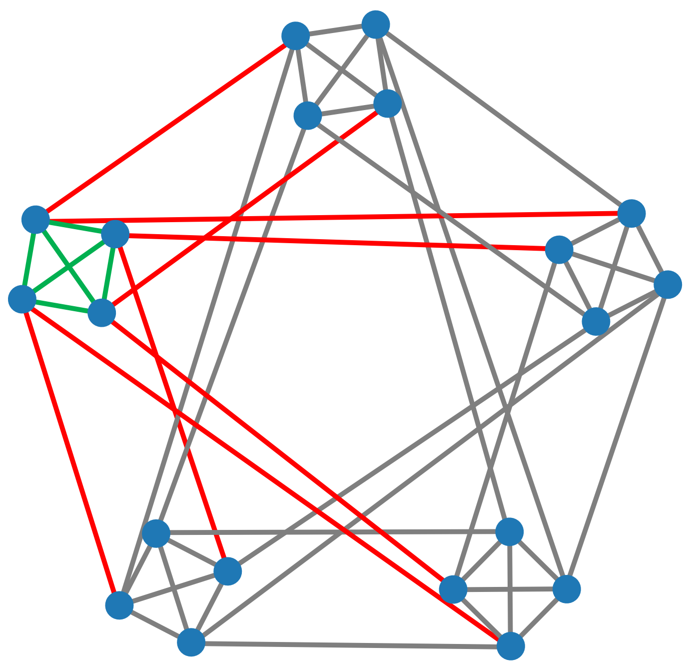


Graph Partitioning

- A graph partitioning problem can be formulated as minimizing the edge cut set, AKA mincut problem

$$\min_{\mathcal{A}_k \subset V} R(\mathcal{A}_1, \ldots, \mathcal{A}_K)$$

- This formulation often does not yield optimal clusters



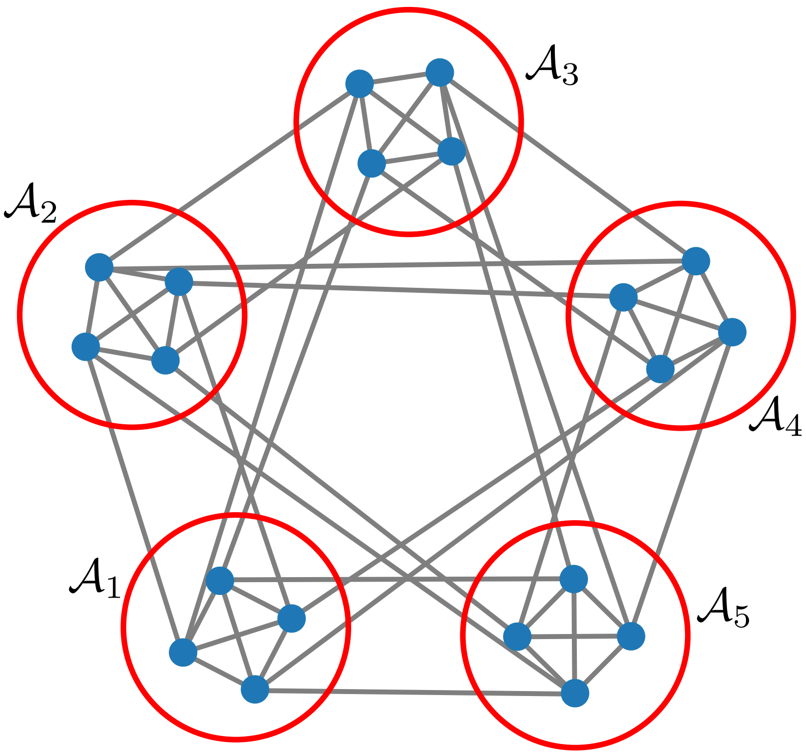
Graph Partitioning

- A graph partitioning problem can be formulated as minimizing the edge cut set, AKA mincut problem

$$\min_{\mathcal{A}_k \subset V} R(\mathcal{A}_1, \ldots, \mathcal{A}_K)$$

- This formulation often does not yield optimal clusters

$R(\mathcal{A}_1, \ldots, \mathcal{A}_5) = 20$



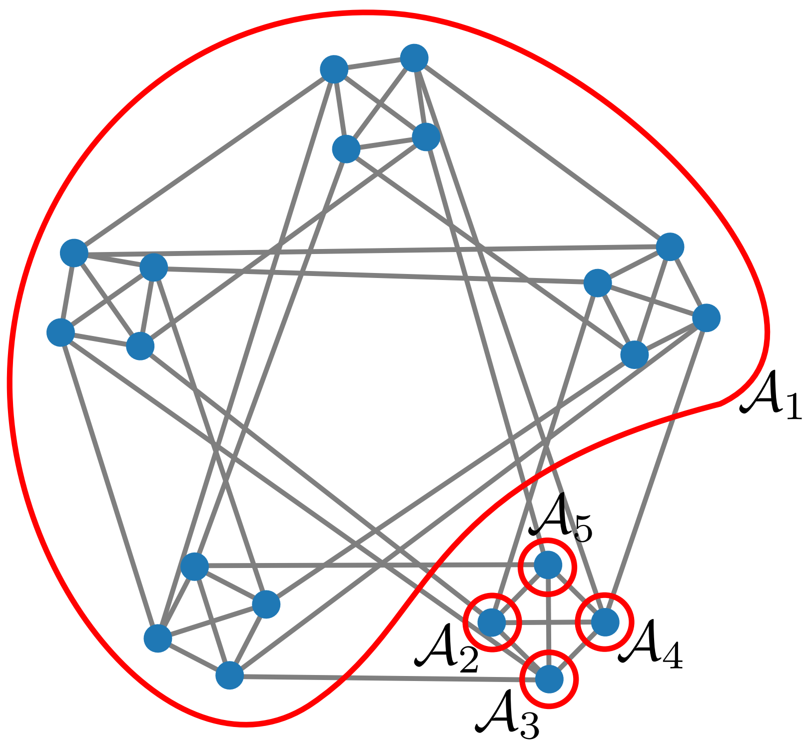
Graph Partitioning

- A graph partitioning problem can be formulated as minimizing the edge cut set, AKA mincut problem

$$\min_{\mathcal{A}_k \subset V} R(\mathcal{A}_1, \ldots, \mathcal{A}_K)$$

- This formulation often does not yield optimal clusters

$R(\mathcal{A}_1, \ldots, \mathcal{A}_5) = 16$

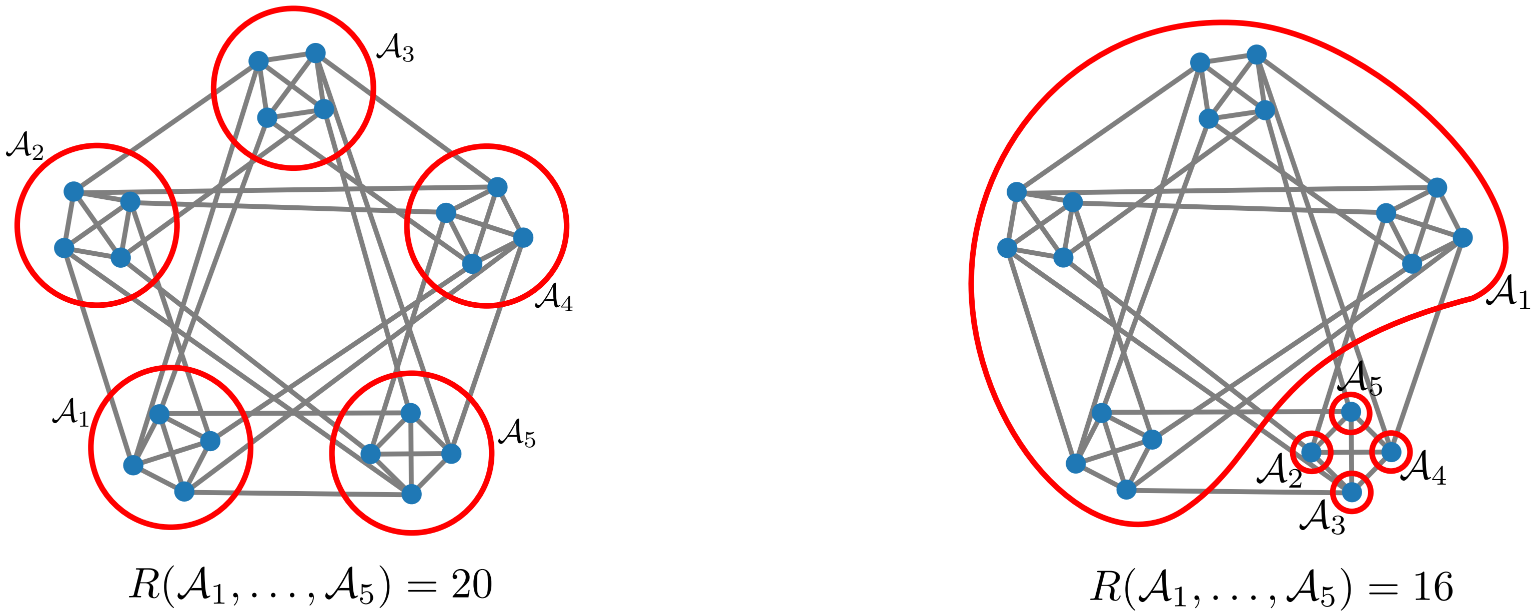


Graph Partitioning

- A graph partitioning problem can be formulated as minimizing the edge cut set, AKA mincut problem

$$\min_{\mathcal{A}_k \subset V} R(\mathcal{A}_1, \ldots, \mathcal{A}_K)$$

- This formulation often does not yield optimal clusters.
- Minimizing this objective results in clusters of size 1.



Ratio Cut

- One strategy is to promote larger clusters while minimizing the cut size.
- This can be done by penalizing small partition sizes.
- Ratio cut size objective function minimizes

$$R^{\text{RatioCut}}\left(\mathcal{A}_1, \ldots, \mathcal{A}_K\right) = \frac{1}{2} \sum_{k=1}^{K} \frac{\left|(u, v) \in \mathcal{E} : u \in \mathcal{A}_k, v \in \overline{\mathcal{A}}_k\right|}{|\mathcal{A}_k|}$$

  where $|\mathcal{A}_k|$ is the number of nodes in of each partition $\mathcal{A}_k$.
- Ratio cut measures the size of a partition by its number of nodes.

Normalized Cut

- Volume of a subgraph $G' = (V', E')$ is the total number of edges incident to nodes v_k of that subgraph.

$$\mathrm{vol}\left(G'\right) = \sum_{v_i \in V'} d_i$$

- Normalized Cut (NCut), alternatively, penalizes the objective function for smaller cluster volumes

$$R^{\mathrm{NCut}}\left(\mathcal{A}_1, \ldots, \mathcal{A}_K\right) = \frac{1}{2} \sum_{k=1}^{K} \frac{\left|(v_i, v_j) \in E : v_i \in \mathcal{A}_k, v_j \in \bar{\mathcal{A}}_k\right|}{\mathrm{vol}\left(\mathcal{A}_k\right)}$$

- NCut measures size of a subset by the number of its edges.
- Minimizing the NCut objective function promotes equal number of edges incident to the nodes of each partition $\mathcal{A}_k$.

Graph Laplacian

- Laplacian is a widely used representation for graphs.
- Laplacian of an undirected graph is a $|V|\times|V|$ symmetric matrix defined as

$$L_{ij} = \begin{cases} d_j & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (v_i, v_j) \in \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

- We can represent this using index notation as

$$L_{ij} = -A_{ij} + d_i \delta_{ij}$$

Graph Laplacian

- Laplacian is a widely used representation for graphs.
- Laplacian of an undirected graph is a $|V| \times |V|$ symmetric matrix defined as

$$L_{ij} = \begin{cases} d_j & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (v_i, v_j) \in \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

- We can represent this using index notation as

$$L_{ij} = -A_{ij} + d_i \delta_{ij}$$

- In matrix notation

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

Degree matrix (D) **Adjacency matrix** (A)

- It is only suitable for undirected graphs.

Graph Laplacian

- Rows of Laplacian matrix sum to zero

$$
\begin{aligned}
\sum_j L_{ij} &= \sum_j -A_{ij} + d_i \delta_{ij} \\
&= -\sum_j A_{ij} + \sum_j d_i \delta_{ij} \\
&= -d_i + d_i = 0
\end{aligned}
$$

Graph Laplacian

- Rows of Laplacian matrix sum to zero

$$\begin{aligned}\sum_j L_{ij} &= \sum_j -A_{ij} + d_i\delta_{ij} \\ &= -\sum_j A_{ij} + \sum_j d_i\delta_{ij} \\ &= -d_i + d_i = 0\end{aligned}$$

- All eigenvalues of L are non-negative

$$\begin{aligned}\lambda = \mathbf{v}^T\mathbf{L}\mathbf{v} &= \sum_i\sum_j v_i L_{ij} v_j \\ &= \sum_i\sum_j \left(-A_{ij} + d_i\delta_{ij}\right) v_i v_j \\ &= -\sum_i\sum_j A_{ij} v_i v_j + \sum_i\sum_j d_i\delta_{ij} v_i v_j\end{aligned}$$

Graph Laplacian

- Then,

$$\begin{aligned}
\lambda =& -\sum_i \sum_j A_{ij} v_i v_j + \frac{1}{2} \sum_i \sum_j d_i \delta_{ij} v_i v_j + \frac{1}{2} \sum_j \sum_i d_j \delta_{ji} v_j v_i \\
=& -\sum_i \sum_j A_{ij} v_i v_j + \frac{1}{2} \sum_i d_i v_i^2 + \frac{1}{2} \sum_j d_j v_j^2 \\
=& -\sum_i \sum_j A_{ij} v_i v_j + \frac{1}{2} \sum_i \sum_j A_{ij} v_i^2 + \frac{1}{2} \sum_i \sum_j A_{ij} v_j^2 \\
=& \frac{1}{2} \sum_i \sum_j A_{ij} \left(-2 v_i v_j + v_i^2 + v_j^2\right) \\
=& \frac{1}{2} \sum_i \sum_j A_{ij} \left(v_i - v_j\right)^2 \geqslant 0
\end{aligned}$$

Graph Laplacian

- **L** has at least one 0 eigenvalue.

$$\lambda = 0 = \mathbf{v}^T \mathbf{L} \mathbf{v} = \frac{1}{2} \sum_i \sum_j A_{ij} \left(v_i - v_j \right)^2$$

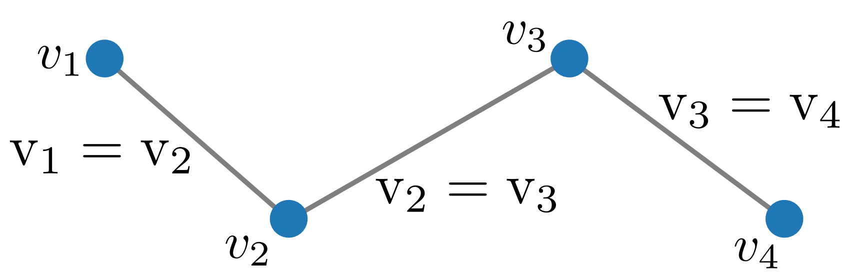
- This indicates that either $A_{ij} = 0$ or $A_{ij} = 1$ and $v_i = v_j$.

Graph Laplacian

- **L** has at least one 0 eigenvalue.

$$\lambda = 0 = \mathbf{v}^T \mathbf{L} \mathbf{v} = \frac{1}{2} \sum_i \sum_j A_{ij} (v_i - v_j)^2$$

- This indicates that either $A_{ij} = 0$ or $A_{ij} = 1$ and $v_i = v_j$.



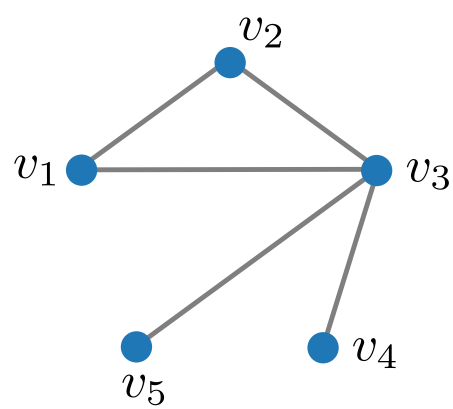
- Therefore, eigenvalue $\lambda = 0$ has a corresponding $\mathbf{v}$ with equal elements.
- Alternatively, since all elements of L sum to 1, we have

$$\mathbf{L}\,\vec{\mathbf{1}} = 0$$

Incidence Matrix

- For a graph $G = (V, E)$, an incidence matrix is a $|E|\times|V|$ matrix describing the membership of a node in edges of the graph

$$B_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$$

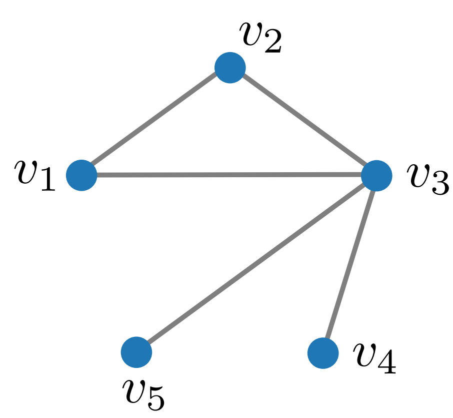


Incidence Matrix

- For a graph $G = (V, E)$, an incidence matrix is a $|E| \times |V|$ matrix describing the membership of a node in edges of the graph

$$B_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$$

	v_1	v_2	v_3	v_4	v_5
(v_1, v_2)					
(v_2, v_3)					
(v_1, v_3)					
(v_3, v_4)					
(v_3, v_5)					

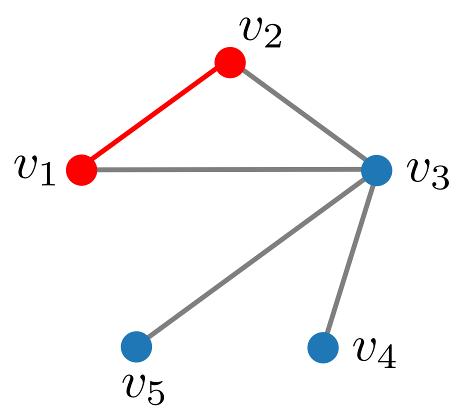


Incidence Matrix

- For a graph $G = (V, E)$, an incidence matrix is a $|E| \times |V|$ matrix describing the membership of a node in edges of the graph

$$B_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$$

	v_1	v_2	v_3	v_4	v_5
(v_1, v_2)	1	1			
(v_2, v_3)					
(v_1, v_3)					
(v_3, v_4)					
(v_3, v_5)					

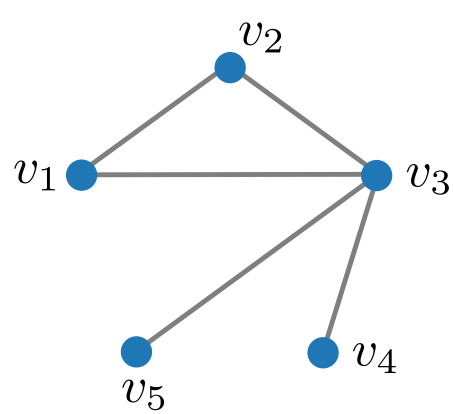


Incidence Matrix

- For a graph $G = (V, E)$, an incidence matrix is a $|E| \times |V|$ matrix describing the membership of a node in edges of the graph

$$B_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$$

	v_1	v_2	v_3	v_4	v_5
(v_1, v_2)	1	1			
(v_2, v_3)		1	1		
(v_1, v_3)	1		1		
(v_3, v_4)			1	1	
(v_3, v_5)			1		1



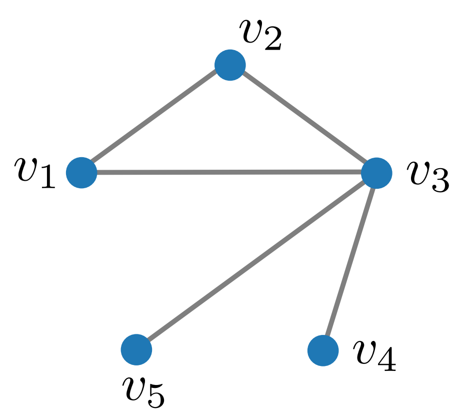
- Perform projection in bipartite graphs.

Incidence Matrix

- An oriented incidence matrix for undirected graph is defined by any orientation of the graph.
- For $\varepsilon_{ij} \in E$, we use the convention

$$B_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \text{ and } i < j, \\ -1 & \text{if } (v_i, v_j) \in E \text{ and } i > j, \\ 0 & \text{otherwise.} \end{cases}$$

	v_1	v_2	v_3	v_4	v_5
(v_1, v_2)					
(v_2, v_3)					
(v_1, v_3)					
(v_3, v_4)					
(v_3, v_5)					

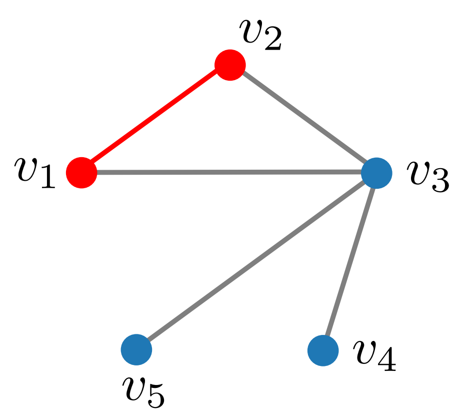


Incidence Matrix

- An oriented incidence matrix for undirected graph is defined by any orientation of the graph.
- For $\varepsilon_{ij} \in E$, we use the convention

$$B_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \text{ and } i < j, \\ -1 & \text{if } (v_i, v_j) \in E \text{ and } i > j, \\ 0 & \text{otherwise.} \end{cases}$$

	v_1	v_2	v_3	v_4	v_5
(v_1, v_2)	1	-1			
(v_2, v_3)					
(v_1, v_3)					
(v_3, v_4)					
(v_3, v_5)					

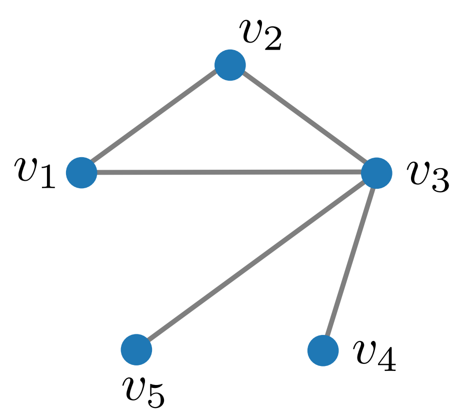


Incidence Matrix

- An oriented incidence matrix for undirected graph is defined by any orientation of the graph.
- For $\varepsilon_{ij} \in E$, we use the convention

$$B_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \text{ and } i < j, \\ -1 & \text{if } (v_i, v_j) \in E \text{ and } i > j, \\ 0 & \text{otherwise.} \end{cases}$$

	v_1	v_2	v_3	v_4	v_5
(v_1, v_2)	1	-1			
(v_2, v_3)		1	-1		
(v_1, v_3)	1		-1		
(v_3, v_4)			1	-1	
(v_3, v_5)			1		-1



Laplacian

- For a multivariate function $f\colon \mathbb{R}^n \rightarrow \mathbb{R}$, the Laplacian operator is defined as the divergence of the gradient of f

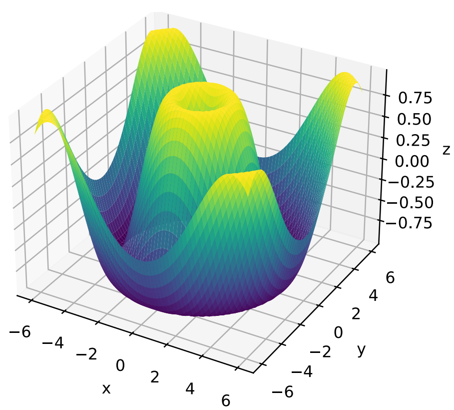
$$\Delta f(\mathbf{x}) = \nabla \cdot \nabla f(\mathbf{x})$$

- Gradient of f, $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined as the vector

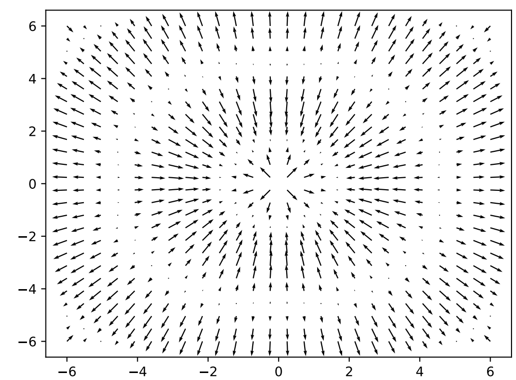
$$\nabla f(\mathbf{x}) = \left[\frac{\partial}{\partial x_1} f(\mathbf{x}), \ldots, \frac{\partial}{\partial x_n} f(\mathbf{x})\right]$$

- Gradient represents the rate and direction of the steepest ascent of function f at **x**.

$f(\mathbf{x})$



$\nabla f(\mathbf{x})$



Laplacian

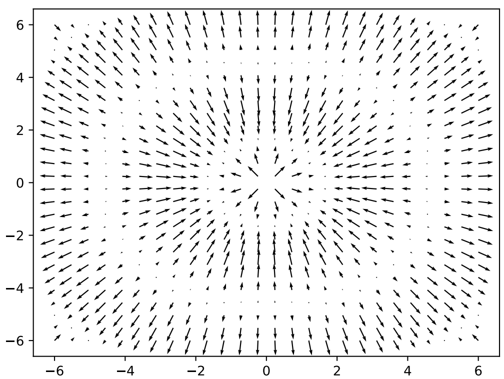
- Divergence of a vector field is the outward flux of the vector field at point $\mathbf{x}$.

$$\nabla \cdot \mathbf{F} = \frac{\partial}{\partial x_1}\mathbf{F} + \cdots + \frac{\partial}{\partial x_n}\mathbf{F}$$

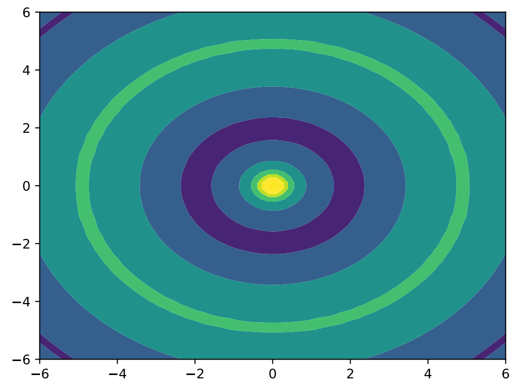
- Laplacian is then derived as

$$\nabla \cdot \nabla f(\mathbf{x}) = \frac{\partial^2}{\partial x_1^2} f(\mathbf{x}) + \cdots + \frac{\partial^2}{\partial x_n^2} f(\mathbf{x})$$

$\nabla f(\mathbf{x})$



$\nabla \cdot \nabla f(\mathbf{x})$

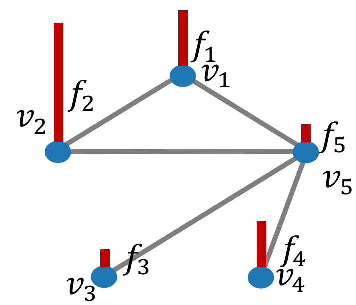


Graph Laplacian

- A real function f on graph is defined as a map from nodes to real number. This is discrete analogue of a scalar field.

$$f : V \to \mathbb{R}$$

$$\boldsymbol{f} := \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 \end{bmatrix}^T$$



- Gradient operator on graph ∇: $V \to E$ is defined as

$$[\nabla \boldsymbol{f}]_{ij} = f_i - f_j$$

- Given incident matrix **B**, we can compute gradient as $\nabla f = \boldsymbol{B} f$

$$\begin{bmatrix} 1 & -1 & \cdot & \cdot & \cdot \\ \cdot & 1 & -1 & \cdot & \cdot \\ 1 & \cdot & -1 & \cdot & \cdot \\ \cdot & \cdot & 1 & -1 & \cdot \\ \cdot & \cdot & 1 & \cdot & -1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix} = \begin{bmatrix} f_1 - f_2 \\ f_2 - f_3 \\ f_1 - f_3 \\ f_3 - f_4 \\ f_3 - f_5 \end{bmatrix}$$

Graph Laplacian

- Divergence operator on a graph $\text{div}: E \rightarrow V$ sums over values associated with the edges of the graph.

$$\text{div}(\mathbf{g})_i = \sum_{v_j \in N(v_i)} \mathrm{g}_{ij}$$

- Given incident matrix **B**, we can compute divergence as

$$\text{div } \mathbf{g} = \mathbf{B}^T \mathbf{g}$$

$$\begin{bmatrix} 1 & \cdot & 1 & \cdot & \cdot \\ -1 & 1 & \cdot & \cdot & \cdot \\ \cdot & -1 & -1 & 1 & 1 \\ \cdot & \cdot & \cdot & -1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & -1 \end{bmatrix} \begin{bmatrix} g_{(1,2)} \\ g_{(2,3)} \\ g_{(1,3)} \\ g_{(3,4)} \\ g_{(3,5)} \end{bmatrix} = \begin{bmatrix} g_{(1,2)} + g_{(1,3)} \\ -g_{(1,2)} + g_{(2,3)} \\ -g_{(2,3)} - g_{(1,3)} + g_{(3,4)} + g_{(3,5)} \\ -g_{(3,4)} \\ -g_{(3,5)} \end{bmatrix}$$

Graph Laplacian

- Divergence operator on a graph $\text{div}: E \to V$ sums over values associated with the edges of the graph.

$$\text{div}(\mathbf{g})_i = \sum_{v_j \in N(v_i)} g_{ij}$$

- Given incident matrix **B**, we can compute divergence as

$$\text{div}\ \mathbf{g} = \mathbf{B}^T \mathbf{g}$$

- Laplacian operator on graph $\Delta: V \to V$ is defined as

$$\mathbf{B}^T \mathbf{B} \boldsymbol{f} = \mathbf{L}\mathbf{f}$$

Graph Laplacian

- To prevent too much effect of nodes with large degree, we can use normalized Laplacian matrices.
- There are two variants of normalized graph Laplacian:

➤ Symmetric normalized Laplacian

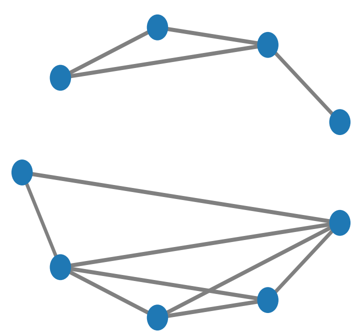
$$\hat{\mathbf{L}}_{Sym} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$$

➤ Random-Walk normalized Laplacian

$$\hat{\mathbf{L}}_{RW} = \mathbf{D}^{-1}\mathbf{L}$$

Spectral Partitioning

- Disconnected graphs have an adjacency matrix with nonzero square blocks on the diagonal and zeros outside of these blocks.



Block diagonal matrix

$\boldsymbol{A}$:

```
[[0 1 0 0 1 0 0 0 0]
 [1 0 1 1 1 0 0 0 0]
 [0 1 0 1 1 0 0 0 0]
 [0 1 1 0 1 0 0 0 0]
 [1 1 1 1 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 1 0 1 1]
 [0 0 0 0 0 0 1 0 1]
 [0 0 0 0 0 0 1 1 0]]
```

- Since $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, $\boldsymbol{L}$ also has a block diagonal form in disconnected graphs.
- Number of the zero eigenvalues of Laplacian represents the number of components in the graph.

Spectral Partitioning

- Consider the cut size

$$R = \frac{1}{4} \sum_{i \in V} \sum_{j \in V} A_{ij} \left(1 - s_i s_j\right)$$

$$= \frac{1}{4} \sum_{i \in V} \sum_{j \in V} \left[A_{ij} - A_{ij} s_i s_j\right]$$

- Since

$$s_i = \begin{cases} 1 & v_i \in \mathcal{A}_1 \\ -1 & v_i \in \mathcal{A}_2 \end{cases}$$

therefore, $s_i^2 = 1$

$$\sum_{i \in V} \sum_{j \in V} A_{ij} = \sum_{i \in V} \sum_{j \in V} A_{ij} s_i^2$$

$$= \sum_{i \in V} d_i s_i^2 = \sum_{i \in V} \sum_{j \in V} d_i s_i s_j \delta_{ij}$$

Spectral Partitioning

❖ Plugging in

$$
\begin{aligned}
R &= \frac{1}{4} \sum_{i \in V} \sum_{j \in V} d_i s_i s_j \delta_{ij} - A_{ij} s_i s_j \\
&= \frac{1}{4} \sum_{i \in V} \sum_{j \in V} \underbrace{\left(d_i s_{ij} - A_{ij}\right)} s_i s_j
\end{aligned}
$$

Spectral Partitioning

- Plugging in

$$R = \frac{1}{4} \sum_{i \in V} \sum_{j \in V} d_i s_i s_j \delta_{ij} - A_{ij} s_i s_j$$

$$= \frac{1}{4} \sum_{i \in V} \sum_{j \in V} \underbrace{(d_i s_{ij} - A_{ij})}_{\text{Laplacian}} s_i s_j = \frac{1}{4} \sum_{i \in V} \sum_{j \in V} L_{ij} s_i s_j$$

- In matrix notation, the cut size is

$$R = \frac{1}{4} \mathbf{s}^T \mathbf{L} \mathbf{s}$$

Graph structure ($\mathbf{L}$) — Division in graph ($\mathbf{s}$)

- We can express the graph partitioning problem as finding partitioning s that minimizes R.

Spectral Partitioning

- Approximation to ratio cut using spectral methods

$$s_i = \begin{cases} +\sqrt{\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|}} & \text{if} \quad v_i \in \mathcal{A} \\ -\sqrt{\frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|}} & \text{if} \quad v_i \in \bar{\mathcal{A}} \end{cases}$$

$$\begin{aligned} \mathbf{s}^T \mathbf{L} \mathbf{s} &= \frac{1}{2} \sum_i \sum_j A_{ij} \left(s_i - s_j\right)^2 \\ &= \sum_{v_i \in \mathcal{A}} \sum_{v_j \in \bar{\mathcal{A}}} A_{ij} \left(s_i - s_j\right)^2 \\ &= \sum_{v_i \in \mathcal{A}} \sum_{v_j \in \bar{\mathcal{A}}} A_{ij} \left(\sqrt{\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|}} + \sqrt{\frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|}}\right)^2 \\ &= \left(\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|} + \frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|} + 2\right) \sum_{v_i \in \mathcal{A}} \sum_{v_j \in \bar{\mathcal{A}}} A_{ij} \end{aligned}$$

Spectral Partitioning

$$\begin{aligned}
\mathbf{s}^T\mathbf{L}\mathbf{s} &= \left(\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|} + \frac{|\mathcal{A}|}{|\mathcal{A}|} + \frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|} + \frac{|\bar{\mathcal{A}}|}{|\bar{\mathcal{A}}|}\right) \sum_{v_i \in \mathcal{A}} \sum_{v_j \in \bar{\mathcal{A}}} A_{ij} \\
&= \left(\frac{|\bar{\mathcal{A}}| + |\mathcal{A}|}{|\mathcal{A}|} + \frac{|\mathcal{A}| + |\bar{\mathcal{A}}|}{|\bar{\mathcal{A}}|}\right) \sum_{v_i \in \mathcal{A}} \sum_{v_j \in \bar{\mathcal{A}}} A_{ij} \\
&= |V| \left(\frac{1}{|\mathcal{A}|} + \frac{1}{|\bar{\mathcal{A}}|}\right) \sum_{v_i \in \mathcal{A}} \sum_{v_j \in \bar{\mathcal{A}}} A_{ij} \\
&= |V| \frac{1}{2} \sum_{k=1}^{2} \sum_{v_i \in \mathcal{A}_k} \sum_{v_j \in \bar{\mathcal{A}}_k} \frac{A_{ij}}{|\mathcal{A}_k|} \\
&= |V| R^{\mathrm{RatioCut}}(\mathcal{A}, \bar{\mathcal{A}})
\end{aligned}$$

$$R^{\mathrm{RatioCut}}(\mathcal{A}, \bar{\mathcal{A}}) = \frac{1}{|V|}\mathbf{s}^T\mathbf{L}\mathbf{s}$$

Spectral Partitioning

- Note that

$$\sum_{v_i \in V} s_i = \sum_{v_i \in \mathcal{A}} \sqrt{\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|}} - \sum_{v_i \in \bar{\mathcal{A}}} \sqrt{\frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|}}$$

$$= |\mathcal{A}| \sqrt{\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|}} - |\bar{\mathcal{A}}| \sqrt{\frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|}} = \sqrt{|\mathcal{A}||\bar{\mathcal{A}}|} - \sqrt{|\bar{\mathcal{A}}||\mathcal{A}|} = 0$$

- Therefore $\boldsymbol{s} \perp \vec{\mathbf{1}}$.
- Also,

$$\|\mathbf{s}\|^2 = \sum_{v_i \in \mathcal{A}} \left(\sqrt{\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|}} \right)^2 + \sum_{v_i \in \bar{\mathcal{A}}} \left(\sqrt{\frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|}} \right)^2$$

$$= |\mathcal{A}| \frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|} + |\bar{\mathcal{A}}| \frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|} = |V|$$

- $$\min_{\mathcal{A} \subset V} \mathbf{s}^T \mathbf{L} \mathbf{s} \quad \text{s.t.} \quad \mathbf{s} \perp \vec{\mathbf{1}} \text{ and } \|\mathbf{s}\|^2 = |V|$$

K-means Clustering

- Feature-based clustering approach.
- It clusters data points into K clusters by minimizing their feature vector's distance from cluster mean.

$$\arg\min_{k} \|\boldsymbol{z}_n - \boldsymbol{\mu}_k\|_2^2$$

- Mean $\boldsymbol{\mu}_k$ is computed at each iteration based on the points assigned to the cluster k at that iteration

$$\boldsymbol{\mu}_k = \frac{1}{|\mathcal{A}_k|} \sum_{\boldsymbol{z}_n \in \mathcal{A}_k} \boldsymbol{z}_n$$

- In matrix notation,

$$\min_{\mathbf{S}} \left\| \mathbf{Z} - \mathbf{S}\mathbf{M}^T \right\|_F^2$$

Cluster Assignment Matrix

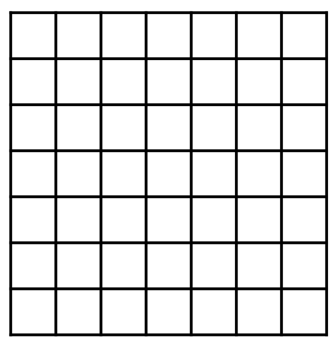
Feature Matrix

Cluster Means

Generalized Spectral Clustering

- K-means clustering can cluster nodes of a graph based on feature vectors built upon eigen-decomposition of $\boldsymbol{L}$.
  - Find eigen-decomposition of Laplacian $\boldsymbol{LV} = \boldsymbol{\Lambda V}$.

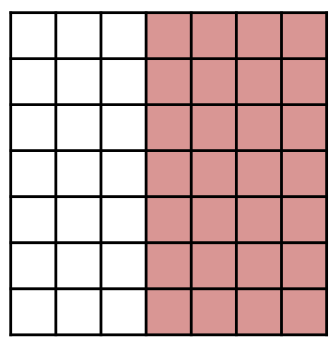
$\boldsymbol{V}$:



Generalized Spectral Clustering

- K-means clustering can cluster nodes of a graph based on feature vectors built upon eigen-decomposition of $\boldsymbol{L}$.
  - Find eigen-decomposition of Laplacian $\boldsymbol{LV} = \boldsymbol{\Lambda V}$.
  - Construct feature matrix $\boldsymbol{Z}$ by separating the eigenvectors corresponding to the K smallest eigenvalues $\boldsymbol{V}_{|V|-K:|V|}$.

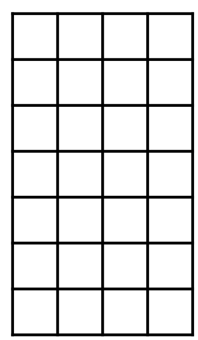
$\boldsymbol{V}$:



Generalized Spectral Clustering

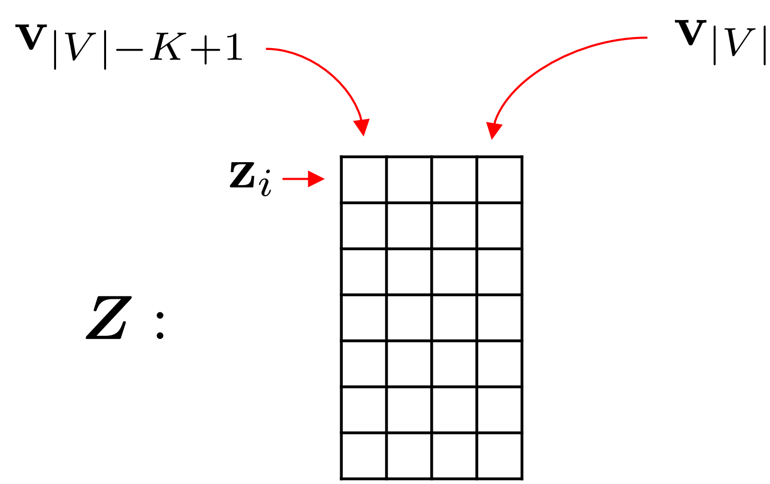
- K-means clustering can cluster nodes of a graph based on feature vectors built upon eigen-decomposition of $\boldsymbol{L}$.
  - Find eigen-decomposition of Laplacian $\boldsymbol{LV} = \boldsymbol{\Lambda V}$.
  - Construct feature matrix $\boldsymbol{Z}$ by separating the eigenvectors corresponding to the K smallest eigenvalues $\boldsymbol{V}_{|V|-K:|V|}$.

$\boldsymbol{Z}$:



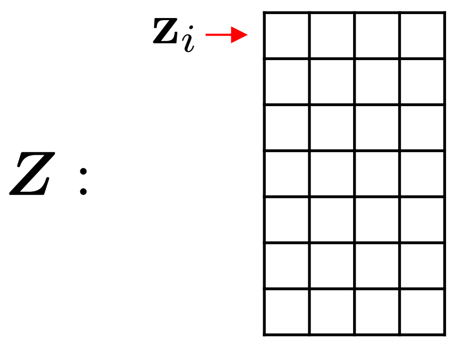
Generalized Spectral Clustering

- K-means clustering can cluster nodes of a graph based on feature vectors built upon eigen-decomposition of $\boldsymbol{L}$.
  - Find eigen-decomposition of Laplacian $\boldsymbol{LV} = \boldsymbol{\Lambda V}$.
  - Construct feature matrix $\boldsymbol{Z}$ by separating the eigenvectors corresponding to the K smallest eigenvalues $\boldsymbol{V}_{|V|-K:|V|}$.
  - i-th row of $\boldsymbol{Z}$ represents the embedding z_i for node v_i.



Generalized Spectral Clustering

- K-means clustering can cluster nodes of a graph based on feature vectors built upon eigen-decomposition of $\boldsymbol{L}$.
  - Find eigen-decomposition of Laplacian $\boldsymbol{LV} = \boldsymbol{\Lambda V}$.
  - Construct feature matrix $\boldsymbol{Z}$ by separating the eigenvectors corresponding to the K smallest eigenvalues $\boldsymbol{V}_{|V|-K:|V|}$.
  - i-th row of $\boldsymbol{Z}$ represents the embedding z_i for node v_i.
  - Run K-means clustering on features z_i, where $v_i \in V$.



Summary

- Graph Partitioning
- Cut Set
- Ratio Cut
- Normalized Cut
- Graph Laplacian
  - Properties
  - Intuition
- Incidence matrix
- Spectral Partitioning
- K-means clustering
- Generalized Spectral Clustering