This Class: Spectral Clustering

- Finding good cuts via Laplacian eigenvectors.
- Start analysis via the stochastic block model.
GRAPH CLUSTERING
A very common task is to **partition or cluster** vertices in a graph based on similarity/connectivity.
A very common task is to **partition or cluster** vertices in a graph based on similarity/connectivity.

**Community detection in naturally occurring networks.**

(a) Zachary Karate Club Graph
A very common task is to **partition or cluster** vertices in a graph based on similarity/connectivity.

**Community detection in naturally occurring networks.**
A very common task is to **partition or cluster** vertices in a graph based on similarity/connectivity.

**Linearly separable data.**
A very common task is to **partition or cluster** vertices in a graph based on similarity/connectivity.

**Non-linearly separable data** $k$-nearest neighbor graph.
A very common task is to **partition or cluster** vertices in a graph based on similarity/connectivity.

**Non-linearly separable data** *k*-nearest neighbor graph.
A very common task is to partition or cluster vertices in a graph based on similarity/connectivity.

Non-linearly separable data $k$-nearest neighbor graph.
A very common task is to **partition or cluster** vertices in a graph based on similarity/connectivity.

**Non-linearly separable data** $k$-nearest neighbor graph.

Can find this cut using eigendecomposition!
**Simple Idea:** Partition clusters along minimum cut in graph.

![Zachary Karate Club Graph](image)

(a) Zachary Karate Club Graph
**Simple Idea:** Partition clusters along minimum cut in graph.

Small cuts are often not informative.
Simple Idea: Partition clusters along minimum cut in graph.

Small cuts are often not informative.

Solution: Encourage cuts that separate large sections of the graph.
**Simple Idea:** Partition clusters along minimum cut in graph.

Small cuts are often not informative.

**Solution:** Encourage cuts that separate large sections of the graph.

- Let $\vec{v} \in \mathbb{R}^n$ be a **cut indicator**: $\vec{v}(i) = 1$ if $i \in S$. $\vec{v}(i) = -1$ if $i \in T$. Want $\vec{v}$ to have roughly equal numbers of 1s and −1s. I.e., $\vec{v}^T \vec{1} \approx 0$. 

(a) Zachary Karate Club Graph
For a graph with adjacency matrix $A$ and degree matrix $D$, $L = D - A$ is the graph Laplacian.
For a graph with adjacency matrix \( A \) and degree matrix \( D \), \( L = D - A \) is the graph Laplacian.

For any vector \( \vec{v} \), its ‘smoothness’ over the graph is given by:

\[
\sum_{(i,j) \in E} (\vec{v}(i) - \vec{v}(j))^2 = \vec{v}^T L \vec{v}.
\]
Lemma:

\[
\sum_{(i,j) \in E} (\vec{v}(i) - \vec{v}(j))^2 = \vec{v}^T L \vec{v}
\]
Lemma:

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

Proof:
Lemma:

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

Proof:

• Let \( L_e \) be the Laplacian for graph just containing edge \( e \).
Rewriting Laplacian

Lemma:
\[ \sum_{(i,j) \in E} (\vec{v}(i) - \vec{v}(j))^2 = \vec{v}^T L \vec{v} \]

Proof:
- Let $L_e$ be the Laplacian for graph just containing edge $e$.
- By linearity,
  \[ \vec{v}^T L \vec{v} = \sum_{e \in E} \vec{v}^T L_e \vec{v} \]
Lemma:

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

Proof:

• Let \( L_e \) be the Laplacian for graph just containing edge \( e \).
• By linearity,
  \[
  \vec{v}^T L \vec{v} = \sum_{e \in E} \vec{v}^T L_e \vec{v}
  \]
• If \( e = (i, j) \), then \( \vec{v}^T L_e \vec{v} = (v(i) - v(j))^2 \)
For a cut indicator vector $\vec{v} \in \{-1, 1\}^n$ with $\vec{v}(i) = -1$ for $i \in S$ and $\vec{v}(i) = 1$ for $i \in T$:

1. $\vec{v}^T L \vec{v} = \sum_{(i,j) \in E} (\vec{v}(i) - \vec{v}(j))^2 = 4 \cdot \text{cut}(S, T)$. 
For a cut indicator vector $\vec{v} \in \{-1, 1\}^n$ with $\vec{v}(i) = -1$ for $i \in S$ and $\vec{v}(i) = 1$ for $i \in T$:

1. $\vec{v}^T \mathbf{L} \vec{v} = \sum_{(i,j) \in E} (\vec{v}(i) - \vec{v}(j))^2 = 4 \cdot \text{cut}(S, T)$.
2. $\vec{v}^T \mathbf{1} = |T| - |S|$. 


For a cut indicator vector $\vec{v} \in \{-1, 1\}^n$ with $\vec{v}(i) = -1$ for $i \in S$ and $\vec{v}(i) = 1$ for $i \in T$:

1. $\vec{v}^T L \vec{v} = \sum_{(i,j) \in E} (\vec{v}(i) - \vec{v}(j))^2 = 4 \cdot \text{cut}(S, T)$. 
2. $\vec{v}^T \vec{1} = |T| - |S|$. 

Want to minimize both $\vec{v}^T L \vec{v}$ (cut size) and $\vec{v}^T \vec{1}$ (imbalance).
For a cut indicator vector $\vec{v} \in \{-1, 1\}^n$ with $\vec{v}(i) = -1$ for $i \in S$ and $\vec{v}(i) = 1$ for $i \in T$:

1. $\vec{v}^T L \vec{v} = \sum_{(i,j) \in E} (\vec{v}(i) - \vec{v}(j))^2 = 4 \cdot \text{cut}(S, T)$.

2. $\vec{v}^T \vec{1} = |T| - |S|$.

Want to minimize both $\vec{v}^T L \vec{v}$ (cut size) and $\vec{v}^T \vec{1}$ (imbalance).

**Next Step:** See how this dual minimization problem is naturally solved by eigendecomposition.
Assuming the graph is connected, the smallest eigenvector of the Laplacian is:

\[ \vec{v}_n = \frac{1}{\sqrt{n}} \cdot \vec{1} = \arg \min_{\vec{v} \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1} \vec{v}^T L \vec{v} \]

with eigenvalue \( \vec{v}_n^T L \vec{v}_n = 0 \).

\( n \): number of nodes in graph, \( A \in \mathbb{R}^{n \times n} \): adjacency matrix, \( D \in \mathbb{R}^{n \times n} \): diagonal degree matrix, \( L \in \mathbb{R}^{n \times n} \): Laplacian matrix \( L = D - A \).
Assuming the graph is connected, the smallest eigenvector of the Laplacian is:

\[ \vec{v}_n = \frac{1}{\sqrt{n}} \cdot \vec{1} = \arg \min_{\vec{v} \in \mathbb{R}^n \text{ with } \|\vec{v}\| = 1} \vec{v}^T L \vec{v} \]

with eigenvalue \( \vec{v}_n^T L \vec{v}_n = 0 \). Why?

---

\( n \): number of nodes in graph, \( A \in \mathbb{R}^{n \times n} \): adjacency matrix, \( D \in \mathbb{R}^{n \times n} \): diagonal degree matrix, \( L \in \mathbb{R}^{n \times n} \): Laplacian matrix \( L = D - A \).
By Courant-Fischer, the second smallest eigenvector is given by:

\[ \vec{v}_{n-1} = \arg \min_{\vec{v} \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1, \ \vec{v}^T \vec{v}=0} \vec{v}^T L \vec{v} \]
By Courant-Fischer, the second smallest eigenvector is given by:

\[ \vec{v}_{n-1} = \arg \min_{\vec{v} \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1, \; \vec{v}_{n-1}^T \vec{v}=0} \vec{v}^T L \vec{v} \]

If \( \vec{v}_{n-1} \) were in \( \left\{ -\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}} \right\}^n \) it would have:

- \( \vec{v}_{n-1}^T L \vec{v}_{n-1} = \frac{4}{n} \cdot \text{cut}(S, T) \) as small as possible given that

\[
\vec{v}_{n-1}^T \vec{v}_n = \frac{1}{\sqrt{n}} \vec{v}_{n-1}^T \vec{1} = \frac{|T| - |S|}{\sqrt{n}} = 0.
\]
By Courant-Fischer, the second smallest eigenvector is given by:

\[\vec{v}_{n-1} = \arg\min_{\vec{v} \in \mathbb{R}^n \text{ with } \|\vec{v}\| = 1} \vec{v}^T L \vec{v}\]

If \(\vec{v}_{n-1}\) were in \((-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}})^n\) it would have:

- \(\vec{v}_{n-1}^T L \vec{v}_{n-1} = \frac{4}{n} \cdot \text{cut}(S, T)\) as small as possible given that
  \[\vec{v}_{n-1}^T \vec{v}_{n-1} = \frac{1}{\sqrt{n}} \vec{v}_{n-1}^T \vec{1} = \frac{|T| - |S|}{\sqrt{n}} = 0\]

- I.e., \(\vec{v}_{n-1}\) would indicate the smallest perfectly balanced cut.
By Courant-Fischer, the second smallest eigenvector is given by:

\[ \vec{v}_{n-1} = \arg \min_{\vec{v} \in \mathbb{R}^n \text{ with } \|\vec{v}\| = 1, \vec{v}_n^T \vec{v} = 0} \vec{v}^T L \vec{v} \]

If \( \vec{v}_{n-1} \) were in \( \left\{ -\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}} \right\}^n \) it would have:

- \( \vec{v}_{n-1}^T L \vec{v}_{n-1} = \frac{4}{n} \cdot \text{cut}(S, T) \) as small as possible given that

\[ \vec{v}_{n-1}^T \vec{v}_{n-1} = \frac{1}{\sqrt{n}} \vec{v}_{n-1}^T \vec{1} = \frac{|T| - |S|}{\sqrt{n}} = 0. \]

- I.e., \( \vec{v}_{n-1} \) would indicate the smallest perfectly balanced cut.

- The eigenvector \( \vec{v}_{n-1} \in \mathbb{R}^n \) is not generally binary, but still satisfies a ‘relaxed’ version of this property.
Find a good partition of the graph by computing

\[ \vec{v}_{n-1} = \arg \min_{\vec{v} \in \mathbb{R}^d \text{ with } \|\vec{v}\|=1, \vec{v}^T \vec{1} = 0} \vec{v}^T L \vec{v} \]

Set \( S \) to be all nodes with \( \vec{v}_{n-1}(i) < 0 \), \( T \) to be all with \( \vec{v}_{n-1}(i) \geq 0 \).
Find a good partition of the graph by computing

\[
\vec{v}_{n-1} = \arg \min_{\vec{v} \in \mathbb{R}^d \text{ with } \|\vec{v}\|=1, \ \vec{v}^T \vec{1} = 0} \vec{v}^T L \vec{v}
\]

Set \( S \) to be all nodes with \( \vec{v}_{n-1}(i) < 0 \), \( T \) to be all with \( \vec{v}_{n-1}(i) \geq 0 \).
Find a good partition of the graph by computing

$$\vec{v}_{n-1} = \arg \min_{\vec{v} \in \mathbb{R}^d \text{ with } \|\vec{v}\| = 1, \vec{v}^T \vec{1} = 0} \vec{v}^T L \vec{v}$$

Set $S$ to be all nodes with $\vec{v}_{n-1}(i) < 0$, $T$ to be all with $\vec{v}_{n-1}(i) \geq 0$. 