COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Andrew McGregor

Lecture 18

SUMMARY

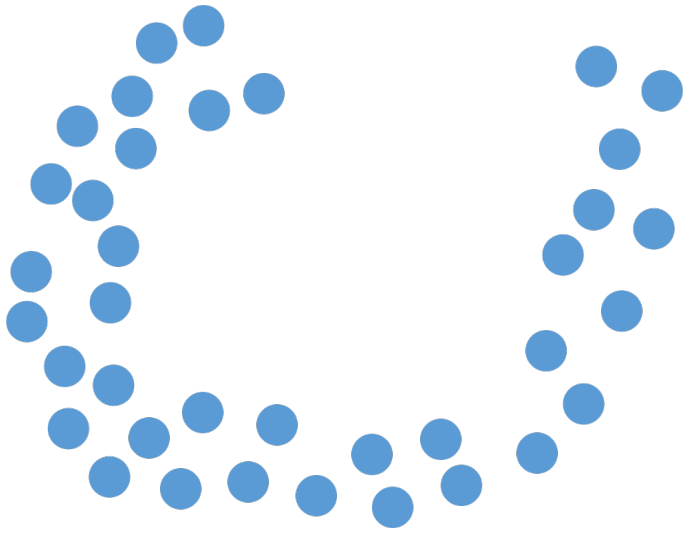
Last Class:

- SVD
- Low-rank matrix completion (predicting missing measurements using low-rank structure) and entity embeddings.

Spectral Graph Theory & Spectral Clustering.

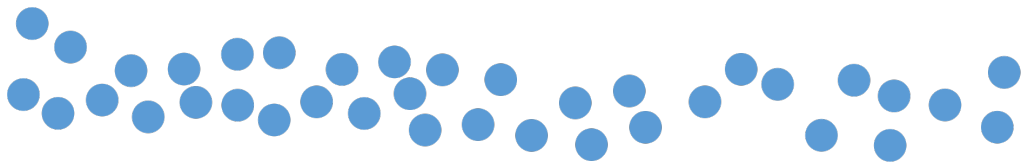
- Low-rank approximation on graph adjacency matrix for non-linear dimensionality reduction.
- Eigendecomposition to partition graphs into clusters.

NON-LINEAR DIMENSIONALITY REDUCTION



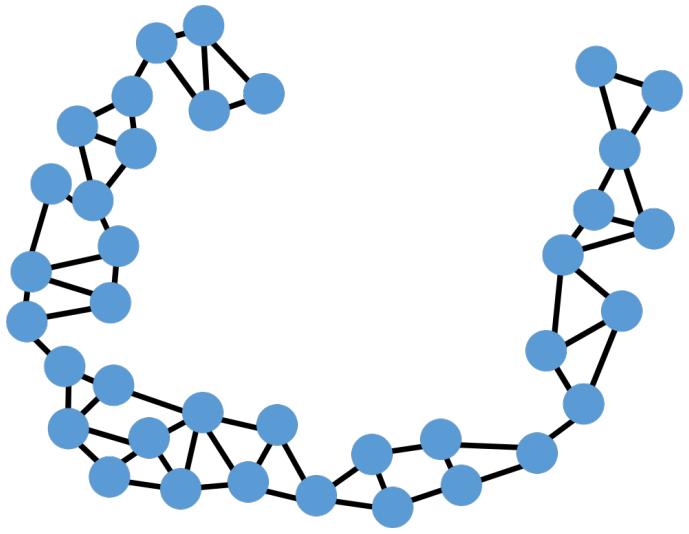
Is this set of points compressible? Does it lie close to a low-dimensional subspace? (A 1-dimensional subspace of $\mathbb{R}^d$.)

NON-LINEAR DIMENSIONALITY REDUCTION



Is this set of points compressible? Does it lie close to a low-dimensional subspace? (A 1-dimensional subspace of $\mathbb{R}^d$.)

NON-LINEAR DIMENSIONALITY REDUCTION



Is this set of points compressible? Does it lie close to a low-dimensional subspace? (A 1-dimensional subspace of $\mathbb{R}^d$.)

A common way of automatically identifying this non-linear structure is to connect data points in a graph. E.g., a k-nearest neighbor graph.

- Connect items to similar items, possibly with higher weight edges when they are more similar.

LINEAR ALGEBRAIC REPRESENTATION OF A GRAPH

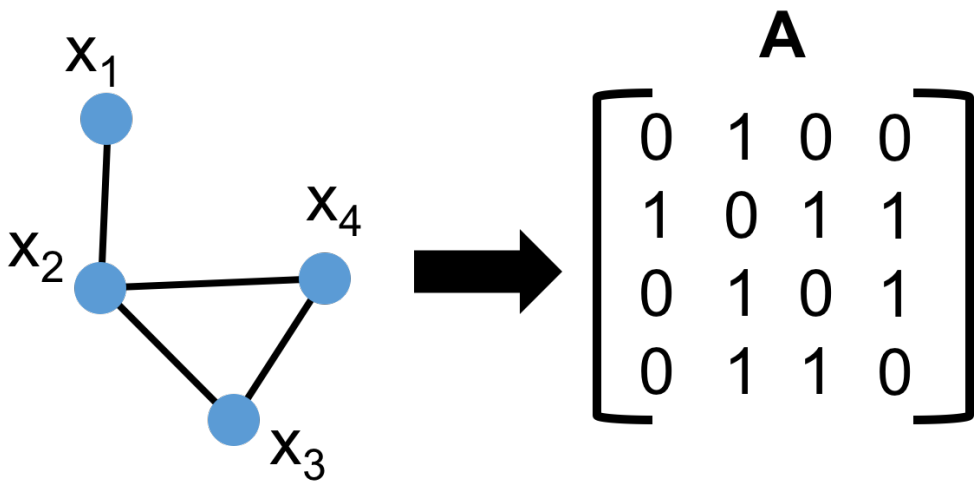
Once we have connected n data points $x_1, \ldots, x_n$ into a graph, we can represent that graph by its (weighted) adjacency matrix.

$$\mathbf{A} \in \mathbb{R}^{n \times n} \text{ with } \mathbf{A}_{i,j} = \text{ edge weight between nodes } i \text{ and } j$$

LINEAR ALGEBRAIC REPRESENTATION OF A GRAPH

Once we have connected n data points $x_1, \ldots, x_n$ into a graph, we can represent that graph by its (weighted) adjacency matrix.

$$\mathbf{A} \in \mathbb{R}^{n \times n} \text{ with } \mathbf{A}_{i,j} = \text{ edge weight between nodes } i \text{ and } j$$



ADJACENCY MATRIX EIGENVECTORS

How do we compute an optimal low-rank approximation of $\mathbf{A}$?

- Project onto the top k eigenvectors of $\mathbf{A}^T\mathbf{A} = \mathbf{A}^2$. These are just the eigenvectors of $\mathbf{A}$.

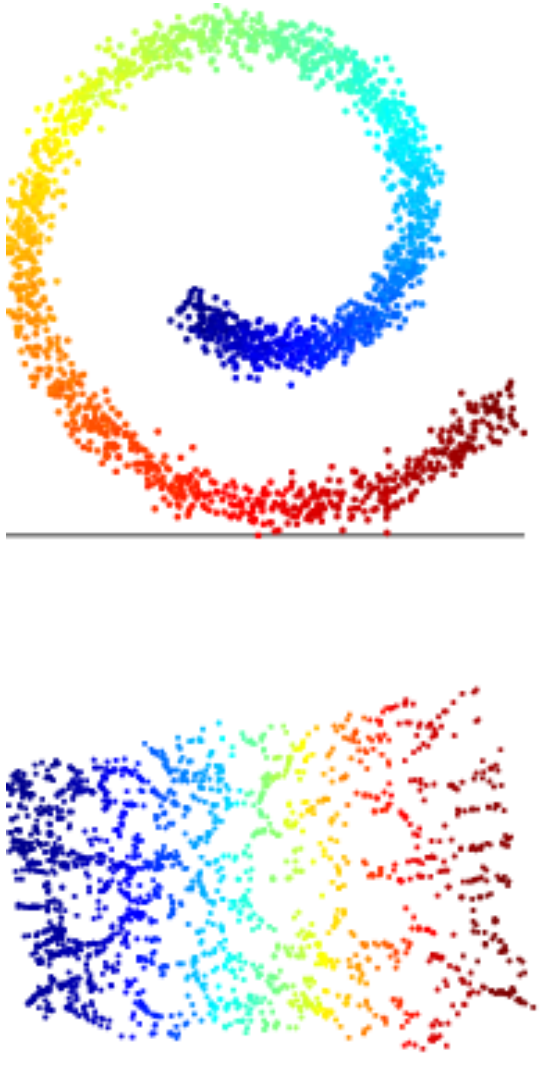
ADJACENCY MATRIX EIGENVECTORS

How do we compute an optimal low-rank approximation of $\mathbf{A}$?

- Project onto the top k eigenvectors of $\mathbf{A}^T\mathbf{A} = \mathbf{A}^2$. These are just the eigenvectors of $\mathbf{A}$.

- Similar vertices (close with regards to graph proximity) should have similar embeddings.

SPECTRAL EMBEDDING

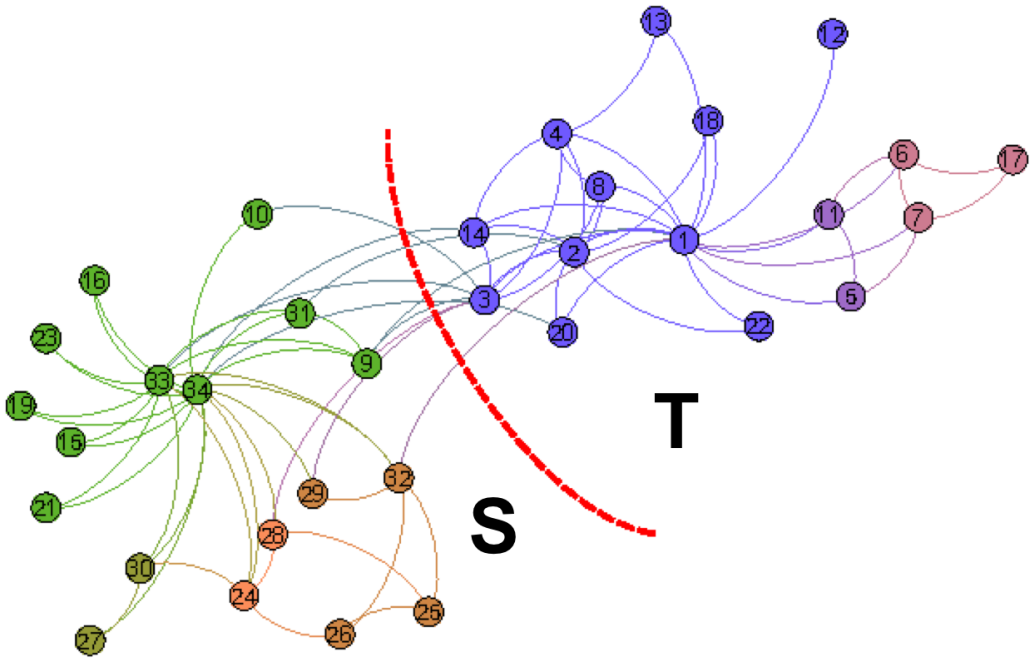


SPECTRAL CLUSTERING

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.

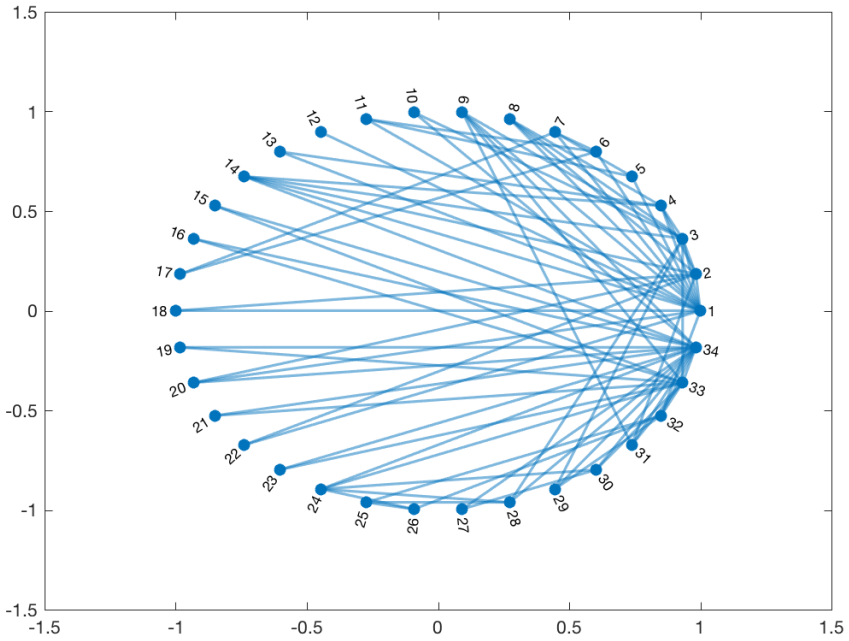
Community detection in naturally occurring networks.



(a) Zachary Karate Club Graph

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

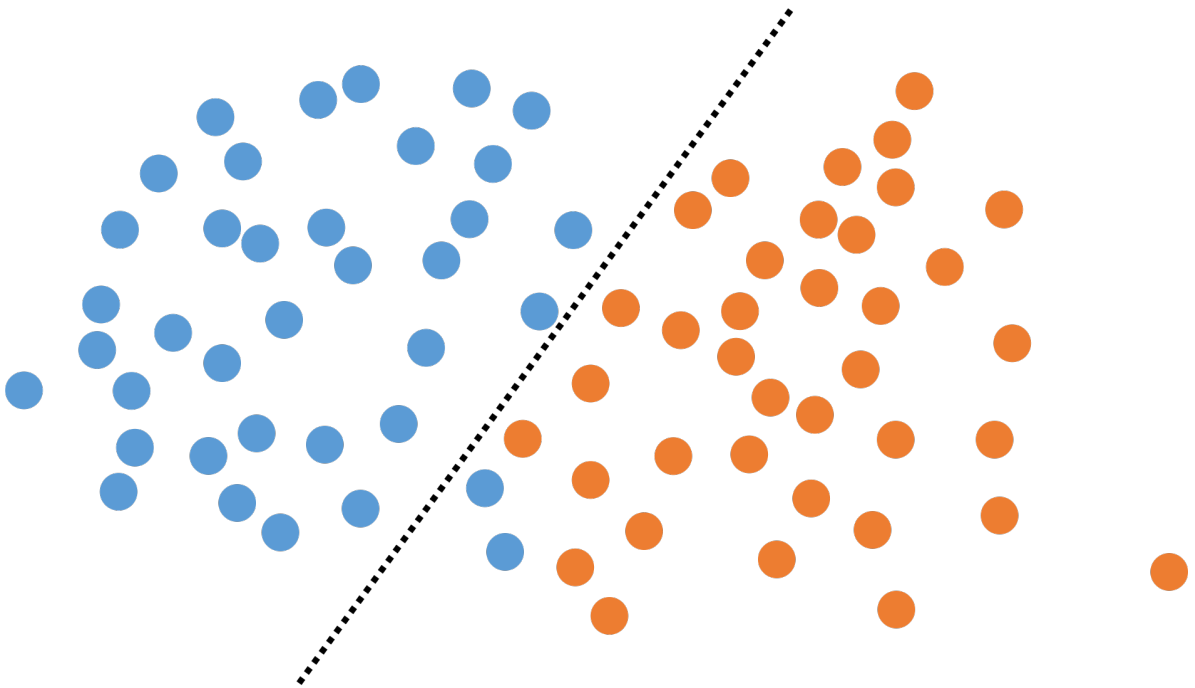
Community detection in naturally occurring networks.



SPECTRAL CLUSTERING

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

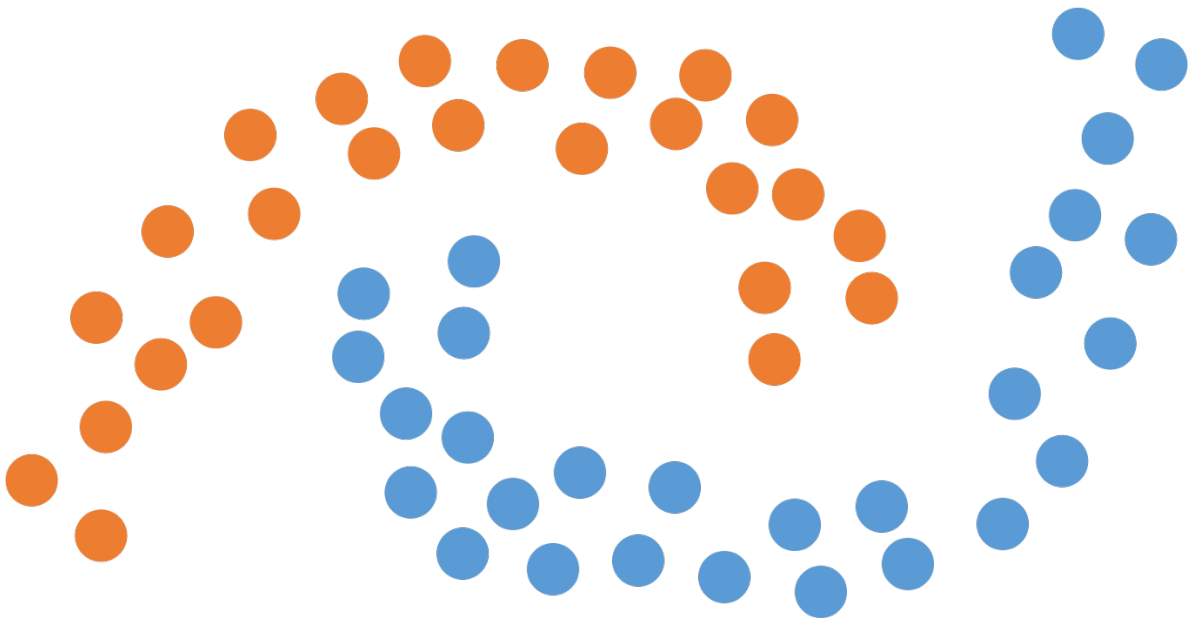
Linearly separable data.



SPECTRAL CLUSTERING

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

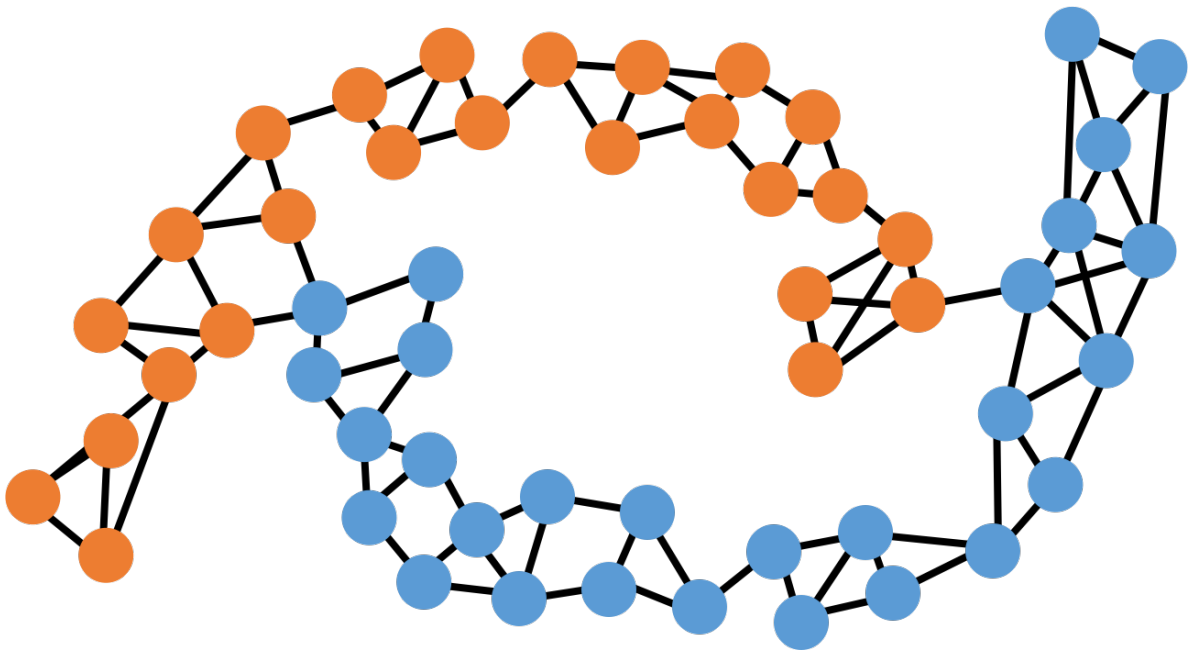
Non-linearly separable data k-nearest neighbor graph.



SPECTRAL CLUSTERING

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

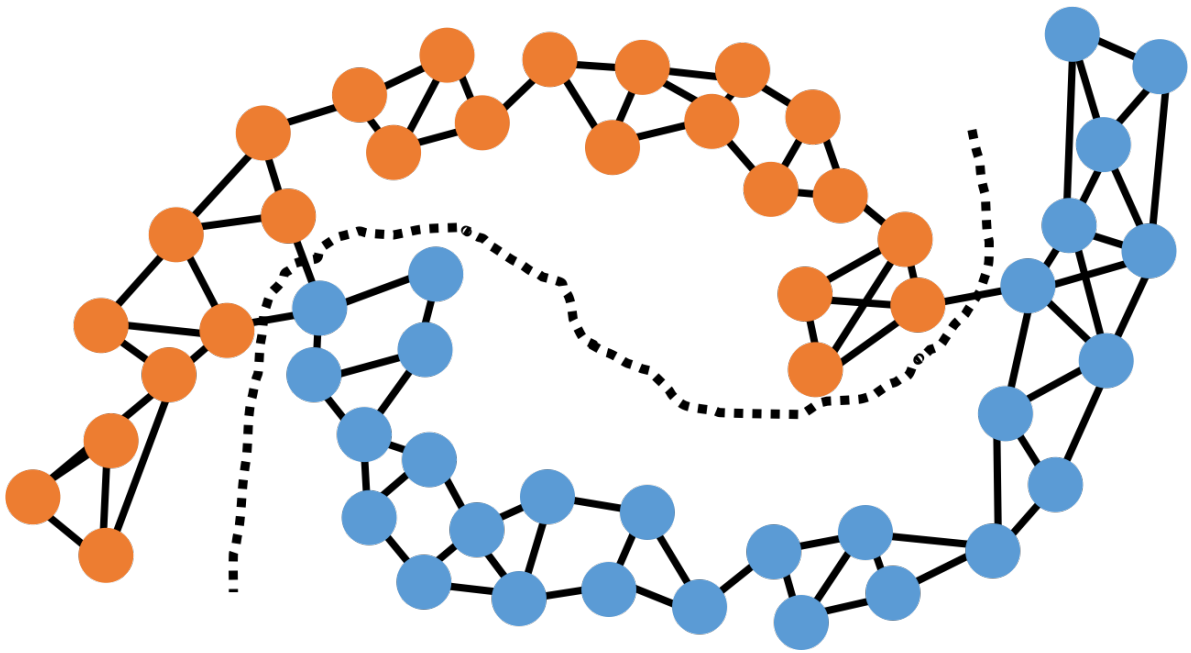
Non-linearly separable data k-nearest neighbor graph.



SPECTRAL CLUSTERING

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

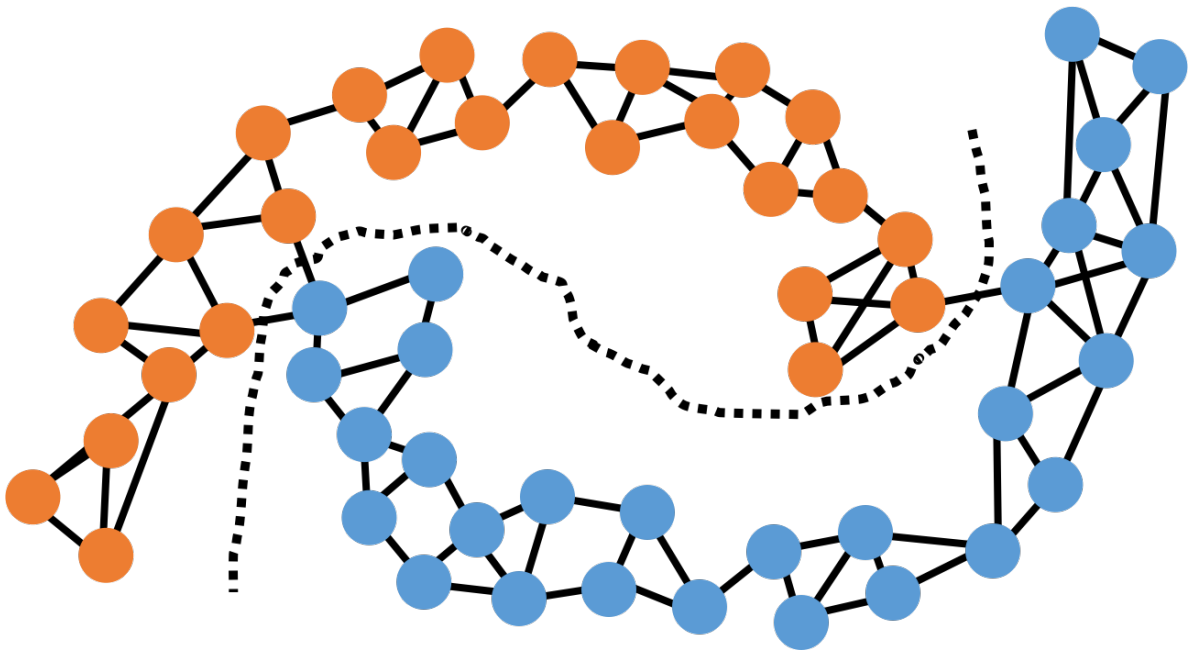
Non-linearly separable data k-nearest neighbor graph.



SPECTRAL CLUSTERING

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

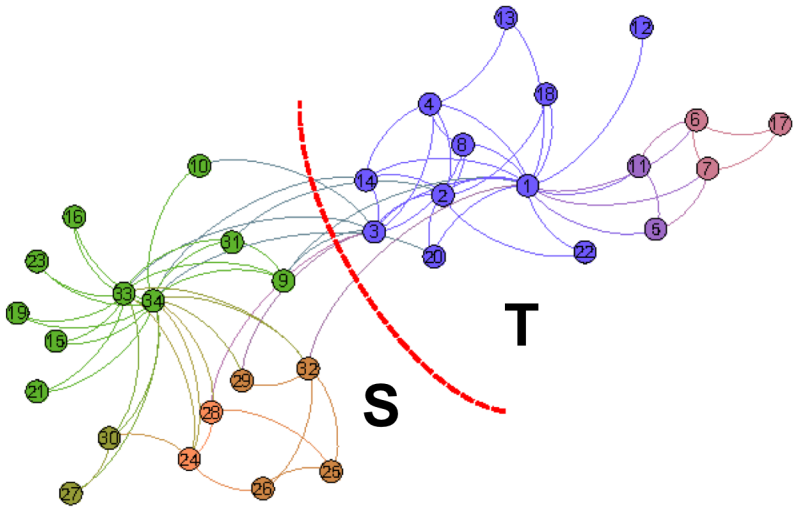
Non-linearly separable data k-nearest neighbor graph.



Can find this cut using eigendecomposition!

CUT MINIMIZATION

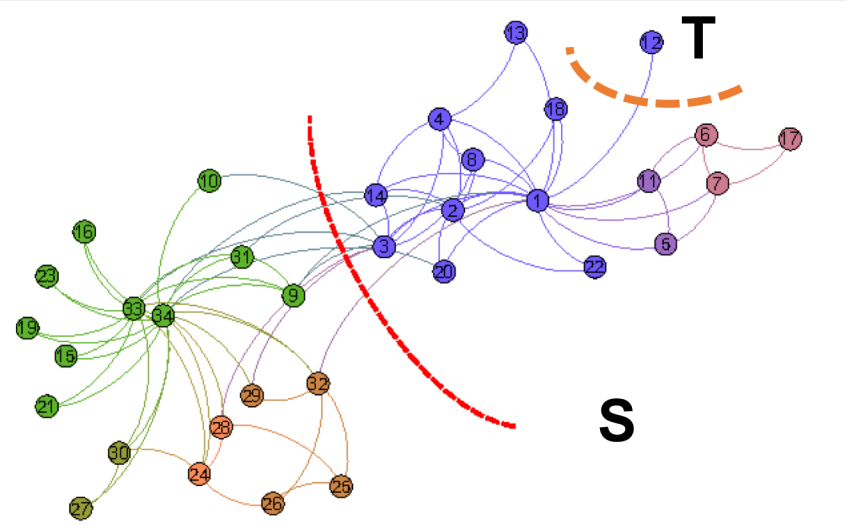
Simple Idea: Partition clusters along minimum cut in graph.



(a) Zachary Karate Club Graph

CUT MINIMIZATION

Simple Idea: Partition clusters along minimum cut in graph.

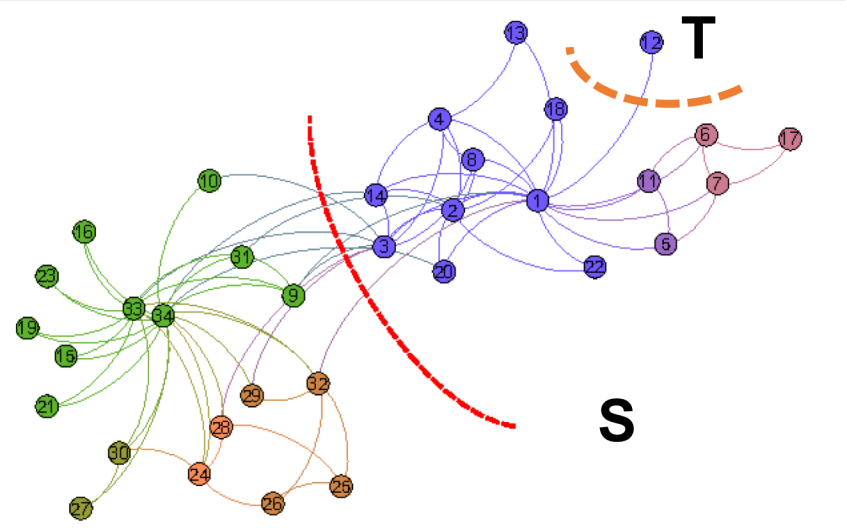


(a) Zachary Karate Club Graph

Small cuts are often not informative.

CUT MINIMIZATION

Simple Idea: Partition clusters along minimum cut in graph.



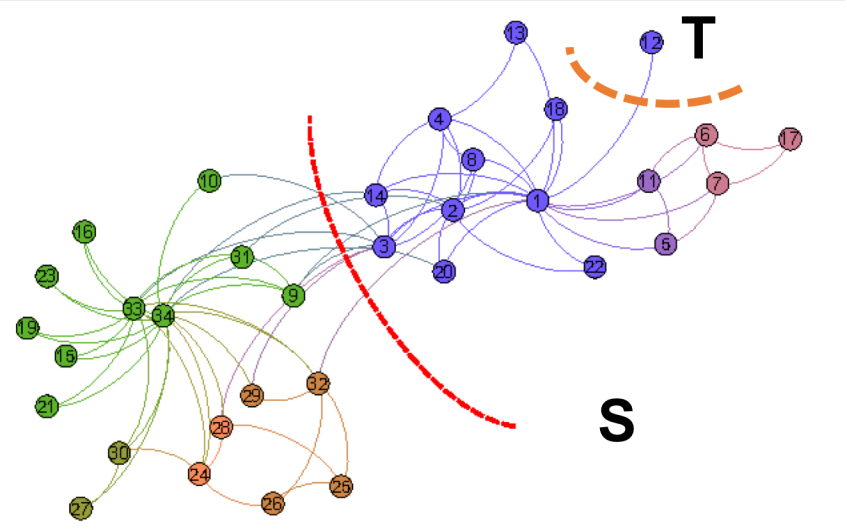
(a) Zachary Karate Club Graph

Small cuts are often not informative.

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

CUT MINIMIZATION

Simple Idea: Partition clusters along minimum cut in graph.



(a) Zachary Karate Club Graph

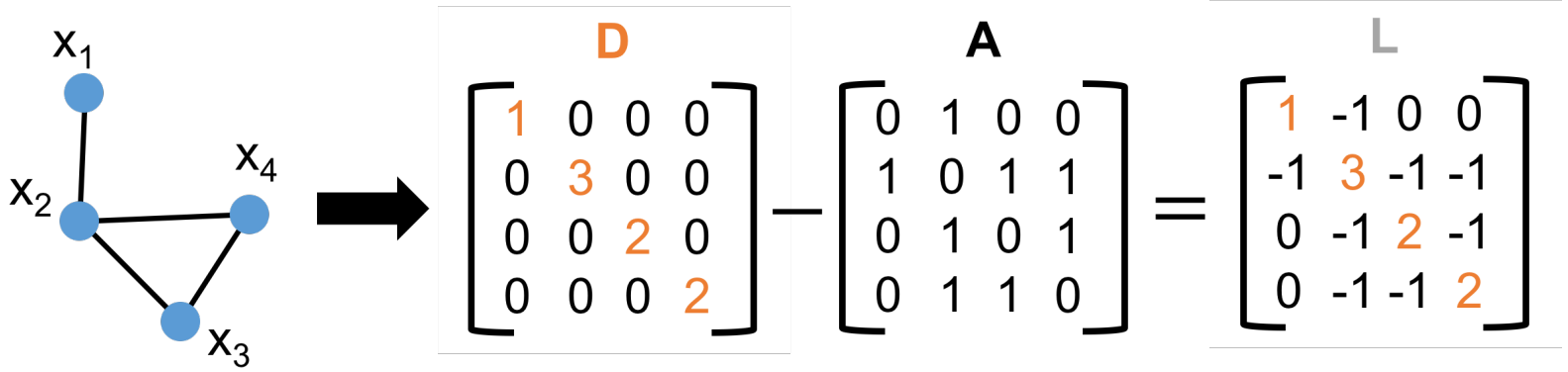
Small cuts are often not informative.

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., $\vec{v}^T \vec{1} \approx 0$.

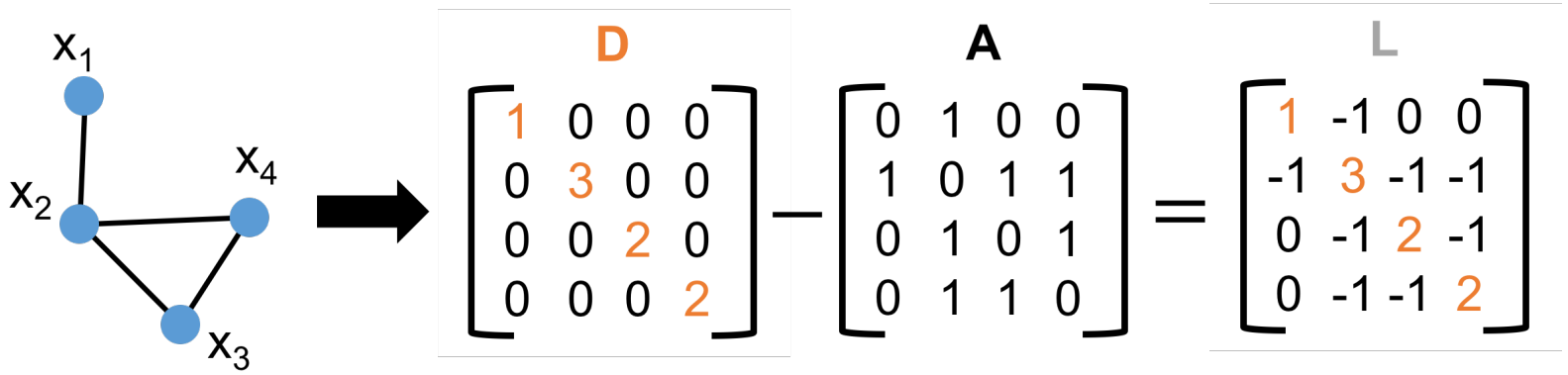
THE LAPLACIAN VIEW

For a graph with adjacency matrix **A** and degree matrix **D**, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the graph Laplacian.



THE LAPLACIAN VIEW

For a graph with adjacency matrix $\mathbf{A}$ and degree matrix $\mathbf{D}$, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the graph Laplacian.



For any vector $\vec{v}$, its 'smoothness' over the graph is given by:

$$\sum_{(i,j)\in E} (\vec{v}(i) - \vec{v}(j))^2 = \vec{v}^T \mathbf{L} \vec{v}.$$

REWRITING LAPLACIAN

Lemma:

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

REWRITING LAPLACIAN

Lemma:

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

Proof:

REWRITING LAPLACIAN

Lemma:

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

Proof:

- Let L_e be the Laplacian for graph just containing edge e.

REWRITING LAPLACIAN

Lemma:

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

Proof:

- Let L_e be the Laplacian for graph just containing edge e.
- By linearity,

$$\vec{v}^T \mathbf{L} \vec{v} = \sum_{e\in E} \vec{v}^T \mathbf{L}_e \vec{v}$$

REWRITING LAPLACIAN

Lemma:

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

Proof:

- Let L_e be the Laplacian for graph just containing edge e.
- By linearity,

$$\vec{v}^T \mathbf{L} \vec{v} = \sum_{e\in E} \vec{v}^T \mathbf{L}_e \vec{v}$$

- If $e = (i, j)$, then $\vec{v}^T \mathbf{L}_e \vec{v} = (v(i) - v(j))^2$