COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Fall 2020.

Lecture 19

LOGISTICS

- Week 10 Quiz is due Monday at 8pm.

SUMMARY

Last Class: Spectral Graph Theory

- View of a graph in terms of adjacency matrix and Laplacian.
- Spectral embedding for non-linear dimensionality reduction.
- Start on graph clustering for community detection and non-linear clustering.
- Idea of finding small cuts that separate large sets of nodes.

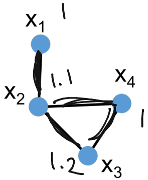
Last Class: Spectral Graph Theory

- View of a graph in terms of adjacency matrix and Laplacian.
- Spectral embedding for non-linear dimensionality reduction.
- Start on graph clustering for community detection and non-linear clustering.
- Idea of finding small cuts that separate large sets of nodes.

This Class: Spectral Clustering and the Stochastic Block Model

- Spectral clustering: finding good cuts via Laplacian eigenvectors.
- Stochastic block model: A simple clustered graph model where we can prove the effectiveness of spectral clustering.

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



$$\overset{\mathbf{D}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}} - \overset{\mathbf{A}}{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}} = \overset{\mathbf{L}}{\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}}$$

$v^T L v$ measures

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

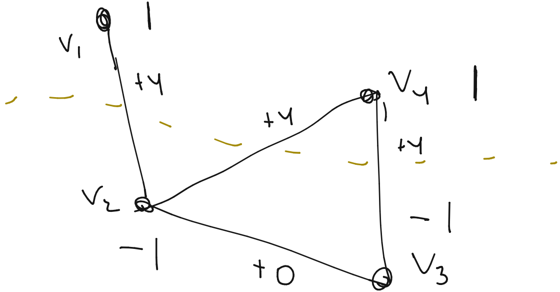
$$\sum_{(i,j)\in E} (\vec{v}(i) - \vec{v}(j))^2 = \vec{v}^T \mathbf{L} \vec{v}.$$

smaller v is "smoother"

THE LAPLACIAN VIEW

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 $i \in T$:

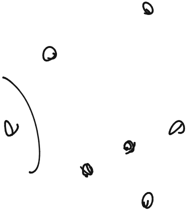
1. $\vec{v}^T \mathbf{L} \vec{V} = \sum_{(i,j)\in E} (\vec{v}(i) - \vec{v}(j))^2 = 4 \cdot cut(S, T)$.



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

$$\begin{bmatrix} 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \sum_{i=1}^{n} v(i)$$



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

Want to minimize both $\vec{v}^T \mathbf{L} \vec{v}$ (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 $i \in T$:

scalars

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

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

Next Step: See how this dual minimization problem is naturally solved (sort of) by eigendecomposition.

$v^T \underline{A} v = \underline{\sum v(i)v(j)} A_{ij}$ "quadratic form"

The smallest eigenvector of the Laplacian is:

$\|v\|_2^2 = 1$

$\sum_{i=1}^{n} v(i)^2 = \sum_{i=1}^{n} \frac{1}{\sqrt{n}^2} = 1$

$$\vec{v}_n = \frac{1}{\sqrt{n}} \cdot \vec{1} = \underset{v \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1}{\arg\min} \vec{v}^T \mathbf{L} \vec{v}$$

$$v_n = \begin{bmatrix} 1/\sqrt{n} \\ 1/\sqrt{n} \\ 1/\sqrt{n} \\ \vdots \\ 1/\sqrt{n} \end{bmatrix}$$

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

$v_n^T \lambda_n v_n = \lambda_n \|v_n\|_2^2$

$= \lambda_n$

for any v,

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

$\geq 0.$

$\underline{v_n^T L v_n} = \sum_{i,j \in E} [v_n(i) - v_n(j)]^2$

$\sum_{i,j \in E} [1/\sqrt{n} - 1/\sqrt{n}]^2 = \boxed{0}$

L

$\begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & & -1 \\ & & & \end{bmatrix} \begin{bmatrix} v_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$.

$$V_n = \begin{bmatrix} 1/\sqrt{n} \\ \vdots \\ 1/\sqrt{n} \end{bmatrix}$$

By Courant-Fischer, the second smallest eigenvector is given by:

$$\vec{v}_{n-1} = \underset{v \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1,\ \vec{v}_n^T\vec{v}=0}{\arg\min} \vec{v}^T \mathbf{L} \vec{v}$$

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n\times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n\times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n\times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$. S, T: vertex sets on different sides of cut.

By Courant-Fischer, the second smallest eigenvector is given by:

$$\vec{v}_{n-1} = \underset{v \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1,\ \vec{v}_n^T \vec{v}=0}{\arg\min} \vec{v}^T \mathbf{L} \vec{v}$$

If $\vec{v}_{n-1}$ were in $\left\{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right\}^n$ it would have: $\frac{1}{\sqrt{n}}\begin{bmatrix}1\\-1\\1\\ \vdots\\-1\end{bmatrix}$

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

$\begin{bmatrix}1 & -1 & 1 & -1\end{bmatrix}\begin{bmatrix}1\\1\\ \vdots\\1\end{bmatrix}$

$\frac{1}{\sqrt{n}}\vec{1}$

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n\times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n\times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n\times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$. S, T: vertex sets on different sides of cut.

eigenvectors of graph Laplacian $L = D - A$

By Courant-Fischer, the second smallest eigenvector is given by:

$$\vec{v}_{n-1} = \underset{v \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1,\ \vec{v}_n^T \vec{v}=0}{\arg\min} \vec{v}^T \mathbf{L} \vec{v}$$

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

- $\vec{v}_{n-1}^T L \vec{v}_{n-1} = \frac{4}{\sqrt{n}} \cdot cut(S,T)$ as small as possible given that $\vec{v}_{n-1}^T \vec{v}_n = \frac{1}{\sqrt{n}} \vec{v}_{n-1}^T \vec{1} = \frac{|T|-|S|}{n} = 0$.
- I.e., $\vec{v}_{n-1}$ would indicate the smallest perfectly balanced cut.

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$. S, T: vertex sets on different sides of cut.

By Courant-Fischer, the second smallest eigenvector is given by:

$$\vec{v}_{n-1} = \underset{v \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1,\ \vec{v}_n^T \vec{v}=0}{\arg\min} \vec{v}^T \mathbf{L} \vec{v}$$

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

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

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$. S, T: vertex sets on different sides of cut.

Find a good partition of the graph by computing

$$\vec{v}_{2} = \underset{v \in \mathbb{R}^d \text{ with } \|\vec{v}\|=1,\ \vec{v}_{2}^T \vec{1}=0}{\arg\min} \underline{\vec{v}^T \mathbf{L} \vec{v}}$$

Set S to be all nodes with $\vec{v}_2(i) < 0$, T to be all with $\vec{v}_2(i) \geq 0$.

Find a good partition of the graph by computing

$v_{n-1}^T 1 = 0$

$\sum_{i=1}^{n} v_{n-1}(i) = 0$

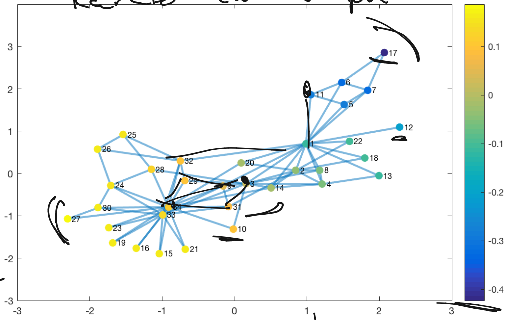
$$\vec{v}_2 = \operatorname*{arg\,min}_{v \in \mathbb{R}^d \text{ with } \|\vec{v}\|=1,\ \vec{v}^T\vec{1}=0} \vec{v}^T L \vec{v}$$

Set S to be all nodes with $\vec{v}_2(i) < 0$, T to be all with $\vec{v}_2(i) \geq 0$.

Karate Club Graph

32 nodes

$v_{n-1} = \begin{bmatrix} -1 \\ 2 \\ -.5 \\ 6 \\ .7 \\ \vdots \end{bmatrix}_{32}$



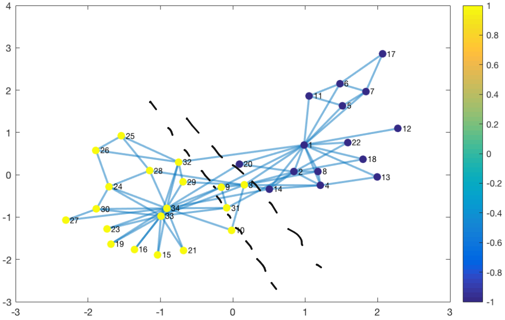
electrical connectivity

"relaxation"

Find a good partition of the graph by computing

$$\vec{v}_2 = \underset{v \in \mathbb{R}^d \text{with } \|\vec{v}\|=1,\ \vec{v}_2^T\vec{1}=0}{\arg\min} \vec{v}^T \mathbf{L} \vec{V}$$

Set S to be all nodes with $\vec{v}_2(i) < 0$, T to be all with $\vec{v}_2(i) \geq 0$.



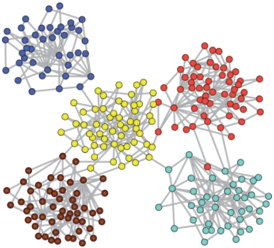
The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\bar{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$.

SPECTRAL PARTITIONING IN PRACTICE

The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\bar{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

Important Consideration: What to do when we want to split the graph into more than two parts?



n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$.

The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\overline{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

Important Consideration: What to do when we want to split the graph into more than two parts?

Spectral Clustering:

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n\times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n\times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n\times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$.

SPECTRAL PARTITIONING IN PRACTICE

The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\bar{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

Important Consideration: What to do when we want to split the graph into more than two parts?

$$\begin{bmatrix} L \end{bmatrix} \to n \overbrace{\begin{bmatrix} v_{n-1} & \cdots & v_{n-k} \end{bmatrix}}^{k}$$

Spectral Clustering:

- Compute smallest k nonzero eigenvectors $\vec{v}_{n-1}, \ldots, \vec{v}_{n-k}$ of $\bar{\mathbf{L}}$.

> n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$.

SPECTRAL PARTITIONING IN PRACTICE

The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\overline{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

Important Consideration: What to do when we want to split the graph into more than two parts?

Spectral Clustering:

- Compute smallest k nonzero eigenvectors $\vec{v}_{n-1}, \ldots, \vec{v}_{n-k}$ of $\overline{\mathbf{L}}$.
- Represent each node by its corresponding row in $\mathbf{V} \in \mathbb{R}^{n \times k}$ whose ~~rows~~ columns are $\vec{v}_{n-1}, \ldots \vec{v}_{n-k}$.

> n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$.

The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\overline{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

Important Consideration: What to do when we want to split the graph into more than two parts?

Spectral Clustering:

- Compute smallest k nonzero eigenvectors $\vec{v}_{n-1}, \ldots, \vec{v}_{n-k}$ of $\overline{\mathbf{L}}$.
- Represent each node by its corresponding row in $\mathbf{V} \in \mathbb{R}^{n \times k}$ whose rows are $\vec{v}_{n-1}, \ldots \vec{v}_{n-k}$.
- Cluster these rows using k-means clustering (or really any clustering method).

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$.

The smallest eigenvectors of $\mathbf{L} = \mathbf{D} - \mathbf{A}$ give the orthogonal 'functions' that are smoothest over the graph. I.e., minimize

$v^T L v$

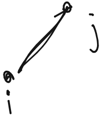
$$\vec{v}^T \mathbf{L} \vec{v} = \sum_{(i,j) \in E} [\vec{v}(i) - \vec{v}(j)]^2.$$

LAPLACIAN EMBEDDING

The smallest eigenvectors of $\mathsf{L} = \mathsf{D} - \mathsf{A}$ give the orthogonal 'functions' that are smoothest over the graph. I.e., minimize

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

Embedding points with coordinates given by $e_j =$ $[\vec{v}_{n-1}(j), \vec{v}_{n-2}(j), \ldots, \vec{v}_{n-k}(j)]$ ensures that coordinates connected by edges have minimum total squared Euclidean distance.



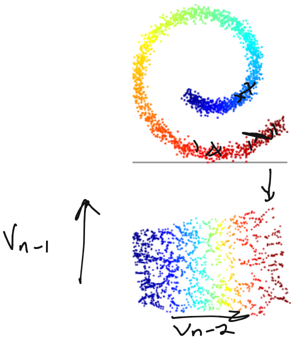
$$\|e_i - e_j\|_2^2 = \sum_{\ell=1}^{k} [V_{n-\ell}(i) - V_{n-\ell}(j)]^2$$

LAPLACIAN EMBEDDING

The smallest eigenvectors of $\mathbf{L} = \mathbf{D} - \mathbf{A}$ give the orthogonal 'functions' that are smoothest over the graph. I.e., minimize

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

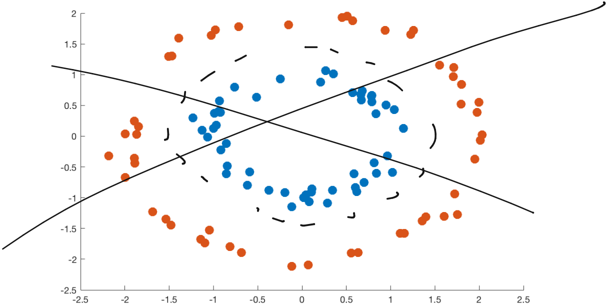
Embedding points with coordinates given by $[\vec{v}_{n-1}(j), \vec{v}_{n-2}(j), \ldots, \vec{v}_{n-k}(j)]$ ensures that coordinates connected by edges have minimum total squared Euclidean distance.



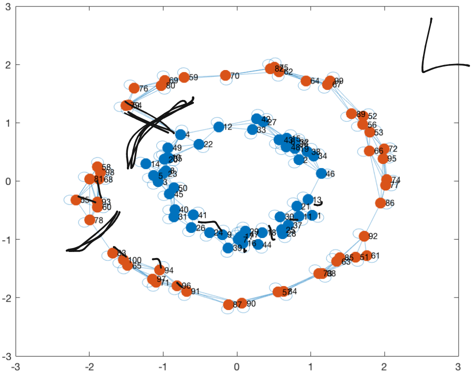
- Spectral Clustering
- Laplacian Eigenmaps
- Locally linear embedding
- Isomap
- Node2Vec, DeepWalk, etc. (variants on Laplacian)

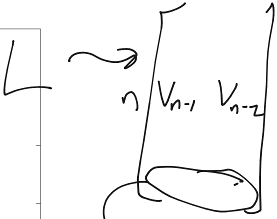
spectral clustering
to non-linearly separable.

Original Data: (not linearly separable)



k-Nearest Neighbors Graph:

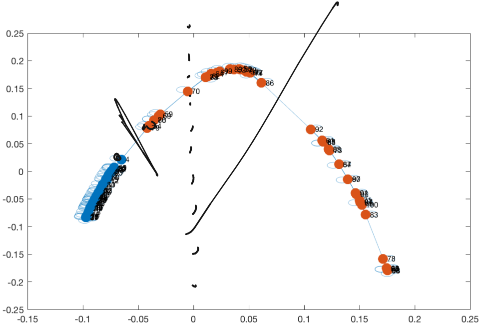




2-d embedding for each node

Embedding with eigenvectors $\vec{v}_{n-1}, \vec{v}_{n-2}$: (linearly separable)

V_{n-2}



V_{n-1}

So Far: Have argued that spectral clustering partitions a graph effectively, along a small cut that separates the graph into large pieces. But it is difficult to give any formal guarantee on the 'quality' of the partitioning in general graphs.

So Far: Have argued that spectral clustering partitions a graph effectively, along a small cut that separates the graph into large pieces. But it is difficult to give any formal guarantee on the 'quality' of the partitioning in general graphs.

Common Approach: Give a natural generative model for random inputs and analyze how the algorithm performs on inputs drawn from this model.

- Very common in algorithm design for data analysis/machine learning (can be used to justify least squares regression, k-means clustering, PCA, etc.)

Stochastic Block Model (Planted Partition Model): Let $G_n(p, q)$ be a distribution over graphs on n nodes, split randomly into two groups B and C, each with $n/2$ nodes.

Stochastic Block Model (Planted Partition Model): Let $G_n(p, q)$ be a distribution over graphs on n nodes, split randomly into two groups B and C, each with $n/2$ nodes.

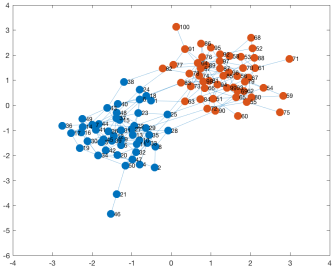
$\approx .2$

- Any two nodes in the same group are connected with probability p (including self-loops).
- Any two nodes in different groups are connected with prob. $q < p$.
- Connections are independent.

$\approx .05$

Stochastic Block Model (Planted Partition Model): Let $G_n(p, q)$ be a distribution over graphs on n nodes, split randomly into two groups B and C, each with $n/2$ nodes.

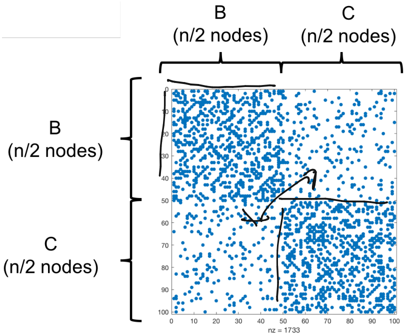
- Any two nodes in the same group are connected with probability p (including self-loops).
- Any two nodes in different groups are connected with prob. $q < p$.
- Connections are independent.



Erdos Renyi

Let G be a stochastic block model graph drawn from $G_n(p, q)$.

- Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of G, ordered in terms of group ID.

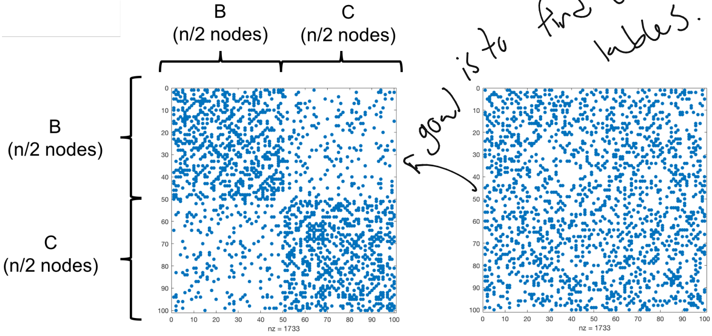


> $G_n(p, q)$: stochastic block model distribution. B, C: groups with $n/2$ nodes each. Connections are independent with probability p between nodes in the same group, and probability q between nodes not in the same group.

Let G be a stochastic block model graph drawn from $G_n(p,q)$.

- Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of G, ordered in terms of group ID.

goal is to find community lables.



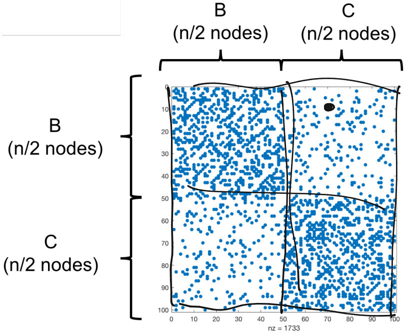
$G_n(p,q)$: stochastic block model distribution. B, C: groups with $n/2$ nodes each. Connections are independent with probability p between nodes in the same group, and probability q between nodes not in the same group.

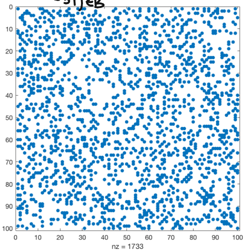
Let G be a stochastic block model graph drawn from $G_n(p, q)$.

- Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of G, ordered in terms of group ID. What is $\mathbb{E}[\mathbf{A}]$? — block matrix

$$\mathbb{E}_{i \in B, j \in C}[A_{ij}] = 1 \cdot q + 0 \cdot (1-q) = q$$

$$\mathbb{E}_{i \in B, j \in B}[A_{ij}] = p$$

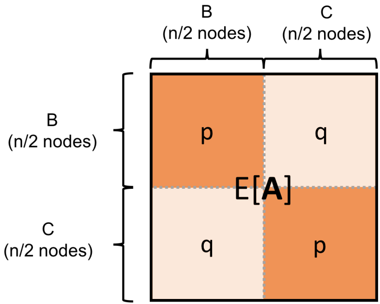




$G_n(p, q)$: stochastic block model distribution. B, C: groups with $n/2$ nodes each. Connections are independent with probability p between nodes in the same group, and probability q between nodes not in the same group.

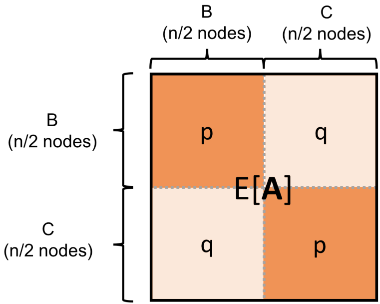
EXPECTED ADJACENCY SPECTRUM

Letting G be a stochastic block model graph drawn from $G_n(p, q)$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ be its adjacency matrix. $(\mathbb{E}[\mathbf{A}])_{i,j} = p$ for i, j in same group, $(\mathbb{E}[\mathbf{A}])_{i,j} = q$ otherwise.



> $G_n(p, q)$: stochastic block model distribution. B, C: groups with $n/2$ nodes each. Connections are independent with probability p between nodes in the same group, and probability q between nodes not in the same group.

Letting G be a stochastic block model graph drawn from $G_n(p,q)$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ be its adjacency matrix. $(\mathbb{E}[\mathbf{A}])_{i,j} = p$ for i,j in same group, $(\mathbb{E}[\mathbf{A}])_{i,j} = q$ otherwise.



What is rank$(\mathbb{E}[\mathbf{A}])$? What are the eigenvectors and eigenvalues of $\mathbb{E}[\mathbf{A}]$?

$G_n(p,q)$: stochastic block model distribution. B, C: groups with $n/2$ nodes each. Connections are independent with probability p between nodes in the same group, and probability q between nodes not in the same group.