Spectral Clustering

- Limitations of Distance Based Clustering
- Similarity Graphs: ϵ-neighborhood & (mutual) $k$nn graphs
- Graph Partition and Cuts
- Spectral Graph Theory
- (Un)Normalized Graph Laplacians
- Relation of Graph Laplacian and Partition
- Spectral Clustering into 2 Clusters
- Spectral Clustering into k Clusters

Imdad ullah Khan

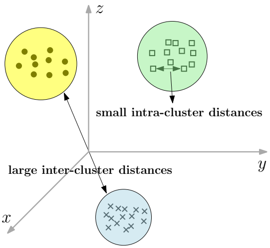
Limitations of Distance Based Clustering

Clustering: Definition

Clustering/cluster analysis/data segmentation

Grouping of objects into clusters such that objects in the same cluster are more similar and objects in different clusters are less similar

- Intra-cluster distances (between pairs of points in the same cluster)
- Inter-cluster distances (between pairs of points in different clusters)

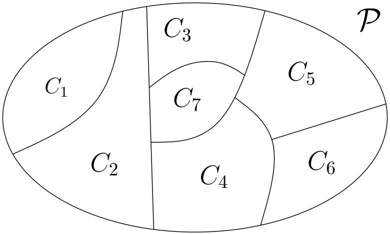


clustering hypothesis: Points in same cluster are semantically similar

Clustering: Definition

Generally, clustering produces a partition $[C_1, C_2, \ldots, C_k]$ of the dataset $\mathcal{P}$

- Each $C_i \subseteq \mathcal{P}$
- For $i \neq j$, $C_i \cap C_j = \emptyset$
- $\bigcup_{i=1}^{k} C_i = \mathcal{P}$



Clustering: Definition

Generally, clustering produces a partition $[C_1, C_2, \ldots, C_k]$ of the dataset $\mathcal{P}$

Two broadly different ways of clustering depending on input

Input: Given a dataset (feature vectors) and a proximity measure

Output: Clusters of the dataset into k clusters

Alternatively,

Input: Given pairwise proximity values for a (abstractly described) dataset (e.g. distance or similarity matrix)

Output: Clusters of the dataset into k clusters

The number of clusters k may or may not be part of the input (fixed)

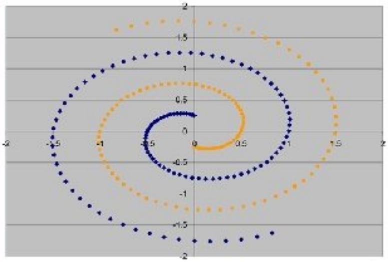
Basic Clustering Methods

- Broadly clustering methods are
  - Distance Based
  - Density and grid-based methods
  - Generative Model based
  - Other methods used for specific data types
    - e.g. for graph data we used connectivity based clustering
- It is possible that different clustering methods generate different clusterings of same data set

Distance measure does not always capture semantic clustering in the data

Limitation of distance-based clustering

Distance measure does not always capture semantic clustering in the data



Dataset exhibits complex cluster shapes

⇒ **K-means** performs very poorly in this space due bias toward dense spherical clusters.

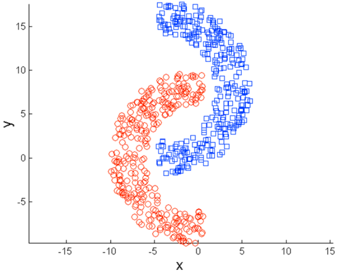
Relationship vs. Geometry Distance

source: J. Fan, UNC

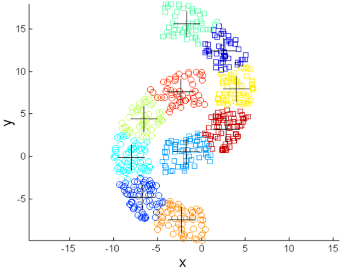
Limitation of distance-based clustering

Distance measure does not always capture semantic clustering in the data

source: J. Fan, UNC



Original Points in 2 Clusters

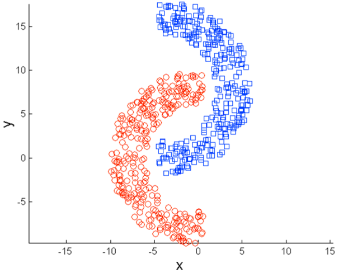


Output of k-means $(k = 10)$

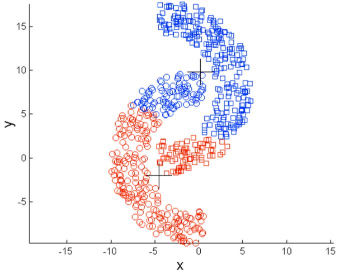
Limitation of distance-based clustering

Distance measure does not always capture semantic clustering in the data

source: J. Fan, UNC



Original Points in 2 Clusters



Output of k-means ($k = 2$)

Similarity Graphs

Similarity Graphs

Graph-based Representation of Data Relationships

Using Graphs to summarize proximity between pairs of Points

Some datasets are already Graphs – Assume adjacency capture relationships

- Web graphs
- Protein-Protein Interaction Networks
- Social Networks
- Coauthorship or Citation Networks

Similarity Graphs

Pairwise proximity is represented with graphs

Given similarity information ▷ e.g. proximity matrix of abstract objects

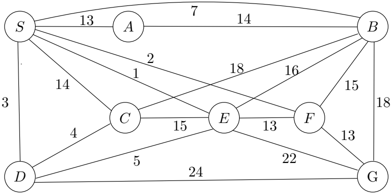
- $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ ▷ Feature vectors or abstractly described
- $n \times n$ proximity matrix: $S(i,j) = sim(\mathbf{x}_i, \mathbf{x}_j)$ ▷ Could be distance

Represent the data by a weighted graph $G = (V, E)$

- $V = \mathcal{P}$
- E : Make an edge b/w vertices with weight = pairwise similarity

Similarity Matrix

	S	A	B	C	D	E	F	G
S	∞	13	7	14	3	1	2	0
A	13	∞	14	0	0	0	0	0
B	7	14	∞	18	0	16	15	18
C	14	0	18	∞	4	15	0	0
D	3	0	0	4	∞	5	0	24
E	1	0	16	15	5	∞	13	22
F	2	0	15	0	0	13	∞	13
G	0	0	18	0	24	22	13	∞



Similarity Graph: ϵ-neighborhood Graph

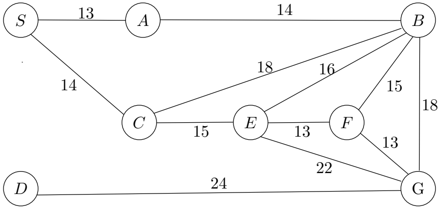
ϵ-neighborhood graphs: points are vertices and two vertices are adjacent if there distance is at most ϵ

No need for weights, as similarities are thresholded

Similarity Matrix

	S	A	B	C	D	E	F	G
S	∞	13	7	14	3	1	2	0
A	13	∞	14	0	0	0	0	0
B	7	14	∞	18	0	16	15	18
C	14	0	18	∞	13	15	0	0
D	3	0	0	13	∞	12	0	24
E	1	0	16	15	12	∞	13	22
F	2	0	15	0	0	13	∞	13
G	0	0	18	0	24	22	13	∞

Similarity Graph thresholded by 9



ϵ-neighborhood graphs are usually constructed from normalized distance matrix

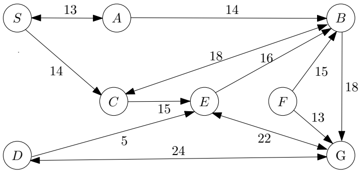
Similarity Graph: k-NN Graph

k-NN Graph: $(v_i, v_j) \in E$ if $v_j \in k\text{-NN}(v_i)$ OR $v_i \in k\text{-NN}(v_j)$

Similarity Matrix

	S	A	B	C	D	E	F	G
S	∞	13	7	14	3	1	2	0
A	13	∞	14	0	0	0	0	0
B	7	14	∞	18	0	16	15	18
C	14	0	18	∞	4	15	0	0
D	3	0	0	4	∞	5	0	24
E	1	0	16	15	5	∞	12	22
F	2	0	15	0	0	12	∞	13
G	0	0	18	0	24	22	13	∞

Every vertex has 2 nearest neighbors as outneighbors



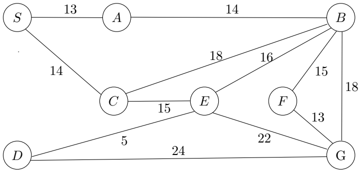
Make edge v_i to $v_j \in k\text{-NN}(v_i)$ ▷ nearest neighbors are not symmetric

Make G undirected by ignoring directions ▷ **OR** of nearest neighbors

Similarity Matrix

	S	A	B	C	D	E	F	G
S	∞	13	7	14	3	1	2	0
A	13	∞	14	0	0	0	0	0
B	7	14	∞	18	0	16	15	18
C	14	0	18	∞	4	15	0	0
D	3	0	0	4	∞	5	0	24
E	1	0	16	15	5	∞	12	22
F	2	0	15	0	0	12	∞	13
G	0	0	18	0	24	22	13	∞

A pair is adjacent if either is 2-NN of the other



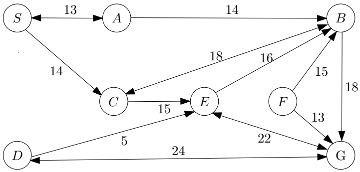
Similarity Graph: Mutual k-NN Graph

Mutual k-NN Graph: $(v_i, v_j) \in E$ if $v_j \in k\text{-NN}(v_i)$ AND $v_i \in k\text{-NN}(v_j)$

Similarity Matrix

	S	A	B	C	D	E	F	G
S	∞	13	7	14	3	1	2	0
A	13	∞	14	0	0	0	0	0
B	7	14	∞	18	0	16	15	18
C	14	0	18	∞	4	15	0	0
D	3	0	0	4	∞	5	0	24
E	1	0	16	15	5	∞	12	22
F	2	0	15	0	0	12	∞	13
G	0	0	18	0	24	22	13	∞

Every vertex has 2 nearest neighbors as outneighbors



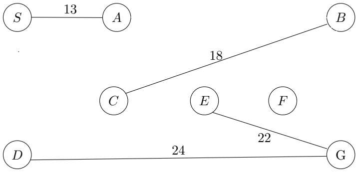
Make edge v_i to $v_j \in k\text{-NN}(v_i)$ ▷ nearest neighbors are not symmetric

Keep only bidirectional edges ▷ **AND** of nearest neighbors

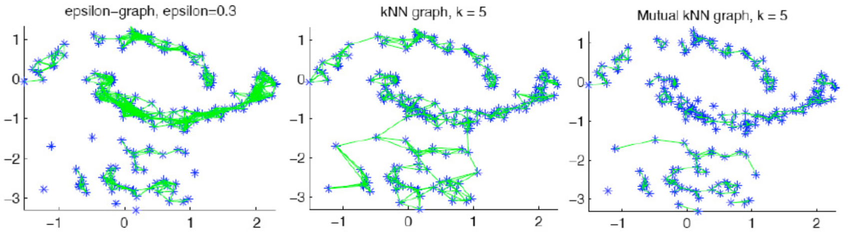
Similarity Matrix

	S	A	B	C	D	E	F	G
S	∞	13	7	14	3	1	2	0
A	13	∞	14	0	0	0	0	0
B	7	14	∞	18	0	16	15	18
C	14	0	18	∞	4	15	0	0
D	3	0	0	4	∞	5	0	24
E	1	0	16	15	5	∞	12	22
F	2	0	15	0	0	12	∞	13
G	0	0	18	0	24	22	13	∞

Two vertices are adjacent if both are each other's 2-NN



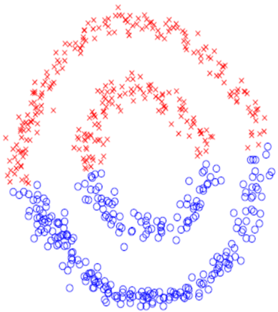
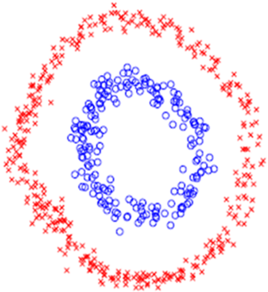
Similarity Graph: Advantages



source: J. Fan, UNC

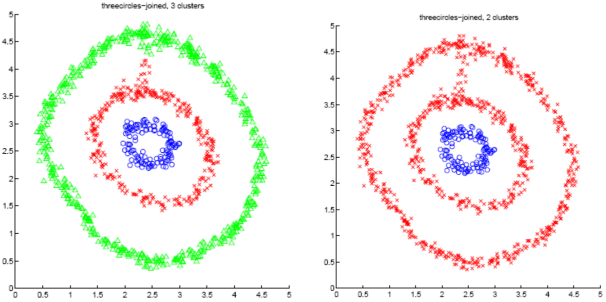
- Graphs capture local neighborhoods
- Can reliably indicate which points are "similar" or close
- Similarity values reliably encode local structure
- The similarity matrix doesn't capture global structure

Similarity Graph: Advantages

- Graphs capture local neighborhoods
- Can reliably indicate which points are "similar" or close
- Similarity values reliably encode local structure
- The similarity matrix doesn't capture global structure

Similarity Graph: Advantages



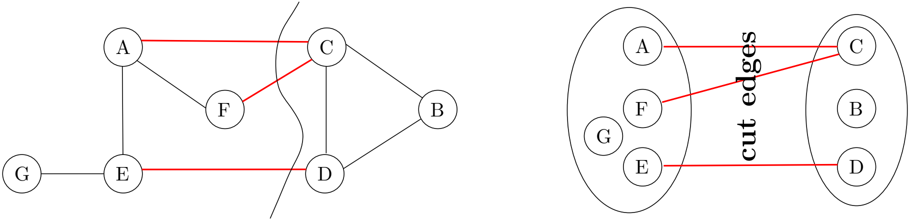
- Graphs capture local neighborhoods
- Can reliably indicate which points are "similar" or close
- Similarity values reliably encode local structure
- The similarity matrix doesn't capture global structure

Graph Partition Using Cut

Cuts in Graphs

A cut in G is a subset $S \subset V$

- Denoted as $[S, \overline{S}]$
- $S = \emptyset$ and $S = V$ are trivial cuts, we assume that $\emptyset \neq S \neq V$
- A graph on n vertices has 2^n cuts
- An edge (u, v) is crossing the cut $[S, \overline{S}]$, if $u \in S$ and $v \in \overline{S}$

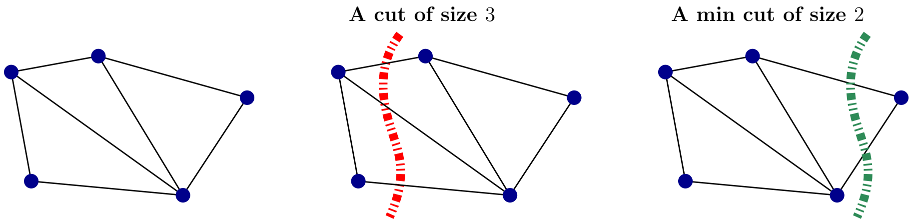


The MIN-CUT(G) problem

A cut in G is a subset $S \subset V$

- Denoted as $[S, \overline{S}]$
- An edge (u, v) is crossing the cut $[S, \overline{S}]$, if $u \in S$ and $v \in \overline{S}$

Size (or cost) of a cut in the number of crossing edges



- In weighted graph size of cut is the sum of weights of crossing edges

Graph Bi-Partition Using Cut

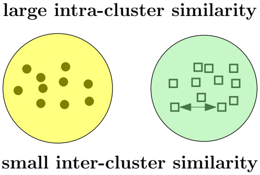
We can find a minimum-cut in the graph to separate clusters of objects

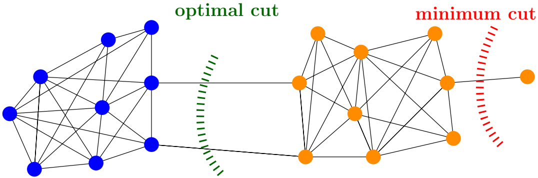
- Partition V into $[S, \overline{S}]$, that minimizes

$$cut(S, \overline{S}) = \sum_{(u,v) \in E, u \in A, v \in \overline{S}} \mathbf{1}$$

Attempts to minimize inter-cluster(s) similarity but

- Does not consider maximizing intra-cluster(s) similarity
- May find trivial cut $([\{v\}, \overline{\{v\}}])$, i.e. doesn't consider size of clusters





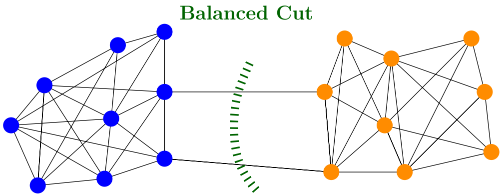
Graph BiPartition Using Balanced -Cut

To avoid trivial cuts we change the objective function of cut

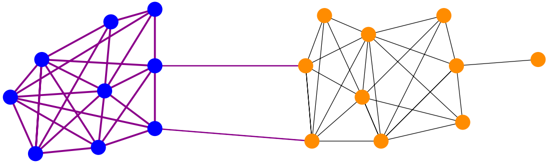
- Partition V into $[S, \overline{S}]$, such that $|S| = |\overline{S}|$ that minimizes

$$cut(S, \overline{S}) = \sum_{(u,v) \in E, u \in A, v \in \overline{S}} \mathbf{1}$$

- Technically, one requires $|S| = |\overline{S}| \pm 1$
- More generally, $|S|, |\overline{S}| \geq \alpha n$



Graph BiPartition Using Ratio-Cut



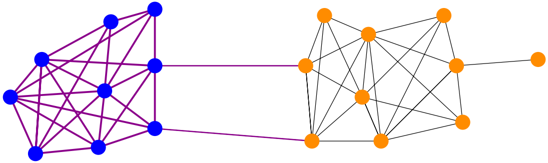
A slightly relaxed way for good and balanced cut is to minimize Ratio-Cut

- Partition V into $[S, \overline{S}]$ that minimizes

$$\text{Ratio-Cut}(S, \overline{S}) \quad := \quad \frac{cut(S, \overline{S})}{|S|} + \frac{cut(S, \overline{S})}{|\overline{S}|}$$

Graph BiPartition Using Normalized-Cut

For $A \subset V$, let $vol(A) = \sum_{x \in A} deg(x)$



- Consider connectivity between groups relative to density of each group
- Partition V into $[S, \overline{S}]$, that minimizes

$$\text{normalized-cut}(S, \overline{S}) \ = Ncut(S, \overline{S}) := \frac{cut(S, \overline{S})}{vol(S)} + \frac{cut(S, \overline{S})}{vol(\overline{S})}$$

- Consider both inter-cluster(s) and intra-cluster similarity
- Generally produces more balanced partitions

Graph Partition Using Cut

Finding minimum cut is easy (recall max-flow-min-cut theorem and Karger-Stein Algo)

Finding optimal balanced, ratio and normalized-cut are NP-HARD

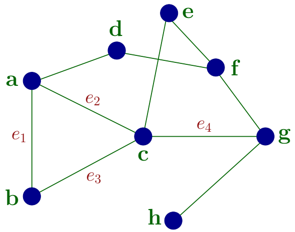
Spectral Graph Theory

Spectral Graph theory

- Spectral graph theory is using techniques from linear algebra to solve graph theory problems
- Particularly, what combinatorial properties of the graphs are implied by the eigenvalues and eigenvectors of the matrices associated with graphs

Adjacency Matrix of Graphs

Adjacency Matrix



$A =$

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

A is symmetric and real $\implies$ all eigenvectors are real and orthogonal

Let $\mathbf{x} \in \mathbb{R}^n$ ▷ coordinates corresponding to vertices (fixed order)

Meaning of $A\mathbf{x} = \mathbf{y}$? ▷ $\mathbf{y}_i$ is sum of $\mathbf{x}_j$ of all neighbors of $v_i \in V$

Adjacency Matrix of Graphs

Adjacency Matrix

Suppose G is a d-regular connected graph ▷ every vertex has degree d

Let $\mathbf{x} = \left[1, 1, \ldots, 1\right]^T$

$$A\mathbf{x} = d\mathbf{x}$$

$\mathbf{x}$ is an eigenvector of A with eigenvalue d

Adjacency Matrix

Suppose G is d-regular with 2 components $\{v_1, \ldots, v_k\}$, $\{v_{k+1}, \ldots, v_n\}$

$$\text{Let } \mathbf{x_1} \;=\; \Big[\underbrace{1, 1, \ldots, 1}_{k}, \underbrace{0, 0, \ldots, 0}_{n-k}\Big]^T$$

$$A\mathbf{x_1} \;=\; d\mathbf{x_1}$$

$$\text{Let } \mathbf{x_s} \;=\; \Big[\underbrace{0, 0, \ldots, 0}_{k}, \underbrace{1, 1, \ldots, 1}_{n-k}\Big]^T$$

$$A\mathbf{x_2} \;=\; d\mathbf{x_2}$$

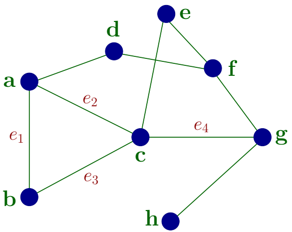
$\mathbf{x_1}$ and $\mathbf{x_2}$ are eigenvectors of A with eigenvalues d

Degree Matrix of Graphs

Degree Matrix

D is an $n \times n$ diagonal matrix

$D = [d_{ii}]$ $\quad\triangleright\ d_{ii} = deg(v_i)$



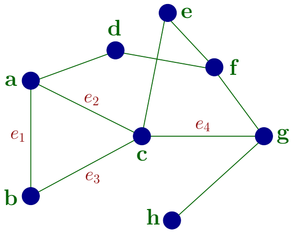
$D =$

	a	b	c	d	e	f	g	h
a	3	0	0	0	0	0	0	0
b	0	2	0	0	0	0	0	0
c	0	0	4	0	0	0	0	0
d	0	0	0	2	0	0	0	0
e	0	0	0	0	2	0	0	0
f	0	0	0	0	0	3	0	0
g	0	0	0	0	0	0	3	0
h	0	0	0	0	0	0	0	1

Laplacian Matrix of Graphs

Laplacian Matrix, $L = D - A$

$$L(i,j) = \begin{cases} d_i & \text{if } i = j \\ -1 & \text{if } i \sim j \\ 0 & \text{else} \end{cases}$$



$L =$

	a	b	c	d	e	f	g	h
a	3	-1	-1	-1	0	0	0	0
b	-1	2	-1	0	0	0	0	0
c	-1	-1	4	0	-1	0	-1	0
d	-1	0	0	2	0	-1	0	0
e	0	0	-1	0	2	-1	0	0
f	0	0	0	-1	-1	3	-1	0
g	0	0	-1	0	0	-1	3	-1
h	0	0	0	0	0	0	-1	1

Laplacian Matrix of Graphs

Laplacian Matrix, $L = D - A$

$$L(i,j) = \begin{cases} d_i & \text{if } i = j \\ -1 & \text{if } i \sim j \\ 0 & \text{else} \end{cases}$$

Let $\mathbf{x} = \left[1, 1 \ldots, 1\right]^T$

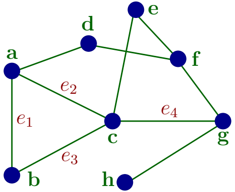
$L\mathbf{x} = \mathbf{0} = 0\mathbf{x}$

$\mathbf{x}$ is a (trivial) eigenvector

Laplacian Matrix, $L = D - A$

For $e = (u, v) \in E$, define matrix L_e as

$$L_e(i,j) = \begin{cases} 1 & \text{if } i = j \ \& \ i \in \{u, v\} \\ -1 & \text{if } (i,j) = (u,v) \text{ or } (i,j) = (v,u) \\ 0 & \text{else} \end{cases}$$



$\mathbf{L_{e_1}}$	a	b	c	d	e	f	g	h
a	1	-1	0	0	0	0	0	0
b	-1	1	0	0	0	0	0	0
c	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0
e	0	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0	0
g	0	0	0	0	0	0	0	0
h	0	0	0	0	0	0	0	0

$\mathbf{L_{e_4}}$	a	b	c	d	e	f	g	h
a	0	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0	0
c	0	0	1	0	0	0	-1	0
d	0	0	0	0	0	0	0	0
e	0	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0	0
g	0	0	-1	0	0	0	1	0
h	0	0	0	0	0	0	0	0

Laplacian Matrix of Graphs

Laplacian Matrix, $L = D - A$

$$L(i,j) = \begin{cases} d_i & \text{if } i = j \\ -1 & \text{if } i \sim j \\ 0 & \text{else} \end{cases}$$

For $e = (u, v) \in E$, matrix L_e

$$L_e(i,j) = \begin{cases} 1 & \text{if } i = j \,\&\, i \in \{u, v\} \\ -1 & \text{if } (i,j) = (u,v) \text{ or } (i,j) = (v,u) \\ 0 & \text{else} \end{cases}$$

L_e is actually contribution of $e = (u, v)$ to L. So

$$L = \sum_{e \in E} L_e$$

Graph Laplacian

For edge $e = (u, v)$ $\quad L_e(i,j) = \begin{cases} 1 & \text{if } i = j \ \& \ i \in \{u, v\} \\ -1 & \text{if } (i,j) = (u,v) \text{ or } (i,j) = (v,u) \\ 0 & \text{else} \end{cases}$

Graph Laplacian, $\quad L = D - A = \sum_{e \in E} L_e$

For any vector $\mathbf{x} \in \mathbb{R}^n$, $\quad \mathbf{x}^T L_e \mathbf{x} = (x_i - x_j)^2$

$$\mathbf{x}^T L\mathbf{x} = \mathbf{x}^T (L_{e_1} + L_{e_2} + \ldots + L_{e_s})\mathbf{x} = \mathbf{x}^T \sum_{(i,j) \in E} L_{(i,j)} \mathbf{x}$$

$$= \sum_{(i,j) \in E} \left(\mathbf{x}^T L_{(i,j)} \mathbf{x}\right) = \sum_{(i,j) \in E} (x_i - x_j)^2$$

Relation of Graph Laplacian with Balanced Bi-partition

Graph Laplacian, $L = D - A$

For any vector $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^T L\mathbf{x} = \sum_{(i,j)\in E} (x_i - x_j)^2$

The optimization problem of balanced bipartition

Assign labels to vertices $x_i = \begin{cases} 1 & \text{if } i \in S \\ -1 & \text{if } i \in \overline{S} \end{cases}$

Find $\mathbf{x} \in \{-1, 1\}^n$ such that

- $\sum_{(i,j)\in E} (x_i - x_j)^2$ is minimum ▷ few cut edges
- $\sum_{x_i=-1} = \sum_{x_i=1} \implies \sum_i x_i = 0$ ▷ Balanced parts

Graph Laplacian

Graph Laplacian, $L = D - A$

- L is real & symmetric
- For any $\mathbf{x} \in R^n$ $\quad \mathbf{x}^T L\mathbf{x} = \sum_{(i,j)\in E} (x_i - x_j)^2$
- L is positive semidefinite $\quad \forall \mathbf{x} \in \mathbb{R}^n, \ \mathbf{x} \neq \mathbf{0} \ \ \mathbf{x}^T L\mathbf{x} \geq 0$

Theorem (Spectral Theorem)

If M is a real, symmetric and positive semidefinite $n \times n$ matrix, then M has n real, non-negative, orthonormal eigen values and eigen vectors

By Spectral Theorem, L has n real eigen values $\lambda_1, \ldots, \lambda_n$

$$0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

Graph Laplacian and Graph Components

Relation of Graph Laplacian with Graph partition

Find $\mathbf{x} \in \{-1, 1\}^n$ such that

- $\sum_{(i,j)\in E} (x_i - x_j)^2$ is minimum ▷ few cut edges
- $\sum_{x_i=-1} = \sum_{x_i=1} \implies \sum_i x_i = 0$ ▷ Balanced

Since $\mathbf{x}^T L\mathbf{x} = \sum_{(i,j)\in E} (x_i - x_j)^2$, minimizing $\mathbf{x}^T L\mathbf{x}$ is a continuous relaxation of the discrete optimization problem

In the following we first see connections between graph connectivity and eigenvalues/vectors of L, which provides the basis of spectral clustering

Relation of Graph Laplacian with Graph partition

For any $\mathbf{x} \in \mathbb{R}^n$, $\quad \mathbf{x}^T L\mathbf{x} \;=\; \sum_{(i,j)\in E} (x_i - x_j)^2$

$$0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

If $\lambda \mathbf{x} \;=\; L\mathbf{x}$, $\quad$ then $\quad \lambda \;=\; \dfrac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$ $\qquad \triangleright$ Called quadratic form

$\lambda_1 = 0$, $\quad$ consider $\quad \mathbf{x} = \Big[1, 1, \ldots, 1\Big]^T \in \mathbb{R}^n$

$$\mathbf{x}^T L\mathbf{x} \;=\; \sum_{(i,j)\in E} (x_i - x_j)^2 \;=\; 0$$

or $L\,\mathbf{1} = 0 \cdot \mathbf{1}$

Thus $\lambda_1 = 0$ for any L, and $\mathbf{x}_1 = \Big[1, 1, \ldots, 1\Big]^T$ is the eigenvector

Relation of Graph Laplacian with Graph partition

Let L be the Laplacian of a graph $G = (V, E)$, $|V| = n$

$$0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

If $\lambda \mathbf{x} = L\mathbf{x}$, then $\lambda = \dfrac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$

$$\lambda_1 = \min_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \quad \text{(say } \mathbf{x_1} \text{ is a solution)}$$

$$\lambda_2 = \min_{\mathbf{x} \neq 0, \mathbf{x} \perp \mathbf{x_1}} \frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \quad \text{(say } \mathbf{x_2} \text{ is a solution)}$$

$$\lambda_3 = \min_{\mathbf{x} \neq 0, \mathbf{x} \perp \mathbf{x_1}, \mathbf{x_2}} \frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \quad \text{(yield } \mathbf{x_3} \text{ and so on)}$$

Relation of Graph Laplacian with Graph partition

Let L be the the Laplacian of a graph $G = (V, E)$, $|V| = n$

$$0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

Theorem

$\lambda_1 = 0$, *and* $\mathbf{x_1} = \left[1, 1, \ldots, 1\right]^T \in \mathbb{R}^n$ *is the corresponding eigenvector*

Observe it for a few graphs, at least the following

Relation of Graph Laplacian with Graph partition

Theorem

If G is connected, then $\lambda_2 > 0$

$$\mathbf{x_1} = \left[1, \ldots, 1\right]^T, \quad \mathbf{x} \perp \mathbf{x_1} \implies \mathbf{x}^T \mathbf{1} = 0 \implies \sum_{i=1}^{n} x_i = 0$$

$$\lambda_2 = \min_{\mathbf{x} \neq 0, \mathbf{x} \perp \mathbf{x_1}} \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \implies \lambda_2 = \min_{\mathbf{x} \neq \mathbf{0}, \sum_{i=1}^{n} x_i = 0} \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$$

$$\lambda_2 = \min_{\mathbf{x} \neq \mathbf{0}, \sum_{i=1}^{n} x_i = 0} \sum\nolimits_{(i,j) \in E} (x_i - x_j)^2$$

$$\mathbf{x_2} = argmin_{\mathbf{x} \neq \mathbf{0}, \sum_{i=1}^{n} x_i = 0} \sum_{(i,j) \in E} (x_i - x_j)^2$$

Relation of Graph Laplacian with Graph partition

Theorem

If G is connected, then $\lambda_2 > 0$

$$\lambda_2 = 0 \implies \min_{\mathbf{x} \neq \mathbf{0}, \sum_{i=1}^{n} x_i = 0} \sum_{(i,j) \in E} (x_i - x_j)^2 = 0$$

- for all edges $(u, v) \in E$ we have $x_u = x_v$
- x_i's must be equal across any path
- G is connected, there is a path b/w all u and v
- x_i should be equal for all vertices
- $\mathbf{x} = \alpha \cdot [1, 1, \ldots, 1]^T$
- $\mathbf{x}$ is not orthogonal to $\mathbf{x_1}$, or $\mathbf{x} = \mathbf{0}$

Relation of Graph Laplacian with Graph partition

Theorem

If G is connected if and only if $\lambda_2 > 0$

$$\lambda_2 = 0 \implies \min_{\mathbf{x} \neq \mathbf{0}, \sum_{i=1}^{n} x_i = 0} \sum_{(i,j) \in E} (x_i - x_j)^2 = 0$$

Suppose G has two components, $S, \overline{S}$, then

$$\mathbf{x_1} = \Big[\underbrace{1, 1, \ldots, 1}_{i \in S}, \underbrace{0, 0, \ldots, 0}_{i \in \overline{S}}\Big]^T \quad \text{and} \quad \mathbf{x_2} = \Big[\underbrace{0, 0, \ldots, 0}_{i \in S}, \underbrace{1, 1, \ldots, 1}_{i \in \overline{S}}\Big]^T$$

are clearly eigen vectors with eigen values 0

This very easy to verify by reasoning very similar to the above

Relation of Graph Laplacian with Graph partition

Theorem

If G is connected if and only if $\lambda_2 > 0$

Theorem

The multiplicity k of the 0 *eigen values of* $L(G)$ *is equal to the number of connected components of G*

This can be proved with the same reasoning as above

We want to see how robust these results are.

If $\lambda_2 = 0$, graph has 2 components

If $\lambda_2 = 0.00001$, graph has 2 sparsely connected parts

Relation of Graph Laplacian with Graph partition

Theorem

The multiplicity k of the 0 eigen values of $L(G)$ is equal to the number of connected components of G

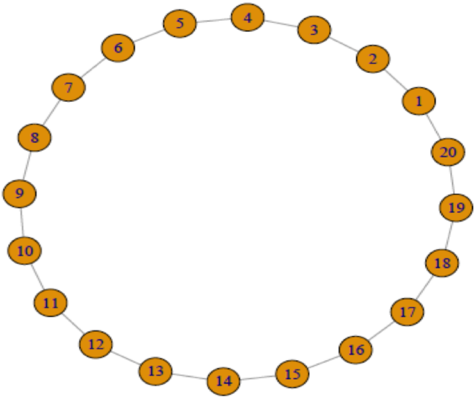
We want to see how robust these results are.

If $\lambda_2 = 0$, graph has 2 components

If $\lambda_2 = 0.00001$, graph has 2 sparsely connected parts

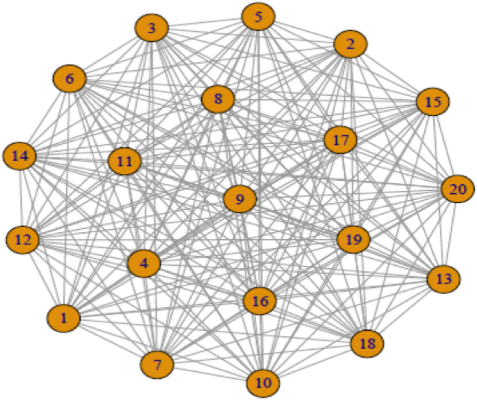
Lets observe the above theorem with the few examples

Relation of Graph Laplacian with Graph partition



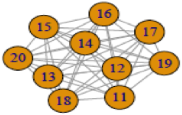
λ_1 1.78E-15	λ_2 9.79E-02	λ_3 9.79E-02
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
-0.2236068	0.3162278	0
-0.2236068	0.3007505	0.09771975
-0.2236068	0.2558336	0.185874
-0.2236068	0.185874	0.2558336
-0.2236068	0.09771975	0.3007505
-0.2236068	1.143599E-14	0.3162278
-0.2236068	-0.09771975	0.3007505
-0.2236068	-0.185874	0.2558336
-0.2236068	-0.2558336	0.185874
-0.2236068	-0.3007505	0.09771975
-0.2236068	-0.3162278	4.664828E-16
-0.2236068	-0.3007505	-0.09771975
-0.2236068	-0.2558336	-0.185874
-0.2236068	-0.185874	-0.2558336
-0.2236068	-0.09771975	-0.3007505
-0.2236068	1.104329E-14	-0.3162278
-0.2236068	0.09771975	-0.3007505
-0.2236068	0.185874	-0.2558336
-0.2236068	0.2558336	-0.185874
-0.2236068	0.3007505	-0.09771975

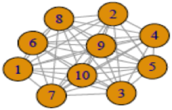
Relation of Graph Laplacian with Graph partition



λ_1 0	λ_2 20	λ_3 20
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
-0.2236068	-0.020559652	0
-0.2236068	0.108807659	0.04273661
-0.2236068	-0.803711507	0.0115485
-0.2236068	-0.191901761	-0.10229988
-0.2236068	-0.222135891	0.06658875
-0.2236068	-0.123007583	-0.15801353
-0.2236068	-0.084541146	0.45265136
-0.2236068	-0.041279066	0.10869576
-0.2236068	0.006026719	-0.39124179
-0.2236068	0.006026719	0.23205375
-0.2236068	0.045614464	0.02944336
-0.2236068	0.045614464	-0.56047404
-0.2236068	0.102436277	-0.29818941
-0.2236068	0.095619485	0.1104163
-0.2236068	0.095619485	0.1104163
-0.2236068	0.132014323	0.1210102
-0.2236068	0.201951467	0.27869041
-0.2236068	0.201951467	0.05981768
-0.2236068	0.208431342	-0.01290196
-0.2236068	0.237022732	-0.10094836

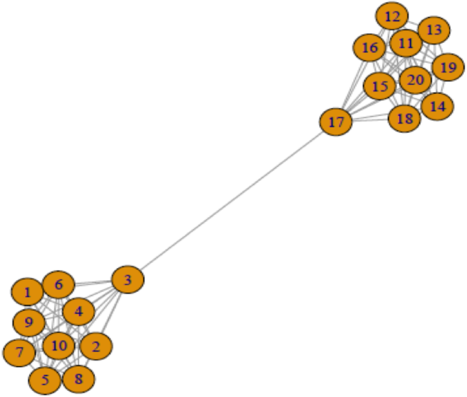
Relation of Graph Laplacian with Graph partition





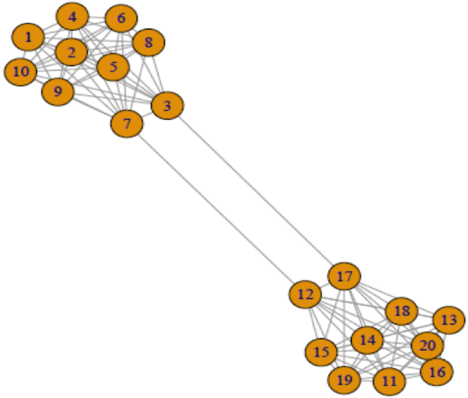
λ_1 -3.55E-15	λ_2 -3.55E-15	λ_3 1.00E+01
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
-0.3162278	0	0
-0.3162278	0	-0.07607884
-0.3162278	0	0.08239587
-0.3162278	0	0.10999263
-0.3162278	0	0.23285916
-0.3162278	0	0.42651362
-0.3162278	0	0.1775714
-0.3162278	0	-0.83645416
-0.3162278	0	-0.0826799
-0.3162278	0	-0.03411978
0	-0.3162278	0
0	-0.3162278	0
0	-0.3162278	0
0	-0.3162278	0
0	-0.3162278	0
0	-0.3162278	0
0	-0.3162278	0
0	-0.3162278	0
0	-0.3162278	0
0	-0.3162278	0

Relation of Graph Laplacian with Graph partition



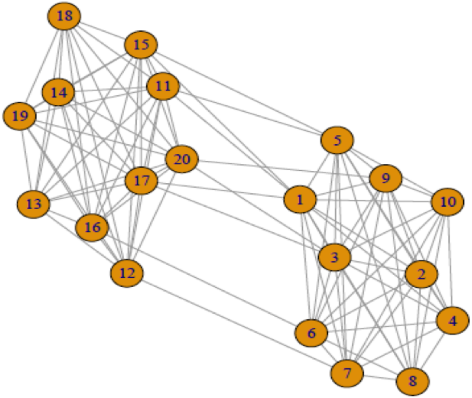
λ_1	λ_2	λ_3
1.24E-14	1.69E-01	1.00E+01
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
0.2236068	-0.2271498	0
0.2236068	-0.2271498	-0.187336735
0.2236068	-0.1887505	0.180331703
0.2236068	-0.2271498	0.026000794
0.2236068	-0.2271498	0.240657356
0.2236068	-0.2271498	-0.081758805
0.2236068	-0.2271498	0.109554976
0.2236068	-0.2271498	0.481534166
0.2236068	-0.2271498	-0.764973527
0.2236068	-0.2271498	-0.004009928
0.2236068	0.2271498	-0.047287728
0.2236068	0.2271498	-0.014797013
0.2236068	0.2271498	-0.042955103
0.2236068	0.2271498	-0.004581612
0.2236068	0.2271498	-0.008986301
0.2236068	0.2271498	-0.004581612
0.2236068	0.1887505	0.180331703
0.2236068	0.2271498	-0.019047445
0.2236068	0.2271498	-0.019047445
0.2236068	0.2271498	-0.019047445

Relation of Graph Laplacian with Graph partition



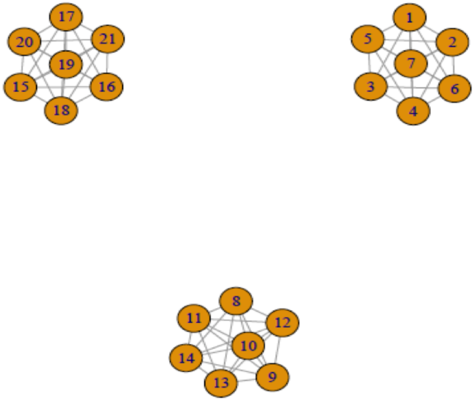
λ_1	λ_2	λ_3
-1.78E-15	3.43E-01	1.00E+01
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
0.2236068	-0.2309699	0
0.2236068	-0.2309699	-0.257554654
0.2236068	-0.1913417	0.417497079
0.2236068	-0.2309699	-0.196918966
0.2236068	-0.2309699	0.229066071
0.2236068	-0.2309699	0.339796713
0.2236068	-0.1913417	-0.207932922
0.2236068	-0.2309699	0.252674051
0.2236068	-0.2309699	-0.450305694
0.2236068	-0.2309699	-0.126321678
0.2236068	0.2309699	-0.052770238
0.2236068	0.1913417	-0.207932922
0.2236068	0.2309699	-0.029543845
0.2236068	0.2309699	-0.010168603
0.2236068	0.2309699	-0.005308371
0.2236068	0.2309699	-0.005308371
0.2236068	0.1913417	0.417497079
0.2236068	0.2309699	-0.052770238
0.2236068	0.2309699	-0.001792058
0.2236068	0.2309699	-0.051902431

Relation of Graph Laplacian with Graph partition



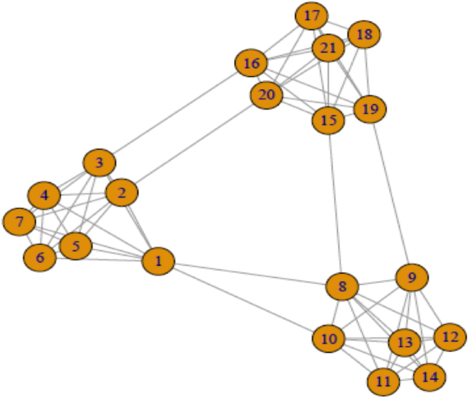
λ_1 1.42E-14	λ_2 1.66E+00	λ_3 1.00E+01
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
-0.2236068	-0.1460852	-0.08215129
-0.2236068	-0.2639678	-0.1809219
-0.2236068	-0.1784494	-0.08215129
-0.2236068	-0.2639678	-0.16323835
-0.2236068	-0.1777733	-0.08215129
-0.2236068	-0.2129005	-0.19414333
-0.2236068	-0.2129005	0.4683991
-0.2236068	-0.2639678	0.5743843
-0.2236068	-0.2169969	-0.08215129
-0.2236068	-0.2639678	-0.17587465
-0.2236068	0.1815737	-0.08215129
-0.2236068	0.2129005	0.4683991
-0.2236068	0.2639678	0.10115772
-0.2236068	0.2639678	-0.03308852
-0.2236068	0.1815737	-0.08215129
-0.2236068	0.2129005	-0.19414333
-0.2236068	0.1815083	-0.08215129
-0.2236068	0.2639678	0.0026285
-0.2236068	0.2639678	-0.01634829
-0.2236068	0.174649	-0.08215129

Relation of Graph Laplacian with Graph partition



λ_1	λ_2	λ_3
4.44E-15	4.44E-15	4.44E-15
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
-0.3779645	0	0
-0.3779645	0	0
-0.3779645	0	0
-0.3779645	0	0
-0.3779645	0	0
-0.3779645	0	0
-0.3779645	0	0
0	-0.3779645	0
0	-0.3779645	0
0	-0.3779645	0
0	-0.3779645	0
0	-0.3779645	0
0	-0.3779645	0
0	-0.3779645	0
0	0	-0.3779645
0	0	-0.3779645
0	0	-0.3779645
0	0	-0.3779645
0	0	-0.3779645
0	0	-0.3779645
0	0	-0.3779645

Relation of Graph Laplacian with Graph partition

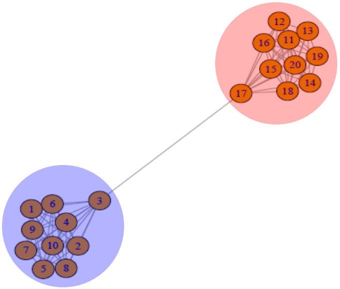


λ_1 1.78E-15	λ_2 6.39E-01	λ_3 7.34E-01
$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
-0.2182179	-0.12384629	-0.20165418
-0.2182179	-0.2208987	-0.17027215
-0.2182179	-0.2208987	-0.17027215
-0.2182179	-0.2395617	-0.23928643
-0.2182179	-0.2395617	-0.23928643
-0.2182179	-0.2395617	-0.23928643
-0.2182179	-0.2395617	-0.23928643
-0.2182179	0.22602667	-0.06051003
-0.2182179	0.27444989	-0.0407094
-0.2182179	0.2623651	-0.10699853
-0.2182179	0.32307908	-0.09189205
-0.2182179	0.32307908	-0.09189205
-0.2182179	0.32307908	-0.09189205
-0.2182179	0.32307908	-0.09189205
-0.2182179	-0.0414664	0.27727069
-0.2182179	-0.10218038	0.26216421
-0.2182179	-0.08351738	0.33117848
-0.2182179	-0.08351738	0.33117848
-0.2182179	-0.03488819	0.27999583
-0.2182179	-0.10218038	0.26216421
-0.2182179	-0.08351738	0.33117848

Spectral Clustering

Three Steps in Spectral Bi-Partition

1. Pre-processing
   - Build Graph Data Representation
   - Compute the Graph Laplacian, L
2. Decomposition
   - Find eigenpair $(\lambda_2, \mathbf{x_2})$ of L
   - Map vertices to coordinates of $\mathbf{x_2}$
   - $\mathbf{x}_2$ is 1d representation of vertices
3. Grouping
   - Split vertices based on values in $\mathbf{x}_2$
   - e.g. split by positive/negative, or
   - split at mean or median or
   - split by k-means on this 1d data

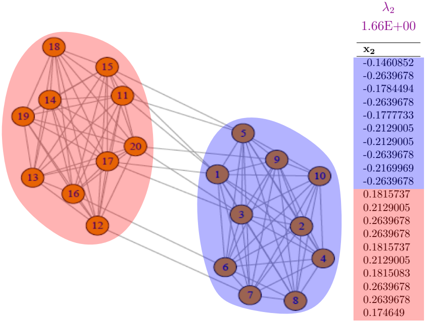


λ_2 1.69E-01
$\mathbf{x_2}$
-0.2271498
-0.2271498
-0.1887505
-0.2271498
-0.2271498
-0.2271498
-0.2271498
-0.2271498
-0.2271498
-0.2271498
0.2271498
0.2271498
0.2271498
0.2271498
0.2271498
0.2271498
0.1887505
0.2271498
0.2271498
0.2271498

Spectral Clustering into 2 clusters

Three Steps in Spectral Bi-Partition

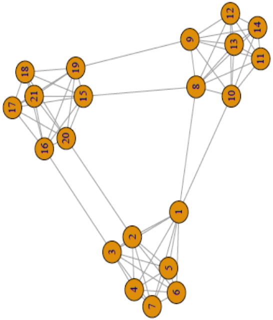
1. Pre-processing
   - Build Graph Data Representation
   - Compute the Graph Laplacian, L
2. Decomposition
   - Find eigenpair $(\lambda_2, \mathbf{x_2})$ of L
   - Map vertices to coordinates of $\mathbf{x_2}$
   - $\mathbf{x}_2$ is 1d representation of vertices
3. Grouping
   - Split vertices based on values in $\mathbf{x}_2$
   - e.g. split by positive/negative, or
   - split at mean or median or
   - split by k-means on this 1d data



Spectral Clustering into k clusters

Three Steps in Spectral Partitioning into k clusters

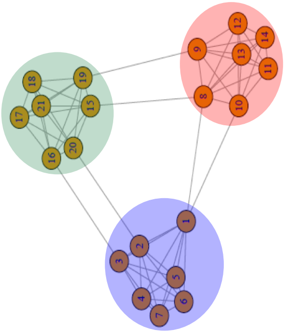
1. Pre-processing
   - Build Graph Data Representation
   - Compute the Graph Laplacian, L
2. Decomposition
   - Find k eigenvectors $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}$ of L
   - Map v_i to $\left[\mathbf{x_1}(i)\mathbf{x_2}(i)\ldots\mathbf{x_k}(i)\right]$
   - $\left[\mathbf{x_2}(i)\ \ldots\ \mathbf{x_k}(i)\right]$ is kd representation of v_i
3. Grouping
   - Cluster the vectors in $\mathbb{R}^k$
     - e.g. using k-means algorithm



λ_2 6.39E-01	λ_3 7.34E-01
$\mathbf{x_2}$	$\mathbf{x_3}$
-0.12384629	-0.20165418
-0.2208987	-0.17027215
-0.2208987	-0.17027215
-0.2395617	-0.23928643
-0.2395617	-0.23928643
-0.2395617	-0.23928643
-0.2395617	-0.23928643
0.22602667	-0.06051003
0.27444989	-0.0407094
0.2623651	-0.10699853
0.32307908	-0.09189205
0.32307908	-0.09189205
0.32307908	-0.09189205
0.32307908	-0.09189205
-0.0414664	0.27727069
-0.10218038	0.26216421
-0.08351738	0.33117848
-0.08351738	0.33117848
-0.03488819	0.27999583
-0.10218038	0.26216421
-0.08351738	0.33117848

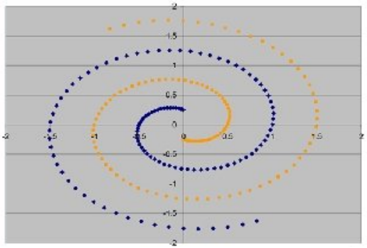
Three Steps in Spectral Partitioning into k clusters

1. Pre-processing
   - Build Graph Data Representation
   - Compute the Graph Laplacian, L
2. Decomposition
   - Find k eigenvectors $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}$ of L
   - Map v_i to $[\mathbf{x_1}(i)\mathbf{x_2}(i)\ldots\mathbf{x_k}(i)]$
   - $[\mathbf{x_2}(i)\ \ldots\ \mathbf{x_k}(i)]$ is kd representation of v_i
3. Grouping
   - Cluster the vectors in $\mathbb{R}^k$
     - e.g. using k-means algorithm



λ_2 6.39E-01	λ_3 7.34E-01
$\mathbf{x_2}$	$\mathbf{x_3}$
-0.12384629	-0.20165418
-0.2208987	-0.17027215
-0.2208987	-0.17027215
-0.2395617	-0.23928643
-0.2395617	-0.23928643
-0.2395617	-0.23928643
-0.2395617	-0.23928643
0.22602667	-0.06051003
0.27444989	-0.0407094
0.2623651	-0.10699853
0.32307908	-0.09189205
0.32307908	-0.09189205
0.32307908	-0.09189205
0.32307908	-0.09189205
-0.0414664	0.27727069
-0.10218038	0.26216421
-0.08351738	0.33117848
-0.08351738	0.33117848
-0.03488819	0.27999583
-0.10218038	0.26216421
-0.08351738	0.33117848

Naive Spectral Clustering



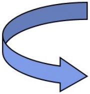
Dataset exhibits complex cluster shapes

⇒ **K-means** performs very poorly in this space due bias toward dense spherical clusters.

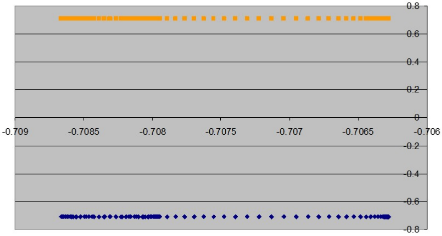
➡ Relationship vs. Geometry Distance

source: J. Fan, UNC

source: J. Fan, UNC



In the embedded space given by two leading eigenvectors, clusters are trivial to separate.



Normalized Graph Laplacians

We saw that spectral properties of Laplacian $L = D - A$ coincides with Ratio-Cut objectives

We define a Laplacian with similar spectral properties as L and coinciding with the Normalized-Cut

Symmetric Normalized Laplacian

$$L_{sym} = D^{-1/2} L D^{-1/2} = \mathbb{I} - D^{-1/2} A D^{-1/2}$$

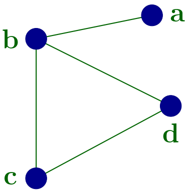
Assumes graph has no isolated vertices, all degree ≥ 1

Normalized Graph Laplacians

Symmetric Normalized Laplacian

$$L_{sym} = D^{-1/2} L D^{-1/2} = \mathbb{I} - D^{-1/2} A D^{-1/2}$$

Assumes graph has no isolated vertices, all degree ≥ 1



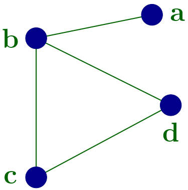
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Normalized Graph Laplacians

Symmetric Normalized Laplacian

$$L_{sym} = D^{-1/2} L D^{-1/2} = \mathbb{I} - D^{-1/2} A D^{-1/2}$$

Assumes graph has no isolated vertices, all degree ≥ 1



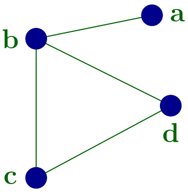
$$\overset{\mathbf{D}^{-1/2}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{3} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1/\sqrt{2} \end{bmatrix}} \overset{\mathbf{A}}{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}} \overset{\mathbf{D}^{-1/2}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{3} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1/\sqrt{2} \end{bmatrix}}$$

Normalized Graph Laplacians

Symmetric Normalized Laplacian

$$L_{sym} = D^{-1/2} L D^{-1/2} = \mathbb{I} - D^{-1/2} A D^{-1/2}$$

Assumes graph has no isolated vertices, all degree ≥ 1



$$\mathbf{L_{sym}} = \mathbb{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

$$\begin{bmatrix} 1 & -1/\sqrt{3} & 0 & 0 \\ -1/\sqrt{3} & 1 & -1/\sqrt{6} & -1/\sqrt{6} \\ 0 & -1/\sqrt{6} & 1 & -1/2 \\ 0 & -1/\sqrt{6} & -1/2 & 1 \end{bmatrix}$$

Normalized Graph Laplacians

The Algorithm based on Unnormalized Laplaian is due to Shi and Malick

The algorithm based on normalized Laplacian is due to Ng, Jordan, and Weiss

All spectral algorithms work for weighted graphs too, with weighted degree of a vertex defined as sum of weights of all edges incident on it