# SPECTRAL CLUSTERING

Let $x_1, \ldots, x_n$ be objects we wish to cluster.

Spectral clustering refers to the following:

(A)

- form a _____ _____
- construct an $n \times n$ matrix, called the _____ _____, from this graph.
- infer the clusters from the _____ or _____ decomposition of this matrix.

The ingredients that determine a sim. graph are

- 
- 

Given these two ingredients, the weighted adjacency matrix is

$$w_{ij} = \begin{cases} s_{ij} & \text{if } x_i, x_j \text{ adjacent} \\ 0 & \text{otherwise} \end{cases}$$

Examples

▷ $\epsilon$-neighborhood graph

- $x_i, x_j$ adjacent $\iff \|x_i - x_j\| \leq \epsilon$
- $s_{ij} = 1$

$\Rightarrow$ locality captured entirely by graph

▷ complete (fully connected) graph

- all $x_i, x_j$ adjacent
- $s_{ij} = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}$, $\sigma^2 > 0$

$\Rightarrow$ locality captured entirely by similarities

▷ $l$-nearest neighbor graph

- $x_i, x_j$ adjacent $\Longleftrightarrow$ $x_i$ is an $l$-nearest neighbor of $x_j$ or vice-versa.
- $s_{ij} = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}$

▷ mutual $l$-nearest neighbor graph

- $x_i, x_j$ adjacent $\iff$ $x_i$ is an $l$-nearest neighbor of $x_j$ and vice versa

Unfortunately, these similarity graphs have free parameters $(\varepsilon, \sigma, l)$ that can be difficult to tune.

The reason for constructing a similarity graph is that it reduces the problem of clustering to _____ _____ :

(c) Find a partition of the graph such that

- edges between clusters have _____ weights
- edges within clusters have _____ weights

# Graph Laplacians

## Definitions

- The degree of a vertex $x_i$ is

(D) $$d_i :=$$

- The degree matrix is the diagonal matrix

$$D :=$$

- The unnormalized graph Laplacian is

$$L :=$$

## Note

$L$ is independent of the self-similarity weights $w_{ii}$, because

$$L_{ii}$$

**Proposition** (Properties of $L$)

1) For every $f \in \mathbb{R}^n$

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

2) $L$ is symmetric and positive semi-definite

3) The smallest eigenvalue of $L$ is $0$

with corresponding eigenvector $\underline{1} = [1\ 1 \cdots 1]^T$

Hence $L$ has $n$ non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

**Proof**

1) $f^T L f = f^T D f - f^T W f$

$$= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j$$

$$= \frac{1}{2} \left( \sum_i d_i f_i^2 - 2 \sum_{i,j} w_{ij} f_i f_j + \sum_j d_j f_j^2 \right)$$

$$= \frac{1}{2} \sum_i \sum_j w_{ij} \left( f_i^2 - 2 f_i f_j + f_j^2 \right)$$

$$= \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

2) Symmetry follows from symmetry of $W$

PSD follows from 1) because, for any $f \in \mathbb{R}^n$,

(E)

$$f^T L f =$$

$$\geq$$

3)

$$L\underline{1} = \begin{bmatrix} d_1 & & 0 \\ & d_2 & \\ 0 & & \ddots \\ & & & d_n \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} - \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$=$$

$$=$$

The unnormalized graph Laplacian encodes many properties of the graph. The following is one such property that is relevant for clustering.

**Notation** Let $A \subseteq \{x_1, \ldots, x_n\}$ be a cluster.

Define the indicator vector

$$\underline{1}_A = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \in \mathbb{R}^n$$

where $f_i = 1$ if $x_i \in A$ and $f_i = 0$ if $x_i \notin A$.

**Proposition** Suppose the graph has connected components $A_1, \ldots, A_K$. Then the _____ of $L$ has dimension ____ and is spanned by

(F)

Recall | The nullspace of $L$ is

$$N(L) := \{ f : Lf = 0 \}$$

It suffices to show:

- $\underline{1}_{A_k} \in N(L)$ for each $k = 1, \ldots, K$
- If $f \in N(L)$, then

$$f =$$

for some $\alpha_1, \ldots, \alpha_K \in \mathbb{R}$.

Since $\underline{1}_{A_1}, \ldots, \underline{1}_{A_K}$ are linearly independent, it follows that

$$\dim(N(L)) =$$

Proof

- $K=1$: we have already seen $\underline{1} \in N(L)$. We need to show that if $Lf = 0$, then

$$f = \alpha \cdot \underline{1}$$

for some $\alpha \in \mathbb{R}$.

If $Lf = 0$, then

$$0 = f^T L f = \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

If $x_i, x_j$ are adjacent, then

$$w_{ij} > 0 \implies f_i = f_j$$

Since $K=1$, any two points can be connected by a path

$$\implies f_i = \text{constant}$$

- $K > 1$: Suppose WLOG that the data are ordered according to connected component. Then $W$ and $L$ are block diagonal

$$L = \begin{bmatrix} L_1 & & 0 \\ & L_2 & \\ 0 & & \ddots \\ & & & L_K \end{bmatrix}$$

Note each $L_k$ is a graph Laplacian on $A_k$. Then by the previous case we know

- $L \mathbb{1}_{A_k} = 0$ for each $k$
- If $Lf = 0$, then $f_i$ is constant on each $A_k$

$$\Rightarrow f = \sum_{k=1}^{K} \alpha_k \mathbb{1}_{A_k}$$

# Spectral Clustering

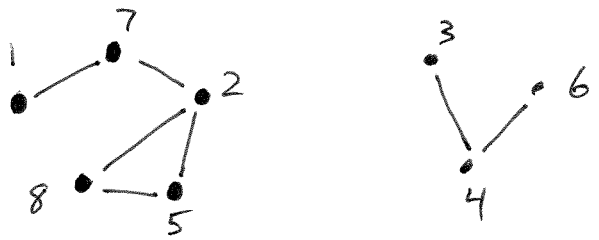
How can we use this result to devise a clustering algorithm?

## Ideal case

In the ideal case where there are $K$ connected components and $K$ is known, we could proceed as follows:

- compute $L$
- compute a basis $u_1, \dots, u_k$ for $N(L)$.
  - $\hookrightarrow$ span $\{u_1, \dots, u_k\}$ = span $\{\mathbb{1}_{A_1}, \dots, \mathbb{1}_{A_k}\}$
- define
$$y_i = (u_{i1}, \dots, u_{ik})$$
- if $x_i, x_j$ in same component then

Example




$$L = \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix} \qquad \underline{1}_{A_1} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}, \quad \underline{1}_{A_2} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

Use spectral decomposition of $L$ to find vectors $u_1, u_2$ that span $N(L)$.

Write
$$u_1 = \alpha_1 \underline{1}_{A_1} + \beta_1 \underline{1}_{A_2}$$
$$u_2 = \alpha_2 \underline{1}_{A_1} + \beta_2 \underline{1}_{A_2}$$

Set
$$y_i = (u_{1i} \quad u_{2i})$$

Then
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} & \\ & \\ & \\ & \\ & \\ & \\ & \\ & \end{bmatrix} \Rightarrow$$

determines connected components

In conclusion, we can use the spectral decomposition of the graph Laplacian $L$ to determine the connected components.

However, there are two problems with this approach

1) There are simpler ways of finding the connected components of a graph

2) It is very difficult to construct a similarity graph such that its connected components are clusters.

Non ideal case

More realistically, the components of our similarity graph will not coincide with clusters.

In practice, we should choose the similarity graph such that

$$\text{number of connected components} \ll \text{desired number of clusters}$$

Then there will be edges connecting points in different clusters, but the weights

(J) between clusters should be ——— ——— the weights within clusters.

Then, if points are indexed according to cluster, we have

$$L = \begin{bmatrix} \boxed{\text{large}} & \text{small} & \\ \text{small} & \boxed{\text{large}} & \\ & & \boxed{\text{large}} \end{bmatrix} = \begin{bmatrix} \boxed{\text{large}} & 0 & \\ 0 & \boxed{\text{large}} & \\ & & \boxed{\text{large}} \end{bmatrix} + \begin{bmatrix} & & \\ & \text{small} & \\ & & \end{bmatrix}$$

$\uparrow$ $\uparrow$

# Perturbation Theory

Perturbation theory establishes results that show: if we perturb a matrix by another matrix with small entries, then the eigenvectors and eigenvalues of the matrix are perturbed by a correspondingly small amount.

Recall the nullspace of $L$ is also the set of eigenvectors with eigenvalue 0. We can use the eigenvectors of $L$ with the smallest $K$ eigenvalues as an approximation to the nullspace of an idealized $L$ based on the true clusters.

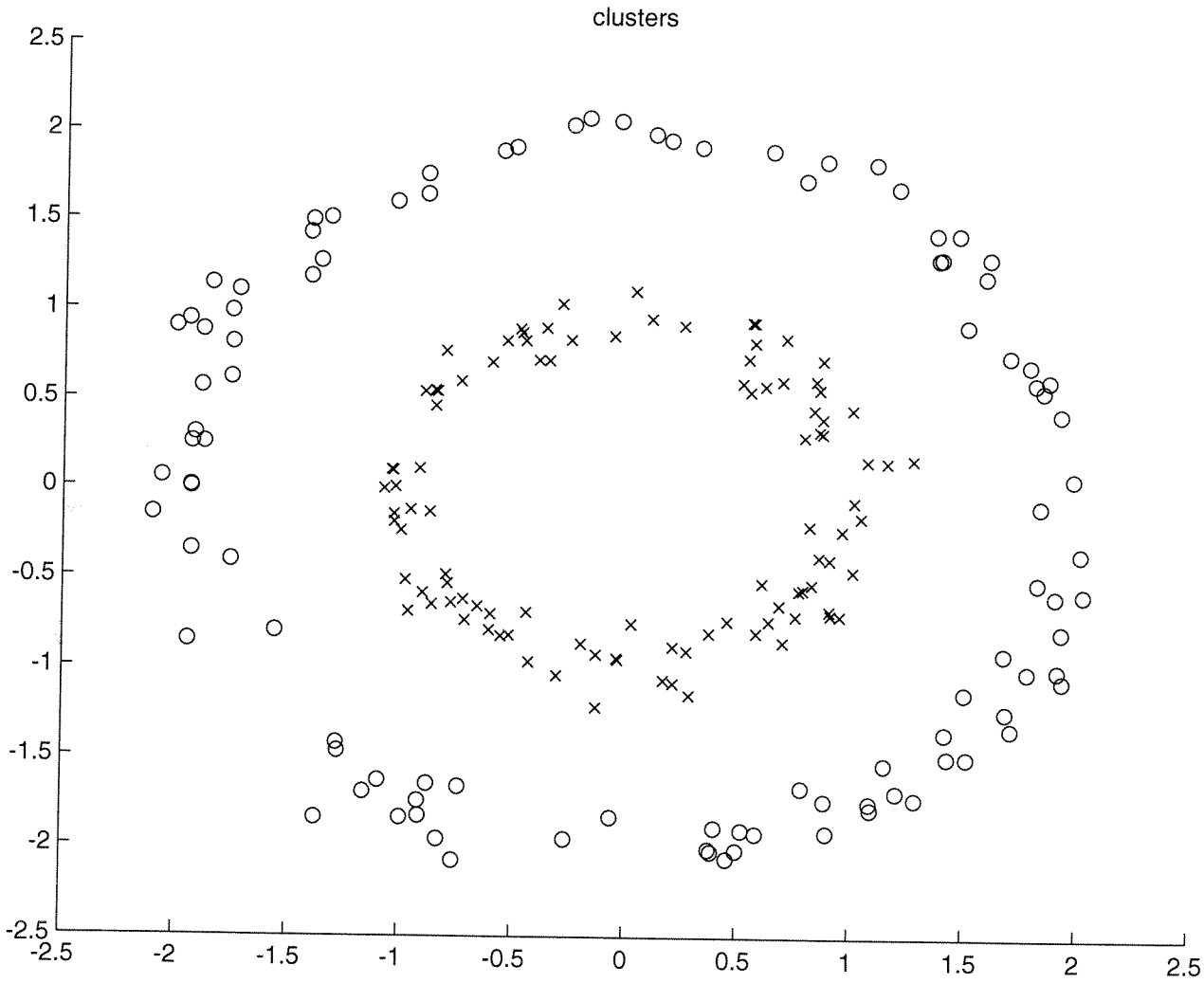
# SPECTRAL CLUSTERING

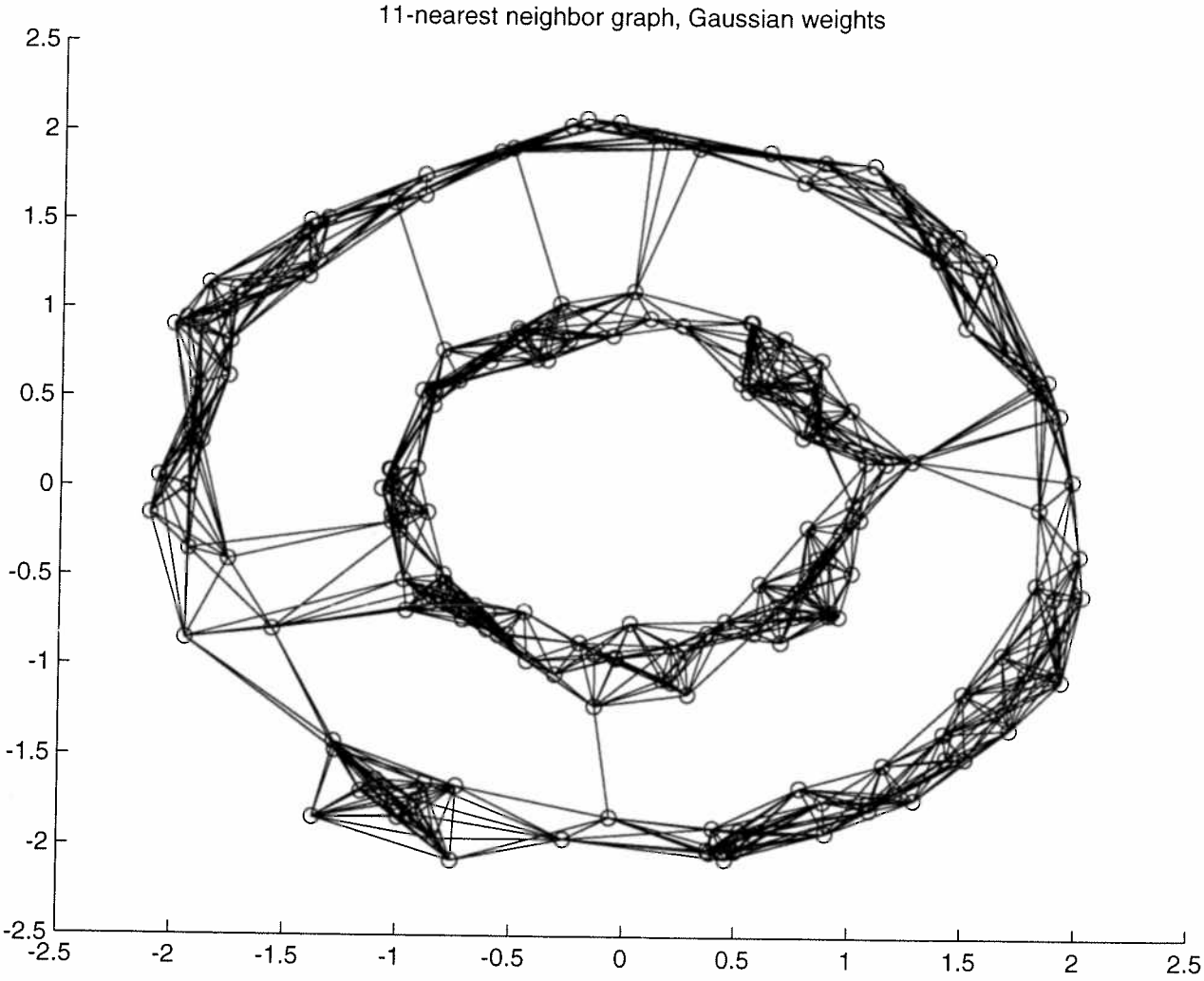
- Construct similarity graph
- Form graph Laplacian $L \in \mathbb{R}^{n \times n}$
- Determine the $K$ smallest eigenvalues of $L$,

  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_K$, and corresponding

  eigenvectors $u_1, \dots, u_K \in \mathbb{R}^n$
- Define $y_i = (u_{1i}, u_{2i}, \dots, u_{Ki})$, $i = 1, \dots, n$
- Cluster $\{y_i\}_{i=1}^n$ using $K$-means clustering

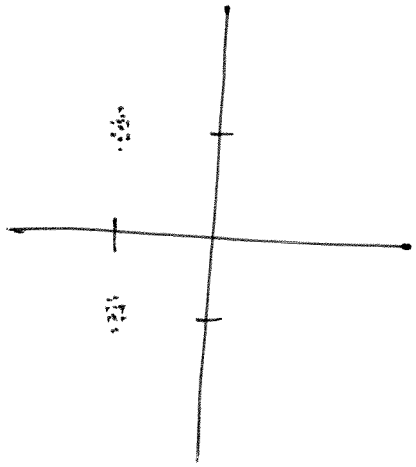
  and assign $\{x_i\}_{i=1}^n$ to corresponding clusters


clusters
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5


11-nearest neighbor graph, Gaussian weights
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

| | |
|---|---|
| -0.070711 | -0.07071 |
| -0.070711 | -0.07071 |
| -0.070711 | -0.070709 |
| -0.070711 | -0.07071 |
| -0.070711 | -0.070708 |
| -0.070711 | -0.070708 |
| -0.070711 | -0.070709 |
| -0.070711 | 0.071153 |
| -0.070711 | 0.071682 |
| -0.070711 | 0.070093 |
| -0.070711 | 0.071059 |
| -0.070711 | -0.070708 |
| -0.070711 | -0.070709 |
| -0.070711 | 0.070098 |
| -0.070711 | -0.07071 |
| -0.070711 | 0.071489 |
| -0.070711 | -0.070709 |
| -0.070711 | -0.07071 |
| -0.070711 | 0.070098 |
| -0.070711 | -0.070708 |
| -0.070711 | 0.070098 |
| -0.070711 | -0.07071 |
| -0.070711 | 0.071682 |
| -0.070711 | -0.070709 |
| -0.070711 | -0.07071 |
| -0.070711 | 0.070098 |
| -0.070711 | -0.07071 |
| -0.070711 | 0.071489 |
| -0.070711 | -0.07071 |
| -0.070711 | -0.070708 |
| -0.070711 | -0.070708 |
| -0.070711 | -0.07071 |
| -0.070711 | -0.070708 |
| -0.070711 | 0.071682 |
| -0.070711 | 0.070098 |
| -0.070711 | 0.070095 |
| -0.070711 | -0.07071 |
| -0.070711 | -0.070708 |
| -0.070711 | -0.07071 |
| -0.070711 | 0.070093 |
| -0.070711 | 0.071682 |
| -0.070711 | -0.070709 |
| -0.070711 | 0.070096 |
| -0.070711 | 0.071153 |
| -0.070711 | 0.071153 |
| -0.070711 | 0.070099 |
| -0.070711 | -0.07071 |
| -0.070711 | 0.070093 |
| -0.070711 | 0.070093 |
| -0.070711 | 0.070098 |

a few randomly selected y_i's



# Normalized Spectral Clustering

(K) Define the _____ _____ _____

$$\tilde{L} :=$$

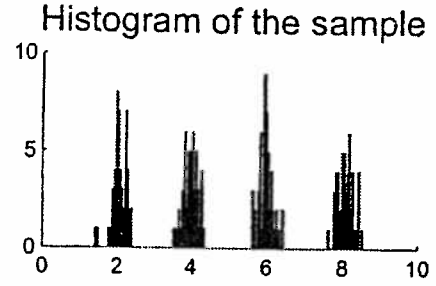
**Proposition 1**

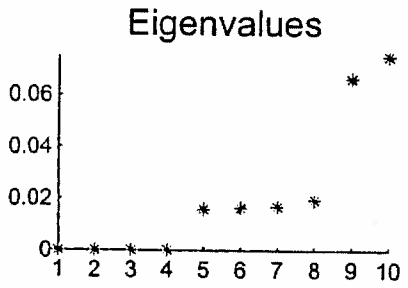
1) $\tilde{L}$ is positive semidefinite with real, nonnegative eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

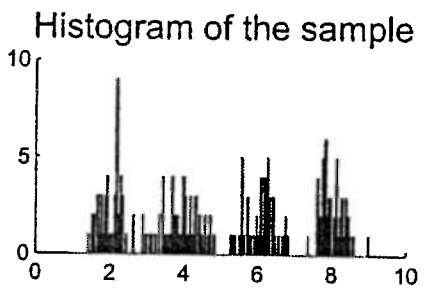
2) 0 is an eigenvalue of $\tilde{L}$ with corresponding eigenvector $\underline{1} = [1\ 1 \cdots 1]^T$.

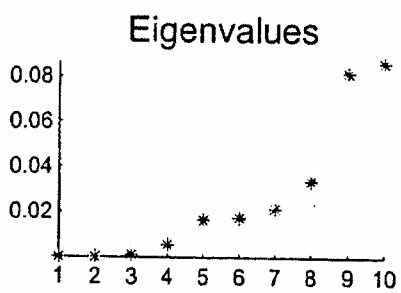
3) Suppose the graph has connected components $A_1, \ldots, A_K$. Then the nullspace of $\tilde{L}$ has dimension $K$ and is spanned by $\underline{1}_{A_1}, \ldots, \underline{1}_{A_K}$

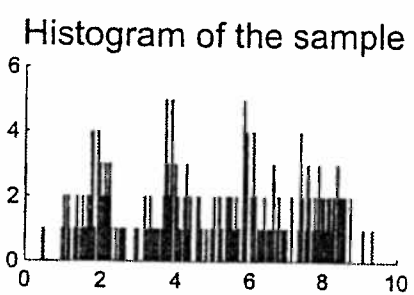
$\Longrightarrow$ we can substitute $\tilde{L}$ for $L$ and get another spectral clustering algorithm.

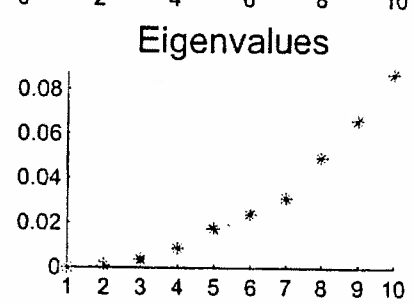



The first large gap in the spectrum can be used to infer the number of clusters automatically.

Results shown are for <u>normalized</u> spectral clustering.

# Graph Cuts

Spectral clustering attempts to find a partition $A_1, \ldots, A_K$ of the similarity graph such that

- $w_{ij}$ large if $x_i, x_j$ in _____ cluster
- $w_{ij}$ small if $x_i, x_j$ in _____ clusters

using properties of graph Laplacians.

It is also possible to approach this goal directly.

The _____ problem is to minimize

$$\text{cut}(A_1, \ldots, A_K) :=$$

with respect to $A_1, \ldots, A_K$, where

$$W(A, B) :=$$

The mincut problem is efficiently solvable when $K=2$. Unfortunately, it tends to lead to small (and often singleton) clusters.

Therefore researchers have examined modified criteria that favor larger clusters:

(M)

- <u>Ratio Cut</u> (Hagen and Kahng, 1992)

$$\text{RatioCut}(A_1, \ldots, A_K) :=$$

where $|A| :=$

- <u>Normalized Cut</u> (Shi and Malik, 2000)

$$\text{Ncut}(A_1, \ldots, A_K) :=$$

$$=$$

where $\text{vol}(A_i) :=$

Unfortunately, minimizing these criteria is NP hard.

Remarkably, however, spectral clustering may be used to solve _____ of these problems

(N)

In particular,

- unnormalized spectral clustering $\Longrightarrow$
- normalized spectral clustering $\Longrightarrow$

## Approximating Ratio Cut for K=2

We wish to solve

$$\min_A \text{Ratio Cut}(A, \bar{A}) = \min_A \left[ \frac{\text{cut}(A, \bar{A})}{|A|} + \frac{\text{cut}(A, \bar{A})}{|\bar{A}|} \right]$$

Given a subset $A \subseteq \{1, 2, \ldots, n\}$, define $f_A := (f_{A1}, \ldots, f_{An})^T$ by

$$f_{Ai} := \begin{cases} \sqrt{|\bar{A}|/|A|} & \text{if } i \in A \\ -\sqrt{|A|/|\bar{A}|} & \text{if } i \notin A. \end{cases}$$

**Claim** $f_A^T L f_A = n \, \text{Ratio Cut}(A, \bar{A})$

Proof

$$f_A^T L f_A = \frac{1}{2} \sum_{ij} w_{ij} (f_{Ai} - f_{Aj})^2$$

$$= \frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} \left( \sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 + \frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} \left( -\sqrt{\frac{|A|}{|\bar{A}|}} - \sqrt{\frac{|\bar{A}|}{|A|}} \right)$$

$$= \frac{1}{2} cut(A, \bar{A}) \left( \frac{|\bar{A}|}{|A|} + \frac{|A|}{|\bar{A}|} + 2 \right) + \frac{1}{2} cut(A, \bar{A}) \left( \frac{|A|}{|\bar{A}|} + \frac{|\bar{A}|}{|A|} + 2 \right)$$

$$= cut(A, \bar{A}) \left( \frac{|\bar{A}|}{|A|} + \frac{|A|}{|\bar{A}|} + 2 \right)$$

$$= cut(A, \bar{A}) \left( \frac{|A| + |\bar{A}|}{|A|} + \frac{|\bar{A}| + |A|}{|\bar{A}|} \right)$$

$$= \underbrace{(|A| + |\bar{A}|)}_{n} \cdot \underbrace{\left[ \frac{cut(A, \bar{A})}{|A|} + \frac{cut(\bar{A}, A)}{|\bar{A}|} \right]}_{\text{Ratio Cut}(A, \bar{A})}$$

Furthermore, $f_A$ satisfies

(0) • $\underline{1}^T f_A = \sum_{i=1}^{n} f_{Ai}$

$=$

$=$

• $\| f_A \|^2 = \sum_{i=1}^{n} f_{Ai}^2$

$=$

$=$

Therefore, the RatioCut problem can be written as the following ——— optimization problem:

$$\min_{A} \quad f_A^T L f_A$$

$$\text{s.t.} \quad \underline{1}^T f_A =$$

$$\| f_A \| =$$

If we allow $f \in \mathbb{R}^n$ we have the following relaxation:

$$\min_{f \in \mathbb{R}^n} \; f^T L f$$

$$\text{s.t.} \quad \mathbb{1}^T f = 0$$

$$\|f\| = \sqrt{n}$$

← { No longer require $f = f_A$ for some $A$

The solution is

(P) $f =$

To recover a solution to the original discrete problem, we can use K-means, $K = 2$, to cluster the vectors

$$y_i := (1 \quad f_i)$$

Therefore, the approximate solution is given by unnormalized spectral clustering.

A similar analysis applies to $K > 2$ and to $N_{cut}$.

## Final comments

- The $l$-nn graph with Gaussian kernel similarity is most common, although the choice of similarity graph is largely an art.
- Another normalized graph Laplacian is

$$\tilde{L} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$$

(Ng, Jordan, & Weiss, 2002). It can also be used for spectral clustering but the algorithm requires some modification.

- Which method is preferred? $\tilde{L}$ is recommended by

U. von Luxburg, "A Tutorial on Spectral Clustering," 2007.

- b-matching is an interesting alternative to nearest neighbor graphs: it ensures that each node has the same number of incident edges (unweighted degree). See Jebara et al, ICML 2009

- graph Laplacians are also used in

(@) _____ _____ learning.

For example, suppose we are in a regression setting with data

$$(x_i, y_i)_{i=1}^{m}$$

$$(x_i)_{i=m+1}^{n}$$

To make sure the estimated function doesn't "wiggle" too much, we could minimize

$$\frac{1}{m}\sum_{i=1}^{m}\left(y_i - f(x_i)\right)^2 + \frac{\lambda}{2}\underbrace{\sum_{i,j=1}^{n} w_{ij}\left(f(x_i) - f(x_j)\right)^2}_{f^T L f}$$

Key

A. similarity graph, graph Laplacian, eigenvalue/spectral

B. a graph, a similarity matrix $S = [s_{ij}]_{ij=1}^{n}$

C. graph partitioning, low, high

D. $d_i = \sum_{j=1}^{n} w_{ij}$, $D = \begin{bmatrix} d_1 & & 0 \\ & d_2 & \\ & & \ddots \\ 0 & & d_n \end{bmatrix}$, $L = D - W$

$$L_{ii} = d_i - w_{ii} = \sum_{j \neq i} w_{ij}$$

E. $f^T L f = \sum w_{ij} (f_i - f_j)^2 \geq 0$

$$L\underline{1} = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} - \begin{bmatrix} \sum w_{1j} \\ \vdots \\ \sum w_{nj} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = 0 \cdot \underline{1}$$

L is PSD

F. nullspace (0-eigenspace); K; $\underline{1}_{A_1}, \ldots, \underline{1}_{A_K}$

G. $f = \sum_{k=1}^{K} \alpha_k \underline{1}_{A_k}$, $\dim(N(L)) = K$ = multiplicity of 0 as an eigenvalue

H. $y_i = y_j$

I. $\underline{1}_{A_1} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$, $\underline{1}_{A_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$,

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \\ \beta_1 & \beta_2 \\ \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \\ \alpha_1 & \alpha_2 \\ \alpha_1 & \alpha_2 \end{bmatrix}$$

J. less than, ideal case, noise/perturbation

K. normalized graph Laplacian $\tilde{L} = D^{-1}L = I - D^{-1}W$

L. same, different, mincut, $cut(A_1, \dots, A_k) = \frac{1}{2} \sum_{k=1}^{K} W(A_k, \bar{A}_k)$, $W(A,B) = \sum_{i \in A} \sum_{j \in B} w_{ij}$

M. $\text{Ratio Cut}(A_1, \dots, A_k) = \frac{1}{2} \sum_{k=1}^{K} \frac{W(A_k, \bar{A}_k)}{|A_k|} = \sum_{k=1}^{K} \frac{cut(A_k, \bar{A}_k)}{|A_k|}$

$|A_k|$ = # of nodes in $A$

$Ncut(A_1, \dots, A_k) = \frac{1}{2} \sum_{k=1}^{K} \frac{W(A_k, \bar{A}_k)}{vol(A_k)} = \sum_{k=1}^{K} \frac{cut(A_k, \bar{A}_k)}{vol(A_k)}$

$vol(A_k) = \sum_{i \in A} \sum_{j \in A} w_{ij}$

N. relaxations, Ratio Cut, Ncut

O. $\underline{1}^T f_A = \sum_{i \in A} \sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i \in \bar{A}} \sqrt{\frac{|A|}{|\bar{A}|}} = \sqrt{|A| \cdot |\bar{A}|} - \sqrt{|A| \cdot |\bar{A}|} = 0$

$\|f_A\|^2 = \sum_{i \in A} \frac{|\bar{A}|}{|A|} + \sum_{i \in \bar{A}} \frac{|A|}{|\bar{A}|} = |\bar{A}| + |A| = n$

discrete, $0$, $\sqrt{n}$

P. eigenvector of $L$ corresponding to second smallest eigenvalue

Q. semi-supervised