SPECTRAL CLUSTERING

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

Spectral clustering refers to the following:

Ⓐ

- 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

▷ <u>ε-neighborhood graph</u>

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

$\Longrightarrow$ locality captured entirely by graph

▷ <u>complete (fully connected) graph</u>

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

⟹ locality captured entirely by similarities

▷ <u>l-nearest neighbor graph</u>

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

▷ <u>mutual ℓ-nearest neighbor graph</u>

- x_i, x_j adjacent $\Longleftrightarrow$ x_i is an ℓ-nearest neighbor of x_j <u>and</u> vice versa

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

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

ⓒ 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 $\underline{0}$ with corresponding eigenvector $\underline{1} = [1 \; 1 \cdots 1]^T$

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

__Proof__

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

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

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

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

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

2) Symmetry follows from symmetry of

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

$$f^T L f =$$

Ⓔ

$$\geq$$

3)

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

$$=$$

$$=$$

4) Follows because L is

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

<u>Notation</u> | Let $A \subseteq \{x_1,\dots,x_n\}$ be a cluster.

Define the indicator vector

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

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

<u>Proposition</u> | Suppose the graph has connected

Ⓕ components $A_1,\dots,A_K$. Then the ______________

of L has dimension ______ and is

spanned by

Recall | The nullspace of L is

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

The proposition is claiming

- $\mathbb{1}_{A_k} \in N(L)$ for each $k = 1, \dots, K$

- If $f \in N(L)$, then

Ⓖ $$f =$$

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

Since $\mathbb{1}_{A_1}, \dots, \mathbb{1}_{A_K}$ 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

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

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

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

$\blacksquare$

Spectral Clustering

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

Ideal case

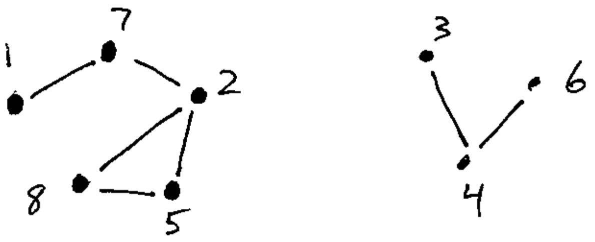
In the ideal case where there are K connected components and K is known, we could proceed as follows:

- compute L
- compute a basis $u_1, \dots, u_K$ for $N(L)$.

  $\hookrightarrow$ $\text{span}\{u_1, \dots, u_K\} = \text{span}\{\underline{1}_{A_1}, \dots, \underline{1}_{A_k}\}$
- define

  $$y_i = (u_{i1}, \dots, u_{iK})$$
- if x_i, x_j in same components then

Example



$L =$

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

Use <u>spectral</u> 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

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

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

Ⓙ between clusters should be _____ _______

the weights within clusters.

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

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

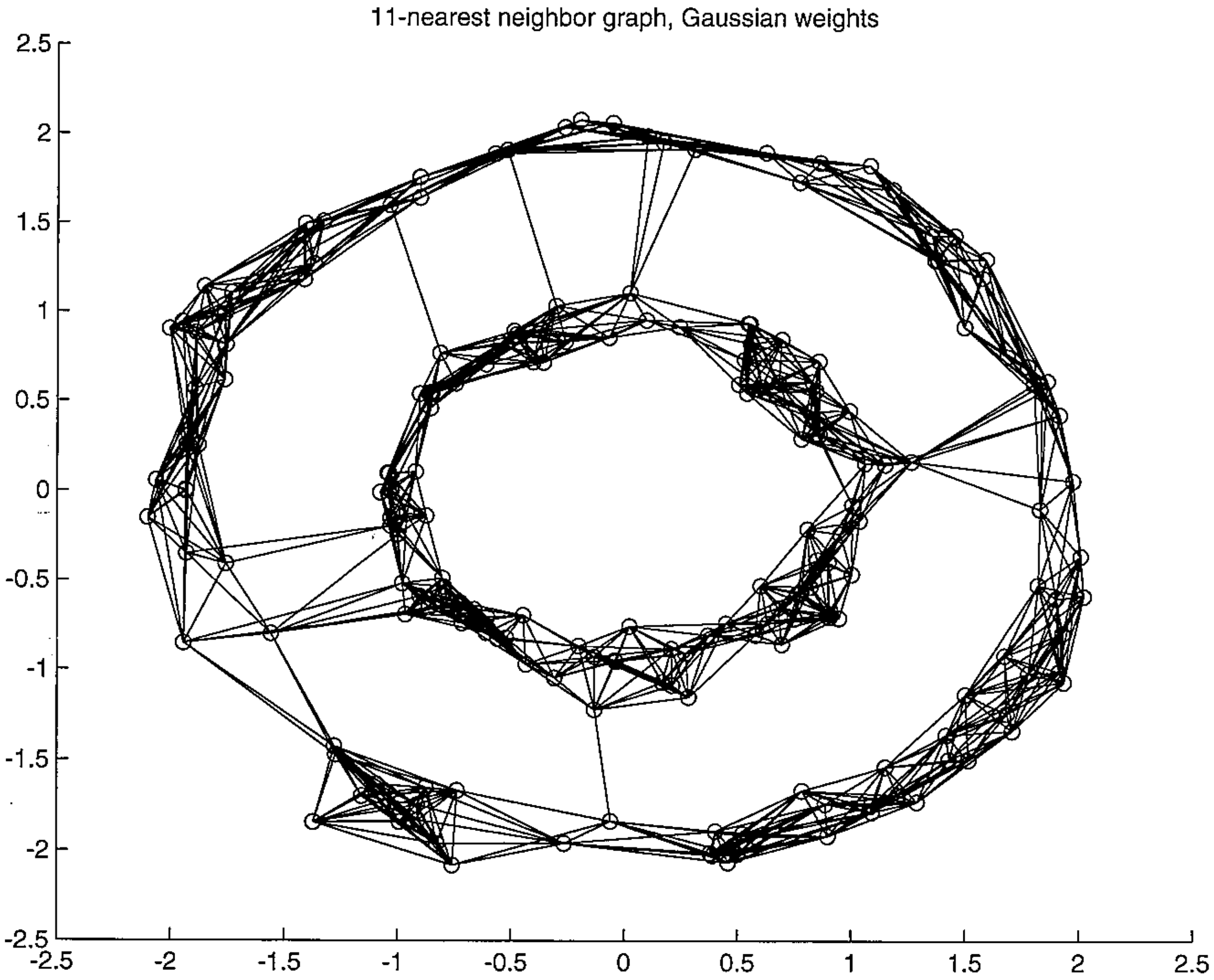
Perturbation Theory

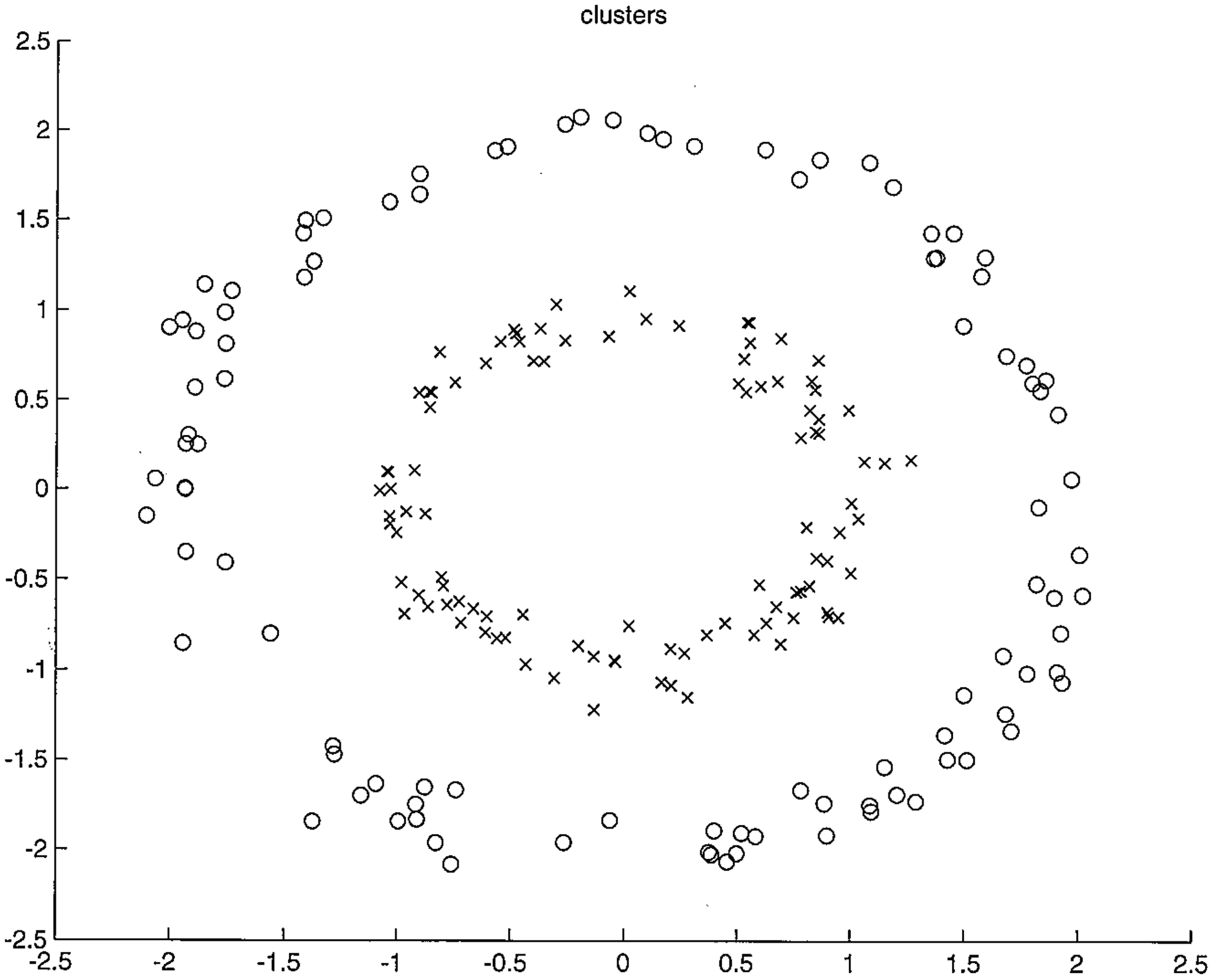
Perturbation theory establishes results that show: if we perturb a matrix by another matrix with small entries, then the eigenvectors and eigenvalues of the matrix are perturbed by a correspondingly small amount.

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

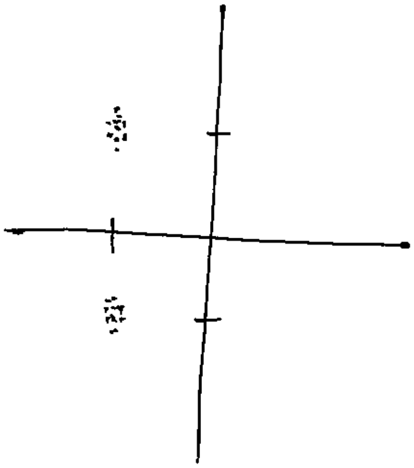
- Construct similarity graph
- Form graph Laplacian $L \in \mathbb{R}^{n \times n}$
- Determine the K smallest eigenvalues of L, $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_K$, and corresponding eigenvectors $u_1, \dots, u_K \in \mathbb{R}^n$
- Define $y_i = (u_{1i}, u_{2i}, \dots, u_{Ki})$, $i = 1, \dots, n$
- Cluster $\{y_i\}_{i=1}^n$ using K-means clustering and assign $\{x_i\}_{i=1}^n$ to corresponding clusters


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


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

-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

Ⓚ Define the ____________ ________ ___________

$$\tilde{L} :=$$

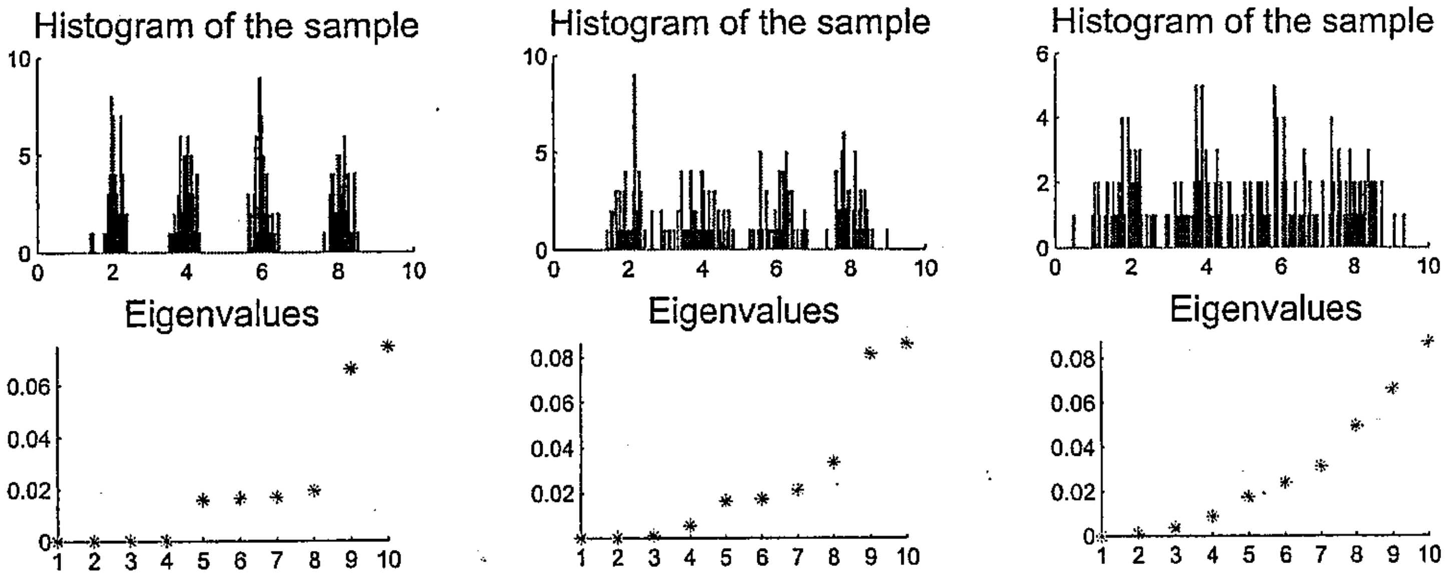
Proposition 1

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

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 $A_1, \dots, A_K$. Then the nullspace of $\tilde{L}$ has dimension $\underline{K}$ and is spanned by $\underline{1}_{A_1}, \dots, \underline{1}_{A_K}$

$\Longrightarrow$ 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 normalized spectral clustering.

Graph Cuts

Spectral clustering attempts to find a partition $A_1, \dots, A_K$ 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

$$\mathrm{cut}(A_1, \dots, A_K) :=$$

with respect to $A_1, \dots, A_K$, 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:

Ⓜ

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

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

  where $|A| :=$

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

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

  $=$

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

Unfortunately, minimizing these criteria is NP hard.

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

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 $A \subseteq \{1,2,\dots,n\}$, define $f_A := (f_{A1},\dots,f_{An})^T$ 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] $f_A^T L f_A = n \text{ Ratio Cut}(A,\bar{A})$

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

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

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

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

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

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

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:

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

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

$$\| f_A \| =$$

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

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

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

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

The solution is

Ⓟ $f =$

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 Ncut.

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" to much, we could minimize

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

Key

A. similarity graph , graph Laplacian , eigenvalue/spectral

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

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 ; $\underline{1}_{A_1}, \dots, \underline{1}_{A_K}$

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 , $\text{cut}(A_1, \dots, A_k) =$

$$\frac{1}{2} \sum_{k=1}^{K} W(A_k, \bar{A}_k) , \quad W(A,B) = \sum_{i \in A} \sum_{j \in B} w_{ij}$$

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

$|A_k| =$ # of nodes in A

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

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

N. relaxations, RatioCut, 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