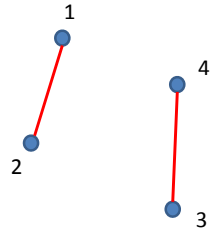
Spectral clustering

Lecture 2

Spectral clustering algorithms

A simple example

- Two ideal clusters, with two points each



$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

• Ideally permuted
• Ideal affinities

Indicator vectors

- Each cluster has an *indicator vector*, represented by a binary vector that contains ``1'' for points in the cluster and ``0'' otherwise:

$$\mathbf{c}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{c}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Indicator vectors of distinct clusters must be orthogonal!

A simple example

- Clearly, we can decompose **A** as

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$\mathbf{c}_1$ and $\mathbf{c}_2$

$$\mathbf{A} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 1 \end{pmatrix}$$

Eigensystem of **A**

- An eigenvalue decomposition of **A** gives

$$\text{normalized eigenvectors} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

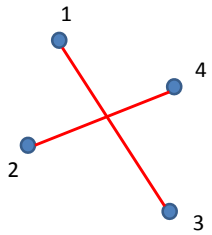
$$\text{corresponding eigenvalues} = \begin{pmatrix} 2 & 2 & 0 & 0 \end{pmatrix}$$

Initial idea

- To each cluster there is a non-zero eigenvalue in **A**
  - Number of clusters = number of non-zero eigenvalues in **A**
- To each such eigenvalue/cluster, the corresponding normalized eigenvector is a scaled version of the corresponding indicator vector

Permutations of **A**

- Two ideal cluster, with two points each



$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Eigensystem of permuted **A**

- An eigenvalue decomposition of **A** gives

$$\text{normalized eigenvectors} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\text{corresponding eigenvalues} = \begin{pmatrix} 2 & 2 & 0 & 0 \end{pmatrix}$$

Initial idea holds: permutations of the points carries over to permutations of the elements of the eigenvectors

Eigensystem of permuted **A**

In an algebraic sense:

- The goal of spectral clustering is to determine the permutation of **A** that turns it into a block diagonal form
- This is done by analyzing the eigensystem of **A**

A glitch (I)

- In this case: the non-zero eigenvalues are equal
  - Any linear combination of the first two eigenvectors is also an eigenvector of the same eigenvalue
  - Any small perturbation of **A** can make a large change in the eigenvectors
  - Eigenvectors will not correspond to the indicator vectors

A glitch (I)

$$\mathbf{A} = \begin{pmatrix} 1 & 0.99 & 0.01 & 0.02 \\ 0.99 & 1 & 0.01 & 0.03 \\ 0.01 & 0.01 & 1 & 0.98 \\ 0.02 & 0.03 & 0.98 & 1 \end{pmatrix}$$

Again ideally ordered but with some noise

Approximate numerical values

$$\text{normalized eigenvectors} = \begin{pmatrix} 0.53 & -0.46 & -0.28 & 0.65 \\ 0.54 & -0.46 & 0.27 & -0.65 \\ 0.46 & 0.54 & -0.65 & -0.27 \\ 0.47 & 0.53 & 0.65 & 0.27 \end{pmatrix}$$

$$\text{corresponding eigenvalues} = \begin{pmatrix} 2.02 & 1.95 & 0.02 & 0.01 \end{pmatrix}$$

A glitch (I)

- It is still the case the there are two dominant eigenvalues, corresponding to the two separate clusters
- But the corresponding eigenvectors do not directly reveal the points of each cluster
  - A linear combination of them, however, will!

Fixing the glitch (I)

- Define, for *n* points and *k* clusters:

  U = $n \times k$ matrix containing the normalized eigenvectors of the *k* largest eigenvalues of **A** in its columns
  - Each row in **U** corresponds to a data point

Fixing the glitch (I)

- In the last numerical example:

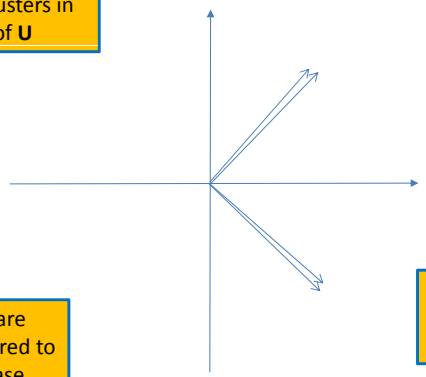
$$\begin{pmatrix} 0.53 & -0.46 & -0.28 & 0.65 \\ 0.54 & -0.46 & 0.27 & -0.65 \\ 0.46 & 0.54 & -0.65 & -0.27 \\ 0.47 & 0.53 & 0.65 & 0.27 \end{pmatrix}$$

= **U**

We notice that rows of **U** corresponding to the same cluster are approximately equal

Fixing the glitch (I)

Points belonging to the same cluster clusters in the row space of **U**



Cluster points are rotated compared to the previous case

Use *k*-means clustering to find these clusters in the row space of **U**

A clustering algorithm

- Assume *n* points and *k* clusters
- Compute $n \times n$ affinity matrix **A**
- Compute the eigensystem of **A**
- There should be *k* non-zero eigenvalues
- Set **U** to hold the corresponding normalized eigenvectors in its columns
- Apply *k*-means clustering on the row space of **U** to find the *k* clusters

An observation

- The self-affinity of each point is a constant value found in the diagonal of **A**
- Changing this constant means adding a term to **A** that is proportional to the identity matrix:

$$\mathbf{A}' = \mathbf{A} + \alpha\,\mathbf{I}$$

An observation

- **A** and **A**' have the same eigenvectors but their eigenvalues differ:

$$\lambda'_k = \lambda_k + \alpha$$

$$\mathbf{A}' = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Same eigenvectors as before

With α = -1

$$\text{corresponding eigenvalues} = \begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix}$$

An observation

- In the literature it is common to set the self-affinity to zero
  - All diagonal elements of **A** are zero
- The phrase

  "*k* eigenvalues of **A** are non-zero"

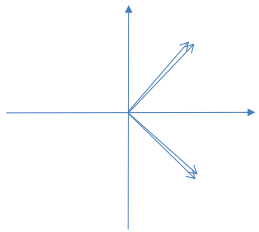
  should then be replaced by

  "*k* eigenvalues of **A** are large"

An observation (II)

In the previous numerical example:

- Not only are the row vectors of **U** for points in different clusters distinct, they are orthogonal



- This is not a coincidence!

An observation (II)

- Assuming that the *k* largest eigenvalues of **A** are approximately equal (to λ):

$$\mathbf{A} + \alpha \mathbf{I} = \lambda \mathbf{U} \mathbf{U}^T$$

The inner product of rows from different clusters correspond to zero affinity in an ideal **A**

In the ideal case: rows in **U** belonging to different clusters must be orthogonal

  - But not necessarily of unit length!

- The *k*-means clustering step should be robust

A clustering algorithm (II)

- Assume *n* points and *k* clusters
- Compute $n \times n$ affinity matrix **A** (0 in diagonal!)
- Compute eigensystem of **A**
- There should be *k* "large" eigenvalues which are approximately equal
- Set **U** to hold the corresponding normalized eigenvectors in its columns
- Apply *k*-means clustering on the row space of **U** to find the *k* clusters

Fiedler's method for *k* = 2

- The Laplacian **L** always (even for noisy data) has an eigenvalue $\lambda_1 = 0$
- Corresponding eigenvector $\mathbf{e}_1$ is **1**
- If *k* = 2, there should be a second eigenvalue = 0, or at least close to zero
- Corresponding eigenvector denoted $\mathbf{e}_2$
- The row space of $\{\mathbf{e}_1, \mathbf{e}_2\}$ should form clusters in two orthogonal directions

Fiedler's method for *k* = 2

- Consequently, the signs of the elements in $\mathbf{e}_2$ must be indicators of the two classes
- For example:

  "+" means class 1

  "-" means class 2

- We don't really need $\mathbf{e}_1$
- Only the signs of the elements in $\mathbf{e}_2$
  - $\mathbf{e}_2$ is often referred to as the *Fiedler vector*

An observation (III)

- Using the "larger" or "significant" eigenvalues of **A** can be replaced with looking for zero or close-to-zero eigenvalues of related matrices
- We need to modify **A** accordingly
- Leads to the Laplacian **L** of **A**, and we do clustering based on the eigensystem of **L** instead of **A**

Degree matrix

- We define

  D = diagonal matrix { d_{ii} }
  where d_{ii} = sum of row/column *i* in **A**

  as the *degree* matrix of **A**

A simple example

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

$$\mathbf{c}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \mathbf{c}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

In the ideal case:
The indicator vectors are eigenvectors both to **A** and **D** and have eigenvalues {1, 2} relative both **A** and **D**

Laplacian

- Formally, we define

  L = **D** – **A**

  as the *Laplacian* of **A**

- The indicator vectors are eigenvectors also of **L**, with eigenvalue 0

Properties of **L**

In the ideal case:

- **L** has the same eigenvectors as **A** and **D**
- **L** has eigenvalues = 0 for the indicator vectors

In general (also with noise):

$a_{ij} \geq 0$ for affinity matrix **A**

$$\mathbf{u}^T \mathbf{L}\, \mathbf{u} = \frac{1}{2} \sum_{i,j=1}^{n} a_{ij}(u_i - u_j)^2$$

L is positive semi-definite!

Properties of **L**

In the general case (also with noise):

- Sum along rows/columns of **L** vanishes
- There is always one eigenvalue = 0 in **L**
- Corresponding eigenvector = is **1** (constant 1)
  - **1** is the sum of all indicator vectors!

Properties of **L**

From this follows:

1. If **u** is a cluster indicator vector ⇒
   u is an eigenvector of **L** with eigenvalue 0
2. If **u** is an eigenvector of **L** with eigenvalue 0 ⇒
   u is a linear combination of the cluster indicator vectors

From this follows:

1. The number of eigenvalues = 0 in **L** is = *k*
   (*k*= number of clusters)
2. The corresponding eigenvectors span the space of indicator vectors

A clustering algorithm (III)

Unnormalized spectral clustering

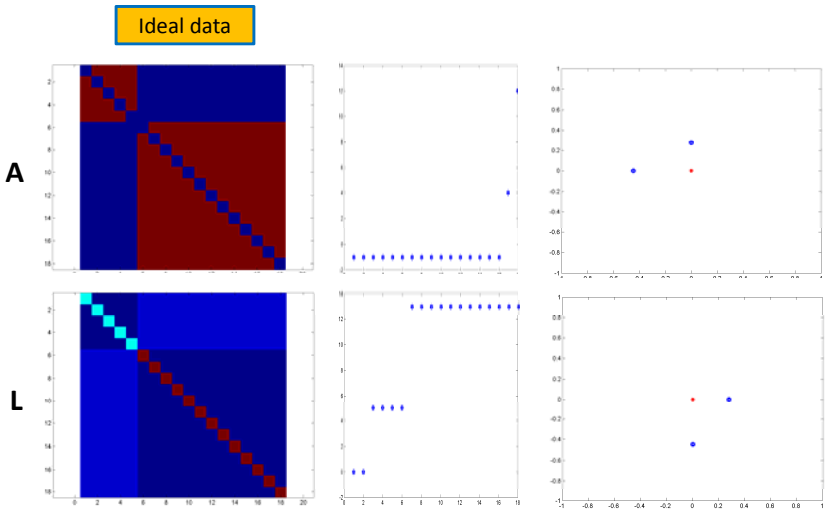
- Assume *n* points and *k* clusters
- Compute $n \times n$ affinity matrix **A**
- Compute **D**, and compute **L** = **D** - **A**
- Compute eigensystem of **L**
- There should be *k* "zero" eigenvalues
- Set **U** to hold the corresponding normalized eigenvectors in its columns
- Apply *k*-means clustering on the row space of **U** to find the *k* clusters

An observation (IV)

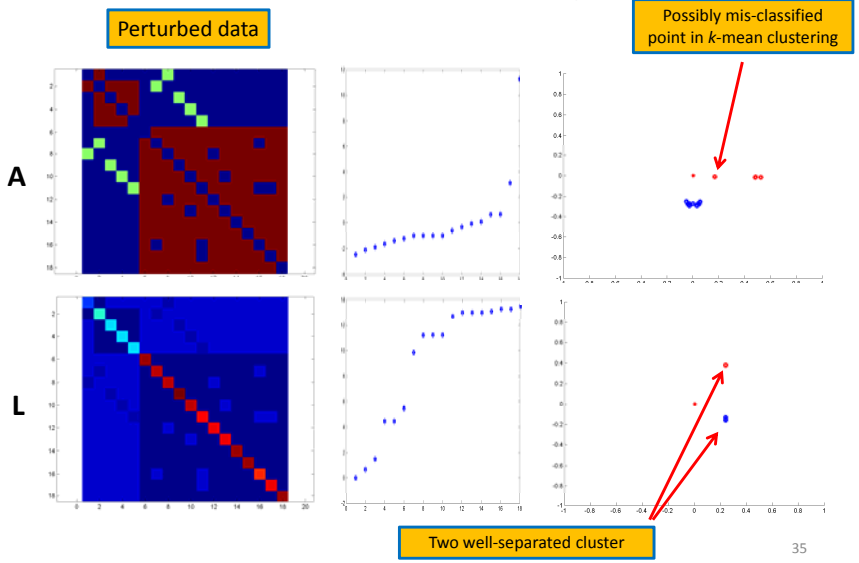
- Should we do clustering on **A** or on **L**?
- For ideal data
  - full connections internally in each segment
  - no connects between segments

  there is, in general, no difference in the result
- For non-ideal data, (= in practice) the results differ
  - Normally: clustering on **L** is better!

A numerical example



A numerical example



Analysis

- It can be shown that the clustering on **A** is equivalent to solving the *mincut* problem of the corresponding graph [see von Luxburg]
- Prefers to cut fewer edges, even if they have higher affinity, than more edges even when each has lower affinity
- In our example: there is a risk of cutting the edge between point 1 and the rest of the points in the first cluster

Analysis

- It can be shown that the clustering on **L** is for $k = 2$ approximates the solution of the *Ratio-cut* problem of the corresponding graph [see von Luxburg]
- Normalizes the cost of a cut with the number of vertices of each sub-graph
- In our example: reduces the risk of cutting the edge between point 1 and the rest of the points in the first cluster

A glitch (II)

- The last clustering algorithm works well for arbitrary k, but assumes that the number of points in each cluster, n_k, is approximately equal
- Otherwise, eigenvalues which are "zero" and "non-zero" may mix in the data of real data

A simple example

- An ideal **A** with $k = 2$ and n_1 and n_2 points in each cluster

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \dots & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & 0 & \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{matrix} \} n_1 \\ \\ \\ \\ \\ \} n_2 \\ \\ \\ \\ \end{matrix}$$

A simple example

- Eigensystem of **A**

$$\mathbf{c}_1 = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \mathbf{c}_2 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{matrix} \} n_1 \\ \\ \} n_2 \end{matrix}$$

$$\text{corresponding eigenvalues} = \begin{pmatrix} n_1 - 1 & n_2 - 1 & \underbrace{-1 \quad \dots \quad -1}_{n-2} \end{pmatrix}$$

A simple example

$$\mathbf{D} = \begin{pmatrix} n_1-1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & n_1-1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & n_1-1 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & n_1-1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & n_2-1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & n_2-1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & n_2-1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n_2-1 \end{pmatrix} \begin{matrix} \} n_1 \\ \\ \\ \\ \\ \} n_2 \\ \\ \\ \\ \\ \end{matrix}$$

A simple example

- Eigensystem of **D**

$$\mathbf{c}_1 = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \mathbf{c}_2 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{matrix} \} n_1 \\ \\ \\ \\ \} n_2 \\ \\ \\ \\ \end{matrix}$$

$$\downarrow \qquad \downarrow$$

$$(\; n_1 - 1 \quad n_2 - 1 \quad \underbrace{n_1 - 1 \quad \dots \quad n_1 - 1}_{n_1 - 1} \quad \underbrace{n_2 - 1 \quad \dots \quad n_2 - 1}_{n_2 - 1} \;)$$

A simple example

- Eigensystem of **L**

$$\mathbf{c}_1 = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \mathbf{c}_2 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{matrix} \} n_1 \\ \\ \\ \\ \} n_2 \\ \\ \\ \\ \end{matrix}$$

$$\downarrow \qquad \downarrow$$

$$(\; 0 \quad 0 \quad \underbrace{n_1 \quad \dots \quad n_1}_{n_1 - 1} \quad \underbrace{n_2 \quad \dots \quad n_2}_{n_2 - 1} \;)$$

A glitch (II)

- For this example:
  - There are 2 eigenvalues approximately = 0
  - There are n_1 - 1 eigenvalues approximately = n_1
  - There are n_2 - 1 eigenvalues approximately = n_2
- If $n_2 >> n_1$ and with sufficiently noisy data:
  - The first two types of eigenvalues can mix
  - Also their eigenvectors will mix
  - Poor clustering performance

Fixing the glitch (II)

- There are (at least) two ways of fixing this glitch, where both normalize the Laplacian **L** before computing the eigensystem:
  - Normalized spectral clustering according to Shi & Malik (2000)
  - Normalized spectral clustering according to Ng *et al* (2002)

Fixing the glitch (II)

- Consider the generalized eigenvalue equation:

$$\mathbf{L}\,\mathbf{u} = \lambda\,\mathbf{D}\,\mathbf{u}$$

Since **L** and **D** share eigenvectors, any such eigenvector is also an eigenvector of this generalized eigenvalue equation

- Eigenvalues are different!

A simple example

Using the previous example:

- The indicator vectors $\mathbf{c}_1$ and $\mathbf{c}_2$ are both eigenvectors of **L**, with eigenvalues 0
  - They are also generalized eigenvectors of (**L**, **D**), with eigenvalue 0
- The remaining eigenvectors of **L** have eigenvalues n_1 ($n_1 - 1$ copies) or n_2 ($n_2 - 1$ copies)

A simple example

- The eigenvalues of the remaining generalized eigenvectors relative (**L**, **D**) are then given by

$$n_k \,/\, (n_k - 1)$$

- This provides a normalization of the eigenvalues that makes the clustering less sensitive to the cluster sizes

A clustering algorithm (IV)
Normalized spectral clustering (Shi-Malik)

- Assume n points and k clusters
- Compute $n \times n$ affinity matrix **A**, and its **D**
- Compute **L** = **D** - **A**
- Compute generalized eigensystem of (**L**, **D**)
- There should be k "zero" eigenvalues
- Set **U** to hold the corresponding normalized eigenvectors in its columns
- Apply k-means clustering on the row space of **U** to find the k clusters

An observation (V)

- We notice that the eigensystem of

$$\mathbf{L}\,\mathbf{u} = \lambda\,\mathbf{D}\,\mathbf{u}$$

is the same as for the standard eigenvalue problem

$$\mathbf{D}^{-1}\,\mathbf{L}\,\mathrm{u} = \lambda\,\mathbf{u}$$

An observation (V)

- We define a normalized Laplacian as

$$\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L}$$

- Referred to as the *normalized random walk* Laplacian [see von Luxburg for explanation]
- In general: **L** is not symmetric!
  - is symmetric in the ideal case!

A clustering algorithm (V)
Variant of Shi-Malik

- Assume n points and k clusters
- Compute $n \times n$ affinity matrix **A**, and its **D**
- Compute **L** = **D** – **A**
- Compute $\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L}$
- Compute eigensystem of $\mathbf{L}_{rw}$
- There should be k "zero" eigenvalues
- Set **U** to hold the corresponding normalized eigenvectors in its columns
- Apply k-means clustering on the row space of **U** to find the k clusters

An observation (VI)

- Alternatively, we define a normalized Laplacian as

$$\mathbf{L}_{sym} = \mathbf{D}^{-1/2}\,\mathbf{L}\,\mathbf{D}^{-1/2}$$

- Referred to as the *normalized symmetric* Laplacian

An observation (VI)

- Is symmetric and has the same eigenvalues as $\mathbf{L}_{rw}$ in the ideal case: $\{0, n_k / (n_k - 1)\}$
- In general, if $\mathbf{v}$ is an eigenvector of $\mathbf{L}_{rw}$, then $\mathbf{D}^{1/2}\mathbf{v}$ is an eigenvector of $\mathbf{L}_{sym}$
- The cluster indicator vectors are eigenvectors also of $\mathbf{L}_{sym}$, with eigenvalues = 0
- We can consider the eigensystem of $\mathbf{L}_{sym}$ instead!

A glitch (III)

A simple example with three ideal clusters

- n_1, n_2, n_3 points each
- The indicator vectors $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$ are eigenvectors of $\mathbf{L}_{sym}$ with eigenvalue 0
- Normalized to unit norm they become

$$\hat{\mathbf{c}}_1 = \begin{pmatrix} 1/\sqrt{n_1} \\ \vdots \\ 1/\sqrt{n_1} \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \hat{\mathbf{c}}_2 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1/\sqrt{n_2} \\ \vdots \\ 1/\sqrt{n_2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \hat{\mathbf{c}}_3 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1/\sqrt{n_3} \\ \vdots \\ 1/\sqrt{n_3} \end{pmatrix} \begin{matrix} \} n_1 \\ \} n_2 \\ \} n_3 \end{matrix}$$

A glitch (III)

- In the practical case, these is some noise and the three eigenvectors if $\mathbf{L}_{sym}$ corresponding to eigenvalue "zero" are linear combinations of the previous vectors
  - Normalized linear combinations!
  - Correspond to rotations of the previous vectors
  - Therefore we do *k*-means clustering on the row space of **U** to find the clusters
  - If n_1, n_2, n_3 are of different magnitudes:
    - Clusters with many points are found close to the origin

Fixing the glitch (III)

- We normalize the rows of **U** before the final *k*-means clustering
- The resulting rows lie on a unit hyper-sphere
- This leads to a better separation of the clusters in the row space of **U**
- We return to the issue of clustering points on a sphere in the following lecture

A clustering algorithm (VI)
Ng et al (2002)

- Assume n points and k clusters
- Compute $n \times n$ affinity matrix **A**, and its **D**
- Compute $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- Compute $\mathbf{L}_{sym} = \mathbf{D}^{-1/2}\,\mathbf{L}\,\mathbf{D}^{-1/2}$
- Compute eigensystem of $\mathbf{L}_{sym}$
- There should be k "zero" eigenvalues
- Set **U** to hold the corresponding normalized eigenvectors in its columns
- Set **T** = **U** but with each row normalized to unit norm
- Apply *k*-means clustering on the row space of **T** to find the k clusters

Does it matter with algorithm we use?

- The unnormalized algorithm is attractive since it is simple, but
  - Use it only when you know that the clusters have the same order of points
- The other two are approximately of the same order of additional computations
  - Von Luxburg suggests using $\mathbf{L}_{rw}$ instead of $\mathbf{L}_{sym}$ as the normalized Laplacian
  - In practice $\mathbf{L}_{sym}$ appears to work as well

Does it matter with algorithm we use?

- For large set of data, the Shi-Malik approach has the advantage of an implicit normalization by solving a generalized eigenvalue problem:

$$\mathbf{L}\,\mathbf{u} = \lambda\,\mathbf{D}\,\mathbf{u}$$

without having to modify any matrix, and may therefore be faster than using normalized Laplacians

Summary

- 3 basic algorithms for spectral clustering
  - Unnormalized: $\mathbf{L}\,\mathbf{u} = \lambda\,\mathbf{u}$
  - Shi-Malik
    - Generalized eigenvalue problem: $\mathbf{L}\,\mathbf{u} = \lambda\,\mathbf{D}\,\mathbf{u}$
    - Standard eigenvalue problem: $\mathbf{L}_{rw}\,\mathbf{u} = \lambda\,\mathbf{u}$
  - Ng, et al: $\mathbf{L}_{sym}\,\mathbf{u} = \lambda\,\mathbf{u}$
- Spectral properties of **A**, **D**, **L**
  - Relations to the cluster indicator vectors

Next lecture

- Practical aspects, parameter tuning
- Extensions of spectral clustering algorithms
- Applications to real problems
  - Mainly in computer vision