One Week Online Workshop on Statistics and Machine Learning in Practice
Brahmananda Keshab Chandra College

# Spectral Clustering


Aparajita Khan
Senior Research Fellow
Indian Statistical Institute

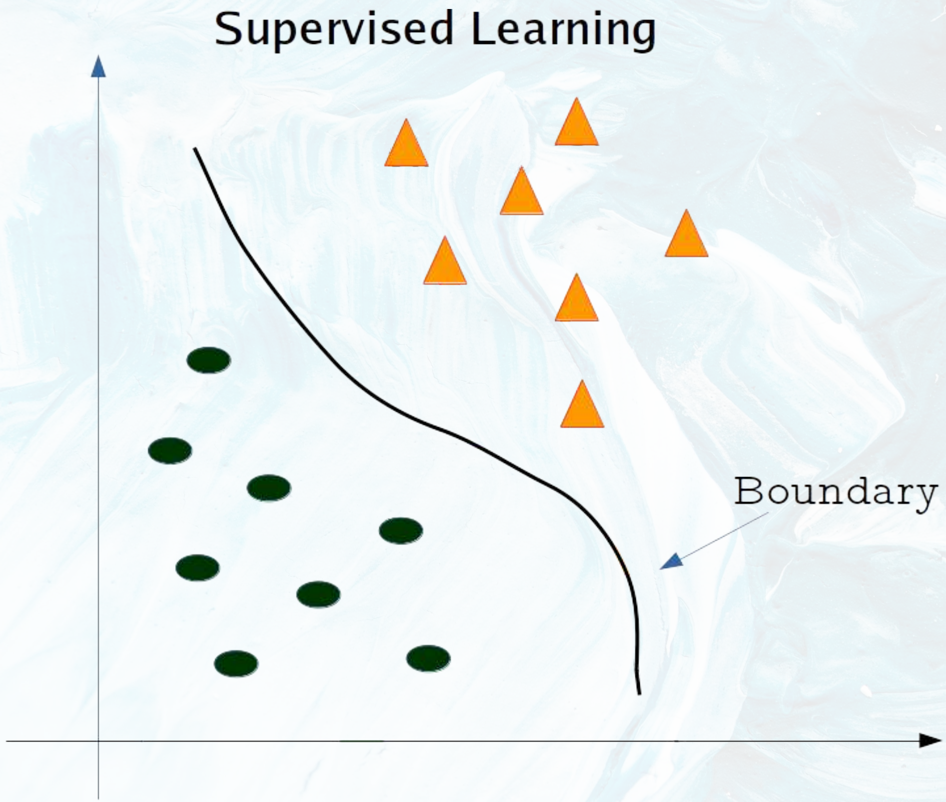
July 29, 2020

# Clustering

The study of natural groupings in data

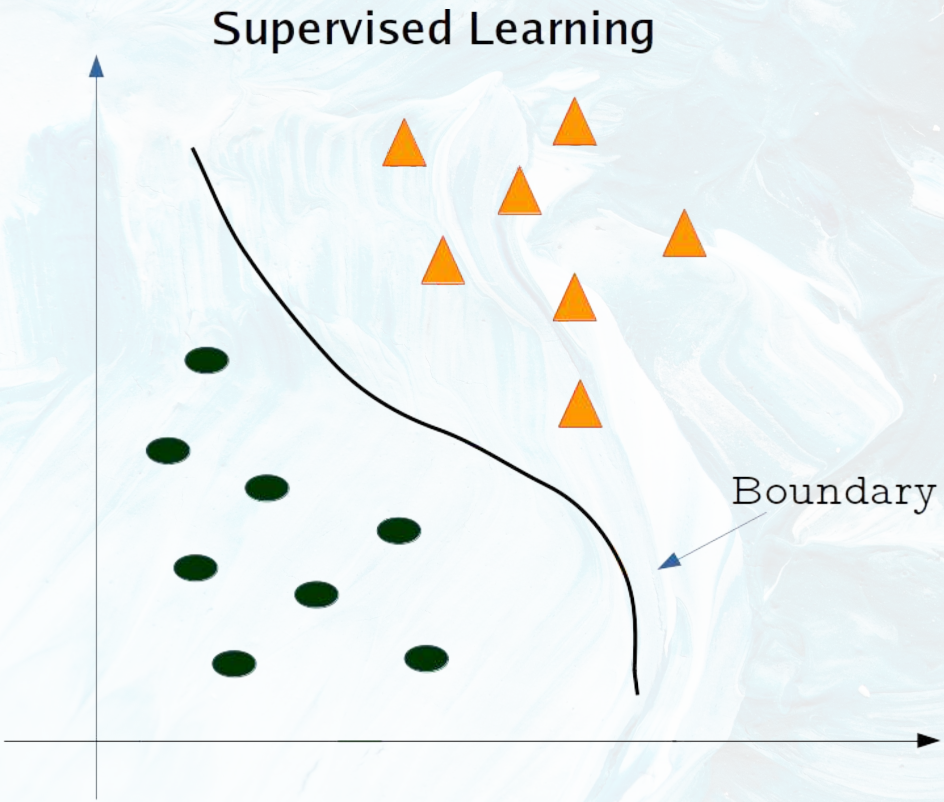
# Clustering

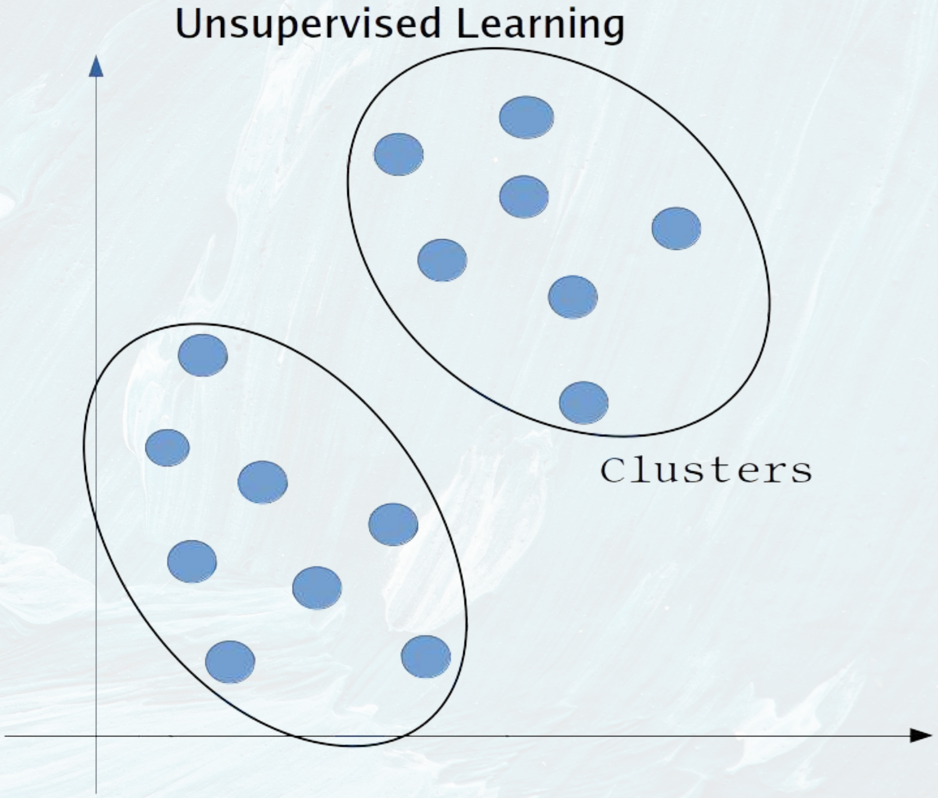
The study of natural groupings in data



# Clustering

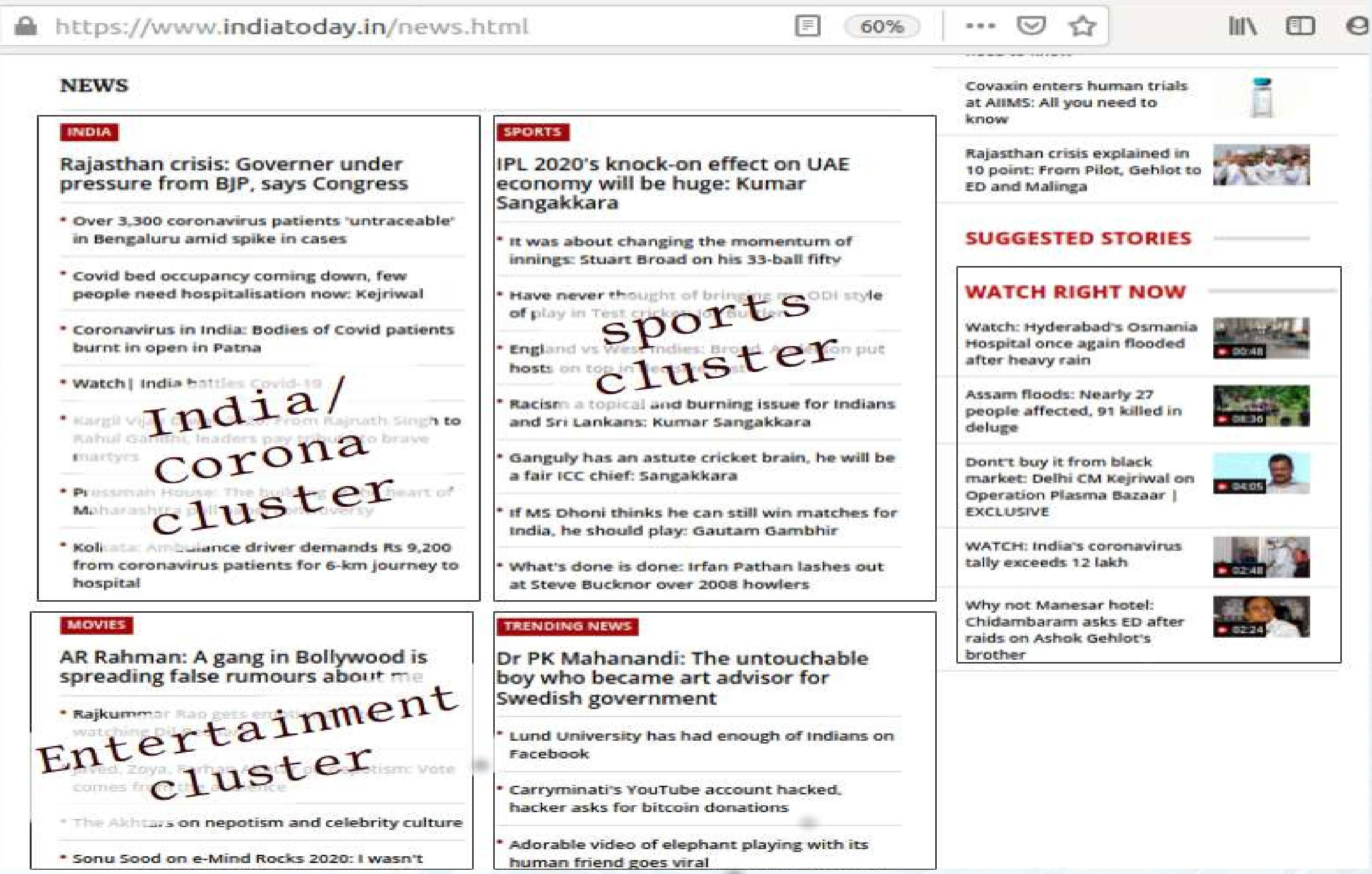
The study of natural groupings in data



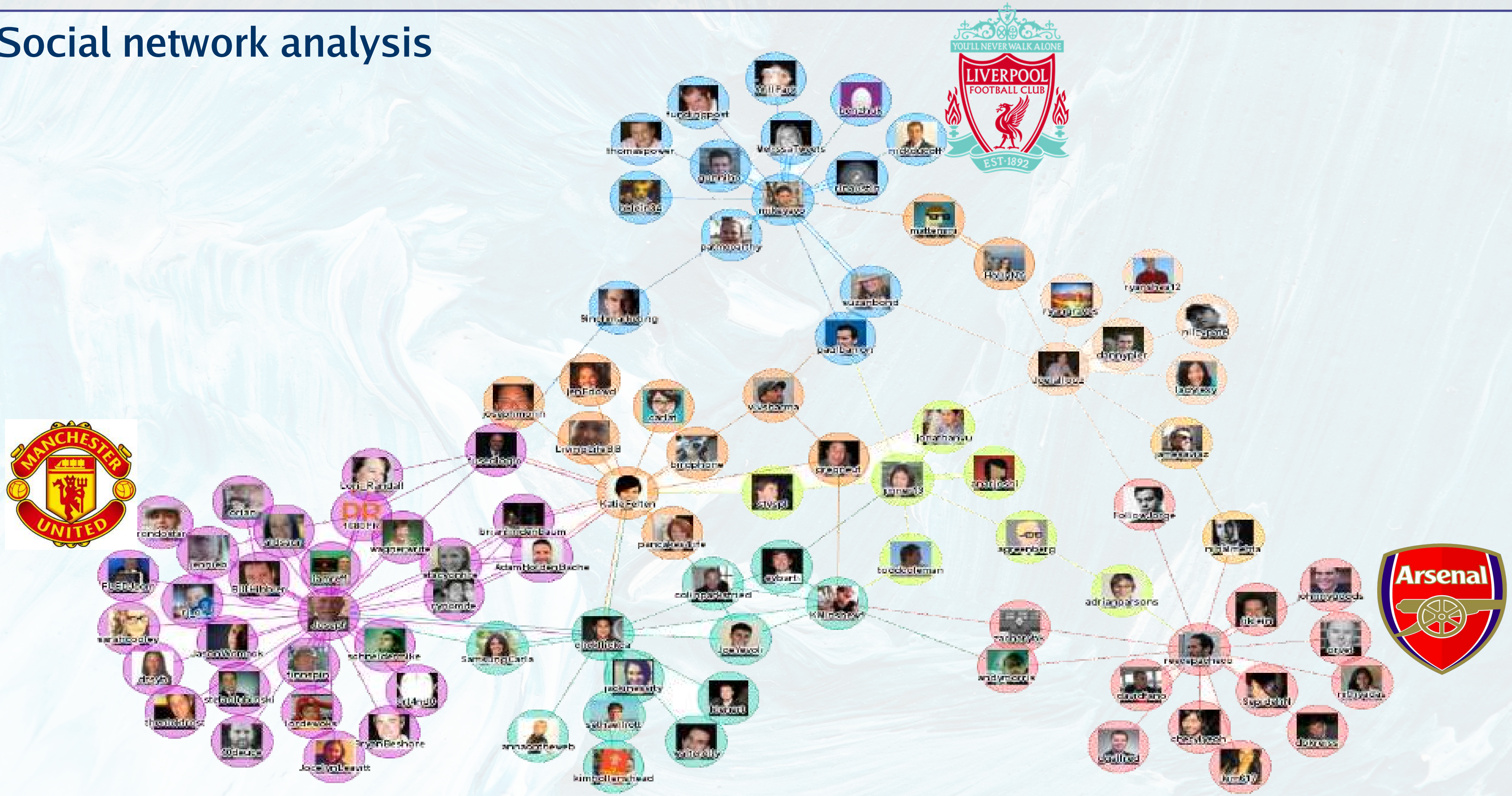
# CLUSTERING: Applications

Grouping similar news articles



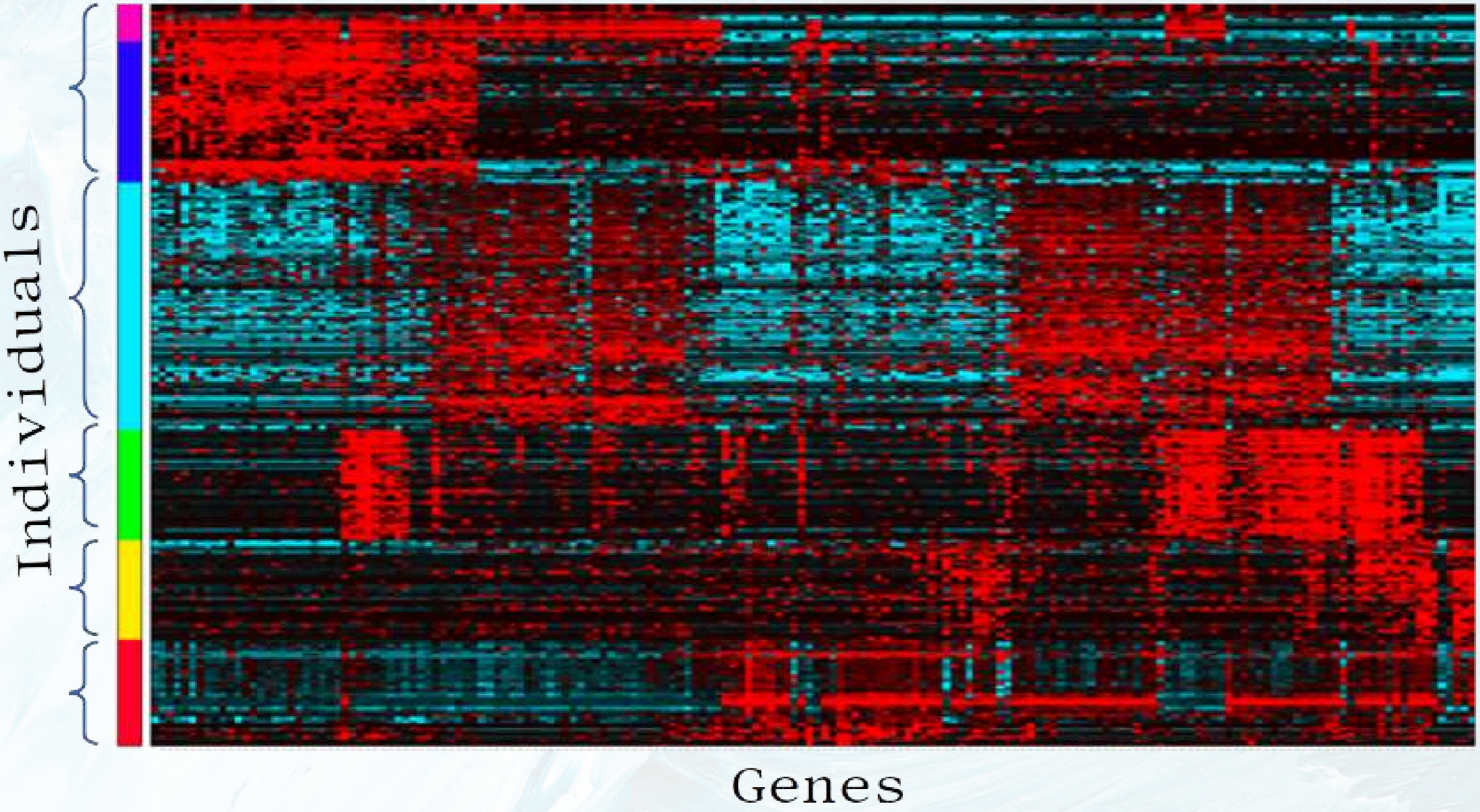
# Clustering: Applications

## Social network analysis



# Clustering: Applications

**Bioinformatics: Identifying disease subtypes, patient groups**



# Clustering: Applications

Image segmentation

# Feature-space vs Graph Clustering

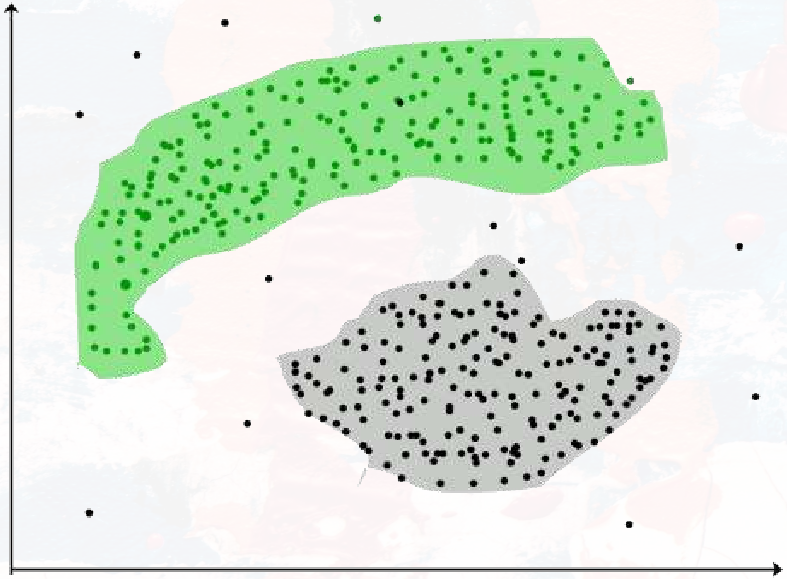
## Feature-space: *k*-Means Clustering

$\mathbf{X} \in \Re^{n \times d}$ `n samples, d features`

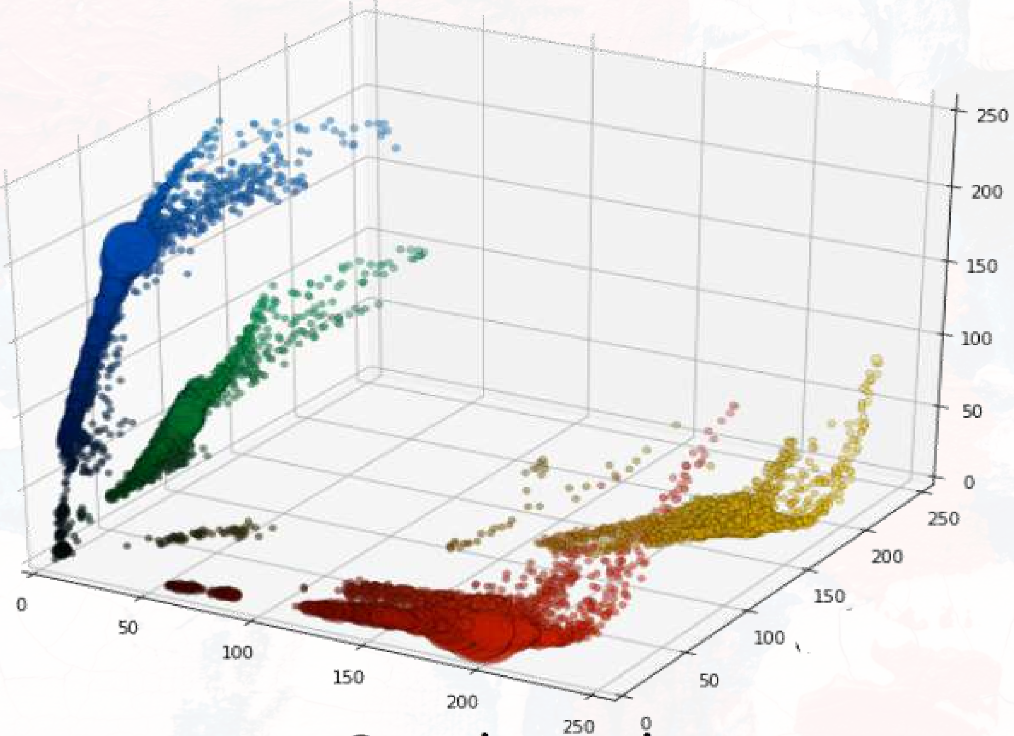
# Feature-space vs Graph Clustering

## Feature-space: $k$-Means Clustering

$\mathbf{X} \in \Re^{n \times d}$ n samples, d features



2 Dimenions




3 Dimenions

# Feature-space vs Graph Clustering

## Feature-space: *k*-Means Clustering

- But, what about high dimensions?

  Disease subtyping: ~1,000 samples   ~20,000 genes

  Object recognition: 1024x1024 images -> ~1M pixels

# Feature-space vs Graph Clustering

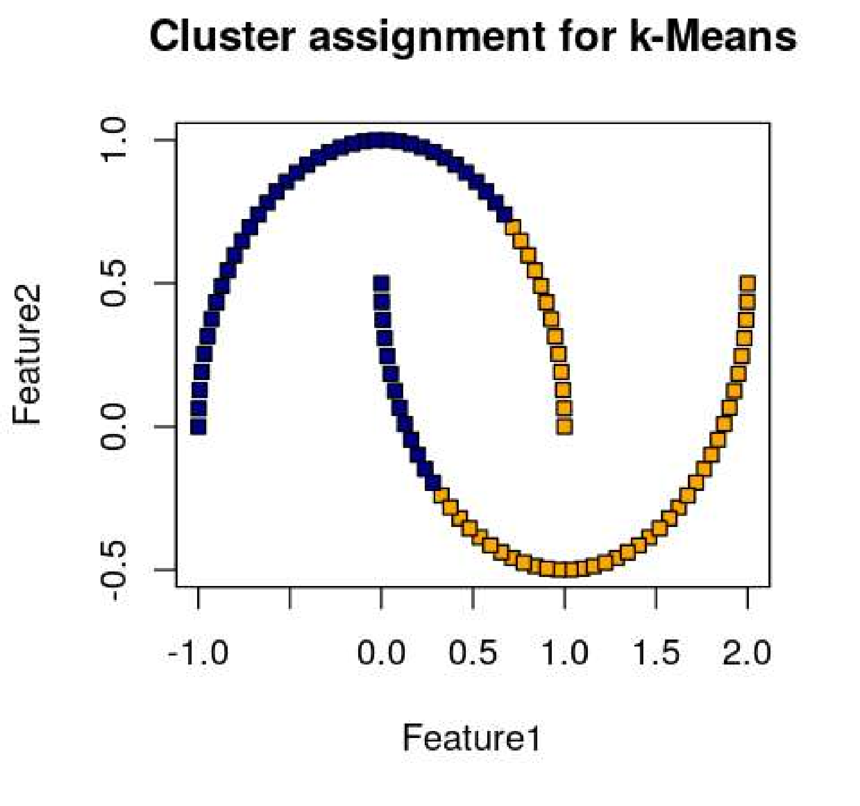
## Feature-space: *k*-Means Clustering

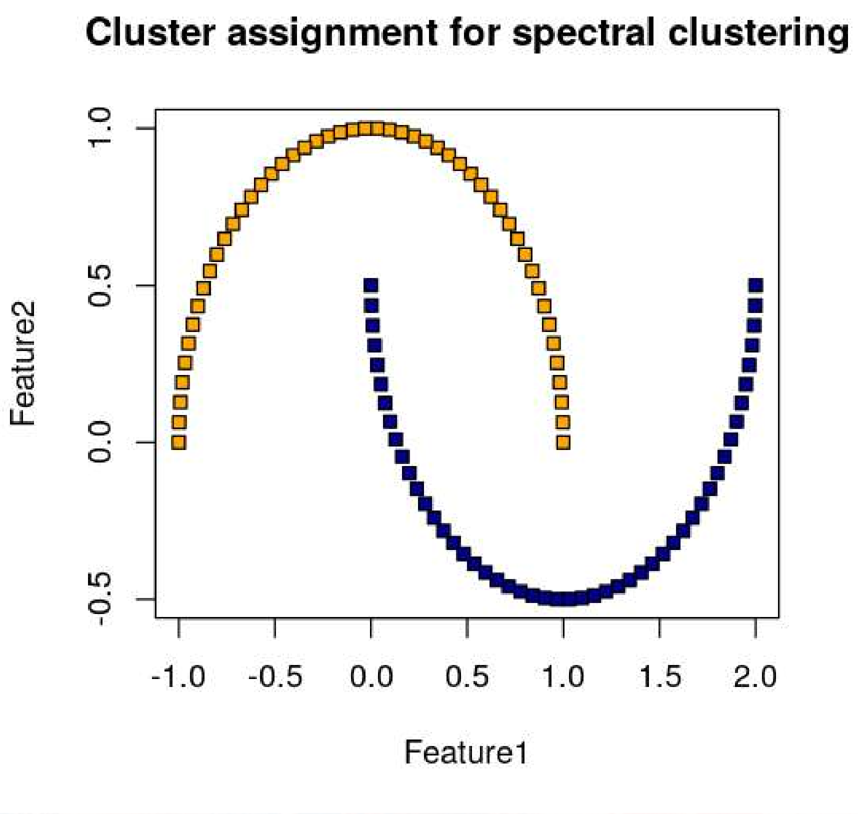
- But, what about high dimensions?
  Disease subtyping: ~1,000 samples ~20,000 genes
  Object recognition: 1024x1024 images -> ~1M pixels

- In such high dimensions:
  Data becomes geometrically sparse
  Distance between nearby points roughly same as distance between far away points

# Feature-space vs Graph Clustering

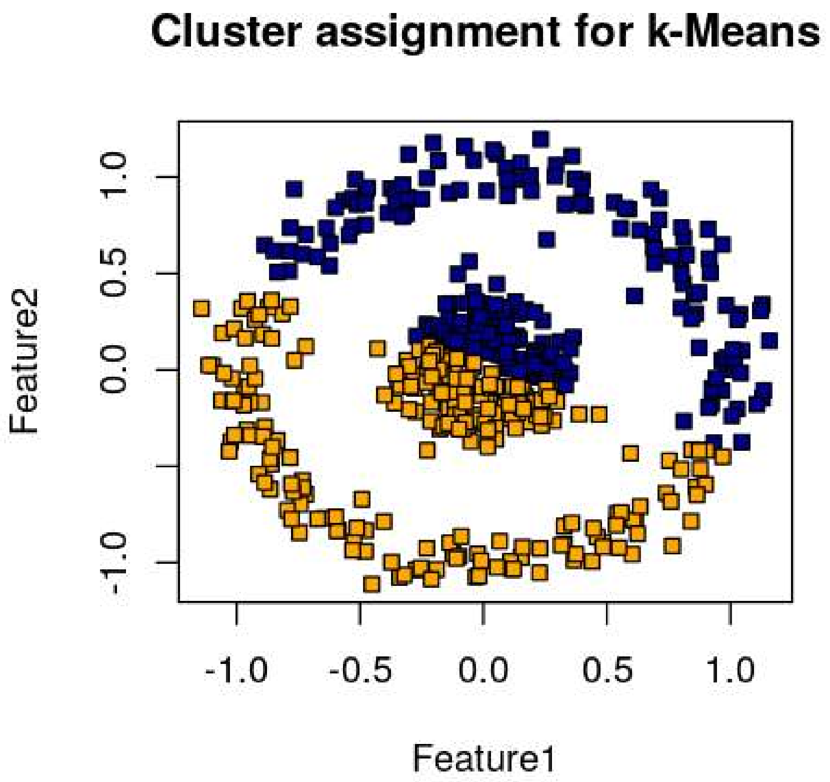
## Handling non-lineartiy

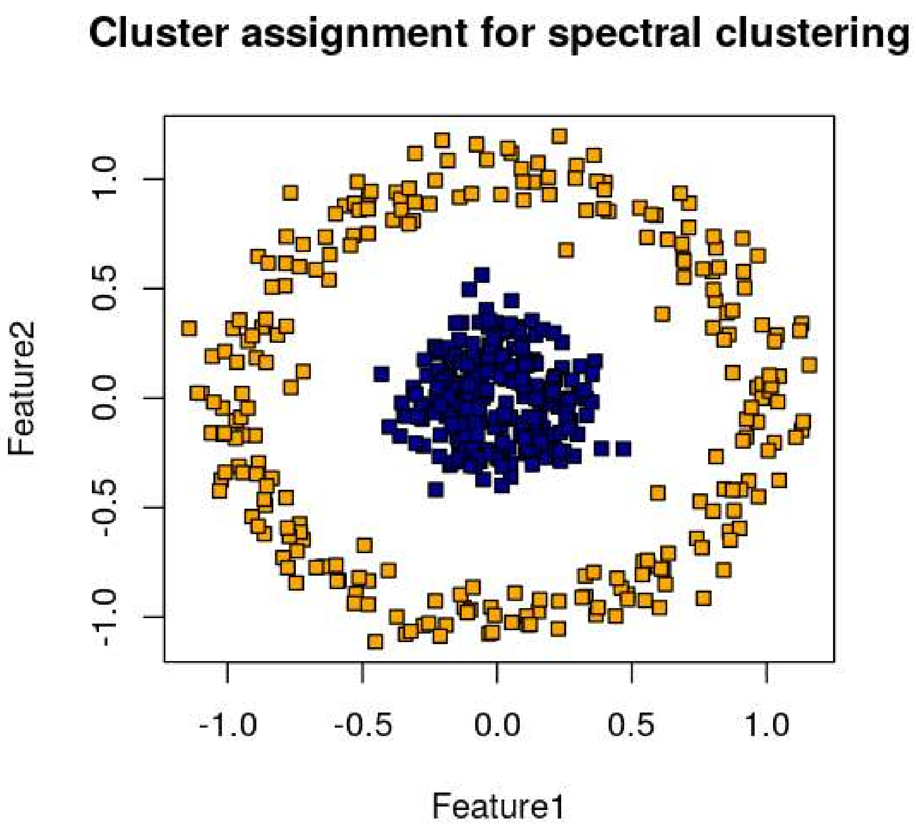


# Feature-space vs Graph Clustering

## Handling non-lineartiy



# Feature-space vs Graph Clustering

Feature-space based Representation

Graph based representation

# Feature-space vs Graph Clustering

## Feature–space

$$\mathbf{X} \in \Re^{n \times d}$$

```
Samples: n
Features/Dimension: d
```

## Graph

# Feature-space vs Eigenspace Clustering

## Feature–space

$$\mathbf{X} \in \Re^{n \times d}$$

Samples: n
Features/Dimension: d

$$\kappa(x_i, x_j) = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right)$$

Gaussian similarity kernel

## Graph

$$\mathbf{W} \in \Re^{n \times n}$$

Pairwise similarity matrix

# Feature-space vs Graph Clustering

**Feature–space**

$\mathbf{X} \in \Re^{n \times d}$

Samples: n
Features/Dimension: d

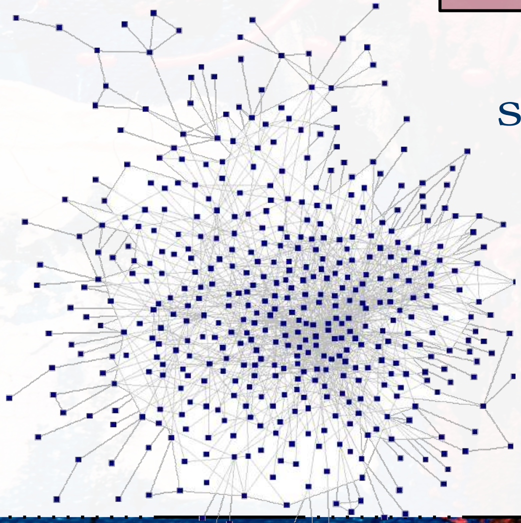
$$\kappa(x_i, x_j) = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right)$$

Gaussian similarity kernel

**Graph**

$\mathbf{W} \in \Re^{n \times n}$

Pairwise
similarity
matrix

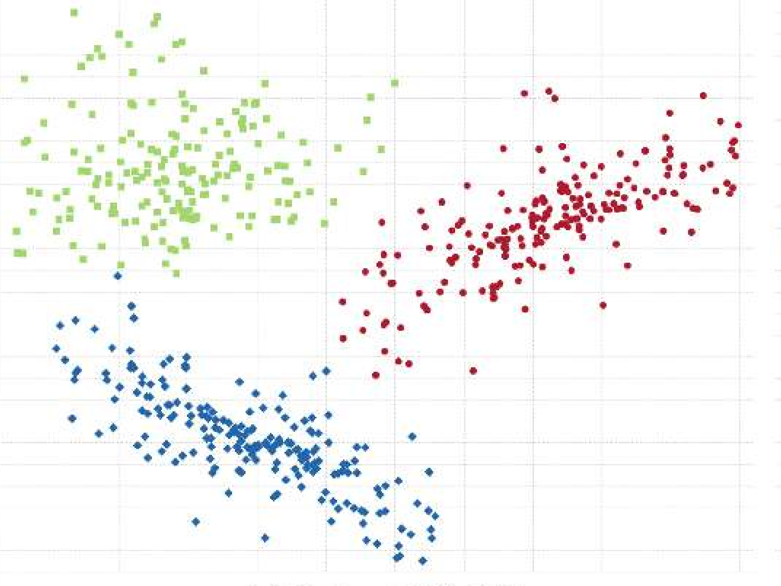


weighted
adjacency

**G**

# Feature-space vs Graph Clustering

## Feature-space

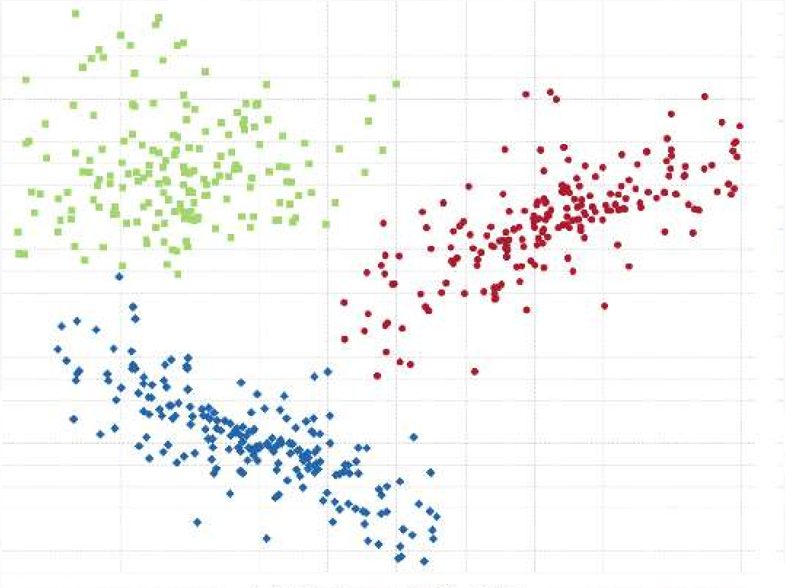


Clusters in dataset

## Graph

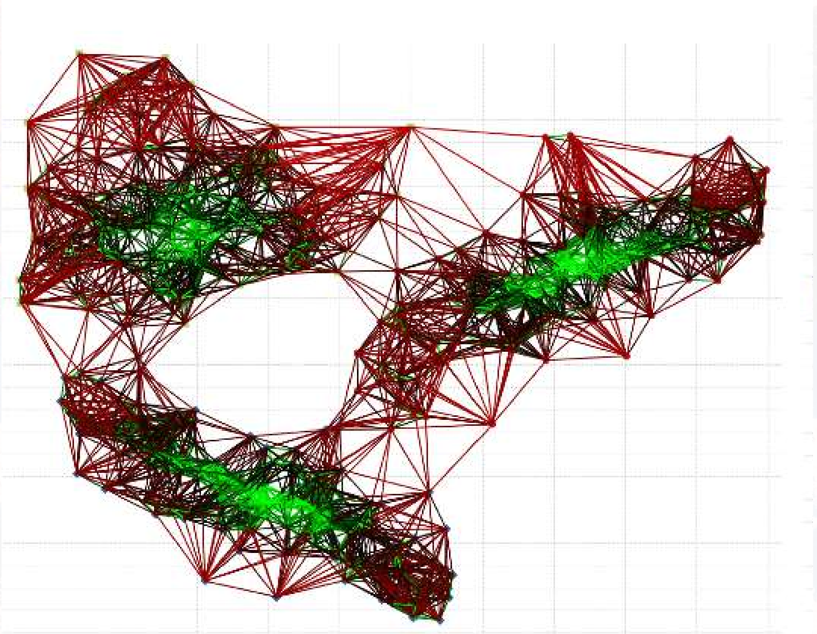
# Feature-space vs Graph Clustering

## Feature-space



Clusters in dataset

## Graph

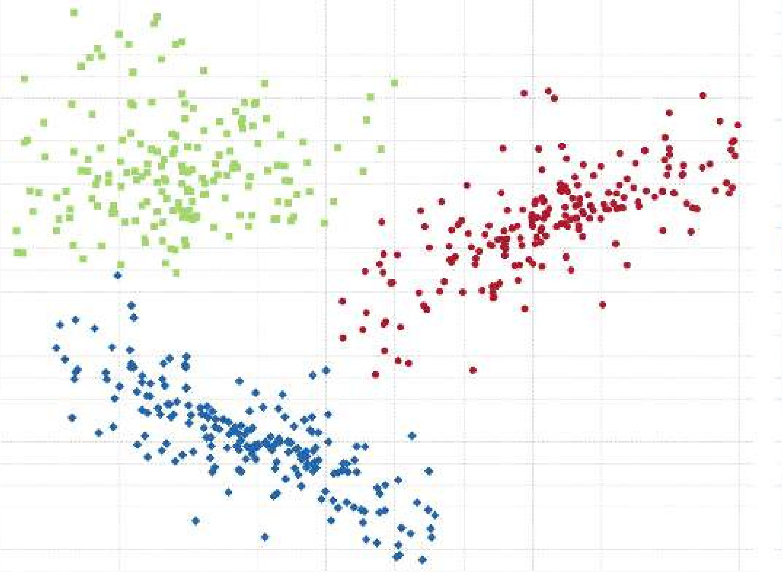


Strongly connected components in graph

Image: Barton, Tomás et al. “Chameleon 2.,” ACM Transactions on Knowledge Discovery from Data, vol. 13, pp. 1 - 27, 2019.

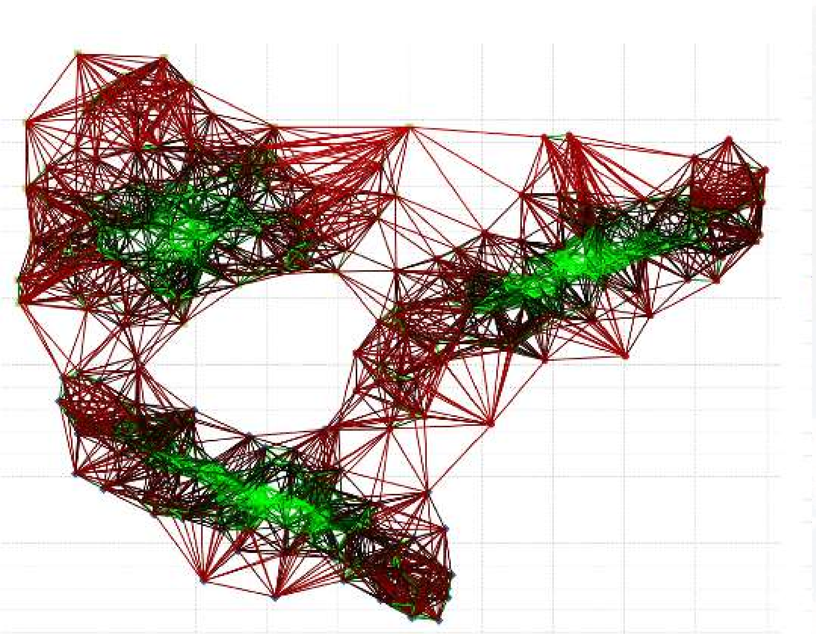
# Feature-space vs Graph Clustering

**Feature-space**



$$\text{minimize} \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \left\| x_i - \frac{1}{|\mathcal{C}_j|} \sum_{t \in \mathcal{C}_j} x_t \right\|_2^2$$

**Graph**



**Graph-cut problem**

# The Graph-Cut

Find a k-way partition of graph such that:

Edges between different components have very low weight

(points in different clusters are dissimilar)

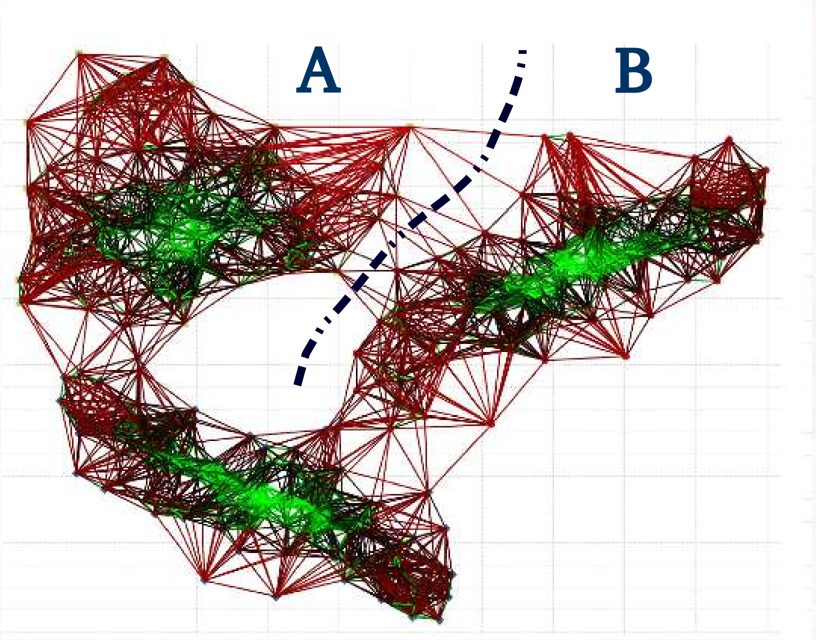
Edges within a component have high weight

(points within same cluster are similar)

# The Graph-Cut

$$\mathbf{W} = [w_{ij}]_{n \times n}$$

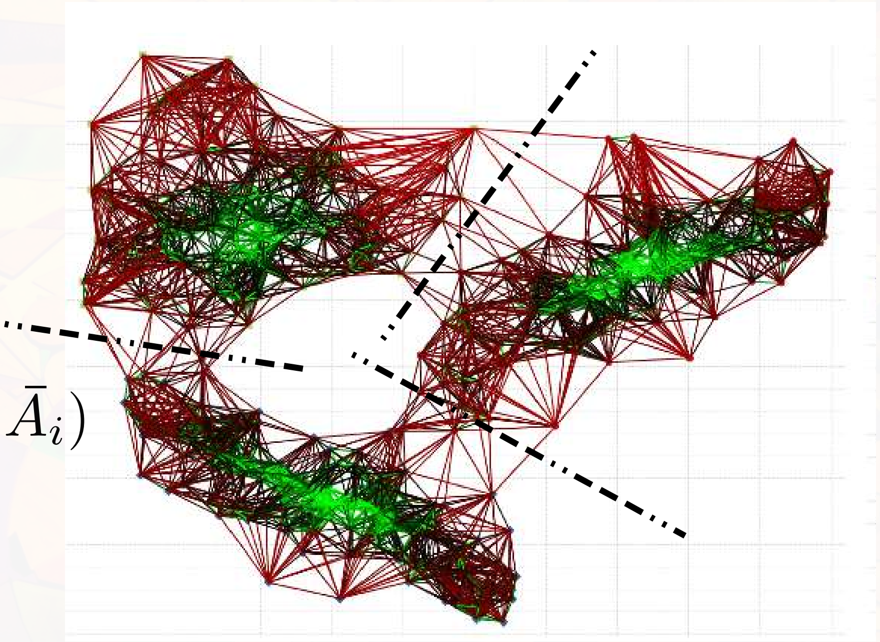
$$\text{cut } W(A, B) := \sum_{i \in A, j \in B} w_{ij}$$



# The Graph-Cut

$$\mathbf{W} = [w_{ij}]_{n \times n}$$

$$\text{cut}\ W(A,B) := \sum_{i \in A, j \in B} w_{ij}$$



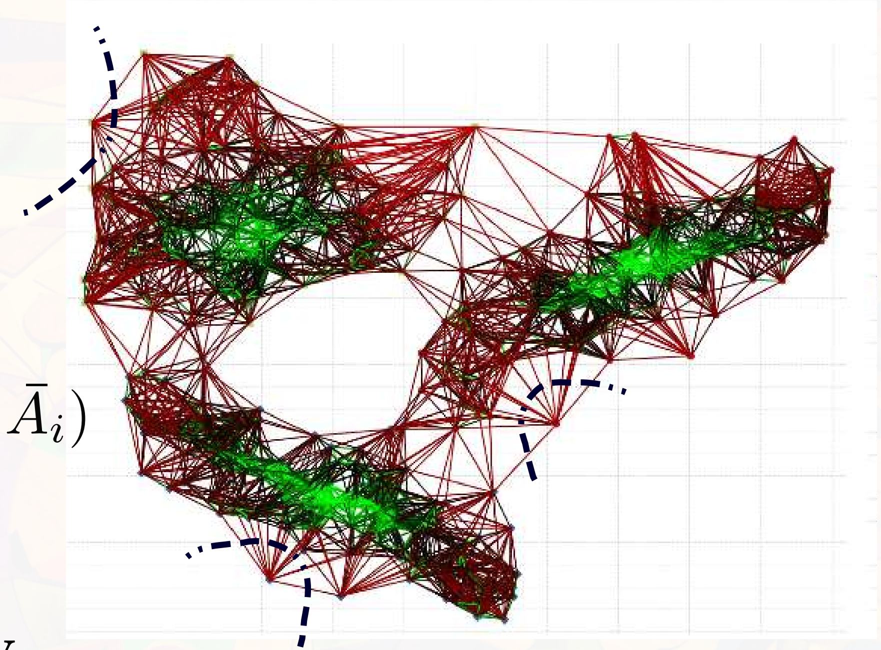
k-way cut:

$$\text{minimize cut}(\mathrm{A}_1, A_2, ..., A_k) := \sum_{i=1}^{k} W(A_i, \bar{A}_i)$$

# The Graph-Cut

$$\mathbf{W} = [w_{ij}]_{n \times n}$$

$$\text{cut } W(A, B) := \sum_{i \in A, j \in B} w_{ij}$$



k-way cut:

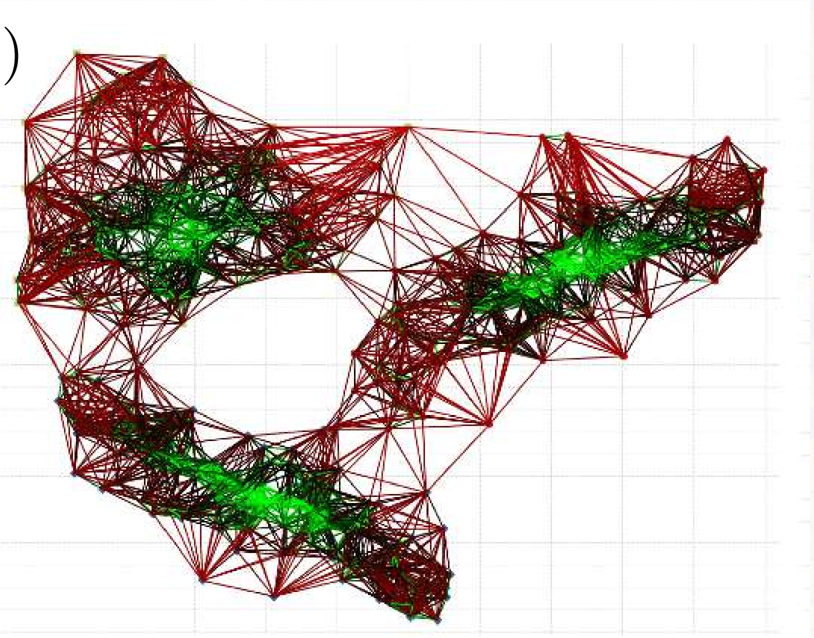
$$\text{minimize cut}(\mathrm{A}_1, A_2, ..., A_k) := \sum_{i=1}^{k} W(A_i, \bar{A}_i)$$

- In many cases, mincut simply separates individual vertices from rest of the graph

# The Graph-Cut

$$\text{minimize } \mathrm{cut}(\mathrm{A}_1, A_2, ..., A_k) := \sum_{i=1}^{k} W(A_i, \bar{A}_i)$$

Explicitly request subsets to be “reasonably large”



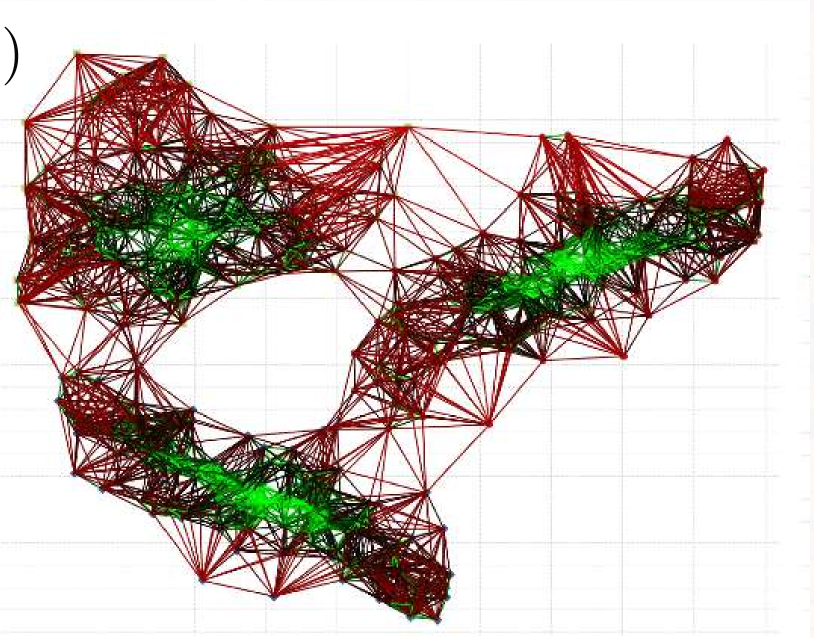
# The Graph-Cut

$$\text{minimize } \mathrm{cut}(\mathrm{A}_1, A_2, ..., A_k) := \sum_{i=1}^{k} W(A_i, \bar{A}_i)$$

Explicitly request subsets to be "reasonably large"

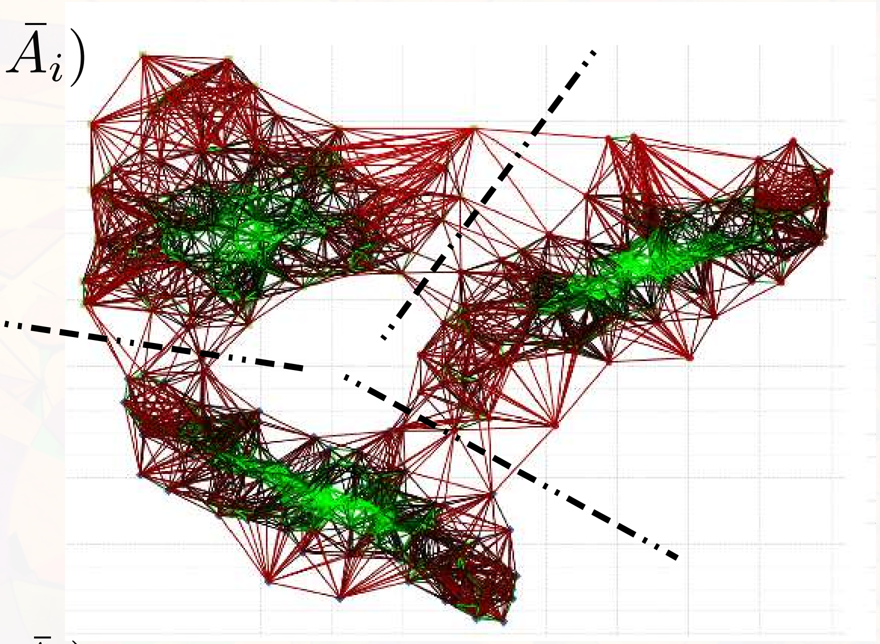
degree of vertex $v_i$: $d_i = \sum_{j=1}^{n} w_{ij}$

size of subset: $\mathrm{vol}(A_i) := \sum_{i \in A} d_i$



# The Graph-Cut

$$\text{minimize } \mathrm{cut}(\mathrm{A}_1, A_2, \ldots, A_k) := \sum_{i=1}^{k} W(A_i, \bar{A}_i)$$



Explicitly request subsets to be “reasonably large”

$$\text{degree of vertex } v_i\text{: } d_i = \sum_{j=1}^{n} w_{ij}$$

$$\text{size of subset: } \mathrm{vol}(A_i) := \sum_{i \in A} d_i$$

Normalized cut

$$\text{minimize } \mathrm{Ncut}(\mathrm{A}_1, A_2, \ldots, A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(\mathrm{A_i})}$$

# Solving the Graph-Cut

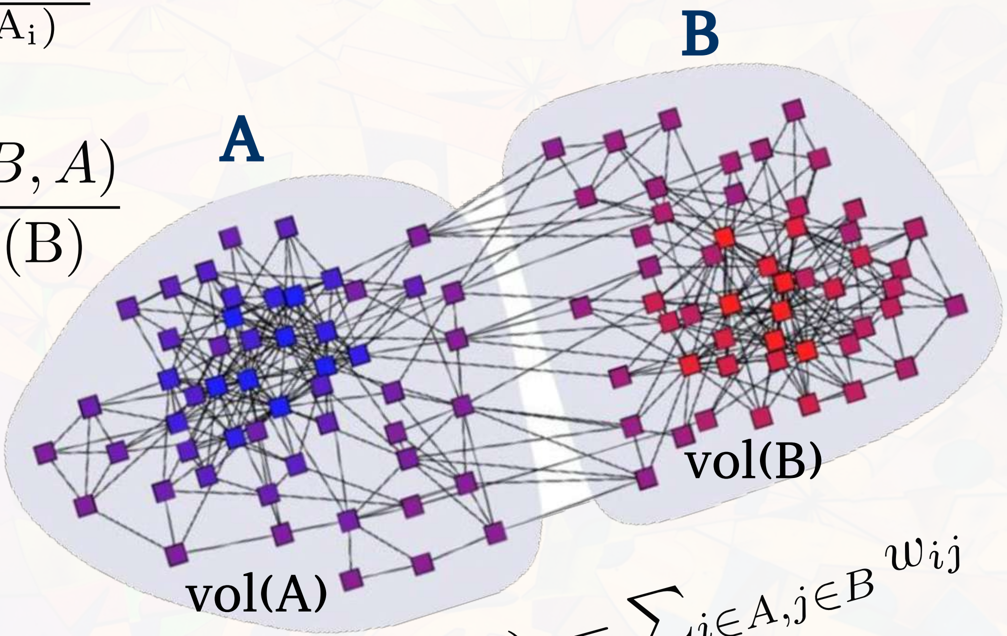
$$\text{minimize } \mathrm{Ncut}(\mathrm{A}_1, A_2, \ldots, A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(\mathrm{A_i})}$$

# Solving the Graph-Cut

$$\text{minimize } \mathrm{Ncut}(\mathrm{A}_1, A_2, ..., A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(\mathrm{A_i})}$$

Solving 2-way cut

$$\text{minimize } \mathrm{Ncut}(A, B) := \frac{W(A, B)}{\mathrm{vol(A)}} + \frac{W(B, A)}{\mathrm{vol(B)}}$$




$$\mathrm{W(A,B)} := \sum_{i \in \mathrm{A}, j \in \mathrm{B}} w_{ij}$$

# Solving the Graph-Cut

$$\text{minimize } \mathrm{Ncut}(\mathrm{A}_1, A_2, \ldots, A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(\mathrm{A_i})}$$

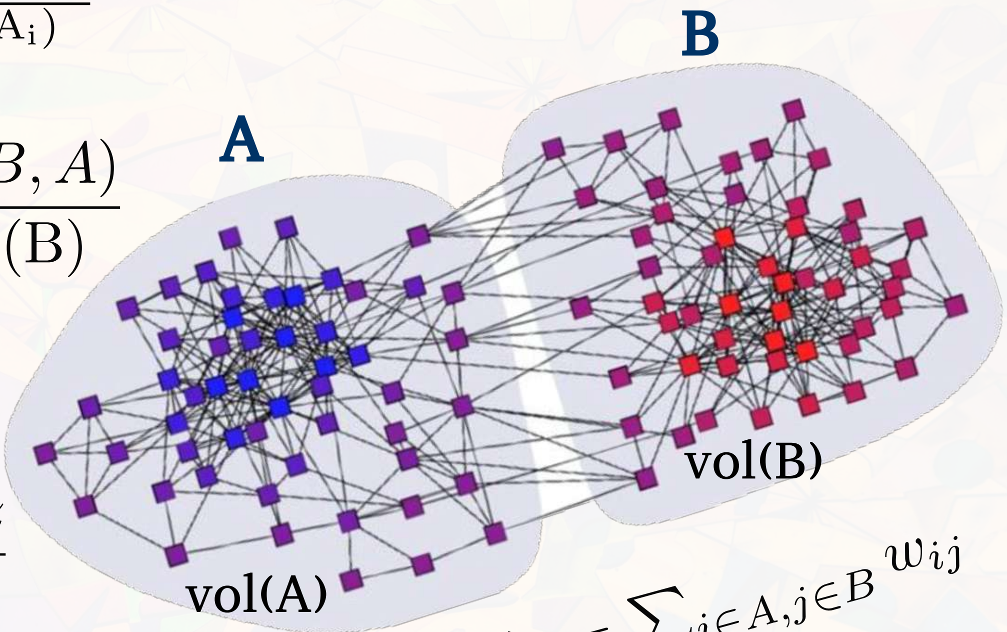
Solving 2-way cut

$$\text{minimize } \mathrm{Ncut}(A, B) := \frac{W(A, B)}{\mathrm{vol(A)}} + \frac{W(B, A)}{\mathrm{vol(B)}}$$

Reduces to

$$\underset{z \in \Re^n}{\text{minimize}} \; \frac{z^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} z}{z^T z}$$

$$\text{such that } \; \mathrm{z}^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$




$$\mathrm{W(A,B)} := \sum_{i \in A, j \in B} w_{ij}$$

Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, Aug. 2000, doi: 10.1109/34.868688.

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \; \frac{z^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} z}{z^T z}$$

$$\text{such that} \quad \mathrm{z}^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \; \frac{z^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} z}{z^T z}$$

$$\text{such that} \quad \mathrm{z}^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$

$\mathbf{W}$=Pairwise-similarity matrix of size $n \times n$

$$\mathbf{D} = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \text{ Degree matrix}$$

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \ \frac{z^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} z}{z^T z}$$

$$\text{such that} \quad \mathrm{z}^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$

The normalized graph **Laplacian**

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}}$$

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \; \frac{z^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} z}{z^T z}$$

$$\text{such that} \;\; \mathrm{z}^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$

The normalized graph **Laplacian**

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}}$$

- Symmetric positive semi-definite

$$v^T \mathbf{L} v \geq 0, \forall v \in \Re^n, v \neq \mathbf{0}$$

$$\text{all the eigenvalues are} \geq 0$$

- $\mathbf{D}^{\frac{1}{2}} \mathbf{1}$ is an eigenvector of $\mathbf{L}$ with eigenvalue 0

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \; \frac{z^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} z}{z^T z}$$

$$\text{such that} \quad \mathrm{z}^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$

The normalized graph **Laplacian**

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}}$$

- Symmetric positive semi-definite

$$v^T \mathbf{L} v \geq 0, \forall v \in \Re^n, v \neq \mathbf{0}$$

all the eigenvalues are $\geq 0$

- $\mathbf{D}^{\frac{1}{2}} \mathbf{1}$ is an eigenvector of $\mathbf{L}$ with eigenvalue 0

Rayleigh quotient:

Let $A$ be a real symmetric matrix
Constraint: $v$ is orthogonal to $j-1$ smallest eigenvectors $v_1, \ldots, v_{j-1}$, the quotient $\frac{v^T A v}{v^T v}$ is minimized by next smallest eigenvector $v_j$ and eigenvalue $\lambda_j$

G.H. Golub and C.F. Van Loan, “Matrix Computations,” John Hopkins Press, 1989.

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \ \frac{z^T \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}} z}{z^T z}$$

$$\text{such that} \ \ z^T \left(\mathbf{D}^{\frac{1}{2}}\mathbf{1}\right) = 0$$

Minimized by:
second smallest eigenvector
of $\mathbf{L}$ and its eigenvalue

The normalized graph **Laplacian**

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}$$

- Symmetric positive semi-definite

$$v^T \mathbf{L} v \geq 0, \forall v \in \Re^n, v \neq \mathbf{0}$$

all the eigenvalues are $\geq 0$

- $\mathbf{D}^{\frac{1}{2}}\mathbf{1}$ is an eigenvector of $\mathbf{L}$ with eigenvalue $0$

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \; \frac{z^T \mathbf{L} z}{z^T z} \text{ such that } z^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \ \frac{z^T \mathbf{L} z}{z^T z} \ \text{such that} \ z^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$

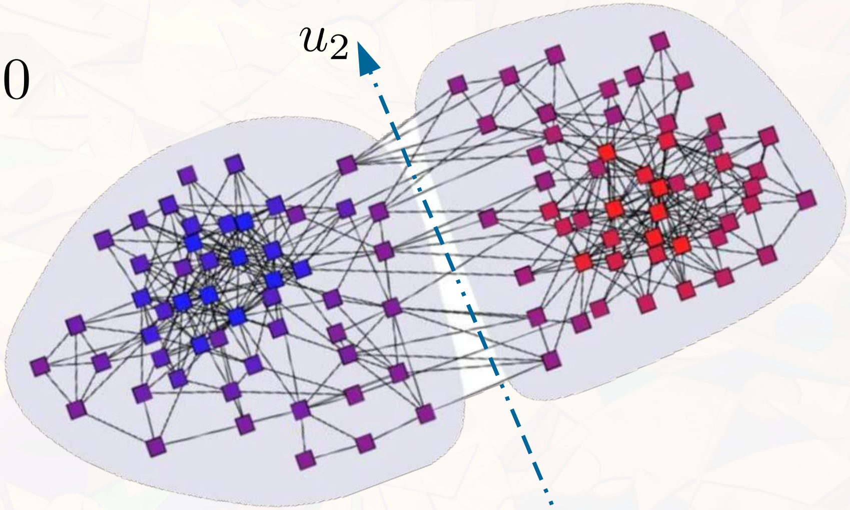
Eigen decomposition $\mathbf{L} = U \Sigma U^T$

$$\begin{bmatrix} \sqrt{d_1} & | & | & & | \\ \vdots & u_2 & u_3 & \dots & u_n \\ \sqrt{d_n} & | & | & & | \end{bmatrix} \qquad \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

(arranged in ascending order)

# Solving the Graph-Cut

$$\underset{z \in \Re^n}{\text{minimize}} \; \frac{z^T \mathbf{L} z}{z^T z} \; \text{such that } z^T \left( \mathbf{D}^{\frac{1}{2}} \mathbf{1} \right) = 0$$




Eigen decomposition $\mathbf{L} = U \Sigma U^T$

$$U = \begin{bmatrix} \sqrt{d_1} & | & | & & | \\ \vdots & u_2 & u_3 & \dots & u_n \\ \sqrt{d_n} & | & | & & | \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

# Solving the Graph-Cut

Solving k-way cut

$$\text{minimize } \mathrm{Ncut}(\mathrm{A}_1, A_2, \ldots, A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol(A_i)}}$$

# Solving the Graph-Cut

Solving k-way cut

$$\text{minimize } \mathrm{Ncut}(\mathrm{A}_1, A_2, \ldots, A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(\mathrm{A_i})}$$

$$\text{minimize } trace(U^T \mathbf{L} U)$$

$$\text{such that } U \in \Re^{n \times k}, U^T U = \mathbf{I}_k$$

U. Von Luxburg,"A tutorial on spectral clustering," Statistics and Computing, vol. 17, no. 4, pp. 395-416, 2007.

# Solving the Graph-Cut

Solving k-way cut

$$\text{minimize } \mathrm{Ncut}(\mathrm{A}_1, A_2, \ldots, A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(\mathrm{A_i})}$$

$$\text{minimize } trace(U^T \mathbf{L} U)$$

$$\text{such that } U \in \Re^{n \times k}, U^T U = \mathbf{I}_k$$

$$= \sum_{i=1}^{k} \lambda_i(\mathbf{L})$$ [Ky-Fan theorem]

$U^* =$ Eigenvectors corresponding to $k$ smallest eigenvalues of $\mathbf{L}$

Fan, K. "On a Theorem of Weyl Concerning Eigenvalues of Linear Transformations: II." Proceedings of the National Academy of Sciences of the United States of America vol. 36, no. 1, pp. 31-35, 1950.

# Solving the Graph-Cut

Solving k-way cut

$$\text{minimize Ncut}(\mathrm{A}_1, A_2, ..., A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(\mathrm{A_i})}$$

$$\text{minimize } trace(U^T \mathbf{L} U)$$

$$\text{such that } U \in \Re^{n \times k}, U^T U = \mathbf{I}_k$$

Eigen decomposition $\mathbf{L} = U \Sigma U^T$

$$\begin{bmatrix} | & | & & | & | & & | \\ u_1 & u_2 & \dots & u_k & u_{k+1} & \dots & u_n \\ | & | & & | & | & & | \end{bmatrix}$$

# Solving the Graph-Cut

Solving k-way cut

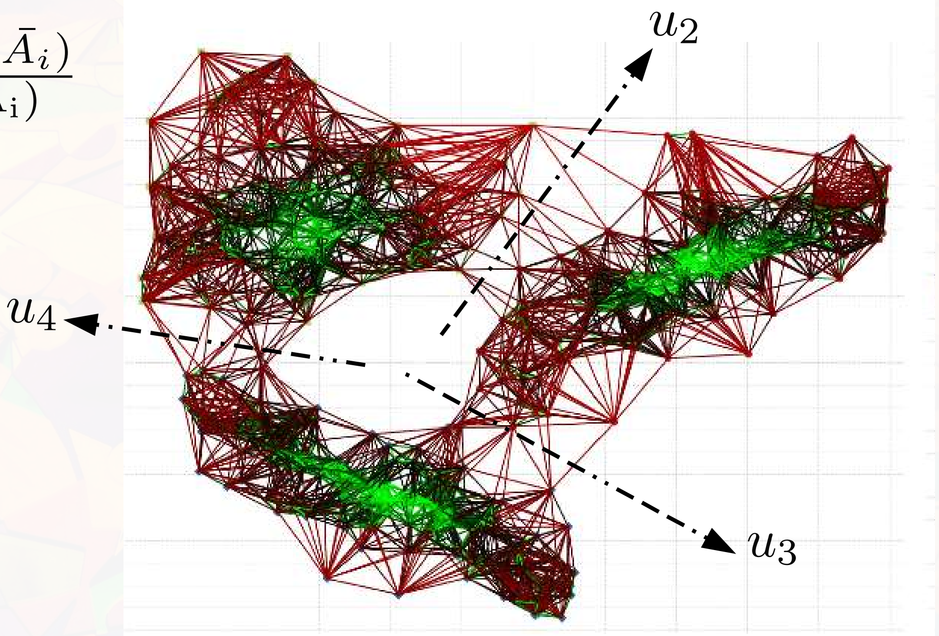
$$\text{minimize } \mathrm{Ncut}(\mathrm{A}_1, A_2, ..., A_k) := \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(\mathrm{A_i})}$$

$$\text{minimize } trace(U^T \mathbf{L} U)$$

$$\text{such that } U \in \Re^{n\times k}, U^T U = \mathbf{I}_k$$

$$\text{Eigen decomposition} \quad \mathbf{L} = U \Sigma U^T$$

$$\begin{bmatrix} | & | & & | & | & & | \\ u_1 & u_2 & \dots & u_k & u_{k+1} & \dots & u_n \\ | & | & & | & | & & | \end{bmatrix}$$



# Spectral Clustering

Why the word "**Spectral**"?

F. R. Chung, Spectral graph theory. American Mathematical Society, no. 92, 1997.

# SPECTRAL CLUSTERING

Why the word "**Spectral**"?

- Derieved from "spectrum"

F. R. Chung, Spectral graph theory. American Mathematical Society, no. 92, 1997.

# Spectral Clustering

Why the word "**Spectral**"?

- Derieved from "spectrum"
- Eigenvalues of a matrix are called its spectrum

F. R. Chung, Spectral graph theory. American Mathematical Society, no. 92, 1997.

# Spectral Clustering

Why the word "**Spectral**"?

- Derieved from "spectrum"
- Eigenvalues of a matrix are called its spectrum
- Here, clustering solutions are obtained from eigenvalues and eigenvectors of some matrix **L**

F. R. Chung, Spectral graph theory. American Mathematical Society, no. 92, 1997.

# Spectral Clustering

Why the word "**Spectral**"?

- Derieved from "spectrum"
- Eigenvalues of a matrix are called its spectrum
- Here, clustering solutions are obtained from eigenvalues and eigenvectors of some matrix **L**
- Clustering using *spectrum* of **L**

"Spectral Clustering"!

F. R. Chung, Spectral graph theory. American Mathematical Society, no. 92, 1997.

# Spectral Clustering Algorithm

Normalized spectral clustering by Ng, Jordan, and Weiss (2002)

**Input** Similarity matrix $\mathbf{W}$, number of clusters $k$.

**Output** Clusters $A_1, \ldots, A_k$.

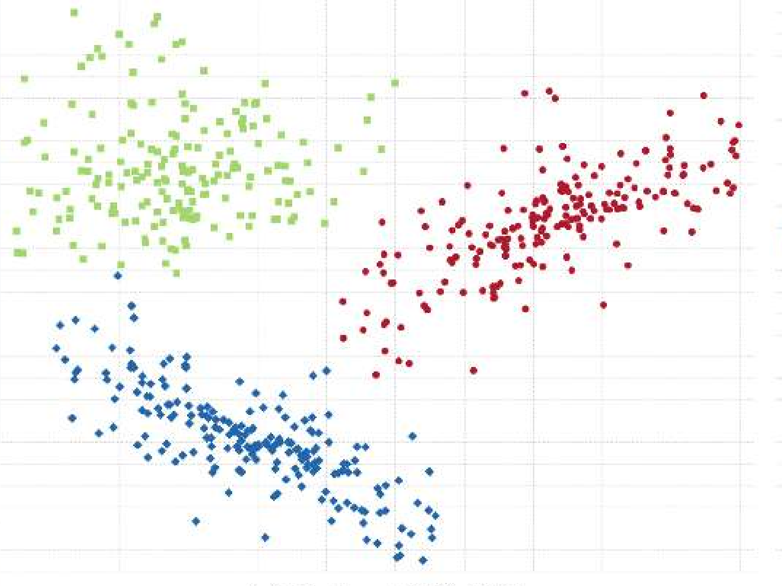
1. Construct degree matrix $\mathbf{D}$ and normalized Laplacian $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}$
2. Find eigenvectors $U = [u_1 \ldots u_k]$ corresponding to $k$ smallest eigenvalues of matrix $\mathbf{L}$.
3. Perform clustering on the rows of $U$ using $k$-means algorithm.

**Return** clusters $A_1, \ldots, A_k$ from $k$-means clustering.

A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in Proc. Advances in Neural Information Processing Systems, pp. 849–856, 2002.

# Spectral Clustering

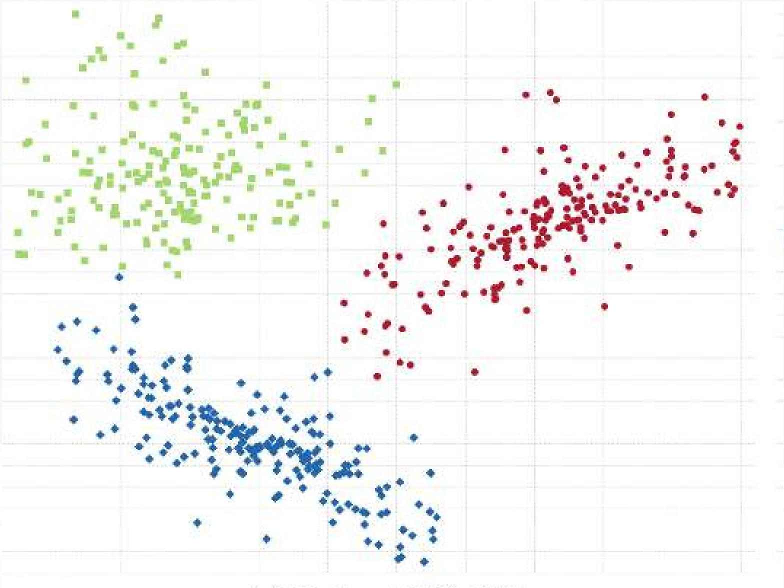
## Feature-space



$$\mathbf{X} \in \Re^{n \times d}$$

# Spectral Clustering

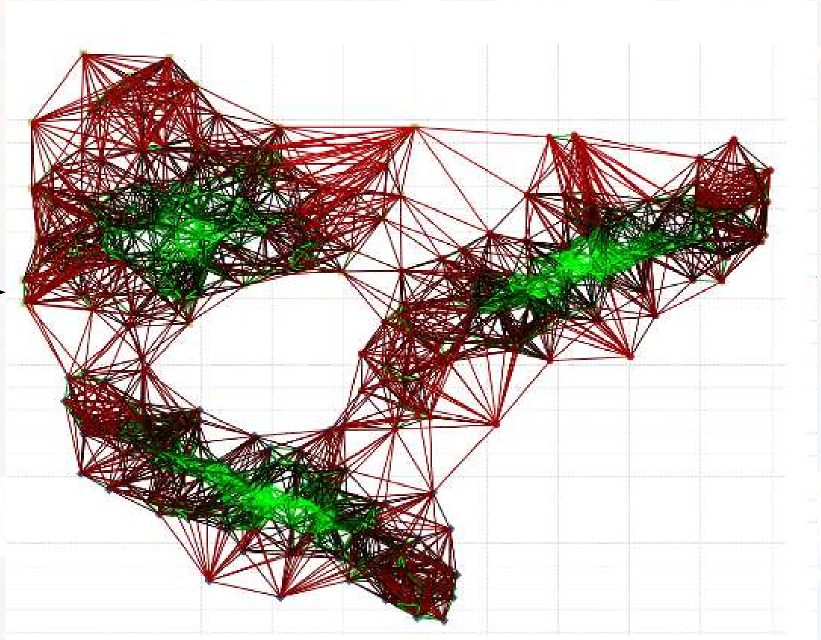
## Feature–space



$\mathbf{X} \in \Re^{n \times d}$

$\xrightarrow{\text{similarity measure}}$

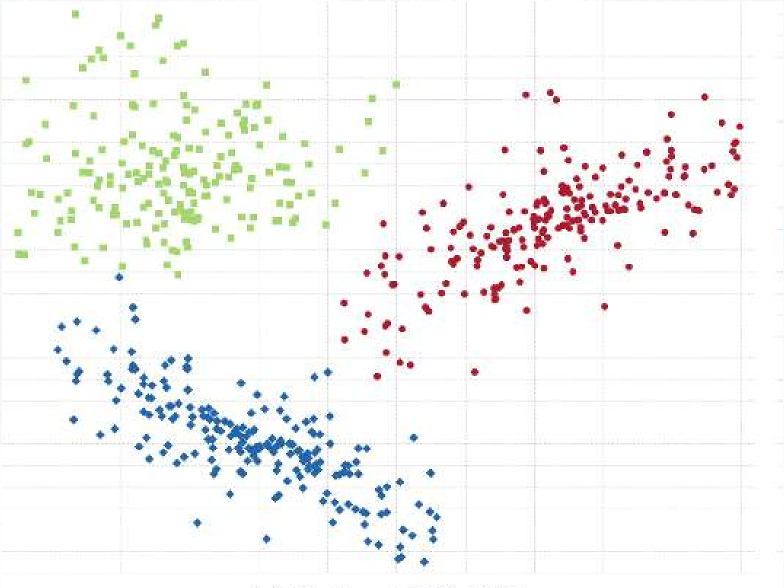
## Graph



$\mathbf{W}$ and Laplacian $\mathbf{L} \in \Re^{n \times n}$

# SPECTRAL CLUSTERING

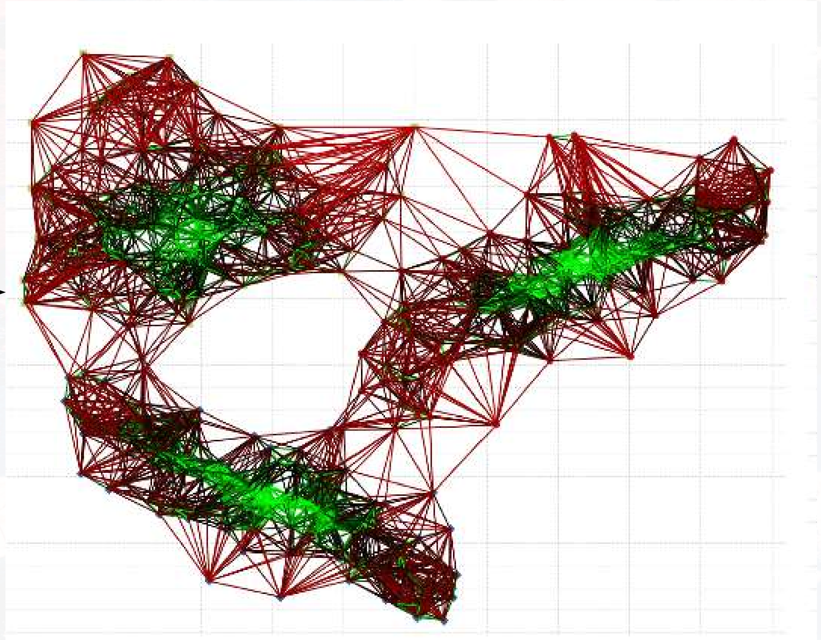
**Feature–space**



$\mathbf{X} \in \Re^{n \times d}$

$\longrightarrow$ similarity measure

**Graph**



$\mathbf{W}$ and Laplacian $\mathbf{L} \in \Re^{n \times n}$

Eigenvectors $U \in \Re^{n \times k}$

$k -$ means clustering on low-rank $U$ $\longleftarrow$

# Spectral Clustering

We will study the
performance of spectral
clustering in lab session.

# Thank You