IMPERIAL

10th March 2025

Graph Based Learning

Clustering and Dimensionality Reduction
Methods in Data Science
MATH70026

Dr Hardik Rajpal

menti.com | code 3393 8461

h.rajpal15@imperial.ac.uk

Any Marvel Fans?

What happened in the Karate Club?

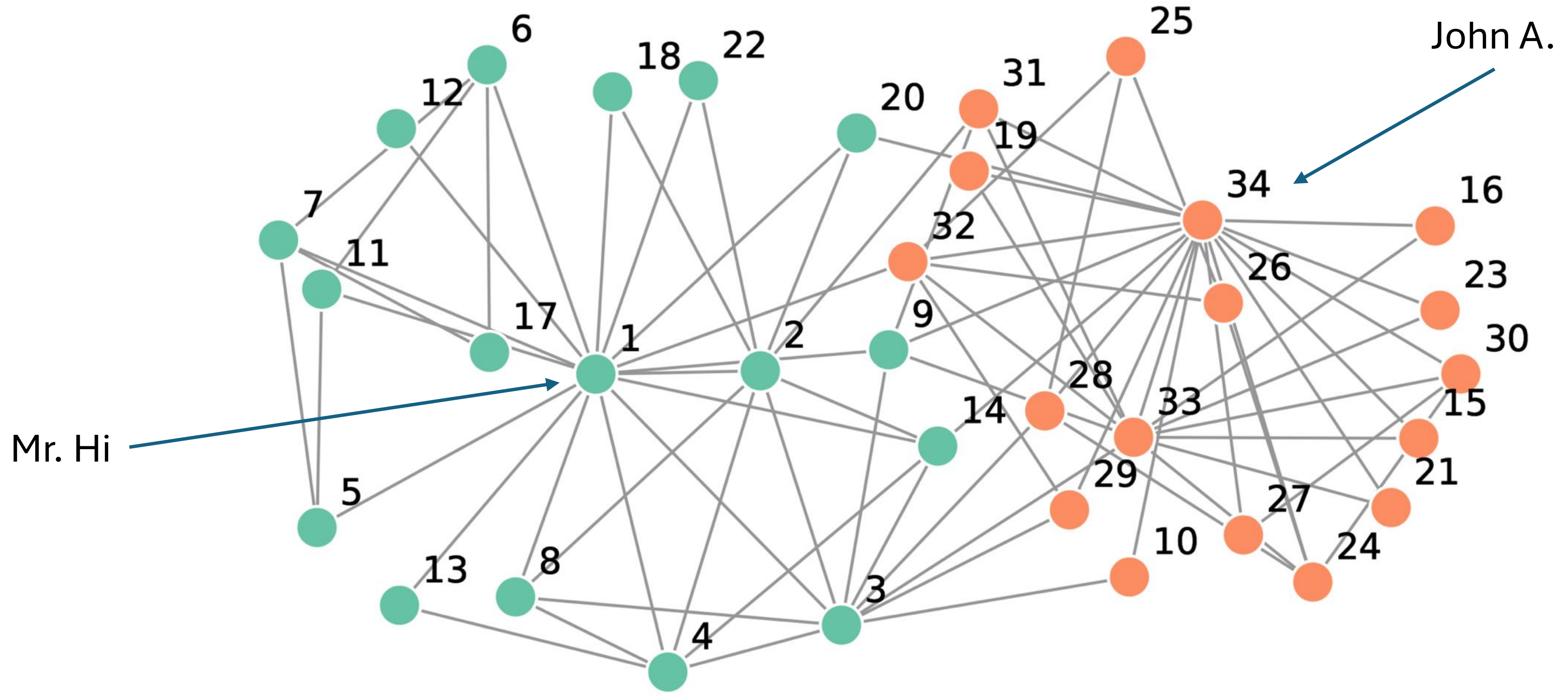
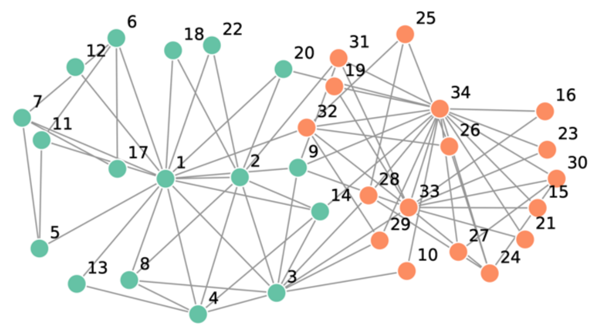
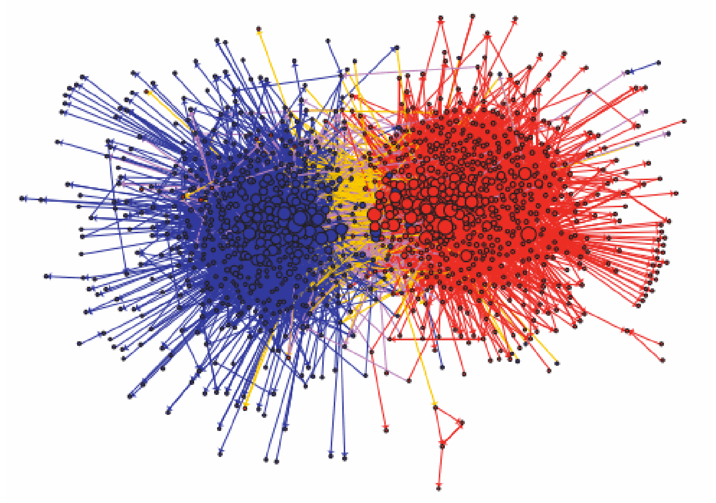


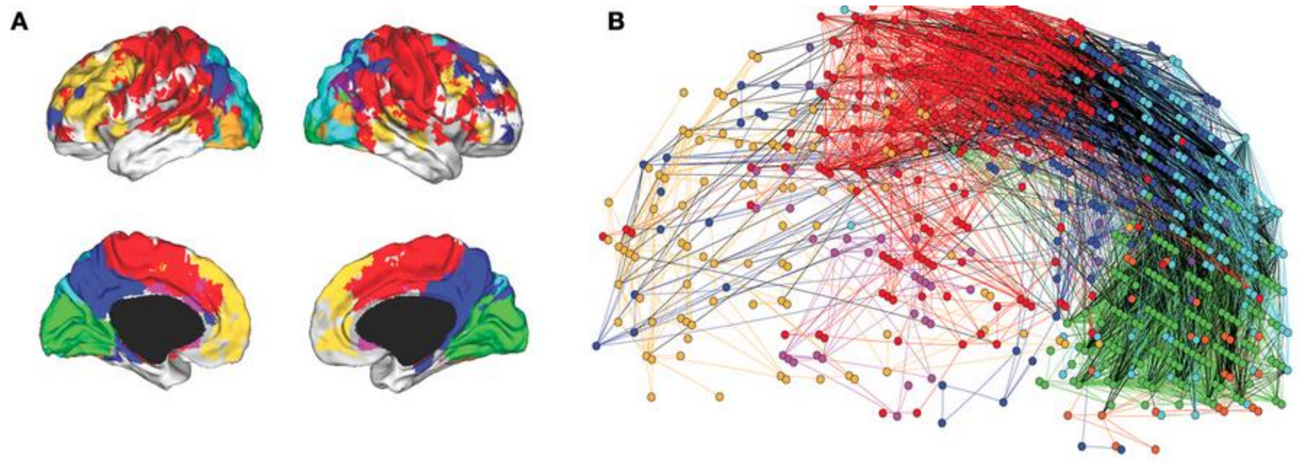
Image by Paresnah, Zachary Karate Club, Wikipedia 2022

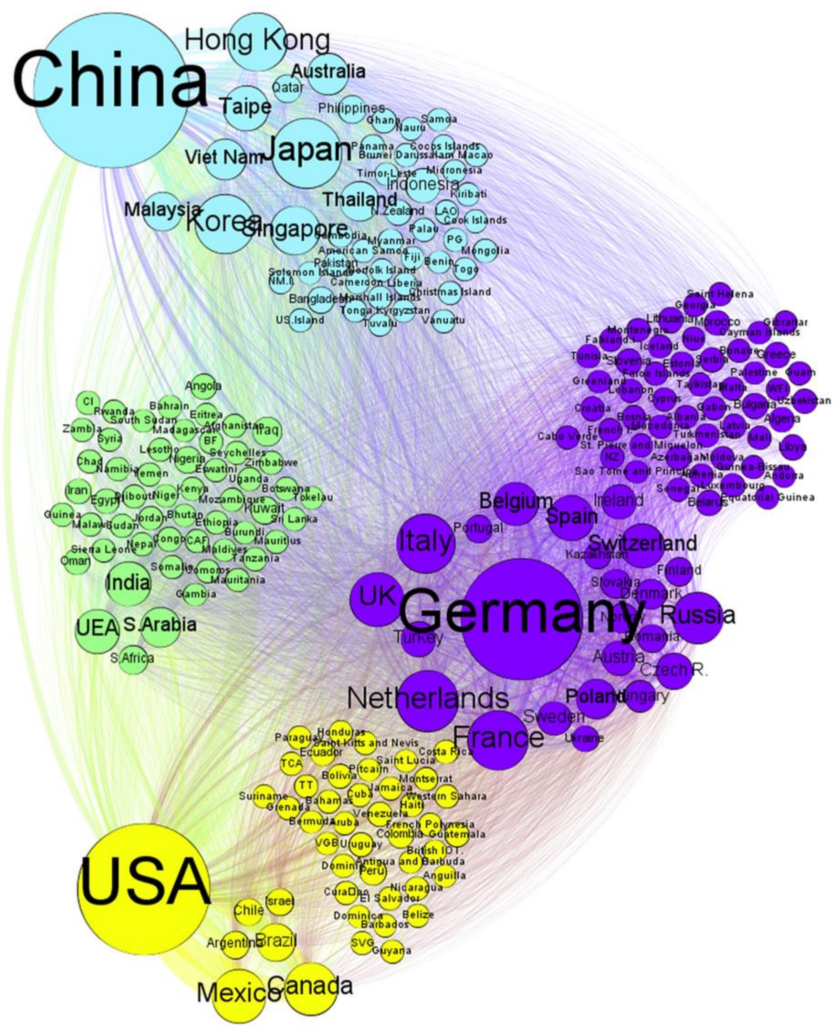
Zachary, W. W. (1977). An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, *33*(4), 452–473. http://www.jstor.org/stable/3629752

The importance of pairwise relationships





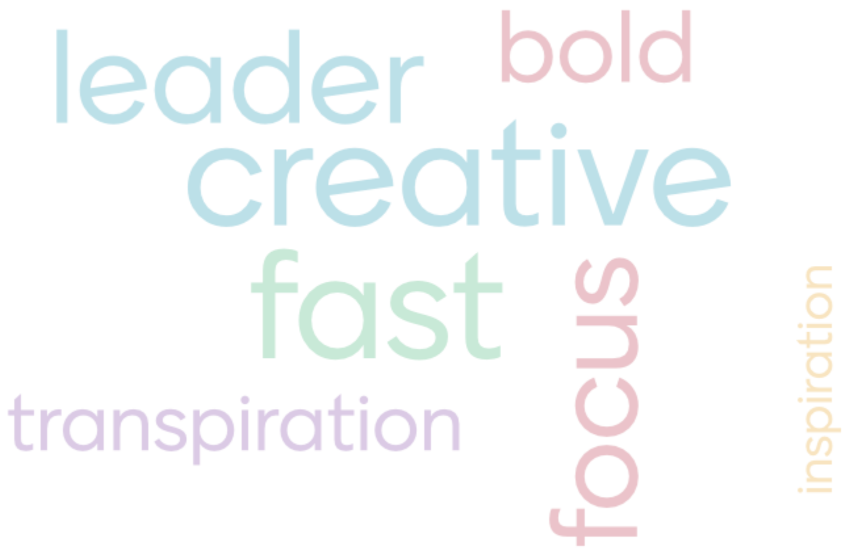




Join at menti.com | use code 3393 8461

Mentimeter

What other systems can the networks help us understand?

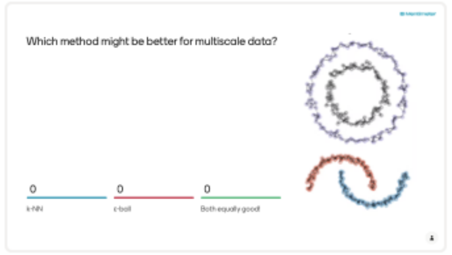


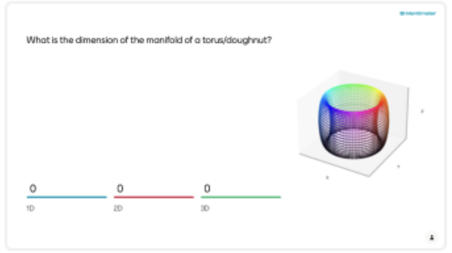
Menti

Graph Based Learning

Choose a slide to present

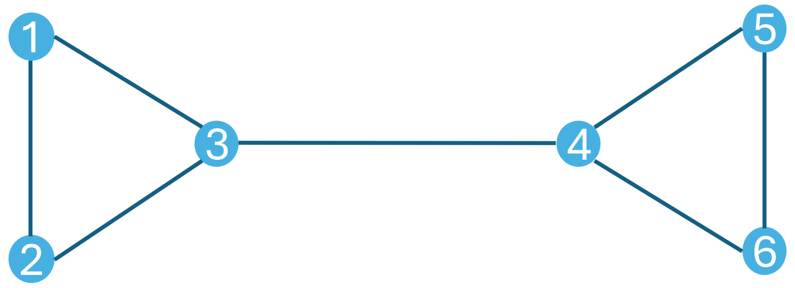
What other systems can the networks help us understand?


Which method might be better for multiscale data?


What is the dimension of the manifold of a torus/doughnut?

Who is the most central character in the Marvel Universe?

What is a Graph?



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

$$Nodes\ V = \{i\}_{i=1}^{N}$$

$$Edges\ E = \{(i,j)\}_{i\sim j}$$

- For undirected graphs , A is symmetric.
- For directed graphs d_{in} and d_{out} are defined separately as,

$$d_{in} = A\mathbf{1} \quad and\ d_{out} = A^T\mathbf{1}$$

For weighted graphs,
$A_{ij} = w_{ij}$, where $w_{ij} \in \mathbb{R}\ \forall\ A_{ij} > 0$

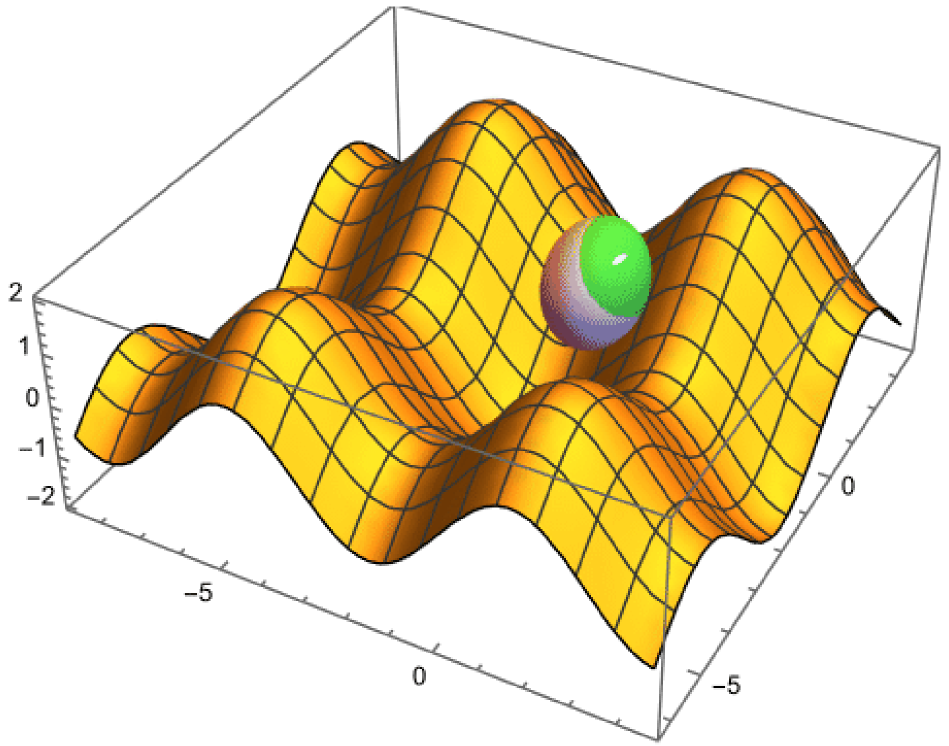
What is a Graph Laplacian?

The Laplacian is a second-order differential operator over space, which measures how a quantity changes at a point relative to its surroundings.

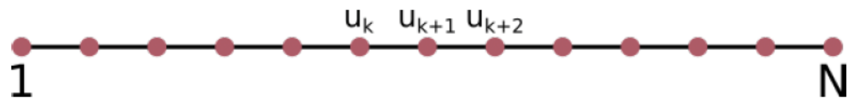
In the diffusion equation, it measures how a quantity disperses in space over time.

$$u = u(x,t), \qquad \frac{\partial u}{\partial t} = \nabla^2 u$$

Laplacian Operator $$\nabla^2 u = \frac{d}{dx}\left(\frac{d}{dx}u\right)$$

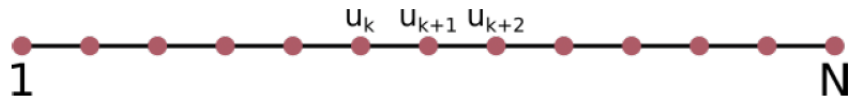


The 1D discrete space can be represented as a graph of nodes along a line



What is a Graph Laplacian?

Now , we can rewrite the diffusion equation in the matrix form on this graph.



$$\frac{d\mathbf{u}}{dt} = \begin{pmatrix} -1 & 1 & & & & & \\ 1 & -2 & 1 & & & 0 & \\ & \ddots & \ddots & \ddots & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & 0 & & & 1 & -2 & 1 \\ & & & & & 1 & -1 \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_N \end{pmatrix}$$

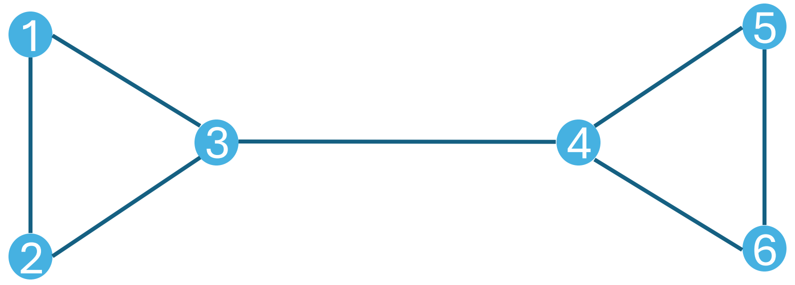
$$A = \begin{pmatrix} 0 & 1 & & \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{pmatrix} \qquad d = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 2 \\ 1 \end{bmatrix}$$

$$\frac{d\boldsymbol{u}}{dt} = -(diag(d) - A)\,\boldsymbol{u} = -\mathcal{L}\boldsymbol{u}$$

$$\mathcal{L} = diag(d) - A = \mathcal{D} - A$$

$\mathcal{L}$ is known as the combinatorial graph Laplacian

What is a Graph Laplacian?



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

$$Nodes\ V = \{i\}_{i=1}^{N}$$

$$Edges\ E = \{(i,j)\}_{i\sim j}$$

$$\mathcal{L} = diag(d) - A = \mathcal{D} - A = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Pairwise relationships in data

We can represent our dataset of N samples as a vector in the p - dimensional space.

$$\{\mathbf{x}^{(i)}\}_{i=1}^{N} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^p \qquad \mathbf{x}^{(i)} = \begin{pmatrix} x_1^{(i)} \\ \vdots \\ x_p^{(i)} \end{pmatrix}$$

$$X_{N\times p} = \begin{pmatrix} \mathbf{x}^{(1)^T} \\ \vdots \\ \mathbf{x}^{(N)^T} \end{pmatrix}$$

In this course, we have explored meaningful metrics that quantify pairwise similarity (or dissimilarity) among data samples.

$$(\textit{cosine similarity}) \qquad S_{ij} = \frac{\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}}{\|\mathbf{x}^{(i)}\| \, \|\mathbf{x}^{(j)}\|}$$

$$(\textit{statistical similarity}) \qquad S_{ij} = \frac{\operatorname{cov}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})}{\sqrt{\operatorname{cov}(\mathbf{x}^{(i)}, \mathbf{x}^{(i)})}\sqrt{\operatorname{cov}(\mathbf{x}^{(j)}, \mathbf{x}^{(j)})}}$$

$$(\textit{Euclidean distance}) \qquad D_{ij} = \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|$$

$$(\textit{kernel matrix}\ ) \qquad D_{ij} = \exp\left[\frac{-\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{c}\right]$$

We can leverage these measures of similarity among data samples to construct a weighted graph of N nodes and edges as the (dis)similarity matrix with adjacency matrices $D_{N\times N}$ or $S_{N\times N}$

Pairwise relationships in data: Quick Aside

In this lecture we will focus on the network of N samples for clustering and dimensionality reduction.

However, in many other applications, the network of descriptors/features is more informative.

Especially when features represent data from different parts of a system.
E.g. Timeseries data from different brain regions.

Statistical interdependence relationships can be quantified amongst the features using Pearson correlation, mutual information, granger causality, etc.

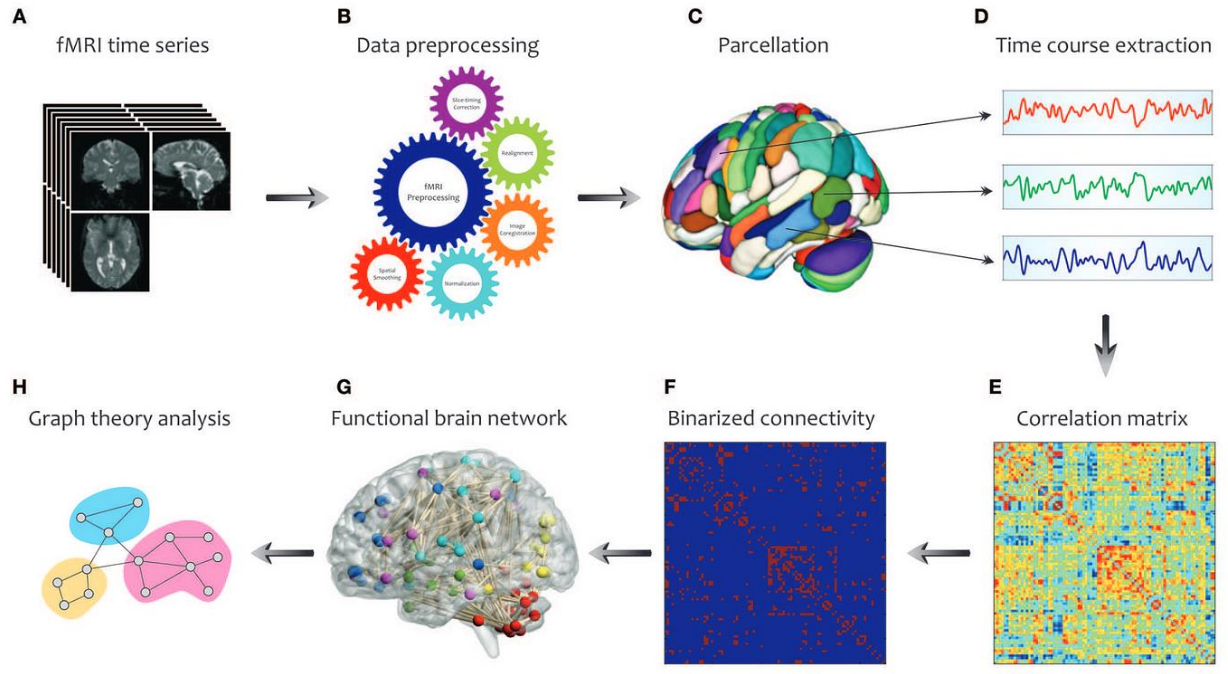
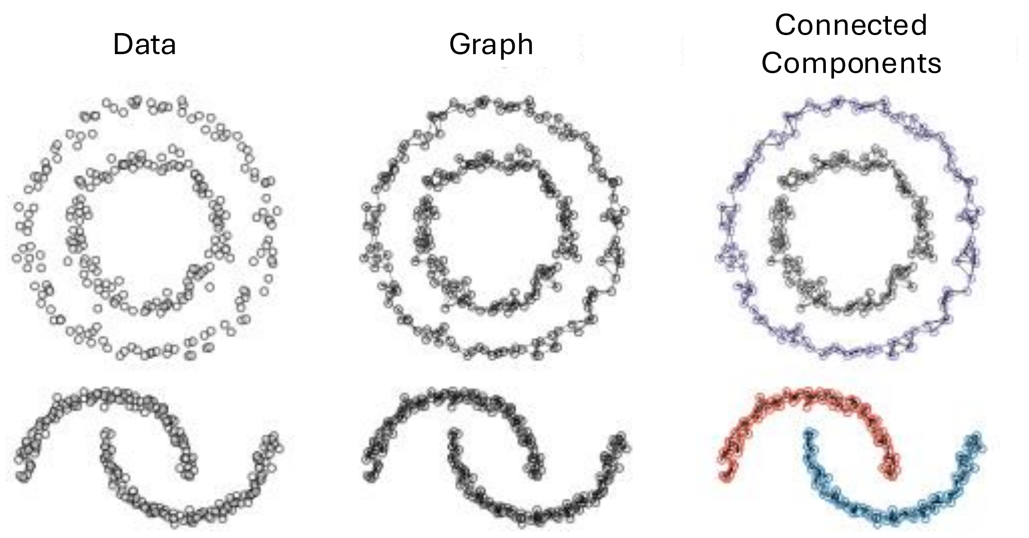


Image from: Touihiri, Mariem. *Decoding Brain Connections: Exploring Neural Function Networks Through fMRI Graph Analysis*, Medium 2023

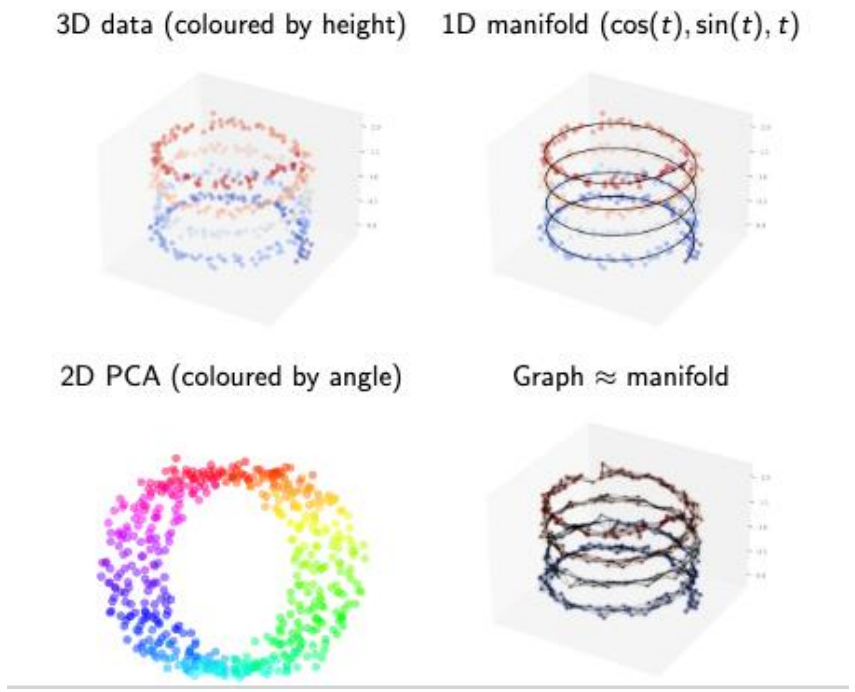
What can graphs tell us?

Clustering



Graphs leverage the intrinsic geometry of the data to facilitate unsupervised learning.

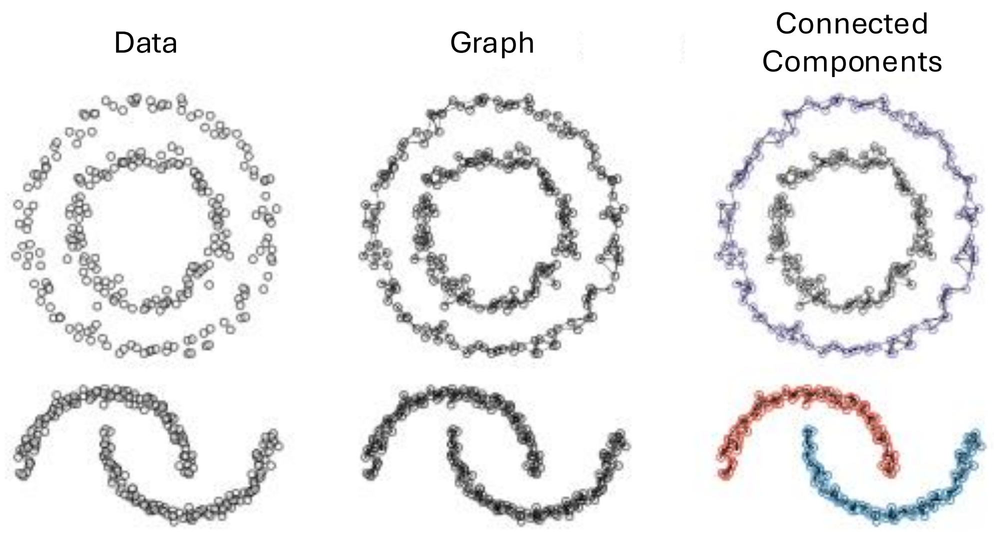
Dimensionality Reduction



Graphs from Data

Although the (dis)similarity matrices $D_{N \times N}$ or $S_{N \times N}$, can be used as the adjacency matrix of the graph.

Their fully connected nature hides the underlying geometry, and graph sparsification methods are needed to reduce the density of the graph. Revealing the underlying geometry.

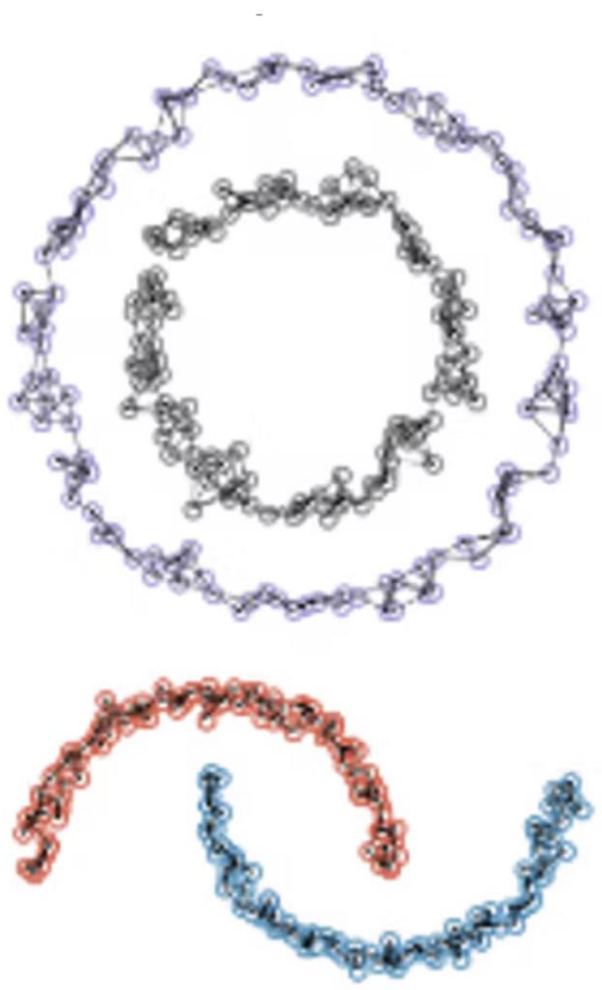


The most common ones are,

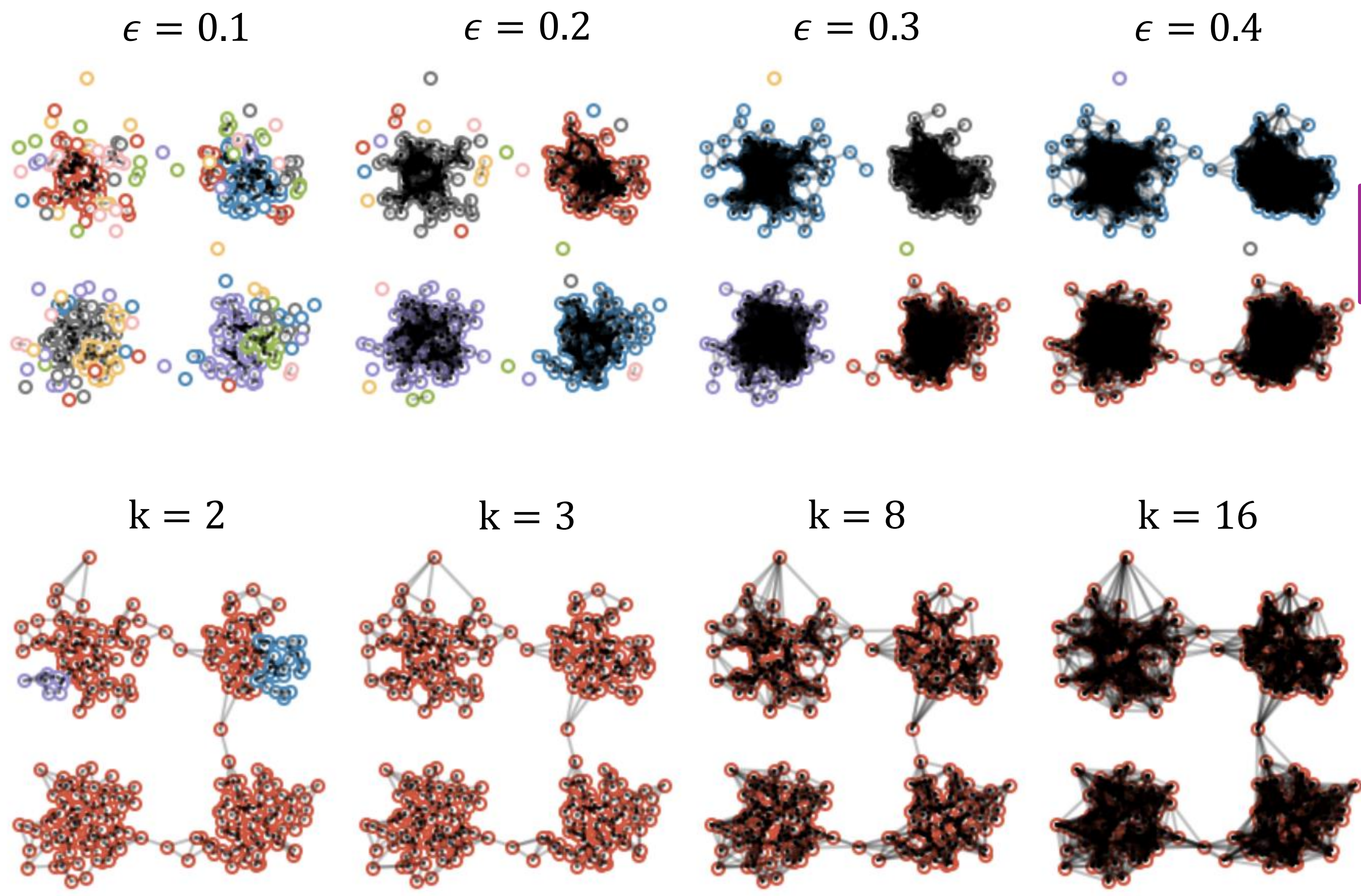
- Thresholding – Removes all edges below a threshold
- ϵ-ball – Nodes within ϵ distance from each other form an edge
- k-NN – The top k nearest neighbours form an edge

Join at menti.com | use code 3393 8461

Mentimeter

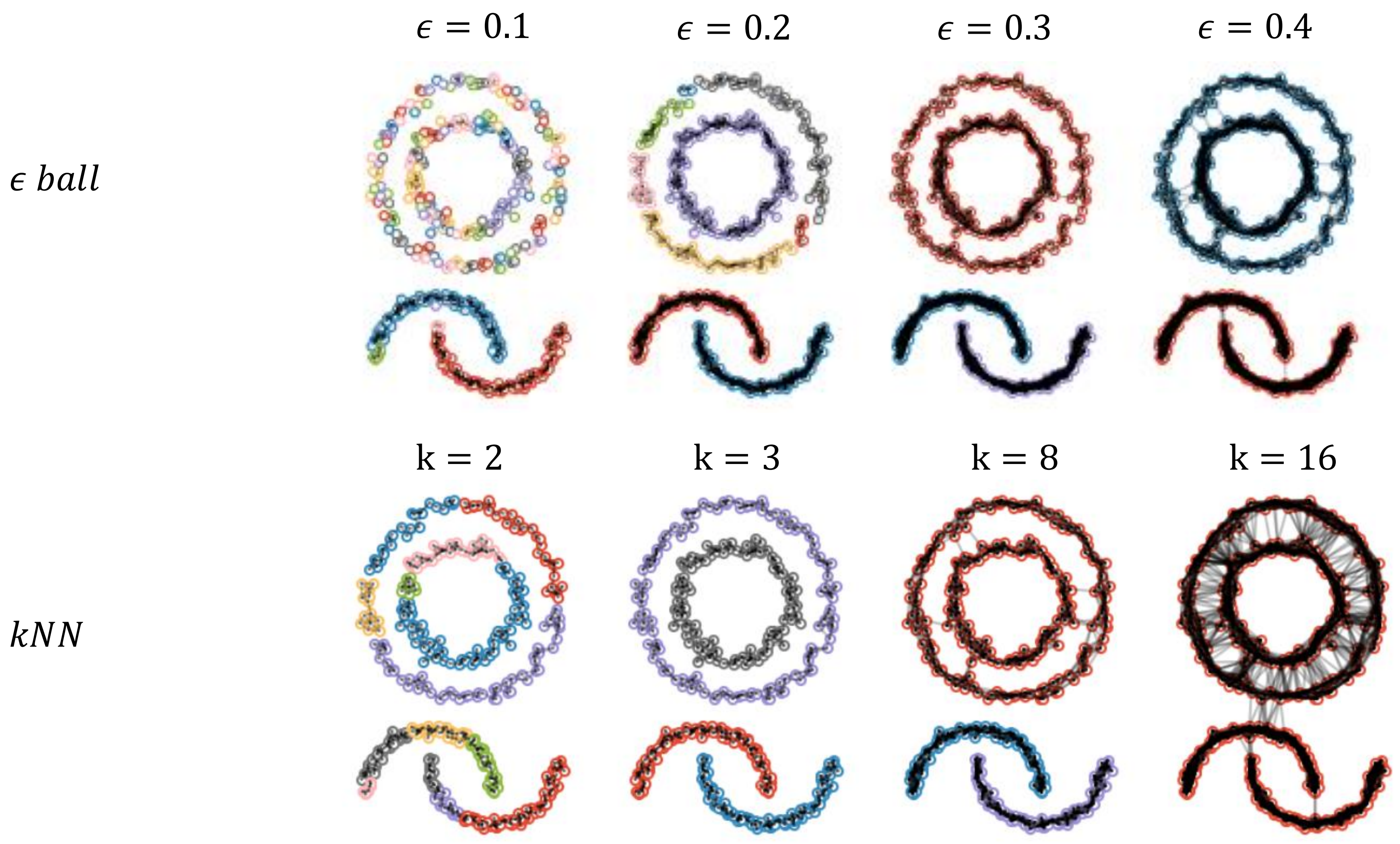


Clustering connected components



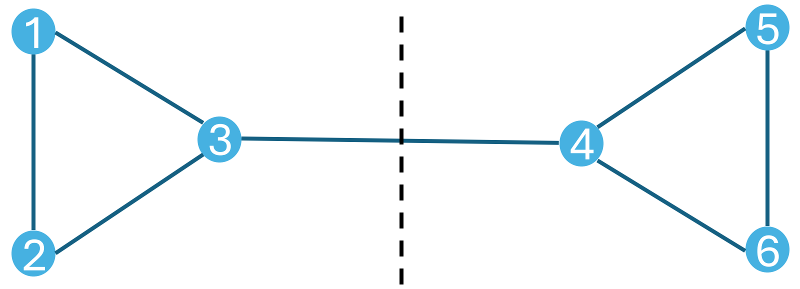
Works when data has a single scale

Clustering connected components



Works when data has a multiple scale

Clustering on (connected) Graphs



Suppose we wish to bipartition this graph into two clusters, S_1 and S_2, **minimizing the number of edges between the two clusters**

$$C = \frac{1}{2} \sum_{i \in S_1, j \in S_2} A_{ij}$$

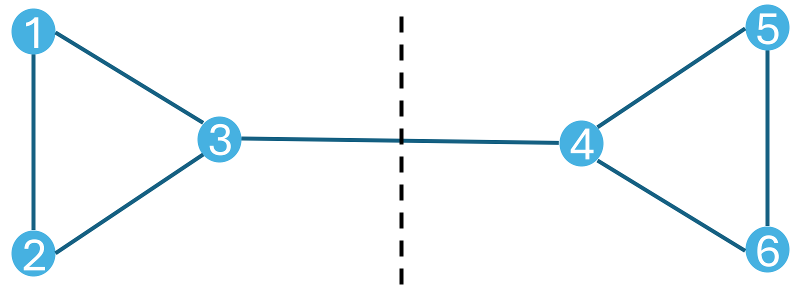
$$s_i = \begin{cases} +1 & \text{if } i \in S_1 \\ -1 & \text{if } i \in S_2 \end{cases} \qquad t_{ij} = \frac{1}{2}(1 - s_i s_j) = \begin{cases} 1 \\ 0 \end{cases}$$

$$C = \frac{1}{4} \sum_{i,j} t_{ij} A_{ij} = \frac{1}{4} \left(\sum_{i,j} A_{ij} - \sum_{i,j} A_{ij} s_i s_j \right)$$

$$\sum_{i,j} A_{ij} = \sum_i d_i = \sum_i d_i s_i^2 = \sum_{i,j} d_i s_i s_j \delta_{ij}$$

$$C = \frac{1}{4} \sum_{i,j} \left(\underbrace{d_i \delta_{ij} - A_{ij}} \right) s_i s_j = \frac{1}{4} \mathbf{s}^T L \mathbf{s}$$

Clustering on (connected) Graphs



$$C = \frac{1}{2} \sum_{i \in S_1, j \in S_2} A_{ij} = \frac{1}{4} \mathbf{s}^T L \mathbf{s}$$

$$\min_{\mathbf{s} \in \{-1,+1\}^N} \mathbf{s}^T L \mathbf{s} \qquad \text{under the constraints} \begin{cases} \mathbf{s}^T \cdot \mathbf{s} & = n_1 + n_2 = N \\ \mathbf{s}^T \cdot \mathbf{1} & = n_1 - n_2 \end{cases}$$

$$\mathcal{C}(\mathbf{s}, \lambda, \mu) = \mathbf{s}^T L \mathbf{s} + \lambda(N - \mathbf{s}^T \cdot \mathbf{s}) + 2\mu\big((n_1 - n_2) - \mathbf{s}^T \cdot \mathbf{1}\big)$$

$$\nabla_{\mathbf{s}} \mathcal{C}\big|_{\mathbf{s}^\star} = 0.$$

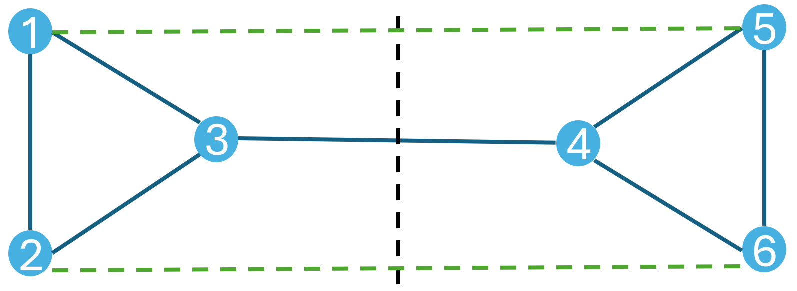
$$\boxed{C^\star_{\min} = \lambda_2 \frac{n_1 n_2}{N}}$$

The lowest value of C^*_{min} is achieved by the eigenvector v_2 of the graph Laplacian with the second smallest or the first non-trivial eigenvalue λ_2

v_2 is also known as the Fiedler vector.

The number of 0 eigenvalues of the Laplacian gives the number of connected components of the graph.

Clustering on (connected) Graphs



$$\mathcal{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \qquad \mathcal{L}' = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & -1 & 3 \end{bmatrix}$$

$v_2 = (-0.465, -0.465, -0.261, 0.261, 0.465, 0.465)$ $\qquad \lambda_2 = 0.438 \qquad \lambda_2 = 2$

$v_2 = (0.408, 0.408, 0.408, -0.408, -0.408, -0.408)$

The sign of each node in the Fiedler vector specifies the partition it belongs to.

λ_2 is also known as the *algebraic connectivity* of the graph. It is a measure of how difficult it is to disconnect the graph.

Beyond Biparitions

- Real networks often have more than 2 clusters.
  Here, we discuss two ways of extending the framework using the graph Laplacian.

- ***Sequential Biparition***
  Iteratively identify all partitions of the system

- ***Spectral Clustering*** using more Laplacian Eigenvectors.

  - ➢ Consider a $N \times m$ matrix of the m eigenvectors corresponding to the smallest m non-trivial eigenvalues of the graph Laplacian.
  - ➢ Each row characterises the N nodes in the m-dimensional eigenspace.
  - ➢ Carry out K-means clustering on the row vectors of the $N \times m$ matrix to find K clusters.

Gene Regulatory Network of E. Coli

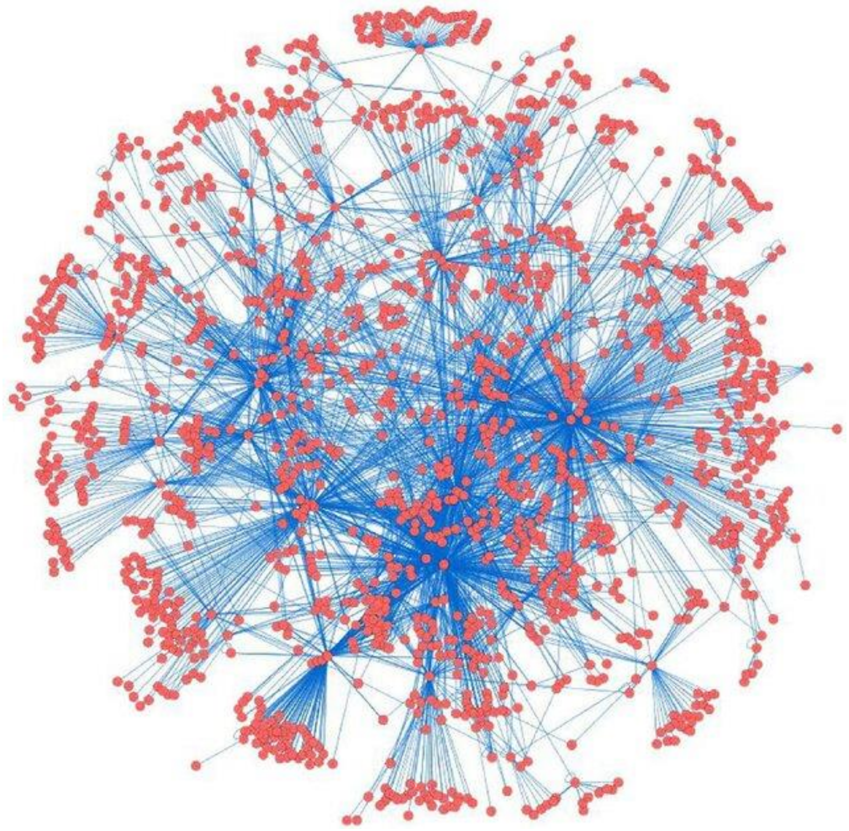
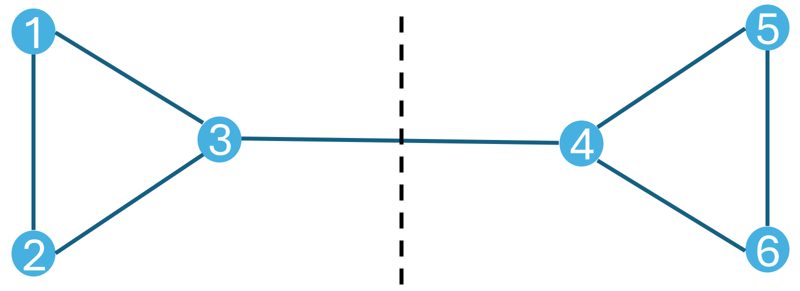


Figure from, Allen, J.D., Xie, Y., Chen, M., Girard, L. and Xiao, G., 2012. Comparing statistical methods for constructing large scale gene networks. *PloS one*

Modularity



$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \frac{1}{2} H^T A H = \begin{bmatrix} 3 & \frac{1}{2} \\ \frac{1}{2} & 3 \end{bmatrix}$$

The modularity algorithm aims to cluster the graph by *labelling nodes, making the adjacency matrix* ***maximally block-diagonal***.

If H is the $N \times k$ membership matrix then,
The total number of edges within the defined clusters can be written as,

$$\frac{1}{2} Tr[H^T A H]$$

Along with maximising $Tr[H^T A H]$ to find the optimal partition, the algorithm also compares this quantity to a null model with the same degree distribution with the adjacency matrix,

$$R = \frac{1}{2E} d d^T$$

Thus, the optimization function for the algorithm can be written as,

$$Q = \frac{1}{2E} Tr\left[H^T \underbrace{\left[A - \frac{1}{2E} d d^T \right]}_{Z} H \right]$$

Modularity

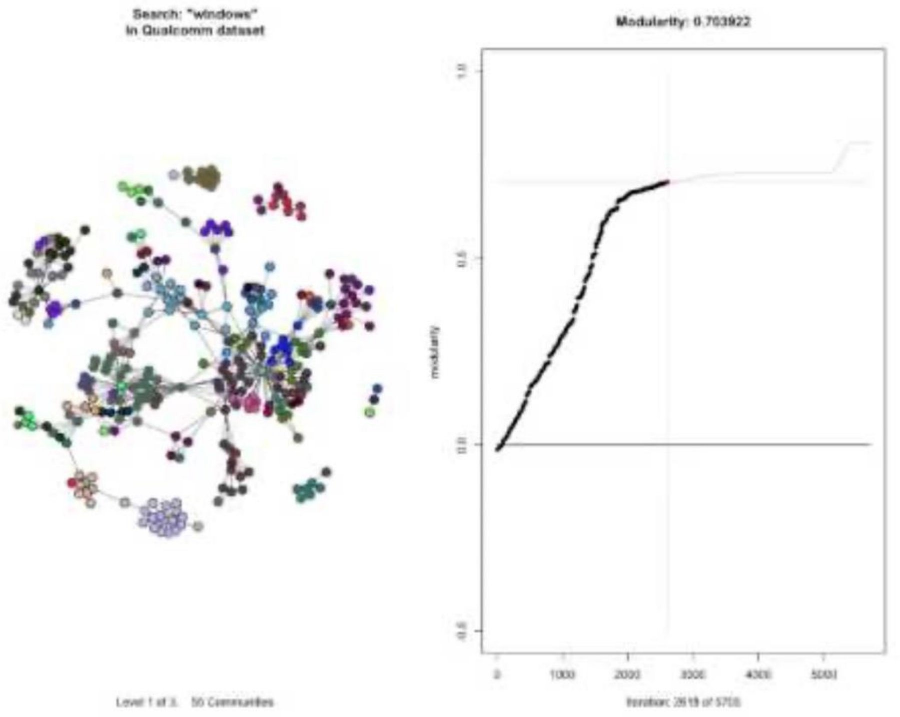
$$Q = \frac{1}{2E} Tr\left[H^T \left[\underbrace{A - \frac{1}{2E} dd^T}_{Z}\right] H\right]$$

The Modularity method aims to **maximise the modularity function $\boldsymbol{Q}$** over all membership matrices H *to find the optimal partition*

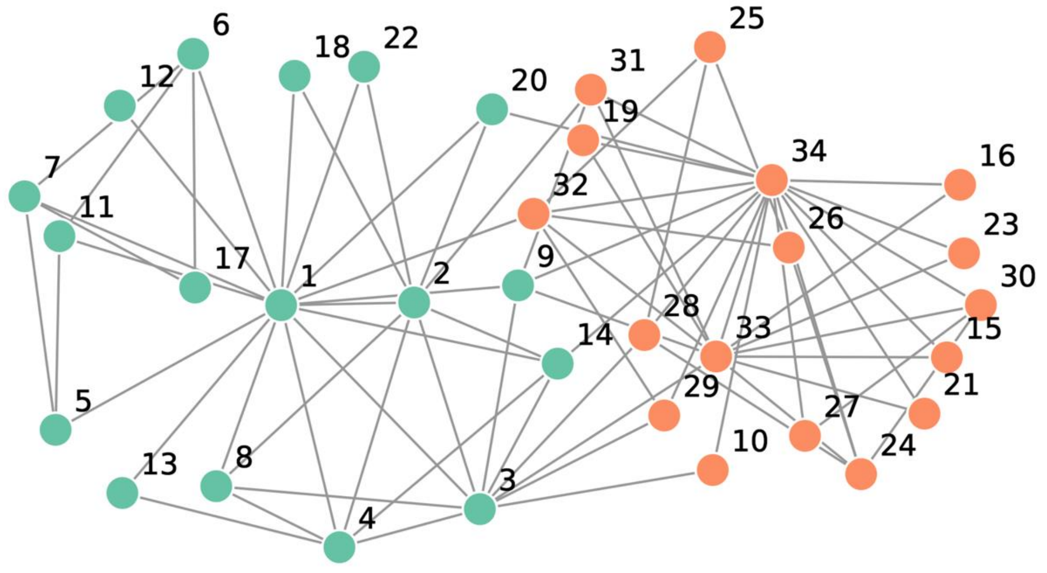
$$\max_H Q$$

Practical algorithms to maximize Q

- Iterative bipartitions of Z using the leading eigenvector until $\Delta Q < 0$
- The *Louvain Algorithm*
  A greedy agglomerative algorithm that adds nodes of the graph to a partition until no improvement of Q is achieved.



Spectral and Modularity Bipartitions



```python
import networkx as nx

zkc = nx.karate_club_graph()
gt_membership = [zkc.nodes[v]['club'] for v in zkc.nodes()]
```

Fun Exercise

Compare the bipartitions obtained using modularity optimization and using the Fiedler vector v_2 to ground truth partition using the Normalised Variation of Information between any learned partition and the ground truth.

Mutual Information:

$$MI(X,Y) = \sum_{i=1}^{K}\sum_{j=1}^{K'} P_{XY}(i,j)\log\frac{P_{XY}(i,j)}{P_X(i)P_Y(j)}$$

Clustering entropy

$$H(X) = -\sum_{i=1}^{K} P_X(i)\log P_X(i)$$

Normalised Variation of Information:

$$NVI(X,Y) = \frac{H(X)+H(Y)-2MI(X,Y)}{H(X)+H(Y)-MI(X,Y)}$$

and others, see lecture notes

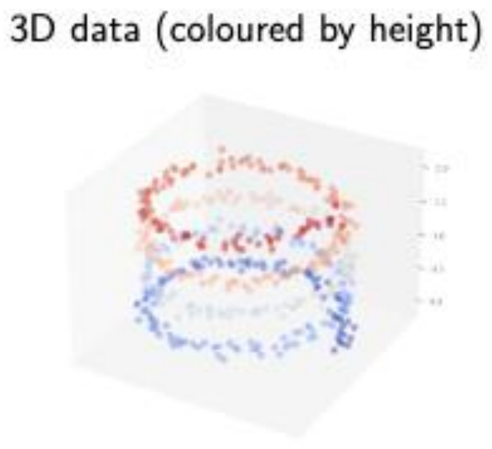
Break?

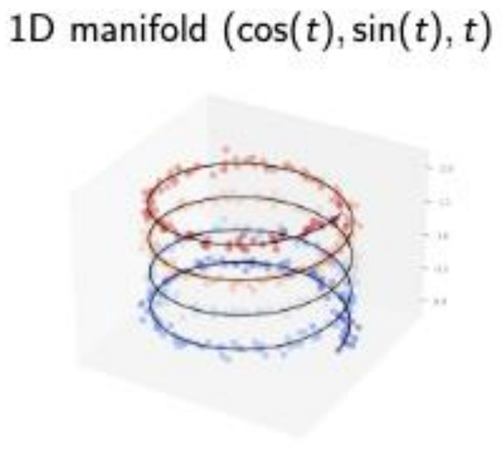
Dimensionality reduction using Graphs

Manifold Assumption: The data lie on some low dimensional manifold of dimensionality m, embedded in a high-dimensional space of dimensionality $p > m$

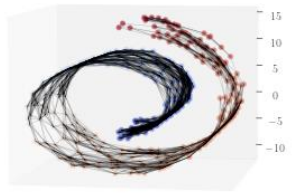
Manifold is a collection of points such that local neighbourhood of each point resembles a Euclidean space of dimension m

Since graphs capture the interactions between a node and its neighbours connected by edges, ***constructing a graph that preserves the local structure*** *of the data can approximate the low dimensional manifold*.

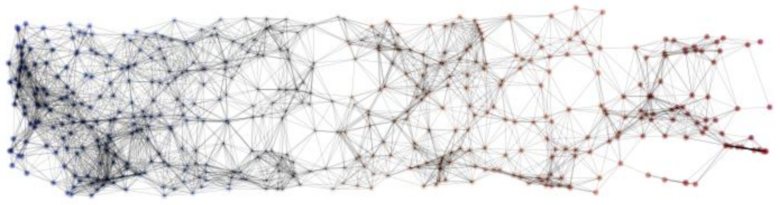




3D data

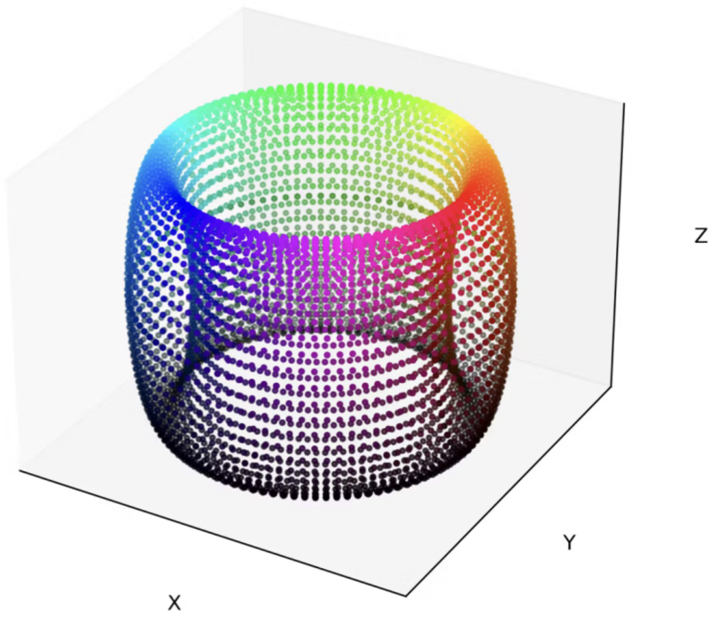


2D Manifold

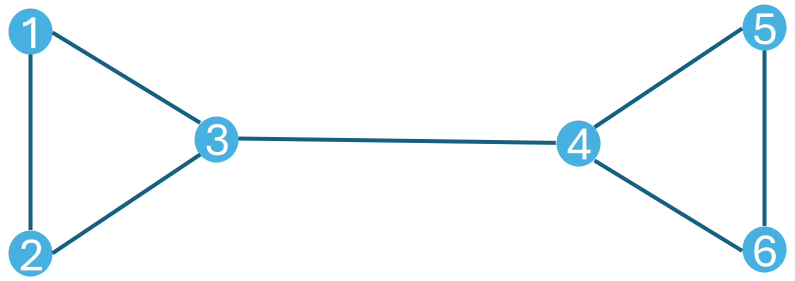


Join at menti.com | use code 3393 8461

Mentimeter



Spectral Embedding



$v_2 = (-0.465, -0.465, -0.261, 0.261, 0.465, 0.465)$

While spectral embedding maps neighbouring nodes to a close distance in the manifold, it doesn't ensure nodes far away in the graph are mapped far away in the cost function.

Preserves the local structure of the data.

Which means if i and j are neighbours in the graph then they should be close together in the mapped lower dimensional space.

$$C := \frac{1}{2} \sum_{i,j} \|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(j)}\|_2^2 A_{ij},$$

$$C = \operatorname{tr}(\widehat{X}^T L \widehat{X}),$$

Solution: the m eigenvectors of the graph Laplacian where eigenvalues are in ascending order
(ignoring the trivial zero eigenvalues)
First shown in *Belkin and Niyogi (2002)*

Isomap

Preserves the global structure of the data.

Which means distance between i and j in the graph should be similar to the distance between them in the mapped lower dimensional space.

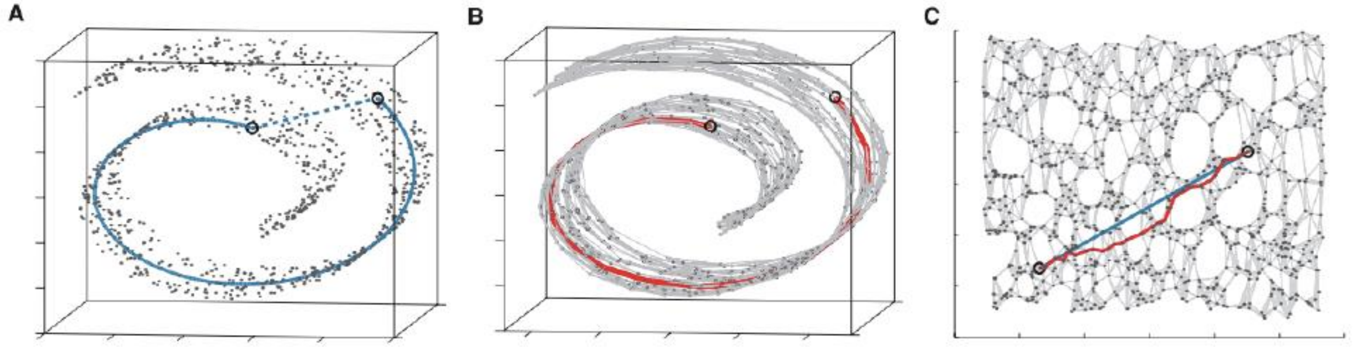
$$C := \sum_{i,j} \left[\|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(j)}\|_2^2 - d(i,j)^2 \right]^2$$

Where $d(i,j)$ is the geodesic distance or the shortest path length on the graph.

This transformation is equivalent to a kernel PCA where $K_{ij} = -\frac{1}{2} d(i,j)^2$
Tenebaum, de Silva & Langford (2000)

The Isomap algorithm can be summarised as follows,

- Construct a (weighted) graph using k-NN or ϵ-ball, with edge weights as Euclidean distances.
- Compute all pairwise geodesic distances using graph algorithms like Dijkstra's algorithm (for positive edge weights).
- Apply multidimensional scaling (MDS) to the matrix $D_{N\times N}$ of geodesic distances.



Graph Centrality Measures

Graph centrality measures rank nodes based on their level of importance.

Different measures capture different perspectives on what it means to be “important”.

Degree Centrality: Total degree of the node normalised by total number of edges.

$$c_d = \frac{d}{2E}$$

Betweenness Centrality: Total number of shortest paths that pass through a node, normalised by the total number of shortest paths.

Eigenvector Centrality: Nodes that are themselves connected to highly central nodes. The components of the leading eigenvector of the adjacency matrix define the eigenvector centrality of each node.

Page Rank: The algorithm assigns higher centrality to nodes where most random walkers on the graph are likely to end up.

$$\mathbf{c}_{\mathrm{PR}\ t+1} = \alpha \left(A\mathcal{D}^{-1}\right)\ \mathbf{c}_{\mathrm{PR}\ t} + (1-\alpha)\,\frac{1}{N}\mathbf{1}$$

$$\mathbf{c}_{\mathrm{PR}} = \alpha \left(A\mathcal{D}^{-1}\right)\ \mathbf{c}_{\mathrm{PR}} + (1-\alpha)\,\frac{1}{N}\mathbf{1}, \quad \alpha \in [0,1]$$

Join at menti.com | use code 3393 8461

Mentimeter

Who is the most central character in the Marvel Universe?

0 responses

Marvel Universe Social Graph

The top ten according to,

Degree	**Betweenness**	**Eigenvector**	**PageRank**
‘CAPTAIN AMERICA’,	'SPIDER-MAN/PETER PAR’,	‘CAPTAIN AMERICA’,	‘CAPTAIN AMERICA’,
'SPIDER-MAN/PETER PAR’,	‘CAPTAIN AMERICA’,	'IRON MAN/TONY STARK ‘,	'SPIDER-MAN/PETER PAR’,
'IRON MAN/TONY STARK ‘,	'HAVOK/ALEX SUMMERS ‘,	'SCARLET WITCH/WANDA ‘,	'IRON MAN/TONY STARK ‘,
'THING/BENJAMIN J. GR’,	'IRON MAN/TONY STARK ‘,	'THING/BENJAMIN J. GR’,	'THING/BENJAMIN J. GR’,
'MR. FANTASTIC/REED R’,	'WOLVERINE/LOGAN ‘,	'SPIDER-MAN/PETER PAR’,	'MR. FANTASTIC/REED R’,
'HUMAN TORCH/JOHNNY S’,	'THING/BENJAMIN J. GR’,	'MR. FANTASTIC/REED R’,	'HUMAN TORCH/JOHNNY S’,
'SCARLET WITCH/WANDA ‘,	'MR. FANTASTIC/REED R’,	'VISION '	'WOLVERINE/LOGAN ‘,
'WOLVERINE/LOGAN ‘,	'THOR/DR. DONALD BLAK’,	'HUMAN TORCH/JOHNNY S’,	'SCARLET WITCH/WANDA ‘,
'THOR/DR. DONALD BLAK’,	'HAWK’,	'WOLVERINE/LOGAN ‘,	'THOR/DR. DONALD BLAK’,
'VISION '	'HUMAN TORCH/JOHNNY S’,	'BEAST/HENRY &HANK& P'	'INVISIBLE WOMAN/SUE '

Hero Social Network Data (csv)

6426 Nodes
167207 Edges

Key Takeaways – Graph Based Learning

- Graphs capture pairwise relationships and the underlying geometry of the data to facilitate unsupervised learning.
- Constructing graphs from data using (dis)similarity matrices among samples along with graph sparsification methods like k-NN and ϵ-ball.
- Graph Laplacian plays an important role in both clustering and dimensionality reduction of the data.
- Graphs help uncover the low-dimensional manifold, enabling non-linear dimensionality reduction that preserves the data's local (Spectral Embedding) and global (Isomap) structure.
- Centrality measures provide a way to rank the nodes based on different measures of importances.