Algebraic Graph Analysis summary

N. Kilis, Prof. Ioannis Pitas,
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 3.5.1

Artificial Intelligence & Information Analysis Lab

Outline

- Why choose Graphs
- Graph Basics
- Graph Matrix Representations
- Graph-shift Operator (GSO)
- Eigen-decomposition of GSO
- Graph Building Blocks
- Community detection
- Graph Clustering
  - Spatial domain
  - Spectral domain

Why choose Graphs

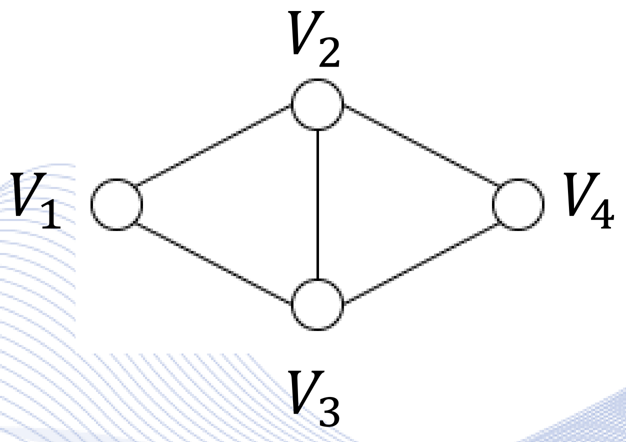
- ***Graphs***:
  - represent a data structure,
  - are more than data structures,
  - in several applications are an inherent part of the system,
  - are models of physical systems with multiple agents:
    - Decentralized Control of Autonomous Systems,
    - Wireless Communication Networks.
  - are usually the source of the problem.

- The challenge is that ***goals are global*** whereas ***information is local***.

Graph Basics

Graph definition: $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$

- $\mathcal{V}$: set of nodes,
- $\mathcal{E}$: set of edges,
- $\mathcal{W}$: set of edge weights.
- N: number of nodes
- E: number of edges



Graph types:

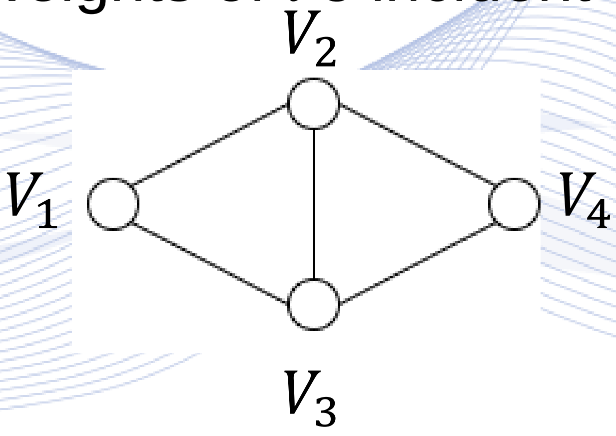
- Directed / Undirected or Symmetric,
- Weighted / Unweighted.

Graph Basics

- ***Neighborhood*** $\mathcal{N}_i$ of node $i = 1, \dots, N$, is the set of nodes j that are connected to i:

$$\mathcal{N}_i = \{j: (i, j) \in \mathcal{E}\}$$

- ***Degree*** of node i: sum of weights of i's incident edges.



Graph Matrix Representations

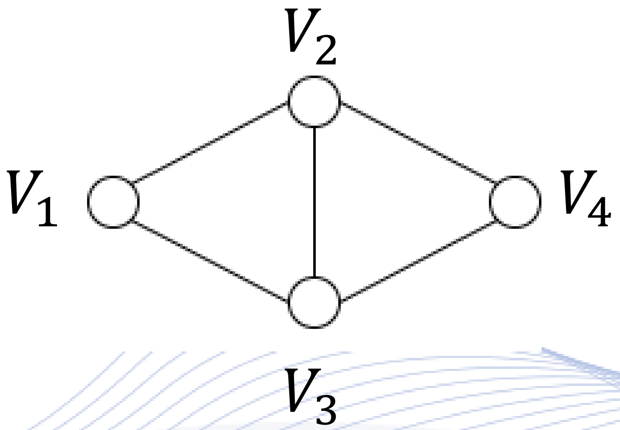
Linear algebra graph descriptors:

- $\mathbf{D} \in \mathbb{R}^{N \times N}$: ***Degree matrix***, describes the #edges connected to each node.
- $\mathbf{A} \in \mathbb{R}^{N \times N}$: ***Adjacency matrix***, describes the connectivity of the graph.
- $\mathbf{L} \in \mathbb{R}^{N \times N}$: ***Laplacian matrix***, of a (sub-)graph consisting of N nodes:

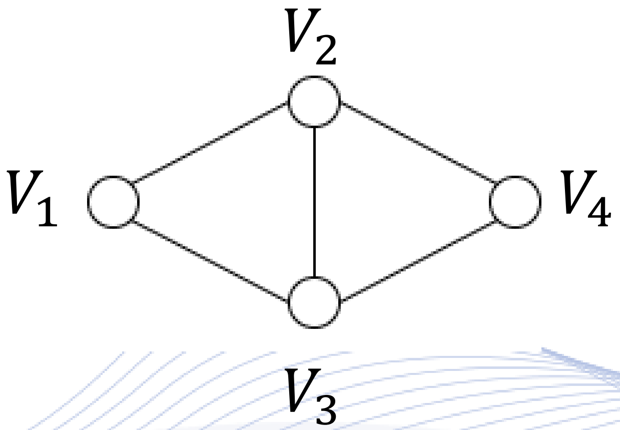
$$\mathbf{L} = \mathbf{D} - \mathbf{A}.$$

Artificial Intelligence & Information Analysis Lab

Graph Matrix Representations

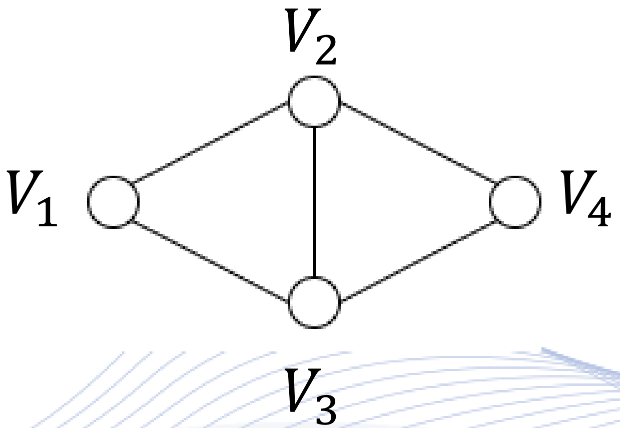


Graph Matrix Representations



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

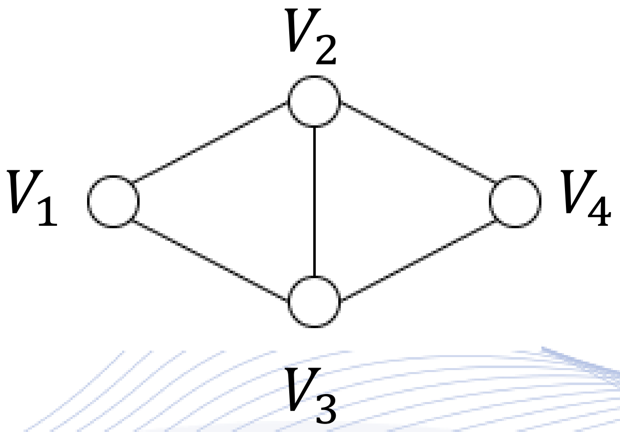
Graph Matrix Representations



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

$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Graph Matrix Representations



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

$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$\mathbf{L} = \mathbf{D} - \mathbf{A} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

Graph Matrix Representations

Graph-Shift Operator (***GSO***):

$$\boldsymbol{S} \in \mathbb{R}^{N \times N}, \qquad S_{ij} \neq 0 \text{ if } i = j \text{ and/or } (i, j) \in \mathcal{E}.$$

- It enables matrix representations of graphs.
- It captures the local graph structure.
- If the graph is symmetric, $\boldsymbol{S}$ is also symmetric.

Graph Matrix Representations

- Various algebraic choices of $\boldsymbol{S}$:
  - Adjacency matrix: $\mathbf{S} = \mathbf{A}$,
  - Graph Laplacian matrix (Directed Graphs):

$$\mathbf{S} = \mathbf{L}_{in} = \mathbf{D}_{in} - \mathbf{A}, \qquad \mathbf{S} = \mathbf{L}_{out} = \mathbf{D}_{out} - \mathbf{A}$$

$$[\mathbf{D}_{in}]_{ii} = \sum_{j=1}^{N} \mathbf{A}_{ji}, \qquad [\mathbf{D}_{out}]_{ii} = \sum_{j=1}^{N} \mathbf{A}_{ij}$$

  - Symmetric Graph Laplacian (Undirected Graphs):

$$\mathbf{S} = \mathbf{L} = \mathbf{D} - \mathbf{A}, \qquad \mathbf{D} = \mathbf{D}_{in} = \mathbf{D}_{out}$$

- The choice matters in practice, however ***the analysis results hold for any selection***.

Graph Fourier-like Basis

Eigen-decomposition of GSO:

$$\boldsymbol{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T,$$

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N] \in \mathbb{R}^{N \times N},$$

$$\boldsymbol{\Lambda} = diag(\lambda_1, \dots, \lambda_N) \in \mathbb{R}^{N \times N}.$$

- Holds for Adjacency and Graph Laplacian matrices.
- Holds for undirected graphs (real-valued $\mathbf{U}$ and $\boldsymbol{\Lambda}$).
- Holds for directed graphs, if $\boldsymbol{S}$ normal ($\mathbf{U}$ and $\boldsymbol{\Lambda}$ complex conjugate pairs).

Graph Fourier-like Basis

Eigen-decomposition of GSO:

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T,$$

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N] \in \mathbb{R}^{N \times N},$$

$$\mathbf{\Lambda} = diag(\lambda_1, \dots, \lambda_N) \in \mathbb{R}^{N \times N}.$$

- Eigen-pair system $\{(\lambda_i, \mathbf{u}_i)\}$, for $i = 1,2, \dots, N$: Fourier-like interpretation.
- $\mathbf{u}_1, \dots, \mathbf{u}_N \in \mathbb{R}^N$: Eigenvectors $\rightarrow$ Graph Fourier modes.
- $\lambda_1, \dots, \lambda_N \in \mathbb{R}^N$: Eigenvalues $\rightarrow$ Graph Spectral Frequencies.

Building Blocks of Graphs

- ***Motifs***:

  - Appear more frequently that random, as small induced overlapping subgraphs,

  - Characterize the whole network structure.

Building Blocks of Graphs

- ***Motifs***: Random Graph $\mathcal{G}'$ with a given degree sequence
  - Appear more frequently that ***random***, as small induced overlapping subgraphs,
  - Characterize the whole network structure.

Building Blocks of Graphs

- ***Motifs***:

Take all the edges between the nodes

  - Appear more frequently that random, as small ***induced*** overlapping subgraphs,
  - Characterize the whole network structure.

Building Blocks of Graphs

- ***Motifs***:
  - Appear more frequently that random, as small induced overlapping subgraphs,
  - Characterize the ***whole network structure***.

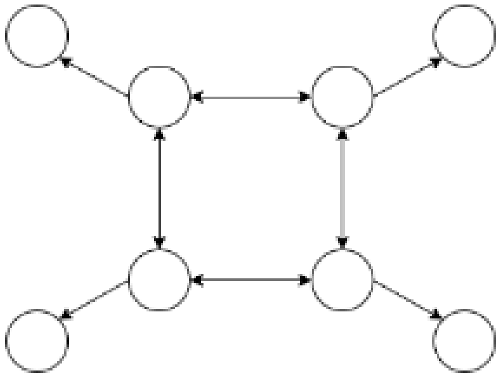
how it works

how it will react

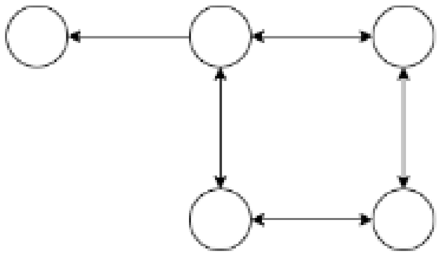
Artificial Intelligence & Information Analysis Lab

Building Blocks of Graphs

- Graph:



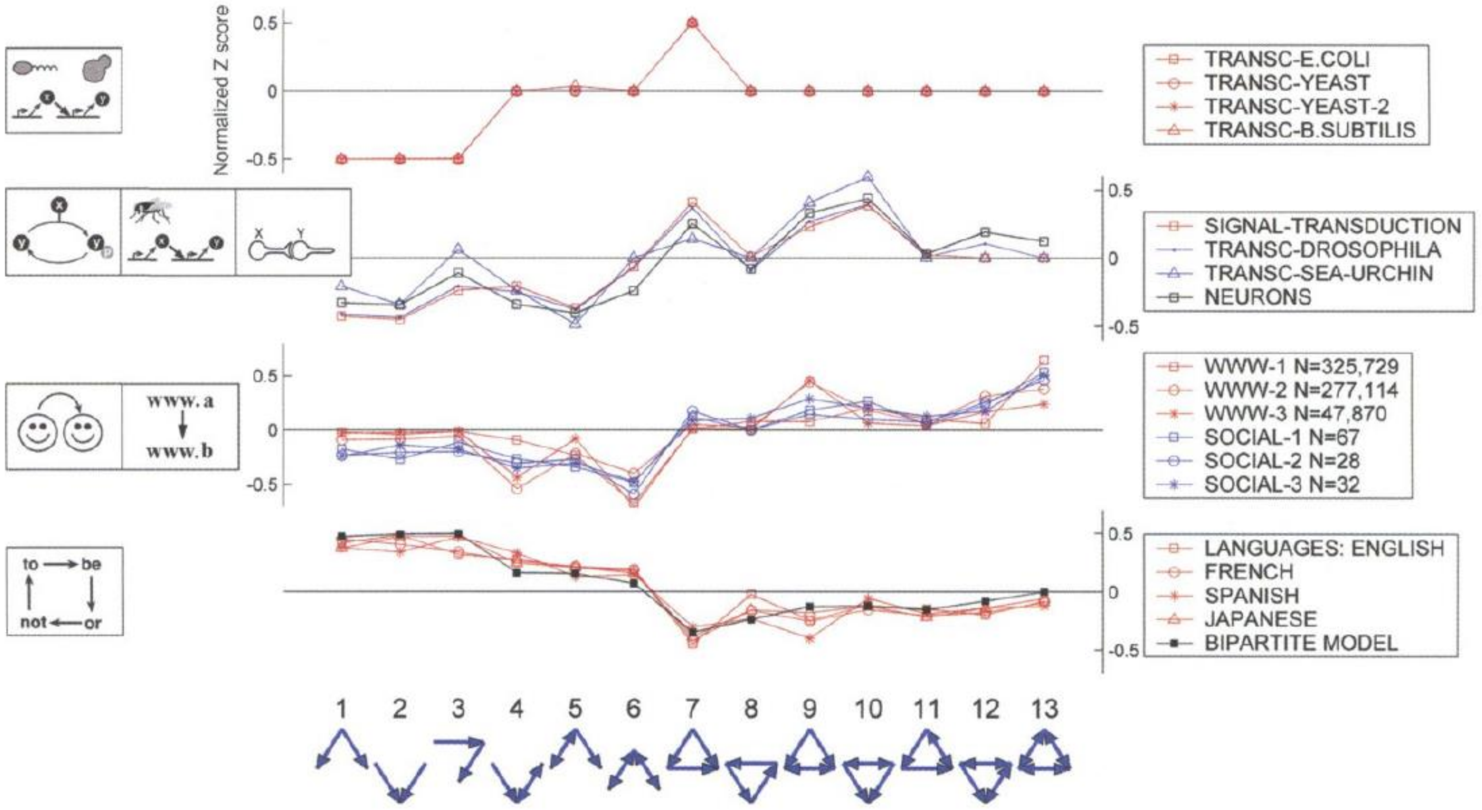
- Motif of interest:



- We observe 4 occurrences of this motif.

Building Blocks of Graphs

- ***Network Significance Profile*** (***SP***):



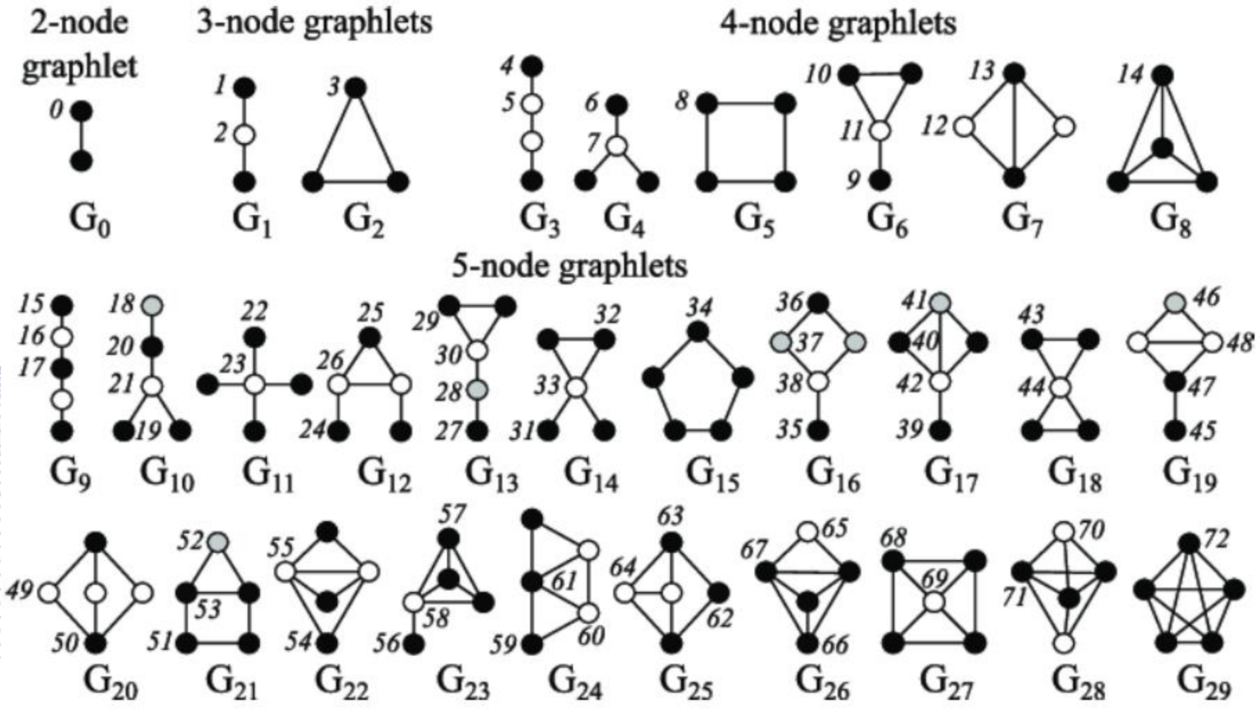
- Image source [LIN2008].

Building Blocks of Graphs

- ***Graphlets*** (generalization of motifs):
  - Connected non-isomorphic subgraphs rooted at any node.
  - Characterize network structure around a node (***neighborhood***).

Building Blocks of Graphs

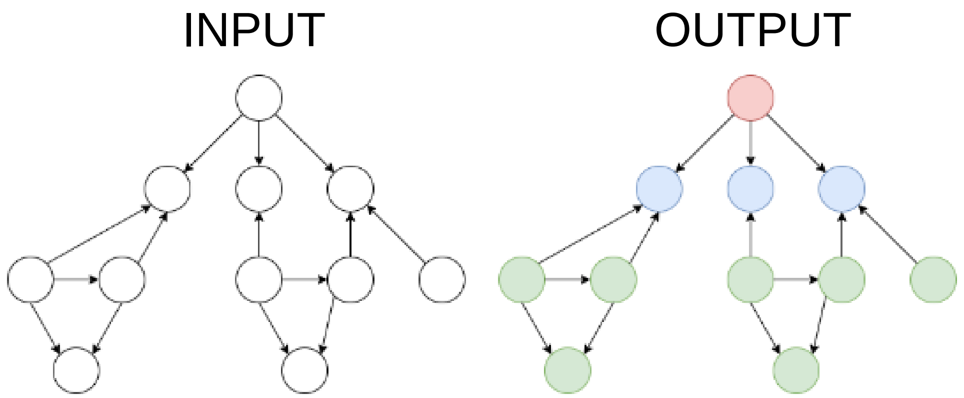
- ***Graphlets***:
- For $N = 3,4,5, \dots, 10$ there are $2,6,21, \dots, 11716571$ graphlets.
- Induced subgraphs of any frequency:

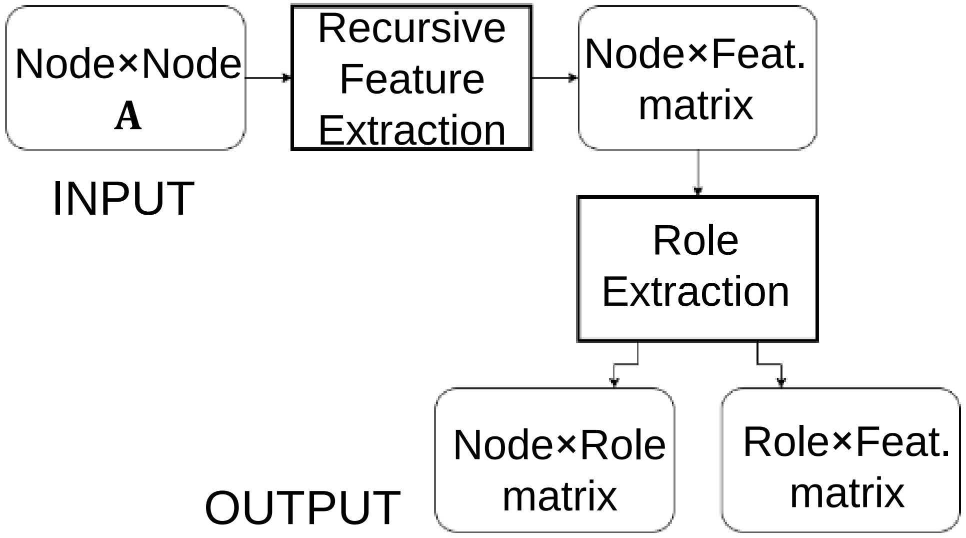


- Image source [PRZULJ2004].

Building Blocks of Graphs

- Automatic discovery of Roles:
  - ***RolX*** algorithm [HEND2012]
    - Unsupervised learning.
    - No prior knowledge.
    - Mixed-membership of roles to each node:
      - Role discovery,
      - How to assign nodes to those roles.
    - Scales linearly (#edges).





Community detection

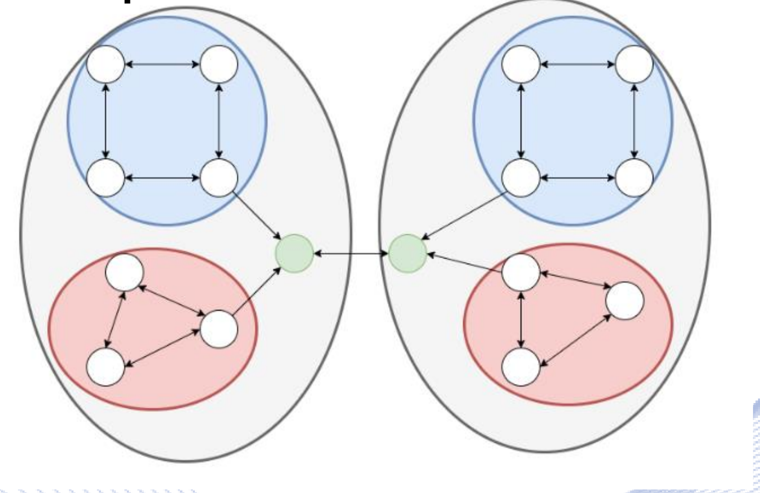
- ***Communities***:
  - Group of nodes with many internal connections and a few external ones.

- ***Modularity Q***:
  - Metric of how well-partitioned into communities a network is.

Community detection

- Discover Communities by maximizing Modularity:
  - ***Louvain*** algorithm:
    - Greedy algorithm.
    - $O(n\log n)$ run time – fast convergence.
    - Supports weighted Graphs.
    - Provides hierarchical Communities.
    - High Modularity output.

Graph and Communities



Dendrogram

Artificial Intelligence & Information Analysis Lab

Community detection

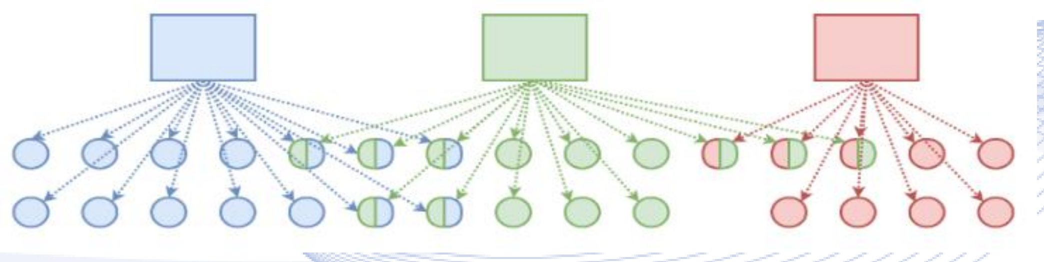
- Discover overlapping Communities:
  - ***Community Affiliation Graph Model*** (***AGM***):
    - Assume that the real Graph $\mathcal{G}$ is generated by AGM.
    - Fit model parameters that generate $\mathcal{G}$.
    - The parameters will reveal which nodes belong to which Communities.

Artificial Intelligence &
Information Analysis Lab

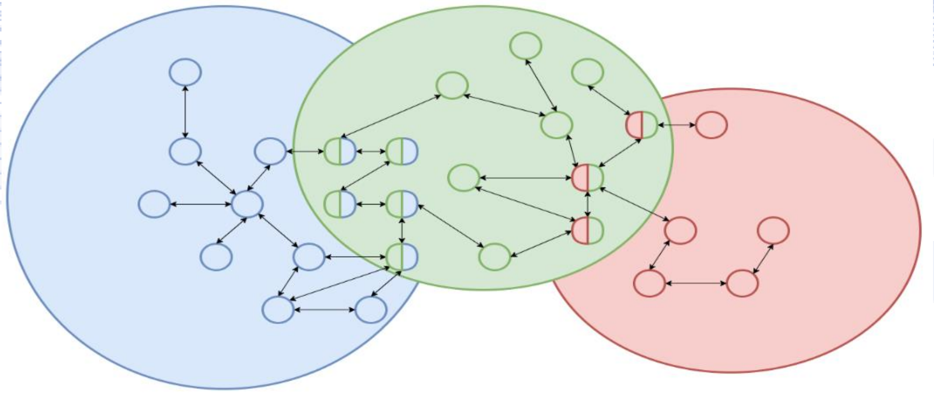
Community detection

- Discover overlapping Communities:
  - ***Community Affiliation Graph Model*** (***AGM***):
    - Assume that the real Graph $\mathcal{G}$ is generated by AGM.
    - Fit model parameters that generate $\mathcal{G}$.
    - The parameters will reveal which nodes belong to which Communities.

Model

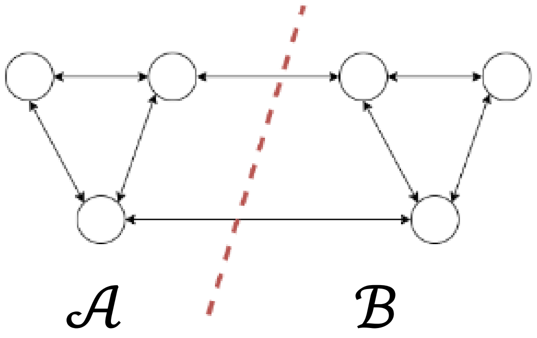


Graph



Graph Clustering

- Graph Partitioning:
  - Modularity → random model comparison (physics view of networks).
  - Conductance → optimization (computer science view of networks).
    - Approximation guarantees on how well the methods work.
  - Goal:
1. Maximize the # within-group connections,
2. Minimize the # between-group connections.

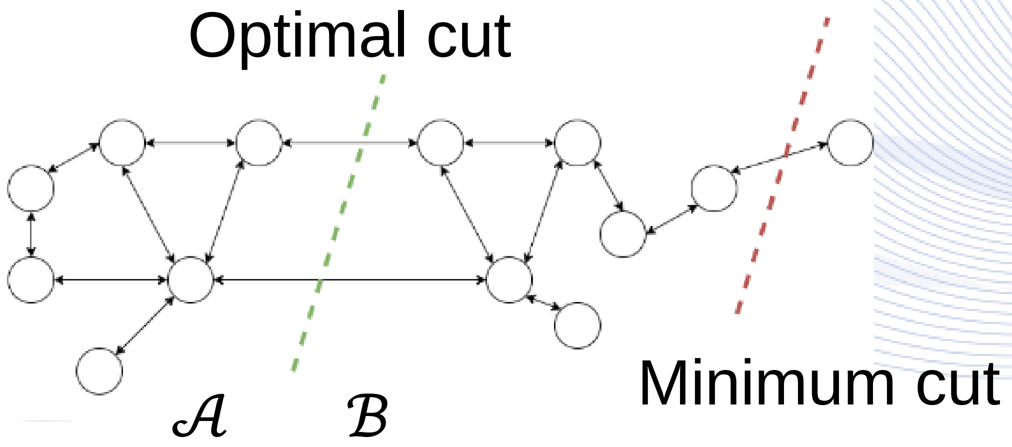


Graph Clustering – Spatial domain

- Notion of a ***Cut***:

$$CUT(\mathcal{A},\mathcal{B}) = \sum_{i\in\mathcal{A},j\in\mathcal{B}} W_{ij}$$

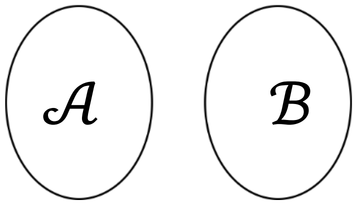
- $\mathcal{A} \subset \mathcal{V}, \mathcal{B} \subset \mathcal{V}$: subsets of the graph node set $\mathcal{V}$.
- Danger of finding the minimum and not the optimal cut:



Graph Clustering – Spectral domain

- D-regular Graphs:
  - Graph with multiple connected components:
    - Multiplicity of the largest eigenvalue (how many different eigenvectors correspond to that eigenvalue) reveals how many connected components there are.

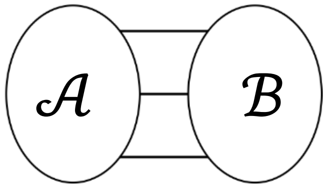
- Disconnected Graph of two components:
  - largest eigenvalue = second largest eigenvalue.



$$\lambda_N - \lambda_{N-1} = 0$$

Graph Clustering – Spectral domain

- D-regular Graphs:
  - Almost disconnected Graph:
    - largest eigenvalue ≈ second largest eigenvalue.
    - Second largest eigenvalue ⇒ what node should be in what Graph component:
      - Largest eigenvector: $\mathbf{u}_N = [1, \dots, 1]^T$.
      - Orthogonality constraint ⇒ Second eigenvector's ($\mathbf{u}_{N-1}$) components must sum to 0.
      - $\mathbf{u}_{N-1}$ splits the nodes into two groups (some values positive, some negative).



$$\lambda_N - \lambda_{N-1} \approx 0$$

Artificial Intelligence & Information Analysis Lab

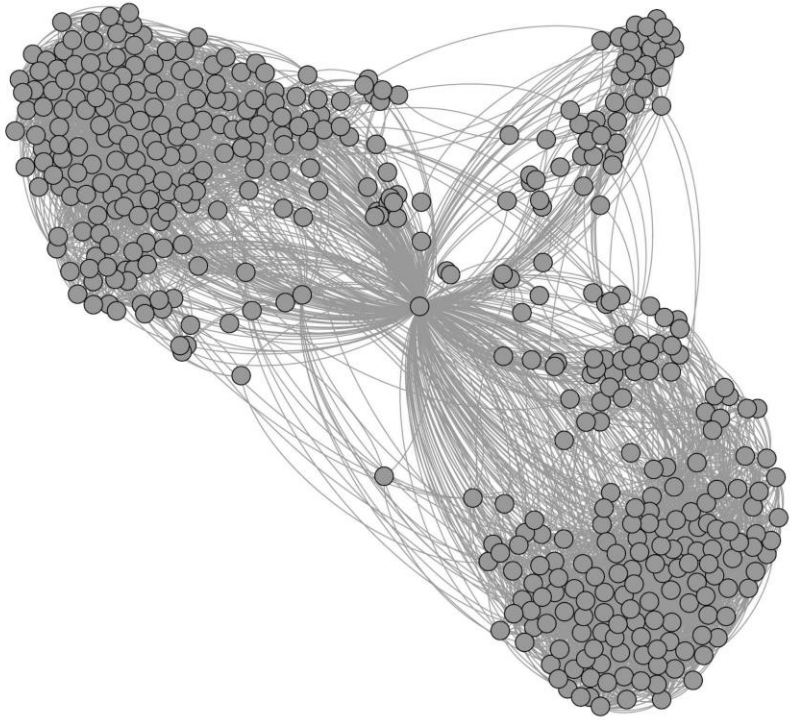
Graph Clustering versus Community detection

- Graph clustering and community detection share many commonalities [COSCIA2011].

- There is a rough distinction between them:
  - ***Clustering***: Group sets of points based on their ***features***.
  - ***Community detection***: Group sets of points based on their ***connectivity***.

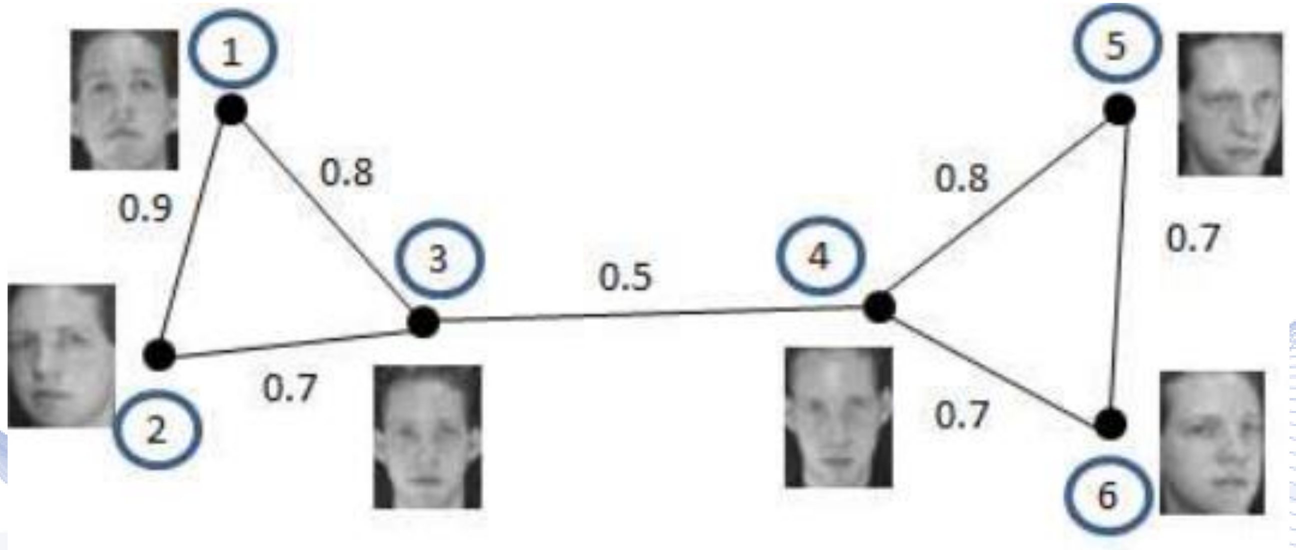
- Related paper [GUID2017].

Graph-based Clustering



Data graph visualization.

Graph-based Clustering



1	0.9	0.8	0	0	0
0.9	1	0.7	0	0	0
0.8	0.7	1	0.5	0	0
0	0	0.5	1	0.8	0.7
0	0	0	0.8	1	0.7
0	0	0	0.7	0.7	1

a) Similarity graph; b) Similarity matrix.

Graph-based Clustering

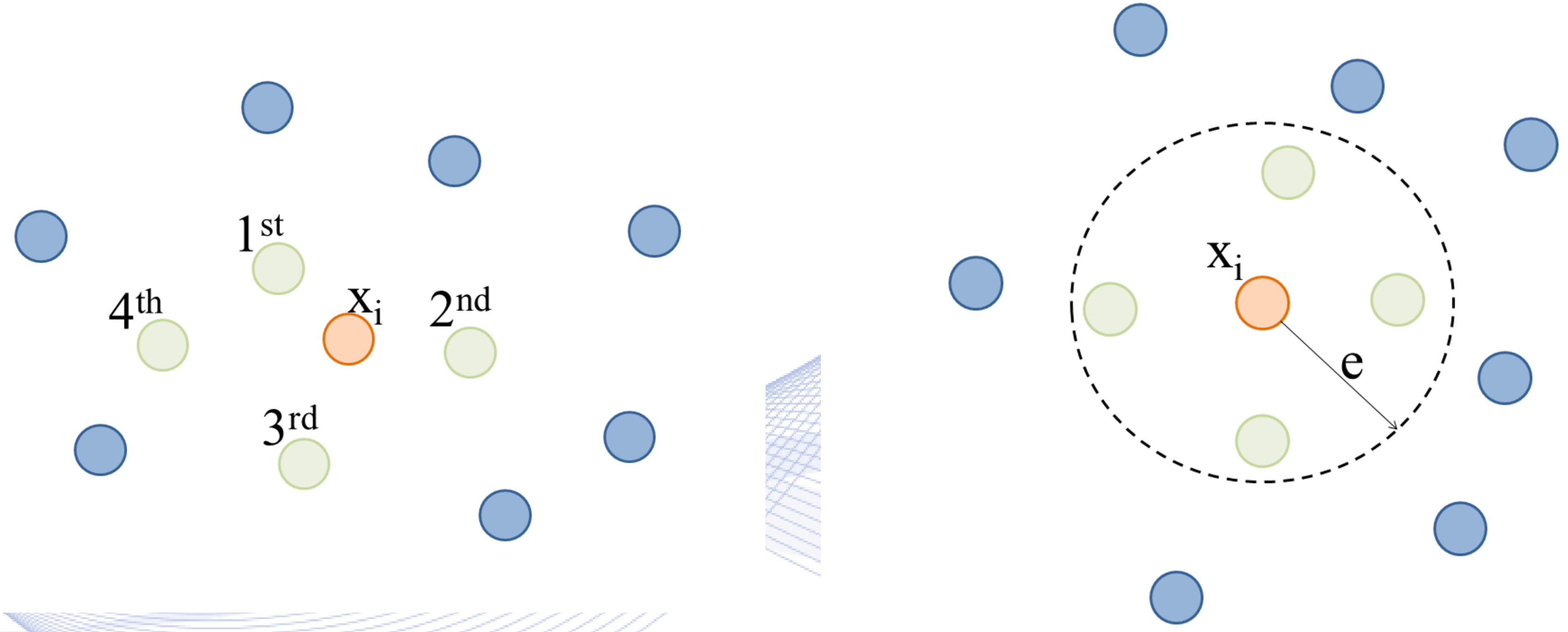
- ***Vertex degree***: number of vertex connection in $\mathbf{A}$.
- ***Gaussian kernel*** for edge weight calculation:

$$W(i,j) = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}}, & \text{if } \left|\left|\mathbf{x}_i - \mathbf{x}_j\right|\right| < e, \\ 0, & \text{otherwise.} \end{cases}$$

- e: is a user-defined constant.
- || . || is Euclidean norm.

Graph-based Clustering

Nearest neighbor graphs



a) k-nearest neighbors graph; b) e-neighborhood graph.

Graph-based Clustering

Graph Clustering

- Cluster graph vertices (data vectors) into tightly linked clusters.

- Vertices of the same cluster are:
  - Strongly connected to each other and
  - sparsely connected to the rest of the graph.

- ***Intra-cluster connectivity***: measured by the cluster density.
- ***Inter-cluster connectivity***: measured by ***graph cut*** cardinality.

Graph-based Clustering

Laplacian matrix eigenanalysis:

- Non-decreasing eigenvalue order:

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_N.$$

- ***Graph spectrum*** is the eigenvalue set: $\{\lambda_i, \quad i = 1, \dots, N\}$
- It is invariant to ***graph isomorphism***
  - Graph vertex permutations.
- Non-isomorphic graphs can be co-spectral.

Graph-based Clustering

- ***Algebraic connectivity*** (eigenvalue λ_2):

- If $\lambda_2 > 0$:
  - graph $\mathcal{G}$ is connected.
- else:
  - multiplicity of eigenvalue 0 is equal number of connected graph components.

Artificial Intelligence & Information Analysis Lab

Graph-based Clustering

- Graph comprised of k disjoint ***cliques***:
  - k smallest eigenvalues of normalized Laplacian matrix are 0.
  - i-th corresponding eigenvector $(0 \leq i \leq k-1)$ has non-zero values for vertices of the i –th clique.

- Adding edges cause the eigenvalues to increase and change slightly corresponding eigenvectors.

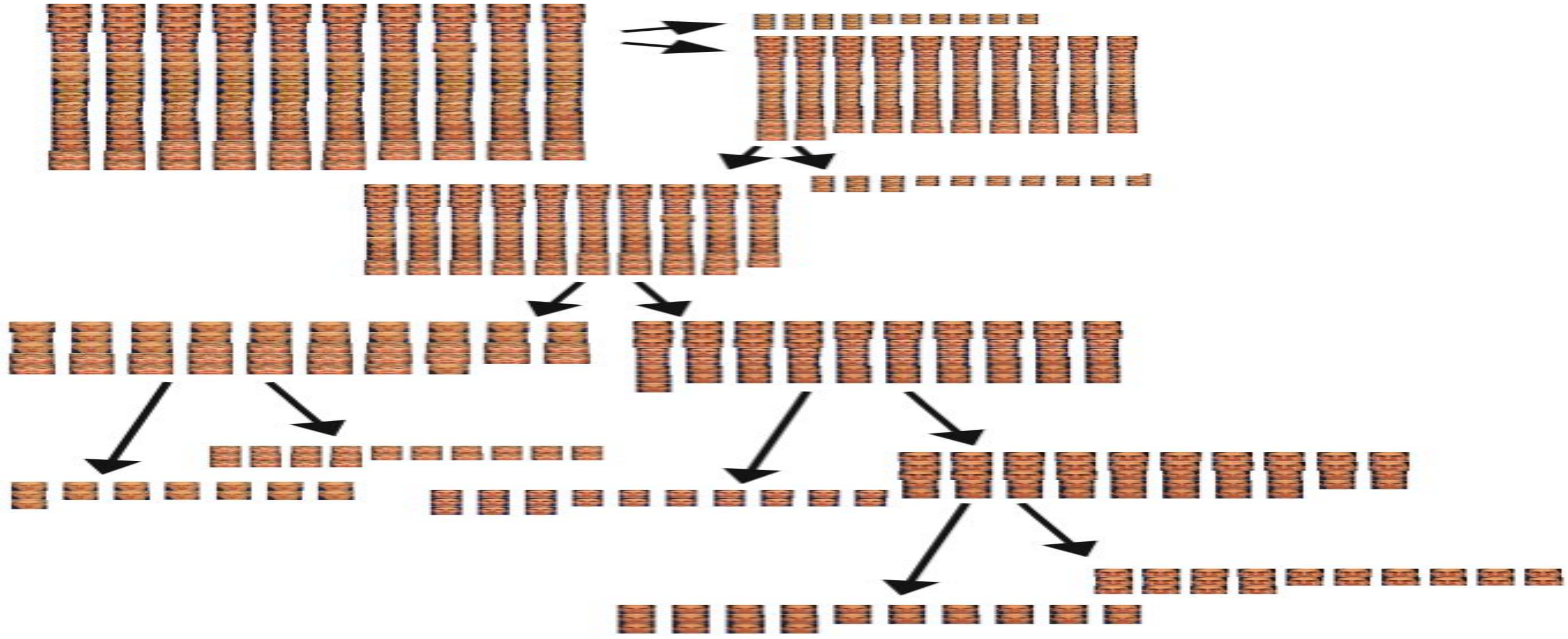
Graph-based Clustering

Graph clustering based on ***spectral bisection***:

- 2-way graph partitioning.

- It uses the so-called ***Fiedler vector***:
  - eigenvector $\mathbf{u}_2$ corresponding to eigenvalue λ_2 of Laplacian matrix.

Graph-based Clustering



N-Cut Graph Clustering (2-way partitioning).

Graph-based Clustering

Edge-based bisection:

- Compute Fiedler vector.
- Split vertices into 2 groups:
  - their relevant Fiedler vector entries are below/above the Fiedler vector entries median.
- Edges between these two groups are cut.

Graph-based Clustering

Vertex-based bisection:

- Compute Fiedler vector.
- Find the largest gap in Fiedler vector entries
- Split Fiedler vector entries accordingly.
- Split the graph at the cut provides the best cut quotient.

Graph-based Clustering

Spectral graph clustering:

- Perform eigenanalysis on one of the normalized Laplacians.
- Extract r eigenvectors corresponding to the smallest eigenvalues excluding λ_1.
- Store eigenvectors in a $\mathrm{N} \times r$ matrix $\mathbf{U}$.
- Its rows are the new data representation.
- Use any standard clustering algorithm to cluster them.

Graph-based Clustering

Graph-based clustering properties:

- Little user input is needed.
- Trivial clusters easily avoided.
- Unlikely to get bad clustering results.
- They cannot be employed in extremely large graphs:
  - Memory limitations.
- Eigenanalysis has $O(N^3)$ computational complexity.

Bibliography

[PIT2016] I. Pitas (editor), “Graph analysis in social media”, CRC Press, 2016.
[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.
[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr