# Spectral Graph Embedding

**Social Networks Analysis and Graph Algorithms**

Prof. Carlos Castillo — https://chato.cl/teach

Universitat Pompeu Fabra Barcelona

# Sources

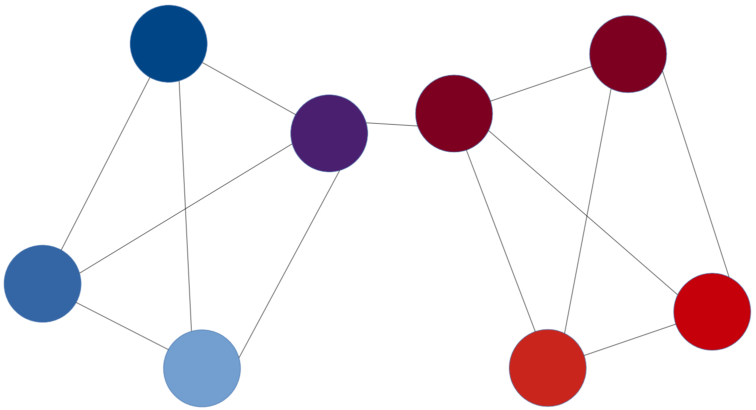
- **J. Leskovec (2016). Defining the graph laplacian** [video]
- E. Terzi (2013). Graph cuts — The part on spectral graph partitioning
- D. A. Spielman (2009): The Laplacian
- URLs cited in the footer of slides

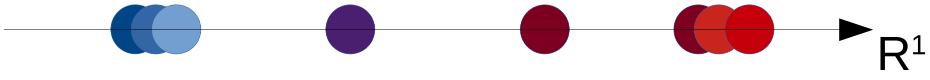
# Many algorithms are not suitable for graphs

- Many algorithms need a notion of similarity or distance (both are interchangeable)
- **Data mining**: clustering, outlier detection, ...
- **Retrieval/search**: nearest neighbors, ...

# Graphs are nice, but ...

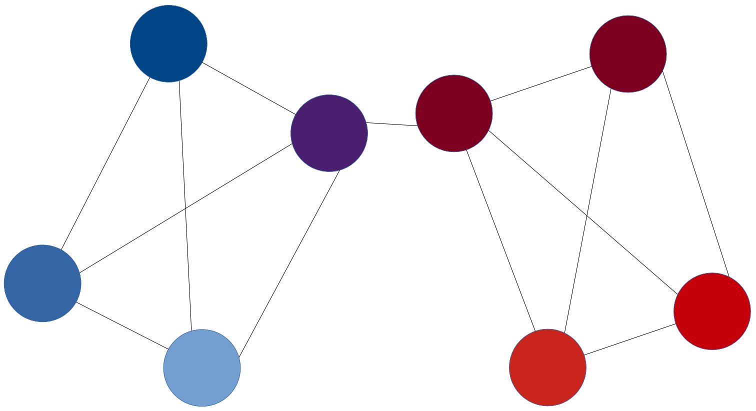
- They describe only local relationships
- We would like to understand a global structure
- We will try to transform a graph into a more familiar object: a cloud of points in $R^k$





# Graphs are nice, but ...

- They describe only local relationships
- We would like to understand a global structure
- We will try to transform a graph into a more familiar object: a cloud of points in $R^k$




**Distances should be somehow preserved**

# What is a graph embedding?

- A graph **embedding** (or graph **projection**) is a mapping from a graph to a vector space
- If the vector space is $\mathbb{R}^2$ you can think of an embedding as a way of ***drawing*** a graph on paper

# Exercise: draw this graph

V = {v1, v2, ..., v8}

E = { (v1, v2), (v2, v3), (v3, v4), (v4, v1), (v5, v6), (v6, v7), (v7, v8),
(v8, v5), (v1,v5), (v2, v6), (v3, v7), (v4, v8) }

**Draw this graph on paper, upload a photo**

What constitutes a good drawing?

Padlet: https://upfbarcelona.padlet.org/chato/3hwnctvqb7p1z6xx

# In a good graph embedding ...

- Pairs of nodes that are **connected** to each other should be **close**
- Pairs of nodes that are not connected should be far
- Compromises will need to be made

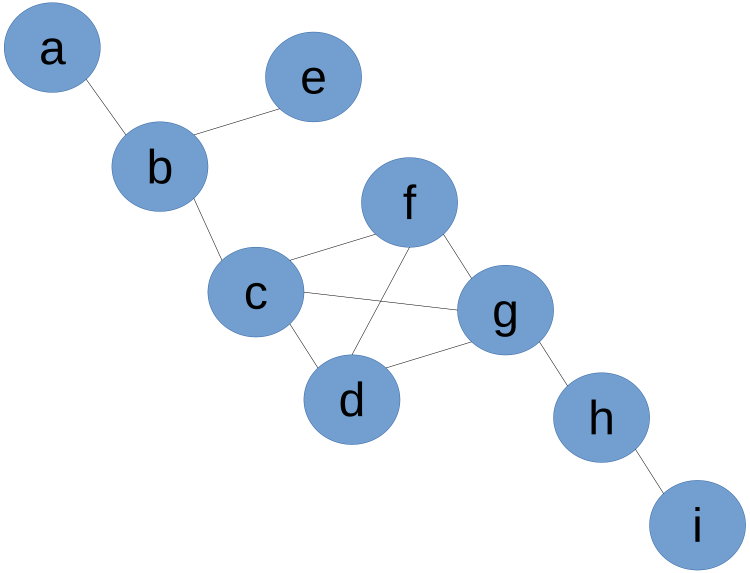
# Random projections

# Random graph projection (2D)

- Start a BFS from a random node, that has x=1, and nodes visited have ascending x
- Start a BFS from another random node, which has y=1, and nodes visited have ascending y
- Project node i to position $(x_i, y_i)$

# Exercise: random projection

- Given this graph
- Pick a random node $u$
  - Distances from $u$ are the x positions
- Pick a random node v
  - Distances from $v$ are the y positions
- Draw the graph in an $\mathbb{R}^2$ plane




Padlet: https://upfbarcelona.padlet.org/chato/9pd56scbpko5svdj

# Refresher about eigenvectors/eigenvalues

# Eigenvectors of symmetric matrices

- In general $Av = \lambda v$ means $A$ has an eigenvector $v$ of eigenvalue $\lambda$
- In symmetric matrices ($A$=$A^T$), eigenvectors are orthogonal

  Suppose $v_1$, $v_2$ are eigenvectors of eigenvalues $\lambda_1$, $\lambda_2$ with $\lambda_1 \neq \lambda_2$

$$\lambda_1 \langle v_1, v_2 \rangle = \langle \lambda_1 v_1, v_2 \rangle = \langle A v_1, v_2 \rangle = \langle v_1, A^T v_2 \rangle$$
$$= \langle v_1, A v_2 \rangle = \langle v_1, \lambda_2 v_2 \rangle = \lambda_2 \langle v_1, v_2 \rangle$$

For any real matrix
$$\langle Ax, y \rangle = \langle x, A^T y \rangle$$

- Therefore:

$$(\lambda_1 - \lambda_2) \langle v_1, v_2 \rangle = 0 \wedge (\lambda_1 - \lambda_2) \neq 0 \Rightarrow \langle v_1, v_2 \rangle = 0$$

# In symmetric matrices

- The **multiplicity** of an eigenvalue $\lambda$ is the dimension of the space of eigenvectors of eigenvalue $\lambda$
- Every $n$ x $n$ symmetric matrix has $n$ eigenvalues counted with multiplicity
- Hence, it has an orthonormal basis of eigenvectors

# Rayleigh quotient

In symmetric matrices $M$, the second smallest eigenvalue is

$$\lambda_2 = \min_x \frac{x^T M x}{x^T x}$$

https://en.wikipedia.org/wiki/Rayleigh_quotient

# Eigenvectors of the adjacency matrix (of an unweighted graph)

# Adjacency matrix (of unweighted graph)

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- How many non-zeros are in every **row** of A?

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

https://www.youtube.com/watch?v=AR7iFxM-NkA

# Adjacency matrix of G=(V,E)

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

**Can you write $\boldsymbol{y_i}$ using $\boldsymbol{E}$ ?**

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

https://www.youtube.com/watch?v=AR7iFxM-NkA

# Adjacency matrix of G=(V,E)

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- **What is $\boldsymbol{Ax}$?** Think of $x$ as a set of labels/values:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$y_i = \sum_{j:(i,j)\in E} x_j$$

Ax is a vector whose $i^{th}$ coordinate contains the sum of the $x_j$ who are in-neighbors of i

https://www.youtube.com/watch?v=AR7iFxM-NkA

# Spectral graph theory ...

- Studies the eigenvalues and eigenvectors of a graph matrix
  - Adjacency matrix $Ax = \lambda x$
  - Laplacian matrix (next)

- Suppose graph is d-regular: $k_i = d \; \forall i$
- Multiply its adjacency by $\vec{1}$
- Look at the result, what does it imply?

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = ?$$

# An eigenvector of a d-regular graph

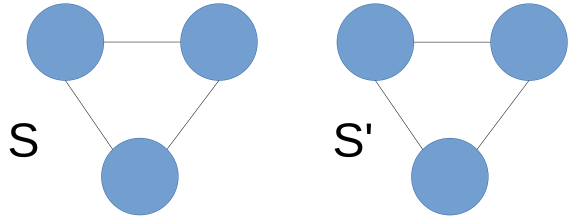
- Suppose graph is d-regular, i.e. all nodes have degree d:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} d \\ d \\ \vdots \\ d \end{bmatrix} = d \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

- **Hence, $[1, 1, \ldots, 1]^T$ is an eigenvector of eigenvalue d**

# Disconnected graphs

- Suppose the graph is regular **and disconnected**

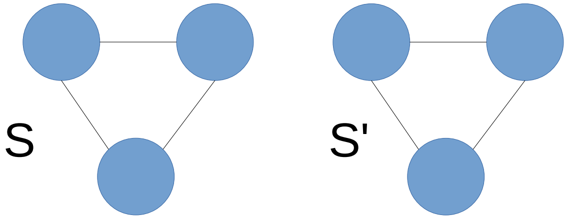



- Then its adjacency matrix has block structure:

$$A = \begin{bmatrix} S & 0 \\ 0 & S' \end{bmatrix}$$

# Disconnected graphs

- Suppose the graph is regular **and disconnected**

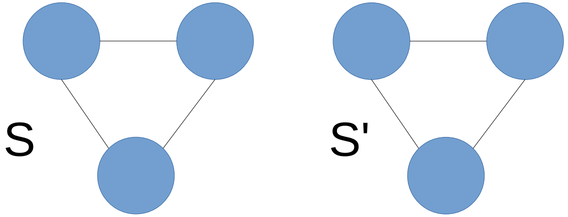



$$\text{Let } x_i^S = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \in S' \end{cases}$$

$$\begin{bmatrix} S & 0 \\ 0 & S' \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = ?$$

# Disconnected graphs

- Suppose the graph is regular **and disconnected**



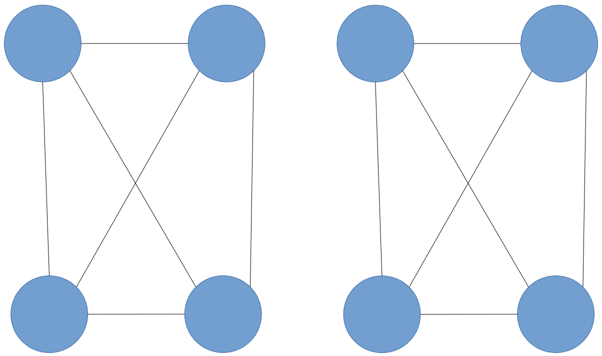

$$Ax^S = dx^S$$

$$Ax^{S'} = dx^{S'}$$

- What is the multiplicity of eigenvalue d?
- What happens if there are more than 2 connected components?

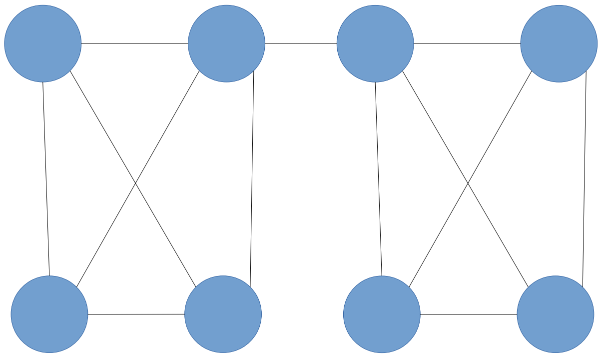
# In general

Disconnected graph



$$\lambda_1 = \lambda_2$$

*Almost* disconnected graph



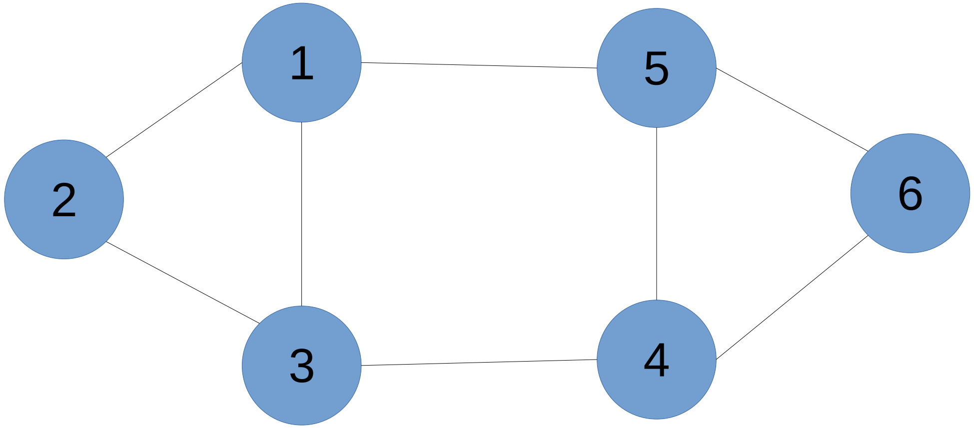
$$\lambda_1 \approx \lambda_2$$

Small “**eigengap**”

# Graph Laplacian

# Adjacency matrix

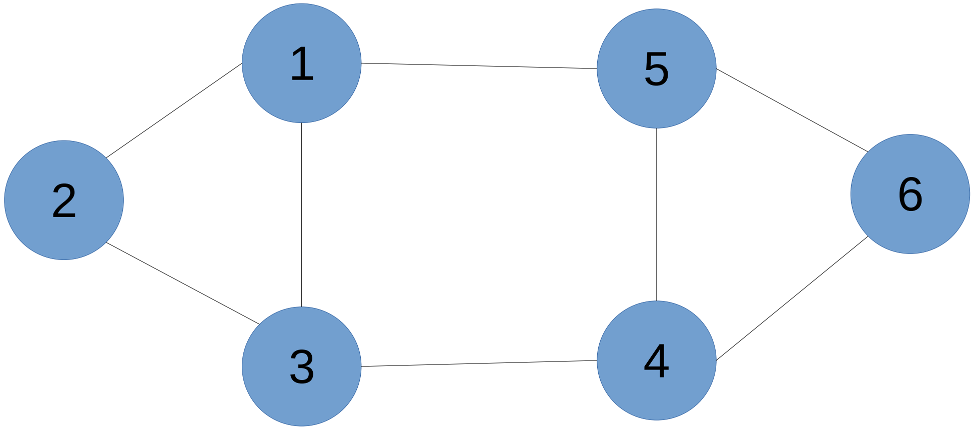
$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$




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

# Degree matrix

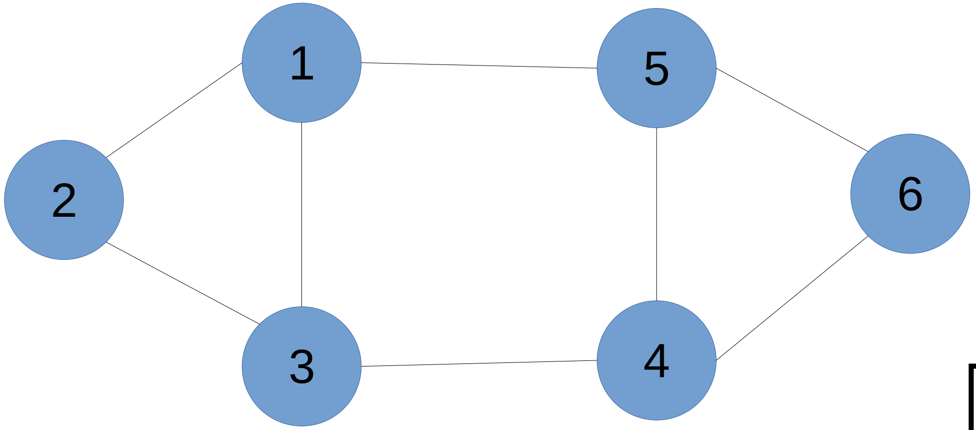
$$D_{ij} = \begin{cases} k_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$




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

# Laplacian matrix

$$L = D - A$$




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

Because A is symmetric, and we have only changed the diagonal, **L is symmetric.**

# Laplacian matrix L = D - A

$$L\vec{1} = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = ?$$

# The constant vector is an eigenvector of L

The constant vector $x=[1,1,...,1]^T$ is an eigenvector of the Laplacian, and has eigenvalue $0$

$$Lx = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Does it need to be this specific graph? Why?

Does it need to be the vector [1, 1, …, 1]? Why?

# If the graph is disconnected

- If the graph is disconnected into two components, the same argument as for the adjacency matrix applies, and $\lambda_1 = \lambda_2 = 0$
- The multiplicity of eigenvalue 0 is equal to the number of connected components

$$x^T L x$$

# Prove this!

Prove that $\mathrm{x}^T L x = \sum_{(i,j)\in E}(x_i - x_j)^2$

$$L_{ij} = D_{ij} - A_{ij}$$

$$D_{ij} = \begin{cases} k_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \qquad A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Assume that E only contains each edge in one direction

Think of this quantity as the "stress" produced by the assignment of node labels x

# As shown before, the constant vector is one of the eigenvectors of L, with eigenvalue 0

- If $x$ is such that $x_i = x_j$ for all $i,j$:

$$x^T L x = \sum_{(i,j)\in E} (x_i - x_j)^2 = 0 \Rightarrow Lx = 0$$

- This means the constant vector is an eigenvector of L with eigenvalue 0

# The eigenvector $x$ of $\lambda=0$ is the constant vector if the graph is connected

- If $x$ is the eigenvector of eigenvalue 0, $Lx = 0$
- Then $x^T L x = \sum_{(i,j)\in E} (x_i - x_j)^2 = 0$

From this, we deduct that $x_i = x_j$ for any pair $i, j$ even if $i$ and $j$ are not directly connected by an edge. Why?

# The eigenvector $x$ of $\lambda=0$ is the constant vector if the graph is connected

- If $x$ is the eigenvector of eigenvalue 0, $Lx = 0$
- Then $x^T L x = \sum_{(i,j)\in E} (x_i - x_j)^2 = 0$
- Hence, for any pair of nodes $(i,j)$ connected by an edge, $x_i = x_j$
- Given the graph is connected, there is a path between any two nodes ⇒

  for **any** pair of nodes $(i,j)$, **even the ones not connected by an edge**, $x_i = x_j$
- Hence $x$ is a constant vector

# All the eigenvalues of the Laplacian are non-negative

- If $v$ is an eigenvector of $L$ of eigenvalue $\lambda$:

$$\lambda v^T v = v^T L v = \sum_{(i,j)\in E} (v_i - v_j)^2 \geq 0$$

- This means all eigenvalues $\lambda$ are non-negative

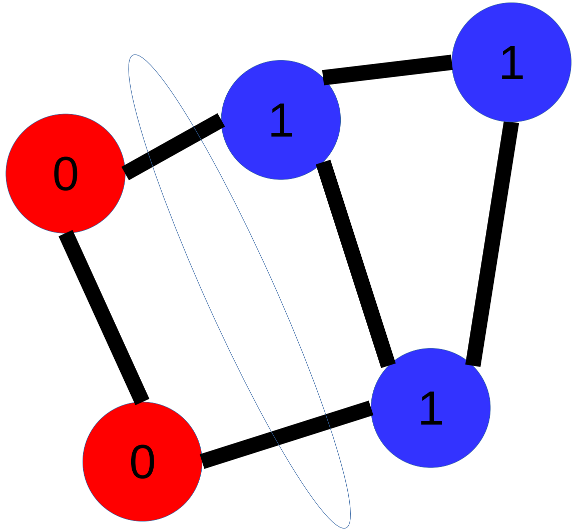
# In summary, the Laplacian matrix L = D - A

- Is symmetric, eigenvectors are orthogonal
- Has $N$ eigenvalues that are non-negative
- *0* is one eigenvalue $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$
- The multiplicity of eigenvalue *0* equals the number of connected components of the graph

# The second smallest eigenvalue of the Laplacian

# $x^TLx$ and graph cuts

- Suppose *c(S, S')* is a cut of graph G
- Set $x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \in S' \end{cases}$




$$|c(S, S')| = 2$$

$$x^T L x = \sum_{(i,j)\in E} (x_i - x_j)^2 = \sum_{(i,j)\in c(S,S')} 1^2 = |c(S, S')|$$

# Remember

- For symmetric matrices

$$\lambda_2 = \min_x \frac{x^T M x}{x^T x}$$

- If $x$ is an eigenvector, $\frac{x^T M x}{x^T x}$ is its eigenvalue

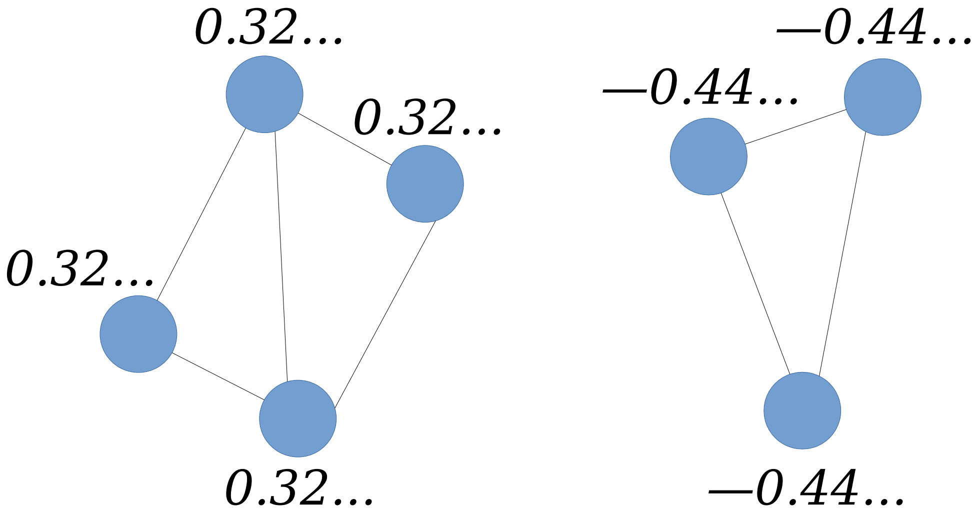
https://en.wikipedia.org/wiki/Rayleigh_quotient

# Second eigenvector

- Orthogonal to the first one: $x \cdot \vec{1} = 0 \Rightarrow \sum_i x_i = 0$
- Normal: $\sum_i x_i^2 = 1$

$$\lambda_2 = \min_x \frac{x^T L x}{x^T x} = \min_{x:\sum x_i = 0} \frac{x^T L x}{\sum x_i^2} = \min_{x:\sum x_i = 0 \wedge \sum x_i^2 = 1} \sum_{(i,j)\in E} (x_i - x_j)^2$$

# The second eigenvalue in a disconnected graph




If the graph is divided into two connected components of sizes $N_1$ and $N_2$, you can use this assignment

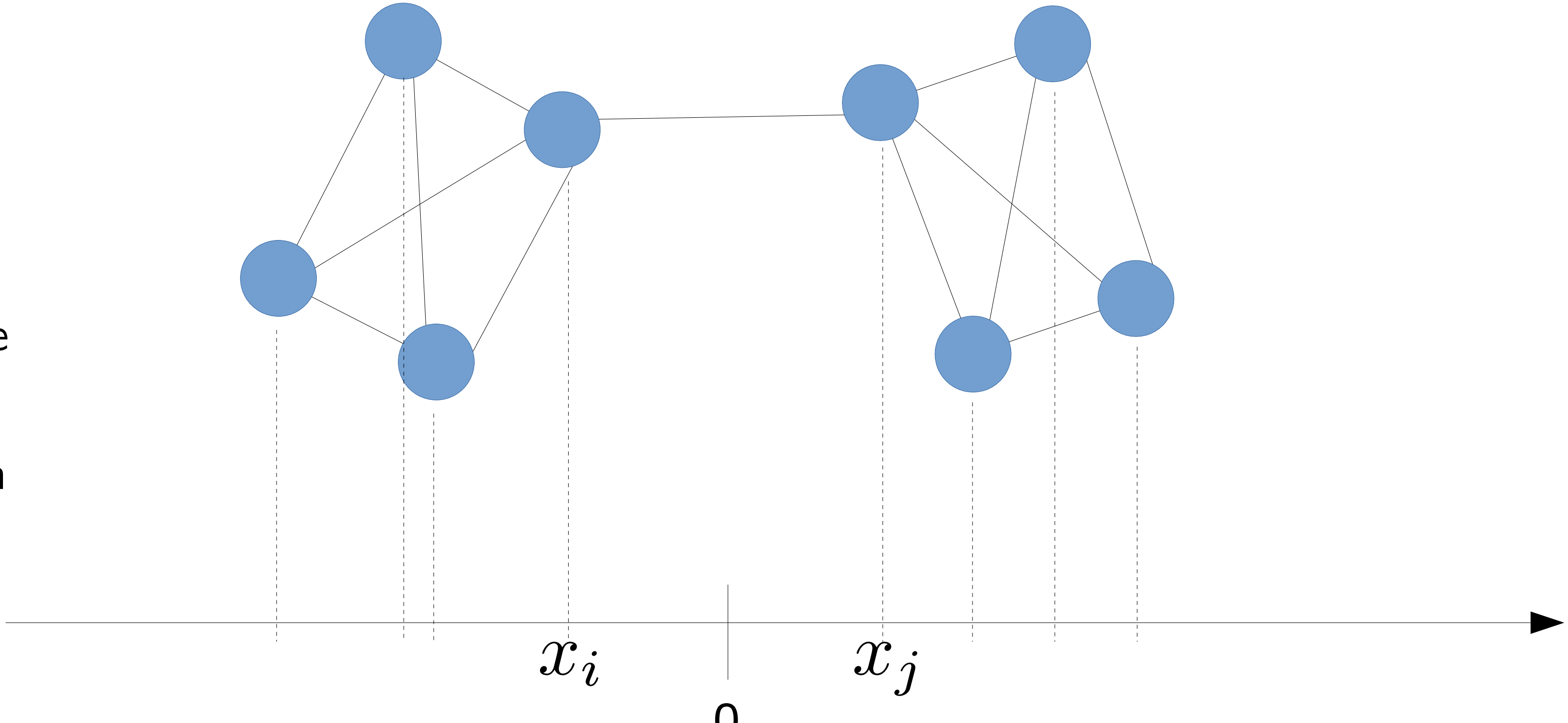
What's its eigenvalue?

$$\lambda_2 = \min_{x:\sum x_i = 0 \wedge \sum x_i^2 = 1} \sum_{(i,j)\in E} (x_i - x_j)^2$$

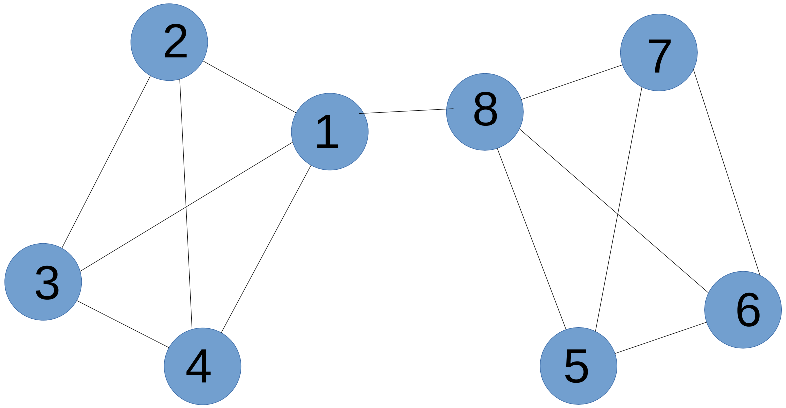
# The second eigenvalue tells us how well the graph can be partitioned into two

$$\lambda_2 = \min_{x:\sum x_i=0 \wedge \sum x_i^2=1} \sum_{(i,j)\in E} (x_i - x_j)^2$$

If the graph is connected but almost partitioned into two component, the optimal $x$ should have values similar to each other in each partition

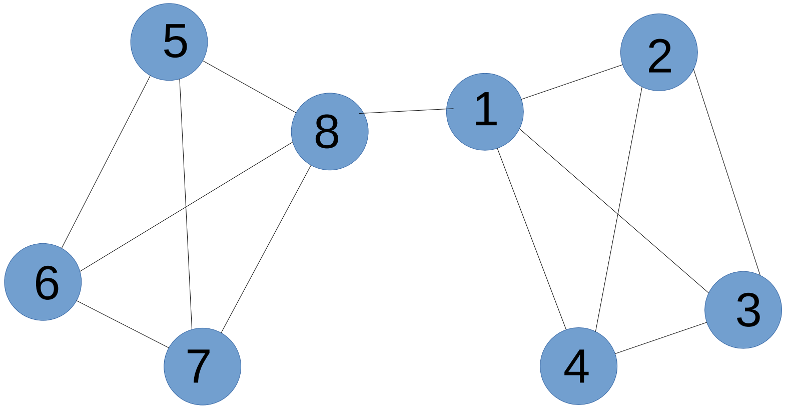


# Example Graph 1




$$L = \begin{bmatrix} 4 & -1 & -1 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & -1 & -1 & 4 \end{bmatrix}$$

# Example Graph 1 (second eigenvalue of L)

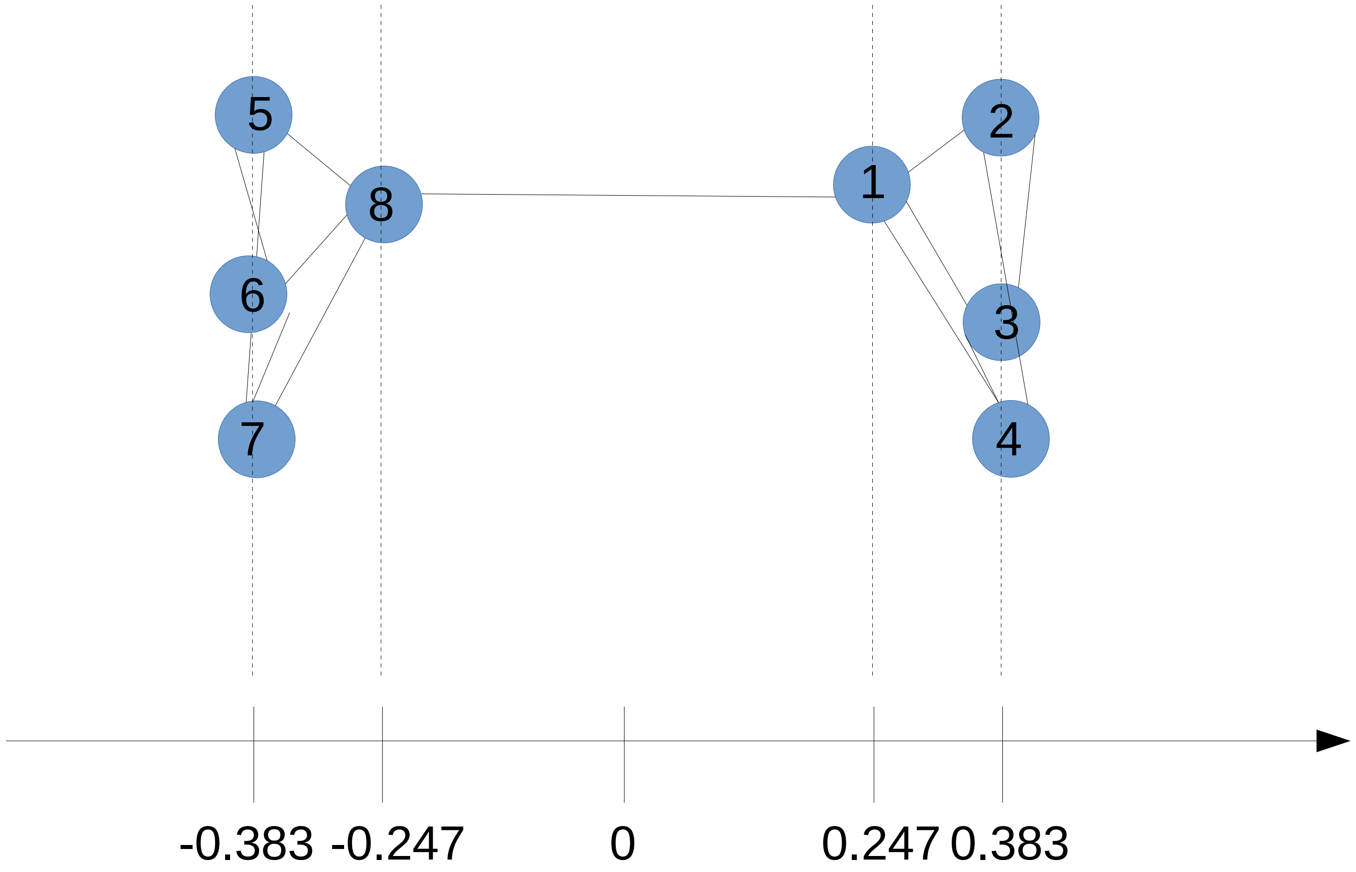



$$\lambda_1 = 0$$
$$\lambda_2 = 0.354$$

$$L = \begin{bmatrix} 4 & -1 & -1 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & -1 & -1 & 4 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} 0.247 \\ 0.383 \\ 0.383 \\ 0.383 \\ -0.383 \\ -0.383 \\ -0.383 \\ -0.247 \end{bmatrix}$$

# Example Graph 1, projected in $\mathbb{R}^1$

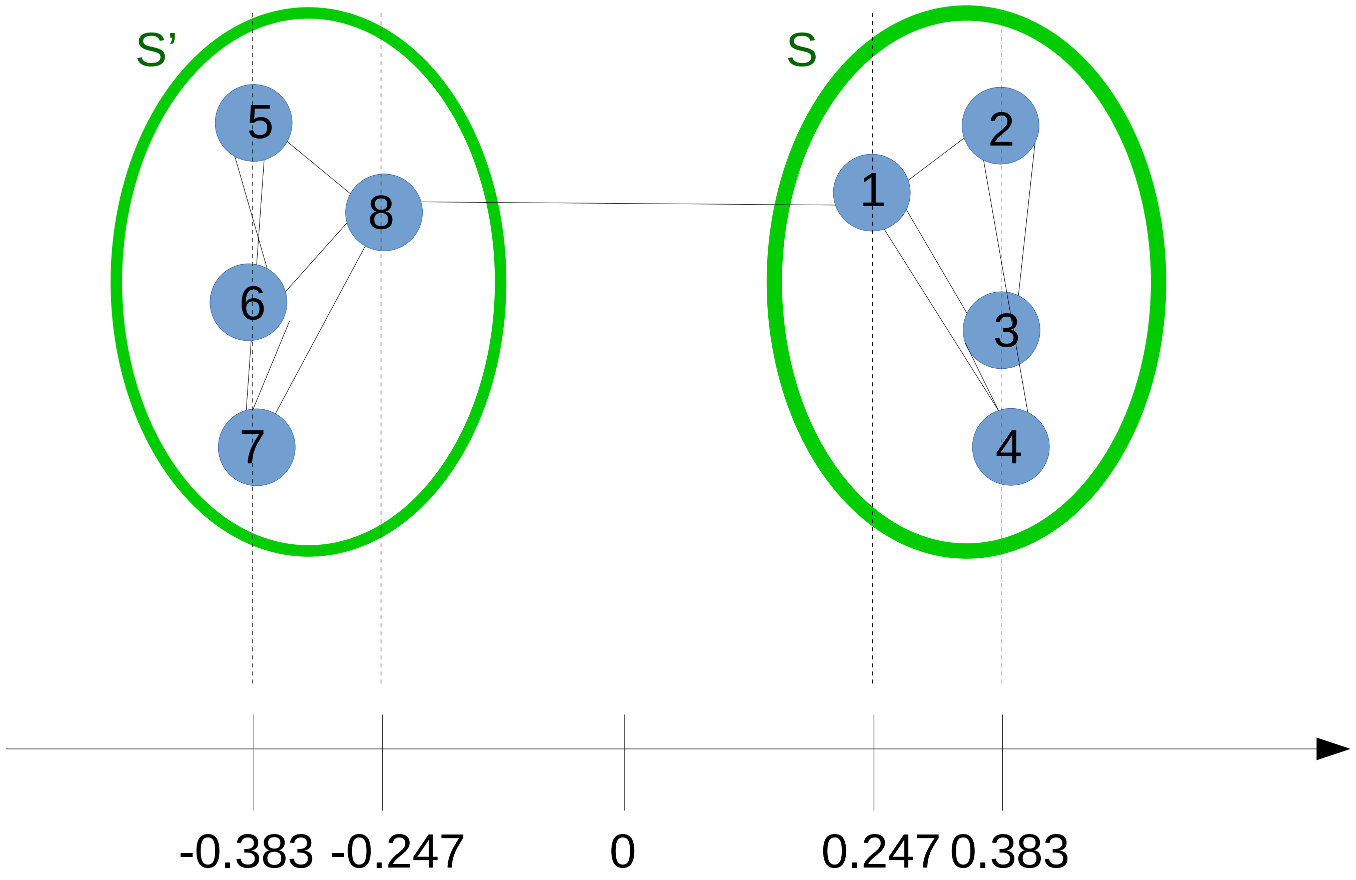



$$\lambda_1 = 0$$

$$\lambda_2 = 0.354$$

$$v_2 = \begin{bmatrix} 0.247 \\ 0.383 \\ 0.383 \\ 0.383 \\ -0.383 \\ -0.383 \\ -0.383 \\ -0.247 \end{bmatrix}$$

# Example Graph 1, communities

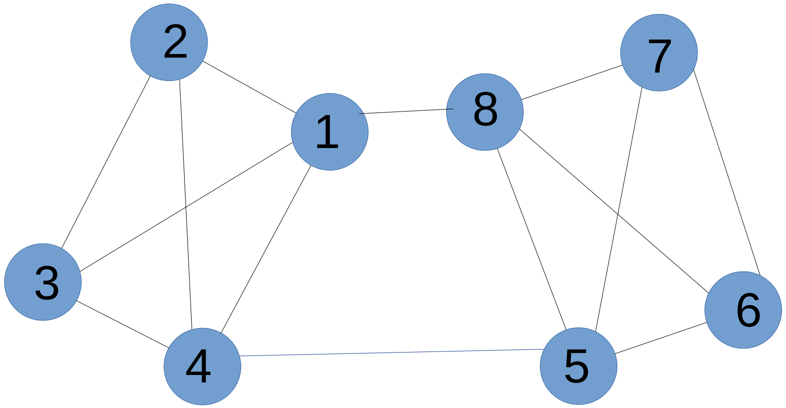



$$\lambda_1 = 0$$

$$\lambda_2 = 0.354$$

$$v_2 = \begin{bmatrix} 0.247 \\ 0.383 \\ 0.383 \\ 0.383 \\ -0.383 \\ -0.383 \\ -0.383 \\ -0.247 \end{bmatrix}$$

# Example Graph 2



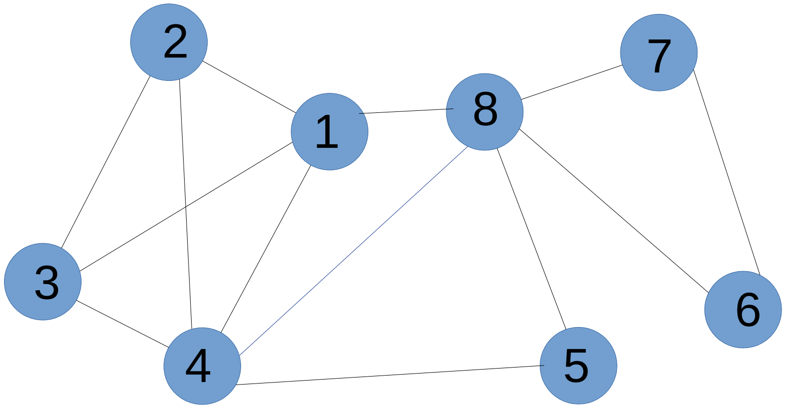

$$\lambda_1 = 0$$

$$\lambda_2 = 0.764$$

$$L = \begin{bmatrix} 4 & -1 & -1 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & -1 & -1 & 4 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} 0.263 \\ 0.425 \\ 0.425 \\ 0.263 \\ -0.263 \\ -0.425 \\ -0.425 \\ -0.263 \end{bmatrix}$$

# Example Graph 3

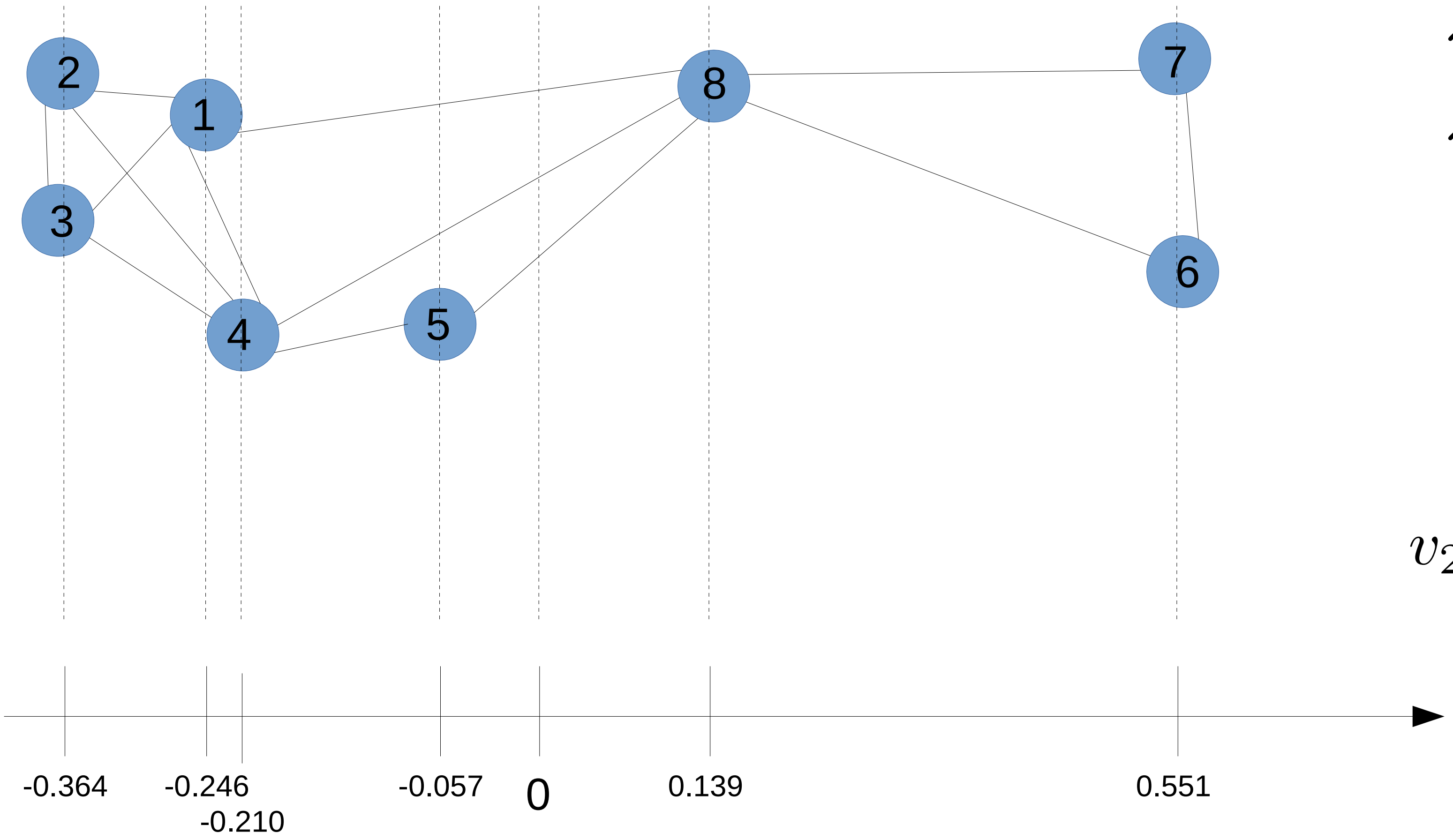



$$\lambda_1 = 0$$
$$\lambda_2 = 0.748$$

$$L = \begin{bmatrix} 4 & -1 & -1 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 5 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & -1 & -1 & -1 & 5 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} -0.246 \\ -0.364 \\ -0.364 \\ -0.210 \\ -0.057 \\ 0.551 \\ 0.551 \\ 0.139 \end{bmatrix}$$

# Example Graph 3, projected (where to cut?)

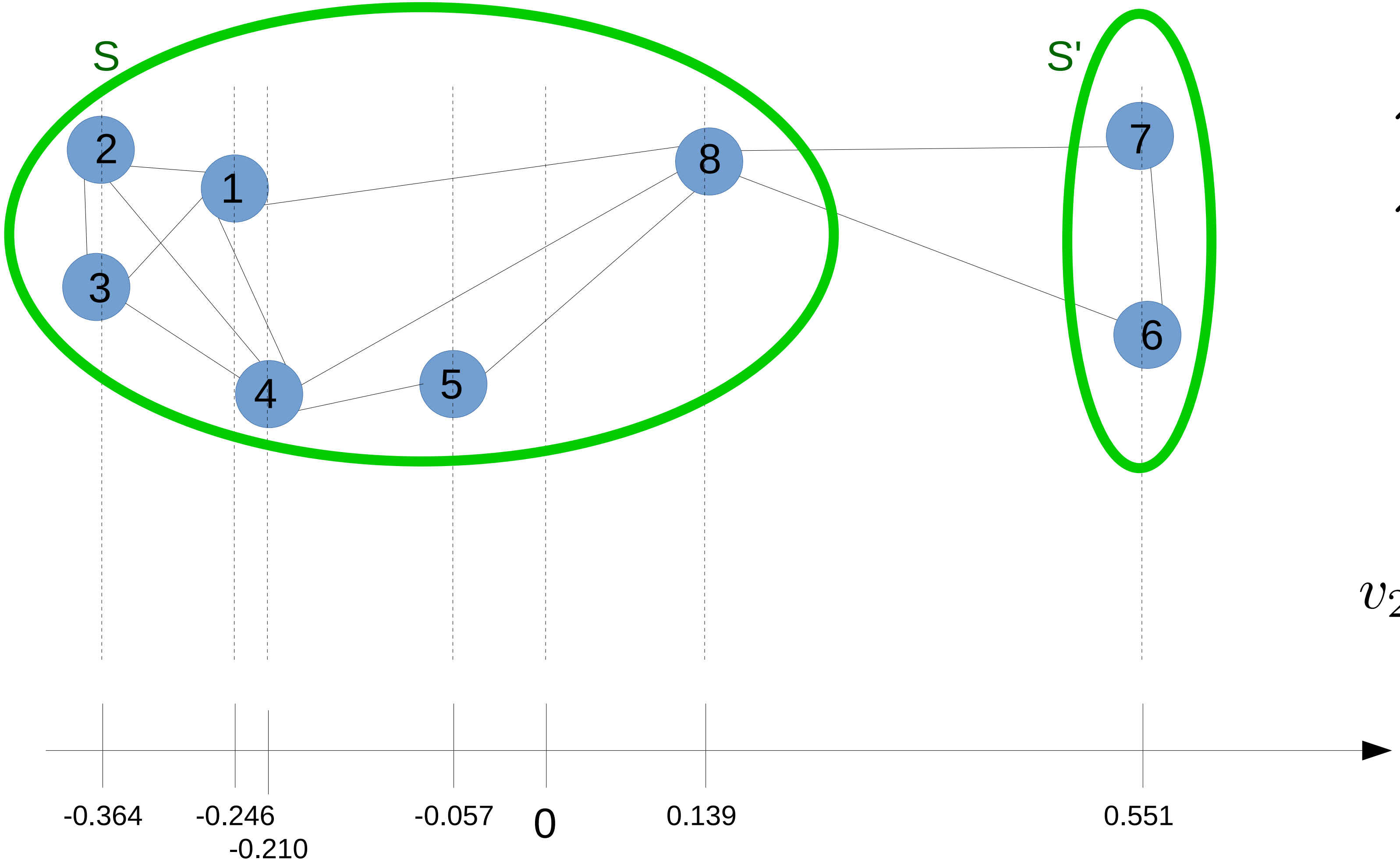



$$\lambda_1 = 0$$

$$\lambda_2 = 0.748$$

$$v_2 = \begin{bmatrix} -0.246 \\ -0.364 \\ -0.364 \\ -0.210 \\ -0.057 \\ 0.551 \\ 0.551 \\ 0.139 \end{bmatrix}$$

# Example Graph 3, projected (where to cut?)

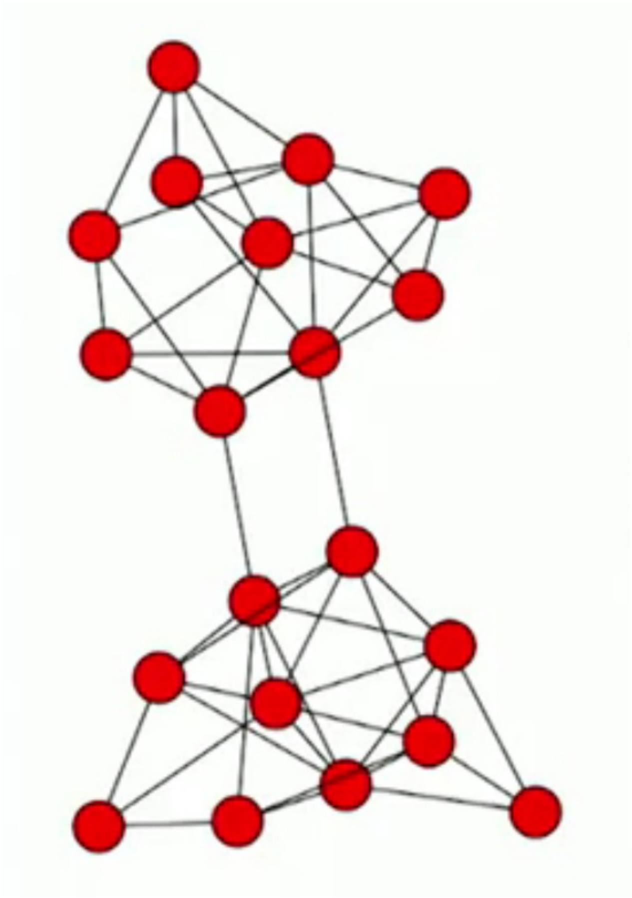


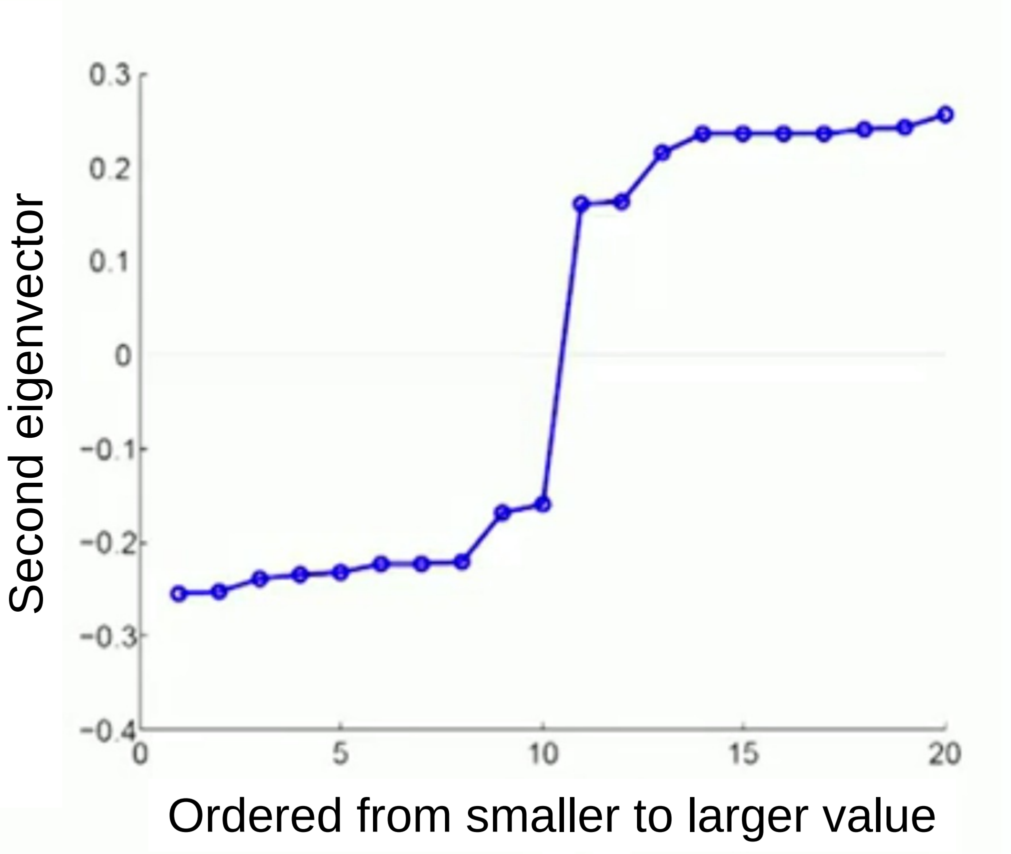

$$\lambda_1 = 0$$

$$\lambda_2 = 0.748$$

$$v_2 = \begin{bmatrix} -0.246 \\ -0.364 \\ -0.364 \\ -0.210 \\ -0.057 \\ 0.551 \\ 0.551 \\ 0.139 \end{bmatrix}$$

# A graph with two communities in $\mathbb{R}^1$

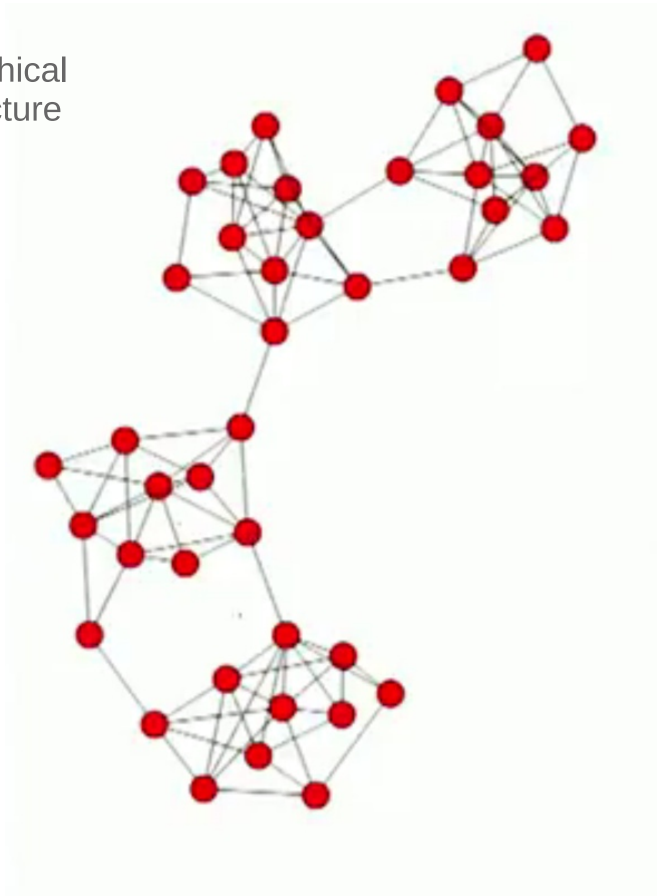


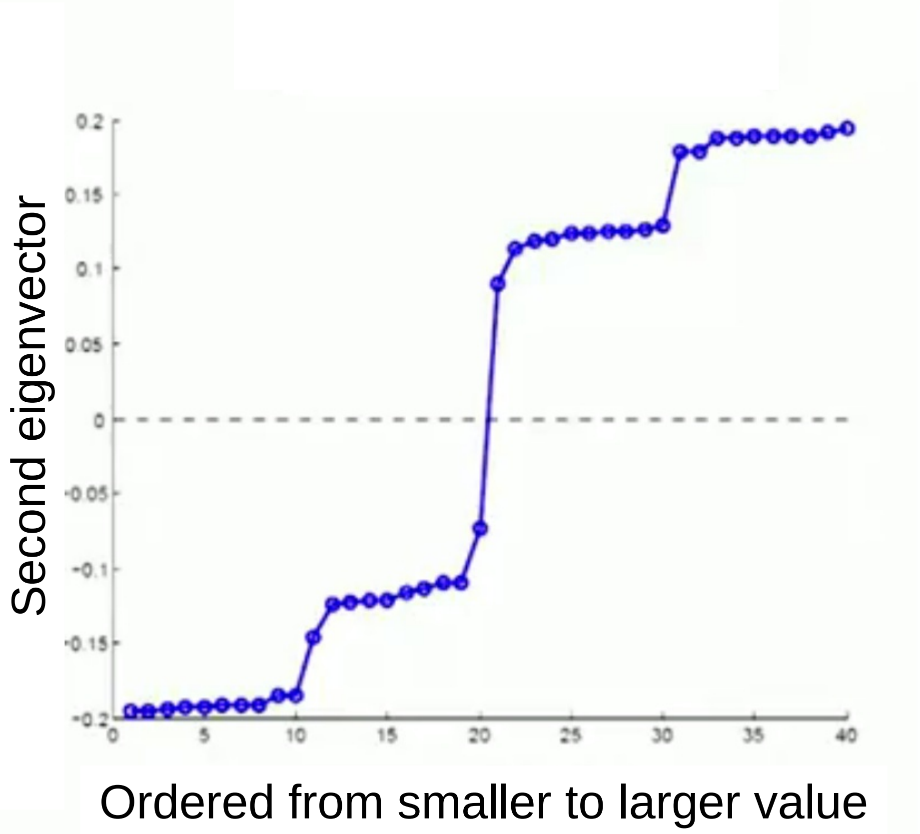



https://www.youtube.com/watch?v=jpTjj5PmcMM

# A graph with four communities in $\mathbb{R}^1$

Note the hierarchical community structure

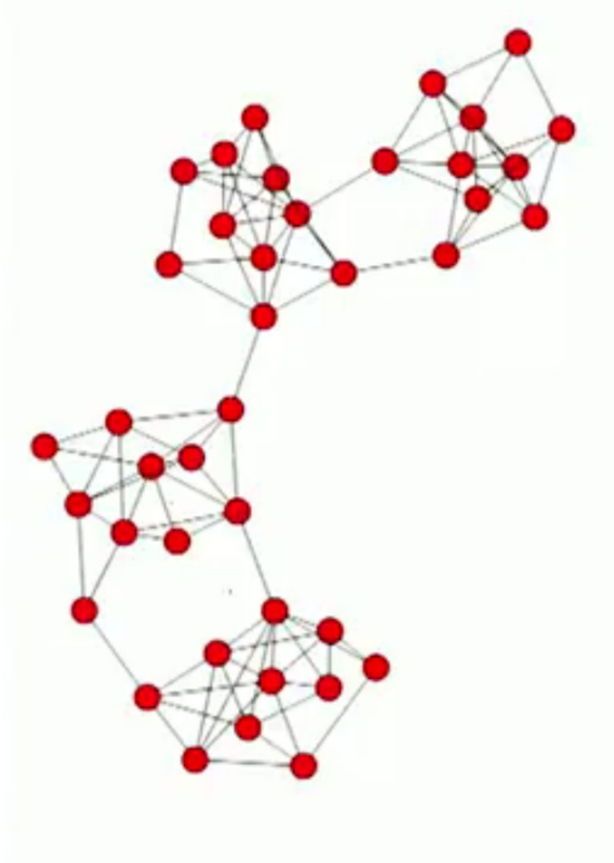


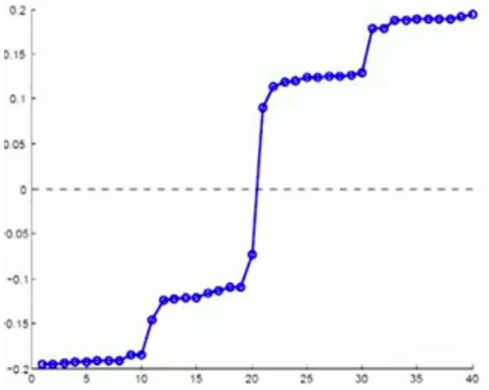



https://www.youtube.com/watch?v=jpTjj5PmcMM

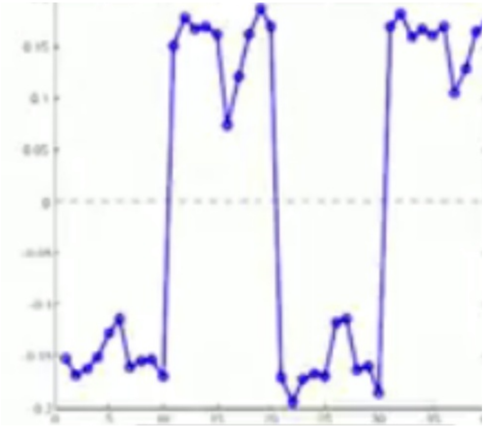
# Application: graph drawing

# A graph with four communities in $\mathbb{R}^2$

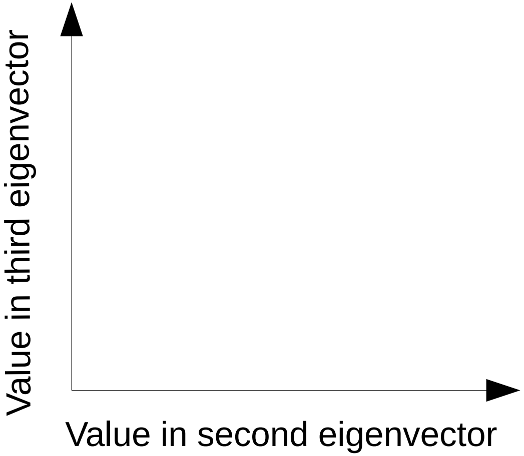




Second eigenvector

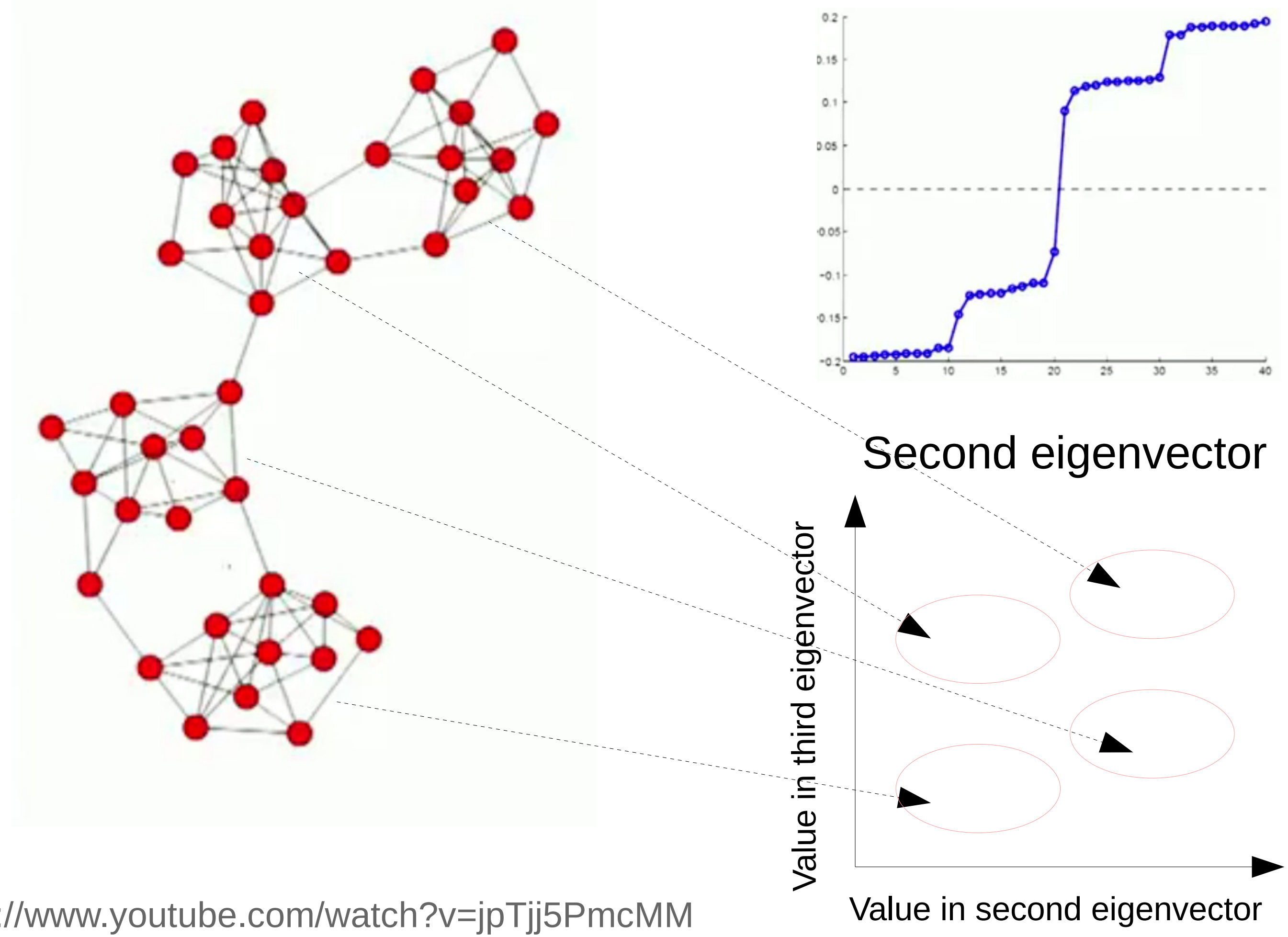


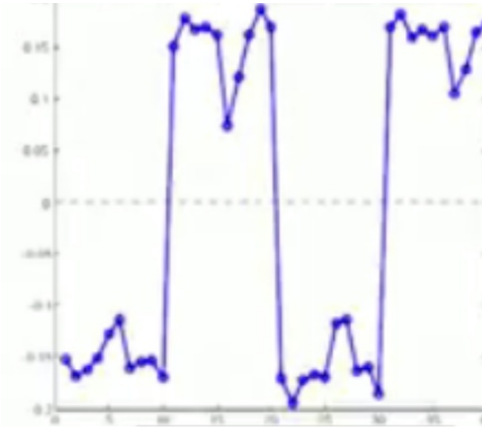
Third eigenvector




https://www.youtube.com/watch?v=jpTjj5PmcMM

# A graph with four communities in $R^2$ (cont)



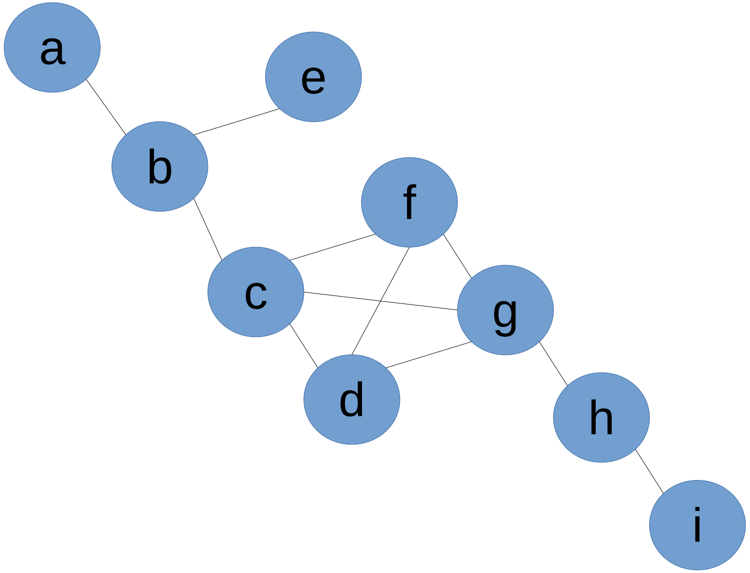



Third eigenvector

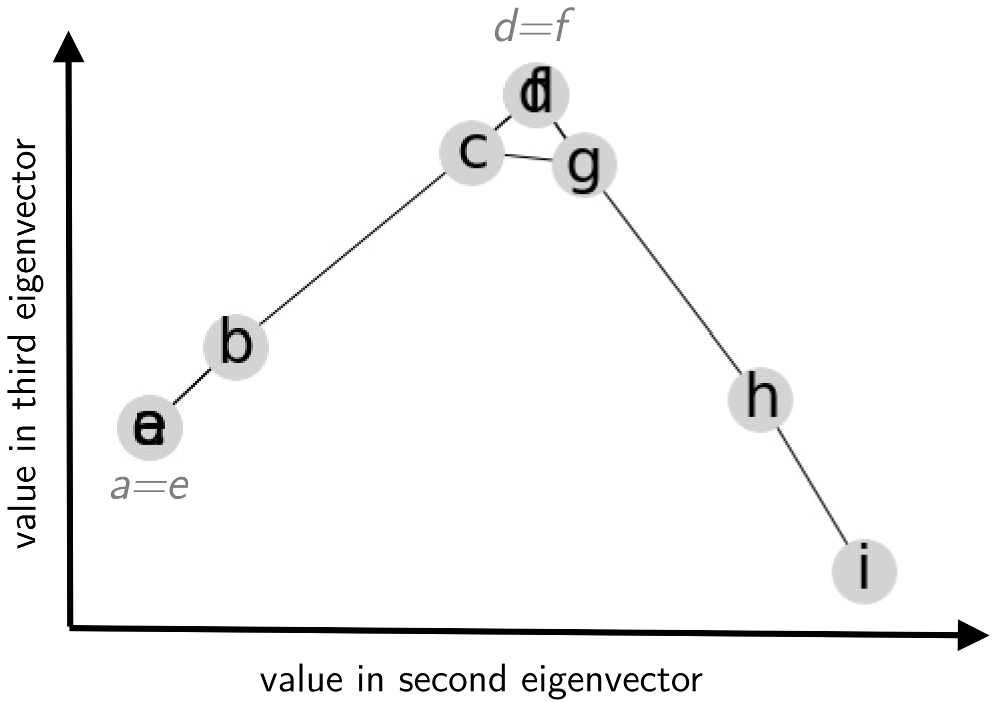
This can be used to draw the graph in $R^2$

https://www.youtube.com/watch?v=jpTjj5PmcMM

# The graph from the initial exercise




**Input nodes and edges**

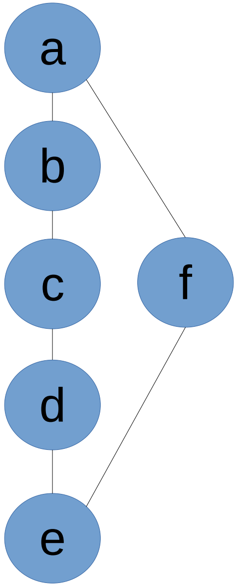



**Spectral embedding**

# Exercise: spectral projection

- Write the Laplacian
- Get the second and third eigenvector

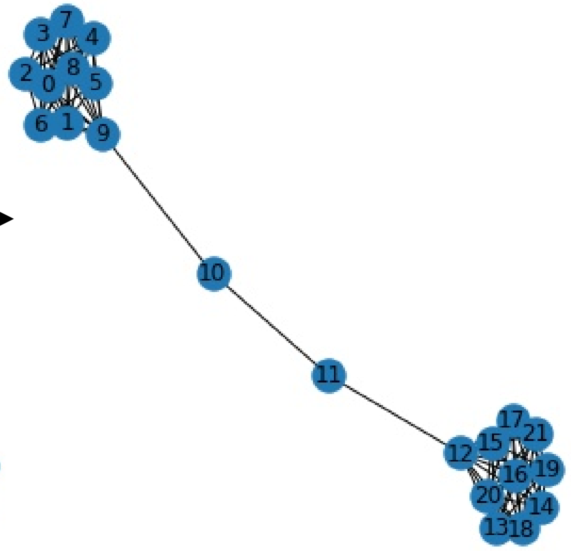
  (e.g., "online eigenvector calculator")
- Obtain projection



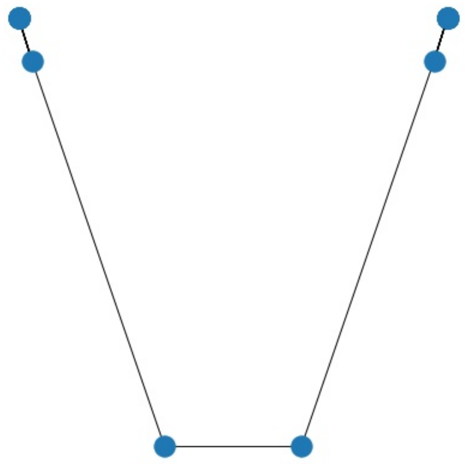

Link to spreadsheet: https://upfbarcelona.padlet.org/chato/shyq9m6f2g2dh1bw

# A barbell graph in $R^2$ (code)

```
B = nx.barbell_graph(10,2)

plt.figure(figsize=(6,6))
nx.draw_networkx(B)
_ = plt.show()
```

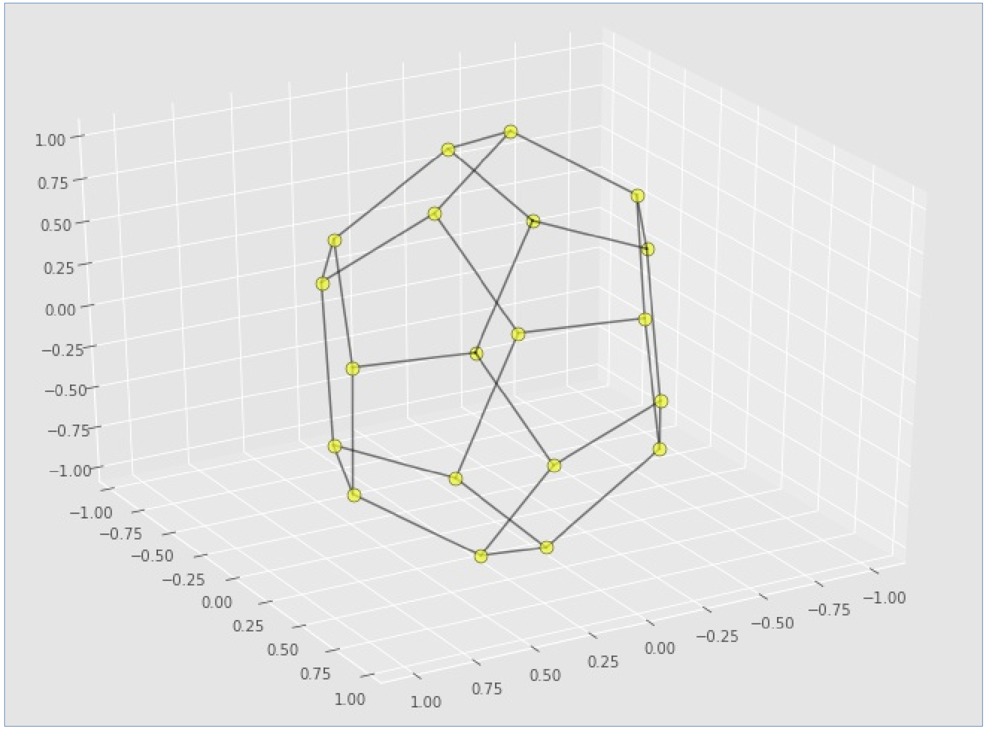



```
plt.figure(figsize=(6,6))
nx.draw_spectral(B)
_ = plt.show()
```



# Dodecahedral graph in 3D

```
g = nx.dodecahedral_graph()
pos = nx.spectral_layout(g, dim=3)
network_plot_3D_alt(g, 60, pos)
```

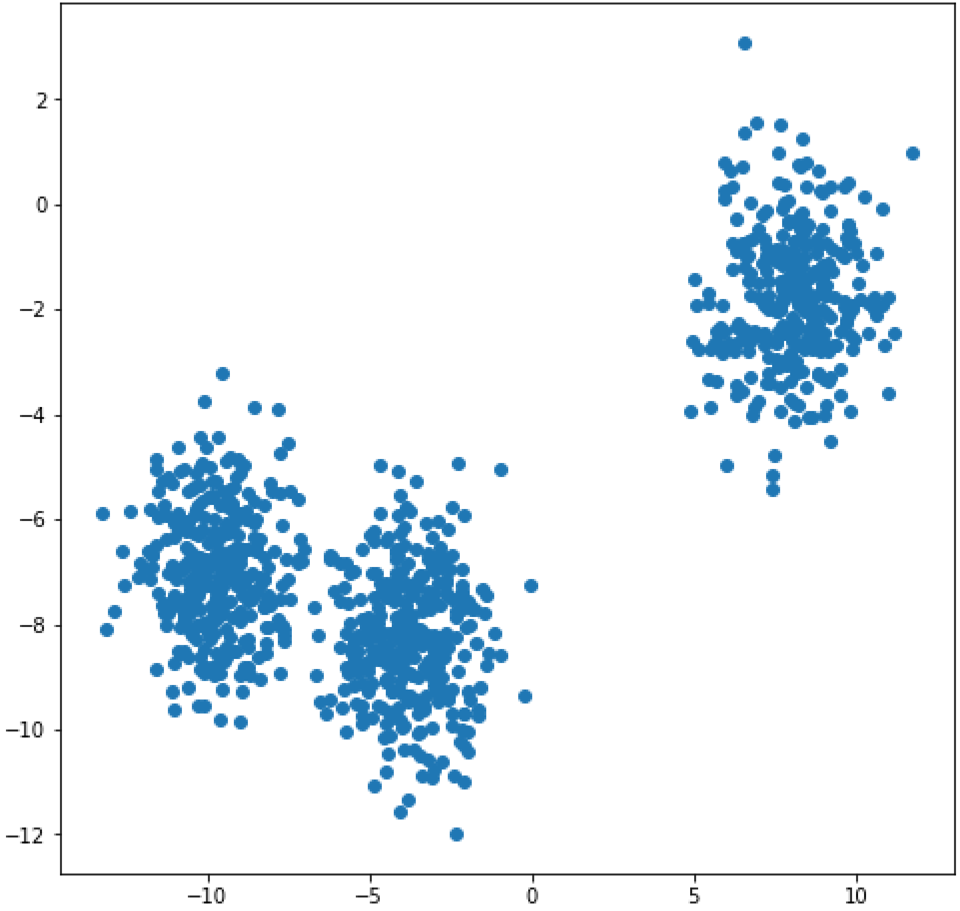



https://www.idtools.com.au/3d-network-graphs-python-mplot3d-toolkit/

# Application: spectral clustering

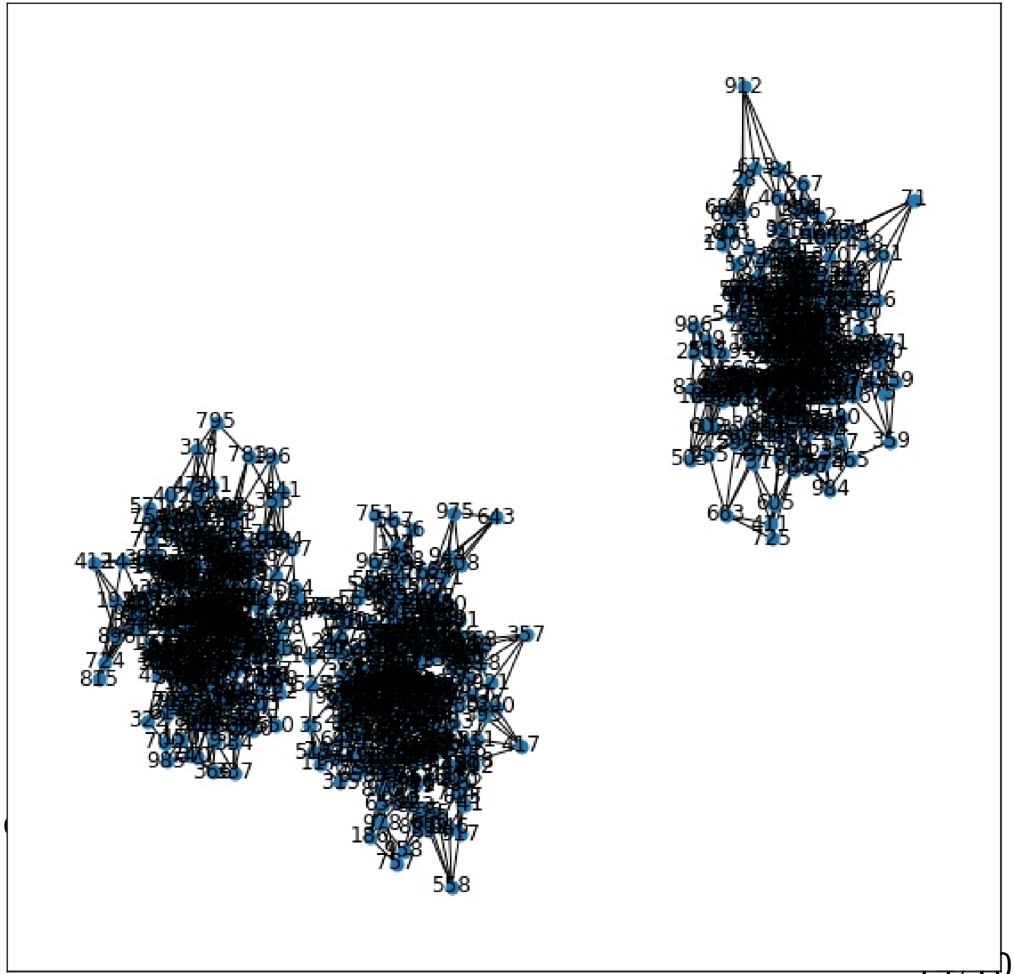
# Generating data

```
from sklearn.datasets import
   make_blobs

N = 1000

x, _ = make_blobs(
   n_samples=N,
   centers=3,
   cluster_std=1.2)


plt.figure(figsize=(8,8))

plt.scatter(x[:,0], x[:,1])

plt.show()
```



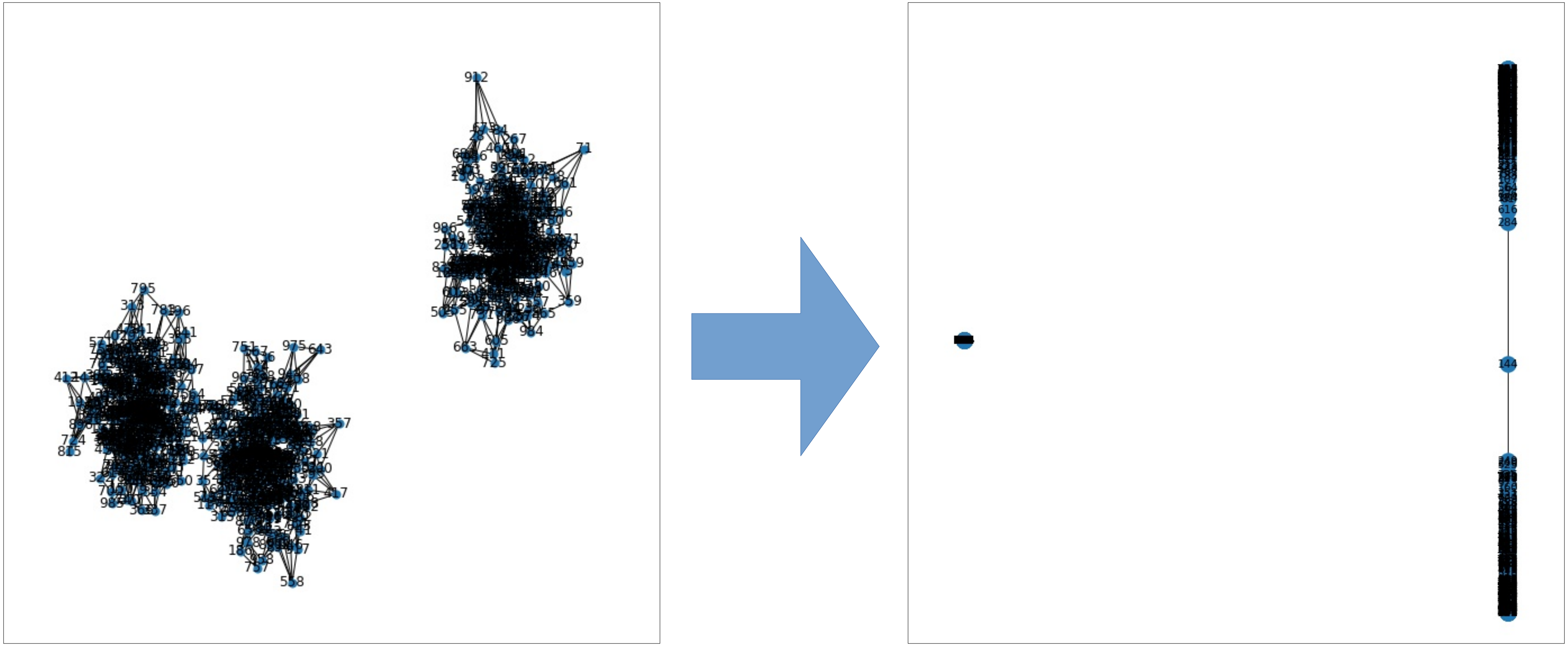
# Connect nodes to k=5 nearest neighbors

```
from sklearn.neighbors
   import NearestNeighbors

nbrs = NearestNeighbors(
    n_neighbors=6,         # includes self
    algorithm='ball_tree')
    .fit(x)

distances, neighbors =
    nbrs.kneighbors(x)

G = nx.Graph()

for neighbor_list in neighbors:

    source_node = neighbor_list[0]

    for target_index in range(1,
        len(neighbor_list)):

        target_node = neighbor_list[target_inde

        G.add_edge(source_node, target_node)
```



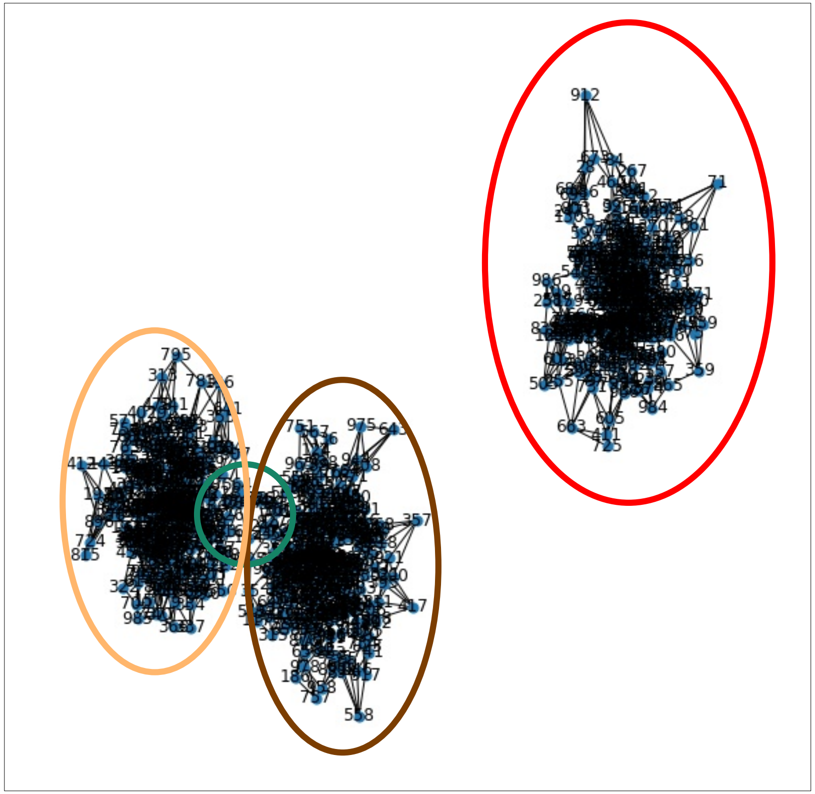
# Perform spectral embedding

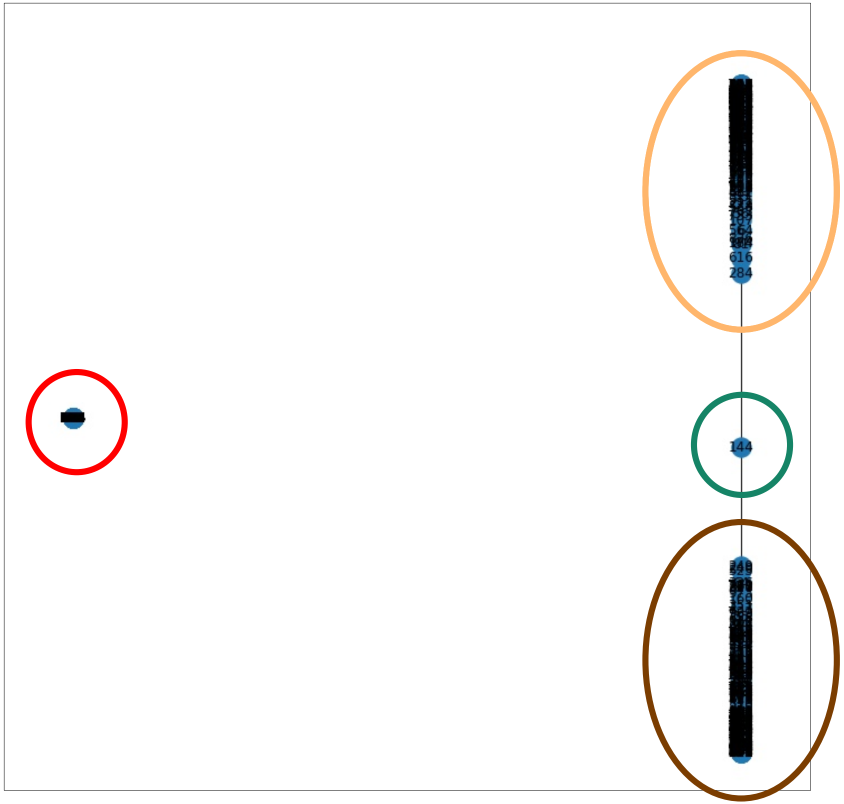
```
nx.draw_spectral(G, with_labels=True)
```



# Perform spectral embedding

```
nx.draw_spectral(G, with_labels=True)
```





# Summary

# Things to remember

- Graph Laplacian
- Laplacian and graph components
- Spectral graph embedding

# Exercises for this topic

- Mining of Massive Datasets (2014) by Leskovec et al.
  - Exercises 10.4.6