GRAPH LAPLACEANS AND THEIR APPLICATIONS

- Back to graphs - define graph Laplaceans
- Properties of graph Laplaceans
- Graph partitioning –
- Introduction to clustering

Graph Laplaceans - Definition

➤ "Laplace-type" matrices associated with general undirected graphs – useful in many applications

➤ Given a graph $G = (V, E)$ define

- A matrix W of weights w_{ij} for each edge
- Assume $w_{ij} \geq 0$,, $w_{ii} = 0$, and $w_{ij} = w_{ji}\ \forall (i, j)$
- The diagonal matrix $D = diag(d_i)$ with $d_i = \sum_{j \neq i} w_{ij}$

➤ Corresponding *graph Laplacean* of G is:

$$L = D - W$$

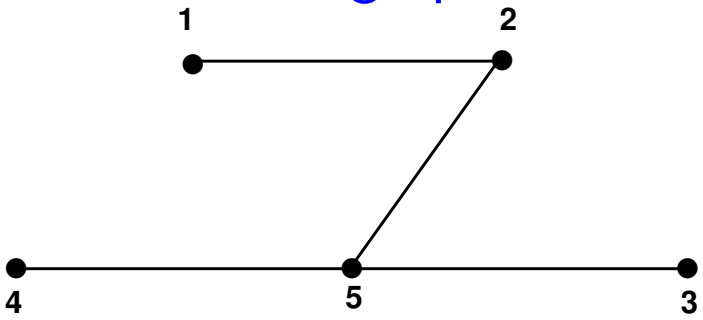
➤ Gershgorin's theorem $\longrightarrow L$ is positive semidefinite.

➤ Simplest case:

$$w_{ij} = \begin{cases} 1 \text{ if } (i,j) \in E \& i \neq j \\ 0 \quad \text{else} \end{cases} \qquad D = \text{diag}\left[d_i = \sum_{j \neq i} w_{ij} \right]$$

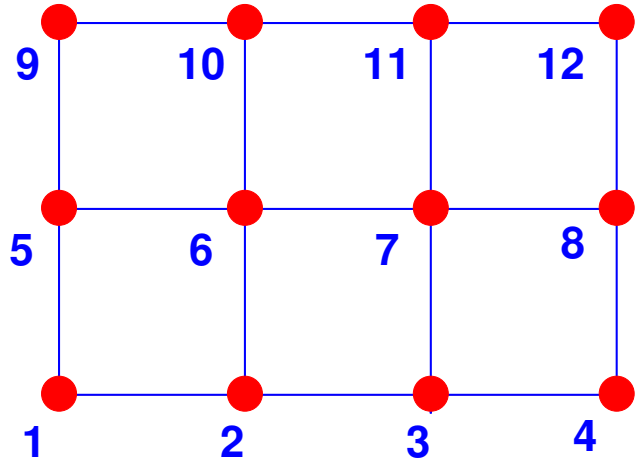
Example:

Consider the graph



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

✍1 Define the graph Laplacean for the graph associated with the simple mesh shown next. [use the simple weights of 0 or 1]. What is the difference with the discretization of the Laplace operator for case when mesh is the same as this graph?



Proposition:

(i) $\boldsymbol{L}$ is symmetric semi-positive definite.

(ii) $\boldsymbol{L}$ is singular with $\mathbb{1}$ as a null vector.

(iii) If $\boldsymbol{G}$ is connected, then $\mathbf{Null}(\boldsymbol{L}) = \mathbf{span}\{\mathbb{1}\}$

(iv) If $\boldsymbol{G}$ has $\boldsymbol{k} > 1$ connected components $\boldsymbol{G_1}, \boldsymbol{G_2}, \cdots, \boldsymbol{G_k}$, then the nullity of $\boldsymbol{L}$ is $\boldsymbol{k}$ and $\mathbf{Null}(\boldsymbol{L})$ is spanned by the vectors $\boldsymbol{z^{(j)}}$, $\boldsymbol{j} = 1, \cdots, \boldsymbol{k}$ defined by:

$$(\boldsymbol{z^{(j)}})_i = \begin{cases} \mathbf{1} \text{ if } \boldsymbol{i} \in \boldsymbol{G_j} \\ \mathbf{0} \text{ if not.} \end{cases}$$

Proof: (i) and (ii) seen earlier and are trivial. (iii) Clearly $\boldsymbol{u} = \mathbb{1}$ is a null vector for $\boldsymbol{L}$. The vector $\boldsymbol{D}^{-1/2}\boldsymbol{u}$ is an eigenvector for the matrix $\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ associated with the smallest eigenvalue. It is also an eigenvector for $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ associated with the largest eigenvalue. By the Perron Frobenius theorem this is a simple eigenvalue... (iv) Can be proved from the fact that $\boldsymbol{L}$ can be written as a direct sum of the Laplacian matrices for $\boldsymbol{G}_1, \cdots, \boldsymbol{G}_k$. ■

A few properties of graph Laplaceans

Define: oriented incidence matrix $\boldsymbol{H}$: (1)First orient the edges $i \sim j$ into $i \rightarrow j$ or $j \rightarrow i$. (2) Rows of $\boldsymbol{H}$ indexed by vertices of $\boldsymbol{G}$. Columns indexed by edges. (3) For each (i, j) in $\boldsymbol{E}$, define the corresponding column in $\boldsymbol{H}$ as $\sqrt{w(i,j)}(e_i - e_j)$.

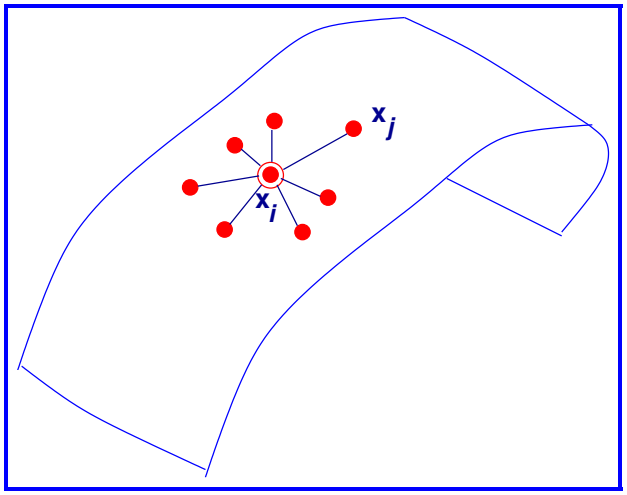
Example: In previous example (P. 11-3) orient $i \rightarrow j$ so that $j > i$ [lower triangular matrix representation].
Then matrix $\boldsymbol{H}$ is: $\longrightarrow$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 \end{bmatrix}$$

Property 1 $\boxed{L = HH^T}$

✍2 Re-prove part (iv) of previous proposition by using this property.

A few properties of graph Laplaceans



Strong relation between $x^T L x$ and local distances between entries of x

➤ Let $L =$ any matrix s.t. $L = D - W$, with $D = diag(d_i)$ and

$$w_{ij} \geq 0, \qquad d_i = \sum_{j \neq i} w_{ij}$$

Property 2: for any $x \in \mathbb{R}^n$:

$$x^\top L x = \frac{1}{2} \sum_{i,j} w_{ij} |x_i - x_j|^2$$

Property 3: (generalization) for any $\boldsymbol{Y} \in \mathbb{R}^{d \times n}$:

$$\mathrm{Tr}\left[\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^\top\right] = \frac{\mathbf{1}}{\mathbf{2}} \sum_{i,j} \boldsymbol{w}_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$$

➤ Note: $\boldsymbol{y}_j = \boldsymbol{j}$-th colunm of $\boldsymbol{Y}$. Usually $\boldsymbol{d} < \boldsymbol{n}$. Each column can represent a data sample.

Property 4: For the particular $\boldsymbol{L} = \boldsymbol{I} - \frac{\mathbf{1}}{\boldsymbol{n}} \mathbb{1}\,\mathbb{1}^\top$

$$\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^\top = \bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^\top == \boldsymbol{n} \times \text{Covariance matrix}$$

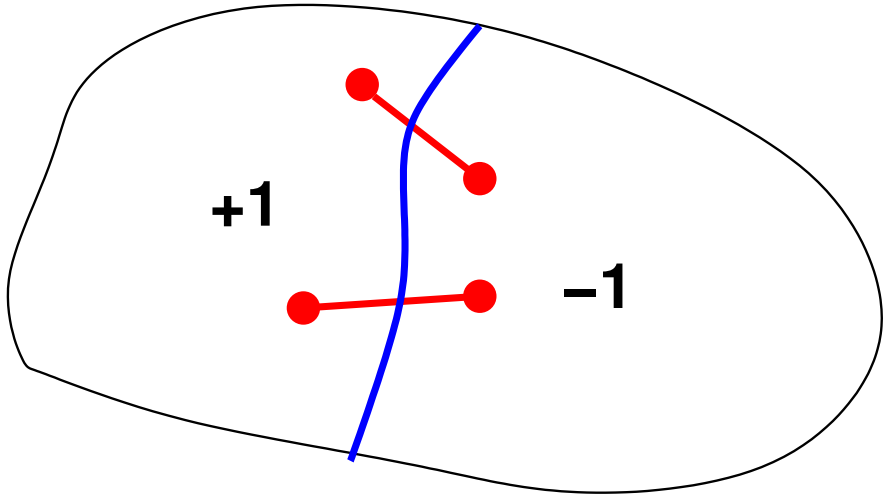
Property 5: $\boldsymbol{L}$ is singular and admits the null vector $\mathbb{1}$ =ones(n,1)

Property 6: (Graph partitioning) Consider situation when $w_{ij} \in \{0, 1\}$. If x is a vector of signs (± 1) then

$$x^\top L x = 4 \times (\text{'number of edge cuts'})$$

edge-cut = pair (i, j) with $x_i \neq x_j$

- Consequence: Can be used to partition graphs



➤ Would like to minimize $(\boldsymbol{Lx}, \boldsymbol{x})$ subject to $\boldsymbol{x} \in \{-\mathbf{1}, \mathbf{1}\}^n$ and $\boldsymbol{e}^T\boldsymbol{x} = \mathbf{0}$ [balanced sets]

➤ WIl solve a relaxed form of this problem

✍3 What if we replace $\boldsymbol{x}$ by a vector of ones (representing one partition) and zeros (representing the other)?

✍4 Let $\boldsymbol{x}$ be any vector and $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\alpha}\ \mathbb{1}$ and $\boldsymbol{L}$ a graph Laplacean. Compare $(\boldsymbol{Lx}, \boldsymbol{x})$ with $(\boldsymbol{Ly}, \boldsymbol{y})$.

➤ Consider any symmetric (real) matrix A with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and eigenvectors $u_1, \cdots, u_n$

➤ Recall that:
(Min reached for $x = u_1$)

$$\min_{x \in \mathbb{R}^n} \frac{(Ax, x)}{(x, x)} = \lambda_1$$

➤ In addition:
(Min reached for $x = u_2$)

$$\min_{x \perp u_1} \frac{(Ax, x)}{(x, x)} = \lambda_2$$

➤ For a graph Laplacean $u_1 = \mathbb{1} =$ vector of all ones and

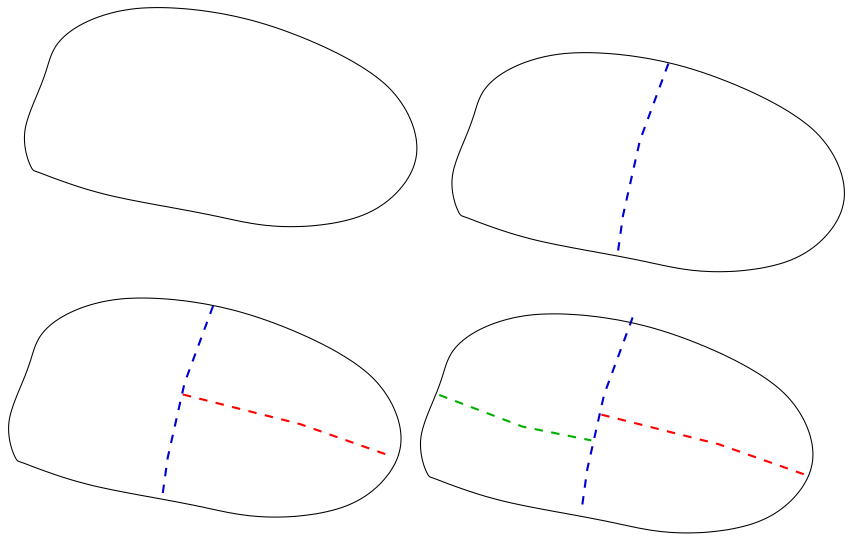
➤ ...vector u_2 is called the Fiedler vector. It solves a relaxed form of the problem -

$$\min_{x \in \{-1,1\}^n;\ \mathbb{1}^T x = 0} \frac{(Lx, x)}{(x, x)} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n;\ \mathbb{1}^T x = 0} \frac{(Lx, x)}{(x, x)}$$

- Define $v = u_2$ then $lab = sign(v - med(v))$

Recursive Spectral Bisection

1. Form graph Laplacean
2. Partition graph in 2 based on Fielder vector
3. Partition largest subgraph in two recursively ...
4. ... Until the desired number of partitions is reached

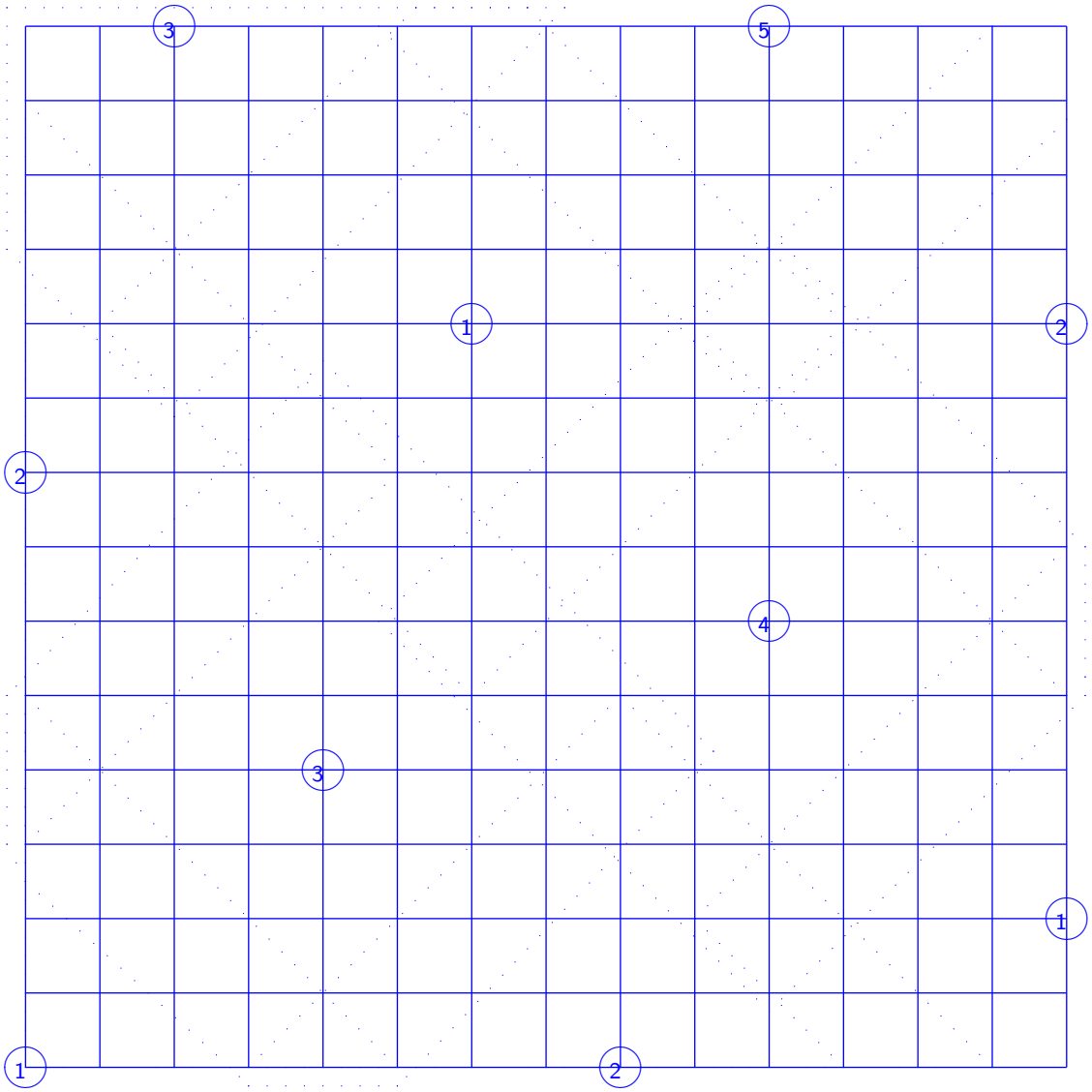


Three approaches to graph partitioning:

1. Spectral methods - Just seen + add Recursive Spectral Bisection.
2. Geometric techniques. Coordinates are required. [Houstis & Rice et al., Miller, Vavasis, Teng et al.]
3. Graph Theory techniques – multilevel,... [use graph, but no coordinates]
   - Currently best known technique is Metis (multi-level algorithm)
   - Simplest idea: Recursive Graph Bisection; Nested dissection (George & Liu, 1980; Liu 1992]
   - Advantages: simplicity – no coordinates required

Example of a graph theory approach

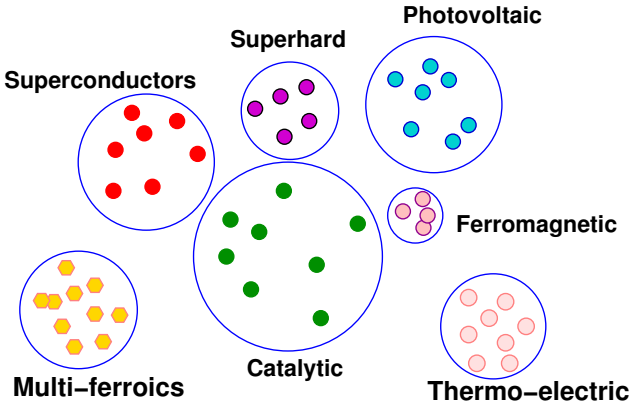
- Level Set Expansion Algorithm
- Given: p nodes 'uniformly' spread in the graph (roughly same distance from one another).
- Method: Perform a level-set traversal (BFS) from each node simultaneously.
- Best described for an example on a 15×15 five – point Finite Difference grid.
- See [Goehring-Saad '94, See Cai-Saad '95]
- Approach also known under the name 'bubble' algorithm and implemented in some packages [Party, DibaP]


3
5
1
2
2
4
3
1
1
2

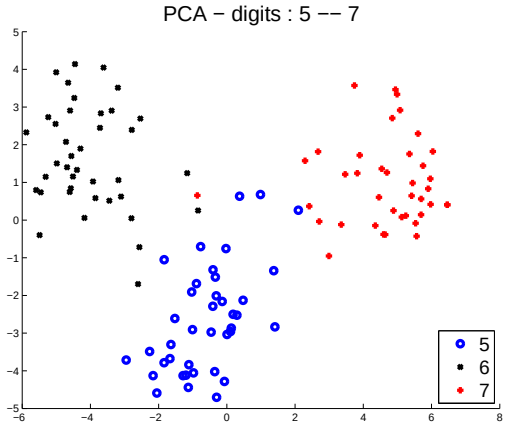
Clustering

➤ Problem: we are given n data items: $x_1, x_2, \cdots, x_n$. Would like to *'cluster'* them, i.e., group them so that each group or cluster contains items that are similar in some sense.

➤ Example: materials



➤ Example: Digits



➤ Refer to each group as a 'cluster' or a 'class'

➤ 'Unsupervised learning'

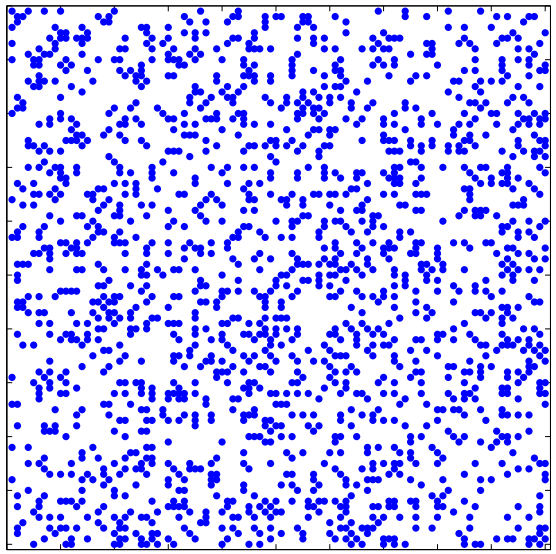
What is Unsupervised learning?

"Unsupervised learning" : methods do not exploit labeled data

- Example of digits: perform a 2-D projection
- Images of same digit tend to cluster (more or less)
- Such 2-D representations are popular for visualization
- Can also try to find natural clusters in data, e.g., in materials
- Basic clusterning technique: K-means

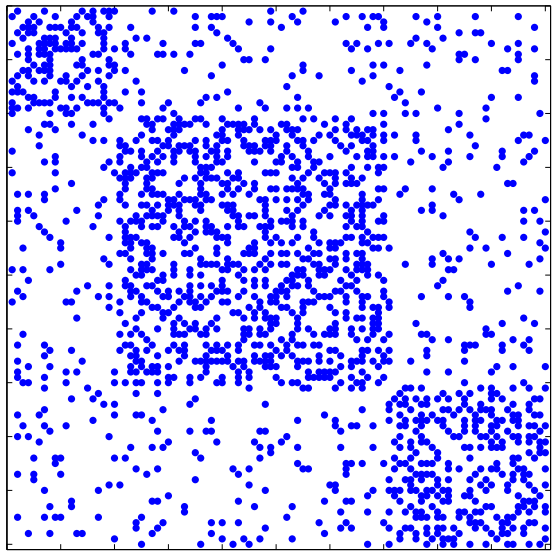
Example: Community Detection

- Communities modeled by an 'affinity' graph [e.g., 'user A sends frequent e-mails to user B']
- Adjacency Graph represented by a sparse matrix



← Original matrix

Goal: Find ordering so blocks are as dense as possible →



- Use 'blocking' techniques for sparse matrices
- Advantage of this viewpoint: need not know # of clusters.

[data: www-personal.umich.edu/~mejn/netdata/]

Example of application Data set from :

http://www-personal.umich.edu/~mejn/netdata/

➤ Network connecting bloggers of different political orientations [2004 US presidentual election]

➤ 'Communities': liberal vs. conservative

➤ Graph: $\mathbf{1,490}$ vertices (blogs) : first $\mathbf{758}$: liberal, rest: conservative.

➤ Edge: $i \longrightarrow j$: a citation between blogs i and j

➤ Blocking algorithm (Density theshold=0.4): subgraphs [note: density $= |E|/|V|^2$.]

➤ Smaller subgraph: conservative blogs, larger one: liberals