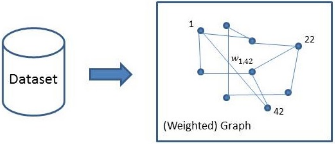
Lecture 1: From Data to Graphs, Weighted Graphs and Graph Laplacian

Radu Balan

February 5, 2018

From Data to Graphs
Datasets



Datasets diversity:

- Social Networks: Set of individuals ("agents", "actors") interacting with each other (e.g., Facebook, Twitter, joint paper authorship, etc.)
- Communication Networks: Devices (phones, laptops, cars) communicating with each other (emails, spectrum occupancy)
- Biological Networks: Macroscale: How animals interact with each other; Microscale: How proteins interact with each other.
- Databases of signals: speech, images, movies; graph relationship tends to reflect signal similarity: the higher the similarity, the larger the weight.

Weighted Graphs
W

The main goal this lecture is to introduce basic concepts of weighted and undirected graphs, its associated graph Laplacian, and methods to build weight matrices.

Graphs (and weights) reflect either similarity between nodes, or functional dependency.

- SIMILARITY: Distance, similarity between nodes $\Rightarrow$ weight $w_{i,j}$
- PREDICTIVE: How node i is predicted by its nighbor node j $\Rightarrow$ weight $w_{i,j}$

Definitions

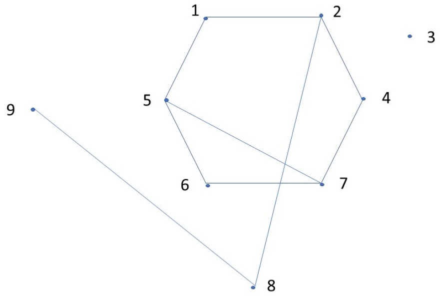
$G = (\mathcal{V}, \mathcal{E})$ and $G = (\mathcal{V}, \mathcal{E}, w)$

An *undirected graph* G is given by two pieces of information: a set of *vertices* $\mathcal{V}$ and a set of *edges* $\mathcal{E}$, $G = (\mathcal{V}, \mathcal{E})$.
A *weighted graph* has three pieces of information: $G = (\mathcal{V}, \mathcal{E}, w)$, the set of vertices $\mathcal{V}$, the set of edges $\mathcal{E}$, and a *weight function* $w : \mathcal{E} \to \mathbb{R}$.

Definitions

$G = (\mathcal{V}, \mathcal{E})$ and $G = (\mathcal{V}, \mathcal{E}, w)$

An *undirected graph* G is given by two pieces of information: a set of *vertices* $\mathcal{V}$ and a set of *edges* $\mathcal{E}$, $G = (\mathcal{V}, \mathcal{E})$.
A *weighted graph* has three pieces of information: $G = (\mathcal{V}, \mathcal{E}, w)$, the set of vertices $\mathcal{V}$, the set of edges $\mathcal{E}$, and a *weight function* $w : \mathcal{E} \to \mathbb{R}$.



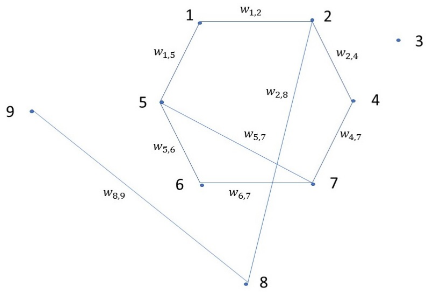
$\mathcal{V} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
$\mathcal{E} = \{(1, 2), (2, 4), (4, 7), (6, 7), (1, 5), (5, 6), (5, 7), (2, 8), (8, 9)\}$
9 *vertices* , 9 *edges*
Undirected graph, edges are not oriented. Thus $(1, 2) \sim (2, 1)$.

Definitions
$G = (\mathcal{V}, \mathcal{E})$

A weighted graph $G = (\mathcal{V}, \mathcal{E}, w)$ can be directed or undirected depending whether $w(i,j) = w(j,i)$.
Symmetric weights == Undirected graphs

Definitions
$G = (\mathcal{V}, \mathcal{E})$

A weighted graph $G = (\mathcal{V}, \mathcal{E}, w)$ can be directed or undirected depending whether $w(i,j) = w(j,i)$.
Symmetric weights == Undirected graphs



Example of a Weighted Graph
UCINET IV Datasets: Bernard & Killworth Office

Available online at:
http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm
Content: Two 40 $\times$ 40 matrices: symmetric (B) and non-symmetric (C)
Bernard & Killworth, later with the help of Sailer, collected five sets of data on human interactions in bounded groups and on the actors' ability to recall those interactions. In each study they obtained measures of social interaction among all actors, and ranking data based on the subjects' memory of those interactions. The names of all cognitive (recall) matrices end in C, those of the behavioral measures in B.
These data concern interactions in a small business office, again recorded by an "unobtrusive" observer. Observations were made as the observer patrolled a fixed route through the office every fifteen minutes during two four-day periods. BKOFFB contains the observed frequency of interactions; BKOFFC contains rankings of interaction frequency as recalled by the employees over the two-week period.

Example of a Weighted Graph
UCINET IV Datasets: Bernard & Killworth Office

bkoff.dat
DL
N=40 NM=2
FORMAT = FULLMATRIX DIAGONAL PRESENT
LEVEL LABELS:
BKOFFB
BKOFFC
DATA:
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 2 1 0
0 0 4 8 0 3 3 0 1 1 0 0 3 0 0 0 2 1 1 0 0 2 2 0 0 0
0 1 0 0 2 9 0 1 1 0 0 3 0 0
0 4 0 2 1 0 14 0 0 0 1 0 0 1 1 1 0 0 2 0 0 10 0 1 1 0
0 0 0 4 0 1 0 1 8 1 0 0 2 1
...

Example of a Weighted Graph

UCINET IV Datasets: Bernard & Killworth Office

...

0 0 1 3 0 0 2 0 0 0 0 1 0 5 0 0 0 0 0 0 0 5 0 0 0 0

0 1 1 0 0 0 2 4 5 0 0 0 0 0

0 27 3 36 23 34 14 19 13 9 3 26 21 25 1 8 22 12 11 4 2 37 35 17 5 20

7 33 32 39 38 16 28 30 29 24 6 10 18 31

...

29 38 17 4 31 37 6 35 36 22 17 24 39 20 19 26 12 30 32 28 25 1 18 14 33 34

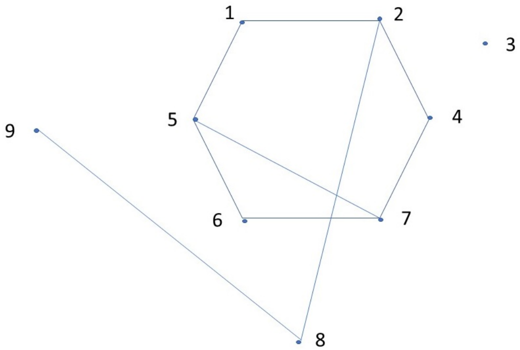
27 8 9 21 11 10 5 3 2 15 23 16 13 0

Definitions

Paths

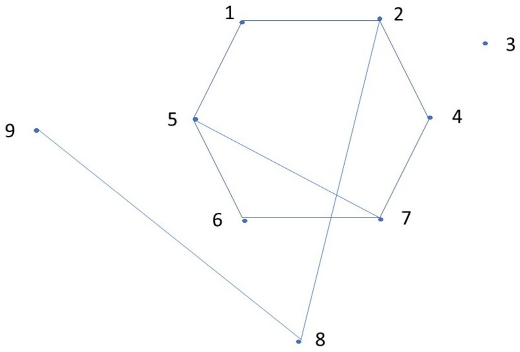
Concept: A path is a sequence of edges where the right vertex of one edge coincides with the left vertex of the following edge.

Example:



Definitions
Paths

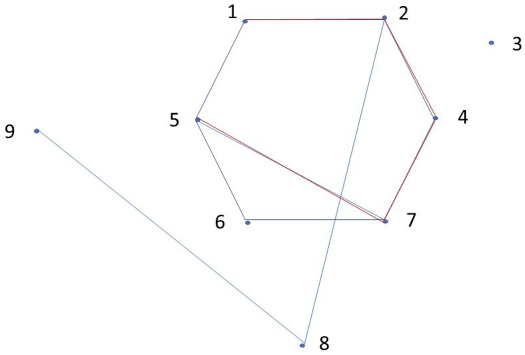
Concept: A path is a sequence of edges where the right vertex of one edge coincides with the left vertex of the following edge.
Example:



$$\{(1,2),(2,4),(4,7),(7,5)\} =$$
$$= \{(1,2),(2,4),(4,7),(5,7)\}$$

Definitions
Paths

Concept: A path is a sequence of edges where the right vertex of one edge coincides with the left vertex of the following edge.
Example:



$$\{(1,2),(2,4),(4,7),(7,5)\} =$$
$$= \{(1,2),(2,4),(4,7),(5,7)\}$$

Definitions
Graph Attributes

Graph Attributes (Properties):

- Connected Graphs: Graphs where any two distinct vertices can be connected through a path.

Definitions
Graph Attributes

Graph Attributes (Properties):

- Connected Graphs: Graphs where any two distinct vertices can be connected through a path.
- Complete (or Totally Connected) Graphs: Graphs where any two distinct vertices are connected by an edge.

Definitions

Graph Attributes

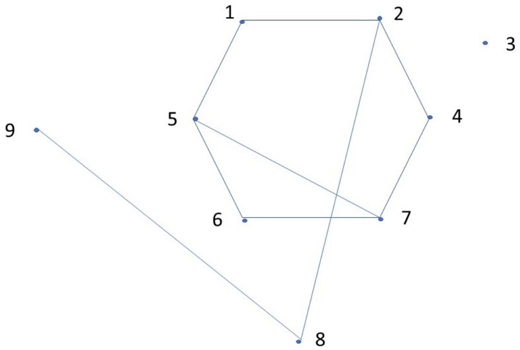
Graph Attributes (Properties):

- Connected Graphs: Graphs where any two distinct vertices can be connected through a path.
- Complete (or Totally Connected) Graphs: Graphs where any two distinct vertices are connected by an edge.

A complete graph with n vertices has $m = \begin{pmatrix} n \\ 2 \end{pmatrix} = \frac{n(n-1)}{2}$ edges.

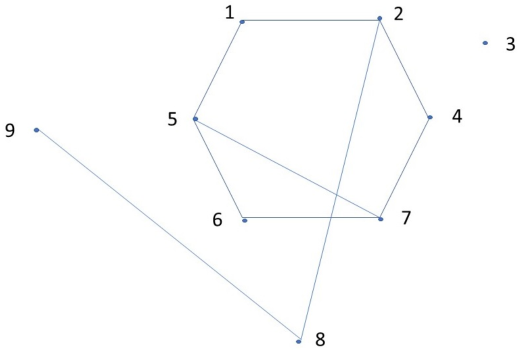
Definitions
Graph Attributes

Example:



Definitions
Graph Attributes

Example:



- This graph is not connected.
- It is not complete.
- It is the union of two connected graphs.

Definitions
Metric

Distance between vertices: For two vertices x, y, the distance $d(x, y)$ is the length of the shortest path connecting x and y. If $x = y$ then $d(x, x) = 0$.

Definitions
Metric

Distance between vertices: For two vertices x, y, the distance $d(x, y)$ is the length of the shortest path connecting x and y. If $x = y$ then $d(x, x) = 0$.
In a connected graph the distance between any two vertices is finite.
In a complete graph the distance between any two distinct vertices is 1.

Definitions
Metric

Distance between vertices: For two vertices x, y, the distance $d(x, y)$ is the length of the shortest path connecting x and y. If $x = y$ then $d(x, x) = 0$.
In a connected graph the distance between any two vertices is finite.
In a complete graph the distance between any two distinct vertices is 1.
The converses are also true:

1. If $\forall x, y \in \mathcal{E}$, $d(x, y) < \infty$ then $(\mathcal{V}, \mathcal{E})$ is connected.
2. If $\forall x \neq y \in \mathcal{E}$, $d(x, y) = 1$ then $(\mathcal{V}, \mathcal{E})$ is complete.

Definitions
Metric

Graph diameter: The diameter of a graph $G = (\mathcal{V}, \mathcal{E})$ is the largest distance between two vertices of the graph:

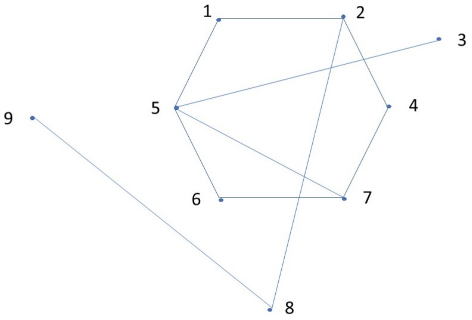
$$D(G) = \max_{x,y \in \mathcal{V}} d(x, y)$$

Definitions
Metric

Graph diameter: The diameter of a graph $G = (\mathcal{V}, \mathcal{E})$ is the largest distance between two vertices of the graph:

$$D(G) = \max_{x,y \in \mathcal{V}} d(x, y)$$

Example:

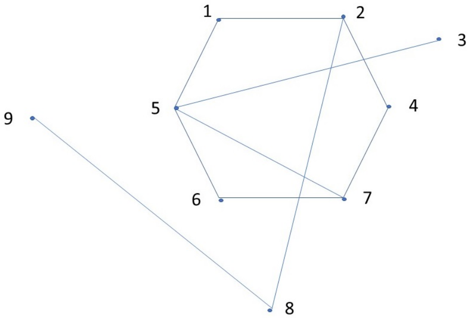


Definitions
Metric

Graph diameter: The diameter of a graph $G = (\mathcal{V}, \mathcal{E})$ is the largest distance between two vertices of the graph:

$$D(G) = \max_{x,y \in \mathcal{V}} d(x, y)$$

Example:



$$D = 5 = d(6, 9) = d(3, 9)$$

Definitions

The Adjacency Matrix

For a graph $G = (\mathcal{V}, \mathcal{E})$ the adjacency matrix is the $n \times n$ matrix A defined by:

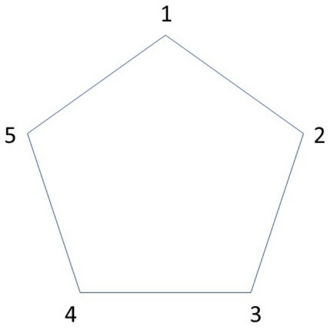
$$A_{i,j} = \begin{cases} 1 & if \quad (i,j) \in \mathcal{E} \\ 0 & \quad otherwise \end{cases}$$

Definitions
The Adjacency Matrix

For a graph $G = (\mathcal{V}, \mathcal{E})$ the adjacency matrix is the $n \times n$ matrix A defined by:

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

Example:

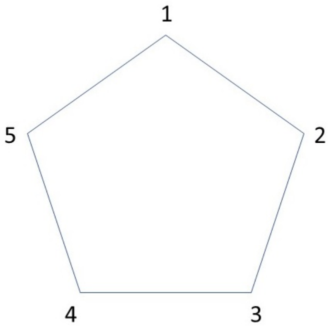


Definitions
The Adjacency Matrix

For a graph $G = (\mathcal{V}, \mathcal{E})$ the adjacency matrix is the $n \times n$ matrix A defined by:

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

Example:



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

Definitions
The Adjacency Matrix

For undirected graphs the adjacency matrix is always symmetric:

$$A^T = A$$

For directed graphs the adjacency matrix may not be symmetric.

Definitions
The Adjacency Matrix

For undirected graphs the adjacency matrix is always symmetric:

$$A^T = A$$

For directed graphs the adjacency matrix may not be symmetric.
For weighted graphs $G = (\mathcal{V}, \mathcal{E}, W)$, the weight matrix W is simply given by

$$W_{i,j} = \begin{cases} w_{i,j} & if \quad (i,j) \in \mathcal{E} \\ 0 & \quad otherwise \end{cases}$$

Degree Matrix
$d(v)$ and D

For an undirected graph $G = (\mathcal{V}, \mathcal{E})$, let $d_v = d(v)$ denote the number of edges at vertex $v \in \mathcal{V}$. The number $d(v)$ is called the degree (or valency) of vertex v. The n-vector $d = (d_1, \cdots, d_n)^T$ can be computed by

$$d = A \cdot 1$$

where A denotes the adjacency matrix, and 1 is the vector of 1's, ones(n,1).
Let D denote the diagonal matrix formed from the degree vector d: $D_{k,k} = d_k$, $k = 1, 2, \cdots, n$. D is called the *degree matrix*.

Degree Matrix
$d(v)$ and D

For an undirected graph $G = (\mathcal{V}, \mathcal{E})$, let $d_v = d(v)$ denote the number of edges at vertex $v \in \mathcal{V}$. The number $d(v)$ is called the degree (or valency) of vertex v. The n-vector $d = (d_1, \cdots, d_n)^T$ can be computed by

$$d = A \cdot 1$$

where A denotes the adjacency matrix, and 1 is the vector of 1's, ones(n,1).
Let D denote the diagonal matrix formed from the degree vector d: $D_{k,k} = d_k$, $k = 1, 2, \cdots, n$. D is called the *degree matrix*.
Key obervation: $(D - A) \cdot 1 = 0$ always holds. This means the matrix $D - A$ has a non-zero null-space (kernel), hence it is rank deficient.

Degree Matrix
$d(v)$ and D

For an undirected graph $G = (\mathcal{V}, \mathcal{E})$, let $d_v = d(v)$ denote the number of edges at vertex $v \in \mathcal{V}$. The number $d(v)$ is called the degree (or valency) of vertex v. The n-vector $d = (d_1, \cdots, d_n)^T$ can be computed by

$$d = A \cdot 1$$

where A denotes the adjacency matrix, and 1 is the vector of 1's, ones(n,1).
Let D denote the diagonal matrix formed from the degree vector d: $D_{k,k} = d_k$, $k = 1, 2, \cdots, n$. D is called the *degree matrix*.
Key obervation: $(D - A) \cdot 1 = 0$ always holds. This means the matrix $D - A$ has a non-zero null-space (kernel), hence it is rank deficient.
Second observation: The dimension of the null-space of $D - A$ equals the number of connected components in the graph.

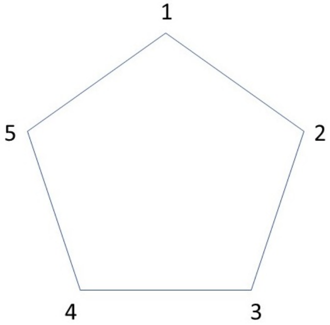
Vertex Degree
Matrix D

For an undirected graph $G = (\mathcal{V}, \mathcal{E})$ of n vertices, we denote by D the $n \times n$ diagonal matrix of degrees: $D_{i,i} = d(i)$.

Vertex Degree

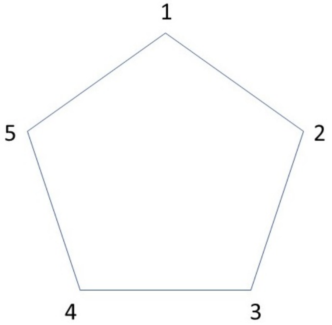
Matrix D

For an undirected graph $G = (\mathcal{V}, \mathcal{E})$ of n vertices, we denote by D the $n \times n$ diagonal matrix of degrees: $D_{i,i} = d(i)$. Example:



Vertex Degree
Matrix D

For an undirected graph $G = (\mathcal{V}, \mathcal{E})$ of n vertices, we denote by D the $n \times n$ diagonal matrix of degrees: $D_{i,i} = d(i)$. Example:



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

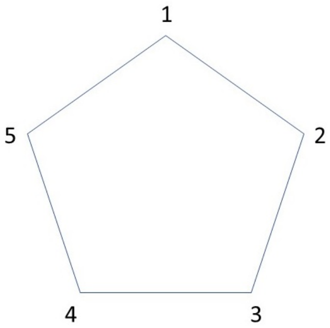
Graph Laplacian

Δ

For a graph $G = (\mathcal{V}, \mathcal{E})$ the graph Laplacian is the $n \times n$ symmetric matrix Δ defined by:

$$\Delta = D - A$$

Example:

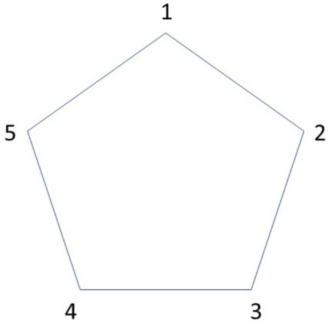


Graph Laplacian
Δ

For a graph $G = (\mathcal{V}, \mathcal{E})$ the graph Laplacian is the $n \times n$ symmetric matrix Δ defined by:

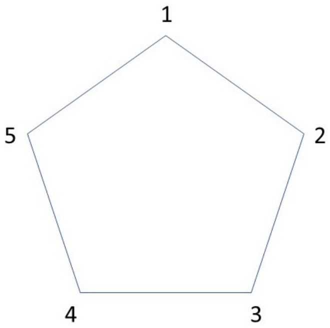
$$\Delta = D - A$$

Example:



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

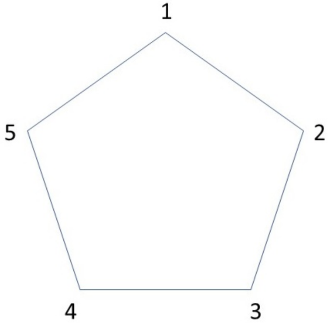
Graph Laplacian
Intuition



Assume $x = [x_1, x_2, x_3, x_4, x_5]^T$ is a signal of five components defined over the graph. The *Dirichlet* energy E, is defined as

$$E = \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2 =$$

Graph Laplacian
Intuition

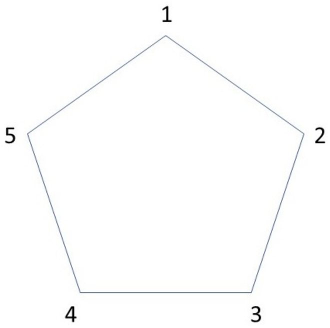


Assume $x = [x_1, x_2, x_3, x_4, x_5]^T$ is a signal of five components defined over the graph. The *Dirichlet* energy E, is defined as

$$E = \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2 = (x_2 - x_1)^2 + (x_3 - x_2)^2 +$$

$$+(x_4 - x_3)^2 + (x_5 - x_4)^2 + (x_1 - x_5)^2.$$

Graph Laplacian
Intuition



Assume $x = [x_1, x_2, x_3, x_4, x_5]^T$ is a signal of five components defined over the graph. The *Dirichlet* energy E, is defined as

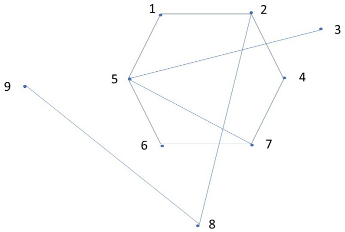
$$E = \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2 = (x_2 - x_1)^2 + (x_3 - x_2)^2 +$$

$$+(x_4 - x_3)^2 + (x_5 - x_4)^2 + (x_1 - x_5)^2.$$

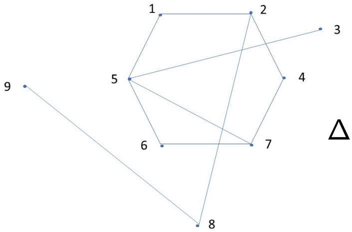
By regrouping the terms we obtain:

$$E = \langle \Delta x, x\rangle = x^T \Delta x = x^T (D - A) x$$

Graph Laplacian
Example



Graph Laplacian
Example



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

Normalized Laplacians
$\tilde{\Delta}$

Normalized Laplacian: (using pseudo-inverses)

$$\tilde{\Delta} = D^{-1/2}\Delta D^{-1/2} = I - D^{-1/2}AD^{-1/2}$$

$$\tilde{\Delta}_{i,j} = \begin{cases} 1 & \text{if} \quad i = j \text{ and } d_i > 0 \text{ (non} - \text{isolated vertex)} \\ -\frac{1}{\sqrt{d(i)d(j)}} & \text{if} \quad (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

Normalized Laplacians
$\tilde{\Delta}$

Normalized Laplacian: (using pseudo-inverses)

$$\tilde{\Delta} = D^{-1/2} \Delta D^{-1/2} = I - D^{-1/2} A D^{-1/2}$$

$$\tilde{\Delta}_{i,j} = \begin{cases} 1 & if \quad i = j \text{ and } d_i > 0 \text{ (non} - \text{isolated vertex)} \\ -\frac{1}{\sqrt{d(i)d(j)}} & if \quad (i,j) \in \mathcal{E} \\ 0 & otherwise \end{cases}$$

Normalized Asymmetric Laplacian:

$$L = D^{-1} \Delta = I - D^{-1} A$$

$$L_{i,j} = \begin{cases} 1 & if \quad i = j \text{ and } d_i > 0 \text{ (non} - \text{isolated vertex)} \\ -\frac{1}{d(i)} & if \quad (i,j) \in \mathcal{E} \\ 0 & otherwise \end{cases}$$

Normalized Laplacians
$\tilde{\Delta}$

Normalized Laplacian: (using pseudo-inverses)

$$\tilde{\Delta} = D^{-1/2}\Delta D^{-1/2} = I - D^{-1/2}AD^{-1/2}$$

$$\tilde{\Delta}_{i,j} = \begin{cases} 1 & if \quad i = j \text{ and } d_i > 0 \text{ (non} - \text{isolated vertex)} \\ -\frac{1}{\sqrt{d(i)d(j)}} & if \quad (i,j) \in \mathcal{E} \\ 0 & otherwise \end{cases}$$

Normalized Asymmetric Laplacian:

$$L = D^{-1}\Delta = I - D^{-1}A$$

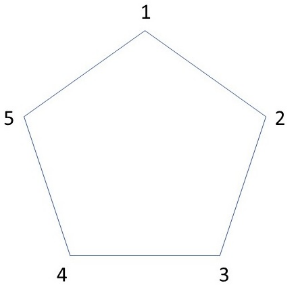
$$L_{i,j} = \begin{cases} 1 & if \quad i = j \text{ and } d_i > 0 \text{ (non} - \text{isolated vertex)} \\ -\frac{1}{d(i)} & if \quad (i,j) \in \mathcal{E} \\ 0 & otherwise \end{cases}$$

Note:

$$\Delta D^{-1} = I - AD^{-1} = L^T \quad ; \quad (D^{-1})_{kk} = (D^{-1/2})_{kk} = 0 \text{ if } d(k) = 0$$

Normalized Laplacians
Example

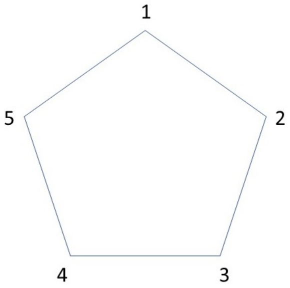
Example:



Normalized Laplacians

Example

Example:



$$\tilde{\Delta} = \begin{bmatrix} 1 & -0.5 & 0 & 0 & -0.5 \\ -0.5 & 1 & -0.5 & 0 & 0 \\ 0 & -0.5 & 1 & -0.5 & 0 \\ 0 & 0 & -0.5 & 1 & -0.5 \\ -0.5 & 0 & 0 & -0.5 & 1 \end{bmatrix}$$

Laplacian and Normalized Laplacian for Weighted Graphs

In the case of a weighted graph, $G = (\mathcal{V}, \mathcal{E}, w)$, the weight matrix W replaces the adjacency matrix A.

Laplacian and Normalized Laplacian for Weighted Graphs

In the case of a weighted graph, $G=(\mathcal{V},\mathcal{E},w)$, the weight matrix W replaces the adjacency matrix A.
The other matrices:

$$D = W\cdot 1 \quad , \quad D_{k,k}=\sum_{j\in\mathcal{V}} W_{k,j}$$

$$\Delta = D - W \quad , \quad \dim\ker(D-W) = \text{number connected components}$$

$$\tilde{\Delta} = D^{-1/2}\Delta D^{-1/2}$$

$$L = D^{-1}\Delta$$

where $D^{-1/2}$ and D^{-1} denote the diagonal matrices:

$$(D^{-1/2})_{k,k} = \begin{cases} \frac{1}{\sqrt{D_{k,k}}} & if \quad D_{k,k}>0 \\ 0 & if \quad D_{k,k}=0 \end{cases} \quad , \quad (D^{-1})_{k,k} = \begin{cases} \frac{1}{D_{k,k}} & if \quad D_{k,k}>0 \\ 0 & if \quad D_{k,k}=0 \end{cases}$$

Laplacian and Normalized Laplacian for Weighted Graphs
Dirichlet Energy

For symmetric (i.e., undirected) weighted graphs, the Dirichlet energy is defined as

$$E = \frac{1}{2}\sum_{i,j\in\mathcal{V}} w_{i,j}|x_i - x_j|^2$$

Expanding the square and grouping the terms together, the expression simplifies to

$$\sum_{i\in\mathcal{V}} |x_i|^2 \sum_j w_{ij} - \sum_{i,j\in\mathcal{V}} w_{i,j}x_i x_j = \langle Dx, x\rangle - \langle Wx, x\rangle = x^T(D-W)x.$$

Hence:

$$E = \frac{1}{2}\sum_{i,j\in\mathcal{V}} w_{i,j}|x_i - x_j|^2 = x^T \Delta x$$

where $\Delta = D - W$ is the weighted graph Laplacian.

Spectral Analysis
Eigenvalues and Eigenvectors

Recall the eigenvalues of a matrix T are the zeros of the characteristic polynomial:

$$p_T(z) = det(zI - T) = 0.$$

There are exactly n eigenvalues (including multiplicities) for a $n \times n$ matrix T. The set of eigenvalues is calles its *spectrum*.

Spectral Analysis
Eigenvalues and Eigenvectors

Recall the eigenvalues of a matrix T are the zeros of the characteristic polynomial:

$$p_T(z) = det(zI - T) = 0.$$

There are exactly n eigenvalues (including multiplicities) for a $n \times n$ matrix T. The set of eigenvalues is calles its *spectrum*.
If λ is an eigenvalue of T, then its associated eigenvector is the non-zero n-vector x such that $Tx = \lambda x$.

Spectral Analysis
Eigenvalues and Eigenvectors

Recall the eigenvalues of a matrix T are the zeros of the characteristic polynomial:

$$p_T(z) = det(zI - T) = 0.$$

There are exactly n eigenvalues (including multiplicities) for a $n \times n$ matrix T. The set of eigenvalues is calles its *spectrum*.
If λ is an eigenvalue of T, then its associated eigenvector is the non-zero n-vector x such that $Tx = \lambda x$.
Recall: If $T = T^T$ then T is called a *symmetric matrix*. Furthermore:

- Every eigenvalue of T is real.
- There is a set of n eigenvectors $\{e_1, \cdots, e_n\}$ normalized so that the matrix $U = [e_1|\cdots|e_n]$ is orthogonal ($UU^T = U^T U = I_n$) and $T = U \Lambda U^T$, where Λ is the diagonal matrix of eigenvalues.

Spectral Analysis

Eigenvalues and Eigenvectors

Recall the eigenvalues of a matrix T are the zeros of the characteristic polynomial:

$$p_T(z) = det(zI - T) = 0.$$

There are exactly n eigenvalues (including multiplicities) for a $n \times n$ matrix T. The set of eigenvalues is calles its *spectrum*.

If λ is an eigenvalue of T, then its associated eigenvector is the non-zero n-vector x such that $Tx = \lambda x$.

Recall: If $T = T^T$ then T is called a *symmetric matrix*. Furthermore:

- Every eigenvalue of T is real.
- There is a set of n eigenvectors $\{e_1, \cdots, e_n\}$ normalized so that the matrix $U = [e_1|\cdots|e_n]$ is orthogonal ($UU^T = U^TU = I_n$) and $T = U\Lambda U^T$, where Λ is the diagonal matrix of eigenvalues.

Remark. Since $det(A_1A_2) = det(A_1)det(A_2)$ and $L = D^{-1/2}\tilde{\Delta}D^{1/2}$ it follows that $eigs(\tilde{\Delta}) = eigs(L) = eigs(L^T)$.

Spectral Analysis
UCINET IV Database: Bernard & Killworth Office Dataset

For the Bernard & Killworth Office dataset (bkoff.dat) dataset we obtained the following results:
The graph is connected. $rank(\Delta) = rank(\tilde{\Delta}) = rank(L) = 39.$

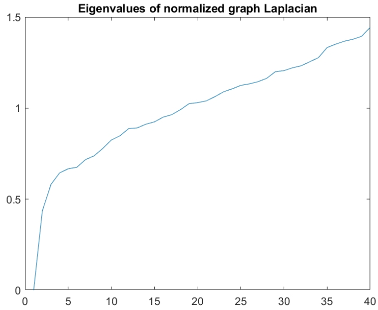


Figure: Adjacency Matrix based Graph Laplacian

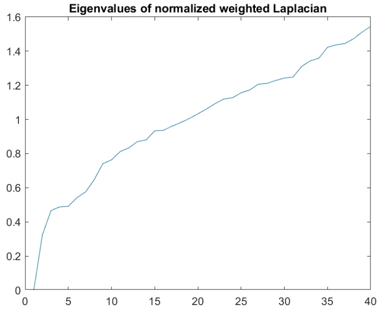


Figure: Weight Matrix based Graph Laplacian

3. Auto-Regressive Processes

Consider a time-series $(x(t))_{t=-\infty}^{\infty}$ where each sample $x(t)$ can be scalar or vector. We say that $(x(t))_t$ is the output of an *Auto-Regressive process of order* p, denoted $AR(p)$, if there are (scalar or matrix) constants $a_1, \cdots, a_p$ so that

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots a_p x(t-p) + \nu(t).$$

Here $(\nu(t))_t$ is a different time-series called the *driving noise*, or the *excitation*.

3. Auto-Regressive Processes

Consider a time-series $(x(t))_{t=-\infty}^{\infty}$ where each sample $x(t)$ can be scalar or vector. We say that $(x(t))_t$ is the output of an *Auto-Regressive process of order p*, denoted $AR(p)$, if there are (scalar or matrix) constants $a_1, \cdots, a_p$ so that

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots a_p x(t-p) + \nu(t).$$

Here $(\nu(t))_t$ is a different time-series called the *driving noise*, or the *excitation*.

Compare the two type of 'noises' we have seen so far:
Measurement Noise: $y_t = Fx_t + r_t$

3. Auto-Regressive Processes

Consider a time-series $(x(t))_{t=-\infty}^{\infty}$ where each sample $x(t)$ can be scalar or vector. We say that $(x(t))_t$ is the output of an *Auto-Regressive process of order p*, denoted $AR(p)$, if there are (scalar or matrix) constants $a_1, \cdots, a_p$ so that

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots a_p x(t-p) + \nu(t).$$

Here $(\nu(t))_t$ is a different time-series called the *driving noise*, or the *excitation*.

Compare the two type of 'noises' we have seen so far:

Measurement Noise: $y_t = Fx_t + r_t$ *Driving Noise*: $x_t = A(x(t-)) + \nu_t$

Scalar AR(p) process

Given a time-series $(x_t)_t$, the least squares estimator of the parameters of an $AR(p)$ process solves the following minimization problem:

$$\min_{a_1,\cdots,a_p} \sum_{t=1}^{T} |x_t - a_1 x(t-1) - \cdots - a_p x(t-p)|^2$$

Scalar AR(p) process

Given a time-series $(x_t)_t$, the least squares estimator of the parameters of an $AR(p)$ process solves the following minimization problem:

$$\min_{a_1,\cdots,a_p} \sum_{t=1}^{T} |x_t - a_1 x(t-1) - \cdots - a_p x(t-p)|^2$$

Expanding the square and rearranging the terms we get
$a^T R a - 2a^T q + \rho(0)$ where

$$R = \begin{bmatrix} \rho(0) & \rho(-1) & \cdots & \rho(p-1) \\ \rho(1) & \rho(0) & \cdots & \rho(p-2) \\ \vdots & & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \cdots & \rho(0) \end{bmatrix}, \quad q = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(p-1) \end{bmatrix}$$

and $\rho(\tau) = \sum_{t=1}^{T} x_t x_{t-\tau}$ is the auto-correlation function.

Scalar AR(p) process

Computing the gradient for the minimization problem

$$\min_{a=[a_1,\cdots,a_p]^T} \quad a^T R a - 2a^T q + \rho(0)$$

produces the closed form solution

$$\hat{a} = R^{-1} q$$

that is, the solution of the linear system $Ra = q$ called the *Yule-Walker system*.
An efficient adaptive (on-line) solver is given by the Levinson-Durbin algorithm.

Multivariate AR(1) Processes

The Multivariate AR(1) process is defined by the linear process:

$$\mathbf{x}(t) = W\mathbf{x}(t-1) + \nu(t)$$

where $\mathbf{x}(t)$ is the n-vector describing the state at time t, and $\nu(t)$ is the driving noise vector at time t. The $n \times n$ matrix W is the unknown matrix of coefficients.

Multivariate AR(1) Processes

The Multivariate AR(1) process is defined by the linear process:

$$\mathbf{x}(t) = W\mathbf{x}(t-1) + \nu(t)$$

where $\mathbf{x}(t)$ is the n-vector describing the state at time t, and $\nu(t)$ is the driving noise vector at time t. The $n \times n$ matrix W is the unknown matrix of coefficients.

In general the matrix W may not have to be symmetric.

However there are cases when we are interested in symmetric $AR(1)$ processes. One such case is furnished by undirected weighted graphs.

Furthermore, the matrix W may have to satisfy additional constraints.

One such constraint is to have zero main diagonal. Alternate case is for W to have constant 1 along the main diagonal.

LSE for Vector AR(1) with zero main diagonal

$$\textit{LS Estimator}: \quad \min_{W \in \mathbb{R}^{n\times n}} \sum_{t=1}^{T} \|\mathbf{x}(t) - W\mathbf{x}(t-1)\|^2$$

$$\text{subject to}: W = W^T$$

$$diag(W) = 0$$

LSE for Vector AR(1) with zero main diagonal

$$LS\ Estimator: \quad \min_{W\in\mathbb{R}^{n\times n}} \sum_{t=1}^{T} \|\mathbf{x}(t) - W\mathbf{x}(t-1)\|^2$$
$$\text{subject to}: W = W^T$$
$$diag(W) = 0$$

How to find W: Rewrite the criterion as a quadratic form in variable $z = vec(W)$, the independent entries in W. If $\mathbf{x}(t) \in \mathbb{R}^n$ is n-dimensional, then z has dimension $m = n(n-1)/2$:

$$z^T = \begin{bmatrix} W_{12} & W_{13} & \cdots & W_{1n} & W_{23} & \cdots & W_{n-1,n} \end{bmatrix}$$

Let $A(t)$ denote the $n \times m$ matrix so that $W\mathbf{x}(t) = A(t)z$. For $n = 3$:

$$A(t) = \begin{bmatrix} \mathbf{x}(t)_2 & \mathbf{x}(t)_3 & 0 \\ \mathbf{x}(t)_1 & 0 & \mathbf{x}(t)_3 \\ 0 & \mathbf{x}(t)_1 & \mathbf{x}(t)_2 \end{bmatrix}$$

LSE for Vector AR(1) with zero main diagonal

Then

$$J(W)=\sum_{t=1}^{T}(\mathbf{x}(t)-A(t)z)^T(\mathbf{x}(t)-A(t)z)=z^TRz-2z^Tq+r_0$$

where

$$R=\sum_{t=1}^{T}A(t)^TA(t)\ \ ,\ \ q=\sum_{t=1}^{T}A(t)^T\mathbf{x}(t)\ \ ,\ \ r_0=\sum_{t=1}^{T}\|\mathbf{x}(t)\|^2.$$

The optimal solution solves the linear system

$$Rz=q\quad\Rightarrow\quad z=R^{-1}q.$$

Then the Least Square estimator W is obtained by reshaping z into a symmetric $n\times n$ matrix of 0 diagonal.

LSE for Vector AR(1) with unit main diagonal

$$LS\ Estimator: \quad \min_{\substack{W \in \mathbb{R}^{n \times n} \\ \text{subject to}: W = W^T \\ diag(W) = ones(n,1)}} \sum_{t=1}^{T} \|\mathbf{x}(t) - W\mathbf{x}(t-1)\|^2$$

LSE for Vector AR(1) with unit main diagonal

$$\textit{LS Estimator}: \quad \min_{W \in \mathbb{R}^{n\times n}} \sum_{t=1}^{T} \|\mathbf{x}(t) - W\mathbf{x}(t-1)\|^2$$
$$\text{subject to}: W = W^T$$
$$diag(W) = ones(n,1)$$

How to find W: Rewrite the criterion as a quadratic form in variable $z = vec(W)$, the independent entries in W. If $\mathbf{x}(t) \in \mathbb{R}^n$ is n-dimensional, then z has dimension $m = n(n-1)/2$:

$$z^T = \left[\begin{array}{ccccccc} W_{12} & W_{13} & \cdots & W_{1n} & W_{23} & \cdots & W_{n-1,n} \end{array} \right]$$

Let $A(t)$ denote the $n \times m$ matrix so that $W\mathbf{x}(t-1) = A(t)z + \mathbf{x}(t-1)$. For $n = 3$:

$$A(t) = \left[\begin{array}{ccc} \mathbf{x}(t-1)_2 & \mathbf{x}(t-1)_3 & 0 \\ \mathbf{x}(t-1)_1 & 0 & \mathbf{x}(t-1)_3 \\ 0 & \mathbf{x}(t-1)_1 & \mathbf{x}(t-1)_2 \end{array} \right]$$

LSE for Vector AR(1) with unit main diagonal

Then

$$J(W)=\sum_{t=1}^{T}(\mathbf{x}(t)-A(t)z-\mathbf{x}(t-1))^T(\mathbf{x}(t)-A(t)z-\mathbf{x}(t-1))=z^TRz-2z^Tq+$$

where

$$R=\sum_{t=1}^{T}A(t)^TA(t),\ q=\sum_{t=1}^{T}A(t)^T(\mathbf{x}(t)-\mathbf{x}(t-1)),\ r_0=\sum_{t=1}^{T}\|\mathbf{x}(t)-\mathbf{x}(t-1)\|^2.$$

The optimal solution solves the linear system

$$Rz=q \quad\Rightarrow\quad z=R^{-1}q.$$

Then the Least Square estimator W is obtained by reshaping z into a symmetric $n\times n$ matrix with 1 on main diagonal.

References

- B. Bollobás, **Graph Theory. An Introductory Course**, Springer-Verlag 1979. **99**(25), 15879–15882 (2002).
- F. Chung, L. Lu, The average distances in random graphs with given expected degrees, Proc. Nat.Acad.Sci.
- R. Diestel, **Graph Theory**, 3rd Edition, Springer-Verlag 2005.
- P. Erdös, A. Rényi, On The Evolution of Random Graphs
- G. Grimmett, **Probability on Graphs. Random Processes on Graphs and Lattices**, Cambridge Press 2010.
- J. Leskovec, J. Kleinberg, C. Faloutsos, Graph Evolution: Densification and Shrinking Diameters, ACM Trans. on Knowl.Disc.Data, **1**(1) 2007.