Data Analysis and Manifold Learning
Lecture 3: Graphs, Graph Matrices, and Graph Embeddings

Radu Horaud
INRIA Grenoble Rhone-Alpes, France
Radu.Horaud@inrialpes.fr
http://perception.inrialpes.fr/
Outline of Lecture 3

- What is spectral graph theory?
- Some graph notation and definitions
- The adjacency matrix
- Laplacian matrices
- Spectral graph embedding
Material for this lecture

- F. R. K. Chung. Spectral Graph Theory. 1997. (Chapter 1)
Spectral graph theory at a glance

- The *spectral graph theory* studies the properties of graphs via the eigenvalues and eigenvectors of their associated graph matrices: the *adjacency matrix*, the *graph Laplacian* and their variants.

- These matrices have been extremely well studied from an algebraic point of view.

- The Laplacian allows a natural link between discrete representations (graphs), and continuous representations, such as metric spaces and manifolds.

- Laplacian embedding consists in representing the vertices of a graph in the space spanned by the smallest eigenvectors of the Laplacian – *A geodesic distance on the graph becomes a spectral distance in the embedded (metric) space.*
Spectral graph theory and manifold learning

- First we construct a graph from $x_1, \ldots, x_n \in \mathbb{R}^D$
- Then we compute the $d$ smallest eigenvalue-eigenvector pairs of the graph Laplacian
- Finally we represent the data in the $\mathbb{R}^d$ space spanned by the corresponding orthonormal eigenvector basis. The choice of the dimension $d$ of the embedded space is not trivial.
- Paradoxically, $d$ may be larger than $D$ in many cases!
Basic graph notations and definitions

We consider *simple graphs* (no multiple edges or loops), $\mathcal{G} = \{V, E\}$:

- $V(\mathcal{G}) = \{v_1, \ldots, v_n\}$ is called the *vertex set* with $n = |V|$;
- $E(\mathcal{G}) = \{e_{ij}\}$ is called the *edge set* with $m = |E|$;
- An edge $e_{ij}$ connects vertices $v_i$ and $v_j$ if they are adjacent or neighbors. One possible notation for adjacency is $v_i \sim v_j$;
- The number of neighbors of a node $v$ is called the *degree* of $v$ and is denoted by $d(v)$, $d(v_i) = \sum_{v_i \sim v_j} e_{ij}$. If all the nodes of a graph have the same degree, the graph is *regular*; The nodes of an *Eulerian* graph have even degree.
- A graph is *complete* if there is an edge between every pair of vertices.
For a graph with $n$ vertices, the entries of the $n \times n$ adjacency matrix are defined by:

$$A := \begin{cases} 
A_{ij} = 1 & \text{if there is an edge } e_{ij} \\
A_{ij} = 0 & \text{if there is no edge} \\
A_{ii} = 0 
\end{cases}$$

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

- A is a real-symmetric matrix: it has \( n \) real eigenvalues and its \( n \) real eigenvectors form an orthonormal basis.
- Let \( \{\lambda_1, \ldots, \lambda_i, \ldots, \lambda_r\} \) be the set of distinct eigenvalues.
- The eigenspace \( S_i \) contains the eigenvectors associated with \( \lambda_i \):
  \[
  S_i = \{x \in \mathbb{R}^n | Ax = \lambda_i x\}
  \]
- For real-symmetric matrices, the algebraic multiplicity is equal to the geometric multiplicity, for all the eigenvalues.
- The dimension of \( S_i \) (geometric multiplicity) is equal to the multiplicity of \( \lambda_i \).
- If \( \lambda_i \neq \lambda_j \) then \( S_i \) and \( S_j \) are mutually orthogonal.
Real-valued functions on graphs

- We consider real-valued functions on the set of the graph’s vertices, $f : V \rightarrow \mathbb{R}$. Such a function assigns a real number to each graph node.
- $f$ is a vector indexed by the graph’s vertices, hence $f \in \mathbb{R}^n$.
- Notation: $f = (f(v_1), \ldots, f(v_n)) = (f_1, \ldots, f_n)$.
- The eigenvectors of the adjacency matrix, $A\mathbf{x} = \lambda \mathbf{x}$, can be viewed as *eigenfunctions*.

![Graph with node values](image)
The adjacency matrix can be viewed as an operator

\[ g = A f; g(i) = \sum_{i \sim j} f(j) \]

It can also be viewed as a quadratic form:

\[ f^\top A f = \sum_{e_{ij}} f(i) f(j) \]
The incidence matrix of a graph

- Let each edge in the graph have an arbitrary but fixed orientation;
- The incidence matrix of a graph is a $|\mathcal{E}| \times |\mathcal{V}|$ ($m \times n$) matrix defined as follows:

$$\nabla := \begin{cases} 
\nabla_{ev} = -1 & \text{if } v \text{ is the initial vertex of edge } e \\
\nabla_{ev} = 1 & \text{if } v \text{ is the terminal vertex of edge } e \\
\nabla_{ev} = 0 & \text{if } v \text{ is not in } e
\end{cases}$$

$$\nabla = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 \\
0 & -1 & 0 & +1 & 0 \\
\end{bmatrix}$$

![Graph with vertices and edges]
The incidence matrix: A discrete differential operator

- The mapping \( f \rightarrow \nabla f \) is known as the **co-boundary mapping** of the graph.

\[
(\nabla f)(e_{ij}) = f(v_j) - f(v_i)
\]

- The matrix representation of the co-boundary mapping is given by:

\[
\begin{pmatrix}
-1 & 1 & 0 & 0 \\
1 & 0 & -1 & 0 \\
0 & -1 & 1 & 0 \\
0 & -1 & 0 & +1
\end{pmatrix}
\begin{pmatrix}
f(1) \\
f(2) \\
f(3) \\
f(4)
\end{pmatrix} =
\begin{pmatrix}
f(2) - f(1) \\
f(1) - f(3) \\
f(3) - f(2) \\
f(4) - f(2)
\end{pmatrix}
\]
The Laplacian matrix of a graph

- \( \mathbf{L} = \nabla^\top \nabla \)
- \( (\mathbf{L}f)(v_i) = \sum_{v_j \sim v_i} (f(v_i) - f(v_j)) \)
- Connection between the Laplacian and the adjacency matrices:
  \[
  \mathbf{L} = \mathbf{D} - \mathbf{A}
  \]
- The degree matrix: \( \mathbf{D} := D_{ii} = d(v_i). \)

\[
\mathbf{L} = \begin{bmatrix}
2 & -1 & -1 & 0 \\
-1 & 3 & -1 & -1 \\
-1 & -1 & 2 & 0 \\
0 & -1 & 0 & 1 \\
\end{bmatrix}
\]
Example: A graph with 10 nodes
The adjacency matrix

\[ A = \begin{bmatrix}
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
\end{bmatrix} \]
The Laplacian matrix

\[ L = \begin{bmatrix}
3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 3 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 4 & -1 & -1 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & -1 & 0 & 4 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 2 & 0 
\end{bmatrix} \]
The Eigenvalues of this Laplacian

\[ \Lambda = \begin{bmatrix}
0.0000 & 0.7006 & 1.1306 & 1.8151 & 2.4011 \\
3.0000 & 3.8327 & 4.1722 & 5.2014 & 5.7462
\end{bmatrix} \]
Matrices of an undirected weighted graph

- We consider *undirected weighted graphs*; Each edge $e_{ij}$ is weighted by $w_{ij} > 0$. We obtain:

  $$\Omega := \begin{cases} 
  \Omega_{ij} = w_{ij} & \text{if there is an edge } e_{ij} \\
  \Omega_{ij} = 0 & \text{if there is no edge} \\
  \Omega_{ii} = 0 
  \end{cases}$$

- The degree matrix: $D = \sum_{i \sim j} w_{ij}$
The Laplacian on an undirected weighted graph

- $L = D - \Omega$
- The Laplacian as an operator:

$$ (Lf)(v_i) = \sum_{v_j \sim v_i} w_{ij} (f(v_i) - f(v_j)) $$

- As a quadratic form:

$$ f^\top Lf = \frac{1}{2} \sum_{e_{ij}} w_{ij} (f(v_i) - f(v_j))^2 $$

- $L$ is symmetric and positive semi-definite $\iff w_{ij} \geq 0$.
- $L$ has $n$ non-negative, real-valued eigenvalues:

$$ 0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n. $$
Other adjacency matrices

- The normalized weighted adjacency matrix
  \[ \Omega_N = D^{-1/2} \Omega D^{-1/2} \]

- The transition matrix of the Markov process associated with the graph:
  \[ \Omega_R = D^{-1} \Omega = D^{-1/2} \Omega_N D^{1/2} \]
Several Laplacian matrices

- The *unnormalized Laplacian* which is also referred to as the *combinatorial Laplacian* $L_C$,
- the *normalized Laplacian* $L_N$, and
- the *random-walk Laplacian* $L_R$ also referred to as the *discrete Laplace operator*.

We have:

\[
\begin{align*}
L_C &= D - \Omega \\
L_N &= D^{-1/2} L_C D^{-1/2} = I - \Omega_N \\
L_R &= D^{-1} L_C = I - \Omega_R
\end{align*}
\]
Relationships between all these matrices

\[ L_C = D^{1/2}L_ND^{1/2} = DL_R \]
\[ L_N = D^{-1/2}L_CD^{-1/2} = D^{1/2}L_RD^{-1/2} \]
\[ L_R = D^{-1/2}L_ND^{1/2} = D^{-1}L_C \]
Some spectral properties of the Laplacians

<table>
<thead>
<tr>
<th>Laplacian</th>
<th>Null space</th>
<th>Eigenvalues</th>
<th>Eigenvectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{L}_C ) = ( \mathbf{U} \Lambda \mathbf{U}^\top )</td>
<td>( \mathbf{u}_1 = \mathbf{1} )</td>
<td>( 0 = \lambda_1 &lt; \lambda_2 \leq \cdots \leq \lambda_n \leq 2 \max_i (d_i) )</td>
<td>( \mathbf{u}_{i&gt;1} \mathbf{1} = 0 ), ( \mathbf{u}_i \mathbf{u}<em>j = \delta</em>{ij} )</td>
</tr>
<tr>
<td>( \mathbf{L}_N ) = ( \mathbf{W} \Gamma \mathbf{W}^\top )</td>
<td>( \mathbf{w}_1 = \mathbf{D}^{1/2} \mathbf{1} )</td>
<td>( 0 = \gamma_1 &lt; \gamma_2 \leq \cdots \leq \gamma_n \leq 2 )</td>
<td>( \mathbf{w}_{i&gt;1} \mathbf{D}^{1/2} \mathbf{1} = 0 ), ( \mathbf{w}_i \mathbf{w}<em>j = \delta</em>{ij} )</td>
</tr>
<tr>
<td>( \mathbf{L}_R ) = ( \mathbf{T} \Gamma \mathbf{T}^{-1} ) ( \mathbf{T} = \mathbf{D}^{-1/2} \mathbf{W} )</td>
<td>( \mathbf{t}_1 = \mathbf{1} )</td>
<td>( 0 = \gamma_1 &lt; \gamma_2 \leq \cdots \leq \gamma_n \leq 2 )</td>
<td>( \mathbf{t}_{i&gt;1} \mathbf{D} \mathbf{1} = 0 ), ( \mathbf{t}_i \mathbf{D} \mathbf{t}<em>j = \delta</em>{ij} )</td>
</tr>
</tbody>
</table>
Spectral properties of adjacency matrices

From the relationship between the normalized Laplacian and adjacency matrix: \( L_N = I - \Omega_N \) one can see that their eigenvalues satisfy \( \gamma = 1 - \delta \).

<table>
<thead>
<tr>
<th>Adjacency matrix</th>
<th>Eigenvalues</th>
<th>Eigenvectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Omega_N = W\Delta W^\top ), ( \Delta = I - \Gamma )</td>
<td>(-1 \leq \delta_n \leq \ldots \leq \delta_2 &lt; \delta_1 = 1)</td>
<td>( w_i^\top w_j = \delta_{ij} )</td>
</tr>
<tr>
<td>( \Omega_R = T\Delta T^{-1} )</td>
<td>(-1 \leq \delta_n \leq \ldots \leq \delta_2 &lt; \delta_1 = 1)</td>
<td>( t_i^\top Dt_j = \delta_{ij} )</td>
</tr>
</tbody>
</table>
The Laplacian of a graph with one connected component

- \( Lu = \lambda u \).
- \( L1 = 0 \), \( \lambda_1 = 0 \) is the smallest eigenvalue.
- The one vector: \( 1 = (1 \ldots 1)^\top \).
- \( 0 = u^\top Lu = \sum_{i,j=1}^n w_{ij} (u(i) - u(j))^2 \).
- If any two vertices are connected by a path, then \( u = (u(1), \ldots, u(n)) \) needs to be constant at all vertices such that the quadratic form vanishes. Therefore, a graph with one connected component has the constant vector \( u_1 = 1 \) as the only eigenvector with eigenvalue 0.
A graph with \( k > 1 \) connected components

- Each connected component has an associated Laplacian. Therefore, we can write matrix \( \mathbf{L} \) as a block diagonal matrix:

\[
\mathbf{L} = \begin{bmatrix}
\mathbf{L}_1 & & \\
& \ddots & \\
& & \mathbf{L}_k
\end{bmatrix}
\]

- The spectrum of \( \mathbf{L} \) is given by the union of the spectra of \( \mathbf{L}_i \).
- Each block corresponds to a connected component, hence each matrix \( \mathbf{L}_i \) has an eigenvalue \( 0 \) with multiplicity 1.
- The spectrum of \( \mathbf{L} \) is given by the union of the spectra of \( \mathbf{L}_i \).
- The eigenvalue \( \lambda_1 = 0 \) has multiplicity \( k \).
The eigenspace of $\lambda_1 = 0$ with multiplicity $k$

- The eigenspace corresponding to $\lambda_1 = \ldots = \lambda_k = 0$ is spanned by the $k$ mutually orthogonal vectors:

  \[
  \mathbf{u}_1 = \mathbf{1}_{L_1} \\
  \ldots \\
  \mathbf{u}_k = \mathbf{1}_{L_k}
  \]

- with $\mathbf{1}_{L_i} = (0000111110000)^\top \in \mathbb{R}^n$

- These vectors are the *indicator vectors* of the graph’s connected components.

- Notice that $\mathbf{1}_{L_1} + \ldots + \mathbf{1}_{L_k} = \mathbf{1}$
The Fiedler vector of the graph Laplacian

- The first non-null eigenvalue $\lambda_{k+1}$ is called the Fiedler value.
- The corresponding eigenvector $u_{k+1}$ is called the Fiedler vector.
- The multiplicity of the Fiedler eigenvalue depends on the graph’s structure and it is difficult to analyse.
- The Fiedler value is the *algebraic connectivity of a graph*, the further from 0, the more connected.
- The Fiedler vector has been extensively used for *spectral bi-partioning*
Eigenvectors of the Laplacian of connected graphs

- \( u_1 = 1, L1 = 0 \).
- \( u_2 \) is the *Fiedler vector* with multiplicity 1.
- The eigenvectors form an orthonormal basis: \( u_i^\top u_j = \delta_{ij} \).
- For any eigenvector \( u_i = (u_i(v_1) \ldots u_i(v_n))^\top \), \( 2 \leq i \leq n \):
  \[
  u_i^\top 1 = 0
  \]
- Hence the components of \( u_i \), \( 2 \leq i \leq n \) satisfy:
  \[
  \sum_{j=1}^{n} u_i(v_j) = 0
  \]
- Each component is bounded by:
  \[-1 < u_i(v_j) < 1\]
Laplacian embedding: Mapping a graph on a line

- Map a weighted graph onto a line such that connected nodes stay as close as possible, i.e., minimize
  \[ \sum_{i,j=1}^{n} w_{ij}(f(v_i) - f(v_j))^2, \]
  or:
  \[ \arg \min_{f} f^\top L f \text{ with: } f^\top f = 1 \text{ and } f^\top 1 = 0 \]

- The solution is the eigenvector associated with the smallest nonzero eigenvalue of the eigenvalue problem: \( L f = \lambda f \), namely the Fiedler vector \( u_2 \).

- Practical computation of the eigenpair \( \lambda_2, u_2 \): the shifted inverse power method (see lecture 2).
Let’s consider the matrix $B = A - \alpha I$ as well as an eigenpair $Au = \lambda u$.

$(\lambda - \alpha, u)$ becomes an eigenpair of $B$, indeed:

$$Bu = (A - \alpha I)u = (\lambda - \alpha)u$$

and hence $B$ is a real symmetric matrix with eigenpairs $(\lambda_1 - \alpha, u_1), \ldots (\lambda_i - \alpha, u_i), \ldots (\lambda_D - \alpha, u_D)$.

If $\alpha > 0$ is chosen such that $|\lambda_j - \alpha| \ll |\lambda_i - \alpha| \forall i \neq j$ then $\lambda_j - \alpha$ becomes the smallest (in magnitude) eigenvalue.

The inverse power method (in conjunction with the $LU$ decomposition of $B$) can be used to estimate the eigenpair $(\lambda_j - \alpha, u_j)$. 

Radu Horaud: Data Analysis and Manifold Learning; Lecture 3
Example of mapping a graph on the Fiedler vector
Laplacian embedding

- Embed the graph in a $k$-dimensional Euclidean space. The embedding is given by the $n \times k$ matrix $F = [f_1 f_2 \ldots f_k]$ where the $i$-th row of this matrix – $f^{(i)}$ – corresponds to the Euclidean coordinates of the $i$-th graph node $v_i$.

- We need to minimize (Belkin & Niyogi ’03):

$$\arg \min_{f_1 \ldots f_k} \sum_{i,j=1}^{n} w_{i,j} \| f^{(i)} - f^{(j)} \|^2 \text{ with: } F^\top F = I.$$

- The solution is provided by the matrix of eigenvectors corresponding to the $k$ lowest nonzero eigenvalues of the eigenvalue problem $Lf = \lambda f$. 
Spectral embedding using the *unnormalized* Laplacian

- Compute the eigendecomposition $L = D - \Omega$.
- Select the $k$ smallest non-null eigenvalues $\lambda_2 \leq \ldots \leq \lambda_{k+1}$
- $\lambda_{k+2} - \lambda_{k+1} = \text{eigengap}$.
- We obtain the $n \times k$ matrix $U = [u_2 \ldots u_{k+1}]$:

\[
U = \begin{bmatrix}
  u_2(v_1) & \ldots & u_{k+1}(v_1) \\
  \vdots & & \vdots \\
  u_2(v_n) & \ldots & u_{k+1}(v_n)
\end{bmatrix}
\]

- $u_i^\top u_j = \delta_{ij}$ (orthonormal vectors), hence $U^\top U = I_k$.
- Column $i$ ($2 \leq i \leq k + 1$) of this matrix is a mapping on the eigenvector $u_i$. 

Radu Horaud
Data Analysis and Manifold Learning; Lecture 3
Examples of one-dimensional mappings

$u_2$

$u_3$

$u_4$

$u_8$
Euclidean L-embedding of the graph’s vertices

- (Euclidean) $L$-embedding of a graph:

$$X = \Lambda_k^{-\frac{1}{2}} U^\top = [x_1 \ldots x_j \ldots x_n]$$

The coordinates of a vertex $v_j$ are:

$$x_j = \begin{pmatrix}
\frac{u_2(v_j)}{\sqrt{\lambda_2}} \\
\vdots \\
\frac{u_{k+1}(v_j)}{\sqrt{\lambda_{k+1}}}
\end{pmatrix}$$

- A formal justification of using this will be provided later.
The Laplacian of a mesh

A mesh may be viewed as a graph: \( n = 10,000 \) vertices, \( m = 35,000 \) edges. ARPACK finds the smallest 100 eigenpairs in 46 seconds.
Example: Shape embedding