The Geometry of Deep Learning. Lecture 5: Spectral Methods in GNN

Rita Fioresi

Fabit

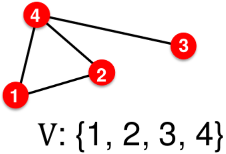
January 23, 2026

Spectral Methods

Main References:

- Stanford course CS224w by Leskovec:
  http://cs224w.stanford.edu/
- Bronstein et al.
  Geometric deep learning: going beyond Euclidean data,
  https://arxiv.org/abs/1611.08097

Adjacency matrix (unweighted graph):



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

Powers of the adjacency matrix:
$(A^n)_{ij}$: number of paths from vertex i to vertex j of length n

$$A^2 = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 3 \end{pmatrix}$$

$D = A^2$ degree matrix, $d_{ii} = (A^2)_{ii}$ number of edges in v_i.

Trainable weight matrices (i.e., what we learn)

$$h_v^{(0)} = x_v$$

$$h_v^{(k+1)} = \sigma\left(W_k \sum_{u \in N(v)} \frac{h_u^{(k)}}{|N(v)|} + B_k h_v^{(k)}\right), \forall k \in \{0..K-1$$

$$z_v = h_v^{(K)}$$

Final node embedding

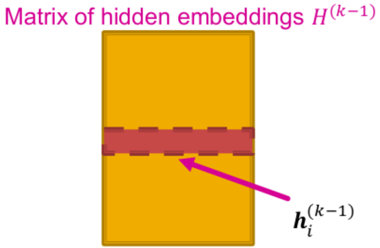
We can feed these **embeddings into any loss function** and run SGD to **train the weight parameters**

h_v^k: the hidden representation of node v at layer k

- W_k: weight matrix for neighborhood aggregation
- B_k: weight matrix for transforming hidden vector of self

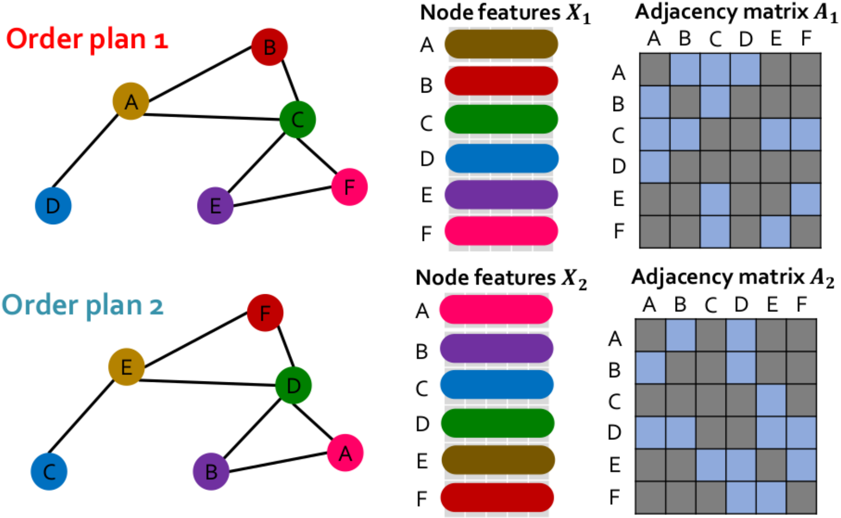
Message Passing: Matrix formulation

- Let $H^{(k)} = [h_1^{(k)} \ldots h_{|V|}^{(k)}]^{\mathrm{T}}$
- Then: $\sum_{u \in N_v} h_u^{(k)} = \mathrm{A}_{v,:}\mathrm{H}^{(k)}$
- Let D be diagonal matrix where
  $D_{v,v} = \mathrm{Deg}(v) = |N(v)|$
  - The inverse of D: D^{-1} is also diagonal:
    $D_{v,v}^{-1} = 1/|N(v)|$
- **Therefore,**



$$\sum_{u \in N(v)} \frac{h_u^{(k-1)}}{|N(v)|} \Longrightarrow H^{(k+1)} = D^{-1} A H^{(k)}$$

- **Graph does not have a canonical order of the nodes!**

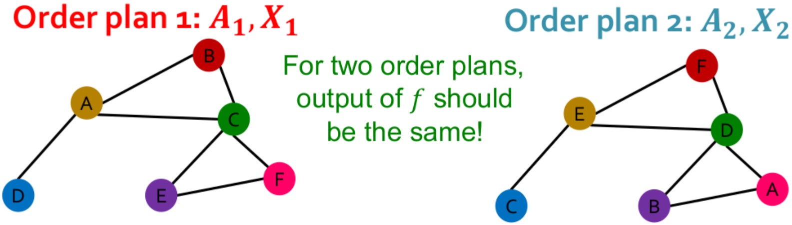


Group equivariance: Encoder function

- The encoder function $V \longrightarrow \mathbb{R}^d$ must be invariant by relabelling of vertices operation.
- But, it must also take into account the adjacency matrix!

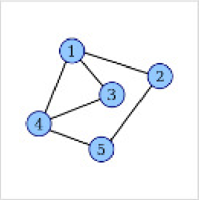
$$f(\boldsymbol{A}_1, \boldsymbol{X}_1) = f(\boldsymbol{A}_2, \boldsymbol{X}_2)$$

$\boldsymbol{A}$ is the adjacency matrix
$\boldsymbol{X}$ is the node feature matrix

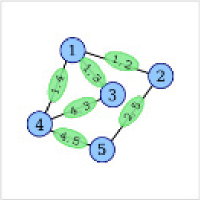


Given a graph G, the line graph $L(G)$:

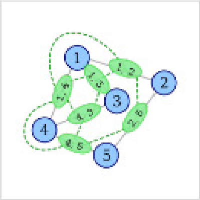
- each vertex of $L(G)$ represents an edge of G
- two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint.



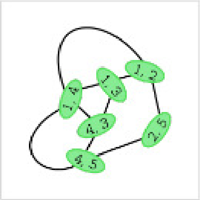
Graph G



Vertices in L(G) constructed from edges in G

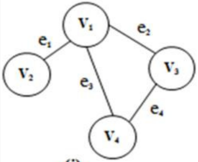


Added edges in L(G)



The line graph L(G)

Incidence matrix $X(G)$ **(or** $X(G)^t$ **for a graph** G:

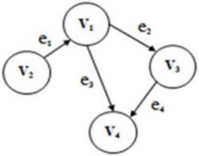


(i)

	e_1	e_2	e_3	e_4
V_1	1	1	1	0
V_2	1	0	0	0
V_3	0	1	0	1
V_4	0	0	1	1

(ii)

Incidence matrix for a directed graph:



(i)

	e_1	e_2	e_3	e_4
V_1	-1	1	1	0
V_2	1	0	0	0
V_3	0	-1	0	1
V_4	0	0	-1	-1

(ii)

Theorem. $A(L(G)) = X(G)^t X(G) - 2I$.

Functions on graphs

Functions on vertices and functions on edges:

$$F(V) = \{f : V \longrightarrow \mathbb{R}\} \quad \text{analogy with } C^\infty(M)$$

$$F(E) = \{f : E \longrightarrow \mathbb{R}\} \quad \text{analogy with } \Omega^1(M)$$

$$f : V \longrightarrow \mathbb{R} \Longleftrightarrow h \in \mathbb{R}^{|V|}$$

$$\text{differential operator on } F(V) \Longleftrightarrow |V| \times |V| \text{ matrix}$$

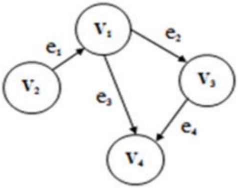
$$\text{exterior derivative } d : C^\infty(M) \longrightarrow \Omega^1(M) \Longleftrightarrow |E| \times |V| \text{ matrix}$$

Incidence matrix as gradient

$G = (V, E)$ directed graph, $f : V \longrightarrow \mathbb{R}$ function on vertices.
Discrete gradient of f:

$$\nabla f : E \longrightarrow \mathbb{R}, \qquad f(i,j) = f(j) - f(i) \quad \text{coboundary operator!}$$

Proposition. The operator ∇ (or ∇^t) is the incidence matrix (obvious).



(i)

	e_1	e_2	e_3	e_4
V_1	-1	1	1	0
V_2	1	0	0	0
V_3	0	-1	0	1
V_4	0	0	-1	-1

(ii)

Note. d is the discrete version of exterior derivative:
$C^\infty(V) = \{$ functions on $V\} \longrightarrow \Omega^1(V) = \{$ differential forms on $V\}$

For ordinary geometry the Laplacian is the operator:

$$\Delta = \partial_1^2 + \cdots + \partial_n^2 = (\partial_1 + \cdots + \partial_n) \cdot (\partial_1 + \cdots + \partial_n) = \nabla^t \cdot \nabla$$

Definition. Let $G(V, E)$ be an directed graph:
$L = D - A$ is the Laplacian.
D degree matrix
A adjacency matrix

Proposition. $L = \nabla^t \cdot \nabla$ (obvious)
∇: discrete gradient.

Example.

$$\nabla = \begin{pmatrix} -1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad \nabla^t \nabla = \begin{pmatrix} -1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

- $\det(L) = 0$, hence one eigenvalue is zero.
  If ∇ is rectangular: obvious. Otherwise: $\mathrm{rk}\nabla = n - 1$ (Godsil, Royce).
- $L = \nabla^t \cdot \nabla$ is symmetric: by Spectral Theorem it is diagonalizable.
- All eigenvalues of L are positive or zero:

$$0 \leq \lambda_0 < \lambda_1 < \dots$$

Sketch. For any matrix $(T^t \cdot T)_{ii} = \mathrm{row}_i \cdot \mathrm{row}_i$

$$\text{For } T = P^t \nabla P, \quad P^t \nabla^t P \cdot P^t \nabla P = \begin{pmatrix} \lambda_0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & \lambda_n \end{pmatrix}$$

Bronstein defines:

$$L_W = M^{-1}(D - W)$$

,
W replaces A: $w_{ij} = 0$ if (i, j) not an edge.
Case 1: $M = I$, $L = D - W$, for $W = A$ we recover previous Laplacian.
Case 2: $M = D^{-1}$, $L_D = I - D^{-1}W$, for $W = A$ is the diffusion/walk matrix.

Proposition. Let $h : V \longrightarrow \mathbb{R}$ (vector of features, here 1 feature only). Then

$$(D^{-1}A\,h)(v) = \sum_{u,(u,v)\,\text{edge}} \frac{h(u)}{d(u)}$$

$d(u)$ degree of u = number of edges in u.
Proof. Obvious (just write it!)

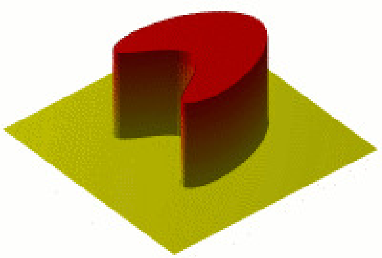
Key Point

$$\text{Feature function } h : V \longrightarrow \mathbb{R} \iff h \in \mathbb{R}^{|V|}$$

Laplacian is $|V| \times |V|$ matrix and acts on features!

Heat Equation

$$(\partial_t h)(v) = -(\Delta h)(v)$$



Important. Look at case 2 and the heat equation with $\Delta = L_D = I - D^{-1}A$:

$$(\partial_t h)(v) = -(\Delta h)(v) = -h(v) + \sum_{u,(u,v)\,\text{edge}} \frac{h(u)}{d(u)}$$

In its time discrete version:

$$h_{t+1}(v) = h_t(v) - h_t(v) + \sum_{u,(u,v)\,\text{edge}} \frac{h_t(u)}{d(u)} = \sum_{u,(u,v)\,\text{edge}} \frac{h_t(u)}{d(u)}$$

Hence:

Message passing = heat equation for diffusion Laplacian

Dirichlet energy

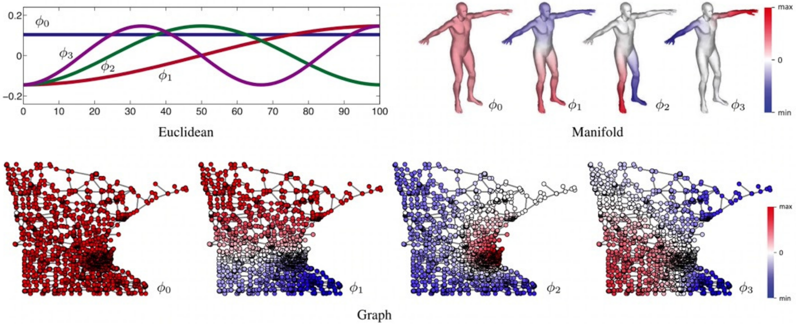
Analogy with the ordinary setting.
Facts:

- The Dirichlet energy measures the smoothness of a function:

$$D(f) = \int \nabla(f) \cdot \nabla(f) dx = -\int f \Delta(f)$$

  (f compact support).
- The Laplacian eigenbasis is a set of orthogonal minimizers of the Dirichlet energy.
- The Laplacian eigenbasis is optimal for representing smooth signals (the features!) in the Fourier expansion.

The eigenfunctions of the laplacian form the smoothest-possible basis function over a specific graph (they minimize the Dirichlet energy).



On Grids. Circulant matrices: convolution operators commuting with Laplacian. Hence: we can use the Laplacian eigenbasis (Spectral domain).

On Graphs. Laplacian eigenvectors: use this basis to expand features $h : V \longrightarrow \mathbb{R}$ or $h \in \mathbb{R}^{|V|}$.

convolution on graphs $\Longleftrightarrow$ laplacian operator

j^{th} neighborhood of a vertex reached by j^{th} power of laplacian

filters $\Longleftrightarrow$ polynomials in the laplacian operator