Spectral Graph Neural Networks

BII Study Group on Graph Neural Networks

Vipul Singhal

January 19, 2023

Genome Institute of Singapore

Prelims

Preliminaries I: Graphs and adjacency

A graph is an ordered pair $G = (V, E)$ of sets, where V is the *vertex* (or node) set, and E is the *edge set*, with

$$E \subset \{\{x, y\} \mid x, y \in V, x \neq y\}. \tag{1}$$

A graph with vertex set $\{x_1, x_2, \ldots, x_n\}$ has an edge between vertices x_i and x_j if $\{x_i, x_j\} \in E$, and these vertices are then called *adjacent*, or *neighbours*.

The *degree* $d(v)$ of a vertex v is the number of vertices that are adjacent to it.

Preliminaries II: Adjacency and Laplacian matrices

Let the vertex set be $V = \{1, 2, \ldots, n\}$. The *adjacency matrix* A_G of the graph G is defined by entries

$$a_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The Laplacian matrix L_G of the graph G is defined by the entries

$$l_{ij} = \begin{cases} -1 & \text{if } \{i, j\} \in E \\ d(i) & \text{if i = j, and} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Recall from previous lectures that $L_G = D_G - A_G$, where the degree matrix D_G, is diagonal, with (i, i)-th entry $d(i)$.

Another definition of L_G

We can decompose L_G edge-wise, resulting in an expansion that illuminates the link between the graph structure and the Laplacian's spectral properties. First, we need some more definitions.

Definition
Suppose $G = (V, E)$ is a graph with $V = \{1, 2, \ldots, n\}$. For an edge $\{u, v\} \in E$, we define an $n \times n$ matrix $L_{G_{\{u,v\}}}$ by

$$L_{G_{\{u,v\}}}(i,j) = \begin{cases} 1 & \text{if } i = j \text{ and } i \in \{u, v\}, \\ -1 & \text{if } i = u \text{ and } j = v, \text{ or vice versa,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Lemma
For a graph $G = (V, E)$, we have the edge-wise decomposition $L_G = \sum_{\{u,v\} \in E} L_{G_{\{u,v\}}}$.

Some useful facts

Lemma
The eigenvalues of a real symmetric matrix, such as L_G, are real.

Proof.
We prove this for the self-adjoint complex case ($H^* = H \in \mathbb{C}^{n\times n}$), and note that the real symmetric case ($H^T = H \in \mathbb{R}^{n\times n}$) is a special case of this.

Let $H \in \mathbb{C}^{n\times n}$ be an arbitrary self-adjoint matrix.

Let x be a non-zero eigenvector with associated eigenvalue $\lambda \in \mathbb{C}$.
We have $x^*Hx = \lambda x^*x$ and $x^*Hx = x^*H^*x = (Hx)^*x = (\lambda x)^*x = \bar{\lambda}x^*x$.
Thus, $\lambda = \bar{\lambda}$, which implies $\lambda \in \mathbb{R}$. □

L_G is positive semidefinite

Lemma
For a signal $x \in \mathbb{R}^n$ defined on the vertices of G, we have the quadratic form $x^T L_{G_{\{u,v\}}} x = (x_u - x_v)^2$.

Lemma
L_G is positive semidefinite, i.e., $x^T L_G x \geq 0,\ \forall x \in \mathbb{R}^n$, and its eigenvalues are non-negative

Proof.
Postive semidefiniteness is immediate from the lemma above and the edge-wise decomposition of L_G.

Non-negativity of eigenvalues is is shown as follows. For an eigen-pair $(\lambda, x \neq 0)$, we have

$$\begin{aligned} 0 &\leq x^T L_G x \\ &= x^T(\lambda x) \\ &= \lambda(x^T x), \end{aligned}$$

which implies $\lambda \geq 0$, noting that $x^T x > 0$. □

Connectedness

Before we can link the spectral properties of L_G with the structure of G, we need a couple more definitions.

Definition (Path)
A *path* is a non-empty graph $P = (V, E)$ of the form (defined without loss of generality)

$$V = \{x_0, x_1, \ldots, x_n\},$$
$$E = \{\{x0, x1\}, \{x1, x2\}, \ldots, \{x_{n-1}, x_n\}\},$$

where all the vertices x_i are distinct.

Definition (Connectedness)
A non-empty graph G is called *connected* if any two of its vertices are contained in a path in G.

A link between the spectrum of L_G and the structure of G

Theorem
For any graph G, $\lambda_1 = 0$ for L_G. If $G = (V, E)$ is a connected graph, where $V = \{1, 2, \ldots, n\}$, then $\lambda_2 > 0$

A link between the spectrum of L_G and the structure of G

Theorem
For any graph G, $\lambda_1 = 0$ for L_G. If $G = (V, E)$ is a connected graph, where $V = \{1, 2, \ldots, n\}$, then $\lambda_2 > 0$

Proof.
Consider a signal $x = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$. Then, the i-th entry of $m = L_G x$ is $m_i = \sum_{k=1}^{n} l_{ik}$. From the first definition of L_G, we know that the rows sum to 0, i.e., $m_i = 0$ for all i, and so $L_G x = 0x = 0$, and 0 is an eigenvalue of L_G.

A link between the spectrum of L_G and the structure of G

Theorem
For any graph G, $\lambda_1 = 0$ for L_G. If $G = (V, E)$ is a connected graph, where $V = \{1, 2, \ldots, n\}$, then $\lambda_2 > 0$

Proof.
Consider a signal $x = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$. Then, the i-th entry of $m = L_G x$ is $m_i = \sum_{k=1}^{n} l_{ik}$. From the first definition of L_G, we know that the rows sum to 0, i.e., $m_i = 0$ for all i, and so $L_G x = 0x = 0$, and 0 is an eigenvalue of L_G.

Let $z \neq 0$ be an eigenvector associated with the 0 eigenvalue, i.e., $L_G z = 0$. Then, $z^T L_G z = \sum_{\{u,v\} \in E} (z_u - z_v)^2 = 0$. This implies that for all $\{u, v\} \in E$, $z_u = z_v$. Since G is connected, this means that $z_i = z_j$ for all $i, j \in V$. That is, $z = \alpha(1, 1, \ldots, 1)$ for some $\alpha \in \mathbb{R}$. Thus, The eigenspace of λ is Span$((1, 1, \ldots, 1))$, and the (geometric) multiplicity of $\lambda = 0$ is 1. This implies $\lambda \neq 0$, and thus $\lambda > 0$. $\square$

Another link between the spectrum and structure of G

Definition (Connected Component)
A connected component of a graph $G = (V, E)$ is a subgraph $G'(V', E')$, with $V' \subset V$, and $E' = \{\{x, y\} \in E \mid x, y \in V'\}$, in which any two vertices $i, j \in V'$ are connected while for any $i \in V'$ and $k \in V\, V'$, i, k are not connected.

Corollary
Let $G = (V, E)$ be a graph. Then, the (geometric) multiplicity of 0 as an eigenvalue of L_G equals the number of connected components of G.

We will leave the proof of this corollary as a homework exercise.

Hint: Use the same argument as in the proof of the theorem above, and letting $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_k = (V_k, E_k)$ be the connected components of G, consider the k signal vectors $\{w_1, w_2, \ldots, w_k\}$ defined by

$$(w_i)_j = \begin{cases} 1 & \text{if } j \in V_i \\ 0 & \text{otherwise} \end{cases}$$

Next session...

- Graph Fourier Transform
- Spectral Graph Neural Networks

Graph Neural Networks

The graph neural networks, high level

We now describe the spectral graph neural network model. The encoder uses both the features matrix and the graph structure to compute embeddings,

$$Z = \mathrm{ENC}(X, W; \Theta^E)$$

This is usually implemented recursively till convergence

$$Z^{t+1} = \mathrm{ENC}(X, W, Z^t; \Theta^E),$$

where the parameters Θ^E are reused at every iteration. Upon convergence ($t = T$), we can generate output graph or node labels,

$$\hat{y}^S = \mathrm{DEC}(X, Z^T; \Theta^S), \tag{5}$$

where Θ^S are the (semi-)supervised loss parameters. The computation of Z^T and $\hat{y}^S$ is repeated in each iteration of the training process (which can be performed via backprop, for example).

The graph convolution framework (GCF) components

Before we can describe the spectral graph neural network, we need to describe some of its components.

- **Node embeddings:** $H^l \in \mathbb{R}^{n \times d_l}$, where d_l are the number of features or "filters" in the l-th layer.
- K **Patch functions:** $\{f_k(W, H^l)\}_{k=1}^K$, which are $n \times n$ matrices defining which nodes interact at each step of the convolution.

The graph convolution framework components (GCF) II

- **Convolution filters' weights:** For patch k and layer l, we can define filter weights $\Theta_k^l \in \mathbb{R}^{d_l \times d_{l_1}}$. For each patch-layer pair,
  - Θ_k^l: each column is a filter, and there is a stack of d_{l+1} filters.
  - d_l and d_{l+1} is similar to the number of channels in layers l and $l+1$ in CNNs.
  - $m_k^{l+1} = f_k(W, H^l)H^l\Theta_k^l$
  - Consider the j-th column of $H^l\Theta_k^l$: each element is the inner product of that node's features with the j-th filter's weights, and you do this for all nodes (and for all d_{l+1} filters) .
  - The i-th row of f_k: This tells you how to sum together the inner products computed by the previous step to get the "convolved" values of the d_{l+1} features for the i-th node (for the k-th patch).

The graph convolution framework components (GCF) III

- **Merging functions:** $H^{l+1} = h(m_1^{l+1}, \ldots, m_K^{l+1})$. These can be a an averaging, concatenation, more complex MLP, followed by a nonlinearity.
- After L convolution layers, we have the nodes' embeddings $Z = H^L$.

Graph convolution via the Fourier transform

We can generalize the Fourier transform defined on Euclidean domains (e.g., grids) to graph structured domains. Let $x \in \mathbb{R}^n$ be a graph signal, and Θ be a filter (linear operator) defined on Euclidean domains, such that Θx is their convolution. We want to find an analogue of this, $x * \theta$, for general graphs.

$$\mathcal{F}(\Theta x) = \mathcal{F}(x) \odot \mathcal{F}(\theta) \qquad \text{LHS not defined on graphs}$$

$$x * \theta = \mathcal{F}^{-1}(\mathcal{F}(x) \odot \mathcal{F}(\theta)) \qquad \text{RHS used to define LHS on graphs}$$

Here I think Θ is the operator that performs convolution on grids, so you can write its operation in matrix notation, Θx (i.e., as a linear operator acting on a signal).

But more generally you want to write convolution using the explicit operator $*$. Then, θ is the symbol used to express the filter (I think in analogy to impulse response in LTI systems), and this is what is convolved with the signal x.

Graph convolution via the Fourier Transform

Let $\tilde{L} = I - D^{1/2}WD^{1/2}$ be the normalized Laplacian of a graph. It is real, symmetric positive semidefinite matrix, and admits an orthonormal eigendecomposition, $\tilde{L} = U\Lambda U^T$. For $x \in \mathbb{R}^n$, the discrete graph Fourier transform and its inverse are defined as

$$\mathcal{F}(x) = \hat{x} = U^T x \quad \text{and} \quad \mathcal{F}^{-1}(\hat{x}) = U\hat{x}$$

Thus,

$$\begin{aligned} x * \theta =& U(U^T x \odot U^T \theta) \\ =& U\text{diag}(U^T\theta)U^T x \end{aligned}$$

Spectral Convolution Neural Networks (SCNNs)

We now consider a special case of the GCF in light of the above definition of the graph Fourier transform. SCNNs learn convolution filters that act on the basis provided by the eigenvectors of the graph Laplacian. Each eigenvector gives a patch function, and we can pick a subset K of the "most informative" eigenvectors as our patch functions.

For the j-th feature on the $l+1$-th layer, we have

$$H^{l+1}_{:,j} = \sigma\left(\sum_{i=1}^{d_l} U_K F^l_{i,j} U_K^T H^l_{:,i}\right) \qquad 1 \leq j \leq d_{l+1} \quad \text{and} \quad i\ eqi \leq d_l,$$

where $F^l_{i,j}$ are $K \times K$ trainable diagonal matrices containing filter weights in the spectral domain, and U_K is a matrix with K of the eigenvectors of $\tilde{L}$, and σ is a non-linearity (applied componentwise).

Compare this to $U\text{diag}(U^T\theta)U^T x = \mathcal{F}^{-1}(\mathcal{F}(x) \odot \mathcal{F}(\theta))$

Spectral Convolution Neural Networks in GCF form

The equation in the previous slide can be reformulated into the GCF form, $H^{l+1} = \sigma\left(\sum_{k=1}^{K} u_k u_k^T H^l \Theta_k^l\right)$. To do this, first note that

$$
\begin{aligned}
H_{:,j}^{l+1} &= \sigma\left(\sum_{i=1}^{d_l} U_K F_{i,j}^l U_K^T H_{:,i}^l\right) \\
&= \sigma(\sum_{k=1}^{K}\sum_{i=1}^{d_l} F_{i,j,k}^l H_{:,i}^l),
\end{aligned}
$$

where u_k are the top K eigenvectors, and $F_{i,j,k}^l$ in the k-th diagonal element of $F_{i,j}^l$. This follows from the standard outer product expansion, $U\text{diag}(\delta)U^T = \sum_k \delta_k u_k u_k^T$, where δ is a vector and u_k is the k-th column of the arbitrary matrix U. We can then just collect vectors into matrices.

Some remarks

There are $\mathcal{O}(d_l d_{l+1} K)$ terms in the learnable parameter set F. This can be computationally demanding, and so Bruna et al. and others have described some smoothness constraints on the parameters to reduce the number of degrees of freedom (via splines). You can read more about this in the paper.