Capstone A: Graph Theory, Networks & Machine Learning

Table of Contents

Graphs and Matrices: The Language of Networks

Linear algebra provides a powerful language to describe and analyze graphs. A graph `G = (V, E)` with `n` vertices can be represented by several key matrices.

Adjacency Matrix (`A`)

- **Definition**: An `n x n` matrix where `A_ij = 1` if there is an edge between vertex `i` and vertex `j`, and `0` otherwise. For weighted graphs, `A_ij` can be the edge weight.
- **Properties**:
  - For an undirected graph, `A` is symmetric.
  - The `(i, j)`-th entry of `A^k` gives the number of walks of length `k` from vertex `i` to vertex `j`.

Degree Matrix (`D`)

- **Definition**: A diagonal matrix where `D_ii` is the degree of vertex `i` (the number of edges connected to it).

Incidence Matrix (`B`)

- **Definition**: An `n x m` matrix (where `n` is #vertices, `m` is #edges) that describes the relationship between vertices and edges. `B_ie = 1` if vertex `i` is an endpoint of edge `e`, and `0` otherwise.

Eigenvalues in Graphs: Spectral Graph Theory

Spectral graph theory is the study of a graph's properties through the eigenvalues and eigenvectors of its associated matrices (primarily the Adjacency and Laplacian matrices).

- **Eigenvalues of the Adjacency Matrix**: Reveal information about the graph's structure. For example, the largest eigenvalue (the *spectral radius*) is related to the graph's density and the growth rate of walks.

- **Eigenvectors**: Can be seen as functions or signals on the vertices of the graph. The eigenvector corresponding to the largest eigenvalue is often used to find the most "important" or central nodes in a network (this is the core idea behind PageRank).

The Laplacian Matrix: The Heart of Graph Analysis

The **Graph Laplacian** is a central tool in modern graph analysis, especially in machine learning.

- **Definition**: `L = D - A` (Degree Matrix - Adjacency Matrix).
- **Key Properties**:
    1. **Positive Semidefinite**: `L` is always positive semidefinite (`x^T L x >= 0`), meaning all its eigenvalues are non-negative ($\lambda_i \geq 0$).
    2. **Smallest Eigenvalue**: The smallest eigenvalue is always `0` ($\lambda_1 = 0$), and its corresponding eigenvector is the constant vector `(1, 1, ..., 1)`.
    3. **Number of Connected Components**: The multiplicity of the `0` eigenvalue equals the number of connected components in the graph. If the graph is fully connected, `$\lambda$_2 > 0`.
    4. **Fiedler Value**: The second-smallest eigenvalue, `$\lambda$_2`, is called the **Fiedler value** or **algebraic connectivity**. It measures how well-connected the graph is. A larger `$\lambda$_2` means the graph is harder to cut into two pieces.
    5. **Fiedler Vector**: The eigenvector corresponding to `$\lambda$_2` can be used to partition the graph into two clusters (this is the basis of **spectral clustering**).

Applications: PageRank, Spectral Clustering, Quantum Walks

PageRank

- **Goal**: Rank the importance of web pages.
- **Linear Algebra Connection**: PageRank finds the stationary distribution of a random walk on the web graph. This is equivalent to finding the eigenvector corresponding to the largest eigenvalue (`$\lambda$=1`) of a modified adjacency matrix (the Google matrix). The entries of this eigenvector give the PageRank scores.

Spectral Clustering

- **Goal**: Partition the nodes of a graph into clusters.
- **Linear Algebra Connection**: It uses the eigenvectors of the Graph Laplacian. Typically, the Fiedler vector (the eigenvector of `$\lambda$_2`) is used. Nodes are clustered based on the sign (positive or negative) of their corresponding entry in the Fiedler vector. This method can uncover cluster structures that simpler methods like k-means might miss.

Quantum Walks

- **Goal**: A quantum mechanical analogue of a classical random walk on a graph.
- **Linear Algebra Connection**: The state of the walk is a vector in a Hilbert space. The evolution of the walk is governed by a unitary operator `U` that depends on the graph's structure (often related to `A` or `L`). Quantum walks can be used to design faster search algorithms and are a fundamental concept in quantum computing.

Bridge to Advanced Topics

- **Manifold Learning**: In machine learning, if you have a high-dimensional dataset, you can construct a graph where nearby points are connected. The Laplacian of this graph can then be used to find a low-dimensional embedding of the data that preserves local structure (Laplacian Eigenmaps).
- **Graph Neural Networks (GNNs)**: The Laplacian matrix is fundamental to how GNNs propagate information between nodes. The core operation in many GNNs is a form of message passing that can be interpreted as a filtering operation on the graph's spectrum.

Practice Problems

1. For a simple graph of a triangle (3 vertices, 3 edges), write down the Adjacency matrix `A`, the Degree matrix `D`, and the Laplacian matrix `L`.
2. Calculate the eigenvalues of the Adjacency matrix for the triangle graph. What do they tell you?
3. Show that for any vector `x`, `x^T L x = sum_{i,j where i~j} (x_i - x_j)^2`, where `i~j` means there is an edge between `i` and `j`. What does this tell you about why `L` is positive semidefinite?
4. If a graph has two separate, disconnected components, why must its Laplacian have at least two eigenvalues equal to 0?
5. **Conceptual**: How could you use the Fiedler vector of a social network graph to find two distinct communities?