CMSC 420: Laplacian Matrices, Graph Clustering, Spanning Trees

Justin Wyss-Gallifent

May 6, 2025

1 Introduction

The goal of this section is to discuss the Laplacian matrix of a graph as a data structure and how it can be used to answer two problems:

- If a graph can be divided into two subgraphs each of which is a strongly-connected cluster and such that the two clusters are weakly connected with one another, how can we partition them algorithmically?
- How many spanning trees does a graph have?

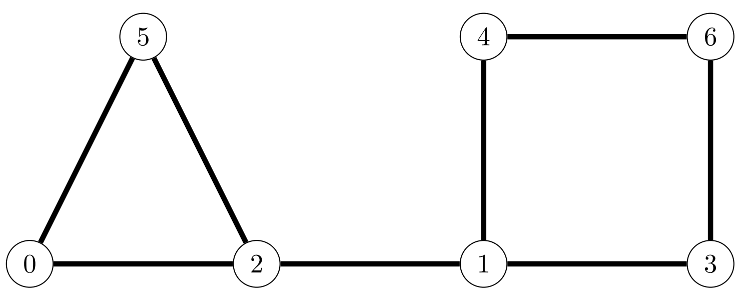
2 Math Part 1

2.1 The Laplacian Matrix

Adjacency matrices are commonly used to store simple graphs. For a graph with n vertices we construct an $n \times n$ matrix A where:

$$A[i,j] = \begin{cases} 1 & \text{if vertices } i \text{ and } j \text{ are connected by an edge.} \\ 0 & \text{if vertices } i \text{ and } j \text{ are not connected by an edge.} \end{cases}$$

Example 2.1. This graph:



Has the adjacency matrix:

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

Note 2.1.1. Observe that we are 0-indexing the rows and columns here, as is typical in CS.

There are two other matrices that arise frequently.

Definition 2.1.1. The degree matrix D has $A[i,i]$ equal to the degree of the vertex i and 0 elsewhere.

Definition 2.1.2. The Laplacian matrix is defined by $L = D - A$.

Example 2.2. Thus for the graph given in the introduction we have:

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$L = D - A = \begin{bmatrix} 2 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 2 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2 & 0 & -1 \\ -1 & 0 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 2 \end{bmatrix}$$

Observe that the Laplacian matrix is sufficient enough to describe the graph completely. The adjacency matrix is, too, but the Laplacian matrix turns out to be more useful.

2.2 The Determinant of a Matrix

Given an $n \times n$ matrix the single most important value associated to this matrix is its determinant.

There are many ways to define the determinant but we'll do it recursively in a very computational way.

Definition 2.2.1. For a 1×1 matrix A (containing a single value) we define the *determinant* $\det(A)$ to be equal to that value.

Example 2.3. If we have $A = [3]$ then $\det(A) = 3$.

Definition 2.2.2. For an $n \times n$ matrix with $n > 1$ we recursively define $\det(A)$ as follows:

If the top row of A consists of the entries a_{11}, a_{12}, ... , a_{1n} then we define:

$$\det(A) = +a_{11}\det(A_{11}) - a_{12}\det(A_{12}) + a_{13}\det(A_{14}) - ...$$

Here the matrix A_{ij} is the *matrix minor* which is the matrix A with row i and column j removed.

Example 2.4. We have:

$$\det\left(\begin{bmatrix}3 & 5\\ 2 & 7\end{bmatrix}\right) = +3\det([7]) - 5\det([2]) = 3(7) - 5(2) = 21 - 10 = 11$$

Observe that in general:

$$\det\left(\begin{bmatrix}a & b\\ c & d\end{bmatrix}\right) = ad - bc$$

Example 2.5. We have:

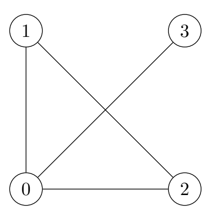
$$\begin{aligned}\det\left(\begin{bmatrix}3 & 5 & 0\\ 2 & 7 & 1\\ 0 & 4 & 6\end{bmatrix}\right) &= +3\det\left(\begin{bmatrix}7 & 1\\ 4 & 6\end{bmatrix}\right) - 5\det\left(\begin{bmatrix}2 & 1\\ 0 & 6\end{bmatrix}\right) + 0\det\left(\begin{bmatrix}2 & 7\\ 0 & 4\end{bmatrix}\right)\\ &= 3(38) - 5(12) + 0(8)\\ &= 54\end{aligned}$$

3 Kirchoff's Theorem for Spanning Trees

We have the following theorem for counting the number of spanning trees for a graph:
Theorem 3.0.1. Suppose a graph has Laplacian matrix L. Then the number of spanning trees for the graph equals $|\det(L_{ij})|$ for any i, j.

Proof. Omit for now. $\mathcal{QED}$

Example 3.1. Consider the graph:



This graph has Laplacian:

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

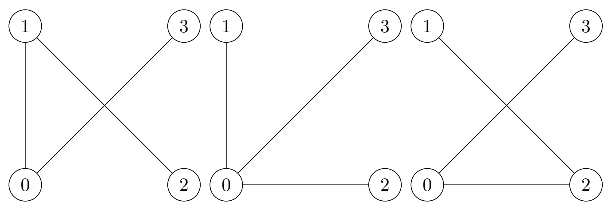
Observe that, for example:

$$|\det(L_{11})| = ... = |3| = 3$$

And just to try a different minor:

$$|\det(L_{12})| = ... = |-3| = 3$$

This tells us that there are three spanning trees. And in fact here they are:



4 Math Part 2

4.1 Matrix and Vector Multiplication

Given an $n \times n$ matrix A and a vector with n entries (an $n \times 1$ matrix) $\bar{v}$ we can perform the multiplication $A\bar{v}$ to form a new vector with n entries.

We do this by multiplying each of the i^{th} entries in $\vec{v}$ by the i^{th} column in A and adding.

Example 4.1. For example:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 1 \\ 5 & 7 & 6 \end{bmatrix} \begin{bmatrix} 10 \\ 15 \\ 30 \end{bmatrix} = 10 \begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix} + 15 \begin{bmatrix} 2 \\ 4 \\ 7 \end{bmatrix} + 30 \begin{bmatrix} 3 \\ 1 \\ 6 \end{bmatrix} = \begin{bmatrix} 130 \\ 90 \\ 335 \end{bmatrix}$$

4.2 Eigenvectors and Eigenvalues - Definitions

Given an $n \times n$ matrix A there are some very special vectors associated to A.

Definition 4.2.1. A vector $\bar{v}$ is an eigenvector for A if $A\vec{v} = \lambda\vec{v}$ for some constant λ. The constant λ is called an eigenvalue.

Example 4.2. Let's define:

$$A = \begin{bmatrix} -5 & 2 \\ -7 & 4 \end{bmatrix} \qquad \text{and} \qquad \bar{v} = \begin{bmatrix} 2 \\ 7 \end{bmatrix}$$

Observe that:

$$A\bar{v} = \begin{bmatrix} -5 & 2 \\ -7 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 7 \end{bmatrix} = \begin{bmatrix} 4 \\ 14 \end{bmatrix} = 2 \begin{bmatrix} 2 \\ 7 \end{bmatrix} = 2\bar{v}$$

Thus we can say that this $\bar{v}$ is an eigenvector with eigenvalue $\lambda = 2$.

4.3 Eigenvectors and Eigenvalues - Finding

Finding eigenvalues and eigenvectors is not overly difficult in theory but the associated calculations can be lengthy and approximations are often necessary.

Because of this nobody finds them by hand except for demonstration of the procedure. Luckly both eigenvectors and eigenvalues can be found easily with software, for example in Python the `numpy` package can do it.

However here is how we find them. The eigenvalues turn out to be the roots of the characteristic polynomial of the matrix. The characteristic polynomial of the matrix may be found as follows:

1. Negate the matrix.
2. Add x to every entry on the diagonal.
3. Take the determinant.

Example 4.3. Consider the matrix:

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

We negate and add x to every entry on the diagonal:

$$\begin{bmatrix} x-2 & 3 \\ 1 & x-4 \end{bmatrix}$$

The characteristic polynomial is then the determinant:

$$(x-2)(x-4)-(3)(1) = x^2-6x+8-3 = x^2-6x+5 = (x-5)(x-1)$$

Thus the eigenvalues are 5 and 1.

To find the eigenvectors for a particular eigenvalue we then solve the corresponding equation.

Example 4.4. In the above example consider the eigenvalue 5. An eigenvector will be a 2×1 vector satisfying:

$$\begin{bmatrix} 2 & -3 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 5 \begin{bmatrix} x \\ y \end{bmatrix}$$
$$\begin{bmatrix} 2x-3y \\ -1x+4y \end{bmatrix} = \begin{bmatrix} 5x \\ 5y \end{bmatrix}$$

This gives us the system:

$$\begin{aligned} 2x-3y &= 5x \\ -1x+4y &= 5y \end{aligned}$$

This simplifies to:

$$\begin{aligned} 3x + 3y &= 0 \\ x + y &= 0 \end{aligned}$$

Observe these two equations are equivalent so as long as we satisfy one of them we satisfy both of them. So for example if $x = 1$ and $y = -1$ we get the eigenvector:

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

5 Graph Partitioning

5.1 The Fiedler Vector and Value

For a connected graph G with n vertices the Laplacian matrix L has n eigenvalues (with some repeats - we won't go into the meaning of this) satisfying:

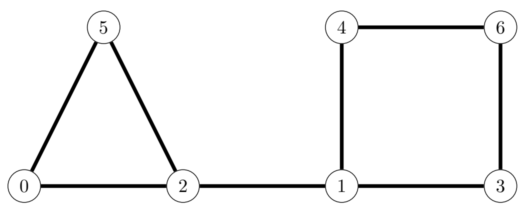
$$0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$$

Moreover in all cases that we'll look at we will in fact have:

$$0 = \lambda_1 < \lambda_2 \leq ... \leq \lambda_n$$

Definition 5.1.1. The smallest positive eigenvalue is called the Fiedler value, named after Miroslav Fiedler (1926 - 2015). An associated eigenvector is called a Fielder vector.

Example 5.1. The graph from the opening of the notes:



We saw has Laplacian matrix:

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

The eigenvalues for this matrix are, with inequalities to reference above:

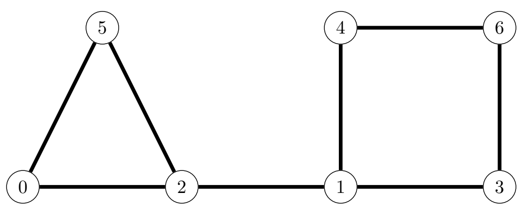
$$0 < 0.3588 < 2.0000 \leq 2.2763 \leq 3.0000 \leq 3.5892 \leq 4.7757$$

The Fielder value is then 0.3588. An associated Fiedler vector is:

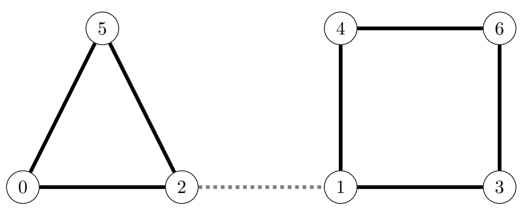
$$\bar{v} = \begin{bmatrix} 0.48 \\ -0.15 \\ 0.31 \\ -0.35 \\ -0.35 \\ 0.48 \\ -0.42 \end{bmatrix}$$

5.2 Graph Partitioning

The goal of this set of notes is to demonstrate a simple way to divide a graph into two "strongly connected" subgraphs when possible. For example consider the graph:



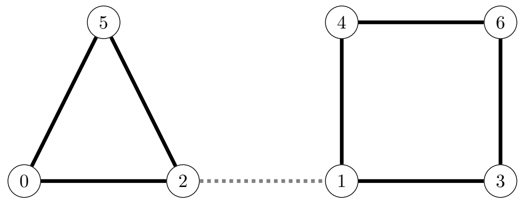
Intuitively this can be viewed as a triangle (on the left, strongly connected) and a square (on the right, strongly connected) with the triangle and square weakly connected to one another, as such:



Given a graph, how can we store this (a data structure!) in a way which allows us to obtain these clusters computationally?

It turns out to be the case that when a graph has a fairly natural partition into two strongly connected subgraphs of equal size then we can identify the nodes in each subgraph by looking at the Fiedler vector and separating the indices by whether the entries are positive or negative. If there happen to be entries which are 0 (happens rarely) they can go either way. We'll group them with the positive entries.

Example 5.2. In the graph we've been examining in our examples the indices in the Fiedler vector which have positive entries are indices $0, 2, 5$ and the indices in the Fiedler vector which have negative entries are indices $1, 3, 4, 6$. If we look at the graph as two subgraphs, one with vertices $0, 2, 5$ and the other with vertices $1, 3, 4, 6$ we see the following:



The Fiedler vector has done what we claimed!

6 Notes

A few notes that may be useful to consider:

1. An eigenvector entry of 0 could go either way in terms of which cluster to put it in.

2. Since any multiple of an eigenvector is also an eigenvector we might wonder what software will do. In practice most software returns a unit eigenvector, meaning it has length 1. However since there are two unit eigenvectors (because we can negate) different software may give different results. I've even seen Matlab give different results!

   Example 6.1. Consider the matrix we saw earlier:

   $$A = \begin{bmatrix} -5 & 2 \\ -7 & 4 \end{bmatrix}$$

   We saw that $\bar{v} = \begin{bmatrix} 2 \\ 7 \end{bmatrix}$ is an eigenvector. To get a unit eigenvector we'd divide by the length $\sqrt{4 + 49} = \sqrt{53}$ and get:

   $$\begin{bmatrix} 2/\sqrt{53} \\ 7/\sqrt{53} \end{bmatrix} \approx \begin{bmatrix} 0.27472112789 \\ 0.96152394764 \end{bmatrix}$$

   Python's `numpy` package gives the negative version of this:

   ```
   % python3
   >>> import numpy as np
   >>> A = np.array([[-5,2],[-7,4]])
   >>> eval,evec = np.linalg.eig(A)
   >>> eval
   array([-3.,  2.])
   ```

```
>>> evec
array([[-0.70710678,  -0.27472113],
       [-0.70710678,  -0.96152395]])
>>>
```

We see the second eigenvalue of 2 corresponding to the second column of the matrix:

$$\begin{bmatrix} -0.27472113 \\ -0.96152395 \end{bmatrix}$$