Linear Algebra - Project

Meet Gera   Yashas S. B.   Rohan

Course Instructor: Dr. Chittraranjan Hens
IIIT Hyderabad

Abstract

Eigenvalues offer a window into the intrinsic properties of graphs, providing a means to comprehend their connectivity, partitioning, symmetry, and dynamics. The exploration of the spectral realm has paved the way for advancements in diverse fields, ranging from computer science and mathematics to social sciences and biology. By leveraging the power of eigenvalues, researchers can unravel the hidden intricacies of graphs and harness their insights to tackle real-world challenges.

1 Introduction

Eigenvalues serve as a key tool in understanding graph connectivity and community structure. By examining the eigenvalues of a graph's adjacency matrix, which captures essential structural information about the graph's connectedness, we can gain valuable insights. The number of zero eigenvalues, known as the algebraic connectivity, provides a measure of the robustness and resilience of the graph. Moreover, the eigenvectors corresponding to the smallest nonzero eigenvalue aid in graph partitioning. Beyond that, eigenvalues unlock further insights into the behavior of graphs. The spectrum of eigenvalues reveals information about the graph's symmetry, regularity, and expansion properties. Large eigenvalues indicate the presence of well-connected clusters or highly influential nodes, while small eigenvalues suggest sparseness or potential bottlenecks. Furthermore, the distribution of eigenvalues allows for the identification of graph families and the comparison of graph similarity. Eigenvalues offer a window into the intrinsic properties of graphs, providing a means to comprehend their connectivity, partitioning, symmetry, and dynamics. The exploration of the spectral realm has paved the way for advancements in diverse fields, ranging from computer science and mathematics to social sciences and biology. By leveraging the power of eigenvalues, researchers can unravel the hidden intricacies of graphs and harness their insights to tackle real-world challenges.

2 Bipartite Graphs

2.1 Mathematical Representation

A graph is a collection of nodes and edges, denoted as $G = (V, E)$. In the case of bipartite graphs, there are two sets of nodes such that there are no edges between any two nodes of the same set. Mathematically, we can represent this as $\exists V_1, V_2$ such that $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$. The representation of bipartite graphs is given by $G = (V_1, V_2, E)$.
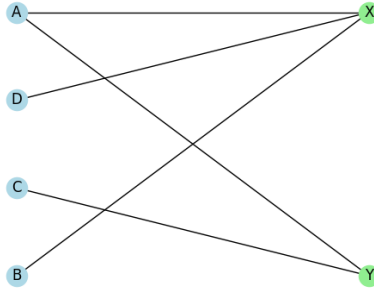

Figure 1: Example of a Bipartite Graph

2.2 Adjacency Matrix

The adjacency matrix $\mathbf{A}$ is a square matrix of size $n \times n$, where n is the number of nodes in the graph. For a bipartite graph, the adjacency matrix is partitioned into four blocks, capturing the connections between nodes in V_1, nodes in V_2, and edges between V_1 and V_2. The adjacency matrix of a bipartite graph takes the form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix}$$

where $\mathbf{0}$ represents the zero matrix and $\mathbf{B}$ captures the edges connecting V_1 and V_2. The adjacency matrix provides a concise representation of the graph's structure and facilitates various graph analysis techniques.

2.3 Eigenvalues of Bipartite Graphs

The eigenvalues of a bipartite graph's adjacency matrix exhibit an interesting property. Let $\mathbf{A}$ be the adjacency matrix of a bipartite graph. By definition, a bipartite graph has its vertices divided into two disjoint sets such that all edges

connect vertices from different sets. The eigenvalues of **A** appear in pairs of $+a$ and $-a$, where a is a real number. This property holds true due to the specific structure of bipartite graphs and has important implications for understanding their spectral properties.
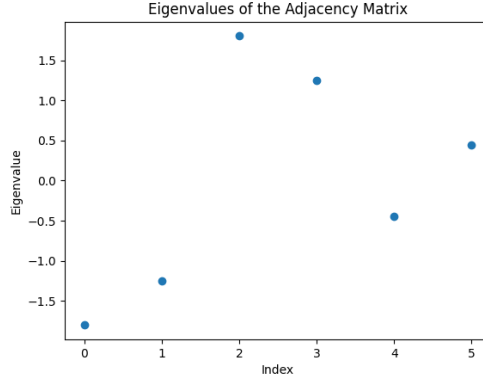

Figure 2: Example of a eigenvalues in Complex Plane

3   Relation with Eigenvalue and Proof

Theorem 1 (Eigenvalues of Bipartite Graphs). *Let* **A** *be the adjacency matrix of a bipartite graph. By definition, a bipartite graph has its vertices divided into two disjoint sets such that all edges connect vertices from different sets. The eigenvalues of* **A** *appear in pairs of* $+a$ *and* $-a$, *where a is a real number.*

Proof. We will prove the statement by considering the properties of the adjacency matrix of a bipartite graph.

Step 1: Matrix A^2

Consider the matrix $\mathbf{A}^2$. The (i, j)th entry of $\mathbf{A}^2$ represents the number of paths of length 2 between vertex i and vertex j. In a bipartite graph, there are no direct edges between vertices within the same set, so all entries on the diagonal of $\mathbf{A}^2$ are zero.

Step 2: Equivalence of A^2v and Av^2

For any vector **v**, we have $(\mathbf{A}^2)\mathbf{v} = \mathbf{A}(\mathbf{v}^2)$. This equation can be proven using matrix multiplication.

Step 3: Eigenvalue λ and eigenvector v

Let λ be an eigenvalue of **A** with eigenvector **v**. By definition, $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$.

Step 4: From $Av = \lambda v$ **to** $Av^2 = \lambda(Av)$

Multiply both sides of the equation by $\mathbf{A}$: $\mathbf{A}(\mathbf{Av}) = \mathbf{A}(\lambda\mathbf{v})$. This yields $\mathbf{A}^2\mathbf{v} = \lambda(\mathbf{Av})$.

Step 5: Substituting Step 2

By substituting Step 2, we have $\mathbf{Av}^2 = \lambda(\mathbf{Av})$. This implies that λ is an eigenvalue of $\mathbf{A}^2$ with eigenvector $\mathbf{v}^2$.

Step 6: Non-negativity of A^2

The eigenvalues of $\mathbf{A}^2$ are non-negative. This is because $\mathbf{A}^2$ is a non-negative matrix representing the number of paths of length 2 in the bipartite graph.

Step 7: Eigenvalue λ **is non-negative**

Since λ is an eigenvalue of $\mathbf{A}^2$, we have $\lambda \geq 0$.

Step 8: Assumption: $\lambda \neq 0$

Let's assume $\lambda \neq 0$. If $\lambda \neq 0$, then the eigenvector $\mathbf{v}^2$ is nonzero.

Step 9: From $Av = \lambda v$ **to** $\lambda = \frac{v^T A^T v}{v^T v}$

By the definition of the eigenvalue, we have $\mathbf{Av} = \lambda\mathbf{v}$. Multiplying both sides of the equation by $\mathbf{v}^T$, we get $(\mathbf{v}^T\mathbf{Av}) = \lambda(\mathbf{v}^T\mathbf{v})$.

Step 10: Symmetric property

Since $\mathbf{A}^T = \mathbf{A}$ for a bipartite graph, we have $\mathbf{A}^T\mathbf{v} = \mathbf{Av}$. This implies that $(\mathbf{v}^T\mathbf{A}^T\mathbf{v}) = (\mathbf{v}^T\mathbf{Av})$.

Step 11: Combining Step 9 and Step 10

Combining steps 9 and 10, we get $\lambda = \frac{v^T A^T v}{v^T v}$.

Step 12: Eigenvalue λ **is real**

Since $\lambda = \frac{v^T A^T v}{v^T v}$, it follows that λ is a real number.

Step 13: Conclusion: Eigenvalues appear in pairs

Therefore, we conclude that the eigenvalues of $\mathbf{A}$ appear in pairs of $+a$ and $-a$, where a is a real number.
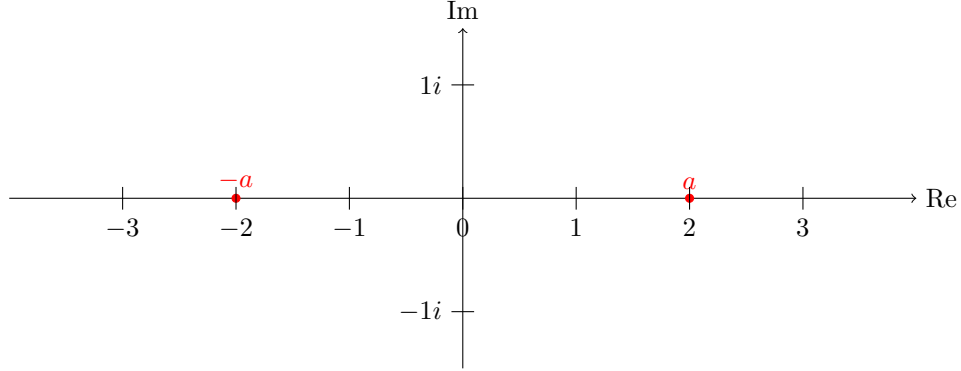
$\square$

Figure 3: Eigenvalues of a Bipartite Graph

4   Conclusion for Bipartite Graphs

In conclusion, the study of bipartite graphs and their eigenvalues holds great promise for understanding graph connectivity, community structure, and various real-world applications. The exploration of eigenvalues provides valuable insights into the robustness, resilience, and behavior of bipartite graphs.

The application of bipartite graphs extends to various domains, including matching and assignment problems, biological networks, and social network analysis. By leveraging the power of eigenvalues, researchers can unravel the hidden intricacies of bipartite graphs and harness their insights to tackle real-world challenges.

Looking to the future, the utilization of bipartite graphs and their eigenvalues is expected to continue advancing. As data becomes increasingly complex and interconnected, bipartite graphs offer a powerful framework for modeling and analyzing large-scale networks. From personalized recommendations to understanding information diffusion, bipartite graphs have the potential to revolutionize various fields and drive innovative solutions.

In conclusion, the study of bipartite graphs and their eigenvalues opens up new avenues for research and practical applications. By delving into the spectral realm of eigenvalues, we can unlock the hidden patterns and structures within bipartite graphs, paving the way for further advancements in graph theory and its applications.

So, let us embrace the power of eigenvalues and embark on a journey to unravel the mysteries of bipartite graphs. With each step, we deepen our understanding of their connectivity, partitioning, symmetry, and dynamics, ultimately enabling us to tackle real-world challenges with confidence and precision.

5 Clustering in Graphs using Eigenvalues

5.1 What is Clustering?

Clustering is the task of grouping objects in a way such that objects belonging to a certain cluster are more similar to each other than to those objects not in the cluster. It is one of the more important techniques in several data analysis fields like pattern recognition, image compression, bioinformatics, machine learning and several others. Clustering in graphs using eigenvalues and eigenvectors of the adjacency matrix is a widely used technique known as spectral clustering.

5.2 Spectral Clustering

Spectral clustering is a technique with roots in graph theory, where the approach is used to identify communities of nodes in a graph based on the edges connecting them. Spectral clustering leverages the spectral properties of the graph Laplacian, which is derived from the adjacency matrix, to partition the graph nodes into cohesive clusters. In statistics, spectral clustering techniques make use of the eigenvalues of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. The similarity matrix which is provided as the input consists of quantitative assessment of the relative similarity of different pairs of points in the dataset.

5.3 Basic Terminologies used in Spectral Clustering

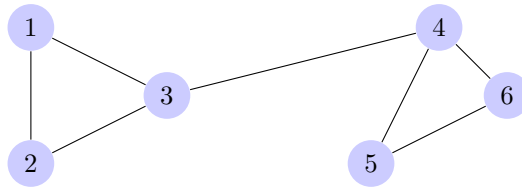We will try to explain the basic terminologies in Spectral Clustering using a graph.

Figure 4: A simple graph with 6 nodes

5.3.1 Adjacency and Affinity Matrix (A)

The graph (or set of data points) can be represented as an Adjacency Matrix, where the row and column indices represent the nodes, and the entries represent the absence or presence of an edge between the nodes (i.e. if the entry in row 0 and column 1 is 1, it would indicate that node 0 is connected to node 1). An Affinity Matrix is like an Adjacency Matrix, except the value for a pair of points expresses how similar those points are to each other. If pairs of points are very dissimilar then the affinity should be 0. If the points are identical, then the

affinity might be 1. In this way, the affinity acts like the weights for the edges on our graph.

Below is the Adjacency/Affinity matrix of the graph above:

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

5.3.2 Degree Matrix (D)

A Degree Matrix is a diagonal matrix, where the degree of a node (i.e. values) of the diagonal is given by the number of edges connected to it. We can also obtain the degree of the nodes by taking the sum of each row in the adjacency matrix.

Below is the Degree matrix of the adjacency matrix we defined above:

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

5.3.3 Laplacian Matrix (L)

This is another representation of the graph/data points, which attributes to the beautiful properties leveraged by Spectral Clustering. One such representation is obtained by subtracting the Adjacency Matrix from the Degree Matrix (i.e. $L = D–A$).

Below is the Laplacian matrix of the graph we defined above:

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

1. **Spectral Gap**: The first non-zero eigenvalue of L is called the Spectral Gap. The Spectral Gap gives us some notion of the density of the graph.

2. **Fiedler Value**: The second eigenvalue of L is called the Fiedler Value, and the corresponding vector is the Fiedler vector. Each value in the Fiedler vector gives us information as to which side of the decision boundary a particular node belongs to.

3. Using L, we find the first large gap between eigenvalues which generally indicates that the number of eigenvalues before this gap is equal to the number of clusters.

5.4 Algorithm for spectral clustering in graphs

The three major steps involved in spectral clustering are: constructing a similarity graph, projecting data onto a lower-dimensional space, and clustering the data.

Step 1: Form a distance matrix.

Step 2: Transform the distance matrix into an affinity matrix A.

Step 3: Compute the degree matrix D and the Laplacian matrix $L = D–A$.

Step 4: Find eigenvalues and eigenvectors of L.

Step 5: With the eigenvectors of k largest eigenvalues computed from the previous step form a matrix.

Step 6: Normalize the vectors.

Step 7: Cluster the data points in k-dimensional space.

5.5 Why use spectral clustering over other forms of clustering?

1. **Flexibility in Handling Complex Data:** Spectral clustering is capable of identifying clusters with complex shapes and structures. It is not limited to spherical or convex clusters like some other algorithms such as k-means. Spectral clustering can handle clusters that are non-linear, disconnected, or overlapping, making it more suitable for a wide range of datasets.

2. **Connectivity-Based Clustering:** Spectral clustering takes into account the connectivity or pairwise relationships between data points. Points that are close in proximity but not directly connected can still be assigned to the same cluster. This allows spectral clustering to capture meaningful relationships between data points and identify clusters based on connectivity patterns.

3. **Scalability with Sparse Graphs:** Spectral clustering can be particularly efficient when working with sparse graphs. The use of sparse matrix representations and efficient algorithms for eigenvalue computation can help reduce computational costs, making it feasible for large-scale datasets.
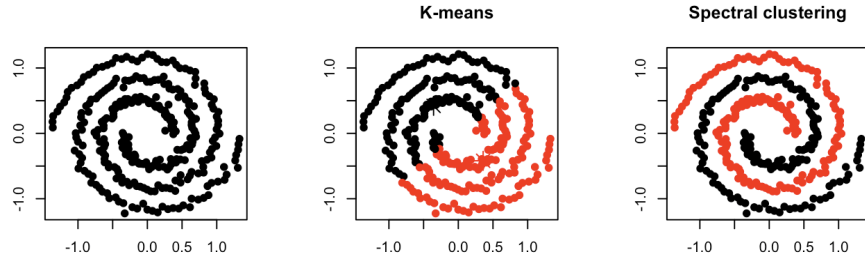
.

Figure 5: Clustering with different clustering algorithms

5.6 Some cons of Spectral Clustering

1. **Computational Complexity:**Spectral clustering involves the computation of eigenvalues and eigenvectors of the similarity matrix or Laplacian matrix, which can be computationally expensive for large datasets. The time and memory requirements of spectral clustering can be higher compared to simpler clustering algorithms like k-means.

2. **Parameter Selection:**Spectral clustering requires the selection of various parameters, such as the number of clusters (k) and the type of affinity matrix or similarity measure to use. Choosing the optimal values for these parameters can be challenging and may require domain knowledge or trial-and-error experimentation.

3. **Lack of Robustness to Noise and Outliers:**Spectral clustering is not inherently robust to noise or outliers in the data. Outliers can influence the computation of the affinity matrix and affect the clustering results. Preprocessing steps or outlier detection techniques may be needed to handle noisy data effectively.

6