33 Spectral Clustering: Finding Hidden Communities in Networks

33.1 Seeing Social Networks as Graphs

In everyday life, we often think about social relationships as a network: each person is a node, and a friendship between two people is a connection, or edge. Representing relationships in this mathematical form allows us to visualize and analyze complex social structures. When a network becomes very large, it is natural to ask questions such as: Who forms a close community? Which groups are tightly connected? Is the network likely to split into separate factions?

To answer such questions, graph theory and linear algebra offer powerful tools. In particular, this chapter introduces spectral clustering, a method that detects communities by studying the structure of a graph through its Laplacian matrix and the eigenvalues and eigenvectors derived from it.

33.1.1 Undirected Graphs: Mapping Mutual Relationships

Graph theory studies structures made up of nodes (also called vertices) and edges, which describe relationships or interactions between nodes. An undirected graph is a graph in which edges have no direction; a connection simply indicates a mutual relationship between two nodes. Many real-world networks, such as friendship networks, can be modeled this way, because "A is connected to B" naturally implies "B is connected to A."

A classic example is the Zachary's Karate Club dataset, shown in Figure 1. In this social network, each node (numbered) represents a club member, and each edge indicates a social tie, such as regular training or personal interaction.

This graph was introduced by Wayne Zachary in a study on social dynamics. During the real event, disagreements between two leaders caused the club to split into two factions, making it an ideal dataset for studying community detection. We will use this graph as our running example when explaining spectral clustering later in this chapter.

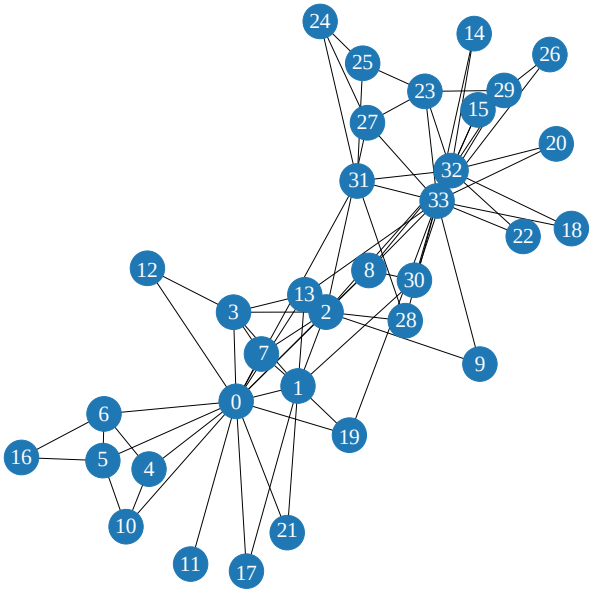

Figure 1. Social network of Zachary's karate club, where nodes represent members and edges represent friendship ties. Figure generated by Ch33_01_spectral_clustering.ipynb.

33.1.2 Directed Graphs: When Connections Have Direction

A directed graph is a graph in which each edge has a direction, indicating a one-way relationship from one node to another. Unlike undirected graphs, where connections are mutual, a directed edge carries additional information: it tells us who influences whom, who points to whom, or who depends on whom. Directed graphs are therefore especially useful for modeling systems that involve directionality, causality, or asymmetric interactions.

Many real-world networks are inherently directional. Information flows from a sender to a receiver, citations go from a newer paper to an older one, and in recommendation systems, a user may "follow" a creator without receiving the same attention in return.

In nature, directional relationships are even more common. For example, Figure 2 illustrates a food chain in an ecosystem. Each node represents a species, and each directed edge shows the direction of energy transfer. Grass is eaten by rabbits, so energy flows from grass to rabbit; rabbits are preyed upon by wolves, so energy flows from rabbits to wolves. This one-way flow reveals not only who depends on whom, but also how the entire ecosystem maintains its balance.

Just as undirected graphs can uncover community structures, directed graphs allow us to analyze influence, hierarchy, dominance, or feedback loops within a network.

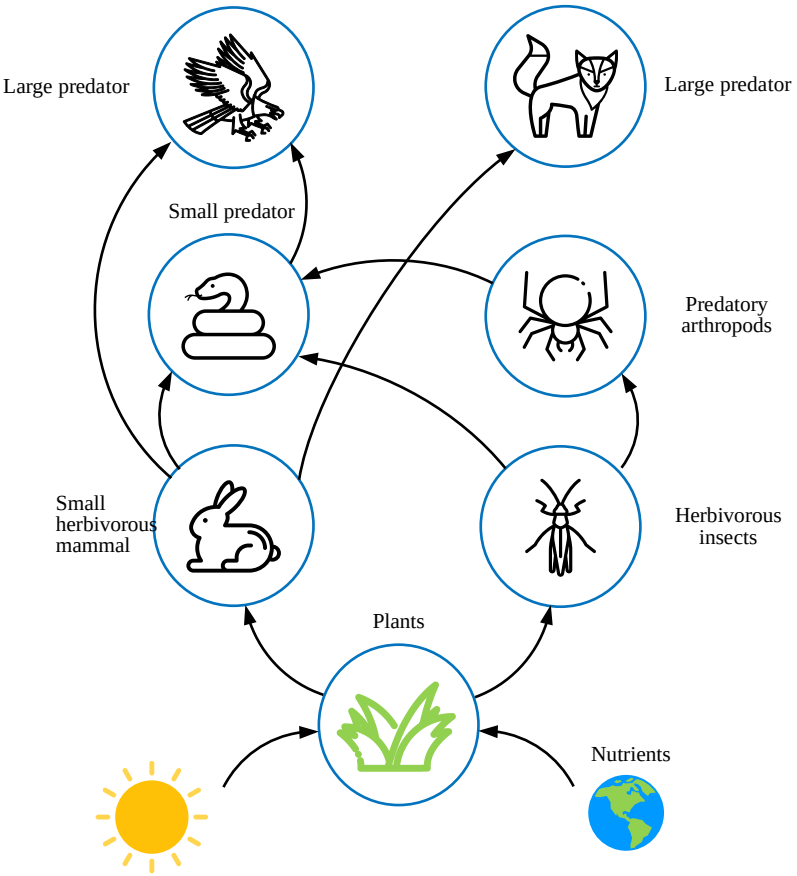

Figure 2. A directed food-chain graph, where arrows indicate the flow of energy between species.

33.1.3 From People to Matrices: Translating Networks into Numbers

Graphs and matrices are deeply connected, and in graph theory. This connection comes from the adjacency matrix, a matrix representation that records how nodes in a graph are linked. The adjacency matrix allows us to translate a visual network into a numerical form that is easy for computers to store, process, and analyze. Because many graph algorithms rely on linear algebra, this matrix-based view becomes extremely powerful.

Figure 3 illustrates the relationship between an unweighted undirected graph and its adjacency matrix. An unweighted undirected graph is a set of nodes connected by edges that have no direction and no numerical weight. Each edge simply indicates a mutual connection between two nodes.

For such a graph, the rule of building its adjacency matrix is simple: if there is an edge between two nodes, the corresponding matrix entry is 1; if no edge exists, the entry is 0. This binary structure captures the entire connectivity pattern of the network in a compact and systematic way.

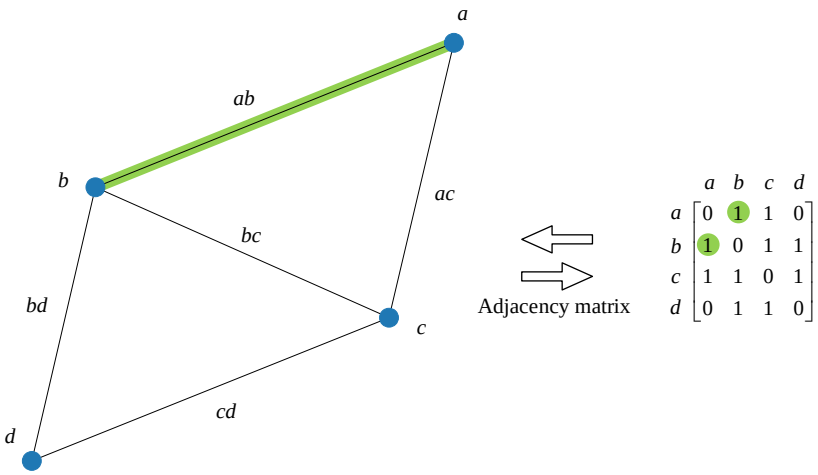

Figure 3. An unweighted undirected graph and its corresponding adjacency matrix.

When the graph has weights, the adjacency matrix becomes even more expressive. In a weighted undirected graph, the matrix entry $w_{i,j}$ represents the weight of the edge between node i and node j.

If there is no edge, then $w_{i,j} = 0$. Because the graph is undirected, the matrix remains symmetric, meaning $w_{i,j} = w_{j,i}$. This symmetric weighted matrix not only tells us whether nodes are connected, but also how strongly they are connected.

By converting graphs into matrices, we make it possible to apply tools from linear algebra—such as eigenvalue decomposition—which will play a central role in spectral clustering later in this chapter.

33.1.4 How Spectral Clustering Works: A Step-by-Step Pipeline

Spectral clustering is a powerful technique for grouping nodes in a graph based on the structure of their connections. Here, we apply spectral clustering to the weighted undirected graph shown in Figure 1.

The main idea behind spectral clustering is to use the graph's connectivity information to find natural clusters. The process can be understood intuitively in several steps.

- First, we compute the graph's adjacency matrix, which encodes which nodes are connected and the strength of their connections.

- Next, we calculate the degree matrix, which captures how strongly each node is connected to the rest of the graph.
- Using these two matrices, we construct the graph Laplacian, a key tool that reflects the graph's structure in a way suitable for analysis. To make the process more stable and comparable across nodes, we compute the normalized Laplacian.
- Once we have the normalized Laplacian, we perform an eigenvalue decomposition to find its eigenvectors. These eigenvectors provide a new, low-dimensional representation of the graph where nodes that are strongly connected tend to be close together.
- By selecting the eigenvectors corresponding to the smallest eigenvalues, we create this low-dimensional space. Finally, we apply a standard clustering method, such as *k*-means, in this space to group the nodes into clusters.

In the following sections, we will go step by step through this process, explaining each component intuitively and showing how it contributes to revealing the underlying cluster structure in the graph.

33.2 Building the Mathematical Foundations of a Graph

33.2.1 The Adjacency Matrix — Capturing Who Connects with Whom

Based on the connections between nodes in the undirected graph shown in Figure 1, we first compute the adjacency matrix $\boldsymbol{A}$, illustrated in Figure 4. In this heatmap, each square represents the strength of a relationship between two members, such as how often they train, socialize, or interact together.

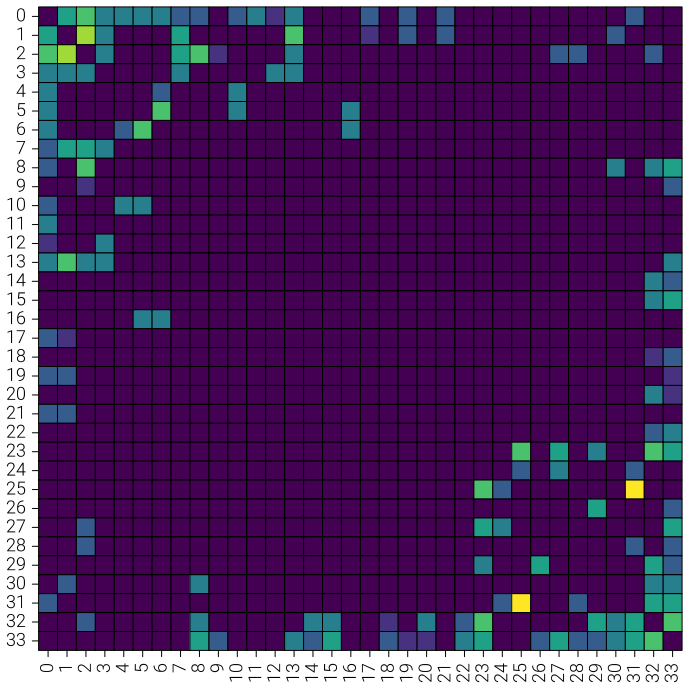

Figure 4. Adjacency matrix of the karate club. Figure generated by Ch33_01_spectral_clustering.ipynb.

Since relationships vary in strength, we assign weights to the edges to reflect these differences. A higher weight (max of 7 in this example) indicates a stronger connection, meaning the two members interact frequently

or share a closer bond, whereas a lower weight (for instance 1) indicates a weaker or occasional interaction. 0 means there is no direct connection between the two members — they do not interact or have no observable relationship within the network.

By including these weights, the adjacency matrix captures not only the presence of a connection but also its intensity.

This weighted undirected graph thus provides a detailed map of social relationships within the club. It allows us to see not only who is connected to whom but also which members are more central or closely connected within the network. Such a representation forms a rich foundation for spectral clustering, as it preserves both the structure and the strength of the relationships among members.

33.2.2 The Degree Matrix — Measuring How Connected Each Node Is

The next step is to compute the degree matrix, denoted as $\boldsymbol{D}$. As shown in Figure 5, the degree matrix is a diagonal matrix that captures the total connection strength of each node. For node i, its degree is defined as the sum of the weights of all edges connected to it.

Thus, each diagonal entry of $\boldsymbol{D}$, for instance, $d_{i,i}$ equals the total number of edges connected to node i, which is the sum of the i-th row (or column) of the adjacency matrix $\boldsymbol{A}$.

In simpler terms, the degree of a node reflects how connected it is within the network. A higher degree indicates that the node is more active or influential, maintaining direct connections with many other nodes. Such nodes often play key roles in spreading information, distributing resources, or maintaining the social structure of the network. Conversely, nodes with lower degrees have fewer connections, suggesting they occupy peripheral positions, participate less in interactions, or may even be relatively isolated.

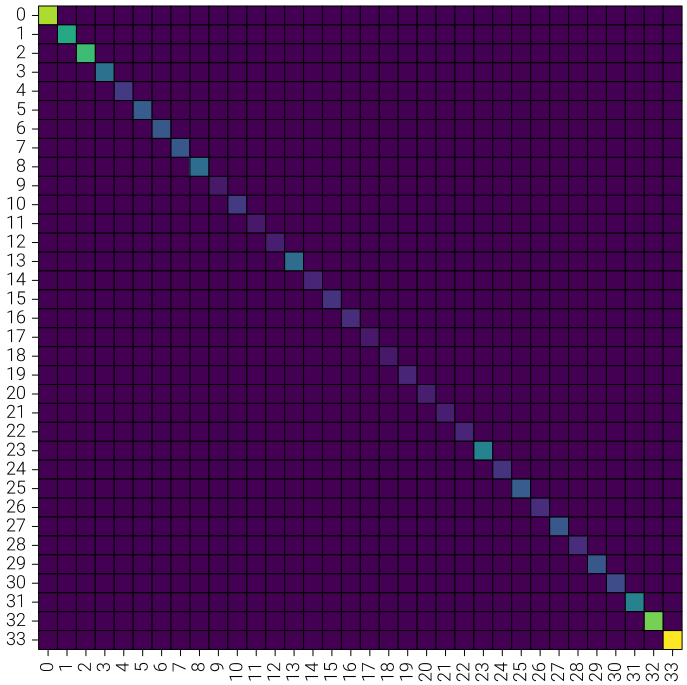

Figure 5. Degree matrix of the karate club graph; Figure generated by Ch33_01_spectral_clustering.ipynb.

Figure 6 visualizes the degree of each node as a bar chart. It is easy to see that nodes 0 and 33 have particularly high degrees, marking them as “core members” of the network.

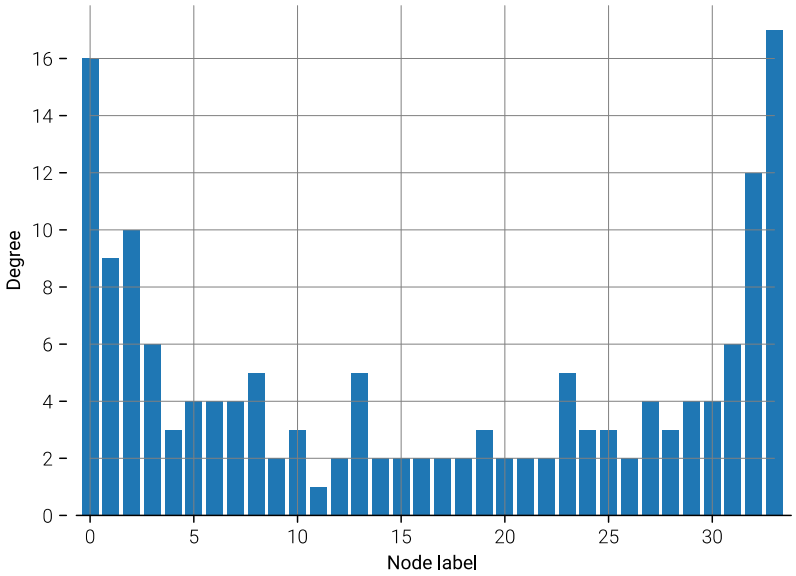

Figure 6. Node degree distribution; Figure generated by Ch33_01_spectral_clustering.ipynb.

Overall, the degree matrix provides a concise way to quantify the direct influence and engagement of each node. In Figure 7, node colors represent their degrees — brighter nodes have higher degrees and are more central in the network. This visualization already suggests the existence of two primary social groups, or "friend circles," within the club.

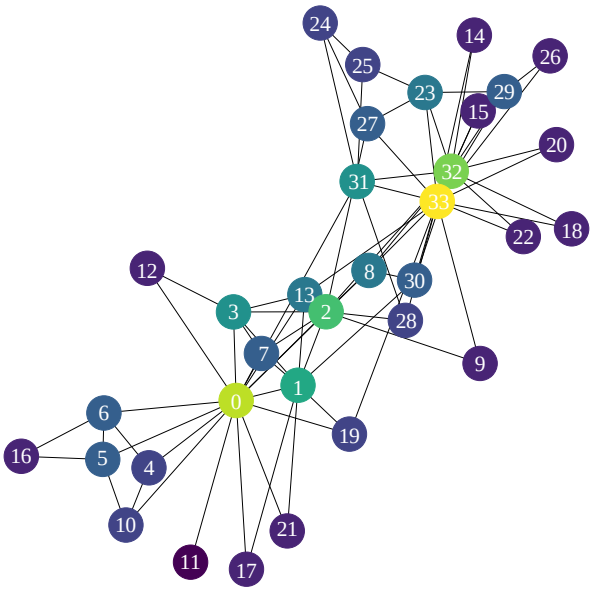

Figure 7. Nodes rendered by degree; Figure generated by Ch33_01_spectral_clustering.ipynb.

33.2.3 The Laplacian Matrix — Revealing the Structure Beneath the Surface

Once we have the degree matrix $\boldsymbol{D}$ and the adjacency matrix $\boldsymbol{A}$, we can compute the graph Laplacian, denoted as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \tag{1}$$

as shown in Figure 8.

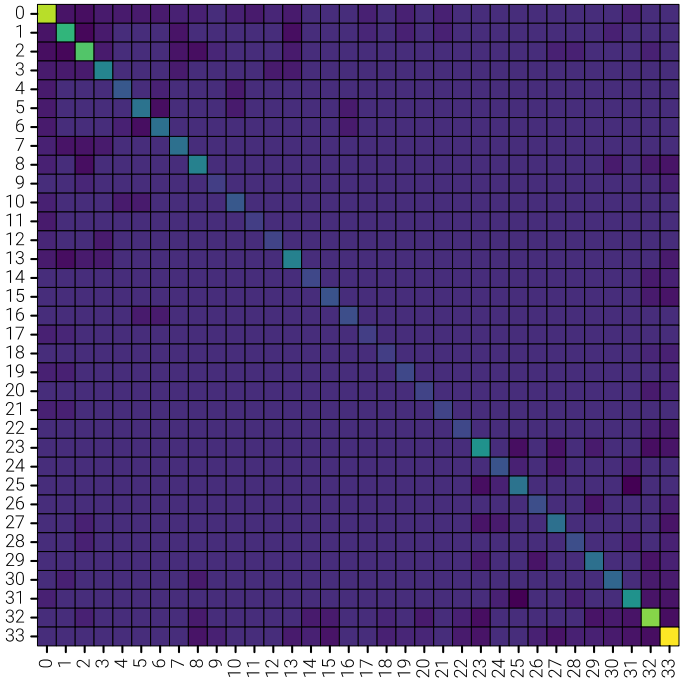

Figure 8. Laplacian matrix of the karate club graph; Figure generated by Ch33_01_spectral_clustering.ipynb.

For an undirected graph, both $\boldsymbol{D}$ and $\boldsymbol{A}$ are symmetric matrices, so the Laplacian $\boldsymbol{L}$ is also symmetric.

The Laplacian matrix is a fundamental representation of a graph, encoding not only the connections between nodes but also the overall structure of the network. Each entry in $\boldsymbol{L}$ reflects how strongly a node is connected to others or how isolated it is relative to its neighbors.

Beyond representing connectivity, the Laplacian has powerful analytical properties. Its eigenvalues and eigenvectors can reveal the internal division of the network, helping us identify clusters or communities within the graph. Intuitively, the Laplacian captures the "flow" of connectivity in the graph, providing the mathematical foundation for techniques like spectral clustering, where these structural patterns are used to group nodes into coherent clusters.

33.2.4 The Normalized Laplacian — Balancing the Influence of Each Node

To reduce the impact of differences in node degrees, we often compute the symmetrically normalized Laplacian, defined as

$$\boldsymbol{L}_{\text{normalized}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{D}^{-1/2}\left(\boldsymbol{D} - \boldsymbol{A}\right)\boldsymbol{D}^{-1/2} \tag{2}$$

as shown in Figure 9.

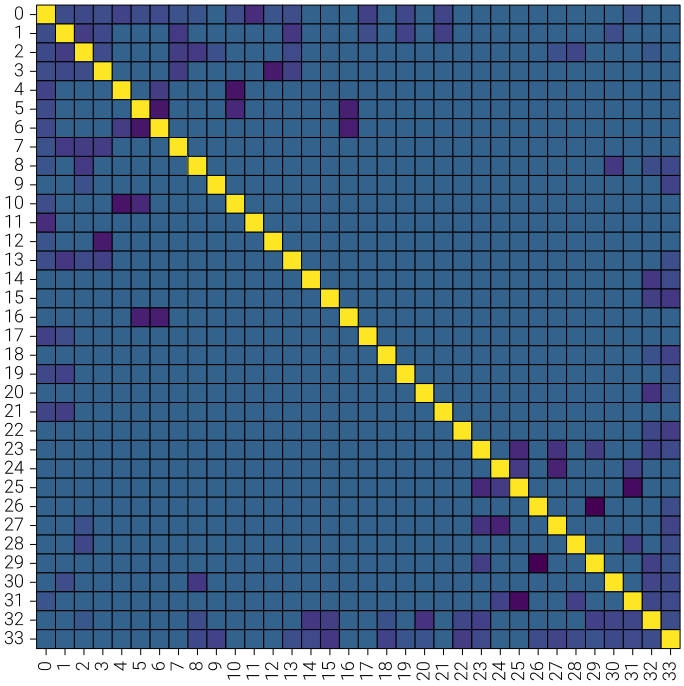

Figure 9. Normalized Laplacian matrix of the karate club graph; Figure generated by Ch33_01_spectral_clustering.ipynb.

This operation scales the original Laplacian $\boldsymbol{L}$ by the inverse square root of the degree matrix on both sides, which is equivalent to rescaling each row and column of $\boldsymbol{L}$.

In the normalized Laplacian, the diagonal elements are all equal to 1. This adjustment ensures that nodes with very high degrees do not dominate the overall structure, which can happen in graphs where some nodes have many more connections than others. Without normalization, clustering or graph partitioning could be biased toward these highly connected nodes.

By normalizing, we effectively standardize the connection strengths of all nodes, giving each node a more balanced influence in subsequent calculations. This creates a fairer representation of the network, which is especially important for spectral clustering, where the normalized Laplacian helps uncover clusters based on the intrinsic structure rather than just the most connected nodes.

33.3 The Spectral Perspective: Learning from Eigenvalues and Eigenvectors

33.3.1 Eigenvalue Decomposition — Turning Structure into Geometry

After computing the normalized Laplacian matrix, the next step is to perform an eigenvalue decomposition

$$\boldsymbol{L}_{\text{normalized}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}} \tag{3}$$

which breaks the matrix into its eigenvalues and eigenvectors, as shown in Figure 10. Because normalized Laplacian matrix is a real symmetric matrix, this decomposition is a spectral decomposition, producing a set of real eigenvalues and an orthogonal matrix of eigenvectors. Thus, in (3), the transpose of $\boldsymbol{V}$ appears instead of its inverse.

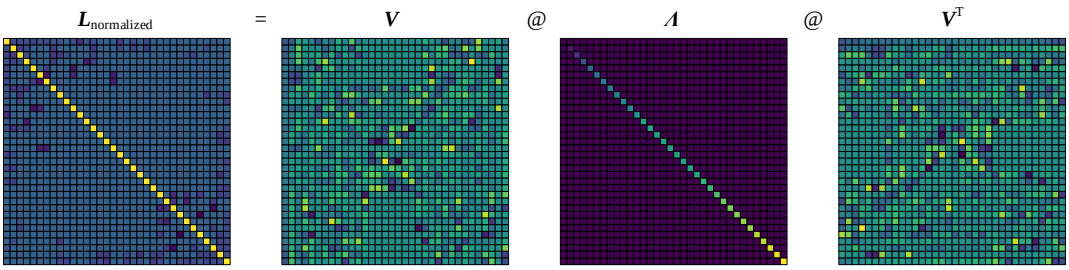

Figure 10. Eigenvalue decomposition of the normalized Laplacian matrix. Figure generated by Ch33_01_spectral_clustering.ipynb.

33.3.2 The Eigenvector Space — Seeing the Network in Two Dimensions

The eigenvalues are arranged in ascending order, and their corresponding eigenvectors are collected as columns in the orthogonal matrix $\boldsymbol{V}$. Each column of $\boldsymbol{V}$ represents a direction in a new space where the graph's connectivity structure can be analyzed.

For our relatively simple undirected graph, we select the two eigenvectors corresponding to the smallest nonzero eigenvalues—these are the first two columns of $\boldsymbol{V}$, denoted as $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Using these two eigenvectors, we construct a low-dimensional representation of the nodes, shown in Figure 11.

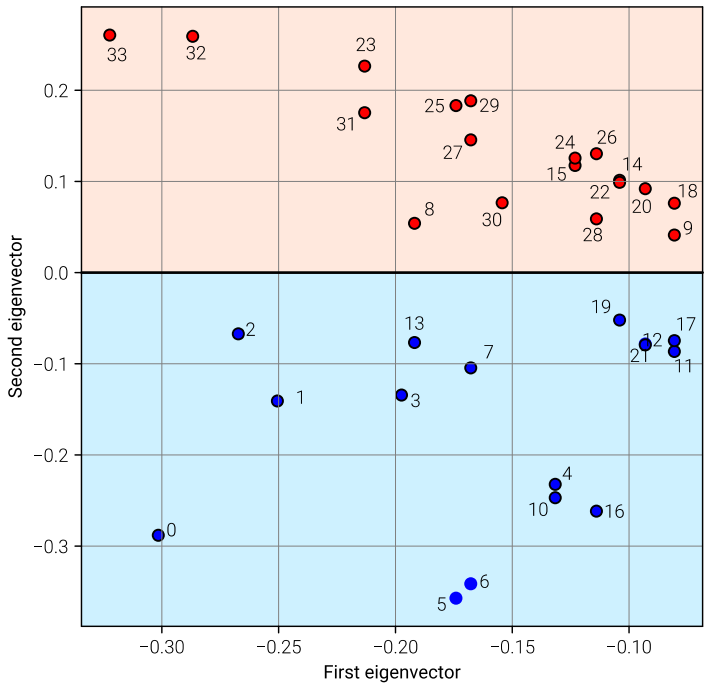

Figure 11. Nodes mapped to the low-dimensional eigenvector space. Figure generated by Ch33_01_spectral_clustering.ipynb.

33.3.3 From Geometry to Clusters — Dividing the Graph into Communities

In Figure 12., the network is divided into two distinct clusters, with red and blue nodes representing the two groups. This separation arises naturally from the signs of the second eigenvector of the normalized Laplacian: nodes with positive values form one cluster (shown in red), while those with negative values form the other (shown in blue).

Intuitively, the normalized Laplacian encodes how strongly nodes are connected to each other. Nodes that are closely linked in the original graph appear near one another in the eigenvector space, whereas weakly connected nodes are positioned farther apart. Unlike Principal Component Analysis (PCA), which seeks directions of maximum variance in data, spectral clustering leverages the eigenvectors of the Laplacian to preserve graph connectivity. This transformation projects the network into a lower-dimensional space where communities become visually and analytically distinct, making the underlying social structure easy to identify.

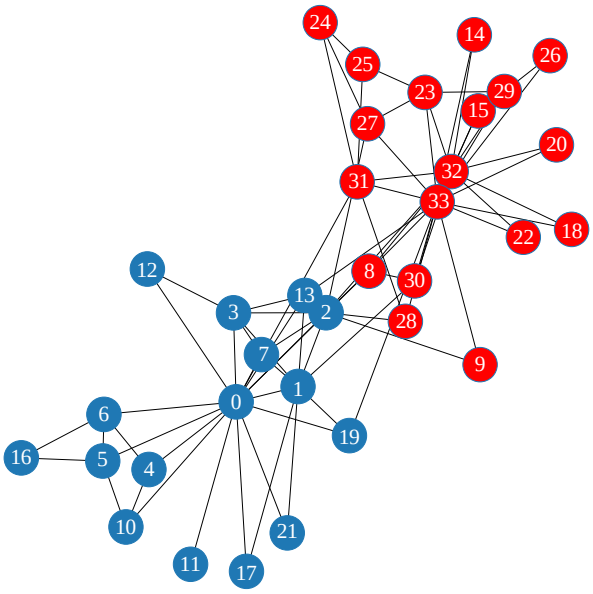

Figure 12. Graph partitioning into two clusters. Figure generated by Ch33_01_spectral_clustering.ipynb.

33.4 Conclusion

Spectral clustering is a method for discovering communities within a network by analyzing its structure mathematically. A network, or graph, consists of nodes representing entities and edges representing relationships. In an undirected graph, edges indicate mutual connections, while in a weighted graph, edge weights capture the strength of these relationships. Spectral clustering leverages this structure by translating the graph into matrices, starting with the adjacency matrix to record connections, and the degree matrix to measure each node's total connectivity.

Using these, we construct the Laplacian matrix, which encodes the network's connectivity patterns and overall structure. To avoid bias from highly connected nodes, we compute the normalized Laplacian, ensuring all nodes contribute fairly. Next, we perform eigenvalue decomposition on the normalized Laplacian, producing eigenvectors that map nodes into a low-dimensional space. In this space, nodes that are strongly connected in the original network are positioned close together, while weakly connected nodes are farther apart.

By selecting eigenvectors corresponding to the smallest eigenvalues, we capture the graph's intrinsic clustering structure. Standard clustering algorithms, like k-means, can then group nodes into coherent communities. This approach transforms complex network connectivity into an intuitive geometric representation, making it easier to detect natural groups, as illustrated using the Zachary Karate Club dataset, where the method successfully separates members into two distinct clusters.