# Spectral Clustering

A Graph-Based Approach to Data Clustering

## What is it?

Spectral Clustering is a clustering algorithm that uses the **eigenvalues and eigenvectors** of a similarity matrix to perform dimensionality reduction before clustering. Unlike traditional methods like K-means that work directly on raw features, it transforms data into a space where clusters are more easily separable.

## The Big Idea

Think of your data points as nodes in a network connected by weighted edges (similarities). Spectral clustering finds the best way to "cut" this network into groups by:

1. Converting similarities into a graph
2. Finding a lower-dimensional embedding that preserves cluster structure
3. Clustering in this new space where groups are naturally separated

**Analogy:** It's like untangling a messy ball of yarn - you first figure out which strands naturally belong together by looking at the overall connectivity pattern, rather than just measuring distances.

## How it Works (Simple Version)

1. **Build a Similarity Graph**
   Calculate pairwise similarities between all points (often using Gaussian/RBF kernel). Create a graph where points are nodes and edges represent similarities.
2. **Construct the Graph Laplacian**
   Build the Laplacian matrix L = D - W, where W is the similarity/adjacency matrix and D is the degree matrix (diagonal matrix of row sums).
3. **Compute Eigenvectors**
   Find the k smallest eigenvectors of the Laplacian. These eigenvectors form a new representation of your data.
4. **Cluster in New Space**
   Treat each eigenvector as a feature and run K-means on this new representation.

## Key Formulas

**Graph Laplacian (Unnormalized):**

$$L = D - W$$

**Normalized Symmetric Laplacian:**

$$L_{sym} = D^{(-1/2)} \cdot L \cdot D^{(-1/2)} = I - D^{(-1/2)} \cdot W \cdot D^{(-1/2)}$$

**Similarity (RBF Kernel):**

$$W_{ij} = \exp(-||x_i - x_j||^2 / 2\sigma^2)$$

## Pros & Cons

### ✓ Pros

- Can identify **non-convex clusters** (arbitrary shapes) that K-means would miss
- Works well when clusters are connected in complex ways
- Based on solid graph theory foundations
- Often more robust to outliers than K-means

### ✗ Cons

- **Computationally expensive** - requires eigendecomposition ($O(n^3)$ for n points)
- Sensitive to choice of similarity metric and scaling parameter (σ)
- Still needs to specify number of clusters k upfront
- Memory intensive for large datasets

## When Should You Use It?

### Use spectral clustering when:

- Your clusters have **complex, non-spherical shapes**
- Points within a cluster are connected but not necessarily close in Euclidean space
- You have good domain knowledge to set similarity metrics
- Dataset is small to medium-sized (< 10,000 points typically)
- K-means or other simple methods fail

### Don't use it when:

- You have very large datasets (too slow)
- Clusters are roughly spherical and well-separated (K-means is faster)
- You lack computational resources

## Common Uses

- Image Segmentation
- Community Detection
- Document Clustering
- Gene Expression Analysis
- Computer Vision
- Market Segmentation

💡 **Key Insight:** Spectral clustering excels when the "shape" of your clusters matters more than their location in feature space.