Spectral Clustering

Presented by:
Media Salih
Matina Rahbar
Mebrooka Rayyana

Dr. Alginahi

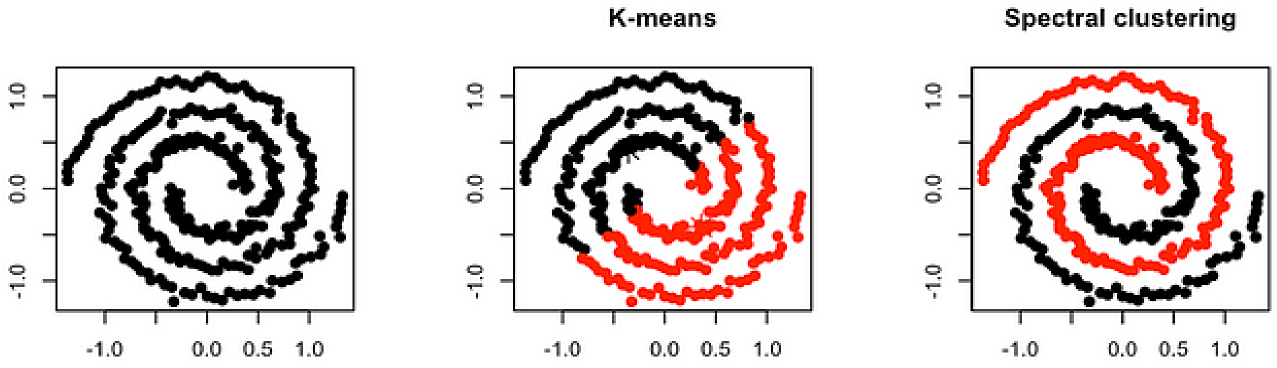
University of Windsor

Nov 10, 2023

Table of Contents

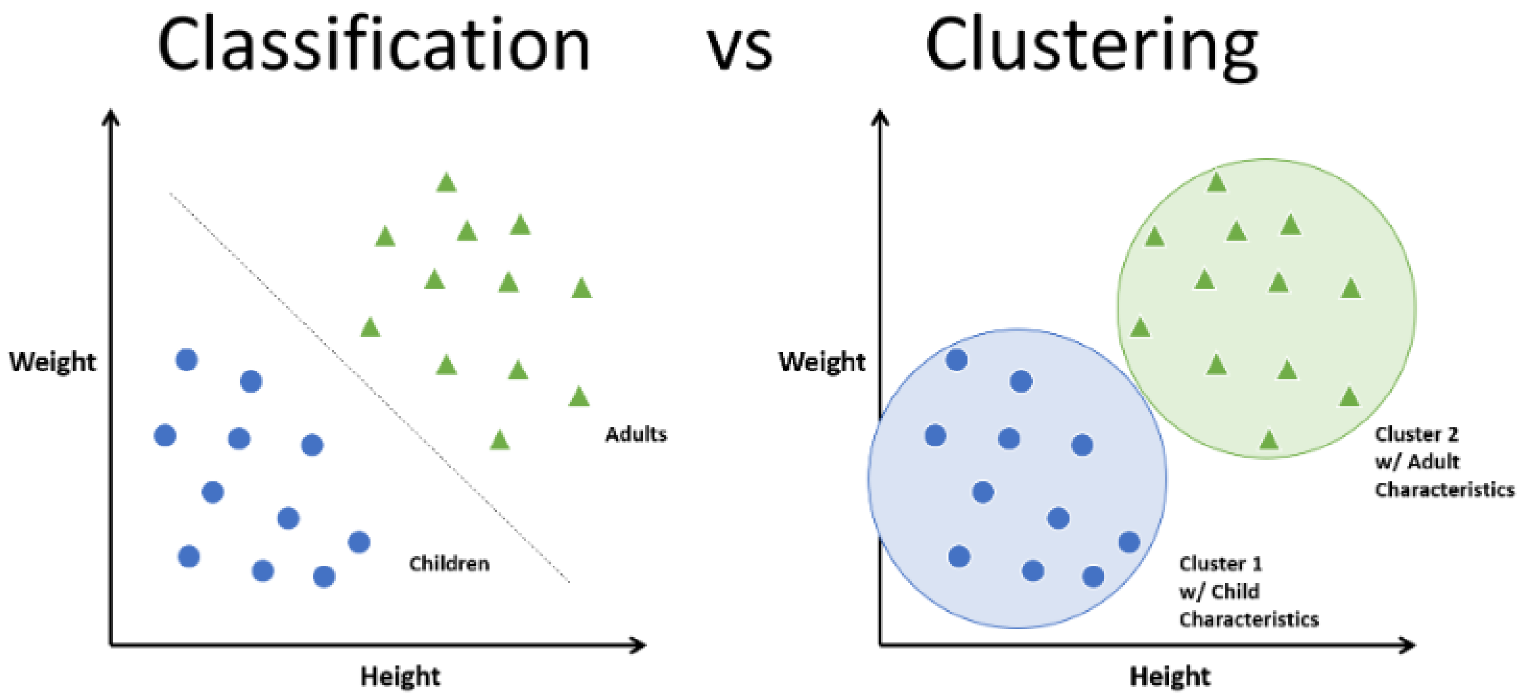
What is Spectral Clustering?

- Data clustering technique that identifies groups of similar data points by leveraging the eigenvalues and eigenvectors of a similarity graph.
- Two ways to approach clustering:
  - **Compactness**: Points that are in close proximity to one another belong to the same cluster
  - **Connectivity**: Points that are linked or directly adjacent to one another are grouped into the same cluster



Source: https://tinyurl.com/5ekmrpz2

Clustering Vs Classification



Source: https://tinyurl.com/e5se3zp3

Why Spectral Clustering?

- Handling complex cluster shapes
- Non-linear separability
- Graph-based representation
- Sensitivity to connectivity
- Community detection in networks
- Image segmentation
- Dimensionality reduction
- Unsupervised learning
- Versatility in data
- Robustness to noise and outliers

Challenges of Traditional Clustering

- Sensitivity to initial centroids
- Assumption of equal-sized cluster
- Assumption of spherical clusters
- Difficulty with non-linear relationships
- Outlier sensitivity
- Influence of noise
- Difficulty with varying clusters densities
- Fixed number of clusters
- Limited capability for graph data

Process: Affinity Matrix Construction

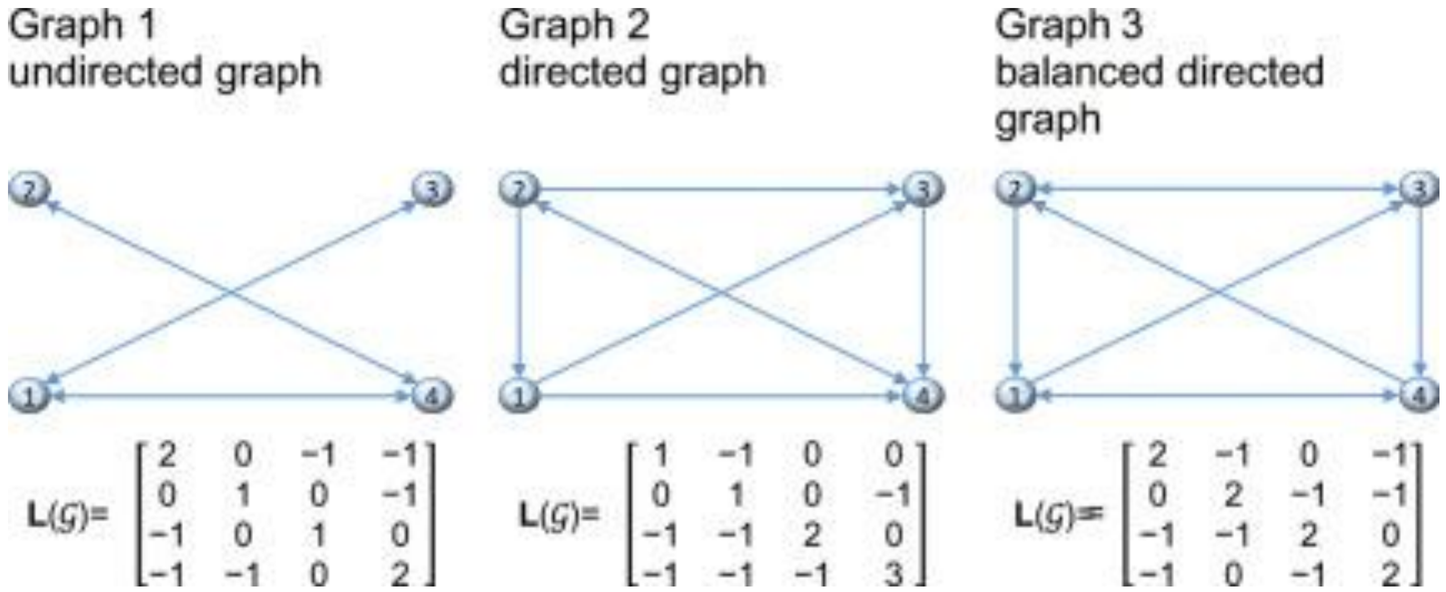
- Given a dataset with n data points, a matrix A is constructed.
- This matrix encodes the pairwise similarity between data points.
- Common similarity measures include Gaussian similarity (based on a radial basis function) $:A_{ij} = \begin{cases} \exp(-d^2(x_i, x_j)/\sigma^2) & i \neq j \\ 0 & i = j \end{cases}$
- Or k-Nearest Neighbors (where A_{ij} is non-zero if data points i and j are among the k nearest neighbors of each other).
- Or kernel

Process: Degree Matrix

- For each row in the affinity matrix, sum up the values to get the total similarity or degree of connection for each data point.
- The degree matrix D is a diagonal matrix where each diagonal element represents the degree of the corresponding node in the graph.
- If A is the affinity matrix, then the degree matrix D is formulated as $Djj = \sum j\ Aij$, where i and j represent row and column indices, respectively.

Process: Laplacian Matrix

- The Laplacian matrix is derived from a graph, where nodes represent data points, and edges represent the relationships between these points. $L = D - A$



Source: https://tinyurl.com/2t5tz7bx

Process: Eigenvalue Problems

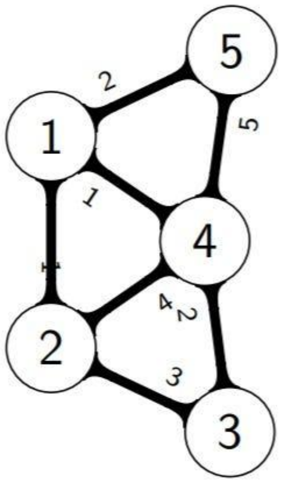
- Eigenvalues characterize the behavior of the linear transformation associated with the matrix.
- In spectral clustering, eigenvalues convey information about the underlying structure and connectivity of the data.
- Smaller eigenvalues correspond to more global patterns, while larger eigenvalues capture finer, more local patterns.

Process: Eigenvector Calculation

- Solve the $Lv = \lambda v$ where λ is the eigenvalues and v is the eigenvectors.
- The eigenvalues are typically sorted in ascending order.
- The selected eigenvectors serve as feature vectors for clustering.

$\mathbf{A}$	adjacency matrix
$\mathbf{W}$	weight matrix
$\mathbf{D}$	(diagonal) degree matrix
$\mathbf{L} = \mathbf{D} - \mathbf{W}$	graph **Laplacian** matrix

$$\mathbf{L} = \begin{pmatrix} 4 & -1 & 0 & -1 & -2 \\ -1 & 8 & -3 & -4 & 0 \\ 0 & -3 & 5 & -2 & 0 \\ -1 & -4 & -2 & 12 & -5 \\ -2 & 0 & 0 & -5 & 7 \end{pmatrix}$$



Source: https://tinyurl.com/65sbbx7p

Process: Embedding

- By using eigenvectors as features, spectral clustering operates in a reduced-dimensional space, potentially enhancing the separation of clusters.
- The computed eigenvectors serve as a new set of features for the data points.
- Each data point is represented as a vector in this new space, often referred to as the spectral embedding.

Process: Clustering in Embedded Space

- K-means can then be applied to the embedded data points to group similar data points into clusters.
- These algorithms work in lower-dimensional space defined by the spectral embedding
- It's important to specify the number of clusters (k) before applying traditional clustering techniques.
- Assign data point to the cluster with the highest probability based on the output.

Example

- https://www.youtube.com/watch?v=rVnOANM0oJE&ab_channel=Udacity

Advantages of Spectral clustering

Facial Recognition: Spectral clustering identifies facial patterns using similarity graphs and eigenvalues.

Advantages Over K-means: It excels in recognizing complex facial patterns and handling noise.

High-Dimensional Data: Spectral clustering is effective in recognizing faces in high-dimensional feature spaces.

Disadvantages to Spectral Clustering

Parameter Sensitivity: Sensitivity to parameter settings, such as the number of clusters.

High Computational Cost: Computationally intensive, particularly for large datasets.

Limited Scalability in High Dimensions: May not work well in high-dimensional spaces without dimensionality reduction

Applications of Spectral Clustering

- Face Clustering
- Expression Analysis
- Age Estimation
- Gender Classification
- Face Verification

Example

- **Scenario:** We have a dataset of images, and we need to group together similar faces
- **Mathematical steps + programming:**
  1. Preprocessing: each m by n image is transformed into a mn-dimensional feature vector
  2. Similarity Matrix Construction: Given n images, the similarity matrix W is a $n \times n$ matrix where W_{ij} represents the similarity between images i and j

Example

- **Mathematical steps + programming:**
  - Code:

```
import numpy as np

def gaussian_similarity(x_i, x_j, sigma):
    return np.exp(-np.linalg.norm(x_i - x_j)**2 / (2 * sigma**2))

def construct_similarity_matrix(X, sigma):
    n = X.shape[0]
    W = np.zeros((n, n))
    for i in range(n):
        for j in range(i, n):
            similarity = gaussian_similarity(X[i], X[j], sigma)
            W[i, j] = similarity
            W[j, i] = similarity  # Since W is symmetric
    return W
```

Example

- **Mathematical steps + programming:**

  3. Degree Matrix: Calculate the degree matrix D, which is a diagonal matrix where Dii is the degree of data point i, i.e., the sum of its edge weights

```
def construct_degree_matrix(W):
    return np.diag(np.sum(W, axis=1))
```

  4.Laplacian Matrix: Compute the Laplacian matrix L=D−W.

```
def construct_laplacian_matrix(D, W):
    return D - W
```

Example

- **Mathematical steps + programming:**

  5. Eigenvector Calculation: Compute the first k eigenvectors of the Laplacian matrix.

```
def compute_eigenvectors(L, k):
    eigenvalues, eigenvectors = np.linalg.eigh(L)
    return eigenvectors[:, :k]
```

  6. Spectral Embedding: The k eigenvectors form the spectral embedding. Each image is represented in the embedded space.

Example

- **Mathematical steps + programming:**

  7. Clustering: Apply a clustering algorithm to the embedded space. For example, use k-means clustering to group the images into k clusters.

```
from sklearn.cluster import KMeans

def cluster_embedding(embedding, k):
    kmeans = KMeans(n_clusters=k, random_state=0).fit(embedding)
    return kmeans.labels_
```

  8. Post-Processing and Interpretation: Analyze the clusters to see if they correspond to meaningful groupings of faces. For example, they might represent different individuals.

Reference

- [1] N. Doshi, “Spectral clustering,” Medium, https://towardsdatascience.com/spectral-clustering-82d3cff3d3b7 (accessed Nov. 5, 2023).
- [2] K. V, “Spectral clustering: A comprehensive guide for beginners,” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/05/what-why-and-how-of-spectral-clustering/ (accessed Nov. 7, 2023).
- [3] “Laplacian matrix,” Laplacian Matrix - an overview | ScienceDirect Topics, https://www.sciencedirect.com/topics/computer-science/laplacian-matrix (accessed Nov. 5, 2023).
- [4] A. Aoullay, “Spectral clustering for beginners,” Medium, https://towardsdatascience.com/spectral-clustering-for-beginners-d08b7d25b4d8 (accessed Nov. 7, 2023).
- [5] “Spectral clustering in machine learning,” GeeksforGeeks, https://www.geeksforgeeks.org/ml-spectral-clustering/ (accessed Nov. 7, 2023).

Questions?