Spectral Clustering

Atou K. Kougbanhoun, Dzodzoenyenye Senanou, Jeremie N. Mabiala

African Masters in Machine Intelligence
AIMS Sénégal, M'bour

April 15, 2024

Outline

Introduction

- As mentioned in [1], **spectral clustering** is a technique that make use of the spectrum (eigenvalues) of the **similarity matrix** of the data and help to perform dimensionality reduction [2] before clustering in low-dimensional spaces.
- Spectral clustering originally stems from spectral graph theory [3] where,for example, they are interested in finding partitions of a graph, called clusters/ communities.
- As for some of the graph based machine learning (ML) algorithms like DBSCAN, LLE, or Isomap [4], **spectral clustering** is also intensively used in ML (in combination with classical clustering techniques like k-means) to find clusters and learn new representation of datasets.

In this talk, we discuss the **spectral clustering** within the perspective of machine learning.

Motivation

- According [?ref1], the **K-means** method is unable to separate clusters that are non-linearly separable in the input space.
- Hence, two approached are to consider:
  1. **kernel K-means**: Before applying **k-means**, first data points are mapped to a higher-dimensional space using a non-linear function, then apply the **k-means** to cluster points in the new space.
  2. **Spectral clustering:** this approach uses the eigenvectors of a Laplacian matrix [?ref2], constructed from an affinity matrix, to obtain the clustering of the data.

Similarity/Distance matrix

- Several dataset can be transformed to a graph representation by the mean of **similarity** or **distance** functions.
- Given data points $\mathcal{D} = \{X_1, \ldots, X_n\}$ with $X_i \in \mathbb{R}^d, \forall i$, the similarity function $S : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a symmetric function that maps $X_i, X_j \in \mathcal{D}$ to a real number $S(X_i, X_j)$.
- The similarity matrix is the matrix of similarities for all pair of the points in $\mathcal{D}$.

Several choices of the function S:

1. **Gaussian kernel:**

$$S(X_i, X_j) = \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right) \tag{1}$$

where σ^2 a parameter that controls smoothness of S, that is how S becomes small.

2. **Polynomial kernel**:

$$S(X_i, X_j) = (c + X_j^T X_j)^d,\ 0 \leq c \in \mathbb{R},\ d \in \mathbb{N}. \tag{2}$$

For the distances, the prominent choice is the euclidean distance $d(X_i, X_j) = \|X_i - X_j\|_2$.

Similarity graph of the data

- We construct the similarity graph as follow:
  1. The data points become **vertices**
  2. **Two points that are "close"/ "similar" are connected by edges**.

One of these technques are proposed in the literature to construct the **the similarity graph**:

- $\epsilon-$ **neighborhood graph:** Connect all the points whose pairwise distance or similarity are less than ϵ. The resulting graph is not weighted.
- $k-$ **Nearest neighbor graph:** Connect the node v_i to the vertex v_j if v_j is in the $k-$neighborhood of the vertex v_i, The resulting graph is not always un-directed since the **neighborhood** is not symmetric.
- **Fully connected graph:** Connect the points with positive similarity with each other, and weight the all the edges by the associate similarity.

Similarity graph of the data

- Let $\mathcal{D} = \{X_1, X_2, X_3, X_4, X_5\}$ a dataset that consists of points

$$X_1 = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}, \; X_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \; X_3 = \begin{pmatrix} -1.4 \\ 7 \end{pmatrix}, X_4 = \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}, \text{ and } X_5 = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \tag{3}$$

- To construct the similarity graph, first choose a a similarity function. We use the RBF from eq. 1 with $\sigma^2 = 1$.

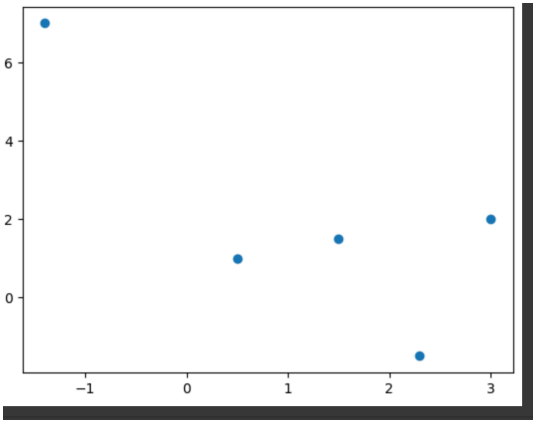


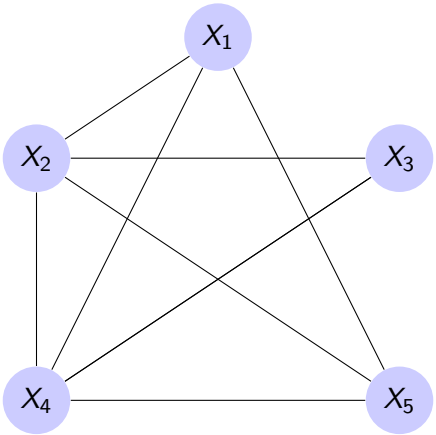
Figure 1: Scatter plot of the dataset $\mathcal{D}$.

It results the affinity matrix below. The entries values are rounded.

$$S = \begin{pmatrix} 1.00 & 0.00 & 0.54 & 0.01 & 0.04 \\ 0.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.54 & 0.00 & 1.00 & 0.01 & 0.29 \\ 0.01 & 0.00 & 0.01 & 1.00 & 0.03 \\ 0.04 & 0.00 & 0.29 & 0.03 & 1.00 \end{pmatrix} \tag{4}$$

where in rows and columns are the data points $X_1, \ldots, X_5$.

With $\epsilon = .1$, the $\epsilon-$ neighborhood graph is:



whose adjacency matrix is

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \tag{5}$$

Graph Laplacian

- The **graph Laplacian matrix** L is defined as

$$L := D - A \tag{6}$$

where A is a the adjacency matrix and D the degree matrix.

- The **degree matrix** for an un-directed and unweighted graph is a diagonal matrix whose elements are given by

$$D_{ii} = \sum_{j}^{n} A_{ij}, \ \forall i = 1, \ldots, n. \tag{7}$$

- n is the number of vertices of the graph,

- For a weighted graph, the graph Laplacian is defined as

$$L := D - W \tag{8}$$

where W is the weight matrix and $D = diag(d_1, \ldots, d_n)$ and

$$d_i = \sum_{j=1} W_{ij}, \ \forall i. \tag{9}$$

- Eqs. (6) and (8) define the **un-normalized graph Laplacian**.
- According to [4], the **normalized graph Laplacian** is defined as

$$L_{rw} := D^{-1}L = I - D^{-1}W \tag{10}$$

- Replace W by the adjacency matrix for un-weighted graphs.

- For the sake of completeness, consider the similarity graph in fig. 2.
- Its Laplacian matrix is given as

$$L := D - A = \begin{pmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & -1 & -1 & 4 & 1 \\ -1 & -1 & 0 & -1 & 3 \end{pmatrix}, \text{ where } D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix} \tag{11}$$

and the matrix A was given in the equation (5).

Spectral Clustering Algorithm [5].

Non-normalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

1. Construct a similarity graph by one of the ways described above. Let W be its weighted adjacency matrix.
2. Compute the non-normalized Laplacian L.
3. Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.
4. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
5. For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
6. Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

Spectral Clustering Algorithm, ctd.

Normalized spectral clustering Shi-Malik method:

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

1. Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
2. Compute the non-normalized Laplacian L.
3. Compute the first k generalized eigenvectors $u_1, \ldots, u_k$ of the generalized eigen problem $Lu = \lambda Du$.
4. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
5. For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
6. Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

Illustrration

- For the un-normalized case, the eigenelements of L are:

$$\lambda_1 = 0 \quad e_1 = (1, \ldots, 1)^{'} \tag{12}$$

$$\lambda_2 = 2 \quad e_2 = (1, 0, -2, 0, 1)^{'}, \tag{13}$$

$$\lambda_3 = 4 \quad e_3 = (1, 0, 0, -0, -1)^{'}, \tag{14}$$

$$\lambda_4 = 5 \quad e_4 = (1, 0, 1, -3, 1)^{'}, \; e_5 = (0, 1, 0, -1, 0)^{'} \tag{15}$$

- $k = 2$. We consider the two smallest eigenvalues , beside the zero eigenvalue.

$$U = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ -2 & 0 \\ 0 & 0 \\ 1 & -1 \end{pmatrix}_{5 \times 2} \tag{16}$$

- It results the **embedded data points**:

$$y_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; y_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \; y_3 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \; y_4 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \; y_5 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad (17)$$

- The transformed data points. Then we can proceed with the $k-$ means for the new dataset.

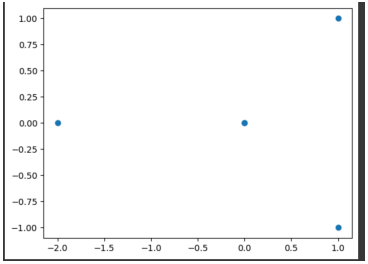


Figure 3: Scatter plots of the embedded data points.

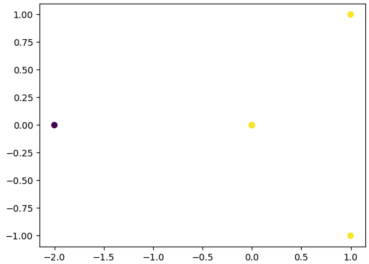


Figure 4: Two clusters obtained after $k-$means, with $k = 2$

The result with $k-$ means with $k = 2$.

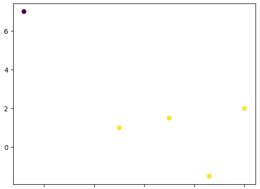


Figure 5: $k-$means, with $k = 2$, applied to the original dataset.

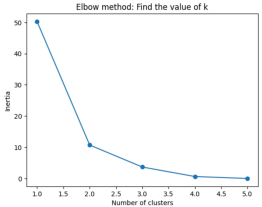


Figure 6: The Elbow method to find the number of clusters.

Implementation

- Refer to `https://colab.research.google.com/drive/1Uq2WrbHBcATPD6Hriemkj-ZQQvEbn5Fb?usp=sharing`

Conclusion

- Spectral clustering is a powerful machine learning method for clustering data based on similarity. Its advantages include handling non-linear data, accommodating complex cluster shapes, and robustness to noise.
- However, it requires careful parameter tuning and can be computationally intensive. Future research aims to improve scalability, automate parameter selection, enhance robustness, and integrate domain knowledge for better applicability.

Reference

[1] Wikipedia, *Spectral clustering*. https://en.wikipedia.org/wiki/Spectral_clustering.

[2] _____, *Dimensionality reduction*. https://en.wikipedia.org/wiki/Dimensionality_reduction.

[3] Daniel A. Spielman, *Spectral Graph Theory and its Applications*. http://cs-www.cs.yale.edu/homes/spielman/sgta/SpectTut.pdf.

[4] Matthias Hein and Ulrike von Luxburg, *Similarity Graphs in Machine Learning*. https://www.ml.uni-saarland.de/code/GraphDemo/HeinLuxburg_SlidesSimilarityGraphs.pdf.

[5] Ulrike von Luxburg, *A Tutorial on Spectral Clustering*. https://arxiv.org/pdf/0711.0189.pdf.