SPECTRAL CLUSTERING: A PYTHON IMPLEMENTATION

PRIYA VARRA AND SHREYA VARRA
SECTION H

1. Introduction

One of the core objectives of computer and data science is extracting meaningful insights from data. This is accomplished by utilizing algorithms and programming to reveal patterns that are present in data sets. One common type of data that we might wish to determine the inherent structure of is data with points that have not been explicitly classified with labels. In this case, clustering, a technique which divides data points into groups called clusters with the criteria that points in the same cluster are similar and points in different clusters are dissimilar to each other, serves as a useful tool for categorizing the data. In fact, there are many interesting and practical real world examples of the application of clustering in analyzing unclassified data. For instance, in bioinformatics, clustering algorithms have successfully been utilized to group similar DNA sequences into gene families. In medicine, clustering has been applied to differentiate between different types of tissues that may be present in three-dimensional images produced by PET scans. In addition, major social media platforms such as Facebook and Twitter rely on clustering to determine circles of friends and communities within their population of users (Wikipedia 2018).

Clustering by itself is not a single algorithm but rather it is a broad term that encompasses the many different algorithms that can classify groups within a set of data. The existence of multiple clustering algorithms is due to the fact that the concept of a cluster is not bound to one single definition. In general, a cluster should satisfy two properties-

- Intra-similarity: All points within the same cluster should be very similar to each other
- Inter-dissimilarity: Any points that are in different clusters should be very dissimilar to each other

However, the metric for similarity is variable and depends on the cluster model and algorithm that is chosen for a data set by the person implementing clustering.

In this paper, we will first discuss a very popular clustering algorithm called k-means clustering, which falls under the centroid-based cluster model, before delving into the spectral clustering algorithm, which falls under the graph theoretic cluster model. (Zagrofos, 2007)

2. Definitions/Notation

Before we discuss the clustering algorithms, we want to introduce some basic definitions and notation that we will utilize in our discussion.

Euclidean Norm - The length of an n-vector in $\mathbb{R}^n$ .If $\bar{x} = [x_1, x_2, ..., x_n] \in \mathbb{R}^n$ then the euclidean norm of $\bar{x}$, denoted $\|\bar{x}\|$, is defined as $\|\bar{x}\| = \sqrt{\bar{x} \cdot \bar{x}} = (\sum_{i=1}^{n} = |x_i^2|)^{\frac{1}{2}}$

Euclidean Distance - The distance between two points in an n-dimensional euclidean space ($\mathbb{R}^n$). For our implementations of k-means clustering and spectral clustering, our metric for similarity between two data points will be the euclidean distance between them. We will calculate the euclidean distance between two data points $\bar{x}$ and $\bar{y}$ in $\mathbb{R}^n$ by computing the standard euclidean norm of the resultant vector $\bar{x} - \bar{y}$.

Undirected Similarity Graph - For a set of data with an associated metric for similarity between each pair of distinct data points, an undirected similarity graph is a mathematical structure that models the pairwise relationships between the data points in the set. We represent a graph G as an ordered pair (V,E) where V is the set of vertices that represents the data points and E is the set of bidirectional edges between distinct vertices with weights given by the similarity between the data points that the vertices represent.

Weighted Adjacency Matrix - For an undirected similarity graph with vertex set V, its corresponding weighted adjacency matrix is a square $|V| \times |V|$ matrix W such that W_{ij} is the value of a chosen similarity metric between vertex i and vertex j. Because the graph is undirected, W must have the property that $W_{ij} = W_{ji}$.

Degree Matrix - The degree matrix D of a matrix W is defined as having diagonals d_i such that $d_i = \sum_{j=1}^{n} W_{ij}$, or in other words, the ith diagonal element in D is the sum of the all of the elements of the ith row in W.

Unnormalized Graph Laplacian Matrix - For a graph G, the unnormalized graph laplacian matrix L is defined as $L = D - W$ where W is the weighted adjacency matrix of G and D is the corresponding degree matrix of W. Since D and W are symmetric, it follows that L is symmetric, and as a consequence, L has real and non-negative eigenvalues.

3. K-Means Clustering

We will now introduce k-means clustering since we have chosen to utilize the algorithm in our implementation of spectral clustering. K-means clustering is a great algorithm for finding clusters in data that has convex, or spherical, boundaries between clusters. Our implementation of the algorithm follows the procedure outlined below (NK, 2017):

(1) Choose the number of clusters, k, that we wish to find in our data.
(2) Initialize k clusters by picking k random data points from the data set to constitute the initial centroids.
(3) Calculate the centroids, u_i, for each of the k clusters. This is accomplished by computing the average of all of the data points in each cluster.
(4) For each x_j in the data, calculate the euclidean distance from x_j to u_i by computing $\|x_j - u_i\|$
(5) Assign each x_j to the cluster with the corresponding centroid, u_i, that is nearest, or more precisely, the centroid from which x_j has the smallest euclidean distance.
(6) Repeat steps 3-5 until the centroids converge or no longer change after an iteration of the algorithm.

One limitation of this algorithm is that its results can vary depending on the initialization of the centroids. In our implementation, to account for this issue we chose to repeat the entire k-means algorithm 10 times with a different random initialization of the k centroids and kept the attempt with the smallest sum of the squared euclidean distances between all of the points and the centroid of their assigned cluster.

Another drawback of the k-means clustering algorithm, as seen in Figure 1, is that it typically fails on data with non-convex boundaries. This is because the algorithm relies on the assumption that the clusters are spherical and separable in a way such that each mean converges towards the center of each cluster (Singh, 2010).

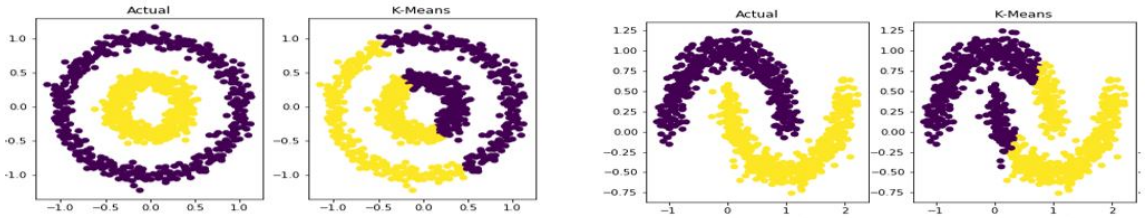

Figure 1: (Left) Result of sklearn.cluster.Kmeans on dataset created from sklearn.datasets.make_circles with k = 2. (Right) Result of sklearn.cluster.Kmeans on dataset created from sklearn.dataset.make_moons with k = 2

As we begin to discuss spectral clustering, we will see that the algorithm becomes particularly useful in this case because it allows us to project clusters with non-convex boundaries onto k-dimensional space from which k-means clustering can effectively cluster the data.

4. Spectral Clustering

The first step in implementing spectral clustering is creating a representation of the relationships between the data points. This is achieved through an undirected similarity graph with the data points from the data set constituting the vertices of the graph. In our implementation, we model this graph by creating a weighted adjacency matrix W according to the k-nearest neighbors algorithm. The overall idea of the k-nearest neighbors algorithm is that for a vertex v_i, we connect it to another vertex

v_j if v_j is one of the k vertices that are closest in euclidean distance to v_i. (Luxburg, 2007) The value of k in the k-nearest neighbors algorithm is not necessarily the same as the number of clusters we are trying to produce. There is no specific method for determining k, so we chose to use $k = 10$ or, if the data set is exceptionally small, the square root of the size of our data. Our steps for creating the weighted adjacency matrix with the k-nearest neighbors algorithm are outlined below:

(1) Calculate the pairwise euclidean distances of every v_i and v_j in our graph and place them in a matrix A such that A_{ij} is the pairwise distance between v_i and v_j
(2) Sort each row in the matrix such that the index corresponding to the closest v_j to v_i will be first and the the index for the farthest v_j from v_i will be last.
(3) For each vertex take the indices corresponding to the k-nearest neighbors by selecting the first k indices of each row of the matrix (not including the first index which represents the pairwise distance between v_i and itself after sorting the matrix).
(4) Create W by letting $W_{ij} = 1$ if j is one of the k indices selected for the ith row (this means that we set $W_{ij} = 1$ if v_j is within the k-nearest neighbors of v_i) and letting $W_{ij} = 0$ if j is not one of the k indices selected for the ith row (meaning we set $W_{ij} = 0$ if v_j is not within the k-nearest neighbors of v_i).

However, this procedure by itself leads to a directed graph because v_i being in the nearest neighbors of v_j does not necessarily mean that v_j is in the nearest neighbors of v_i. To make this graph undirected and, consequently, to make the weighted adjacency matrix symmetric, we ignore the directions of the edges by connecting v_i and v_j with an undirected edge if either v_i is among the k-nearest neighbors of v_j or if v_j is among the k-nearest neighbors of v_i. (Luxburg, 2007)

The next step is creating the degree matrix D of the weighted adjacency matrix W followed by constructing the unnormalized graph laplacian matrix L. As seen in Figure 2, we begin to see how we can derive which points are most similar to each other from our graph laplacian matrix L.

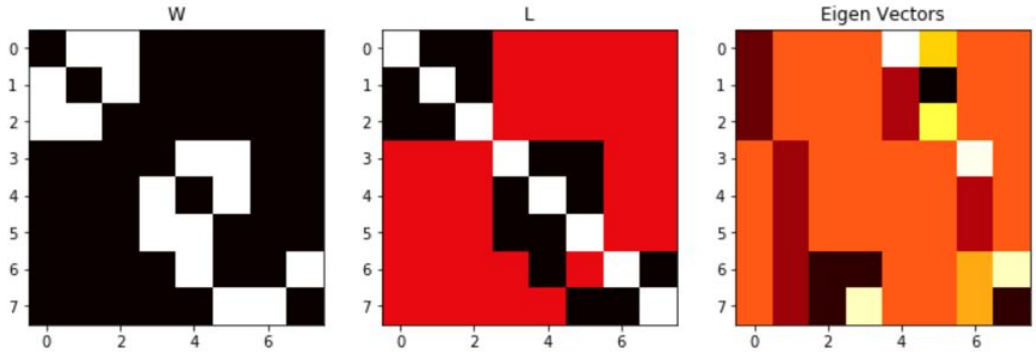

Figure 2: Here, the adjacency matrix W, the graph Laplacian, and the eigenvectors (sorted from smallest eigenvalue to largest eigenvalue) of a data set with 3 clusters have been graphed such that if W_{ij} is closer to 0, the color is darker and if W_{ij} is closer to 1, the color is lighter. Observe the diagonal block structure of the graph Laplacian and how the first 3 eigenvectors show the separation of the clusters.

The vectors that are most similar in our graph Laplacian correspond to the first k eigenvectors, and as such, the k smallest eigenvalues. These eigenvectors represent how the data points are divided based on how similar they are to each other. Thus, when we take the eigenvectors corresponding to the k smallest eigenvalues, we are able to split the data points according to their similarity into k-dimensional space with clearly defined convex borders as exemplified in Figure 3.

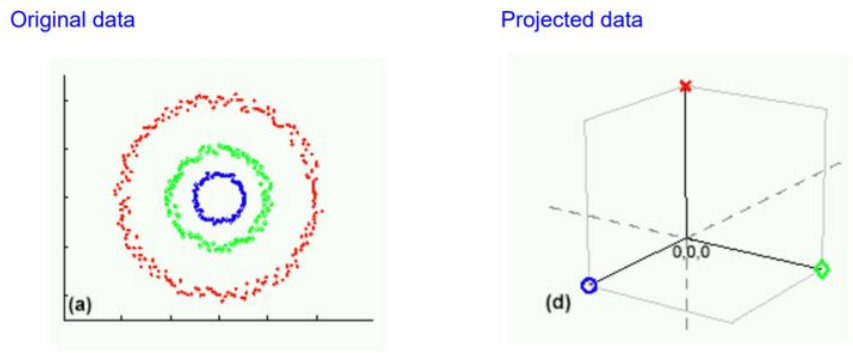

Figure 3: A set of data before and after projection using the eigenvectors corresponding to the 3 smallest eigenvalues of its associated graph Laplacian matrix (Singh, 2010)

Now we will be able to successfully run k-means clustering on the projected data which we achieve by creating a projection matrix with columns consisting of the first k eigenvectors (Meila, 2016). Each row of this matrix is essentially the new location in space of each data point, which means we can cluster the data points by running the k-means clustering algorithm on the rows of the projection matrix. This final step produces the clusters that we return from our implementation of spectral clustering.

5. Conclusion

Our implementation of spectral clustering is one version of many possible options for the algorithm. For instance, in addition to the k nearest neighbor algorithm, popular choices of similarity metrics for creating the adjacency graph include the Gaussian Similarity function and the epsilon nearest neighborhood algorithm. We chose to utilize an unnormalized graph Laplacian matrix but some alternatives also include the normalized graph Laplacian and the symmetric graph Laplacian. In our implementation, we chose to use k-means clustering to cluster the rows of our projection matrix, however, other techniques include advanced post-processing of the eigenvectors and hyperplanes (Luxburg, 2007). Ultimately, there is no specific set of guidelines on choosing from the various options and, instead, the choices are left to the discretion of the person implementing spectral clustering. Across all of these alternatives, however, the underlying concept remains the same of representing the relationships between the data points through a similarity graph, creating a graph Laplacian matrix, projecting the data onto a k-dimensional space according to the eigenvectors of the Laplacian, and finally clustering the projected data into the k clusters.

References

[1] Meila, M. (2016). Spectral Clustering: a Tutorial for the 2010âs. Retrieved December 6, 2018

[2] NK, M. (2017, October 1). K-Means Clustering in Python. Retrieved December 10, 2018, from `https://mubaris.com/posts/kmeans-clustering`

[3] Singh A., (2010). *Spectral Clustering*[PDF document]. Retrieved December 7, 2018 from `https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf`

[4] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4), 395-416. Retrieved December 6, 2018

[5] Wikipedia contributors. (2018, December 6). Cluster analysis. In *Wikipedia, The Free Encyclopedia*. Retrieved December 7, 2018, from `https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=872325062`

[6] Zografos V., Nordberg K. (2011) Introduction to Spectral Clustering[PDF document]. Retrieved December 6, 2018, from `http://www.cvl.isy.liu.se:82/education/graduate/spectral-clustering/SC_course_part1.pdf`