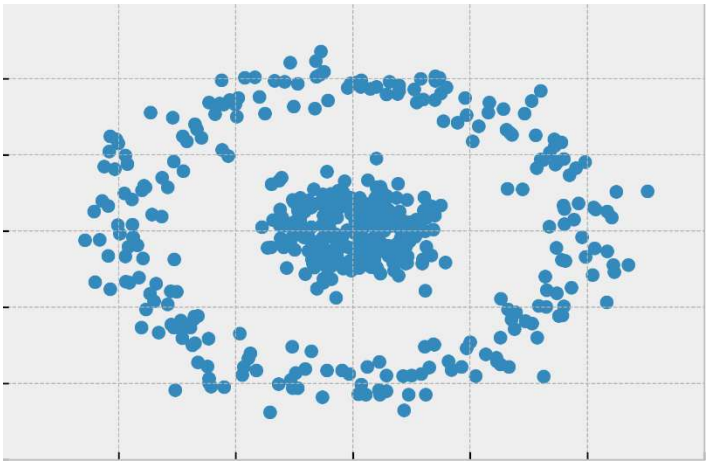
Spectral Clustering



Spectral clustering is used to dentify communities of nodes in a graph based on the edges connecting them.

Algorithm Process (min cut):

1. Represent the graph as an adjacency matrix A $\in R^{m\times m}$ for m data points with binary digits for unweighted graph and binary digits mutipling by weights for weighted graph, where the row and column indices represent the nodes, 1s represent that there are edge connections between the nodes and 0s represent the opposite.
2. Compute vertex degree $d_i = \sum A_{ij}$. For unweighted graph, the vertex degree represents the number of edge connections to each node. Let degree matrix D $\in R^{m\times m}$ denote a diagonal matrix of vertex degree d_i for i = 1,2,…,m.
3. Compute a Laplician matrix L $\in R^{m\times m}$ = D – A.
4. After Eigendecomposition, find eigenvectors of Laplician matrix L corresponding to the smallest eigenvalues.
5. Perform Kmeans on the eigenvectors by treating each row as a data point to determine the assignments.

The above step 3 and 4 can also be (normalized cut):

3. Compute a new matrix B = $D^{-1/2}AD^{-1/2}$
4. Find the eigenvectors of the matrix B corresponding to the largest eigenvalues.

Comment:

- Laplician matrix L is positive semi-definite (meaning all eigenvalues are nonnegative) and it has at least one zero eigenvalue. This makes the algorithm more functional to perform eigendecomposition on Laplician matrix L rather than adjacency matrix A.
- The eigenvectors corresponding to small eigenvalues contain cluster assignment information.
- Represent each row of k eigenvectors as new data points for clustering.
- The number of clusters k is determined by the number of smallest eigenvalues before a large gap between the last selectes eigenvalues and the next smaller eigenvalue.

Reference:

https://towardsdatascience.com/spectral-clustering-aba2640c0d5b