University of Aberdeen
Department of Mathematics

Author	Rufus Behr, 52091262
Date	9.5.2021

MA2508 Linear Algebra
The Unreasonable Effectiveness of Spectral Clustering

Living in an increasingly data-driven world, it is imperative to have an understanding of how data is being used. One common way to represent data is through a structure called a Graph. A Graph is comprised of vertices, commonly known as points or nodes, and edges, the lines that connect the vertices, which makes this structure ideal for modelling relationships in data. The quintessential use-case example of a Graph is to model the relationship between friends and friend groups, where each vertex is a person and if an edge exists between two vertices they are friends. There can also be numerical values associated with the vertices and edges. Every vertex has a degree, which is the number of edges that are connected to it, and both vertices and edges can have a weight, which will typically indicate something about the relationship (often strength). One way of extrapolating meaning from data is through clustering. The two most utilised clustering methods are K-Means Clustering and Spectral Clustering. K-Means Clustering attempts to identify (k) cluster centers, also known as centroids, and then the proximity to these centers determines to which cluster a point belongs, whereas Spectral Clustering accounts more for the underlying geometry [7] through use of the Laplacian Matrix and its eigenvalues & eigenvectors.

But what is the Laplacian Matrix? It is easiest to understand the Laplacian Matrix from the context of a simple unweighted & undirected Graph, and it is defined as the difference between the diagonal Degree Matrix and the Adjacency Matrix of this Graph (L = D - A[1]). The diagonal Degree Matrix is a Matrix, whose diagonal entries ($D_{i,i}$) are the number of edges connected to the vertex i and all other entries in the Matrix are equal to 0, whereas the Adjacency Matrix is a Matrix that at row i and column j has a value of 1 if there exists an edge between vertex i and vertex j (otherwise the value of $A_{i,j}$ is 0)[6]. Consequently, with the Laplacian Matrix being the difference of these two, each row of the Laplacian Matrix will sum to 0 (since the degree of a node is equivalent to the number of edges). Whilst this is somewhat interesting in and of itself, the crux of Spectral Clustering and the importance of using the Laplacian Matrix rests on its eigenvalues and eigenvectors[2] (where there exists an ordering such that $\lambda_0 \leq \lambda_1 .... \leq \lambda_n$). It is actually on account of each row summing to 0 that there exists a trivial eigenvalue and eigenvector: $\lambda_0 = 0$ and $v = < 1, 1, ...1 >$[5, 6]. Moreover, the algebraic multiplicity of $\lambda = 0$, that is the number of times 0 is a root of the characteristic polynomial, is equivalent to the number of connected components[6]. In Godsil's Algebraic Graph Theory, he states "Let X be a graph with n vertices and c connected components. If Q is the Laplacian of X, then rk Q = n - c"[4], which bears some resemblance to the Rank-Nullity Theorem (i.e. the Rank + the Nullity = n, n number of columns). If one rewrote it to explicitly resemble the Rank-Nullity Theorem, the relation becomes even clearer : rk Q + c = n, so c, the number of connected components, directly equals the Nullity (the dimension of the Kernel) and, as "the 0-eigenspace of L is exactly the kernel of L" [1], it is then clear why the algebraic multiplicity of $\lambda = 0$ for the Laplacian Matrix is equivalent to the number of connected components. Intuitively, then, the trivial $\lambda_0 = 0$ makes sense, as there must exist at least one connected component, and, if the Graph is connected, then $\lambda_1 > 0$ and the

[1] In the case of a Weighted Graph, this is will be written as L = D - W

[2] In fact the term Spectral, in Spectral Clustering, references the Spectrum (i.e. the set of eigenvalues) of the the Laplacian Matrix.

larger λ_1 is the greater the algebraic connectivity is of the Graph[6, 5, 3]. But, if there are m connected components, then the ordering of eigenvalues will be $\lambda_0 = \lambda_1 = ... = \lambda_m < \lambda_{m+1}$, and this λ_{m+1}, the first non-zero eigenvalue, also plays a role, and it is known as the spectral gap[6]. The spectral gap is useful in determining how easy it is to separate the graph, and this is seen from the Cheeger Inequality. The Cheeger constant, h_G, is a number showing how efficient of a separation exists within a graph where efficiency is expressed as the ratio of the number of edges that must be cut divided by the minimum of the remaining distinct parts of the Graph (and the minimum of this ratio of all the subsets of the Graph is the Cheeger constant for the Graph), which incentives larger distinct parts to remain[9, 2]. The Cheeger Inequality then directly relates to the spectral gap of the Laplacian Matrix as follows $\frac{h_G^2}{2} \leq \lambda_{m+1} \leq 2h_G$, and, as the aim of clustering is to distinguish parts of the Graph, the Cheeger inequality provides bounds on how easily this can be done[9, 2].

Having established what the Laplacian Matrix of a Graph is and the implications of its eigenvalues & eigenvectors, the Spectral Clustering algorithm can now be understood with ease. The first step of the algorithm is done by computing the Laplacian of the Graph, as seen above[3][6]. In order to determine k clusters, one then calculates the eigenvectors, ordered such that $0 = \lambda_0 \leq \lambda_1 \leq ...\lambda_n$, and the first k are selected to be formed into a new Matrix where each column is one of the k eigenvectors of length n, each row now represents a vertex[6]. This effectively acts as dimension reduction for the data from $\mathbb{R}^n \to \mathbb{R}^k$[7]; this Matrix is a Laplacian Eigenmap[7, 10]. Lastly, this new Matrix is provided as input for a clustering algorithm on the rows of the Matrix, such as K-Means clustering, which then results with the desired Spectral Clustering[7, 3, 11, 8].

But, why would one go through the additional steps if the end result still relies on K-Means Clustering? This is a valid critique of Spectral Clustering. The issue with K-Means Clustering is that the initial centroids (the cluster centers) are randomly initialised and then updated, so there is still a level of randomness and consequently varying degrees of success in its clustering [3]. That being said, through use of the nonlinear Laplacian Eigenmap, Spectral Clustering, unlike K-Means Clustering by itself, can "capute 'the geometry of data' and the local structure"[7, pg. 3]. Its main other critique is that, since it requires computing eigenvalues and eigenvectors, for large datasets it can be very slow [3].

Ultimately, as the world becomes further data-driven, Spectral Clustering should be in the arsenal of any competent Data Scientist for its ease of implementation and for the potential advantages over other prevailing clustering algorithms dependent on the data at hand.

[3]for k clusters, however, the Laplacian matrix must be normalised, so instead of L = D - A, as see above, it would be L = I-D$^{-\frac{1}{2}}$·A·D$^{-\frac{1}{2}}$[6].

Bibliography

[1] *23. Kernel, Rank, Range*. URL: https://www.math.ucdavis.edu/~linear/old/notes23.pdf.

[2] *Cheeger constant (graph theory)*. Mar. 2021. URL: https://en.wikipedia.org/wiki/Cheeger_constant_(graph_theory).

[3] Neerja Doshi. *Spectral clustering*. May 2019. URL: https://towardsdatascience.com/spectral-clustering-82d3cff3d3b7.

[4] Christopher D. Godsil and Gordon F. Royle. *Algebraic graph theory*. Springer, 2010.

[5] University of Illinois at Urbana-Champaign. *Spectral Clustering 01 - Spectral Clustering*. Aug. 2015. URL: https://www.youtube.com/watch?v=zkgm0i77jQ8.

[6] *Laplacian matrix*. May 2021. URL: https://en.wikipedia.org/wiki/Laplacian_matrix.

[7] Shuyang Ling. *NYU MATH-SHU 236 Spectral Clustering, Graph Laplacian*. Mar. 2020. URL: https://cims.nyu.edu/~sling/MATH-SHU-236-2020-SPRING/MATH-SHU-236-Lecture-7-8-spectral.pdf.

[8] Cory Maklin. *Spectral Clustering Algorithm Implemented From Scratch*. July 2019. URL: https://towardsdatascience.com/unsupervised-machine-learning-spectral-clustering-algorithm-implemented-from-scratch-in-python-205c87271045.

[9] *MATH 595 - 1March 2017 Cheeger's constant*. Mar. 2017. URL: https://www.youtube.com/watch?v=t2J-dNzzGq8.

[10] *Nonlinear dimensionality reduction - Laplacian Eigenmaps*. Apr. 2021. URL: https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction#Laplacian_eigenmaps.

[11] *Spectral clustering*. Apr. 2021. URL: https://en.wikipedia.org/wiki/Spectral_clustering.