GRAPH SPECTRAL CLUSTERING

SUMMER @ ICERM 2020

1. Clustering

The grouping of a collection of objects into similar "cliques" or "clusters" is a central goal in data analysis. More precisely, in applications one typically has a very large number of data points, $X = \{x_1, \ldots, x_n\}$ for $n \gg 1$, where each data point can be represented as a point in some Euclidean space $\mathbb{R}^d$ (although more general representations are possible). One goal is the partitioning of these elements into *clusters*, i.e., the identification of k subsets $\{C_j\}_{j=1}^k$ of X such that

$$X = \cup_{j=1}^k C_j, \qquad\qquad C_i \cap C_j = \emptyset,$$

for $i, j = 1, \ldots, k$. If we view the data X as some cloud of points in $\mathbb{R}^d$, then geometrically one could interpret the clustering procedure as seeking to identify k clusters of points that are geometrically distributed in $\mathbb{R}^d$. Clustering naturally requires some notion of "distance" or "similarity" between points. For example, we might have some way to characterize the "similarity" between two points in the set:

$$w_{ij} = \exp(-\mathrm{dist}(x_i, x_j)^2),$$

where dist computes some notion of "distance" between two points. (Points that are small "distance" from each other should be very "similar".) If $\mathrm{dist}(\cdot, \cdot)$ is, say the Euclidean norm on $\mathbb{R}^d$, then the set of weights $\{w_{ij}\}_{i,j=1}^n$ are functions of the collection of pairwise (Euclidean) distances between points, and the clustering problem is straightforward to interpret: we are looking for clusters of points such that the pairwise distance between two points in a cluster is "small". In many practical applications, some non-trivial notions of distance are required to properly interpret data; i.e., Euclidean distance may be misleading, see Figure 1.

2. Spectral clustering

Given the pairwise distances $\{w_{i,j}\}$, one algorithm that can be used to detect and construct clusters is the k-means algorithm. The challenge is that the detection of clusters in high-dimensional space ($d \gg 1$) is both geometrically and computationally challenging. One tool to combat this issue is to preprocess the data via graph-spectral dimension reduction. Here is one basic approach, typically called "unnormalized spectral clustering".

2.1. **The graph Laplacian.**

The first step is to construct the *graph Laplacian.* By considering each individual point x_j as the vertex of a graph, undirected weighted edges are constructed via the pairwise distances. I.e., two vertices x_j and x_k are connected by an edge in the graph of weight w_{ij}. We can represent these edge connections via the weight matrix $W \in \mathbb{R}^{n,n}$, with entries $(W)_{i,j} = w_{i,j}$.

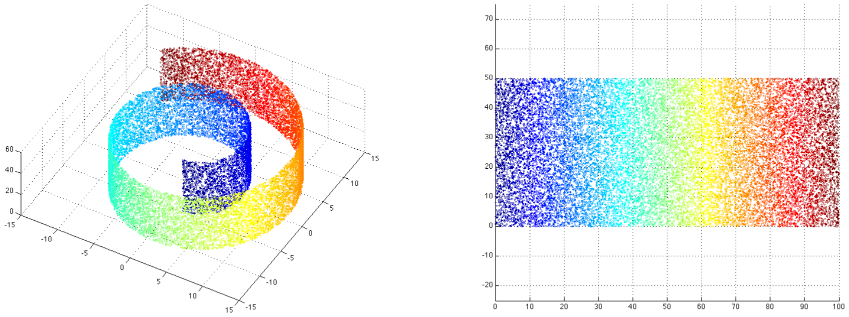

Figure 1. Plot from [1]. The “swiss roll” data set (left). Points in $\mathbb{R}^3$ are distributed according on a two-dimensional submanifold, and from the application point of view, we want points to be “close together” if they are close together with respect to the Euclidean distance on the “unrolled” manifold in $\mathbb{R}^2$ (right). The Euclidean distance on the data set in $\mathbb{R}^3$ is misleading since points points that are colored dark red and light blue look close in the Euclidean distance (left), but the manifold distance of these (right) is large.

The *degree* d_j of a vertex x_j is the sum of the weights connecting it to all the other vertices:

$$d_j \coloneqq \sum_{i=1}^{n} w_{ij}.$$

The degree matrix D is then a diagonal matrix containing the degrees:

$$D = \operatorname{diag}(d_1, \dots, d_n).$$

Finally, the graph Laplacian is defined as the symmetric positive-definite matrix,

$$L \coloneqq D - W.$$

2.2. **Spectral decomposition of the graph Laplacian.** The second step in dimension reduction via graph spectral methods is to replace each high-dimensional point $x_j \in \mathbb{R}^d$ with a low-dimensional point $y_j \in \mathbb{R}^k$ with $m \ll d$. This is accomplished from the eigenvalue decomposition of L:

$$L = V \Lambda V^T,$$

where Λ is a diagonal matrix with entries $\{\lambda_j\}_{j=1}^n$ that are ordered such that $\lambda_j \leq \lambda_{j+1}$. The jth column v_j of the matrix $V \in \mathbb{R}^{n\times n}$ is the eigenvector corresponding to λ_j.

The dimension reduction is accomplished by taking only the first $m < n$ eigenvectors, and truncating the rest. In particular if we form the column-truncated eigenvector matrix, this defines our new points y_j:

$$V_m \coloneqq \underbrace{\begin{pmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{pmatrix}}_{n\times m} = \underbrace{\begin{pmatrix} — & y_1 & — \\ — & y_2 & — \\ & \vdots & \\ — & y_n & — \end{pmatrix}}_{n\times m}$$

2.3. **Clustering on the set Y.** The set Y now contains n points in $\mathbb{R}^m$, but now $m \ll d$. We still identify $y_j \in \mathbb{R}^m$ with $x_j \in \mathbb{R}^d$, but now all the data lies in a much lower dimensional space. The advantage of this is that standard clustering algorithms like k-means algorithms (in general, algorithms that detect geometrical representations of data) are far more effective at identifying geometrical patterns in lower dimensions. Thus, clustering on Y is (frequently much) more effective than clustering on X. The combination of all the previous steps is a graph spectral clustering algorithm.

Graph spectral clustering is flexible since it relies on low-dimensional geometry that is learned only from pairwise distances, and has been successfully used in several real-world applications such as image segmentation [3].

3. Project outlook

This project involves the following investigations:

- analytical investigation of graph and graph clustering properties, for example as in [2]
- numerical implementation of graph clustering algorithms
- comparison of graph clustering methods on real-world data sets
- development of clustering algorithms that are customized for particular applications
- graph sparsification for acceleration of large data sets

Students will program in either Matlab or Python to focus the numerical investigation. A good starting point for learning about spectral clustering is the reference [4].

References

1. John Burgoyne and Stephen Mcadams, *Non-linear scaling techniques for uncovering the perceptual dimensions of timbre*, January 2007.
2. Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, *On spectral clustering: analysis and an algorithm*, Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (Vancouver, British Columbia, Canada), NIPS'01, MIT Press, January 2001, pp. 849–856.
3. Jianbo Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000), no. 8, 888–905, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
4. Ulrike von Luxburg, *A tutorial on spectral clustering*, Statistics and Computing **17** (2007), no. 4, 395–416 (en), arXiv: 0711.0189 [cs.DS].