Community detection and non-linear dimension reduction techniques in data science

Hrishikesh Bodas†, Annika Cleveland‡ and Maati McKinney*

Georgia Institute of Technology Math REU, Summer 2019

†Carnegie Mellon University, *hbodas@andrew.cmu.edu* ‡New Mexico State University, *annikac@nmsu.edu* *Spelman College, *mmckinn6@scmail.spelman.edu*

Introduction

We studied two different approaches to interpreting high dimensional data sets: community detection and non-linear dimension reduction. Under community detection, we reviewed spectral clustering as presented in [3]. We also examined the diffusion maps algorithm, a non-linear dimension reduction technique, as presented in [1, 2].

Both algorithms were implemented and run in Python on synthetic data sets. We explored the efficacy of the algorithms on data sets of different sizes.

Graphs and Neighborhoods

A similarity graph can be constructed on a data set to encode information about a data set. There are several ways to do this:

- The complete neighborhood graph: Connect every pair of vertices and weight with a kernel
- k nearest neighbors: Connect a vertex with it's k nearest neighbors, see Figure 1. Optionally weight edges
- ε-neighborhood: Connect two vertices if they are within ε of each other

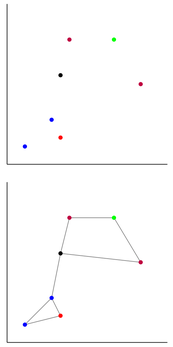

Fig. 1: Construction of a 2-nearest neighbors graph

Markov Chains

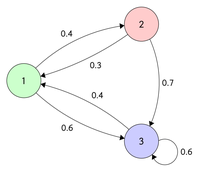

Fig. 2: A simple Markov chain

A Markov chain is a stochastic model that has a set of states and transition probabilities between them. Markov chains can be used to infer global information about a system, like the stationary distribution, by studying local transition probabilities.

We can construct a transition matrix for the Markov chain in Figure 2.

$$M = \begin{bmatrix} 0 & 0.3 & 0.4 \\ 0.4 & 0 & 0 \\ 0.6 & 0.7 & 0.6 \end{bmatrix}$$

Diffusion Maps

The diffusion maps algorithm is a non-linear dimension reduction technique used to recover the underlying manifold geometry of a data set. Let X be the data set, μ a measure on X, and suppose we have a kernel function $k : X \times X \to \mathbb{R}$ that describes similarity or affinity between points. We can normalize k

$$p(x, y) = \frac{k(x, y)}{d(x)} \quad \text{where} \quad d(x) = \int_X k(x, y)\, d\mu(y)$$

to a transition kernel p for a Markov chain on X, and define the diffusion operator

$$Pf(x) = \int_X p(x, y) f(y)\, d\mu(y)$$

Spectral analysis of P and its powers can be used to construct an embedding of the data points into a lower dimensional space. In particular, P has a discrete set of eigenvalues λ_i and eigenfunctions ψ_i. We use eigenfunctions corresponding to the largest eigenvalues to define the family of diffusion maps:

$$\Psi_t(x) = (\lambda_1^t \psi_1(x), \lambda_2^t \psi_2(x), \ \cdots, \lambda_s^t \psi_s(x))$$

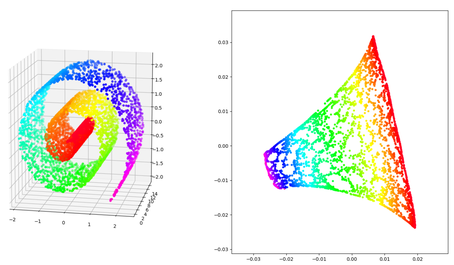

Fig. 3: Left: $n = 5000$ points generated along a Swiss Roll, Right: The embedding generated by the algorithm

The implementation uses k-nearest neighbors with a Gaussian kernel to construct the similarity graph and uses this to compute the low dimensional embedding.

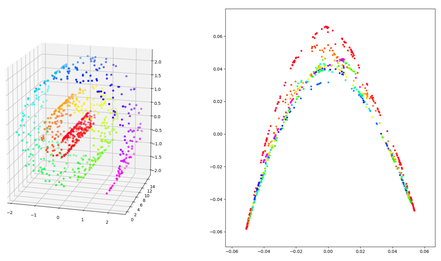

Fig. 4: The diffusion maps algorithm on $n = 800$ points

Running this algorithm with a small number of points does not work as well - there is less information encoded in the similarity graph due to the smaller number of neighbors.

Spectral Clustering

Spectral clustering is a community detection technique that uses the eigenvectors of a graph Laplacian to cluster a data set. We define the unnormalized and normalized Laplacians:

$$L = D - W \quad \text{and} \quad L_{rw} = I - D^{-1}W$$

where D is the degree matrix and W is the weight matrix for the similarity graph. We use the first few eigenvectors of the Laplacian to cluster the data points.

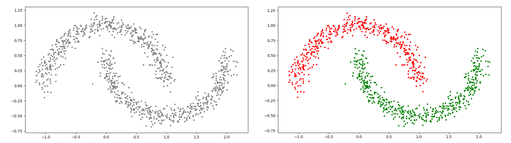

Fig. 5: Left: $n = 1000$ data points generated around two half-moons, Right: Clusters generated by algorithms

Our implementation uses the normalized spectral clustering algorithm by Shi and Malik as presented in [3]. The algorithm uses k-nearest neighbors with a Gaussian kernel to construct a similarity graph.

Spectral clustering more accurately identifies relevant structures in the data set and clusters accordingly; a less sophisticated algorithm would not be able to identify the half-moons in Figure 5.

The eigengaps of the Laplacian provide a heuristic for the ideal number of clusters to make - as seen in Figure 6, 2 is optimal for the half-moons data set in Figure 5.

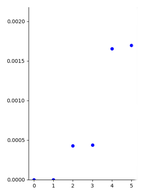

Fig. 6: The eigenvalues of the Laplacian

Acknowledgements

We would like to thank Dr. Michael Lacey, Dr. Wenjing Liao, and Dr. Hao Liu for their expert teaching and guidance throughout the entirety of the program. We would also like to thank the National Science Foundation for providing funding for this REU and to thank all those within the Georgia Tech School of Mathematics who made this program possible.

References

[1] Coifman, R. R. et al. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 102(21), 2005.
[2] Coifman, R. R., & Lafon, S. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21, 5-30, 2006.
[3] Von Luxburg, U. *A Tutorial on Spectral Clustering*. Statistics and Computing, 17(4), 2007.