

Spectral Clustering Implementation and Application

Qi Wang Hanqiu Xia

April 4, 2016

Abstract

Spectral clustering is widely used in image segmentation. The main idea of spectral clustering is to use the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction and then perform clustering in fewer dimensions. In this project, we investigated the normalized spectral decomposition algorithm from the paper "On Spectral Clustering: Analysis and an algorithm" by Ng, Jordan, and Weiss [1]. Their algorithm improved upon the existing spectral clustering algorithm by resolving the issues of inconsistent algorithms of using eigenvectors as well as providing evidence of it resulting in a reasonable clustering. We implemented the algorithm using Python with considering both common and edge cases. We also optimized the Python codes by applying vectorization, Cython, and Just-In-Time compiling. Finally, we tested the algorithm on simulated and real datasets and compare the experimental results with those of k-means clustering to see if spectral clustering dramatically improves the results.

1	Background
2	Implementation
2.1	Algorithm for Spectral Clustering
2.2	Ideal Case
2.3	General Case
3	Testing
3.1	Unit Test for Common Case
3.2	Unit Test for Edge Case
4	Optimization
4.1	Vectorization
4.2	Cython
4.3	JIT (Just-In-Time compiling)
4.4	Improvement result
5	Application and Comparison
5.1	Application on Simulated Data
5.2	Application on Real Data
5.3	Comparison of Spectral clustering and K-means
6	Conclusion

References

- [1] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an algorithm. *Advances in Neural Information Processing*. Vol. 14, No. 2. (2001), pp. 849-856