ID2222 Data Mining
Niklas Barth and Yu Gao

Lab 4 Report

Abstract

In this lab, we are to study, implement and test the spectral graph clustering algorithm described in the paper "On Spectral Clustering: Analysis and an algorithm" by Andrew Y. Ng, Michael I. Jordan, and Yair Weiss.

We are to analyze two sample graphs using our implementation of the K-eigenvector algorithm.

Code Structure

Class Buildingmatrix
This class is designed for building adjacency matrix as well as Laplacian matrix for the input graph data. Please go in the code to see more comments as the code is heavily commented.

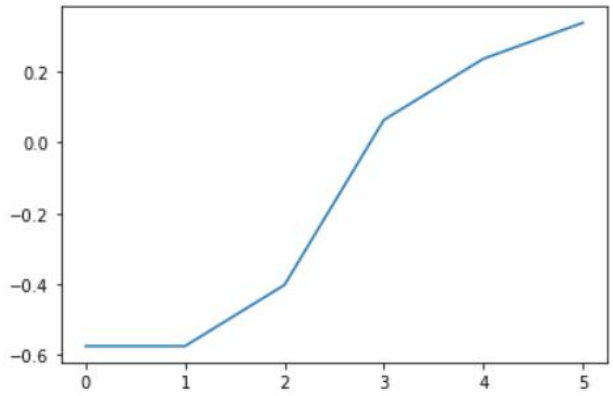
Class Fiedlervector
This class implement the the K-eigenvector algorithm to find the Fiedler vector based on Laplacian matrix. It would tell the user where the Fiedler vector is so that we can cluster points based on their values of Fiedler vectors.

Instructions for use

1. Unzip the project folder. In it you find the dataset in zipped format. Just update the name of files you want to check
2. Run the main class

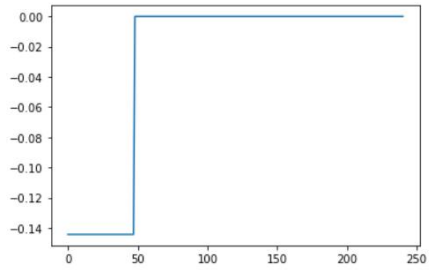
Result

- We choose a similar 7-nodes graph for test if our result is similar with that in instruction
- The results of both given datasets are clearly distributed in 2 clusters. Especially for example 1, there is a sharp gap between two clusters.



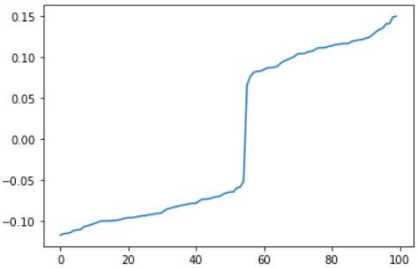
cluster1:[3, 4, 5], cluster2:[0, 1, 2]

Figure 1 Clustering for test graph



cluster1:[117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164], cluster2:[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240]

Figure 2 Clustering for example1 graph



cluster1:[0, 1, 3, 4, 7, 8, 9, 11, 12, 14, 17, 18, 19, 20, 22, 23, 24, 25, 26, 28, 30, 35, 36, 38, 42, 43, 47, 48, 49, 51, 52, 55, 56, 58, 60, 62, 63, 64, 65, 66, 70, 72, 74, 78, 81, 82, 83, 84, 86, 87, 89, 92, 93, 94, 96], cluster2:[2, 5, 10, 13, 15, 16, 21, 27, 29, 31, 32, 33, 34, 37, 39, 40, 41, 44, 45, 46, 50, 53, 54, 57, 59, 61, 67, 68, 69, 71, 73, 75, 76, 77, 79, 80, 85, 88, 90, 91, 95, 97, 98, 99]

Figure 3 Clustering for example2 graph