On Spectral Clustering Analysis and An Algorithm

EE3025 - DSP Lab

K Srikanth - EE18BTECH11023

Indian Institute of Technology Hyderabad

June 8, 2021

Table of Contents

Abstract

- algorithms that cluster points using eigenvectors of matrices de-rived from the data-there are several unresolved issues.

Unresolved Issues

- There are a wide variety of algorithms that use the eigenvectors in slightly different ways
- many of these algorithms have no proof that they will actually compute a reasonable clustering

- The Authors presented a simple spectral clustering algorithm that can be implemented using a few lines of Matlab. Using tools from matrix perturbation theory, we analyze the algorithm, and give conditions under which it can be expected to do well. We also show surprisingly good experimental results on a number of challenging clustering problems

Introduction

- The main application focus of this paper is to clustering points in $\mathbb{R}$ and the spectral methods has recently emerged in a number of fields for clustering and can use the top eigenvectors of a matrix derived from the distance between points.
- **Algorithms Analysis:** One line of analysis makes the link to spectral graph partitioning, in which the second eigenvector of a graph's Laplacian is used to define a semi-optimal cut.

Introduction

- Here, the eigenvector is seen as a solving a relaxation of an NP-hard discrete graph parti-tioning problem, and it can be shown that cuts based on the second eigenvector give a guaranteed approximation to the optimal cut.

- the ma-jority of analyses in spectral graph partitioning appear to deal with partitioning the graph into exactly two parts, these methods are then typically applied recursively to find k clusters. Experimentally it has been observed that using more eigenvectors and directly computing a k way partitioning is better.

Algorithm

Given a set of points $S = \{s_1, s_2..., s_n\}$ in $\mathbb{R}^l$ and the authors want to cluster it into k subsets:

Equations:

Affinity Matrix $A \in \mathbb{R}^{n \times n}$:

$$A_{ij} = \begin{cases} \exp\left(-\left\|s_i - s_j\right\|^2 / 2\sigma^2\right), & i \neq j \\ 0, & i = j \end{cases} \quad (1)$$

Defining D to be the diagonal matrix whose (i, i)-element is the sum of A's i-th row

Construct the below matrix

$$L = D^{-1/2} A D^{-1/2} \quad (2)$$

Algorithm

Finding $\{x_l, x_2, ..., x_k\}$, the k largest eigenvectors of L, and form the matrix $X = [x_l, x_2 ... x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.

Forming the matrix Y from X

$$Y_{ij} = X_{ij} / \Big(\sum_j X_{ij}^2\Big)^{1/2} \tag{3}$$

- Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters via K-means or any other algorithm
- Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.