Tutorial

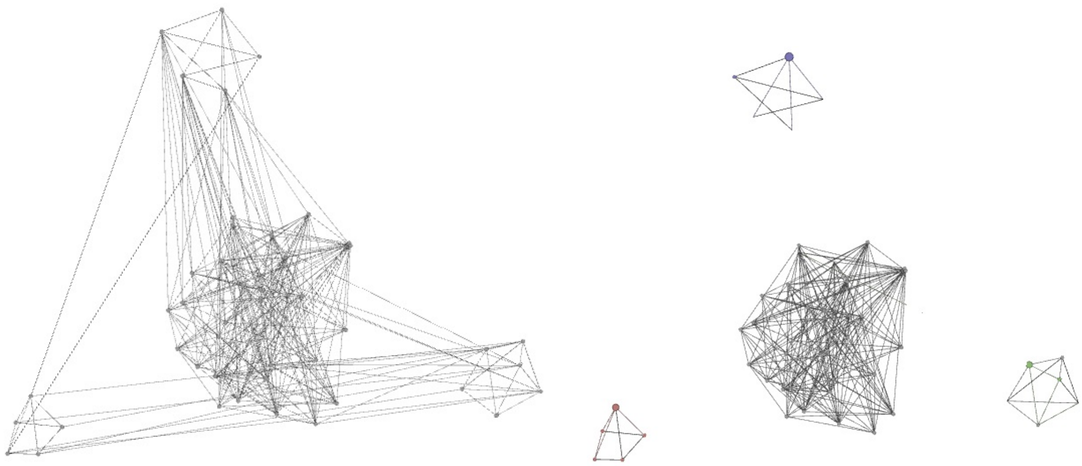
Emad Zahedi and Vahid Mirjalili

Abstract

Here we briefly talk about the Spectral clustering algorithm.

1. Spectral Clustering

Let $G = (V, E)$ be a graph such that V is the set of vertices, and E set of edges. We can transform any data to a graph representation in such a way that vertices are the data points to be clustered and edges are weighted based on similarity between data points Clustering can be viewed as graph partitioning problem.



(a) A graph G (b) Graph partitioning

Figure 1: Each connected component is a cluster.

Basically we look for an objective function to determine what would be the best way cut the edges of the graph.To this end we introduce an algorithm to find the optimal, according to the objective function, partition.

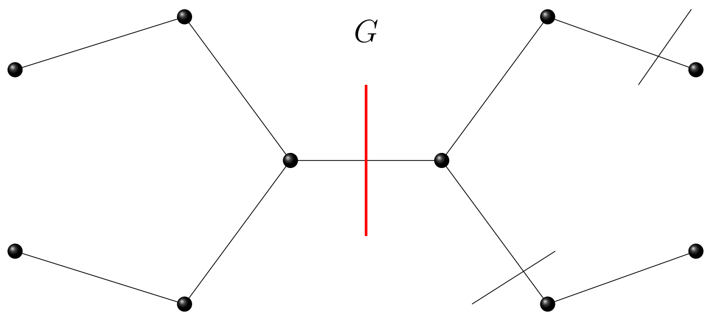


Figure 2: An example for cuts in a graph G.

Suppose we want to partition the set of vertices V into two sets: V_1 and V_2, one possible objective function is to minimize the cut

$$Cut(V_1, V_2) = \sum_{i \in V_1,\ j \in V_2} w_{ij}$$

For most graphs, the optimal solution may correspond to partitioning a single node from the rest of the graph (which is not the desired solution)

We should not only minimize the cut; but should also look for more balanced clusters

$$Ratio\ cut(V_1, V_2) = \frac{Cut(V_1, V_2)}{|V_1|} + \frac{Cut(V_1, V_2)}{|V_2|},$$

$$Normalized\ cut(V_1, V_2) = \frac{Cut(V_1, V_2)}{\sum_{i \in V_1} d_i} + \frac{Cut(V_1, V_2)}{\sum_{i \in V_2} d_i},$$

where $d_i = \sum_j w_{ij}$.
There is an elegant way to optimize the objective function using ideas from spectral graph theory This leads to a class of algorithms known as spectral clustering.

we will examine the spectral properties of a graph, i.e., eigenvalues/eigenvectors of the adjacency matrix representation of a graph. In fact showing that graph partitioning (using the balanced cut criterion) is equivalent to examining the spectral properties of the graph.

Spectral Properties of a Graph Start with a similarity/adjacency matrix, WWW which captures all the information about the relationships between nodes. Next define a diagonal matrix D as follows

$$D_{ij} = \begin{cases} \sum_{k=1}^{n} w_{ik}, & \text{if} \quad i = j \\ 0, & \text{otherwise.} \end{cases}$$

If W is a binary 0/1 matrix, then D_{ii} represents the degree of node i. Now define the laplacian matrix, L as follows

$$L = D - W$$

Clearly $L = DW$ is a symmetric matrix.
L is a positive semi-definite matrix For all real-valued vectors, x: $x^T L x \geq 0$

$$
\begin{aligned}
x^T L x =& x^T (D - W) x = x^T D x - x^T W x \\
=& \sum_i d_i x_i^2 - \sum_{i,j} w_{ij} x_i x_j \\
=& \frac{1}{2} \Big(\sum_{i,j} w_{ij} x_i^2 - 2 \sum_{i,j} w_{ij} x_i x_j + \sum_{i,j} w_{ij} x_i^2 \Big) \\
=& \frac{1}{2} \sum_{i,j=1}^{N} w_{ij} (x_i - x_j)^2.
\end{aligned}
$$

Consequence: all eigenvalues of L are ≥ 0. Vow suppose $e = (1, 1, \cdots, 1)^T$ then

$$
We = \begin{bmatrix} w_{11} & w_{12} \cdots & w_{1d} \\ w_{21} & w_{22} \cdots & w_{2d} \\ \vdots & \ddots & \vdots \\ w_{d1} & w_{d2} \cdots & w_{dd} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} \sum_j w_{1j} \\ \sum_j w_{2j} \\ \vdots \\ \sum_j w_{dj} \end{bmatrix} = \begin{bmatrix} D_{11} \\ D_{22} \\ \vdots \\ D_{dd} \end{bmatrix}
$$

On the other hand we have

$$
De = \begin{bmatrix} D_{11} & 0 \cdots & 0 \\ 0 & D_{22} \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 \cdots & D_{dd} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} D_{11} \\ D_{22} \\ \vdots \\ D_{dd} \end{bmatrix}
$$

Therefore $De = We$ that is $(D - W)e = 0$. Since $L = W - D$ so $Le = 0$. Now using the eigen value equation we have $Le = \lambda e$. Thus $\lambda e = 0$, but $e \neq (0, 0, \cdots, 0)^T$ then $\lambda = 0$.

Corollary 1.1. *The* 0 *is an eigenvalue of* L *with the corresponding eigenvector* $e = (1, 1, \cdots, 1)^T$ *Furthermore, since* L *is positive semi-definite,* 0 *is the smallest eigenvalue of* L.

Spectral properties of a graph (i.e., eigenvalues and eigenvectors) contain information about clustering structure. To find k clusters, examine the first k eigenvectors of the graph Laplacian matrix. Spectral clustering is an algorithm that does this. We saw that clusters can be found by examining the spectral properties of the graph How does this relate to the objective function of minimizing graph cut?
Let x_i indicates the membership of a node in a cluster:

$$
x_i = \begin{cases} \sqrt{V_2/V_1}, & \text{if} \quad v_i \in V_1 \\ -\sqrt{V_1/V_2}, & \text{if} \quad v_i \in V_2. \end{cases}
$$

It is easy to see that

$$x^T Lx = 2|V|\ RatioCut(V_1, V_2)$$

Which means that finding V_1 and V_2 that minimizes $RatioCut$ is equivalent to finding x that minimizes ${}^T Lx$ where L is the graph Laplacian matrix. So we are looking for x in such a wat that

$$min_x\ (x^T Lx)$$

that subjects to the following constraints:

$$x^T 1 = \sum_{i=1}^{n} x_i = \sum_{i \in V_1} \sqrt{V_2/V_1} - \sum_{i \in V_2} \sqrt{V_1/V_2} = 0$$

$$x^T x = \sum_{i=1}^{n} x_i^2 = \sum_{i \in V_1} \left(\sqrt{V_2/V_1}\right)^2 + \sum_{i \in V_2} \left(-\sqrt{V_1/V_2}\right)^2 = |V_1| + |V_2| = n$$

Now we have found the relation between ratio cut and Laplacian matrix L. But what does this have to do with finding eigenvalues and eigenvectors of L?

1.1. Ratio Cut

For the ratio cut in fact we are looking for

$$min_x\ (x^T Lx) \qquad \text{such that} \quad x^T 1 = 0 \quad \text{and} \quad x^T x = n$$

This is a constrained optimization problem where

$$x_i = \begin{cases} \sqrt{V_2/V_1}, & \text{if} \quad v_i \in V_1 \\ -\sqrt{V_1/V_2}, & \text{if} \quad v_i \in V_2. \end{cases}$$

If we solve a relaxation of the problem one can find

$$\phi = x^T Lx - \lambda(x^T x - n)$$

$$\frac{\partial \phi}{\partial x} = Lx - \lambda x = 0 \iff Lx = \lambda x$$

This is really ghood that we can say

$$\min_x\ (x^T Lx) \ = \ \min_x\ (x^T \lambda x) \ = \ min\ \lambda$$

But $\lambda_m in = 0$ with eigenvector $1 = (1, \ldots, 1)^T$ Since we want a solution where $x^T 1 = 0$, so $x \neq 1$. Instead of the smallest eigenvalue, we look for the eigenvector corresponding to the next smallest eigenvalue. Thereby finding the eigenvector that corresponds to the second smallest eigenvalue is a relaxation of the normalized mincut graph partitioning problem.

The Spectral clustering algorithm

....

Consider a data set with N data points.

1: Construct an $N \times N$ similarity matrix, W.

2: Compute the $N \times N$ Laplacian matrix, $L = DW$.

3: Compute the k eigenvectors of L.

a: Each eigenvector v_i is an $N \times 1$ column vector.

b: Create a matrix V containing eigenvectors $v_1 \cdots v_k$ as columns

4: Cluster the rows in V using k-means or any clustering algorithm.

....