Indian Statistical Institute

Spectral Graph Clustering

Name : **Siladittya Manna**

Registration No.: JCSK-CC-0094

Contents

1 Graphs and Graph Laplacian

1.1 Undirected Graphs

Definition : A *graph* (or *undirected graph*) is a pair G=(V,E), where $V = \{v_1,...,v_n\}$ is a set of nodes or vertices, and E is a set of two-element subsets of V (that is, subsets $\{u,v\}$, with u, $v \in V$ and $u \neq v$), called *edges.*

For every node $v \in V$, the *degree* $d(v)$ of v is the number of edges incident to v:

$$d(v) =\mid \{u \in V \mid u, v \in E\} \mid$$

The *degree matrix* $D(G)$, is the diagonal matrix

$$D(G) = diag(d_1, ..., d_n).$$

1.2 Adjacency Matrix

Definition : Given a graph G=(V,E), with $V = \{v_1,...,v_n\}$, the *adjacency matrix* $A(G)$ of G is the symmetric $n \times n$ matrix (a_{ij}) sucht that

$$a_{ij} = \begin{cases} 1 & \text{if there is some edge } \{v_i,v_j\} \in \text{E} \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

The adjacency matrix can be viewed as a *diffusion operator.*

$$\phi' = A\phi = \textstyle\sum_{i \sim j} \phi(j)$$

where ϕ is any real valued function on the set of graph's vertices, $\phi : V(G) \mapsto \mathbb{R}$. Such a function assigns a real number to each vertex.

This gives the amount of diffusion from each vertex, or otherwise stated, it gives the sum of the weights to the neighbours of each vertex in a weighted graph. In an unweighted graph, we get the degree of each vertex, when multiplied by a unit vector of length $\mid V \mid$.

It can also be used in *Quadratic* form

$$\phi^T A\phi = \sum_{e_{ij}} \phi(i)\phi(j) \tag{1.2}$$

Adjacency matrix is symmetric and hence has real and *orthogonal* eigenvectors.

1.3 Weighted Graph

Definition : A *weighted graph* is defined as $G = (V, E, W)$, where $V = \{v_1, ..., v_n\}$ is a set of *nodes* or *vertices* and W is a symmetric matrix called the *weight matrix*, such that $w_{ij} \geq 0 \; \forall i, j \in \{1, ..., n\}$, and $w_{ii} = 0 \; \forall i = 1, ..., m$. We say that a set $\{v_i, v_j\}$ is an edge iff $w_{ii} > 0$. The corresponding graph (V, E) with $E = \{\{v_i, v_j\} \mid w_{ij} > 0\}$, is called the *underlying graph* of G.

For every node $v \in V$, the *degree* $d(v_i)$ of v_i is the sum of the weights of the edges adjacent to v_i:

$$d(v) = \textstyle\sum_{j=1}^{n} w_{ij}$$

The degree matrix D is defined as before, namely by $D(G) = diag(d_1, ..., d_n)$..

1.4 Laplacian of a Graph

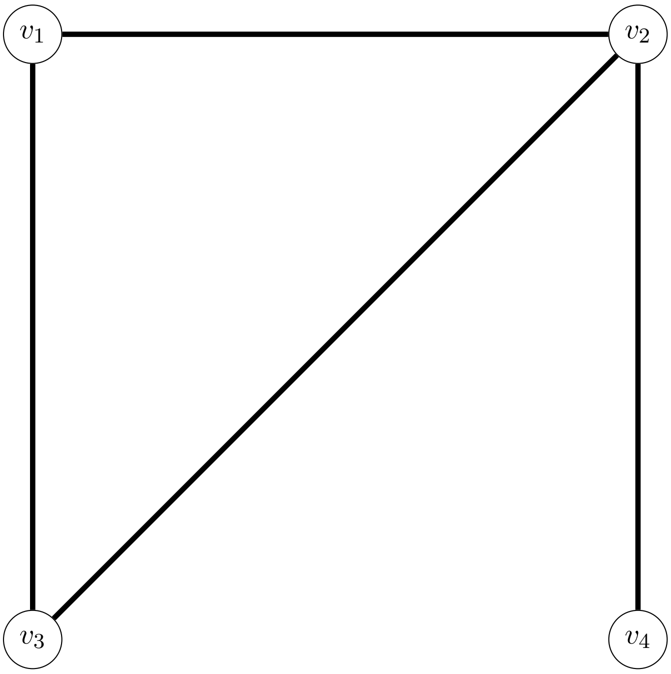
Definition : The *Laplacian* of a graph G is defined as a $N \times N$ matrix L such that

$$l_{ij} = \begin{cases} d_v & \text{if } u = v \\ -1 & \text{if } u \text{ and } v \text{ are adjacent} \end{cases} \tag{1.3}$$

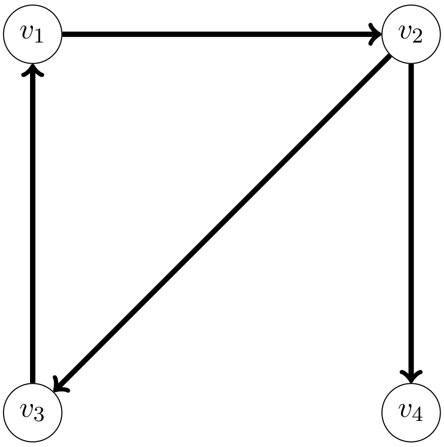
Let each edge of a graph have an arbitrary but fixed direction (clockwise or anticlockwise). The *incidence* matrix of a graph is a matrix of size $\mid E \mid \times \mid V \mid$ and is defined as:

$$\nabla = \begin{cases} \nabla_{ev} = -1 & \text{if } v \text{ is the initial vertex of the edge } e \\ \nabla_{ev} = 1 & \text{if } v \text{ is the terminal vertex of the edge } e \\ \nabla_{ev} = 0 & \text{if } v \text{ is not in } e \end{cases} \tag{1.4}$$

For example take the graph below as an example



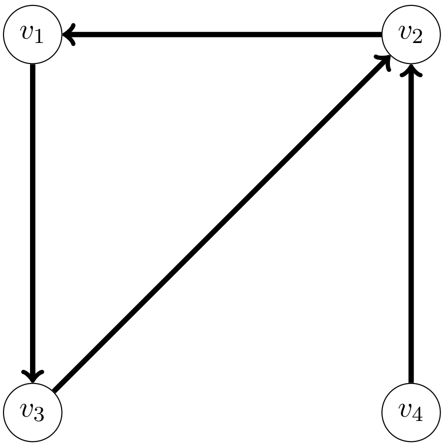
Assigning orientation of the edges in clockwise direction, we get



The incidence matrix, we obtain from this graph is

$$\nabla = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{1.5}$$

Again, assigning orientation of the edges in counterclockwise direction, we get



The incidence matrix, we obtain from this graph is

$$\nabla = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \tag{1.6}$$

This incidence matrix serves a *discrete differential* operator and the mapping $f \mapsto \nabla f$ is known as ***co-boundary mapping***.

$$(\nabla\phi)(e_{ij}) = \phi(v_j) - \phi(v_i) \tag{1.7}$$

Hence, the Laplacian matrix L of a graph G is defined as $\nabla^T\nabla$.

$$L = \nabla^T\nabla \tag{1.8}$$

Thus, when used as an operator,

$$(L\phi)(v_i) = \sum_{v_j \sim v_i} (\phi(v_j) - \phi(v_i)) \tag{1.9}$$

This Laplacian operator in *Quadratic* form

$$\phi^T L\phi = \frac{1}{2}\sum_{e_{ij}}(\phi(i) - \phi(j))^2 \tag{1.10}$$

The Laplace matrix of the example graph G when assigned orientation in the clockwise direction is

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

We can see that when assigned orientation in the counterclockwise direction, we get the same Laplacian matrix L.

From the above example, we can see that the relation between the *Laplacian* matrix L and the *Adjacency* matrix A is

$$L = D - A \tag{1.11}$$

If the graph G under consideration is a weighted undirected graph, where each edge e_{ij} has weight $w_{ij} > 0$, then the equation (1.9) becomes

$$(L\phi)(v_i) = \sum_{v_j \sim v_i} w_{ij}(\phi(v_j) - \phi(v_i)) \tag{1.12}$$

Subsequently, equation (1.10) becomes

$$\phi^T L\phi = \frac{1}{2}\sum_{e_{ij}} w_{ij}(\phi(i) - \phi(j))^2 \tag{1.13}$$

Important properties of the Laplacian matrix L are:

- L is symmetric and positive semi-definite.
- L has n non-negative and real-valued eigenvalues.

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$$

Since, L is symmetric, the eigenvectors of L, corresponding to distinct eigenvalues are orthogonal.

1.5 Normalized Laplacian of a Graph

Definition : Given a weighted undirected graph $G = (V, E, W)$ with no isolated vertex and with $V = \{v_1, v_2, \ldots, v_n\}$, the normalized graph Laplacian L_{sym} and L_{rw} is defined as

$$L_{sym} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \tag{1.14}$$

$$L_{rw} = D^{-1}L = I - D^{-1}W = D^{-\frac{1}{2}}L_{sym}D^{\frac{1}{2}} \tag{1.15}$$

This matrix L_{sym} is a symmetric, positive and semi-definite and can be used as a operator like the Laplacian matrix.

$$\phi^T L_{sym}\phi = \frac{1}{2}\sum_{i,j=1}^{N} w_{ij}\left(\frac{\phi_i}{\sqrt{d_i}} - \frac{\phi_j}{\sqrt{d_j}}\right)^2 \tag{1.16}$$

A nonzero vector u is a solution of the generalized eigenvalue problem $Lu = \lambda Du$ iff $D^{\frac{1}{2}}u$ is an eigenvector of L_{sym} and u is an eigenvector of L_{rw} for the same eigenvalue of λ. Hence, as it is evident that 1_n is in the nullspace of L, as also in the nullspace of L_{rw}, the nonzero vector $D^{\frac{1}{2}}1_n$ is in the nullspace of L_{sym}.

1.6 Laplacian of a Graph with One single component

Let L be the laplacian matrix of a graph G.

To find the eigenvectors and eigenvalues, we use the equation

$$L\psi = \lambda\psi$$

Putting $\psi = 1_n$, in the above equation, it is obvious, that we will get,

$$L1_n = 0$$

Hence, $\lambda_1 = 0$ is an eigenvalue of L, and it is the smallest eigenvalue, since all eigenvalues of L are non-negative.

Since, all entries of ψ are same, equation (1.13) gives

$$\psi^T L\psi = 0 = \frac{1}{2}\sum_{e_{ij}} w_{ij}(\phi(j) - \phi(i))^2$$

For every pair of vertices (v_i,v_j), we have $\psi(v_i) = \psi(v_j)$. By induction, $\psi(v_i) = \psi(v_j)$ for every pair of vertices, since G is connected. Hence, ψ must be a constant vector.
Thus when the graph G is connected, the *Eigenspace* is spanned by a *Eigenvector* ψ with *Eigenvalue* 0 and has dimension 1.

Since the eigenvectors form an orthonormal basis,

$$\psi_i^T \psi_j = \delta_{ij}$$

For any eigenvector $\psi_i = (\psi_i(v_1), \ldots, \psi_i(v_n))^T, 2 \leq i \leq n$

$$\psi_i^T 1_n = 0$$

The above equation translates to

$$\sum_{j=1}^{N} \psi_i(v_j) = 0 \tag{1.17}$$

1.7 Laplacian of a graph with K>1 connected components

Let the graph G is disconnected and consists of two components G_1 and G_2.

$$G = G_1 \cup G_2$$

Let the component G_1 contain vertices $V_1 = \{v_1, \ldots, v_m\}$ and G_2 contain vertices $V_2 = \{v_{m+1}, \ldots, v_n\}$.▌

Let A_1 be the *Adjacency* matrix for the vertices in the component G_1 and A_2 be the *Adjacency* matrix for the component G_2. Since the graph G is disconnected and consists of the two components G_1 and G_2, there is no connection between the vertices in V_1 with any vertex in V_2, hence all the elements outside of A_1 will be *zeros*. Similarly for the vertices in V_2, no connection exists with vertices in V_1, hence all elements outside A_2 will be *zeros*.

Thus, the adjacency matrix can be written as a block diagonal matrix.

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \tag{1.18}$$

From equation (1.11), $L = D - A$, hence, the *Laplacian* matrix will also be a block diagonal matrix. Let the blocks be L_1 and L_2. Since, each component is a connected graph in itself, the Laplacian matrices L_1 and L_2, each have eigenvalues 0, with eigenvectors $\begin{bmatrix} 1_m & 0_{n-m} \end{bmatrix}^T$ and $\begin{bmatrix} 0_m & 1_{n-m} \end{bmatrix}^T$ respectively. Thus we get two linearly independent *eigenvector* with the *eigenvalue* 0, that is, the multiplicity of the eigenvalue 0 is two.

For generalizing the above result, we assume the graph G to consist of K components, namely, $G_1, G_2, \ldots, G_k$. Thus the *Laplacian* matrix of G is of the form

$$L_G = \begin{bmatrix} L_1 & \ldots & & \\ \vdots & L_2 & \ldots & \\ & \vdots & \ddots & \vdots \\ & & \ldots & L_k \end{bmatrix} \tag{1.19}$$

We will get k independent *eigenvectors* with eigenvalue 0. Thus the *eigenspace* will be spanned by k independent and mutually orthogonal vectors $u_i = 1_{L_i}$, where $1_{L_i} = (0...011..110...0)^T \in \mathbb{R}$. Thus, the multiplicity of the *eigenvalue* 0 is equal to the number of connected components in the underlying graph G.

1.8 Fiedler Vector

The first non-null eigenvalue of the Laplacian matrix of the underlying graph G is called the ***Fiedler*** value and the corresponding *eigenvector* is called the ***Fiedler*** vector.

It is interesting to note that, the numtiplicity of the Fiedler value is always 1. The Fiedler value is the *algebraic connectivity* of the graph G. The further it is from 0, the more connected the graph is. If this value becomes 0, then it implies that the graph G is disconnected.

1.9 Rayleigh Quotient and Courant-Fischer Theorem

If $G = (V, E)$ be the graph and L be the Laplacian of the graph G, then by *Courant-Fischer* formula

$$\lambda_2 = \min_{\substack{\phi \neq 0 \\ \phi \perp 1_n}} \frac{\phi^T L \phi}{\phi^T \phi} = \min_{\substack{\phi \neq 0 \\ \phi \perp 1_n}} \frac{\sum_{(i,j) \in E} (\phi(i) - \phi(j))^2}{\sum_{i \in V} \phi(i)^2} \tag{1.20}$$

OR

$$\lambda_n = \max_{\substack{\phi \neq 0 \\ \phi \perp 1_n}} \frac{\phi^T L \phi}{\phi^T \phi} = \max_{\substack{\phi \neq 0 \\ \phi \perp 1_n}} \frac{\sum_{(i,j) \in E} (\phi(i) - \phi(j))^2}{\sum_{i \in V} \phi(i)^2} \tag{1.21}$$

2 Graph Clustering

2.1 Introduction

Given any subset of nodes $V' \subseteq V$, we define the *volume* $vol(V')$ of V' as the sum of the weigths of all edges adjacent to nodes in V':

$$vol(V') = \sum_{v_i \in V'} d(v_i) = \sum_{v_i \in V'} \sum_{j=1}^{m} w_{ij} \tag{2.1}$$

Given any two subsets $V_1, V_2 \subseteq V$, links(V_1, V_2) is defined as

$$links(V_1, V_2) = \sum_{v_i \in V_1, v_j \in V_2} w_{ij} \tag{2.2}$$

Since the weight matrix W is symmetric,

$$links(V_1, V_2) = links(V_2, V_1)$$

and observe that

$$vol(V_1) = links(V_1, V).$$

We define $\overline{V'}$ as the complement of V' in V. Then the quantity

$$cut(V') = links(V', \overline{V'})$$

is often called the *cut* of V'.
The quantity

$$assoc(V') = links(V', V')$$

is called the *association* of V'.
Clearly,

$$cut(V') + assoc(V') = vol(V').$$

2.2 Graph Clustering Using Normalized Cuts

The goal of clustering is to partition the data points into different groups according to some similarity metric. If the data given is in terms of similarity graph G where the weights of the edges w_{ij} between two nodes v_i and v_j is the measure of similarity between the two nodes, then the problem can be stated as follows:

Find a partition $(V_1, V_2, \ldots, V_K)$ of the set of nodes V into different groups such that hte edges between different groups have very low weights or similarity and the edges within a group have high weight or similarity.

The above problem can however be formulated into an optimization problem where the objective is to minimize the quantity $cut(V_1, V_2, \ldots, V_K)$.

$$cut(V_1, V_2, \ldots, V_K) = \frac{1}{2}\sum_{i=1}^{K} cut(V_i) \tag{2.3}$$

If we take $K = 2$, then it becomes a classical mincut problem, but in practice it does not yield satisfcatory partitions. It often separates one single vertex from the rest of the graph. Hence to keep the partitions reasonably large, the *cuts* are normalized by dividing by some measure of each subset V_i. One possible way is to use the size of V_i, and the other one is to use the ***volume*** $vol(V_i)$ of V_i.

Hence, the optimization problem is converted to

$$Ncut(V_1, \ldots, V_K) = \sum_{i=1}^{K} \frac{links(V_i, \overline{V_i})}{vol(V_i)} = \sum_{i=1}^{K} \frac{cut(V_i, \overline{V_i})}{vol(V_i)} \tag{2.4}$$

In a seminal paper of Shi and Malik [2], Normalized Cut was represented as:

$$Ncut(V_1, \ldots, V_K) = \frac{cut(V_1, V - V_1)}{assoc(V_1, V)} + \frac{cut(V_2, V - V_2)}{assoc(V_2, V)} + \ldots + \frac{cut(V_K, V - V_K)}{assoc(V_K, V)} \tag{2.5}$$

The **significance** of using $assoc(V_i, V)$ instead of $vol(V_i)$ lies in the relation between the two. As can be seen from the relation $cut(V_i, V - V_i) + assoc(V_i, V) = vol(V_i)$, using $vol(V_i)$ in the denominator allows $assoc(V_i, V)$ to become **ZERO**, which in this case, that is the application of *Spectral Graph Clustering* to *Image Segmentation* means an ***isolated pixel*** or ***node***, which is undesirable. Hence, using $assoc(V_i, V)$ in the denominator forces it to take **non-zero** value, since the algorithm presented in the paper by Shi and Malik[2] had a prespecified limit on the value of $NCut(V_1, V_2, \ldots, V_K)$, thereby preventing it from taking an *infinite* value and thus does not allow isolated nodes.

2.3 2-way Clustering Using Normalized Cuts

In case of 2-way clustering, in which we need to partition the set of vertices V into two clusters, a single vector X can be used to describe the partition $(V', \overline{V'})$.

The structure of the vector X needs to be chosen in such a way that $Ncut(V', \overline{V'}) = \frac{X^T LX}{X^T DX}$

An extensive proof has been given in the seminal paper of Shi and Malik [2]

Let V' and $\overline{V'}$ be the two partitions of the set of nodes V of graph G. Also let ϕ be an $N = \mid V \mid$ dimensional ***indicator*** vector, such that $\phi_i = 1$ if node i is in V' and -1 otherwise.

With the definition of Normalized cut provided in Shi and Malik [2] $Ncut(V', \overline{V'})$ can be written as:

$$Ncut(V', \overline{V'}) = \frac{cut(V_1, V_2)}{assoc(V_1, V)} + \frac{cut(V_2, V_1)}{assoc(V_2, V)} = \frac{\sum_{(\phi_i>0,\phi_j<0)} -w_{ij}\phi_i\phi_j}{\sum_{\phi_i>0} d_i} + \frac{\sum_{(\phi_i<0,\phi_j>0)} -w_{ij}\phi_i\phi_j}{\sum_{\phi_i<0} d_i} \tag{2.6}$$

Let

$$k = \frac{\sum_{\phi_i>0} d_i}{\sum_i d_i} \tag{2.7}$$

and **1** be an $N \times 1$ vector of all ones.

Using the fact that $\frac{1+\phi}{2}$ and $\frac{1-\phi}{2}$ are indicator vectors for $\phi_i > 0$ and $\phi_i < 0$ respectively, we can rewrite $4[Ncut(\phi)]$ as :

$$4[Ncut(\phi)] = \frac{(1+\phi)^T L(1+\phi)}{k1^T D1} + \frac{(1-\phi)^T L(1-\phi)}{(1-k)1^T D1} = \frac{(\phi^T L\phi + 1^T L1)}{k(1-k)1^T D1} + \frac{(2(1-2k)1^T L\phi)}{k(1-k)1^T D1} \tag{2.8}$$

Let,

$$\alpha(\phi) = \phi^T L\phi \tag{2.9}$$

$$\beta(\phi) = 1^T L\phi \tag{2.10}$$

$$\gamma = 1^T L1 \tag{2.11}$$

$$M = 1^T D1 \tag{2.12}$$

We can rewrite equation (2.8) as

$$\begin{aligned} 4[Ncut(\phi)] &= \frac{(\alpha(\phi)+\gamma) + 2(1-2k)\beta(\phi)}{k(1-k)M} \\ &= \frac{(\alpha(\phi)+\gamma) + 2(1-2k)\beta(\phi)}{k(1-k)M} \\ &\quad - \frac{2(\alpha(\phi)+\gamma)}{M} + \frac{2\alpha(\phi)}{M} + \frac{2\gamma}{M} \end{aligned} \tag{2.13}$$

We can drop the last constant term, since it equals zero. Thus the above equation becomes

$$\begin{aligned}4[Ncut(\phi)] =& \frac{(1-2k+2k^2)(\alpha(\phi)+\gamma)+2(1-2k)\beta(\phi)}{k(1-k)M} + \frac{2\alpha(\phi)}{M} \\ =& \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(\alpha(\phi)+\gamma+\frac{2(1-2k)}{(1-k)^2}\beta(\phi)}{\frac{k}{1-k}M} + \frac{2\alpha(\phi)}{M}\end{aligned} \tag{2.14}$$

Letting $b = \frac{k}{1-k}$ and since $\gamma = 0$, it becomes

$$\begin{aligned}4[Ncut(\phi)] =& \frac{(1+b^2)(\alpha(\phi)+\gamma)+2(1-b^2)\beta(\phi)}{bM} + \frac{2b\alpha(\phi)}{bM} \\ =& \frac{(1+b^2)(\alpha(\phi)+\gamma)}{bM} + \frac{2(1-b^2)\beta(\phi)}{bM} + \frac{2b\alpha(\phi)}{bM} - \frac{2b\gamma}{bM} \\ =& \frac{(1+b^2)(\phi^TL\phi+1^TL1)}{b1^TD1} + \frac{2(1-b^2)1^TL\phi}{b1^TD1} + \frac{2b\phi^TL\phi}{b1^TD1} - \frac{2b1^TL1}{b1^TD1} \\ =& \frac{(1+\phi)^TL(1+\phi)}{b1^TD1} + \frac{b^2(1-\phi)^TL(1-\phi)}{b1^TD1} - \frac{2b(1-\phi)^TL(1+\phi)}{b1^TD1} \\ =& \frac{[(1+\phi)-b(1-\phi)]^TL[(1+\phi)-b(1-\phi)]}{b1^TD1}\end{aligned} \tag{2.15}$$

Now, setting $X = (1+\phi) - b(1-\phi)$, we can see that

$$X^TD1 = \sum_{\phi_i>0} d_i - b\sum_{\phi_i<0} d_i = 0 \tag{2.16}$$

Since, we previously put $b = \frac{k}{1-k} = \frac{\sum_{\phi_i>0} d_i}{\sum_{\phi_i<0} d_i}$ and

$$\begin{aligned}X^TDX &= \sum_{\phi_i>0} d_i + b^2\sum_{\phi_i<0} d_i \\ &= b\sum_{\phi_i<0} d_i + b^2\sum_{\phi_i<0} d_i \\ &= b\left(\sum_{\phi_i>0} d_i + b\sum_{\phi_i<0} d_i\right) \\ &= b1^TD1\end{aligned} \tag{2.17}$$

Putting together everything, we get,

$$\min_{\phi} Ncut(\phi) = \min_{X} \frac{X^T L X}{X^T D X} \tag{2.18}$$

with the condition that $X(i) \in \{1, -b\}$ and $X^T D 1 = 0$.

The right hand side of the above equation is the *Rayleigh Quotient*.
It is natural to assume that the vector X is of the form

$$X = (x_1, \ldots, x_N)$$

The vector X is an indicator vector in the sense that

$$X = \begin{cases} 1 & \text{if } v_i \in V' \\ -b & \text{if } v_i \notin V' \end{cases} \tag{2.19}$$

If X is relaxed to take on only real values, then the above equation (2.18) can be minimized by solving the eigenvalue system,

$$LX = \lambda D X \tag{2.20}$$

The second constraint on X is automatically taken care of by the solution of the generalized eigensystem.

This is done by transforming the equation (2.20) by rewriting it as

$$D^{-\frac{1}{2}} L D^{-\frac{1}{2}} Y = \lambda Y \tag{2.21}$$

where $Y = D^{\frac{1}{2}} X$.

It is evident that $D^{\frac{1}{2}} 1$ is an eigenvector of Y with eigenvalue 0. Also, $D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ is a symmetric positive semidefinite matrix, since L is also positive semidefinite. Hence, translating back to the previous eigensystem, we get, $X_1 = 1$ as the smallest eigenvector with eigenvalue 0 and consequently we get the second smallest eigenvector X_2

$$X_2 = \min_{X^T D 1 = 0} \frac{X^T L X}{X^T D X} \tag{2.22}$$

This second smallest eigenvector is the solution to the normalized cut problem.

Since the first condition on X was relaxed, the condition is not automatically satisfied. However, the second smallest eigenvector exactly minimizes the problem

$$\inf_{X^T D1=0} \frac{\sum_i \sum_j (X(i) - X(j))^2 w_{ij}}{\sum_i X(i)^2 d_i} \tag{2.23}$$

in real valued domain.

This forces the indicator vector X to take up similar values for nodes which are similar. Thus the second smallest eigenvector can be used to bipartition the graph into two components. However, the **GROUPING ALGORITHM** presented in the seminal paper of Shi and Malik [2] had one more step, which involved deciding if the current partition should be subdivided by checking the *stability* of the cut and making sure that the value of $Ncut$ is below the prespecified value and recursively partitioning the segmented parts if necessary. This above mentioned algorithm was used to segment an image, where each pixel was taken as a node and each pair of pixel was connected by an edge, with weight

$$w_{ij} = \exp \frac{-\|F_i - F_j\|_2^2}{\sigma_I^2} * \begin{cases} \exp \frac{-\|X_i - X_j\|_2^2}{\sigma_X^2} & \text{if } \|X_i - X_j\|_2^2 < r \\ 0 & otherwise \end{cases} \tag{2.24}$$

Solving a standard eigenvalue problem for all eigenvectors is very expensive and takes $O(n^3)$ operations, where n is the number of nodes in the graph.

However, it is interesting to note that, using the equation (2.4), gets us rid of a lot of pain.

$$Ncut(V_1, V_2) = \frac{cut(V_1, V_2)}{vol(V_1)} + \frac{cut(V_2, V_1)}{vol(V_2)} \tag{2.25}$$

Since, $cut(V_1, V_2) = cut(V_2, V_1)$, we can rewrite the above equation as,

$$NCut(V_1, V_2) = cut(V_1, V_2) \left(\frac{1}{vol(V_1)} + \frac{1}{vol(V_2)} \right) \tag{2.26}$$

Let $d = 1^T D1$ and $vol(V_1) = \alpha$. Then $vol(V_2) = d - \alpha$.

Assuming the vector X to be an indicator vector

$$X = \begin{cases} a & \text{if } v_i \in V_1 \\ b & \text{if } v_i \notin V_1 \end{cases} \tag{2.27}$$

We have

$$X^T LX = (a - b)^2 cut(V_1, V_2) \tag{2.28}$$

$$X^T DX = \alpha a^2 + (d - \alpha)b^2 \tag{2.29}$$

Equating $Ncut(V_1, V_2)$ and the Rayleigh Quotient, we get

$$NCut(V_1, V_2) = \frac{d}{\alpha(d-\alpha} cut(V_1, V_2) = \frac{(a-b)^2}{\alpha a^2 + (d-\alpha)b^2} cut(V_1, V_2) \tag{2.30}$$

We find the condition

$$a\alpha + b(d-\alpha) = 0 \tag{2.31}$$

Now, X should be orthogonal to $D1$
Shi and Malik [2] use

$$a = 1, b = -\tfrac{\alpha}{d-\alpha}$$

Let,

$$\chi = \{(x_1, \dots, x_N) \mid x_i \in \{a, b\}, a, b \in \mathbb{R}, a, b \neq 0\} \tag{2.32}$$

Our solution set is

$$\kappa = \{\chi \in X \mid X^T D1 = 0\} \tag{2.33}$$

This problem now proceeds as a standard eigenvalue problem, where the second smallest eigenvector can be found out by solving the eigensystem, as in equation (2.21)

2.4 K-Way Clustering Using Normalized Cuts

For partitioning a graph G into K clusters, we formulate the partition $(V_1, V_2, \dots, V_K)$ of the set of nodes V by an $N \times K$ matrix $X = \left[X^1 \dots X^K\right]$, where the columns X_i are indicator vectors of the partitions V_i. Again, we assume the vector X^i is of the form

$$X^i = (x_1^i, \dots, x_N^i)$$

From equation (2.5)

$$NCut(V_1, \dots, V_K) = \sum_i \frac{cut(V_i, \overline{V_i})}{assoc(V_i, V)} \tag{2.34}$$

Now, we have already seen that for the **2-way Clustering** case, that,

$$\min_{\phi} NCut(\phi) = \min_{X} \frac{X^T L X}{X^T D X} \tag{2.35}$$

We can reformulate the above equation for K-way case as

$$\begin{aligned}\min_{\Phi} NCut(V_1, \ldots, V_K) &= \min_{X} \sum_{i} \frac{(X^i)^T L X^i}{(X^i)^T D X^i} \\ &= \min_{X} tr(\Lambda^{-1} X^T L X)\end{aligned} \tag{2.36}$$

where $\Lambda = diag((X^1)^T D X^1, \ldots, (X^K)^T D X^K)$.
subject to

$(X^i)^T D X^j = 0, 1 \leq i, j \leq K, i \neq j,$
$X(X^T X)^{-1} X^T 1 = 1,$
$X \in \chi$

The columns of X are nonzero and D-orthogonal, hence they must be linearly independent, and X has rank K and $X^T X$ is invertible.
We can replace $(X^T X)^{-1} X^T$ by X^+. Therefore the second condition becomes $XX^+ = 1$
Since, the problem is scale-invariant, we can drop Λ^{-1} from the equation (2.36), by introducing the condition that

$$X^T D X = I \tag{2.37}$$

OR

$$(X^j)^T D X^j = 1, \forall 1 \leq j \leq K \tag{2.38}$$

Now, relaxing the condition $X \in \chi$ and converting L to its symmetric form L_{sym} by change of variables $Y = D^{\frac{1}{2}} X$, the optimization problem is converted to
From equation (2.37), changing the variable we get, $YY^T = 1$, hence we have $Y^+ = Y^T$.

$$\begin{aligned}\min_{\Phi} NCut(V_1, \ldots, V_K) &= \min_{Y} tr(Y^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} Y) \\ &= \min_{Y} tr(Y^T L_{sym} Y)\end{aligned} \tag{2.39}$$

subject to

$Y^T Y = 1$
$YY^T D^{\frac{1}{2}} 1 = D^{\frac{1}{2}} 1$

Replacing $Z = D^{-\frac{1}{2}}Y$, we pass from a solution Y of the problem to a solution Z of the problem.

In a bid to solve the above eigensystem, we need to visit the Poincaré Separation Theorem, which provides a lower and upper bound of eigenvalues of a real symmetric matrix of the form B^TAB. Specifically speaking, if A be a $N \times N$ real symmetric matrix and B be $N \times K$ semi-orthogonal matrix such that $B^TB = I_K$, then denoting the eigenvalues of A and B^TAB by λ_i and μ_i respectively, we have

$$\lambda_i \geq \mu_i \geq \lambda_{n-r+i} \tag{2.40}$$

Thus, $tr(Y^TL_{sym}Y)$ is lower bounded by the sum of K smallest eigenvalues of L_{sym}, that is,

$$\sum_{i=1}^{K} \lambda_i \leq tr(Y^TL_{sym}Y) \tag{2.41}$$

If the second constraint in equation (2.39) is temporarily ignored, the minimum of $tr(Y^TL_{sym}Y)$ is achieved by any K unit eigenvectors associated with the smallest eigenvalues

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_K$$

of L_{sym}.

If we assume that $\lambda_2 > 0$, that is the underlying graph is connected, then $Y^1 = D^{\frac{1}{2}}1/\left\|D^{\frac{1}{2}}1\right\|_2$, because $\mathbf{1}$ is in the nullspace of L. Hence the second constraint

$$YY^TD^{\frac{1}{2}}1 = D^{\frac{1}{2}}1$$

is also satisfied.

Thus $Z = D^{-\frac{1}{2}}Y$ yields a minimum of the relaxed problem in the form of equation (2.39).

By equation (1.15) and the properties mention in the subsection **(1.5)**, the solutions of the eigensystem obtained by changing variables by putting $Z = D^{-\frac{1}{2}}Y$, that is, the vectors Z^j are eigenvectors of L_{rw}, associated with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_K$.

From subsection (**1.5**), 1_n is in the nullspace of L_{rw}, hence 1_n is an eigenvector for the eigenvalue $\lambda_1 = 0$ and $Z^1 = 1_n/\left\|D^{\frac{1}{2}}1_n\right\|_2$.

Because, $(Y^i)^TY^j = 0 \ \forall i \neq j$, we have

$$(Z^i)^TDZ^j = 0, \ \forall i \neq j \tag{2.42}$$

This implies that $Z^i \perp D1_n \; \forall i = 2 \dots K$. Thus, Z^i has both some positive and negative components.

However, it is to be kept in mind that the conditions $(Z^i)^T DZ^j = 0$ do not necessarily imply Z^i and Z^j are orthogonal. However, solutions satisfying both $(Z^i)^T Z^j = 0$ and $(Z^i)^T DZ^j = 0$ can be obtained by multiplying Z by some $K \times K$ orthogonal matrix on the right.

The $K \times K$ matrix $Z^T Z$ can be diagonalized by some orthogonal $K \times K$ matrix R as

$$Z^T Z = R \Sigma R^T \tag{2.43}$$

where Σ is a diagonal matrix, and thus,

$$R^T Z^T Z R = (ZR)^T Z R = \Sigma \tag{2.44}$$

which again shows that the columns of ZR are orthogonal.

Now in the equation $X(X^T X)^{-1} X^T 1 = 1$, if we replace X by XR, then also it is satisfied. Thus, ZR satisfies the constraints of the eigenvalue problem obtained by replacing $Z = D^{-\frac{1}{2}} Y$ in equation (2.39). Also, $tr((ZR)^T L(ZR)) = tr(Z^T LZ)$.

The matrix R can be found by *Singualr Value Decomposition* of Z.

References

[1] Jean H. Gallier. Notes on elementary spectral graph theory. applications to graph clustering using normalized cuts. *CoRR*, abs/1311.2492, 2013.

[2] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000.

[3] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

[4] Radu Horaud. A short tutorial on graph laplacians, laplacian embedding, and spectral clustering.

[5] Anne E Marsden. Eigenvalues of the laplacian and their relationship to the connectedness. 2013.

[6] Andries E. Brouwer and Willem H. Haemers. *Spectra of Graphs.* New York, NY, 2012.

[7] László Lovász. Eigenvalues of graphs. 2007.

[8] Ulrike von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.