

[WORK IN PROGRESS]

Report on Paper "A tutorial on spectral clustering"  
by Ulrike von Luxburg

Sourabh Antani

# Chapter 1

## Introduction

This tutorial is setup up as self-contained introduction to spectral clustering. It derives spectral clustering from scratch and present different points of view to why spectral clustering works.

# Chapter 2

## Similarity graphs

Given a set of data points  $x_1, \dots, x_n$  and some notion of similarity  $s_{ij} \geq 0$  between all pairs of data points  $x_i$  and  $x_j$ , the intuitive goal of clustering is to divide the data points into several groups such that the points in the same group are similar and the points in different groups are dissimilar to each other. A nice way of representing the data is in form of *similarity graph*  $G = (V, E)$ . Each vertex  $v_i$  in this graph represents a data point  $x_i$ . Two vertices are connected if the similarity  $s_{ij}$  between the corresponding data points  $x_i$  and  $x_j$  is positive or larger than a certain threshold and the edge is weighted by  $s_{ij}$ . The problem of clustering can now be reformulated using the similarity graph: we want to find the partition of the graph such that the edges between different groups have very low weights (which points to dissimilarity) and the edges within the group have high weights (hence similar).

### 2.1 Some Terminology:

$G = (V, E)$  undirected graph

$V = \{v_1, \dots, v_n\}$  vertex set

$E$  set of weighted edges such that each edge between two vertices  $v_i$  and  $v_j$  carries a non-negative weight  $w_{ij} \geq 0$ .

**Weighted adjacency matrix** of the graph is matrix  $W = (w_{ij})_{i,j=1,\dots,n}$ . If  $w_{ij} = 0$ , vertices  $v_i$  and  $v_j$  are not connected.

$w_{ij} = w_{ji}$  since  $G$  is undirected.

**The degree of a vertex**  $v_i \in V$  is defined as  $d_i = \sum_{j=1}^n w_{ij}$ .

**degree matrix**  $D$  is defined as the diagonal matrix with the degrees  $d_1, \dots, d_n$  on the diagonal.

$\bar{A}$  Complement  $V \setminus A$  of a given subset  $A \subset C$ .

**Indicator vector**  $\mathbf{1}_A$  vector with entries  $f_i = 1$  if  $v_i \in A$ ,  $f_i = 0$  otherwise.

$W(A, B) = \sum_{i \in A, j \in B} w_{ij}$ , for non-necessarily disjoint sets.

$|A|$  = number of vertices in  $A$  = size of set  $A$

$vol(A) = \sum_{i \in A} d_i$

**Connected** subset  $A \subset V$  of a graph is connected if any two vertices in  $A$  can be joined by a path such that all intermediate points also lie in  $A$ .

**Connected Component** Connected subset such that  $A$  and  $\bar{A}$  are disjoint

**Partition** non-empty sets  $A_1, \dots, A_k$  form partition of graph if  $A_i \cap A_j = \emptyset$  and  $A_1 \cup \dots \cup A_k = V$

## 2.2 Different similarity graphs

**$\epsilon$ -neighbourhood graph** Connect points whose pairwise distances are less than  $\epsilon$ . Weight data is not incorporated into the graph.

**$k$ -nearest neighbor graphs** Connect the nearest  $k$  points. Usually directed since  $v_i$  being in the  $k$  nearest neighbors of  $v_j$  does not mean that  $v_j$  is in  $k$  nearest neighbors of  $v_i$

**Fully connected graph** is a graph where all points with positive similarity with each other are connected. The edges here are weighted by  $s_{ij}$ . This type of construction is only useful when the similarity function models local neighborhoods since a graph should represent local neighborhood relationships.

# Chapter 3

## Graph Laplacians and their basic properties

**Assumptions:**  $G$  is an undirected, weighted graph with weight matrix  $W$ , where  $w_{ij} = w_{ji} \geq 0$ . Eigenvectors are not normalized and multiples of an eigenvector are considered to be equivalent. Eigenvalues are always ordered increasingly, respecting multiplicities. So 'the first  $k$  eigenvectors' will refer to eigenvectors corresponding to the  $k$  smallest eigenvalues.

### 3.1 Unnormalized graph Laplacian: $L = D - W$

**Proposition 1** (Properties of  $L$ ) The matrix  $L$  satisfies:

1. for every vector  $f \in \mathbb{R}^n$  we have

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$$

2.  $L$  is symmetric and positive semi-definite

3. The smallest eigenvalue of  $L$  is 0, the corresponding eigenvector is the constant vector  $\mathbf{1}$

4.  $L$  has  $n$  non-negative, real-values eigenvalues  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

*Proof:*

1. By definition of  $d_j = \sum_{i=1}^n w_{ij}$

$$\begin{aligned} f^T L f &= f^T D f - f^T W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n f_i f_j w_{ij} \\ &= \frac{1}{2} \left( \sum_{i=1}^n d_i f_i^2 - 2 \sum_{i,j=1}^n f_i f_j w_{ij} + \sum_{i=j}^n d_j f_j^2 \right) \\ &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \end{aligned}$$

2. Since  $W$  and  $D$  are symmetric  $L = D - W$  is still symmetric. By result of part (a), since  $w_{ij} \geq 0$  and  $(f_i - f_j)^2 \geq 0$ ,  $f^T L f \geq 0$ , hence  $L$  is symmetric and positive semi-definite

3. See proposition 2

4. Since  $L$  is symmetric, all its eigenvalues are real. Also by part (c), the smallest eigenvalue is 0,

Note that the unnormalized graph Laplacian does not depend on the diagonal elements of  $W$ .  $d_{ii} = \sum_{j=1}^n w_{ij}$ , hence the diagonal elements are cancelled by  $d_{ii}$  and the self-edges do not change the Laplacian.

**Proposition 2** (Number of connected components and the spectrum of  $L$ ) *Let  $G$  be an undirected graph with non-negative weights. Then the multiplicity  $k$  of the eigenvalue 0 of  $L$  equals the number of connected components  $A_1, \dots, A_k$  in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors  $\mathbf{1}_{A_1}, \dots, \mathbf{1}_{A_k}$  of those components.*

*Proof:* Starting with  $k = 1$ , i.e. graph is fully connected. Assume that  $f$  is the eigenvector with eigenvalue 0. Then,

$$0 = f^T L f = \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \implies f_i - f_j = 0 \quad [\because w_{ij} > 0]$$

Since we assumed a fully connected graph,  $f$  has to be constant for the above argument to be true for any vertex in the connected component. Hence the vector  $\mathbf{1}$ , which is the indicator vector of the component is an eigenvector corresponding to eigenvalue 0.

Now, for  $k > 1$ , we can easily rearrange the vertices in the matrix to put all the vertices of a connected component together. This would cause the matrix  $L$  to be a block diagonal matrix. For each of these blocks, or components, the respective indicator vector is an eigenvector with eigenvalue 0. Thus the Matrix  $L$  has eigenvalue 0 with multiplicity equal to the number of components and the corresponding eigenvectors would be the indicator vectors of those components.

## 3.2 Normalized Graph Laplacians

There are two normalized graph Laplacians found in literature:

1. Symmetric Laplacian:  $L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$

This can be derived simply by decomposing the diagonal matrix  $D$  into  $D^{1/2} D^{1/2}$  and then left and right multiplying the equation  $L = D - W = D^{1/2} D^{1/2} - W$  by  $D^{-1/2}$

2. Random walk Laplacian:  $L_{rw} = D^{-1}L = I - D^{-1}W$   
This can be derived by left multiplying  $L = D - W$  by  $D^{-1}$

**Proposition 3** (*Properties of  $L_{sym}$  and  $L_{rw}$* )

1. For every  $f \in \mathbb{R}^n$

$$f' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \left( \frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

2.  $\lambda$  is an eigenvalue of  $L_{rw}$  with eigenvector  $u$  iff  $\lambda$  is an eigenvalue of  $L_{sym}$  with eigenvector  $w = D^{1/2}u$ .
3.  $\lambda$  is an eigenvalue of  $L_{rw}$  with eigenvector  $u$  iff  $\lambda$  and  $u$  solve the generalized eigenproblem  $Lu = \lambda Du$
4. 0 is an eigenvalue of  $L_{rw}$  with the constant one vector  $\mathbf{1}$  as eigenvector. 0 is an eigenvalue of  $L_{sym}$  with eigenvector  $D^{1/2}\mathbf{1}$
5.  $L_{sym}$  and  $L_{rw}$  are positive semi-definite and have  $n$  non-negative real-valued eigenvalues  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

*Proof:*

1. This can be proved similar to part 1 of proposition 1.
2. Assume  $\lambda$  is an eigenvalue of  $L_{sym}$  with eigenvector  $w = D^{1/2}u$ .

$$\begin{aligned} L_{sym} D^{1/2}u &= \lambda D^{1/2}u \\ \implies D^{-1/2} L D^{-1/2} D^{1/2}u &= \lambda D^{1/2}u \\ \implies D^{-1} L u &= \lambda u \quad [\text{left multiplying with } D^{-1/2}] \\ \implies L_{rw} u &= \lambda u \end{aligned}$$

In otherwords,  $u$  is eigenvector of  $L_{rw}$ . To prove the other direction, apply the above steps in reverse order.

3. This can be proven in similar manner to above by left multiplying the generalized eigenproblem  $Lu = \lambda Du$  with  $D^{-1}$
4.  $L_{rw}\mathbf{1} = (I - D^{-1}W)\mathbf{1} = \mathbf{1} - \mathbf{1} = 0$ .  
The above is true because  $d_i = \sum_{j=1}^n w_{ij} \implies (D^{-1}W)\mathbf{1}_i = \sum_{j=1}^n \frac{w_{ij}}{d_i}$ , since  $\mathbf{1}$  is the indicator vector. Then using part (b) leads to the second statement of this property.
5. The first part (for  $L_{sum}$ ) can be proved from part (a), similar to Proposition 1. Part 2 essentially states that  $L_{sym}$  and  $L_{rw}$  have the same eigenvalues, hence second statement is also proven.

**Proposition 4** (*Number of connected components and spectra of  $L_{sym}$  and  $L_{rw}$* ) Let  $G$  be an undirected graph with non-negative weights. Then the multiplicity  $k$  of the eigenvalue 0 of both  $L_{sym}$  and  $L_{rw}$  equals to the number of connected components  $A_1, \dots, A_k$  in the graph. for  $L_{rw}$ , the eigenspace of 0 is spanned by the indicator vectors  $\mathbf{1}_{A_i}$  of these components. for  $L_{sym}$  the eigenspace of 0 is spanned by  $D^{1/2}\mathbf{1}_{A_i}$

*Proof:* The proof of this is analogous to that of proposition 2, but using proposition 3.

## Chapter 4

# Spectral clustering algorithms

This chapter lists 3 classic algorithms, one for unnormalized and two for normalized spectral clustering. These have been implemented in python notebooks on the git repository and are excluded here.

## Chapter 5

# Graph cut point of view

The intuition of clustering is to separate points into groups so that the edges between groups have low weight. Thus the clustering can be viewed as mincut problem.

Recall the notation  $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$  and  $\bar{A}$  for complement of  $A$ . Thus choosing the  $k$  partitions  $A_1, \dots, A_k$  is to minimize

$$cut(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^k W(A_i, \bar{A}_i)$$

The factor  $1/2$  is for notational consistency, otherwise the edges in the cut will be counted twice, once from  $A$  to  $\bar{A}$  and then from  $\bar{A}$  to  $A$ . The problem is that in many cases this separates in to 'unbalanced' clusters where some clusters have only one (or very few) nodes while others are fairly large. One way to circumvent is to explicitly request that the sets  $A_1, \dots, A_k$  are 'reasonably large'. Two common approaches are RatioCut (Hagen & Kahng, 1992) where the size of  $A_i$  is measured by the number of vertices  $|A|$  and Normalized cut, or Ncut, (Shi & Malid 2000) where the size  $A_i$  is measured by sum of weights of its edges  $vol(A)$ . The definitions are

$$RatioCut(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{|A_i|}$$

$$Ncut(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{vol(A_i)} = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{vol(A_i)}$$

Note that in both cases, the objective functions take the minimum value if the 'measures' (number of vertices or sum of weights) for all the subsets (i.e. partitions/clusters) coincide (are reasonably close), thus aiding in generating 'balanced' clusters.

Unfortunately adding this 'balancing' condition makes the problem NP-hard. Therefore in practice this is achieved by relaxing the definitions and that is what spectral clustering achieves.

## 5.1 Approximating RatioCut for $k = 2$

The goal is to solve the optimization problem

$$\min_{A \subset V} \text{RatioCut}(A, \bar{A})$$

Given  $A \subset V$ , define  $f = (f_1, \dots, f_n)' \in \mathbb{R}^n$  with entries

$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|}, & \text{if } v_i \in A, \\ \sqrt{|A|/|\bar{A}|}, & \text{if } v_i \in \bar{A} \end{cases}$$

Now, we can write the RatioCut using Unnormalized graph Laplacian

$$\begin{aligned} f'Lf &= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2 \\ &= \frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} \left( \sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right) \\ &\quad + \frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} \left( -\sqrt{\frac{|\bar{A}|}{|A|}} - \sqrt{\frac{|A|}{|\bar{A}|}} \right) \\ &= \frac{1}{2} \sum_{i=1}^k W(A_i, \bar{A}_i) \left( \frac{|\bar{A}_i|}{|A_i|} + \frac{|A_i|}{|\bar{A}_i|} + 2 \right) \\ &= \text{cut}(A, \bar{A}) \left( \frac{|A| + |\bar{A}|}{|A|} + \frac{|A| + |\bar{A}|}{|\bar{A}|} \right) \quad \left[ \because 2 = \frac{|A|}{|A|} + \frac{|\bar{A}|}{|\bar{A}|} \right] \\ &= |V| \cdot \text{RatioCut}(A, \bar{A}) \quad \left[ \because |A| + |\bar{A}| = |V|, \frac{\text{cut}(A, \bar{A})}{|A|} + \frac{\text{cut}(A, \bar{A})}{|\bar{A}|} = \text{RatioCut}(A, \bar{A}) \right] \end{aligned}$$

Also,

$$\begin{aligned} f \cdot \mathbf{1} &= \sum_{i=1}^n f_i = \sum_{i \in A} \sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i \in \bar{A}} \sqrt{\frac{|A|}{|\bar{A}|}} = |A| \sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}| \sqrt{\frac{|A|}{|\bar{A}|}} = 0 \implies f \perp \mathbf{1} \\ \|f\|^2 &= \sum_{i=1}^n f_i^2 = |A| \frac{|\bar{A}|}{|A|} + |\bar{A}| \frac{|A|}{|\bar{A}|} = |A| + |\bar{A}| = n \end{aligned}$$

Thus the goal is to find such a vector  $f$  that minimizes  $f'Lf$  under the conditions  $f \perp \mathbf{1}$  and  $\|f\| = \sqrt{n}$ . Here  $n$  is the total number of vertices in the graph, hence is a constant in this equation. This means that the vector  $f$  that minimizes  $f'Lf$  will also minimize  $\frac{f'Lf}{n} = \frac{f'Lf}{\|f\|^2}$ .

Rayleigh-Ritz ([4]) theorem states that

Let  $A \in \mathbb{C}$  be a Hermitian Matrix, and 'Rayleigh quotient' be the function defined as

$$R : \mathbb{C}^n \setminus \{\mathbf{0}\} \rightarrow \mathbb{R}, R(x) = \frac{x^H A x}{x^H x}, \|x\| \neq 0$$

then the eigenvectors of  $A$  are the critical points of  $R(x)$  and eigenvalues of  $A$  are the values of  $R(x)$  at those critical points.

Consequently,

$$\lambda_{\max} = \max_{\|x\| \neq 0} \frac{x^H A x}{x^H x} \quad \text{and} \quad \lambda_{\min} = \min_{\|x\| \neq 0} \frac{x^H A x}{x^H x}$$

Thus,  $f$  would be the eigenvector corresponding to the smallest eigenvalue of  $L$ . However, we know that the smallest eigenvalue of  $L$  is 0 with eigenvector as  $\mathbf{1}$ . Hence, we can use the second eigenvector of  $L$  as the minimizer of the RatioCut function. This is just the solution to the 'relaxed' problem. Hence to find the partition of the graph, we need to transform this real-valued solution into discrete solution. The simplest way to do this is to use the sign of the  $f$  as the indicator function. i.e. split the graph into  $A$  and  $\bar{A}$  by  $A$  is set of the vertices where the corresponding element is non-negative and  $\bar{A}$  is the remaining. i.e.

$$\begin{cases} v_i \in A, & f_i \geq 0 \\ v_i \in \bar{A} & f_i < 0 \end{cases}$$

In practice, what most spectral clustering algorithms do, instead is to treat the elements (coordinates) of  $f$  as points in  $\mathbb{R}$  and use  $k$ -means to cluster them into two clusters (say  $C_1$  and  $C_2$ ) and then carry over the clustering to the graph by choosing points as

$$\begin{cases} v_i \in A, & f_i \in C_1 \\ v_i \in \bar{A} & f_i \in C_2 \end{cases}$$

This is exactly the unnormalized spectral clustering for  $k = 2$ .

## 5.2 Approximating RatioCut for arbitrary $K$

Generalizing the above method, we partition the set of vertices  $V$  into  $k$  sets  $A_1, \dots, A_k$ . We define  $k$  indicator vectors  $h_j = (h_{1,j}, \dots, h_{n,j})'$  by

$$h_{i,j} = \begin{cases} 1/\sqrt{|A_j|} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases} \quad (i = 1, \dots, n; j = 1, \dots, k)$$

Let  $H \in \mathbb{R}^{n \times k}$  be the matrix of the  $k$  indicator vectors above. Note that the columns of  $H$  are orthonormal to each other, i.e.  $H'H = I$ . Applying the calculations similar to the last section we find that

$$h_i' L h_i = \frac{\text{cut}(A_i)}{|A_i|} = (H' L H)_{ii}$$

Thus,

$$\text{RatioCut}(A_1, \dots, A_k) = \sum_{i=1}^k h_i' L h_i = \sum_{i=1}^k (H' L H)_i = \text{Trace}(H' L H)$$

Hence the goal of clustering would be minimize Trace of  $H' L H$  with  $H$  orthonormal as defined above. Again, a version of Rayleigh-Ritz theorem ([5]) states that this is achieved by taking the first  $k$  eigenvectors of  $L$  corresponding to the  $k$  smallest eigenvalues. Note that  $H$  is the same as matrix  $U$  in unnormalized spectral clustering mentioned in chapter 4. Again converting this to the discretion partition, we get the unnormalized spectral clustering algorithm mentioned in chapter 4.

## 5.3 Approximating Ncut

Similar to RatioCut, for  $k = 2$  we define cluster indicator vector  $f$  by

$$f_i = \begin{cases} \sqrt{\frac{\text{vol}(\bar{A})}{\text{vol}(A)}} & \text{if } v_i \in A, \\ -\sqrt{\frac{\text{vol}(A)}{\text{vol}(\bar{A})}} & \text{if } v_i \in \bar{A} \end{cases}$$

Similar to RatioCut, we can check that  $(Df)' \mathbf{1} = 0$  and  $f' D f = \text{vol}(V)$ . Thus  $f' L f = \text{vol}(V) \text{Ncut}(A, \bar{A})$ . So the problem is now converted to minimizing  $f' L f$  subject to  $f$  as defined above such that  $Df \perp \mathbf{1}$  and  $f' D f = \text{vol}(V)$ .



Again, relaxing the problem by allowing to take arbitrary values, we substitute  $g = D^{1/2}f$ . Now the minimization problem becomes

$$\min_{g \in \mathbb{R}^n} g' D^{-1/2} L D^{-1/2} g, \quad \text{such that } g \perp \mathbf{1}, \|g\|^2 = \text{vol}(V)$$

Since  $D^{-1/2} L D^{-1/2} = L_{sym}$ , the problem becomes, again to minimize  $g' L_{sym} g$  with  $\|g\| = \sqrt{\text{vol}(V)}$ . Once again, Rayleigh-Ritz theorem states that the minimizer is the eigenvector of  $L_{sym}$ . We know that the first eigenvector is  $D^{1/2} \mathbf{1}$  and constant eigenvalue is  $\text{vol}(V)$ , the solution of this minimization problem is given by second eigenvector of  $L_{sym}$ . Re-substituting  $f = D^{-1/2} g$  and using parts (b) and (c) of proposition 3, we can state that, since  $g = D^{1/2} f$  is an eigenvector of  $L_{sym}$ , letting  $\lambda$  be the corresponding eigenvalue,  $f$  is eigenvector of  $L_{rw}$  and solves the generalized eigenvalue problem  $Lu = \lambda Du$ .

Extending this for  $k > 2$ , let

$$h_{i,j} = \begin{cases} 1/\sqrt{\text{vol}(A_j)} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases} \quad (i = 1, \dots, n; j = 1, \dots, k)$$

Now, set the matrix  $H$  as the matrix containing first  $k$  indicator vectors as columns. Observe that  $H'H = I$ ,  $h'_i D h_i = 1$  and  $h'_i L h_i = \text{cut}(A_i, \bar{A}_i)/\text{vol}(A_i)$ . Hence the minimization problem now becomes the trace minimization problem for  $\text{Trace}(H' L H)$  subject to  $H' D H = I$ . Relaxing the discreteness condition and substituting  $T = D^{1/2} H$ , we get  $\min T \in \mathbb{R}^{n \times k} T' L_{sym} T$ , subject to  $T' T = I$ . Once again, combining the Rayleigh-Ritz theorem as applied to RatioCut, and Proposition 3 parts (b) and (c) we can see that the solution is to construct  $H$  with the first  $k$  eigenvectors of  $L_{rw}$ , or the first  $k$  generalized eigenvectors of  $Lu = \lambda Du$ . This yields the normalized spectral clustering algorithm according to Shi and Malik (2000) which is also mentioned in chapter 4, and implemented in a jupyter notebook on github repository.

## 5.4 Comments on relaxation approach

The author notes that the relaxation approach leads to an approximation of the real solution. There are no guarantees about the quality of the cut generated by these algorithms. For example, in the cockroach graph (which looks like a horizontal ladder with few rungs removed), the ideal cut will be vertical where the rungs have been removed. (need to get graphic in here) but the unnormalized spectral cut always generates horizontal cut. Here the ideal RatioCut = 1 but unnormalized clustering generates  $2/k$ , thus order of  $n$  factors worse. it is known that an algorithm that generates a constant RatioCut does not exist.

The relaxation techniques used here are not unique and many such techniques have been devised, mainly because they usually generated a standard Linear algebra problem to solve.

# Bibliography

- [1] need to add citation for lecture slides by Dr. Zhou for Rayleigh quotient
- [2] github repository with implemented algorithms.
- [3] Cheung, Gene, Magli, Enrico, Tanaka, Yuichi, Ng, Michael K. "Graph Spectral Image Processing". Proceedings of the IEEE, vol 106, No. 5, pp. 907-930 May 2018
- [4] <https://planetmath.org/RayleighRitzTheorem>
- [5] <http://people.inf.ethz.ch/arbenz/ewp/Lnotes/chapter13.pdf>