平成 29 年度

筑波大学情報学群情報科学類

卒業研究論文

題目 Title of the Thesis

主専攻　　ソフトウェアサイエンス主専攻

著者　　Farley Soares Oliveira

指導教員　櫻井鉄也

Abstract

Here you write the abstract of your thesis.

Contents

List of Figures

Chap. 1 Introduction

1.1 Notation

Chap. 2 Spectral Clustering

In this chapter we define the clustering problem, describe general ways in which it can be solved, and introduce spectral clustering as a solution to this problem which uses the Courant-Fischer Min-Max Theorem. The material here is based mainly on [4] and [6]. However, we have changed the notations and some of the presentation, selecting only the relevant parts for the rest of this thesis. We have also provided several proofs which were omitted in the original papers. Readers who are already familiar with the derivation of spectral clustering may feel free to skip this chapter.

2.1 The clustering problem

Clustering is currently the most popular way of conducting unsupervised learning. Given a dataset $\mathcal{D}$, the objective of clustering is to find a partition of $\mathcal{D}$, $(\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_k)$, where $k \in \mathbb{Z}_{>0}$ is predetermined by the user of the algorithm, such that the similarity of elements of a same subset $\mathcal{P}_i$ (with $i \in [\![1\,,\,k]\!]$) are as big as possible and the similarity of elements of different subsets $\mathcal{P}_i$ and $\mathcal{P}_j$ (where $(i,j) \in [\![1\,,\,k]\!]^2, i \neq j$) are as small as possible. In other words, a clustering algorithm assigns a label $l \in [\![1\,,\,k]\!]$ to each data instance in $\mathcal{D}$ in such a way that data instances which are similar to each other are assigned the same label. The way the similarity of elements of a same subset and the similarity of elements of different subsets are calculated depends on the clustering algorithm used. We can say the same about the way in which the dataset $\mathcal{D}$ is represented.

Clustering may be achieved by several different approaches, each with its own advantages and disadvantages. Some models and approaches for clustering are as follows:

- Strict partioning clustering: each data instance is classified into one cluster based on its similarity with other instances. The main approach for this type of clustering is k-means: the algorithm works by iteratively assigning a label to each data instance based on its similarity with each cluster. Here, the similarity of a data instance and a cluster is obtained by computing the similarity between the instance and some kind of representative data instance of the cluster, usually some kind of mean vector.

- Hierarchical clustering: the data is divided in clusters which make up a hierarchy. This type of clustering may be achieved by two main approaches: the agglomerative approach, where each data instance starts in its own cluster, and pairs of clusters are merged as we go up in the hierarchy; and the divisive approach, where all data instances start in a same cluster, and clusters are split as we

go down the hierarchy. One advantage of hierarchical clustering is that the algorithm user does not need to set the number of subsets k ahead of time.

- Overlapping clustering: In the final result, each data instance may be an element of more than one cluster. In other words, $(\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_k)$ is not necessarily a partition of $\mathcal{D}$. This approach may be useful when certain data instances naturally pertain to more than one class.

In contrast to the approaches above, spectral clustering works by transforming the data into a graph, constructing a certain matrix associated to this graph called Laplacian, computing the eigenvalues and eigenvectors of the Laplacian, and finally using this eigeninformation to classify the data. Although spectral clustering is often more difficult to implement (requiring, e.g., an algorithm to efficiently solve an eigenproblem), it is more general than the more common approaches such as k-means and hierarchial clustering. This is because spectral clustering may be succesfuly used for data that are arranged in complex shapes (as long as each cluster is connected), since the data is first mapped from their native data space to another one in which connectivity is preserved but geometrical relationships are simplified.

There are two main approaches with which spectral clustering can be derived. The first approach, the *ideal case* approach, considers regular Laplacian matrices as perturbations of an ideal case in which data points that are to be classified into different clusters are infinitely far apart. The second approach, the *relaxation* approach, considers spectral clustering as a approximation algorithm to solve a original NP-complete discrete optimization problem. The first approach is related to the FAST-GE2 algorithm, and the second one is related to the Bethe Hessian spectral clustering algorithm, both of which will be discussed henceforth in this thesis. For this reason, we will explain both approaches in this chapter.

We show the Spectral Clustering Algorithm below and explain why it outputs a valid result in the subsequent sections.

Algorithm 1 Spectral Clustering

Require:

Adjacency Matrix: $A \in \mathbb{R}^{m \times m}$

Number of Clusters: $k \in \mathbb{Z}_{>0}$

Ensure:

Partition of the set of vertices V: $\{C_1, C_2, \cdots, C_k\} \subseteq V$

1: Compute the normalized Laplacian L of A.
2: Compute the first k eigenvectors $(x_1, x_2, \cdots, x_k) \in (\mathbb{R}^{m \times 1})^k$ of L.
3: Let $X \in \mathbb{R}^{m \times k}$ be the matrix containing the vectors $x_1, x_2, \cdots, x_k$ as columns.
4: Form the matrix $Y \in \mathbb{R}^{m \times k}$ by normalizing the columns of X.
5: Let $(y_1, y_2, \cdots, y_m) \in (\mathbb{R}^{1 \times k})^m$ represent the row-vectors of Y.
6: Cluster $(y_1, y_2, \cdots, y_m)$ using k-means into clusters $\{D_1, D_2, \cdots, D_k\}$.
7: For each $i \in [\![1, k]\!]$, set $C_i = \{v_j : y_j \in D_i\}$.

2.2 Preliminary definitions

Before entering in the discussion of each derivation, we will give some definitions common to both approaches. Here we will assume that each data instance $d \in \mathcal{D}$ is a n dimensional column vector, i.e. $\mathcal{D} \subseteq \mathbb{R}^{n\times 1}$.

Definition Let $\mathcal{D}$ be a dataset containing m elements. The *similarity matrix* $A \in \mathbb{R}^{m\times m}$ associated with $\mathcal{D}$ is defined as follows:

$$A_{ij} = s\left(d_i, d_j\right), \text{ for each } (i,j) \in [\![1\,,\, m]\!]^2, \tag{2.1}$$

where s is a similarity measure and $d_i \in \mathcal{D}$ for each $i \in [\![1\,,\, m]\!]$. In this thesis, we will only consider the *Gaussian similarity measure*, which is given by

$$\begin{aligned} s_G : \quad & E^2 \longrightarrow \mathbb{R} \\ & (x,y) \longmapsto \exp\left(-\frac{1}{2\sigma^2}\|x-y\|^2\right), \end{aligned} \tag{2.2}$$

where E is a normed vector space with norm $\|\cdot\|$ and $\sigma \in \mathbb{R}$ is a parameter set by the user which controls the width of the neighborhoods.

We can think of the similarity matrix above as encoding the *adjacency matrix* of a weighted graph $G_{\mathcal{D}} = (V_{\mathcal{D}}, E_{\mathcal{D}})$ representing the dataset $\mathcal{D}$. In this case, each element A_{ij} of A represents the weight of an edge connecting the vertices $(v_i, v_j) \in {V_{\mathcal{D}}}^2$.

Remark It is convenient here to establish a bijective relationship between the similarity matrix of a dataset and the adjacency matrix of graph. In other words, although we have seen that we may obtain a new graph (represented by an adjacency matrix A) from a dataset with similarity matrix A, we may also obtain a new dataset (represented by a similarity matrix A) from a graph with adjacency matrix A.

Definition Let $A \in \mathbb{R}^{m\times m}$ be the adjacency matrix of a graph $G = (V, E)$. The *unnormalized Laplacian matrix* of the graph G is defined by

$$L_0 = D - A, \tag{2.3}$$

where $D \in \mathbb{R}^{m\times m}$ is defined to be the diagonal matrix whose D_{ii} elements are given by the sum of the elements of the matrix A's i-th row, for all $i \in [\![1\,,\, m]\!]$.

Definition Let $L_0 \in \mathbb{R}^{m\times m}$ be the unnormalized Laplacian matrix of a graph $G = (V, E)$. The *normalized Laplacian matrix* of the graph G is defined by

$$L = D^{-1/2} L_0 D^{-1/2}, \tag{2.4}$$

where $D \in \mathbb{R}^{m\times m}$ is defined to be the diagonal matrix whose D_{ii} elements are given by the sum of the elements of the matrix A's i-th row, for all $i \in [\![1\,,\, m]\!]$.

2.3 Ideal case approach

Here we will consider the ideal case for spectral clustering where, for all $(i, j) \in [\![1, m]\!]^2$, $A_{ij} = 0$ whenever d_i and d_j are in different clusters, and $A_{ij} > 0$ otherwise. We will only consider the case where the number of clusters k is 3 and we will assume that, for all $(i, j) \in [\![1, m]\!]^2$, $d_i \in \mathcal{D}$ are ordered in such a way that $i < j$ whenever the label of d_i is smaller than the label of d_j (remember that the labels l are elements of $[\![1, k]\!]$). The argument, however, can be easily extended to a general case.

In this section, for algebraic convenience, we will use an alternative definition for the unnormalized Laplacian matrix: $L_{\text{new}} = D^{-1/2}AD^{-1/2}$, instead of $L = D^{-1/2}L_0D^{-1/2}$. The use of this trick is justified by the fact that, since $L_0 = D - A$, we have that $L_{\text{new}} + L = I_m$, where I_m is the identity matrix of order m. Therefore L_{new} and L possess the same eigenvectors, and, for each $i \in [\![1, m]\!]$, if λ_i is an eigenvalue of L, then $1 - \lambda_i$ is an eigenvalue of L_{new}. Outside of this section, however, we will use the normal definition of unnormalized Laplacian as given in the previous section.

Before entering the derivation, we need to outline a result.

Proposition 2.3.1 *Let G be a connected graph of order m. The normalized Laplacian associated with G has the eigenvalue 1 with positive eigenvector. Furthermore, all the other eigenvalues are strictly smaller than 1.*

This is a basic result in spectral graph theory. A proof may be found in, e.g., [3].

Consider the dataset $\mathcal{D}$ corresponding to the graph G and let $d_i \in \mathcal{D}$ for all $i \in [\![1, m]\!]$. Since $A_{ij} = 0$ whenever d_i and d_j are in different clusters, $A \in \mathbb{R}^{m\times m}$ may be expressed as a block matrix as follows:

$$A = \begin{bmatrix} A^{(1)} & 0_{m_1\times m_2} & 0_{m_1\times m_3} \\ 0_{m_2\times m_1} & A^{(2)} & 0_{m_2\times m_3} \\ 0_{m_3\times m_1} & 0_{m_3\times m_2} & A^{(3)} \end{bmatrix}. \tag{2.5}$$

Here, all the elements of the three matrices $A^{(1)} \in \mathbb{R}^{m_1\times m_1}$, $A^{(2)} \in \mathbb{R}^{m_2\times m_2}$ and $A^{(3)} \in \mathbb{R}^{m_3\times m_3}$ are strictly positive (that is, all their elements are strictly positive) and we have that $m_1 + m_2 + m_3 = m$. From now on in this section, to avoid verbosity, we will omit the subscripts of the 0 matrices.

It follows from the definitions that the normalized Laplacian L and the diagonal matrix D can be expressed as block matrices in a similar way:

$$D = \begin{bmatrix} D^{(1)} & 0 & 0 \\ 0 & D^{(2)} & 0 \\ 0 & 0 & D^{(3)} \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} L^{(1)} & 0 & 0 \\ 0 & L^{(2)} & 0 \\ 0 & 0 & L^{(3)} \end{bmatrix}, \tag{2.6}$$

where $D^{(1)} \in \mathbb{R}^{m_1\times m_1}$, $D^{(2)} \in \mathbb{R}^{m_2\times m_2}$ and $D^{(3)} \in \mathbb{R}^{m_3\times m_3}$ are themselves diagonal matrices and $L^{(1)} \in \mathbb{R}^{m_1\times m_1}$, $L^{(2)} \in \mathbb{R}^{m_2\times m_2}$ and $L^{(3)} \in \mathbb{R}^{m_3\times m_3}$ are strictly positive normalized Laplacians of each element of the partition $(\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3)$. Here, the following relation holds for each $i \in \{1, 2, 3\}$:

$$L^{(i)} = \left(D^{(i)}\right)^{-1/2} A^{(i)} \left(D^{(i)}\right)^{-1/2}. \tag{2.7}$$

Since the matrix L is block-diagonal, its set of eigenvalues σ_L is given by the union of the set of eigenvalues of each block L_1, L_2 and L_3, respectively σ_{L_1}, σ_{L_2} and σ_{L_3}. Furthermore its eigenvectors are the same as the ones of L_1, L_2 and L_3, provided that they are "extended" with 0 elements as necessary. The proof of this claim may also be found in [3].

By Proposition 2.3.1 on the preceding page, we know that each $L^{(i)}$ ($i \in \{1, 2, 3\}$) has 1 as an eigenvalue with positive eigenvector, which we denote by $x_1^{(i)} \in {\mathbb{R}_{>0}}^{m_i \times 1}$. Furthermore, all other eigenvalues of each $L^{(i)}$ are smaller than 1. This implies that L has 1 as an eigenvalue with multiplicity 3. Let X be the matrix containing the eigenvectors associated with these eigenvalues as columns. We have that

$$X = \begin{bmatrix} x_1^{(1)} & 0 & 0 \\ 0 & x_1^{(2)} & 0 \\ 0 & 0 & x_1^{(3)} \end{bmatrix} \in \mathbb{R}^{m \times 3}. \tag{2.8}$$

However, from elementary linear algebra, we know that for a Hermitian matrix if v_1 and v_2 are two eigenvectors associated with a certain eigenvalue, so is $\alpha v_1 + \beta v_2$, for all $(\alpha, \beta) \in \mathbb{R}^2$. Since the normalized Laplacian is Hermitian, we could have picked any other three eigenvectors spanning the same subspace as the ones above. The actual eigenvectors we obtain may depend on the small perturbations in the normalized Laplacian and the eigensolver used. This means that we could have gotten XR instead of X, for any orthogonal matrix $R \in \mathbb{R}^{3 \times 3}$. Therefore, we make the transformation $X \longmapsto XR$ to the matrix above in our analysis.

By normalizing the rows of the matrix X, we construct the matrix $Y \in \mathbb{R}^{m \times 3}$ as follows:

$$Y = \begin{bmatrix} 1_{m_1 \times 1} & 0 & 0 \\ 0 & 1_{m_2 \times 1} & 0 \\ 0 & 0 & 1_{m_3 \times 1} \end{bmatrix} R. \tag{2.9}$$

If we let ${R_1}^T \in \mathbb{R}^{1 \times 3}$, ${R_2}^T \in \mathbb{R}^{1 \times 3}$ and ${R_3}^T \in \mathbb{R}^{1 \times 3}$ represent the rows of the matrix R, equation 2.9 tells us that the i-th row of Y is given by ${R_j}^T$, where $i \in [\![1, m]\!]$, $j \in \{1, 2, 3\}$ and $d_i \in \mathcal{P}_j$ (i.e. the label of the i-th data instance is j).

As a result, the rows of the matrix Y related to the same label i will cluster in the same point ${R_i}^T$. Furthermore, from the fact that R is an orthogonal matrix, we deduce that rows of Y corresponding to different labels will cluster in points (located in the unit sphere) that are perpendicular to each other. This permits us to use the rows of the matrix Y to easily recover the labels of each $d_i \in \mathcal{D}$, with $i \in [\![1, m]\!]$, by e.g. applying the k-means algorithm to these rows.

Needless to say, most matrices we deal with are not in the ideal form we assumed they were in this section's discussion. However, we can think of a general matrix A as being of the norm $A = A_{\text{ideal}} + E$, where A_{ideal} is a matrix in the ideal form we discussed in this section and E represents the perturbation from the ideal case. As long as the norm of E is small enough, it is possible to prove that a spectral algorithm based on the approach of this section works. A more detailed description of the approach used in this section can be found in [4].

2.4 Relaxation approach

In this section we will derive the same spectral clustering algorithm as we did in the last section by framing the clustering problem as a discrete optimization problem and relaxing it so it is not discrete anymore. Before doing that, however, we need to give some definitions and outline some preliminary results.

2.4.1 Background

Definition Let $G = (V, E)$ be a graph of order m with adjacency matrix $A \in \mathbb{R}^{m \times m}$, and let C be a proper subset of the set of vertices V. Set $\overline{C} = V \setminus C$. The *cut* of the subset C is defined as follows:

$$\operatorname{cut}(C) = \sum_{\substack{v_i \in C \\ v_j \notin \overline{C}}} A_{ij}. \tag{2.10}$$

When multiple graphs are under discussion, there are cases in which we write the name of the graph considered as in $\operatorname{cut}_G(C)$ to make things clearer.

The cut of a set of vertices C is a measure of how much the elements of C are connected with the vertices of $\overline{C}$. For that reason, it is minimized when C is a separated component. One may try, then, to conduct clustering by minimizing the cut of the several elements of a partition of V. However, a problem with this idea is that an eventual algorithm trying to achieve this objective might minimize the cut by separating individual vertices from the rest of the graph, which is not what we desire. A possible approach to deal with this complication is to normalize the cut in such a way that small clusters are "penalized". This leads to the next definition.

Definition Let $G = (V, E)$ be a graph of order m with adjacency matrix $A \in \mathbb{R}^{m \times m}$, and let $(C_1, C_2, \cdots, C_k)$, where $k \in [\![1, m]\!]$, be a partition of the set of vertices V. The *normalized cut* of the partition $(C_1, C_2, \cdots, C_k)$ is defined as the following quantity:

$$\operatorname{Ncut}(C_1, C_2, \cdots, C_k) = \sum_{i=1}^{k} \frac{\operatorname{cut}(C_i)}{\operatorname{vol}(C_i)}. \tag{2.11}$$

Here, $\operatorname{vol}(C_i)$ denotes the sum of the degrees of all the vertices $v \in C_i$ for each $i \in [\![1, k]\!]$.

With this definition, we can think of the objective of spectral clustering as follows: given a graph $G = (V, E)$, and the number of clusters k, we wish to find a partition $(C_1, C_2, \cdots, C_k)$ of V such that $\operatorname{Ncut}(C_1, C_2, \cdots, C_k)$ is minimized. Unfortunately, this discrete optimization problem cannot be solved efficiently by brute force (more on this later). Therefore we will show how to derive a way of minimizing this quantity for $k = 2$ by relaxation. For the general case, the reader may consult [6].

In the following proposition, we will show a useful form for expressions of the type $x^T L x$, where L is a Laplacian matrix and x is a real vector. As we will see later, this will come handy when we want to find relationships between Laplacian matrices and the normalized cut of certain partitions.

Proposition 2.4.1 *Let $G = (V, E)$ be a graph of order m with adjacency matrix $A \in \mathbb{R}^{m\times m}$, and let L_0 be the unnormalized Laplacian matrix associated with G. Let $x \in \mathbb{R}^{m\times 1}$ be a real vector. Furthermore, for each $i \in [\![1, m]\!]$, let d_i denote the degree of the vertex v_i. Then we have*

$$x^T L_0 x = \frac{1}{2} \sum_{i,j=1}^{m} A_{ij} \left(x_i - x_j\right)^2 . \tag{2.12}$$

Proof

$$\begin{aligned} x^T L_0 x &= x^T D x - x^T A x \\ &= \sum_{i=1}^{m} d_i {x_i}^2 - \sum_{i,j=1}^{m} x_i x_j A_{ij} \\ &= \frac{1}{2} 2 \sum_{i=1}^{m} \left(\sum_{j=1}^{m} A_{ij} \right) {x_i}^2 - \frac{1}{2} \sum_{i,j=1}^{m} 2 x_i x_j A_{ij} \\ &= \frac{1}{2} \sum_{i,j=1}^{m} (x_i^2 + x_j^2 - 2 x_i x_j) A_{ij} \\ &= \frac{1}{2} \sum_{i,j=1}^{m} A_{ij} \left(x_i - x_j\right)^2 \end{aligned}$$

The proposition above allows us to say the following:
Corollary 2.4.2 *The unnormalized Laplacian L_0 of a graph G has the following properties:*

1. *It is positive semidefinite.*
2. *The vector $1_{m\times 1}$ is one of its eigenvectors with corresponding eigenvalue 0.*
3. *Thus its eigenvalues can be written as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m = 0$.*

And here we prove the generalization of the result above.

Proposition 2.4.3 *Let $G = (V, E)$ be a graph of order m with no isolated vertices and with similarity matrix $A \in \mathbb{R}^{m\times m}$, let L be the normalized Laplacian matrix associated with G, and let $x \in \mathbb{R}^{m\times 1}$ be a real vector. Furthermore, for each $i \in [\![1, m]\!]$, let d_i denote the degree of the vertex v_i. Then we have*

$$x^T L x = \frac{1}{2} \sum_{i,j=1}^{m} A_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 . \tag{2.13}$$

Proof

$$
\begin{aligned}
x^T L x &= x^T D^{-1/2} L_0 D^{-1/2} x \\
&= x^T x - x^T D^{-1/2} A D^{-1/2} x \\
&= \sum_{i=1}^{m} {x_i}^2 - \sum_{i,j=1}^{m} x_i x_j \frac{A_{ij}}{\sqrt{d_i d_j}} \\
&= \frac{1}{2}\left(\sum_{i=1}^{m} {x_i}^2 - 2 \sum_{i,j=1}^{m} \frac{x_i}{\sqrt{d_i}} \frac{x_j}{\sqrt{d_j}} A_{ij} + \sum_{j=1}^{m} {x_j}^2 \right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{m} \left(\frac{x_i}{\sqrt{d_i}} \right)^2 d_i - 2 \sum_{i,j=1}^{m} \frac{x_i}{\sqrt{d_i}} \frac{x_j}{\sqrt{d_j}} A_{ij} + \sum_{j=1}^{m} \left(\frac{x_j}{\sqrt{d_j}} \right)^2 d_j \right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{m} \left(\frac{x_i}{\sqrt{d_i}} \right)^2 \left(\sum_{j=1}^{m} A_{ij} \right) - 2 \sum_{i,j=1}^{m} \frac{x_i}{\sqrt{d_i}} \frac{x_j}{\sqrt{d_j}} A_{ij} + \sum_{j=1}^{m} \left(\frac{x_j}{\sqrt{d_j}} \right)^2 \left(\sum_{i=1}^{m} A_{ij} \right) \right) \\
&= \frac{1}{2} \sum_{i,j=1}^{m} \left(\left(\frac{x_i}{\sqrt{d_i}} \right)^2 - 2 \frac{x_i}{\sqrt{d_i}} \frac{x_j}{\sqrt{d_j}} + \left(\frac{x_j}{\sqrt{d_j}} \right)^2 \right) A_{ij} \\
&= \frac{1}{2} \sum_{i,j=1}^{m} A_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 . \quad \blacksquare
\end{aligned}
$$

The proposition above allows us to say the following:

Corollary 2.4.4 *The normalized Laplacian L of a graph G has the following properties:*

1. *It is positive semidefinite.*
2. *The vector $D^{1/2}\, 1_{m\times 1}$ is one of its eigenvectors with corresponding eigenvalue 0.*
3. *Thus its eigenvalues can be written as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m = 0$.*

Finally, before entering the second derivation of spectral clustering proper, we need to state two theorems which relate optimization of expressions of the form $x^T M x$ and eigenvalues. These theorems are collectively known as *Courant-Fischer Min-Max Theorems*. It is worthy to note here that the Propositions 2.4.1 and 2.4.3 on the preceding page are also important because, as we will see next, the generalized Courant-Fischer Min-Max Theorem requires that one of the matrices concerned be positive semidefinite.

Theorem 2.4.5 *Let m denote a positive integer, let $M \in \mathbb{C}^{m\times m}$ be a Hermitian matrix and denote its eigenvalues by $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. Assume U and V denote linear subspaces of $\mathbb{C}^{m\times 1}$. Then for all $k \in [\![1, m]\!]$ the following holds:*

$$
\lambda_k = \min_{\dim(U)=k} \max_{\substack{x\in U \\ x\neq 0_{m\times 1}}} \frac{x^H M x}{x^H x} = \max_{\dim(V)=m-k+1} \min_{\substack{x\in V \\ x\neq 0_{m\times 1}}} \frac{x^H M x}{x^H x}. \tag{2.14}
$$

Theorem 2.4.6 *Let m denote a positive integer, let $M \in \mathbb{C}^{m\times m}$ be a Hermitian matrix and $N \in \mathbb{C}^{m\times m}$ be a Hermitian positive semidefinite matrix such that $\mathcal{N}(N) \subseteq \mathcal{N}(M)$. Assume U and V denote linear subspaces of $\mathbb{C}^{m\times 1}$. Denote by r the rank of matrix T and by $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$ the generalized eigenvalues of the pencil (M, N). Then for all $k \in [\![1, r]\!]$ the following holds:*

$$\lambda_k = \min_{\substack{\dim U = k \\ U \perp \mathcal{N}(N)}} \max_{x\in U} \frac{x^H M x}{x^H N x} = \max_{\substack{\dim V = r-k+1 \\ V \perp \mathcal{N}(N)}} \min_{x\in V} \frac{x^H M x}{x^H N x}. \tag{2.15}$$

A proof of these theorems can be found in [1].

2.4.2 Derivation

Let m and n be positive integers. Consider the dataset $\mathcal{D} \subseteq \mathbb{R}^{n\times 1}$ and its associated graph $G = (V, E)$ with similarity matrix $A \in \mathbb{R}^{m\times m}$. Let C be a proper subset of V, and let $D \in \mathbb{R}^{m\times m}$ be the diagonal matrix such that D_{ii} is equal to the degree of $v_i \in V$ for all $i \in [\![1, m]\!]$. Our objective is to set C such that

$$\operatorname{Ncut}(C, \overline{C}) \tag{2.16}$$

is minimized. A proof that this optimization problem is NP-complete may be found at [5]. Therefore, we need another approach in order to perform clustering in a graph by minimizing the normalized cut.

Definition The indicator vector $x_C \in \mathbb{R}^{m\times 1}$ is defined by:

$$(x_C)_i = \begin{cases} \sqrt{\operatorname{vol}\left(\overline{C}\right) / \operatorname{vol}\left(C\right)} & \text{if } v_i \in C \\ -\sqrt{\operatorname{vol}\left(C\right) / \operatorname{vol}\left(\overline{C}\right)} & \text{if } v_i \in \overline{C} \end{cases} \tag{2.17}$$

for each $i \in [\![1, m]\!]$.

Our goal here is to find a relationship between ${x_C}^T L x_C$ and the normalized cut of A. Before that, consider the following lemmas:

Lemma 2.4.7 *The following holds:*

$$(Dx_C)^T 1_{m\times 1} = 0. \tag{2.18}$$

Proof Let d_i denote the degree of the vertex v_i for each $i \in [\![1, m]\!]$. We have, then:

$$\begin{aligned}
(Dx_C)^T 1_{m\times 1} &= \sum_{i=1}^{m} d_i \cdot (x_C)_i \\
&= \sum_{v_i \in C} d_i \cdot (x_C)_i + \sum_{v_i \in \overline{C}} d_i \cdot (x_C)_i \\
&= \sum_{v_i \in C} d_i \sqrt{\operatorname{vol}\left(\overline{C}\right) / \operatorname{vol}\left(C\right)} - \sum_{v_i \in \overline{C}} d_i \sqrt{\operatorname{vol}\left(C\right) / \operatorname{vol}\left(\overline{C}\right)} \\
&= \operatorname{vol}\left(C\right) \sqrt{\operatorname{vol}\left(\overline{C}\right) / \operatorname{vol}\left(C\right)} - \operatorname{vol}\left(\overline{C}\right) \sqrt{\operatorname{vol}\left(C\right) / \operatorname{vol}\left(\overline{C}\right)} \\
&= 0. \quad \blacksquare
\end{aligned}$$

Lemma 2.4.8 *The following holds:*

$$x_C{}^T D x_C = \mathrm{vol}\,(V). \tag{2.19}$$

Proof As in the lemma above, let d_i denote the degree of the vertex v_i for each $i \in [\![1\,,\,m]\!]$. We have:

$$\begin{aligned}
x_C{}^T D x_C &= \sum_{i,j=1}^{m} D_{ij} \cdot (x_C)_i \cdot (x_C)_j \\
&= \sum_{i=1}^{m} d_i \cdot (x_C)_i{}^2 \\
&= \sum_{v_i \in C} d_i \left(\sqrt{\frac{\mathrm{vol}\,(\overline{C})}{\mathrm{vol}\,(C)}} \right)^2 + \sum_{v_i \in \overline{C}} d_i \left(\sqrt{\frac{\mathrm{vol}\,(C)}{\mathrm{vol}\,(\overline{C})}} \right)^2 \\
&= \mathrm{vol}\,(C) \frac{\mathrm{vol}\,(\overline{C})}{\mathrm{vol}\,(C)} + \mathrm{vol}\,(\overline{C}) \frac{\mathrm{vol}\,(C)}{\mathrm{vol}\,(\overline{C})} \\
&= \mathrm{vol}\,(V). \qquad \blacksquare
\end{aligned}$$

And here, finally, we relate $x_C{}^T L x_C$ and the normalized cut.

Theorem 2.4.9 *The following holds:*

$$x_C{}^T L_0 x_C = \mathrm{vol}\,(V)\ \mathrm{cut}\,(A, \overline{A}). \tag{2.20}$$

Proof We already know that

$$x_C{}^T L_0 x_C = \frac{1}{2} \sum_{i,j=1}^{m} A_{ij} ((x_C)_i - (x_C)_j)^2.$$

Since whenever $(v_i, v_j) \in C^2$ or $(v_i, v_j) \in \overline{C}^2$ (where $(i, j) \in [\![1\,,\,m]\!]^2$) we have that $(x_C)_i - (x_C)_j = 0$, we can write $x_C{}^T L_0 x_C$ as follows:

$$\begin{aligned}
x_C{}^T L_0 x_C &= \frac{1}{2} \sum_{v_i \in C, v_j \in \overline{C}} A_{ij} \left(\sqrt{\frac{\mathrm{vol}\,(\overline{C})}{\mathrm{vol}\,(C)}} + \sqrt{\frac{\mathrm{vol}\,(C)}{\mathrm{vol}\,(\overline{C})}} \right)^2 + \frac{1}{2} \sum_{v_i \in \overline{C}, v_j \in C} A_{ij} \left(-\sqrt{\frac{\mathrm{vol}\,(\overline{C})}{\mathrm{vol}\,(C)}} - \sqrt{\frac{\mathrm{vol}\,(C)}{\mathrm{vol}\,(\overline{C})}} \right)^2 \\
&= \mathrm{cut}\,(C) \left(\frac{\mathrm{vol}\,(\overline{C})}{\mathrm{vol}\,(C)} + \frac{\mathrm{vol}\,(C)}{\mathrm{vol}\,(\overline{C})} + 2 \right) \\
&= \mathrm{cut}\,(C) \left(\frac{\mathrm{vol}\,(C) + \mathrm{vol}\,(\overline{C})}{\mathrm{vol}\,(C)} + \frac{\mathrm{vol}\,(C) + \mathrm{vol}\,(\overline{C})}{\mathrm{vol}\,(\overline{C})} \right) \\
&= \mathrm{vol}\,(V) \left(\frac{\mathrm{cut}\,(C)}{\mathrm{vol}\,(C)} + \frac{\mathrm{cut}\,(\overline{C})}{\mathrm{vol}\,(\overline{C})} \right) \\
&= \mathrm{vol}\,(V)\ \mathrm{Ncut}\,(A, \overline{A}).
\end{aligned}$$

Here we have used the fact that $\mathrm{cut}\,(C) = \sum_{v_i \in C, v_j \in \overline{C}} A_{ij} = \sum_{v_i \in \overline{C}, v_j \in C} A_{ij} = \mathrm{cut}\,(\overline{C})$. $\blacksquare$

Considering that $\mathrm{vol}\,(V)$ is constant for a given graph, the objective function for clustering,

$$\underset{C}{\operatorname{argmin}}\,\mathrm{Ncut}\ (A, \overline{A}), \tag{2.21}$$

can be expressed as

$$\underset{x_C \in \mathbb{R}^{m\times 1}}{\operatorname{argmin}}\ {x_C}^T L_0 x_C, \tag{2.22}$$

where x_C is defined as in equation 2.17 on page 10.

As discussed before, this is a NP-complete problem. To deal with this issue, we may try to relax the condition that x_C is a indicator vector and treat it as a normal vector in $\mathbb{R}^{m\times 1}$. However, in order not to lose too much information from the optimization constraints, we should also incorporate the two conditions that x_C obeys given by Lemma 2.4.7 and Lemma 2.4.8 on the preceding page in our new constraint. We get, then:

$$\underset{x\in\mathbb{R}^{m\times 1}}{\operatorname{argmin}} x^T L_0 x \ \text{ subject to } (Dx) \perp 1_{m\times 1} \text{ and } x^T D x = \mathrm{vol}\,(V). \tag{2.23}$$

In order to put the constraining problem above in the form given by Courant-Fischer Min-Max Theorem, we can make the substitution $y = D^{1/2}x$ and obtain

$$\underset{y\in\mathbb{R}^{m\times 1}}{\operatorname{argmin}} y^T L y \ \text{ subject to } y \perp (D^{1/2}\, 1_{m\times 1}) \text{ and } y^T y = \mathrm{vol}\,(V). \tag{2.24}$$

Using Theorem 2.4.5 on page 9 for $k = 2$ we know that the second biggest eigenvalue of the matrix L satisfies:

$$\lambda_2 = (1/\,\mathrm{vol}\,(V)) \max_{\dim(V)=m-1} \min_{\substack{y\in V \\ y\perp D^{1/2}1_{m\times 1}}} y^T L y. \tag{2.25}$$

Furthermore, knowing that y is perpendicular to the eigenvector corresponding to the eigenvalue $\lambda_1 = 0$, the eigenvector corresponding to the second largest eigenvalue of L is the solution to the optimization problem given by equation 2.25 and consequently the one given by equation 2.24.

Clearly, obtaining y and consequently x does not give us C immediately. However, we can consider the coordinates of $x \in \mathbb{R}^{m\times 1}$ as points in $\mathbb{R}$, use k-means to cluster them and recover C.

Chap. 3 Spectral Clustering with the Bethe Hessian

Chap. 4 Constrained Spectral Clustering with FAST-GE2

4.1 Introduction

The Information Age has brought with it large incentives to organize and process big amounts of data. Traditionally, two main approaches have been used to deal with this task: classification and clustering. While classification is widely used in situations where training data is abundant, such as recommendation systems, spam detection and speech recognition, this class of methods is not applicable to unlabeled datasets, which have been traditionally handled by clustering algorithms. However, since clustering only makes use of the internal structure of the data, our control over the process is limited. In this context, a new class of semi-supervised algorithms known as constrained clustering has appeared. While these methods do not demand large amounts of labeled data as inputs, they still make it possible for a small amount of training data to influence the final outcome of the clustering process. In this chapter, we describe FAST-GE-2.0, a spectral way of performing constrained clustering and see the theory behind its correctness. This chapter is mainly a survey of the results from [2], although we have changed some of the presentation and notation as to make them fit better with the rest of this thesis.

4.2 Constrained Clustering

In this chapter, m, n, and k represent positive integers.

Given a dataset $\mathcal{D} \subseteq \mathbb{R}^{n\times 1}$ (or equivalently a weighted graph $G = (V, E)$ of order m) and a set of contraints, to perform constrained clustering on the data means to find a partition $(C_1, C_2, \cdots, C_k)$ of V such that:

- For all $i \in [\![1\,,\,k]\!]$, edges of vertices in the same subset C_i have big weights.
- For all $(i, j) \in [\![1\,,\,k]\!]^2$, edges of vertices in different subsets C_i and C_j have small weights.
- Constraints are followed as much as possible.

These constraints are usually small in number and represent whether certain groups of vertices should forcibly stay together or forcibly stay apart. For example, in image segmentation, one of the main applications of constrained spectral clustering, a user selects a small amount of points in an image that she believes should stay in the same segment (e.g. points of a uniform backgroud, or points of a tree).

Then the contrained clustering algorithm tries to divide the image in segments (clusters) such that the points selected by the user stay in the same segment.

There are several ways of representing these constraints, each leading to different algorithms. In this thesis we will work with must-link constraints (ML) and cannot-link constraints (CL) encoded as follows:

A set of constraints is given by p disjoint subjects of V,

$$\{V_1, V_2, \cdots, V_k\} \subseteq V, \tag{4.1}$$

such that: (1) for all $i \in [\![1, k]\!]$, if $(u, v) \in {V_i}^2$ then there exists a ML contraints between the vertices u and v; and (2) for all $(i, j) \in [\![1, k]\!]^2$, if $(u, v) \in V_i \times V_j$ and $i \neq j$ then there exists a CL contraint between the vertices u and v.

An algorithm we may eventually develop, then, must be set up in such a way that violations of ML and CL constraints (such as, e.g., two vertices in different constraint sets V_1 and V_2 being in the same cluster C_1) have a negative effect on its effort to satisfy the objective function.

4.3 FAST-GE-2.0

We will now discuss FAST-GE-2.0, a spectral algorithm proposed by Chengming Jiang, et al, for constrained clustering in [2]. We are given a fully connected graph $G = (V, E)$ of order m with adjacency matrix A. Assume a set of constraints $\{V_1, V_2, \cdots, V_k\}$ is given. The objective of FAST-GE-2.0, in line with our discussion in the last section, is to find a partition $(C_1, C_2, \cdots, C_k)$ of V such that $V_i \subseteq C_i$ for all $i \in [\![1, k]\!]$, where, for each pair $(i, j) \in [\![1, m]\!]^2$, vertices $(u, v) \in {C_i}^2$ have edges with high weight and vertices $(u, v) \in C_i \times C_j$, with $i \neq j$ have edges with low weight. FAST-GE-2.0 manages to satisfy these constraints indirectly by using auxiliary graphs and encoding the ML and CL constraints into the Laplacian matrices dealth with in the algorithm.

4.3.1 Auxiliary graphs

In this subsection we define the auxiliary graphs used in FAST-GE-2.0.

The graph G_M

Definition The graph $G_M = (V, E)$ is defined by its adjacency matrix

$$A_M = \sum_{\ell=1}^{k} A_{M_\ell}, \tag{4.2}$$

where, for each $(\ell, i, j) \in [\![1, k]\!] \times [\![1, m]\!]^2$, the entries of the submatrix $A_{M_\ell} \in \mathbb{R}^{m \times m}$ are given by:

$$(A_{M_\ell})_{ij} = \begin{cases} (d_i d_j)/(d_{\min} d_{\max}) & \text{if } (v_i, v_j) \in {V_\ell}^2 \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

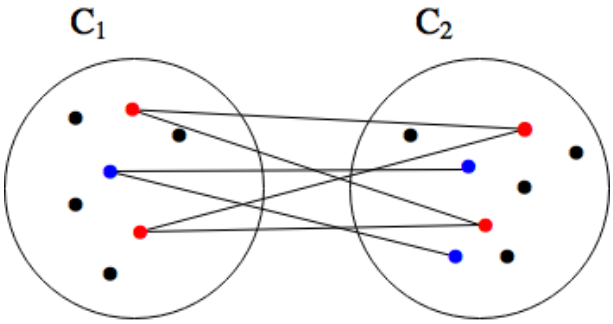

Figure 4.1: Graphical representation of $\operatorname{cut}_{G_M}(C_1)$ for a simple simple graph $G = (V, E)$. In this example, red vertices are elements of V_1, blue vertices are elements of V_2 and black vertices are elements of $V \setminus (V_1 \cup V_2)$. The black lines represent elements of A_M that contribute to the amount of violations of ML given by $\operatorname{cut}_{G_M}(C_1)$. Notice that all non-zero elements of A_M must connect vertices of the same color, and all terms contributing to $\operatorname{cut}_{G_M}(C_1)$ must connect vertices in different clusters; hence the lines in the figure. We want elements of the same color to stay in the same cluster as much as possible. Therefore, we must try to decrease the amount of black lines.

Here, for each $i \in [\![1, k]\!]$, d_i represents the degree of the vertex v_i. Futhermore, $d_{\min}$ and $d_{\max}$ represent the smallest and biggest element of the set $\{d_i\}_{i=1}^{k}$, respectively.

As shown in Figure 4.1, if we define G_M as above, for any given $\ell \in [\![1, k]\!]$, the quantity

$$\operatorname{cut}_{G_M}(C_\ell) = \sum_{\substack{v_i \in C_\ell \\ v_j \overline{C_\ell}}} (A_M)_{ij} \tag{4.4}$$

measures the amount of violations of ML constraints relative to the cut of C_ℓ. Therefore we must try to minimize it as much as possible.

The graph G_H

Chap. 5 Proposed Method

Chap. 6 Numerical Experiments

Acknowledgements

References

[1] Haim Avron, Esmond Ng, and Sivan Toledo. A generalized courant-fischer minimax theorem. 2008.

[2] Chengming Jiang, Huiqing Xie, and Zhaojun Bai. Robust and efficient computation of eigenvectors in a generalized spectral method for constrained clustering. In *Artificial Intelligence and Statistics*, pages 757–766, 2017.

[3] M. W. Mahoney. Lecture Notes on Spectral Graph Methods. *ArXiv e-prints*, August 2016.

[4] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2002.

[5] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[6] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.