Chapter 3

Graph Clustering

3.1 Graph Clustering Using Normalized Cuts

Given a set of data, the goal of clustering is to partition the data into different groups according to their similarities.

When the data is given in terms of a similarity graph G, where the weight $w_{i\,j}$ between two nodes v_i and v_j is a measure of similarity of v_i and v_j, the problem can be stated as follows:

Find a partition $(A_1, \ldots, A_K)$ of the set of nodes V into different groups such that the *edges between different groups have very low weight* (which indicates that the *points in different clusters are dissimilar*), and the *edges within a group have high weight* (which indicates that *points within the same cluster are similar*).

The above graph clustering problem can be formalized as an optimization problem, using the notion of *cut* mentioned at the end of Section 1.1.

Given a subset A of the set of vertices V, we define $\mathrm{cut}(A)$ by

$$\mathrm{cut}(A) = \mathrm{links}(A, \overline{A}) = \sum_{v_i \in A, v_j \in \overline{A}} w_{i\,j}.$$

If we want to partition V into K clusters, we can do so by finding a partition $(A_1, \ldots, A_K)$ that minimizes the quantity

$$\mathrm{cut}(A_1, \ldots, A_K) = \frac{1}{2} \sum_{i=1}^{K} \mathrm{cut}(A_i).$$

For $K = 2$, the mincut problem is a classical problem that can be solved efficiently, but in practice, it does not yield satisfactory partitions.

Indeed, in many cases, the mincut solution separates one vertex from the rest of the graph.

What we need is to design our cost function in such a way that it keeps the subsets A_i "reasonably large" (*reasonably balanced*).

An example of a weighted graph and a partition of its nodes into two clusters is shown in Figure 3.1.

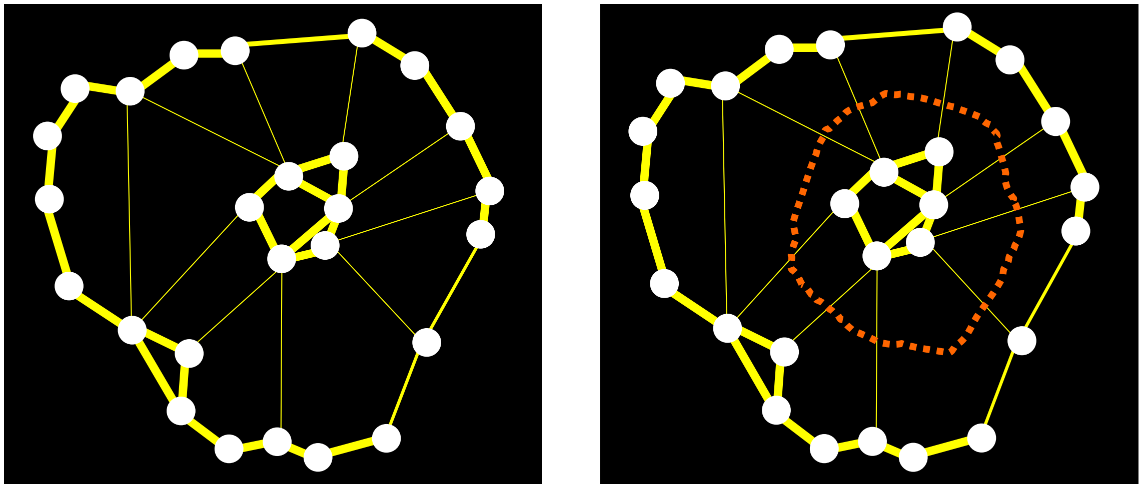

Figure 3.1: A weighted graph and its partition into two clusters.

A way to get around this problem is to *normalize the cuts* by dividing by some measure of each subset A_i.

One possibility is to use the *size* (the number of elements) of A_i.

Another is to use the *volume* $\mathrm{vol}(A_i)$ of A_i.

A solution using the second measure (the volume) (for $K = 2$) was proposed and investigated in a seminal paper of Shi and Malik [13].

Subsequently, Yu (in her dissertation [16]) and Yu and Shi [17] extended the method to $K > 2$ clusters.

The idea is to minimize the cost function

$$\mathrm{Ncut}(A_1, \ldots, A_K) = \sum_{i=1}^{K} \frac{\mathrm{links}(A_i, \overline{A_i})}{\mathrm{vol}(A_i)} = \sum_{i=1}^{K} \frac{\mathrm{cut}(A_i, \overline{A_i})}{\mathrm{vol}(A_i)}.$$

We begin with the case $K = 2$, which is easier to handle.

3.2 Special Case: 2-Way Clustering Using Normalized Cuts

Our goal is to express our optimization problem in matrix form.

In the case of two clusters, a single vector X can be used to describe the partition $(A_1, A_2) = (A, \overline{A})$.

We need to choose the structure of this vector in such a way that $\mathrm{Ncut}(A, \overline{A})$ is equal to the Rayleigh ratio

$$\frac{X^\top L X}{X^\top D X}.$$

It is also important to pick a *vector representation which is invariant under multiplication by a nonzero scalar*, because the Rayleigh ratio is scale-invariant, and it is crucial to take advantage of this fact to make the denominator go away.

Let $N = |V|$ be the number of nodes in the graph G.

In view of the desire for a scale-invariant representation, it is natural to assume that the vector X is of the form

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

where $x_i \in \{a, b\}$ for $i = 1, \ldots, N$, for any two distinct real numbers a, b.

This is an indicator vector in the sense that, for $i = 1, \ldots, N$,

$$x_i = \begin{cases} a & \text{if } v_i \in A \\ b & \text{if } v_i \notin A. \end{cases}$$

The correct interpretation is really to view X as a representative of a point in the *real projective space* $\mathbb{RP}^{N-1}$, namely the point $\mathbb{P}(X)$ of homogeneous coordinates $(x_1 \colon \cdots : x_N)$.

Therefore, from now on, we view X as a vector of homogeneous coordinates representing the point $\mathbb{P}(X) \in \mathbb{RP}^{N-1}$.

Let $d = \mathbf{1}^\top D\mathbf{1}$ and $\alpha = \mathrm{vol}(A)$. Then, $\mathrm{vol}(\overline{A}) = d - \alpha$.

Using Proposition 1.4, we have

$$\begin{aligned} X^\top LX &= (a-b)^2 \,\mathrm{cut}(A, \overline{A}) \\ X^\top DX &= \alpha a^2 + (d-\alpha)b^2. \end{aligned}$$

We obtain

$$\mathrm{Ncut}(A, \overline{A}) = \frac{d}{\alpha(d-\alpha)}\,\mathrm{cut}(A, \overline{A})$$

and

$$\frac{X^\top L X}{X^\top D X} = \frac{(a-b)^2}{\alpha a^2 + (d-\alpha)b^2}\,\mathrm{cut}(A, \overline{A}).$$

In order to have

$$\mathrm{Ncut}(A, \overline{A}) = \frac{X^\top L X}{X^\top D X},$$

we need to find a and b so that

$$\frac{(a-b)^2}{\alpha a^2 + (d-\alpha)b^2} = \frac{d}{\alpha(d-\alpha)}.$$

We find the condition

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

Note that condition ($\dagger$) applied to a vector X whose components are a or b is equivalent to the fact that X is orthogonal to $D\mathbf{1}$.

Various choices for a choice of scale factor appear in the literature.

von Luxburg [15] picks

$$a = \sqrt{\frac{d-\alpha}{\alpha}}, \quad b = -\sqrt{\frac{\alpha}{d-\alpha}}.$$

Shi and Malik [13] use

$$a = 1, \quad b = -\frac{\alpha}{d-\alpha} = -\frac{k}{1-k},$$

with

$$k = \frac{\alpha}{d}.$$

Belkin and Niyogi [2] use

$$a = \frac{1}{\alpha}, \quad b = -\frac{1}{d-\alpha}.$$

However, there is no need to restrict solutions to be of either of these forms.

So, let

$$\mathcal{X} = \big\{(x_1, \ldots, x_N) \mid x_i \in \{a, b\},\ a, b \in \mathbb{R},\ a, b \neq 0\big\},$$

so that our solution set is

$$\mathcal{K} = \big\{X \in \mathcal{X} \mid X^\top D\mathbf{1} = 0\big\},$$

Actually, to be perfectly rigorous, we are looking for solutions in $\mathbb{RP}^{N-1}$, so our solution set is really

$$\mathbb{P}(\mathcal{K}) = \big\{(x_1 \colon \cdots : x_N) \in \mathbb{RP}^{N-1} \mid (x_1, \ldots, x_N) \in \mathcal{K}\big\}.$$

Consequently, our minimization problem can be stated as follows:

Problem PNC1

$$
\begin{aligned}
&\text{minimize} && \frac{X^\top L X}{X^\top D X} \\
&\text{subject to} && X^\top D\mathbf{1} = 0, \qquad X \in \mathcal{X}.
\end{aligned}
$$

It is understood that the solutions are points $\mathbb{P}(X)$ in $\mathbb{RP}^{N-1}$.

Since the Rayleigh ratio and the constraints $X^\top D\mathbf{1} = 0$ and $X \in \mathcal{X}$ are scale-invariant we are led to the following formulation of our problem:

Problem PNC2

$$\begin{aligned} &\text{minimize} && X^\top L X \\ &\text{subject to} && X^\top D X = 1, \quad X^\top D\mathbf{1} = 0, \quad X \in \mathcal{X}. \end{aligned}$$

Problem PNC2 is equivalent to problem PNC1, in the sense that if X is any minimal solution of PNC1, then $X/(X^\top DX)^{1/2}$ is a minimal solution of PNC2 (with the same minimal value for the objective functions), and if X is a minimal solution of PNC2, then λX is a minimal solution for PNC1 for all $\lambda \neq 0$ (with the same minimal value for the objective functions).

Equivalently, problems PNC1 and PNC2 have the same set of minimal solutions as points $\mathbb{P}(X) \in \mathbb{RP}^{N-1}$ given by their homogeneous coordinates X.

Unfortunately, this is an NP-complete problem, as shown by Shi and Malik [13].

As often with hard combinatorial problems, we can look for a *relaxation* of our problem, which means looking for an optimum in a larger continuous domain.

After doing this, the problem is to find a discrete solution which is close to a continuous optimum of the relaxed problem.

The natural relaxation of this problem is to allow X to be any nonzero vector in $\mathbb{R}^N$, and we get the problem:

$$\begin{aligned} &\text{minimize} \quad X^\top L X \\ &\qquad\qquad\qquad \text{subject to} \quad X^\top D X = 1, \quad X^\top D\mathbf{1} = 0. \end{aligned}$$

In order to apply Proposition A.2, we make the change of variable $Y = D^{1/2}X$, so that $X = D^{-1/2}Y$.

We obtain the problem:

$$\begin{aligned} &\text{minimize} \quad Y^\top D^{-1/2} L D^{-1/2} Y \\ &\qquad\qquad\qquad \text{subject to} \quad Y^\top Y = 1, \quad Y^\top D^{1/2}\mathbf{1} = 0. \end{aligned}$$

Because $L\mathbf{1} = 0$, the vector $D^{1/2}\mathbf{1}$ belongs to the nullspace of the symmetric Laplacian $L_{\mathrm{sym}} = D^{-1/2}LD^{-1/2}$.

By Proposition A.2, minima are achieved by any unit eigenvector Y of the second eigenvalue $\nu_2 > 0$ of L_{sym}.

Then, $Z = D^{-1/2}Y$ is a solution of our original relaxed problem

Note that because Z is nonzero and orthogonal to $D\mathbf{1}$, a vector with positive entries, it must have negative and positive entries.

The next question is to figure how close is Z to an exact solution in $\mathcal{X}$.

Actually, because solutions are points in $\mathbb{RP}^{N-1}$, the correct statement of the question is: Find an exact solution $\mathbb{P}(X) \in \mathbb{P}(\mathcal{X})$ which is the closest (in a suitable sense) to the approximate solution $\mathbb{P}(Z) \in \mathbb{RP}^{N-1}$.

However, because $\mathcal{X}$ is closed under the antipodal map, it can be shown that minimizing the distance $d(\mathbb{P}(X), \mathbb{P}(Z))$ on $\mathbb{RP}^{N-1}$ is equivalent to minimizing the Euclidean distance $\|X - Z\|_2$, where X and Z are representatives of $\mathbb{P}(X)$ and $\mathbb{P}(Z)$ on the unit sphere (if we use the Riemannian metric on $\mathbb{RP}^{N-1}$ induced by the Euclidean metric on $\mathbb{R}^N$).

We may assume $b < 0$, in which case $a > 0$.

If all entries in Z are nonzero, due to the projective nature of the solution set, it seems reasonable to say that the partition of V is defined by the signs of the entries in Z.

Thus, A will consist of nodes those v_i for which $x_i > 0$.

Elements corresponding to zero entries can be assigned to either A or $\overline{A}$, unless additional information is available. In our implementation, they are assigned to A.

Here are some examples of normalized cuts found by a fairly naive implementation of the method.

The weight matrix of the first example is

$$W_1 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Its underlying graph has 9 nodes and 9 edges and is shown in Figure 3.2 on the left.

The normalized cut found by the algorithm is shown in the middle; the edge of the cut is shown in magenta, and the vertices of the blocks of the partition are shown in blue and red.

The figure on the right shows the two disjoint subgraphs obtained after deleting the cut edge.

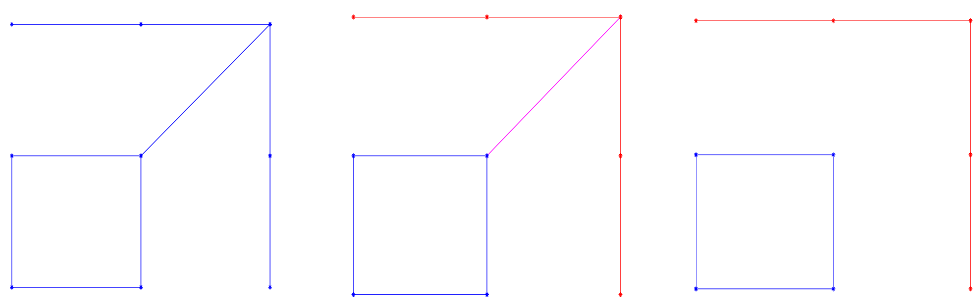

Figure 3.2: Underlying graph of the matrix W_1 (left); normalized cut (middle); blocks of the cut (right).

The weight matrix of the second example is

$$W_2 = \begin{pmatrix} 0 & 3 & 6 & 3 \\ 3 & 0 & 0 & 3 \\ 6 & 0 & 0 & 3 \\ 3 & 3 & 3 & 0 \end{pmatrix}.$$

Its underlying graph has 4 nodes and 5 edges and is shown in Figure 3.3 on the left.

The normalized cut found by the algorithm is shown in the middle; the edges of the cut are shown in magenta, and the vertices of the blocks of the partition are shown in blue and red.

The figure on the right shows the two disjoint subgraphs obtained after deleting the cut edges.

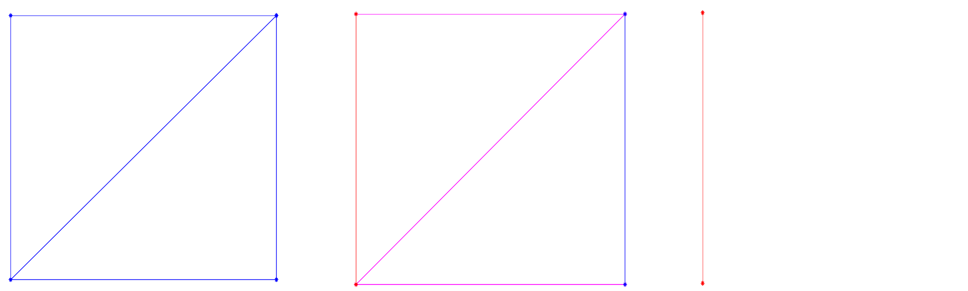

Figure 3.3: Underlying graph of the matrix W_2 (left); normalized cut (middle); blocks of the cut (right).

The weight matrix W_3 of the third example is the adjacency matrix of the complete graph on 12 vertices.

All nondiagonal entries are equal to 1, and the diagonal entries are equal to 0.

This graph has 66 edges and is shown in Figure 3.4 on the left.

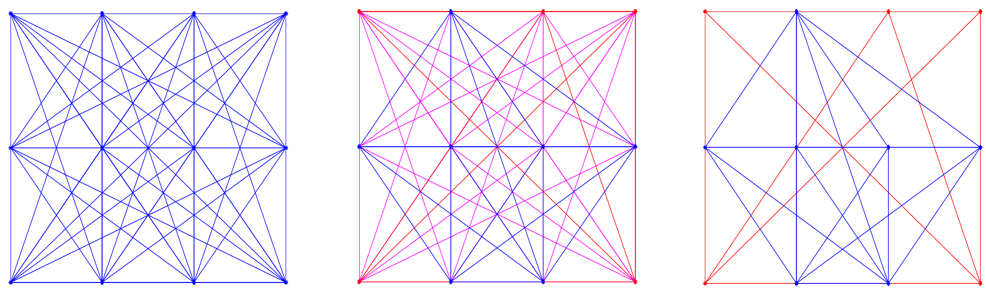

Figure 3.4: Underlying graph of the matrix W_3 (left); normalized cut (middle); blocks of the cut (right).

The normalized cut found by the algorithm is shown in the middle; the edges of the cut are shown in magenta, and the vertices of the blocks of the partition are shown in blue and red.

The figure on the right shows the two disjoint subgraphs obtained after deleting the cut edges.

Our naive algorithm treated zero as a positive entry. Now, using the fact that

$$b = -\frac{\alpha a}{d - \alpha},$$

a better solution is to look for a vector $X \in \mathbb{R}^N$ with $X_i \in \{a, b\}$ which is closest to a minimum Z of the relaxed problem (in the sense that $\|X - Z\|$ is minimized) and with $\|X\| = \|Z\|$.

I implemented such an algorithm, and it seems to do a god job dealing with zero entries in the continuous solution Z.