SPECTRAL CLUSTERING

Notes

Matteo Lugli

1 Intuition

Spectral clustering algorithms all rely on some simple steps:

1. Represent the non-convex space using a graph;
2. Compute the adjacency matrix and the degree matrix of that graph;
3. Compute the Laplacian by doing $L = A - D$;
4. Compute the eigenvector and the eigenvalues of the matrix L;
5. Take the first k eigenvectors (from the smallest corresponding eigenvalue) except for the first one, where k is the dimension of the new space;
6. The eigenvectors are going to be the features of the elements of the new space, so they become the columns of a new matrix P;
7. Use k-means on the matrix P to find the clusters;

But why do we use the Laplacian matrix? Why do we compute the eigenvectors of such matrix?

2 Example

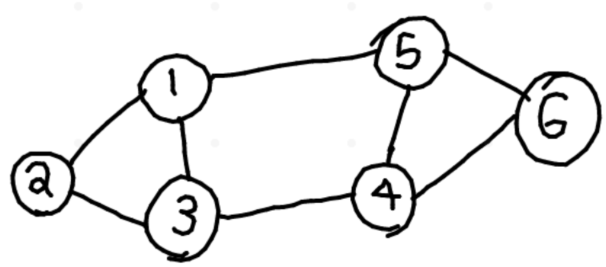

Figure 1: example graph

nodes	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	0	0	0	-1	3	-1
6	0	0	0	-1	-1	2

Above you can see an example graph and the corresponding Laplacian. If you notice, the sum of all of the rows is always zero. You can easily verify that one possible eigenvector is $[1, 1, \ldots, 1]$ with the corresponding eigenvalue being $\lambda = 0$. Also remember that all of the eigenvectors are always real and orthogonal.

The min-max theorem (read Wikipedia page) says that finding the smallest eigenvalue is the same as solving the following equation:

$$\lambda_2 = min \frac{x^t M x}{x^t x} \tag{1}$$

Let's see if plugging the Laplacian in that equation produces something usefull that can relate to our clustering problem:

$$x^T L x = \sum_{i,j=1}^{n} L_{i,j} x_i x_j = \sum_{i,j=1}^{n} (D_{i,j} - A_{i,j}) x_i x_j \tag{2}$$

$$= \sum_i D_{ii} x_i^2 - \sum_{i,j \in E} 2 x_i x_j \tag{3}$$

Going from 2 to 3 is not trivial: you need to consider that D is a diagonal matrix, so when you successfully select an element that is $\neq$ from zero in matrix D it means that i and j are the same. For the second part you need to consider that the in matrix A you find a 1 if node i and node j are connected. So we can replace the original loop with a loop that iterates on the edges. The matrix is symmetrical so we encounter the product $x_i x_j$ twice through the loop.

$$= \sum_i D_{ii} x_i^2 - \sum_{i,j \in E} 2 x_i x_j$$

$$= \sum_{i,j \in E} x_i^2 + x_j^2 - 2 x_i x_j \tag{4}$$

$$= \sum_{i,j \in E} (x_i - x_j)^2 \tag{5}$$

Let's recall that x is a vector that "selects" some of the elements and place them in cluster A, the other ones will be placed in cluster B. Let's re-write the full problem and make some considerations:

$$\lambda_2 = \min \frac{\sum_{i,j \in E} (x_i - x_j)^2}{\sum_i x_i^2} \tag{6}$$

- because x is an eigenvector and L is a Hermitian matrix, his length is 1, so $\sum_i x_i^2 = 1$;
- because eigenvector x in orthogonal to the first eigenvector that we found in the previous example, then $\sum_i x_i \cdot 1 = \sum_i x_i = 0$. This means that our function will place some nodes on one cluster and the some nodes in the other cluster (if you think of labels as 1 and -1).

3 Summary

We learned that finding the eigenvector corresponding to the smallest eigenvalue is the same as solving this:

$$\text{argmin}_{y \in [-1,1]^n} f(y) = \sum_{i,j \in E} (y_i - y_j)^2 = y^t L y \tag{7}$$

which looks really similar to the clustering problem that we want to solve. This particular formulation is obtained because we used the Laplacian matrix!