Spectral Clustering

Parsa Tasbihgou

Prof. Sajedi

Department of Mathematics and

Computer Science, University of Tehran

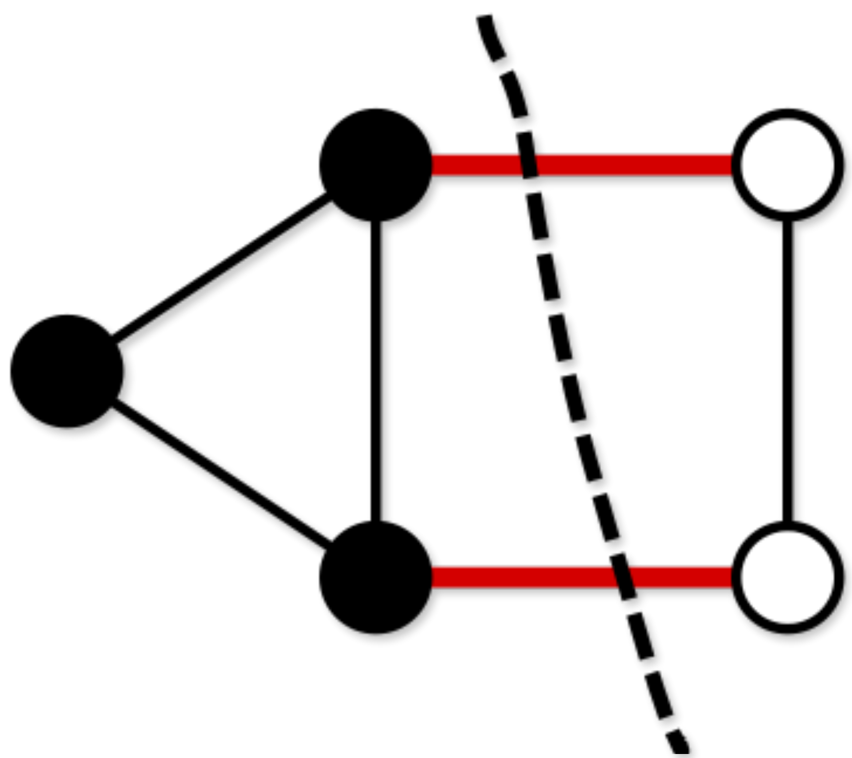
May 20, 2022

Graph partitioning

- Clustering a graph can be viewed as partitioning graph nodes into connected components. Doing so requires making “cuts” to separate current connected components.

- A “cut” is a subset of edges, which removing them will increase the number of connected components.

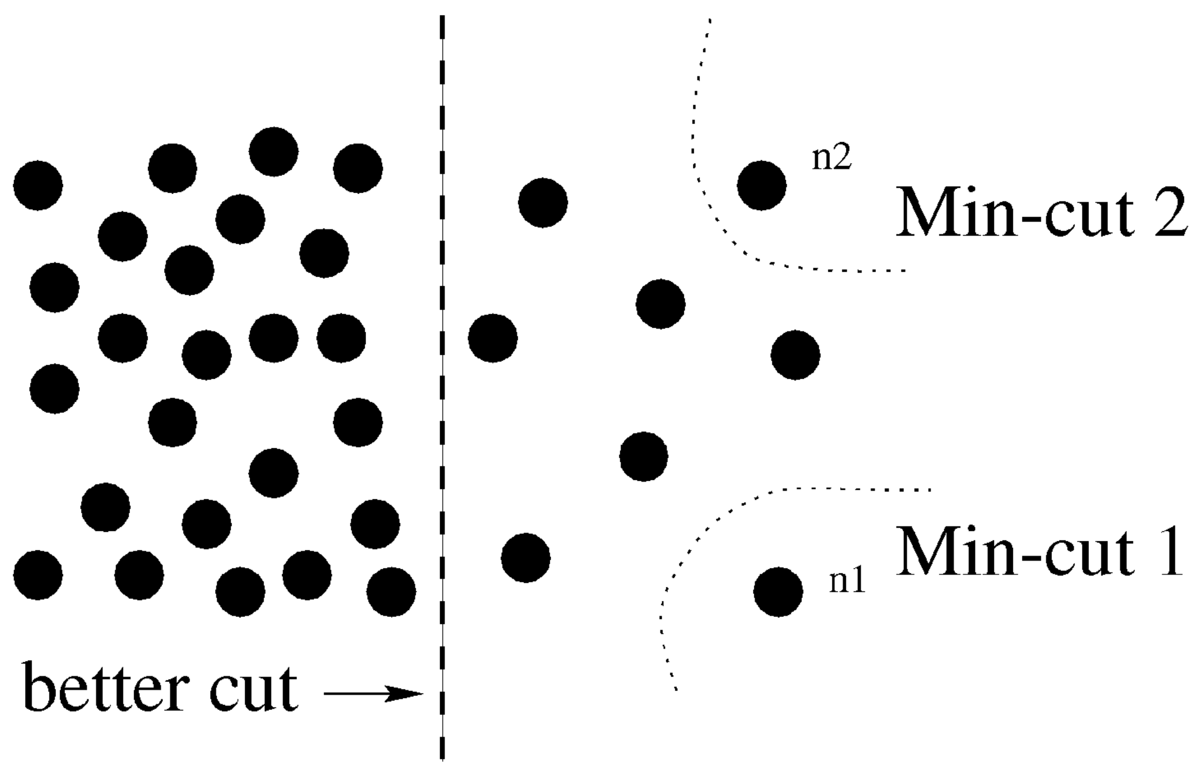
- A “minimal cut” is a cut, that has no proper subset that is also a cut.



A minimum cut

Graph partitioning

- For purposes of clustering, minimum cut is not a suitable choice, because it can favor very small partitions



- A better choice is “minimum normalized cut”. In a normalized cut, the sum of weights of the edges in the cut is normalized by the size of the resulting partitions.

Graph partitioning

- Let G=(V, E) and A and B, be two partitions. We define their corresponding cut:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

  - cut(A, B) is the sum of the weight of the edges between A and B.
  - assoc(A, V) is the sum of the edges with at least one endpoint in A.
  - Ncut, measures a cut; in the same spirit we can define a measure for the resulting partitions:

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$$

- Note that Ncut together with Nassoc, measure intraclass distance and interclass distance.

Graph partitioning

- An important observation is that these measures are naturally dependent:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} = \frac{assoc(A,V) - assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,V) - assoc(B,B)}{assoc(B,V)}$$

$$= 2 - (\frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}) = 2 - Nassoc(A,B)$$

- Therefor minimizing Ncut automatically maximizes Nassoc. intraclass distance and interclass distance respectively.

- Each partitioning (A, B) can be represented by a n-vector “x” where i-th entry is 1 iff vertex “i” is in A and -1 if in B. We can write Ncut:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} = \frac{\sum_{(x_i>0,x_j<0)} - w_{ij}x_ix_j}{\sum_{x_i>0} d_i} + \frac{\sum_{(x_i<0,x_j>0)} - w_{ij}x_ix_j}{\sum_{x_i<0} d_i}$$

Graph partitioning

- Using the previous definition we can define Ncut(x) for every n-vector x, with entries in {-1, 1}.

- So the minimum normalized cut problem can be stated as: $min_x Ncut(x) = min_y \frac{y^T(D - W)y}{y^T D y}$, where D is the diagonal degree matrix, W is weighted adjacency matrix and:

  I) $y_i \in \{-1, b\}$.

  II) $y^T D 1_n = 0$.

groups, are in fact identical and can be satisfied simultaneously. In our algorithm, we will use this normalized cut as the partition criterion.

Unfortunately, minimizing normalized cut exactly is NP-complete, even for the special case of graphs on grids. The proof, due to Papadimitriou, can be found in Appendix A. However, we will show that, when we embed the normalized cut problem in the real value domain, an approximate discrete solution can be found efficiently.

2.1 Computing the Optimal Partition

Given a partition of nodes of a graph, V, into two sets A and B, let $\boldsymbol{x}$ be an $N = |\boldsymbol{V}|$ dimensional indicator vector, $x_i = 1$ if node i is in A and -1, otherwise. Let $\boldsymbol{d}(i) = \sum_j w(i,j)$ be the total connection from node $\boldsymbol{i}$ to all other nodes. With the definitions $\boldsymbol{x}$ and $\boldsymbol{d}$, we can rewrite $Ncut(A,B)$ as:

$$\begin{aligned} Ncut(A,B) &= \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(B,A)}{assoc(B,V)} \\ &= \frac{\sum_{(\boldsymbol{x}_i>0,\boldsymbol{x}_j<0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i>0} \boldsymbol{d}_i} \\ &\quad + \frac{\sum_{(\boldsymbol{x}_i<0,\boldsymbol{x}_j>0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i<0} \boldsymbol{d}_i}. \end{aligned}$$

Let $\mathbf{D}$ be an $N \times N$ diagonal matrix with $\boldsymbol{d}$ on its diagonal, $\mathbf{W}$ be an $N \times N$ symmetrical matrix with W(i, j) $= w_{ij}$,

$$k = \frac{\sum_{x_i>0} \boldsymbol{d}_i}{\sum_i \boldsymbol{d}_i},$$

and $\mathbf{1}$ be an $N \times 1$ vector of all ones. Using the fact $\frac{\mathbf{1}+\boldsymbol{x}}{2}$ and $\frac{\mathbf{1}-\boldsymbol{x}}{2}$ are indicator vectors for $x_i > 0$ and $x_i < 0$, respectively, we can rewrite $4[Ncut(\boldsymbol{x})]$ as:

$$\begin{aligned} &= \frac{(\mathbf{1}+\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{k\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}-\boldsymbol{x})}{(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ &= \frac{(\boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x} + \mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1})}{k(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{2(1-2k)\mathbf{1}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}}{k(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}}. \end{aligned}$$

Let

$$\begin{aligned} \alpha(\boldsymbol{x}) &= \boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}, \\ \beta(\boldsymbol{x}) &= \mathbf{1}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}, \\ \gamma &= \mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1}, \end{aligned}$$

and

$$M = \mathbf{1}^T\mathbf{D}\mathbf{1},$$

we can then further expand the above equation as:

$$\begin{aligned} &= \frac{(\alpha(\boldsymbol{x})+\gamma) + 2(1-2k)\beta(\boldsymbol{x})}{k(1-k)M} \\ &= \frac{(\alpha(\boldsymbol{x})+\gamma) + 2(1-2k)\beta(\boldsymbol{x})}{k(1-k)M} - \frac{2(\alpha(\boldsymbol{x})+\gamma)}{M} + \frac{2\alpha(\boldsymbol{x})}{M} + \frac{2\gamma}{M}. \end{aligned}$$

Dropping the last constant term, which in this case equals 0, we get

$$\begin{aligned} &= \frac{(1-2k+2k^2)(\alpha(\boldsymbol{x})+\gamma) + 2(1-2k)\beta(\boldsymbol{x})}{k(1-k)M} + \frac{2\alpha(\boldsymbol{x})}{M} \\ &= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(\alpha(\boldsymbol{x})+\gamma) + \frac{2(1-2k)}{(1-k)^2}\beta(\boldsymbol{x})}{\frac{k}{1-k}M} \\ &\quad + \frac{2\alpha(\boldsymbol{x})}{M}. \end{aligned}$$

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

$$\begin{aligned} &= \frac{(1+b^2)(\alpha(\boldsymbol{x})+\gamma) + 2(1-b^2)\beta(\boldsymbol{x})}{bM} + \frac{2b\alpha(\boldsymbol{x})}{bM} \\ &= \frac{(1+b^2)(\alpha(\boldsymbol{x})+\gamma)}{bM} + \frac{2(1-b^2)\beta(\boldsymbol{x})}{bM} + \frac{2b\alpha(\boldsymbol{x})}{bM} - \frac{2b\gamma}{bM} \\ &= \frac{(1+b^2)(\boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x} + \mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1})}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ &\quad + \frac{2(1-b^2)\mathbf{1}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ &\quad + \frac{2b\boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} - \frac{2b\mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1}}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ &= \frac{(\mathbf{1}+\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ &\quad + \frac{b^2(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}-\boldsymbol{x})}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ &\quad - \frac{2b(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ &= \frac{[(\mathbf{1}+\boldsymbol{x}) - b(\mathbf{1}-\boldsymbol{x})]^T(\mathbf{D}-\mathbf{W})[(\mathbf{1}+\boldsymbol{x}) - b(\mathbf{1}-\boldsymbol{x})]}{b\mathbf{1}^T\mathbf{D}\mathbf{1}}. \end{aligned}$$

Setting $\boldsymbol{y} = (\mathbf{1}+\boldsymbol{x}) - b(\mathbf{1}-\boldsymbol{x})$, it is easy to see that

$$\boldsymbol{y}^T\mathbf{D}\mathbf{1} = \sum_{x_i>0} \boldsymbol{d}_i - b\sum_{x_i<0} \boldsymbol{d}_i = 0 \tag{4}$$

since $b = \frac{k}{1-k} = \frac{\sum_{x_i>0} \boldsymbol{d}_i}{\sum_{x_i<0} \boldsymbol{d}_i}$ and

$$\begin{aligned} \boldsymbol{y}^T\mathbf{D}\boldsymbol{y} &= \sum_{x_i>0} \boldsymbol{d}_i + b^2\sum_{x_i<0} \boldsymbol{d}_i \\ &= b\sum_{x_i<0} \boldsymbol{d}_i + b^2\sum_{x_i<0} \boldsymbol{d}_i \\ &= b(\sum_{x_i<0} \boldsymbol{d}_i + b\sum_{x_i<0} \boldsymbol{d}_i) \\ &= b\mathbf{1}^T\mathbf{D}\mathbf{1}. \end{aligned}$$

Putting everything together we have,

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\mathbf{D}-\mathbf{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathbf{D}\boldsymbol{y}}, \tag{5}$$

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

Note that the above expression is the Rayleigh quotient [11]. If $\boldsymbol{y}$ is relaxed to take on real values, we can minimize (5) by solving the generalized eigenvalue system,

$$(\mathbf{D}-\mathbf{W})\boldsymbol{y} = \lambda\mathbf{D}\boldsymbol{y}. \tag{6}$$

However, we have two constraints on $\boldsymbol{y}$ which come from the condition on the corresponding indicator vector $\boldsymbol{x}$. First, consider the constraint $\boldsymbol{y}^T\mathbf{D}\mathbf{1} = 0$. We can show this constraint on $\boldsymbol{y}$ is automatically satisfied by the solution of the generalized eigensystem. We will do so by first

Graph partitioning

- This is Rayleigh's quotient. If we relax constraints on y so it can hold real values, the solutions are the eigenvalues of the laplacian matrix.
- For graph G, the "Laplacian matrix" is defined as: L = D - W.
- Laplacian is symmetric positive semi-definite, so all eigenvalues are real non-negative, and eigenvectors are orthogonal.
- Sum of the entries on each row in laplacian is 0, So 1_n is a right eigenvector, and 0 is an eigenvalue
- Let A be a real symmetric matrix. Under the constraint that x is orthogonal to the j-1 smallest eigenvectors $x_1,\ldots,x_{j-1}$, the quotient $\frac{x^T A x}{x^T x}$ is minimized by the next smallest eigenvector x_j and its minimum value is the corresponding eigenvalue λ_j.

Graph partitioning

- The first eigenvector is 1_n and corresponds to trivial partitioning, so the first solution to the relaxed minimized cut problem is the second eigenvector (as it must be orthogonal to 1_n).

- Observe that similarly third eigenvector is a solution if "x" need to be orthogonal to the first two eigenvectors. In fact k-th eigenvector is a solution if "x" need to be orthogonal to the first k-1 vectors (this is useful when making k partitions).

- Note that if y is an eigenvector, constraint (II) is automatically satisfied.

- However eigenvectors do not generally satisfy constraint(I). In fact relaxing this constraint makes solving this problem possible; the original problem is NP-Complete (reduced to subset sum problem).

Graph partitioning

- To obtain a viable solution for the original problem (to discretize y), we search for a threshold "t" and assign vertex "i" to cluster A iff $x_2(i) > t$.
- Note that this is similar to using K-means to obtain a 2-clustering.
- When looking for a k-clustering we use K-means on the first k-1 eigenvectors. Since we need to make k-1 cuts in the graph.
- K-means algorithm is a good choice because this problem is an actual thresholding problem.
- We can see that this is an embedding to a lower space (from R^n to R^k) in which "close" vertices in G correspond to "close" points with Euclidian measure.

References

1.Normalized Cuts and Image Segmentation, Jianbo Shi and Jitendra Malik; IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 8, AUGUST 2000

2. https://en.wikipedia.org/wiki/Laplacian_matrix

3.https://en.wikipedia.org/wiki/Rayleigh_quotient