Spectral Clustering

Wei Wang @ CSE, UNSW

April 8, 2019

Quadratic Form

In mathematics, a quadratic form is a polynomial with terms all of degree two

- Let $\mathbf{A}$ be some $n \times n$ matrix. A是一个n*n的矩阵
- What is $\mathbf{Ax}$? What's the type of the output? What may $\mathbf{x}$ represent?
  - Some numeric assignment to $\{1, 2, \ldots, n\}$ (i.e., think of x_i as $x(i)$).
  - E.g., what if $x_i \in \{0, 1\}$? $x_i \in [0, 1]$? $x_i \in \Re$?
- What is $\mathbf{x}^\top \mathbf{Ax}$? What's the type of the output? Why it is called a qudratic form?

In mathematics, a quadratic form is a polynomial with terms all of degree two

方程中每一项都是二次,可以是xy 也可以是x^2

- Let $\mathbf{A}$ be some $n \times n$ matrix.
- What is $\mathbf{Ax}$? What's the type of the output? What may $\mathbf{x}$ represent? A就是系数矩阵 x代表的就是未知变量的矩阵[x,y]
  - Some numeric assignment to $\{1, 2, \ldots, n\}$ (i.e., think of x_i as $x(i)$). x^T代表的就是未知变量的转置矩阵 [x, y]
  - E.g., what if $x_i \in \{0, 1\}$? $x_i \in [0, 1]$? $x_i \in \Re$?
- What is $\mathbf{x}^\top \mathbf{Ax}$? What's the type of the output? Why it is called a qudratic form? 就是二次式子
  - $\mathbf{x}^\top \mathbf{Ax} = \sum_{i,j} A_{ij} \cdot (x_i x_j)$

Exercise:

- Rewrite $f_1(\mathbf{x}) = (3x_1 - 2x_2) + 4x_3^2$ into a quadratic form.
- Rewrite $f_2(\mathbf{x}) = (3x_1 - 2)^2 + (x_2 + x_1)^2$ into a quadratic form.

二次项只关注二次项,改变一次项或者常数项只是改变图形未知,并不改变图形形状,所以当要写出quadratic form的时候,一次项和常数项可以省略

Unnormalized Graph Laplacian /1

- (See the example graph later) Let **A** is the adjacency matrix of a "normal" (unweighted) *undirected* graph G. $\mathbb{V}$ are the vertices of G and $\mathbb{E}$ are the edges of G
  - An edge between v_i and v_j is modelled as (i, j) and (j, i), i.e., $A_{ij} = A_{ji} = 1$.
  - $A_{ii} =$ ____?
  - Write out A for the example graph.
  - How many edges are there in the example graph? 16

Unnormalized Graph Laplacian /2

- What is $\mathbf{x}^\top \mathbf{I}_n \mathbf{x}$? 这个是拉普拉斯矩阵
- What is $\mathbf{x}^\top \mathbf{D}\mathbf{x}$, where $\mathbf{D} \doteq \mathrm{Diag}(d_1, d_2, \ldots, d_n)$ and $d_i = deg(v_i) = \sum_{(i,j)\in\mathbb{E}} w_{ij}$? 这个是度矩阵
- What is $\mathbf{x}^\top \mathbf{A}\mathbf{x}$? 这个是邻接矩阵
- Now what about $2(\mathbf{x}^\top \mathbf{D}\mathbf{x} - \mathbf{x}^\top \mathbf{A}\mathbf{x})$?

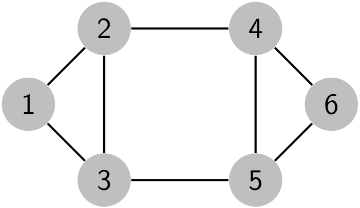
- $\mathbf{x}^\top \mathbf{I} \mathbf{x} = \sum_i x_i x_i$
- $\mathbf{x}^\top \mathbf{D} \mathbf{x} = \sum_i d_i \cdot x_i x_i = \sum_e x_i x_i$.
- $\mathbf{x}^\top \mathbf{A} \mathbf{x} = \sum_{(i,j) \in \mathbb{E}} x_i x_j = \sum_e x_i x_j$
- $2(\mathbf{x}^\top \mathbf{D} \mathbf{x} - \mathbf{x}^\top \mathbf{A} \mathbf{x}) = \sum_e (2x_i^2 - 2x_i x_j) = \sum_e x_i^2 + \sum_e x_j^2 - \sum_e 2x_i x_j = \sum_e (x_i - x_j)^2$

Example

$$\mathbf{x}^\top \mathbf{L} \mathbf{x} = \frac{1}{2} \cdot \sum_{e_{ij} \in \mathbb{E}} (x_i - x_j)^2 \quad \text{, where } \mathbf{L} = \mathbf{D} - \mathbf{A}.$$

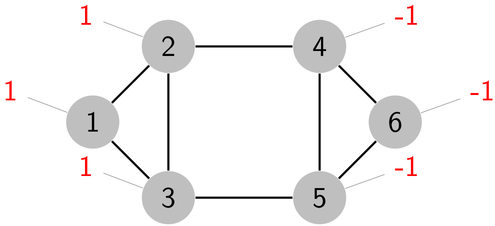
- ℓ_2 differences between assignments on the two ends of an edge, summed over all edges.

Example



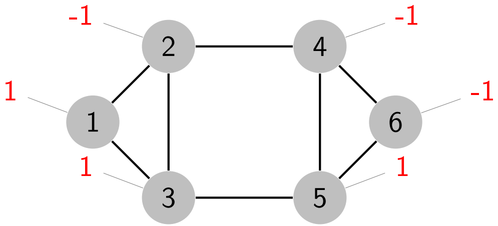
	n_1	n_2	n_3	n_4	n_5	n_6
n_1						
n_2						
n_3						
n_4						
n_5						
n_6						

- $\mathbf{1}_n$ is the one vector.
- $\mathbf{L}\mathbf{1}_n =$ (NB: $\mathbf{L}^\top = \mathbf{L}$) $\implies$ $\lambda_1 = 0, v_1 = \mathbf{1}_n$
- $\mathbf{x}^\top \mathbf{L} \mathbf{x} =$



- $\mathbf{x} =$
- $\mathbf{x}^\top \mathbf{L} \mathbf{x} =$

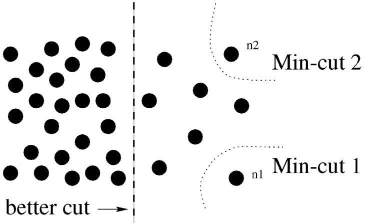
Binary $\mathbf{x}$ induces a Clustering /2



- $\mathbf{x} =$
- $\mathbf{x}^\top \mathbf{L} \mathbf{x} =$
- $\mathbf{x}^\top \mathbf{x} =$

Min Cut vs. Normalized Cut

- Min cuts are not *always* desirable.
  - Biased towards cutting small sets of isolated nodes.



- Cut: $cut(A, B) = \sum_{v_i \in A, v_j \in B} w_{ij}$.
- Normalized cut:

$$ncut(A, B) = \frac{cut(A, B)}{vol(A)} + \frac{cut(A, B)}{vol(B)},$$

where $vol(A) = \sum_{v_i \in A} d_i) = \sum_{v_i \in A, v_j \in \mathbb{V}} w_{i,j}$.

Connection to L

$$ncut(A,B) = cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$$

- Let $x_i = \frac{1}{vol(A)}$ if $v_i \in A$, and $= \frac{-1}{vol(B)}$ otherwise.
- $\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_e w_{ij}(x_i - x_j)^2 = 0 + \sum_{v_i \in A, v_j \in B}\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)^2$
- $\mathbf{x}^\top \mathbf{D}\mathbf{x} = (\mathbf{x}^\top \mathbf{D})\mathbf{x} = \sum_E d_i x_i^2 =$
  $\sum_{v_i \in A} \frac{d_i}{vol(A)^2} + \sum_{v_j \in B} \frac{d_j}{vol(B)^2} = \frac{1}{vol(A)} + \frac{1}{vol(B)}$

$$ncut(A,B) = \frac{\mathbf{x}^\top \mathbf{L}\mathbf{x}}{\mathbf{x}^\top \mathbf{D}\mathbf{x}}$$

Relaxation and Optimization

$$\text{Minimize } ncut(A, B) = \frac{\mathbf{x}^\top \mathbf{L} \mathbf{x}}{\mathbf{x}^\top \mathbf{D} \mathbf{x}} \quad \text{Subject to} \quad x_i \in \left\{ \frac{1}{vol(A)}, \frac{-1}{vol(B)} \right\}$$

- NP-hard to optimize under the discrete constraint.
- Relaxation: grow the feasible region of $\mathbf{x}$ and find the minimum value within the enlarged region.
  - allow $\mathbf{x}$ to be a real vector?
  - Yes, but too large.
  - This gives the constraint: $\mathbf{x}^\top \mathbf{D} \mathbf{1} = 0$ or equivalently $\mathbf{x}^\top \mathbf{D} \perp \mathbf{1}$ (You can verify this by plugging in any discrete vectors)
- Solution: the second smallest eigenvector of the generalized eigen value problem $\mathbf{L}\mathbf{x} = \lambda \mathbf{D}\mathbf{x}$.
- Normalized Laplacian:
  $\mathbf{L}' = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$

- Algorithm $SC\_recursive\_bin\_cut(data, k)$
  - Construct the weighted graph G
  - Construct the special graph laplacian L for G.
  - Compute the smallest non-zero eigenvector for L. This is the new representation of vertices in a new 1-dimensional space (i.e., embedding).
  - Cluster the vertices in the embedding space according to the objective function.
  - For each cluster, recursively call the algorithm if more clusters are needed.

Spectral Clustering Algorithm Framework

- Algorithm $SC\_k\_way\_cut(data, k)$
  - Construct the weighted graph G
  - Construct the special graph laplacian L for G.
  - Compute the smallest t non-zero eigenvector for L. This is the new representation of vertices in a new t-dimensional space (i.e., embedding).
  - Cluster the vertices in the embedding space using another clustering algorithm (e.g., k-means)

- How to construct the weighted graph if only n objects are given?
  - Be based on the similarity or distance among objects.
  - E.g., $w_{ij} = \exp(\frac{\|f(o_i)-f(o_j)\|}{2\sigma^2})$ where $f(o)$ is the feature vector of object o. One can also induce a sparse graph if one caps the raw weights by a threshold.
- Which Laplacian to use?
  - Unnormalized graph laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$.
  - Normalized graph laplacian $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}$.

Comments on Spectral Clustering

- Pros:
    - Usually better quality than other methods.
    - Can be thought of (non-linear) dimensionality reduction or embedding.
    - Freedom to construct a (sparse) G to preserve local similarity/connectivity.
    - Only requires some similarity measure.
    - Could be more efficient than k-means for high-dimensional sparse vectors (esp. if k-means is not fully optimized for such case).
- Cons:
    - Still need to determine k
    - Assumes clusters are of similar sizes.
    - Does not scale well with large datasets; but more scalable variants exist.
    - One of the relaxation of the original NP-hard problem – may not be the tightest relaxation.