Lecture #11: 10-16-18

· Fast recall: let $G=(V,E)$, $E = W\in\mathbb{R}^{n\times n}$, be a weighted graph with vertices corresponding to data $\{x_1,\dots,x_n\}\subset\mathbb{R}^D$ and edges with weight corresponding to affinity between points.

e.g. $W_{ij} = \exp(-\|x_i-x_j\|_2^2/\sigma^2)$, some scaling parameter $\sigma>0$.

· We can try to form clusters in the data by "cutting" the graph in a way such that

1.) Few edges are broken

2.) The volume is not too small for any of the clusters produced.

· For a partition into two clusters, $\{C_1,C_2\}$, set the normalized cut value to be

$$\mathrm{Ncut}(C_1,C_2) = \sum_{\substack{x_i\in C_1\\ x_j\in C_2}} W_{ij}\cdot\left(\frac{1}{\mathrm{vol}(C_1)}+\frac{1}{\mathrm{vol}(C_2)}\right),$$

$$\mathrm{vol}(C_\ell) = \sum_{\substack{x_i\in C_\ell\\ x_j\in V}} W_{ij} \qquad \text{n.b.: not } \sum_{\substack{x_i\in C_\ell\\ x_j\in C_\ell}} W_{ij}, \text{ as}$$

last time.

Remark: Could also do $\displaystyle\sum_{\substack{x_i\in C_1\\ x_j\in C_2}} W_{ij}\cdot\left(\frac{1}{|C_1|}+\frac{1}{|C_2|}\right)$.

- So, a "good" choice of clusters could be $(C_1^*, C_2^*) = \underset{(C_1, C_2)}{\arg\min}\ Ncut(C_1, C_2)$

Problem: Impossible to compute efficiently (NP-hard).

Solution: Recast as a linear algebra problem

- For a partition $(C_1, C_2) = (C, \bar{C})$, let $f^C \in \mathbb{R}^n$ be $f_i^C = \begin{cases} \sqrt{\frac{vol(\bar{C})}{vol(C)}}, & x_i \in C \\ -\sqrt{\frac{vol(C)}{vol(\bar{C})}}, & x_i \notin C. \end{cases}$

- Let $d_i = \sum_{x_j} W_{ij}$ be the degree of x_i, and set $D_{ij} = \begin{cases} d_i, & i = j \\ 0, & \text{else} \end{cases}$

to be the degree matrix.

Exercise (HW 6): 1) $(Df^C)^T \mathbb{1} = 0$, i.e. $\sum_{i=1}^n (Df^C)_i = 0$.

2) $(f^C)^T D f^C = vol(V) = \sum_{x_i, x_j \in V} W_{ij}$

- Let $L = D - W$ be the Laplacian of G. Then:

Exercise (HW 6): $(f^C)^T L f^C = vol(V)\, Ncut(C, \bar{C})$.

- Thus, to minimize $Ncut(C, \bar{C})$, it suffices to minimize $(f^c)^T L f^c$ over all choices of f^c.

$$(\vec{C}^*, \vec{\bar{C}}^*) = \arg\min_{f^c} (f^c)^T L f^c \quad \text{s.t.} \quad \begin{array}{l} 1.)\ (Df^c)^T \mathbb{1} = 0 \\ 2.)\ (f^c)^T D f^c = vol(V). \end{array}$$

- This is still NP-hard, so let's drop the requirement that $f_i^c = \begin{cases} \sqrt{\frac{vol(\bar{C})}{vol(C)}}, & x_i \in C \\ -\sqrt{\frac{vol(C)}{vol(\bar{C})}}, & x_i \notin C \end{cases}$

This yields:

$$(\vec{C}^*, \vec{\bar{C}}^*) = \cancel{\arg\min_{f^c} (f^c)^T L f^c}$$

i.e. $Df \perp \mathbb{1}$

$$\arg\min_{f \in \mathbb{R}^n} f^T L f \quad \text{s.t.} \quad \begin{array}{l} 1)\ (Df)^T \mathbb{1} = 0 \\ 2)\ fDf = vol(V) \end{array} \qquad \star$$

Can we solve this? Yes!

- Need one more trick: let $g = D^{\frac{1}{2}} f$, where $D^{\frac{1}{2}} = \begin{pmatrix} \sqrt{d_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{d_n} \end{pmatrix}$. Then

Since $D^{\frac{1}{2}}$ is invertible (no $d_i = 0$), $\star$ is equivalent to:

$$\underset{g}{\arg\min}\; g^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}T} g \quad \text{s.t. } 1)\; g \perp D^{\frac{1}{2}}\mathbb{1} \qquad (\star\star)$$

$$2)\; \|g\|_2^2 = \mathrm{vol}(V)$$

Notice that $D^{\frac{1}{2}}\mathbb{1}$ is an eigenvector of $D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$

$$= D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$$

$$= \underbrace{I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}}_{} := L_{sym} \;)$$

Since $L_{sym} D^{\frac{1}{2}}\mathbb{1}$

$$= \left(D^{-\frac{1}{2}} L D^{-\frac{1}{2}}\right) D^{\frac{1}{2}}\mathbb{1}$$

$$= D^{-\frac{1}{2}} L \mathbb{1}$$

$$= D^{-\frac{1}{2}}(D - W)\mathbb{1}$$

$$= D^{-\frac{1}{2}}(0)$$

$$= 0.$$

- In particular, the first eigenvalue is $\lambda_1 = 0$.
- Thus, the solution to $(\star\star)$ is g the second eigenvector of L_{sym}.

Setting $f = D^{-\frac{1}{2}} g$, we get that (exercise) f is the second eigenvector of $L_{rw} = I - D^{-\frac{1}{2}} W$.

really, just ϕ_2

- So, to cluster the data, we can: 1.) Compute the eigenvectors of L_{sym} (or L_{rw})

  2.) Threshold: $C_1 = \{x_i \mid \phi(x_i) > 0\}$

  $C_2 = \{x_i \mid \phi(x_i) \le 0\}$

- The Laplacian L_{rw} is the random walk Laplacian... connections to random walks on graphs...