3 Spectral Clustering and Cheeger's Inequality

3.1 Clustering

Clustering is one of the central tasks in machine learning. Given a set of data points, the purpose of clustering is to partition the data into a set of clusters where data points assigned to the same cluster correspond to similar data points (depending on the context, it could be for example having small distance to each other if the points are in Euclidean space).

3.1.1 k-means Clustering

One the most popular methods used for clustering is k-means clustering. Given $x_1, \ldots, x_n \in \mathbb{R}^p$ the k-means clustering partitions the data points in clusters $S_1 \cup \cdots \cup S_k$ with centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^p$ as the solution to:

$$\min_{\substack{\text{partition } S_1,\ldots,S_k \\ \mu_1,\ldots,\mu_k}} \sum_{l=1}^{k} \sum_{i \in S_i} \|x_i - \mu_l\|^2 . \tag{25}$$

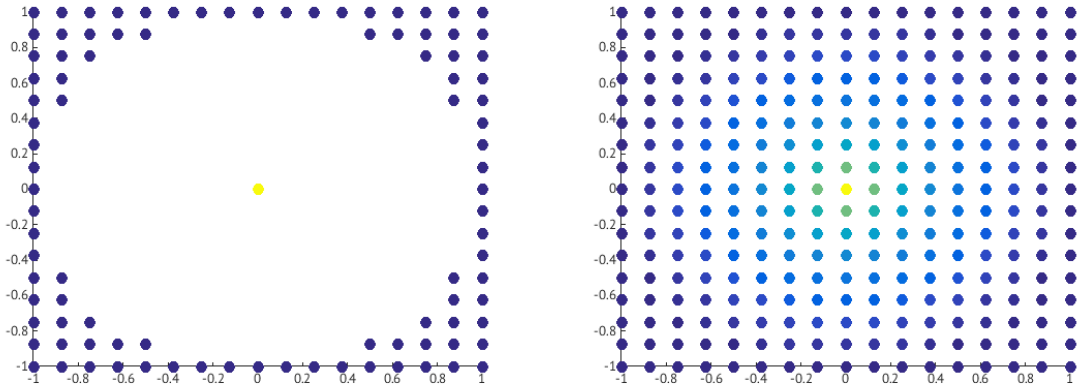

Figure 13: The $d = 2$ example of the use of this method to the example described above, the value of the nodes is given by color coding. For $d = 2$ it appears to smoothly interpolate between the labeled points.

Note that, given the partition, the optimal centers are given by

$$\mu_l = \frac{1}{|S_l|} \sum_{i \in S_l} x_i.$$

Lloyd's algorithm [Llo82] (also known as the k-means algorithm), is an iterative algorithm that alternates between

- Given centers $\mu_1, \ldots, \mu_k$, assign each point x_i to the cluster

$$l = \text{argmin}_{l=1,\ldots,k} \|x_i - \mu_l\|.$$

- Update the centers $\mu_l = \frac{1}{|S_l|} \sum_{i \in S_l} x_i$.

Unfortunately, Lloyd's algorithm is not guaranteed to converge to the solution of (25). Indeed, Lloyd's algorithm oftentimes gets stuck in local optima of (25). A few lectures from now we'll discuss convex relaxations for clustering, which can be used as an alternative algorithmic approach to Lloyd's algorithm, but since optimizing (25) is NP-hard there is not polynomial time algorithm that works in the worst-case (assuming the widely believed conjecture $P \neq NP$)

While popular, k-means clustering has some potential issues:

- One needs to set the number of clusters a priori (a typical way to overcome this issue is by trying the algorithm for different number of clusters).
- The way (25) is defined it needs the points to be defined in an Euclidean space, oftentimes we are interested in clustering data for which we only have some measure of affinity between different data points, but not necessarily an embedding in $\mathbb{R}^p$ (this issue can be overcome by reformulating (25) in terms of distances only).

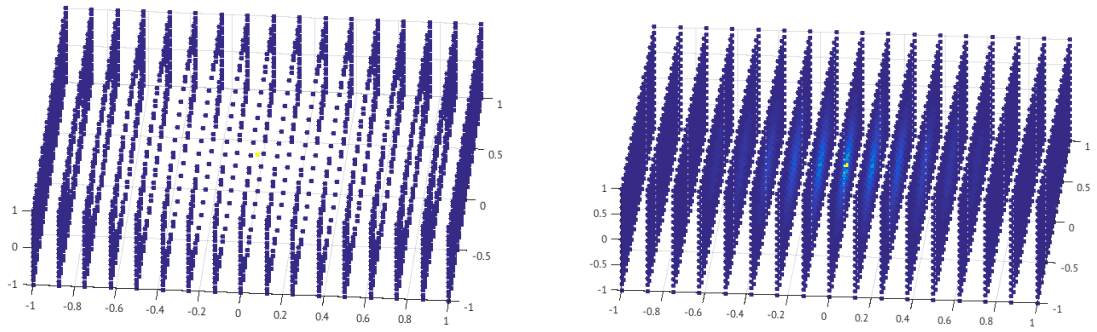

Figure 14: The $d = 3$ example of the use of this method to the example described above, the value of the nodes is given by color coding. For $d = 3$ the solution appears to only learn the label -1.

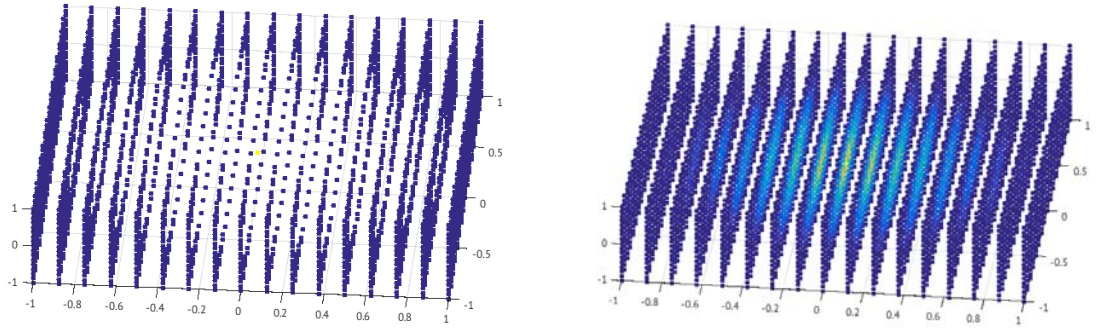

Figure 15: The $d = 3$ example of the use of this method with the extra regularization $f^T L^2 f$ to the example described above, the value of the nodes is given by color coding. The extra regularization seems to fix the issue of discontinuities.

- The formulation is computationally hard, so algorithms may produce suboptimal instances.
- The solutions of k-means are always convex clusters. This means that k-means may have difficulty in finding cluster such as in Figure 17.

3.2 Spectral Clustering

A natural way to try to overcome the issues of k-means depicted in Figure 17 is by using Diffusion Maps: Given the data points we construct a weighted graph $G = (V, E, W)$ using a kernel K_ϵ, such as $K_\epsilon(u) = \exp\left(\frac{1}{2\epsilon}u^2\right)$, by associating each point to a vertex and, for which pair of nodes, set the edge weight as

$$w_{ij} = K_\epsilon\left(\|x_i - x_j\|\right).$$

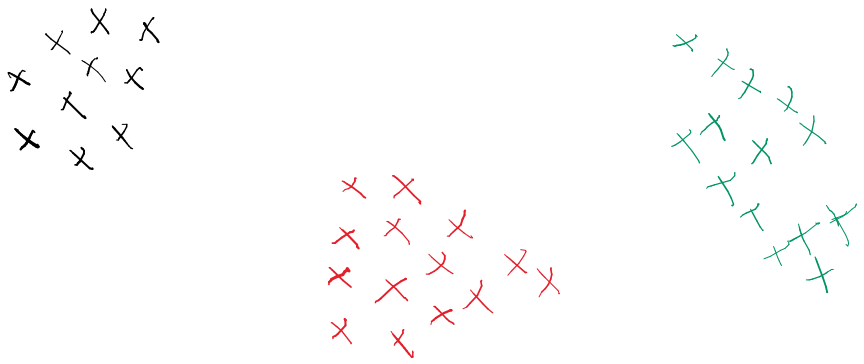

Figure 16: Examples of points separated in clusters.

Recall the construction of a matrix $M = D^{-1}W$ as the transition matrix of a random walk

$$\text{Prob}\{X(t+1) = j | X(t) = i\} = \frac{w_{ij}}{\deg(i)} = M_{ij},$$

where D is the diagonal with $D_{ii} = \deg(i)$. The d-dimensional Diffusion Maps is given by

$$\phi_t^{(d)}(i) = \begin{bmatrix} \lambda_2^t \varphi_2(i) \\ \vdots \\ \lambda_{d+1}^t \varphi_{d+1}(i) \end{bmatrix},$$

where $M = \Phi \Lambda \Psi^T$ where Λ is the diagonal matrix with the eigenvalues of M and Φ and Ψ are, respectively, the right and left eigenvectors of M (note that they form a bi-orthogonal system, $\Phi^T \Psi = I$).

If we want to cluster the vertices of the graph in k clusters, then it is natural to truncate the Diffusion Map to have $k-1$ dimensions (since in $k-1$ dimensions we can have k linearly separable sets). If indeed the clusters were linearly separable after embedding then one could attempt to use k-means on the embedding to find the clusters, this is precisely the motivation for Spectral Clustering.

Algorithm 3.1 (Spectral Clustering) *Given a graph $G = (V, E, W)$ and a number of clusters k (and t), Spectral Clustering consists in taking a $(k-1)$ dimensional Diffusion Map*

$$\phi_t^{(k-1)}(i) = \begin{bmatrix} \lambda_2^t \varphi_2(i) \\ \vdots \\ \lambda_k^t \varphi_k(i) \end{bmatrix}$$

and clustering the points $\phi_t^{(k-1)}(1), \phi_t^{(k-1)}(2), \ldots, \phi_t^{(k-1)}(n) \in \mathbb{R}^{k-1}$ using, for example, k-means clustering.

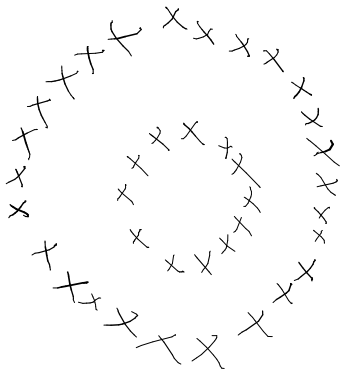

Figure 17: Because the solutions of k-means are always convex clusters, it is not able to handle some cluster structures.

3.3 Two clusters

We will mostly focus in the case of two cluster ($k = 2$). For $k = 2$, Algorithm 3.1 consists in assigning to each vertex i a real number $\varphi_2(i)$ and then clustering the points in the real line. Note in $\mathbb{R}$, clustering reduces to setting a threshold τ and taking $S = \{i \in V : \varphi_2(i) \leq \tau\}$. Also, it is computationally tractable to try all possible thresholds (there are $\leq n$ different possibilities).

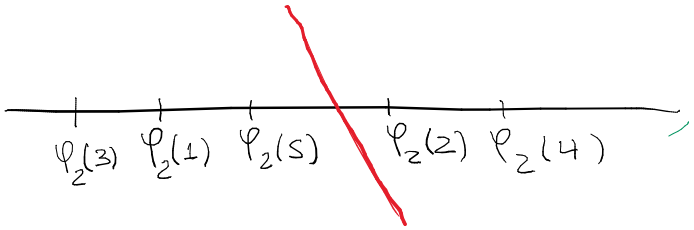

Figure 18: For two clusters, spectral clustering consists in assigning to each vertex i a real number $\varphi_2(i)$, then setting a threshold τ and taking $S = \{i \in V : \varphi_2(i) \leq \tau\}$.

Algorithm 3.2 (Spectral Clustering for two clusters) *Given a graph $G = (V, E, W)$, consider the two-dimensional Diffusion Map*

$$i \to \varphi_2(i).$$

set a threshold τ (one can try all different possibilities) and set

$$S = \{i \in V : \varphi_2(i) \leq \tau\}.$$

In what follows we'll give a different motivation for Algorithm 3.2.

3.3.1 Normalized Cut

Given a graph $G = (V, E, W)$, a natural measure to measure a vertex partition (S, S^c) is

$$\text{cut}(S) = \sum_{i \in S} \sum_{j \in S^c} w_{ij}.$$

Note however that the minimum cut is achieved for $S = \emptyset$ (since $\text{cut}(\emptyset) = 0$) which is a rather meaningless choice of partition.

Remark 3.3 *One way to circumvent this issue is to ask that $|S| = |S^c|$ (let's say that the number of vertices $n = |V|$ is even), corresponding to a balanced partition. We can then identify a partition with a label vector $y \in \{\pm 1\}^n$ where $y_i = 1$ is $i \in S$, and $y_i = -1$ otherwise. Also, the balanced condition can be written as $\sum_{i=1}^n y_i = 0$. This means that we can write the minimum balanced cut as*

$$\min_{\substack{S \subset V \\ |S| = |S^c|}} \text{cut}(S) = \min_{\substack{y \in \{-1,1\}^n \\ \mathbf{1}^T y = 0}} \frac{1}{4} \sum_{i \leq j} w_{ij} (y_i - y_j)^2 = \frac{1}{4} \min_{\substack{y \in \{-1,1\}^n \\ \mathbf{1}^T y = 0}} y^T L_G y,$$

where $L_G = D - W$ is the graph Laplacian.[13]*.*

Since asking for the partition to be balanced is too restrictive in many cases, there are several ways to evaluate a partition that are variations of $\text{cut}(S)$ that take into account the intuition that one wants both S and S^c to not be too small (although not necessarily equal to $|V|/2$). A prime example is Cheeger's cut.

Definition 3.4 (Cheeger's cut) *Given a graph and a vertex partition (S, S^c), the cheeger cut (also known as conductance, and sometimes expansion) of S is given by*

$$h(S) = \frac{\text{cut}(S)}{\min\{\text{vol}(S), \text{vol}(S^c)\}},$$

where $\text{vol}(S) = \sum_{i \in S} \deg(i)$.

Also, the Cheeger's constant of G is given by

$$h_G = \min_{S \subset V} h(S).$$

A similar object is the Normalized Cut, Ncut, which is given by

$$\text{Ncut}(S) = \frac{\text{cut}(S)}{\text{vol}(S)} + \frac{\text{cut}(S^c)}{\text{vol}(S^c)}.$$

Note that $\text{Ncut}(S)$ and $h(S)$ are tightly related, in fact it is easy to see that:

$$h(S) \leq \text{Ncut}(S) \leq 2h(S).$$

[13] W is the matrix of weights and D the degree matrix, a diagonal matrix with diagonal entries $D_{ii} = \deg(i)$.

Both $h(S)$ and $\mathrm{Ncut}(S)$ favor nearly balanced partitions, Proposition 3.5 below will give an interpretation of Ncut via random walks.

Let us recall the construction form previous lectures of a random walk on $G = (V, E, W)$:

$$\mathrm{Prob}\{X(t+1) = j | X(t) = i\} = \frac{w_{ij}}{\deg(i)} = M_{ij},$$

where $M = D^{-1}W$. Recall that $M = \Phi\Lambda\Psi^T$ where Λ is the diagonal matrix with the eigenvalues λ_k of M and Φ and Ψ form a biorthogonal system $\Phi^T\Psi = I$ and correspond to, respectively, the right and left eigenvectors of M. Moreover they are given by $\Phi = D^{-\frac{1}{2}}V$ and $\Psi = D^{\frac{1}{2}}V$ where $V^TV = I$ and $D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = V\Lambda V^T$ is the spectral decomposition of $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$.

Recall also that $M\mathbf{1} = \mathbf{1}$, corresponding to $M\varphi_1 = \varphi_1$, which means that $\psi_1^T M = \psi_1^T$, where

$$\psi_1 = D^{\frac{1}{2}}v_1 = D\varphi_1 = [\deg(i)]_{1\leq i\leq n}.$$

This means that $\left[\frac{\deg(i)}{\mathrm{vol}(G)}\right]_{1\leq i\leq n}$ is the stationary distribution of this random walk. Indeed it is easy to check that, if $X(t)$ has a certain distribution p_t then $X(t+1)$ has a distribution p_{t+1} given by $p_{t+1}^T = p_t^T M$

Proposition 3.5 *Given a graph $G = (V, E, W)$ and a partition (S, S^c) of V, $\mathrm{Ncut}(S)$ corresponds to the probability, in the random walk associated with G, that a random walker in the stationary distribution goes to S^c conditioned on being in S plus the probability of going to S condition on being in S^c, more explicitly:*

$$\mathrm{Ncut}(S) = \mathrm{Prob}\{X(t+1) \in S^c | X(t) \in S\} + \mathrm{Prob}\{X(t+1) \in S | X(t) \in S^c\},$$

where $\mathrm{Prob}\{X(t) = i\} = \frac{\deg(i)}{\mathrm{vol}(G)}$.

Proof. Without loss of generality we can take $t = 0$. Also, the second term in the sum corresponds to the first with S replaced by S^c and vice-versa, so we'll focus on the first one. We have:

$$\begin{aligned}
\mathrm{Prob}\{X(1) \in S^c | X(0) \in S\} &= \frac{\mathrm{Prob}\{X(1) \in S^c \cap X(0) \in S\}}{\mathrm{Prob}\{X(0) \in S\}} \\
&= \frac{\sum_{i\in S}\sum_{j\in S^c}\mathrm{Prob}\{X(1) \in j \cap X(0) \in i\}}{\sum_{i\in S}\mathrm{Prob}\{X(0) = i\}} \\
&= \frac{\sum_{i\in S}\sum_{j\in S^c}\frac{\deg(i)}{\mathrm{vol}(G)}\frac{w_{ij}}{\deg(i)}}{\sum_{i\in S}\frac{\deg(i)}{\mathrm{vol}(G)}} \\
&= \frac{\sum_{i\in S}\sum_{j\in S^c} w_{ij}}{\sum_{i\in S}\deg(i)} \\
&= \frac{\mathrm{cut}(S)}{\mathrm{vol}(S)}.
\end{aligned}$$

Analogously,

$$\mathrm{Prob}\{X(t+1) \in S | X(t) \in S^c\} = \frac{\mathrm{cut}(S)}{\mathrm{vol}(S^c)},$$

which concludes the proof. □

3.3.2 Normalized Cut as a spectral relaxation

Below we will show that Ncut can be written in terms of a minimization of a quadratic form involving the graph Laplacian L_G, analogously to the balanced partition.

Recall that balanced partition can be written as

$$\frac{1}{4}\min_{\substack{y\in\{-1,1\}^n\\ \mathbf{1}^T y=0}} y^T L_G y.$$

An intuitive way to relax the balanced condition is to allow the labels y to take values in two different real values a and b (say $y_i = a$ if $i \in S$ and $y_j = b$ if $i \notin S$) but not necessarily ± 1. We can then use the notion of volume of a set to ensure a less restrictive notion of balanced by asking that

$$a\,\mathrm{vol}\,(S) + b\,\mathrm{vol}\,(S^c) = 0,$$

which corresponds to $\mathbf{1}^T D y = 0$.

We also need to fix a scale/normalization for a and b:

$$a^2\,\mathrm{vol}\,(S) + b^2\,\mathrm{vol}\,(S^c) = 1,$$

which corresponds to $y^T D y = 1$.

This suggests considering

$$\min_{\substack{y\in\{a,b\}^n\\ \mathbf{1}^T D y=0,\, y^T D y=1}} y^T L_G y.$$

As we will see below, this corresponds precisely to Ncut.

Proposition 3.6 *For a and b to satisfy $a\,\mathrm{vol}\,(S) + b\,\mathrm{vol}\,(S^c) = 0$ and $a^2\,\mathrm{vol}\,(S) + b^2\,\mathrm{vol}\,(S^c) = 1$ it must be that*

$$a = \left(\frac{\mathrm{vol}(S^c)}{\mathrm{vol}(S)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}} \quad \text{and} \quad b = -\left(\frac{\mathrm{vol}(S)}{\mathrm{vol}(S^c)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}},$$

corresponding to

$$y_i = \begin{cases} \left(\frac{\mathrm{vol}(S^c)}{\mathrm{vol}(S)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S \\ -\left(\frac{\mathrm{vol}(S)}{\mathrm{vol}(S^c)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S^c. \end{cases}$$

Proof. The proof involves only doing simple algebraic manipulations together with noticing that $\mathrm{vol}(S) + \mathrm{vol}(S^c) = \mathrm{vol}(G)$. □

Proposition 3.7

$$\mathrm{Ncut}(S) = y^T L_G y,$$

where y is given by

$$y_i = \begin{cases} \left(\frac{\mathrm{vol}(S^c)}{\mathrm{vol}(S)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S \\ -\left(\frac{\mathrm{vol}(S)}{\mathrm{vol}(S^c)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S^c. \end{cases}$$

Proof.

$$\begin{aligned}
y^T L_G y &= \frac{1}{2}\sum_{i,j} w_{ij}(y_i - y_j)^2 \\
&= \sum_{i\in S}\sum_{j\in S^c} w_{ij}(y_i - y_j)^2 \\
&= \sum_{i\in S}\sum_{j\in S^c} w_{ij}\left[\left(\frac{\mathrm{vol}(S^c)}{\mathrm{vol}(S)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}} + \left(\frac{\mathrm{vol}(S)}{\mathrm{vol}(S^c)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}}\right]^2 \\
&= \sum_{i\in S}\sum_{j\in S^c} w_{ij}\frac{1}{\mathrm{vol}(G)}\left[\frac{\mathrm{vol}(S^c)}{\mathrm{vol}(S)} + \frac{\mathrm{vol}(S)}{\mathrm{vol}(S^c)} + 2\right] \\
&= \sum_{i\in S}\sum_{j\in S^c} w_{ij}\frac{1}{\mathrm{vol}(G)}\left[\frac{\mathrm{vol}(S^c)}{\mathrm{vol}(S)} + \frac{\mathrm{vol}(S)}{\mathrm{vol}(S^c)} + \frac{\mathrm{vol}(S)}{\mathrm{vol}(S)} + \frac{\mathrm{vol}(S^c)}{\mathrm{vol}(S^c)}\right] \\
&= \sum_{i\in S}\sum_{j\in S^c} w_{ij}\left[\frac{1}{\mathrm{vol}(S)} + \frac{1}{\mathrm{vol}(S^c)}\right] \\
&= \mathrm{cut}(S)\left[\frac{1}{\mathrm{vol}(S)} + \frac{1}{\mathrm{vol}(S^c)}\right] \\
&= \mathrm{Ncut}(S).
\end{aligned}$$

□

This means that finding the minimum Ncut corresponds to solving

$$\begin{array}{ll}
\min & y^T L_G y \\
\text{s. t.} & y \in \{a, b\}^n \text{ for some } a \text{ and } b \\
& y^T D y = 1 \\
& y^T D\mathbf{1} = 0.
\end{array} \tag{26}$$

Since solving (26) is, in general, NP-hard, we consider a similar problem where the constraint that y can only take two values is removed:

$$\begin{array}{ll}
\min & y^T L_G y \\
\text{s. t.} & y \in \mathbb{R}^n \\
& y^T D y = 1 \\
& y^T D\mathbf{1} = 0.
\end{array} \tag{27}$$

Given a solution of (27) we can *round* it to a partition by setting a threshold τ and taking $S = \{i \in V : y_i \leq \tau\}$. We will see below that (27) is an eigenvector problem (for this reason we call (27) a spectral relaxation) and, moreover, that the solution corresponds to y a multiple of φ_2 meaning that this approach corresponds exactly to Algorithm 3.2.

In order to better see that (27) is an eigenvector problem (and thus computationally tractable), set $z = D^{\frac{1}{2}}y$ and $\mathcal{L}_G = D^{-\frac{1}{2}}L_G D^{-\frac{1}{2}}$, then (27) is equivalent

$$\begin{aligned} \min \quad & z^T \mathcal{L}_G z \\ \text{s. t.} \quad & z \in \mathbb{R}^n \\ & \|z\|^2 = 1 \\ & \left(D^{\frac{1}{2}}\mathbf{1}\right)^T z = 0. \end{aligned} \tag{28}$$

Note that $\mathcal{L}_G = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. We order its eigenvalues in increasing order $0 = \lambda_1(\mathcal{L}_G) \leq \lambda_2(\mathcal{L}_G) \leq \cdots \leq \lambda_n(\mathcal{L}_G)$. The eigenvector associated to the smallest eigenvector is given by $D^{\frac{1}{2}}\mathbf{1}$ this means that (by the variational interpretation of the eigenvalues) that the minimum of (28) is $\lambda_2(\mathcal{L}_G)$ and the minimizer is given by the second smallest eigenvector of $\mathcal{L}_G = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, which is the second largest eigenvector of $D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ which we know is v_2. This means that the optimal y in (27) is given by $\varphi_2 = D^{-\frac{1}{2}} v_2$. This confirms that this approach is equivalent to Algorithm 3.2.

Because the relaxation (27) is obtained from (26) by removing a constraint we immediately have that

$$\lambda_2(\mathcal{L}_G) \leq \min_{S \subset V} \mathrm{Ncut}(S).$$

This means that

$$\frac{1}{2}\lambda_2(\mathcal{L}_G) \leq h_G.$$

In what follows we will show a guarantee for Algorithm 3.2.

Lemma 3.8 *There is a threshold τ producing a partition S such that*

$$h(S) \leq \sqrt{2\lambda_2(\mathcal{L}_G)}.$$

This implies in particular that

$$h(S) \leq \sqrt{4h_G},$$

meaning that Algorithm 3.2 is suboptimal at most by a square root factor.

Note that this also directly implies the famous Cheeger's Inequality

Theorem 3.9 (Cheeger's Inequality) *Recall the definitions above. The following holds:*

$$\frac{1}{2}\lambda_2(\mathcal{L}_G) \leq h_G \leq \sqrt{2\lambda_2(\mathcal{L}_G)}.$$

Cheeger's inequality was first established for manifolds by Jeff Cheeger in 1970 [Che70], the graph version is due to Noga Alon and Vitaly Milman [Alo86, AM85] in the mid 80s.

The upper bound in Cheeger's inequality (corresponding to Lemma 3.8) is more interesting but more difficult to prove, it is often referred to as the "the difficult part" of Cheeger's inequality. We will prove this Lemma in what follows. There are several proofs of this inequality (see [Chu10] for four different proofs!). The proof that follows is an adaptation of the proof in this blog post [Tre11] for the case of weighted graphs.

Proof. [of Lemma 3.8]

We will show that given $y \in \mathbb{R}^n$ satisfying

$$\mathcal{R}(y) := \frac{y^T L_G y}{y^T D y} \leq \delta,$$

and $y^T D\mathbf{1} = 0$. there is a "rounding of it", meaning a threshold τ and a corresponding choice of partition

$$S = \{i \in V : y_i \leq \tau\}$$

such that

$$h(S) \leq \sqrt{2\delta},$$

since $y = \varphi_2$ satisfies the conditions and gives $\delta = \lambda_2(\mathcal{L}_G)$ this proves the Lemma.

We will pick this threshold at random and use the probabilistic method to show that at least one of the thresholds works.

First we can, without loss of generality, assume that $y_1 \leq \cdot \leq y_n$ (we can simply relabel the vertices). Also, note that scaling of y does not change the value of $\mathcal{R}(y)$. Also, if $y^D\mathbf{1} = 0$ adding a multiple of $\mathbf{1}$ to y can only decrease the value of $\mathcal{R}(y)$: the numerator does not change and the denominator $(y + c\mathbf{1})^T D(y + c\mathbf{1}) = y^T Dy + c^2\mathbf{1}^T D\mathbf{1} \geq y^T Dy$.

This means that we can construct (from y by adding a multiple of $\mathbf{1}$ and scaling) a vector x such that

$$x_1 \leq ... \leq x_n, \ x_m = 0, \ \text{ and } x_1^2 + x_n^2 = 1,$$

and

$$\frac{x^T L_G x}{x^T D x} \leq \delta,$$

where m be the index for which $\text{vol}(\{1, \ldots, m-1\}) \leq \text{vol}(\{m, \ldots, n\})$ but $\text{vol}(\{1, \ldots, m\}) > \text{vol}(\{m, \ldots, n\})$.

We consider a random construction of S with the following distribution. $S = \{i \in V : x_i \leq \tau\}$ where $\tau \in [x_1, x_n]$ is drawn at random with the distribution

$$\text{Prob}\{\tau \in [a, b]\} = \int_a^b 2|\tau| d\tau,$$

where $x_1 \leq a \leq b \leq x_n$.

It is not difficult to check that

$$\text{Prob}\{\tau \in [a, b]\} = \begin{cases} \left|b^2 - a^2\right| & \text{if } a \text{ and } b \text{ have the same sign} \\ a^2 + b^2 & \text{if } a \text{ and } b \text{ have different signs} \end{cases}$$

Let us start by estimating $\mathbb{E}\,\text{cut}(S)$.

$$\begin{aligned} \mathbb{E}\,\text{cut}(S) &= \mathbb{E}\frac{1}{2}\sum_{i \in V}\sum_{j \in V} w_{ij}\mathbf{1}_{(S,S^c) \text{ cuts the edge } (i,j)} \\ &= \frac{1}{2}\sum_{i \in V}\sum_{j \in V} w_{ij}\,\text{Prob}\{(S, S^c) \text{ cuts the edge } (i, j)\} \end{aligned}$$

Note that $\text{Prob}\{(S, S^c) \text{ cuts the edge } (i, j)\}$ is $\left|x_i^2 - x_j^2\right|$ is x_i and x_j have the same sign and $x_i^2 + x_j^2$ otherwise. Both cases can be conveniently upper bounded by $|x_i - x_j|\,(|x_i| + |x_j|)$. This means that

$$\begin{aligned} \mathbb{E}\,\text{cut}(S) &\leq \frac{1}{2}\sum_{i,j} w_{ij}\,|x_i - x_j|\,(|x_i| + |x_j|) \\ &\leq \frac{1}{2}\sqrt{\sum_{ij} w_{ij}(x_i - x_j)^2}\sqrt{\sum_{ij} w_{ij}(|x_i| + |x_j|)^2}, \end{aligned}$$

where the second inequality follows from the Cauchy-Schwarz inequality.

From the construction of x we know that

$$\sum_{ij} w_{ij}(x_i - x_j)^2 = 2x^T L_G x \leq 2\delta x^T D x.$$

Also,

$$\sum_{ij} w_{ij}(|x_i| + |x_j|)^2 \leq \sum_{ij} w_{ij} 2x_i^2 + 2x_j^2. = 2\left(\sum_i \deg(i) x_i^2\right) + 2\left(\sum_j \deg(j) x_j^2\right) = 4x^T D x.$$

This means that

$$\mathbb{E}\,\mathrm{cut}(S) \leq \frac{1}{2}\sqrt{2\delta x^T D x}\sqrt{4x^T D x} = \sqrt{2\delta}\, x^T D x.$$

On the other hand,

$$\mathbb{E}\min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\} = \sum_{i=1}^n \deg(i)\,\mathrm{Prob}\{x_i \text{ is in the smallest set (in terms of volume)}\},$$

to break ties, if $\mathrm{vol}(S) = \mathrm{vol}(S^c)$ we take the "smallest" set to be the one with the first indices.

Note that m is always in the largest set. Any vertex $j < m$ is in the smallest set if $x_j \leq \tau \leq x_m = 0$ and any $j > m$ is in the smallest set if $0 = x_m \leq \tau \leq x_j$. This means that,

$$\mathrm{Prob}\{x_i \text{ is in the smallest set (in terms of volume)} = x_j^2.$$

Which means that

$$\mathbb{E}\min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\} = \sum_{i=1}^n \deg(i) x_i^2 = x^T D x.$$

Hence,

$$\frac{\mathbb{E}\,\mathrm{cut}(S)}{\mathbb{E}\min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\}} \leq \sqrt{2\delta}.$$

Note however that because $\frac{\mathbb{E}\,\mathrm{cut}(S)}{\mathbb{E}\min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\}}$ is not necessarily the same as $\mathbb{E}\frac{\mathrm{cut}(S)}{\min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\}}$ and so, we do not necessarily have

$$\mathbb{E}\frac{\mathrm{cut}(S)}{\min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\}} \leq \sqrt{2\delta}.$$

However, since both random variables are positive,

$$\mathbb{E}\,\mathrm{cut}(S) \leq \mathbb{E}\min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\}\sqrt{2\delta},$$

or equivalently

$$\mathbb{E}\left[\mathrm{cut}(S) - \min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\}\sqrt{2\delta}\right] \leq 0,$$

which guarantees, by the probabilistic method, the existence of S such that

$$\mathrm{cut}(S) \leq \min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\}\sqrt{2\delta},$$

which is equivalent to

$$h(S) = \frac{\mathrm{cut}(S)}{\min\{\mathrm{vol}\, S, \mathrm{vol}\, S^c\}} \leq \sqrt{2\delta},$$

which concludes the proof of the Lemma. □

3.4 Small Clusters and the Small Set Expansion Hypothesis

We now restrict to unweighted regular graphs $G = (V, E)$.

Cheeger's inequality allows to efficiently approximate its Cheeger number up to a square root factor. It means in particular that, given $G = (V, E)$ and ϕ we can efficiently between the cases where: $h_G \leq \phi$ or $h_G \geq 2\sqrt{\phi}$. Can this be improved?

Open Problem 3.1 *Does there exists a constant $c > 0$ such that it is NP-hard to, given ϕ, and G distinguis between the cases*

1. *$h_G \leq \phi$, and*
2. *$h_G \geq c\sqrt{\phi}$?*

It turns out that this is a consequence [RST12] of an important conjecture in Theoretical Computer Science (see [BS14] for a nice description of it). This conjecture is known [RS10] to imply the Unique-Games Conjecture [Kho10], that we will discuss in future lectures.

Conjecture 3.10 (Small-Set Expansion Hypothesis [RS10]) *For every $\epsilon > 0$ there exists $\delta > 0$ such that it is NP-hard to distinguish between the cases*

1. *There exists a subset $S \subset V$ with $\mathrm{vol}(S) = \delta\,\mathrm{vol}(V)$ such that $\frac{\mathrm{cut}(S)}{\mathrm{vol}(S)} \leq \epsilon$,*
2. *$\frac{\mathrm{cut}(S)}{\mathrm{vol}(S)} \geq 1 - \epsilon$, for every $S \subset V$ satisfying $\mathrm{vol}(S) \leq \delta\,\mathrm{vol}(V)$.*

3.5 Computing Eigenvectors

Spectral clustering requires us to compute the second smallest eigenvalue of $\mathcal{L}_G$. One of the most efficient ways of computing eigenvectors is through the power method. For simplicity we'll consider the case on which we are computing the leading eigenvector of a matrix $A \in \mathbb{R}^{n\times n}$ with m non-zero entries, for which $|\lambda_{\max}(A)| \geq |\lambda_{\min}(A)|$ (the idea is easily adaptable). The power method proceeds by starting with a guess y^0 and taking iterates $y^{t+1} = \frac{Ay^t}{\|Ay^t\|}$. One can show [KW92] that the variantes of the power method can find a vector x in randomized time $\mathcal{O}\left(\delta^{-1}(m+n)\log n\right)$ satisfying $x^TAx \geq \lambda_{\max}(A)(1-\delta)x^Tx$. Meaning that an approximate solution can be found in quasi-linear time.[14]

One drawback of the power method is that when using it, one cannot be sure, a posteriori, that there is no eigenvalue of A much larger than what we have found, since it could happen that all our guesses were orthogonal to the corresponding eigenvector. It simply guarantees us that if such an eigenvalue existed, it would have been extremely likely that the power method would have found it. This issue is addressed in the open problem below.

Open Problem 3.2 *Given a symmetric matrix M with small condition number, is there a quasi-linear time (on n and the number of non-zero entries of M) procedure that certifies that $M \succeq 0$. More specifically, the procedure can be randomized in the sense that it may, with some probably not certify that $M \succeq 0$ even if that is the case, what is important is that it never produces erroneous certificates (and that it has a bounded-away-from-zero probably of succeeding, provided that $M \succeq 0$).*

[14] Note that, in spectral clustering, an error on the calculation of φ_2 propagates gracefully to the guarantee given by Cheeger's inequality.

The Cholesky decomposition produces such certificates, but we do not know how to compute it in quasi-linear time. Note also that the power method can be used in $\alpha I - M$ to produce certificates that have arbitrarily small probability of being false certificates. Later in these lecture we will discuss the practical relevance of such a method as a tool to quickly certify solution produced by heuristics [Ban15b].

3.6 Multiple Clusters

Given a graph $G = (V, E, W)$, a natural way of evaluating k-way clusterign is via the k-way expansion constant (see [LGT12]):

$$\rho_G(k) = \min_{S_1,\ldots,S_k} \max_{l=1,\ldots,k} \left\{ \frac{\mathrm{cut}(S)}{\mathrm{vol}(S)} \right\},$$

where the maximum is over all choice of k disjoin subsets of V (but not necessarily forming a partition).

Another natural definition is

$$\varphi_G(k) = \min_{S:\mathrm{vol}\, S \leq \frac{1}{k}\mathrm{vol}(G)} \frac{\mathrm{cut}(S)}{\mathrm{vol}(S)}.$$

It is easy to see that

$$\varphi_G(k) \leq \rho_G(k).$$

The following is known.

Theorem 3.11 ([LGT12]) *Let $G = (V, E, W)$ be a graph and k a positive integer*

$$\rho_G(k) \leq \mathcal{O}\left(k^2\right)\sqrt{\lambda_k}, \tag{29}$$

Also,

$$\rho_G(k) \leq \mathcal{O}\left(\sqrt{\lambda_{2k}\log k}\right).$$

Open Problem 3.3 *Let $G = (V, E, W)$ be a graph and k a positive integer, is the following true?*

$$\rho_G(k) \leq \mathrm{polylog}(k)\sqrt{\lambda_k}. \tag{30}$$

We note that (30) is known not to hold if we ask that the subsets form a partition (meaning that every vertex belongs to at least one of the sets) [LRTV12]. Note also that no dependency on k would contradict the Small-Set Expansion Hypothesis above.

18.S096 Topics in Mathematics of Data Science
Fall 2015