**18.409 An Algorithmist's Toolkit** September 17, 2009

Lecture 3

*Lecturer: Jonathan Kelner* *Scribe: Andre Wibisono*

# 1 Outline

Today's lecture covers three main parts:

- Courant-Fischer formula and Rayleigh quotients
- The connection of $\lambda_2$ to graph cutting
- Cheeger's Inequality

# 2 Courant-Fischer and Rayleigh Quotients

The Courant-Fischer theorem gives a variational formulation of the eigenvalues of a symmetric matrix, which can be useful for obtaining bounds on the eigevalues.

**Theorem 1 (Courant-Fischer Formula)** *Let $A$ be an $n \times n$ symmetric matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ and corresponding eigenvectors $v_1, \ldots, v_n$. Then*

$$\lambda_1 = \min_{\|x\|=1} x^T A x = \min_{x \neq 0} \frac{x^T A x}{x^T x},$$

$$\lambda_2 = \min_{\substack{\|x\|=1 \\ x \perp v_1}} x^T A x = \min_{\substack{x \neq 0 \\ x \perp v_1}} \frac{x^T A x}{x^T x},$$

$$\lambda_n = \lambda_{\max} = \max_{\|x\|=1} x^T A x = \max_{x \neq 0} \frac{x^T A x}{x^T x}.$$

*In general, for $1 \leq k \leq n$, let $S_k$ denote the span of $v_1, \ldots, v_k$ (with $S_0 = \{0\}$), and let $S_k^\perp$ denote the orthogonal complement of $S_k$. Then*

$$\lambda_k = \min_{\substack{\|x\|=1 \\ x \in S_{k-1}^\perp}} x^T A x = \min_{\substack{x \neq 0 \\ x \in S_{k-1}^\perp}} \frac{x^T A x}{x^T x}.$$

**Proof** Let $A = Q^T \Lambda Q$ be the eigendecomposition of $A$. We observe that $x^T A x = x^T Q^T \Lambda Q x = (Qx)^T \Lambda (Qx)$, and since $Q$ is orthogonal, $\|Qx\| = \|x\|$. Thus it suffices to consider the case when $A = \Lambda$ is a diagonal matrix with the eigenvalues $\lambda_1, \ldots, \lambda_n$ in the diagonal. Then we can write

$$x^T A x = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \sum_{i=1}^n \lambda_i x_i^2.$$

We note that when $A$ is diagonal, the eigenvectors of $A$ are $v_k = e_k$, the standard basis vector in $\mathbb{R}^n$, i.e. $(e_k)_i = 1$ if $i = k$, and $(e_k)_i = 0$ otherwise. Then the condition $x \in S_{k-1}^\perp$ implies $x \perp e_i$ for $i = 1, \ldots, k-1$, so $x_i = \langle x, e_i \rangle = 0$. Therefore, for $x \in S_{k-1}^\perp$ with $\|x\| = 1$, we have

$$x^T A x = \sum_{i=1}^n \lambda_i x_i^2 = \sum_{i=k}^n \lambda_i x_i^2 \geq \lambda_k \sum_{i=k}^n x_i^2 = \lambda_k \|x\|^2 = \lambda_k.$$

On the other hand, plugging in $x = e_k \in S_{k-1}^{\perp}$ yields $x^T A x = (e_k)^T A e_k = \lambda_k$. This shows that

$$\lambda_k = \min_{\substack{\|x\|=1 \\ x \in S_{k-1}^{\perp}}} x^T A x.$$

Similarly, for $\|x\| = 1$,

$$x^T A x = \sum_{i=1}^{n} \lambda_i x_i^2 \leq \lambda_{\max} \sum_{i=1}^{n} x_i^2 = \lambda_{\max} \|x\|^2 = \lambda_{\max}.$$

On the other hand, taking $x = e_n$ yields $x^T A x = (e_n)^T A e_n = \lambda_{\max}$. Hence we conclude that

$$\lambda_{\max} = \max_{\|x\|=1} x^T A x.$$

■

The Rayleigh quotient is the application of the Courant-Fischer Formula to the Laplacian of a graph.

**Corollary 2 (Rayleigh Quotient)** *Let $G = (V, E)$ be a graph and $L$ be the Laplacian of $G$. We already know that the smallest eigenvalue is $\lambda_1 = 0$ with eigenvector $v_1 = \mathbf{1}$. By the Courant-Fischer Formula,*

$$\lambda_2 = \min_{\substack{x \neq 0 \\ x \perp v_1}} \frac{x^T A x}{x^T x} = \min_{\substack{x \neq 0 \\ x \perp \mathbf{1}}} \frac{\sum_{(i,j)\in E} (x_i - x_j)^2}{\sum_{i \in V} x_i^2},$$

$$\lambda_{\max} = \max_{x \neq 0} \frac{x^T A x}{x^T x} = \max_{x \neq 0} \frac{\sum_{(i,j)\in E} (x_i - x_j)^2}{\sum_{i \in V} x_i^2}.$$

We can interpret the formula for $\lambda_2$ as putting springs on each edge (with slightly weird boundary conditions corresponding to normalization) and minimizing the potential energy of the configuration.

Some big matrices are hard or annoying to diagonalize, so in some cases, we may not want to calculate the exact value of $\lambda_2$. However, we can still get an approximation by just constructing a vector $x$ that has a small Rayleigh quotient. Similarly, we can find a lower bound on $\lambda_{max}$ by constructing a vector that has a large Rayleigh quotient. We will look at two examples in which we bound $\lambda_2$.

## 2.1 Example 1: The Path Graph

Let $P_{n+1}$ be the path graph of $n+1$ vertices. Label the vertices as $0, 1, \ldots, n$ from one end of the path to the other. Consider the vector $x \in \mathbb{R}^{n+1}$ given by $x_i = 2i - n$ for vertices $i = 0, 1, \ldots, n$. Note that $\sum_{i=0}^{n} x_i = 0$, so $x \perp \mathbf{1}$. Calculating the Rayleigh quotient for $x$ gives us

$$\frac{\sum_{(i,j)\in E} (x_i - x_j)^2}{\sum_{i \in V} x_i^2} = \frac{4n}{\sum_{i=0}^{n} (2i - n)^2} = \frac{4n}{\Omega(n^3)} = O\left(\frac{1}{n^2}\right).$$

Thus we can bound $\lambda_2 \leq O(1/n^2)$. We knew this was true from the explicit formula of $\lambda_2$ in terms of sines and cosines from Lecture 2, but this is much cleaner and more general of a result.

## 2.2 Example 2: A Complete Binary Tree

Let $G$ be a complete binary tree on $n = 2^h - 1$ nodes. Define the vector $x \in \mathbb{R}^n$ to have the value 0 on the root node, $-1$ on all nodes in the left subtree of the root, and 1 on all nodes in the right subtree of the root.

It is easy to see that $\sum_{i \in V} x_i = 0$, since there are equal numbers of nodes on the left and right subtrees of the root, so $x \perp \mathbf{1}$. Calculating the Rayleigh quotient of $x$ gives us

$$\frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i \in V} x_i^2} = \frac{2}{n-1} = O\left(\frac{1}{n}\right).$$

Thus we get $\lambda_2 \leq O(1/n)$, again with little effort. It turns out in this case that our approximation is correct within a constant factor, and we did not even need to diagonalize a big matrix.

# 3 Graph Cutting

The basic problem of graph cutting is to cut a given graph $G$ into two pieces such that both are "pretty big". Graph cutting has many applications in computer science and computing, e.g. for parallel processing, divide-and-conquer algorithms, or clustering. In each application, we want to divide the problem into smaller pieces so as to optimize some measure of efficiency, depending on the specific problems.

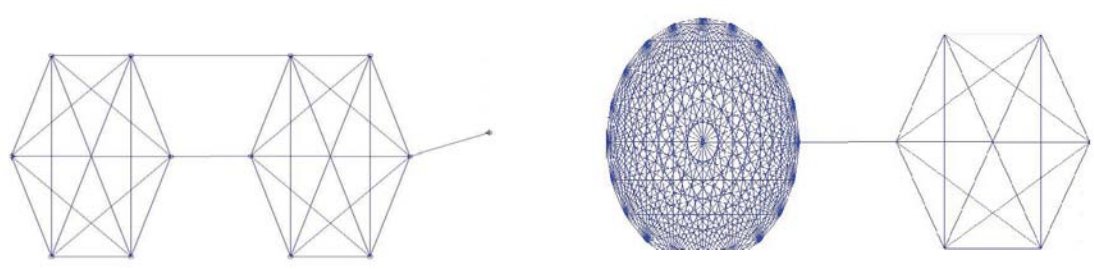
## 3.1 How Do We Cut Graphs?

The first question to ask about graph cutting is what we want to optimize when we are cutting a graph. Before attempting to answer this question, we introduce several notations. Let $G = (V, E)$ be a graph. Given a set $S \subseteq V$ of vertices of $G$, let $\bar{S} = V \setminus S$ be the complement of $S$ in $V$. Let $|S|$ and $|\bar{S}|$ denote the number of vertices in $S$ and $\bar{S}$, respectively. Finally, let $e(S)$ denote the number of edges between $S$ and $\bar{S}$. Note that $e(S) = e(\bar{S})$.

Now we consider some possible answers to our earlier question.

**Attempt 1: Min-cut.** Divide the vertex set $V$ into two parts $S$ and $\bar{S}$ to minimize $e(S)$. This approach is motivated by the intuition that to get a good cut, we do not want to break too many edges. However, this approach alone is not sufficient, as Figure 1(a) demonstrates. In this example, we ideally want to cut the graph across the two edges in the middle, but the min-cut criterion would result in a cut across the one edge on the right.

**Attempt 2: Approximate bisection.** Divide the vertex set $V$ into two parts $S$ and $\bar{S}$, such that $|S|$ and $|\bar{S}|$ are approximately $n/2$ (or at least $n/3$). This criterion would take care of the problem mentioned in Figure 1(a), but it is also not free of problems, as Figure 1(b) shows. In this example, we ideally want to cut the graph across the one edge in the middle that separates the two clusters. However, the approximate bisection criterion would force us to make a cut across the dense graph on the left.



(a) Problem with min-cut

(b) Problem with approximate bisection

**Figure 1**: Illustration for problems with the proposed graph cutting criteria.

Now we propose a criterion for graph cutting that balances the two approaches above.

**Definition 3 (Cut Ratio)** *The* **cut ratio** $\phi$ *of a cut* $S - \bar{S}$ *is given by*

$$\phi(S) = \frac{e(S)}{\min(|S|, |\bar{S}|)}.$$

*The* **cut of minimum ratio** *is the cut that minimizes* $\phi(S)$*. The* **isoperimetric number** *of a graph* $G$ *is the value of the minimum cut,*

$$\phi(G) = \min_{S \subseteq V} \phi(S).$$

As we can see from the definition above, the cut ratio is trying to minimize the number of edges across the cut, while penalizing cuts with small number of vertices. This criterion turns out to be a good one, and is widely used for graph cutting in practice.

## 3.2 An Integer Program for the Cut Ratio

Now that we have a good definition of graph cutting, the question is how to find the optimal cut in a reasonable time. It turns out that we can cast the problem of finding cut of minimum ratio as an integer program as follows.

Associate every cut $S - \bar{S}$ with a vector $x \in \{-1, 1\}^n$, where

$$x_i = \begin{cases} 1, & \text{if } i \in S, \text{ and} \\ -1, & \text{if } i \in \bar{S}. \end{cases}$$

Then it is easy to see that we can write

$$e(S) = \frac{1}{4} \sum_{(i,j) \in E} (x_i - x_j)^2.$$

For a boolean statement $A$, let $[A]$ denote the characteristic function on $A$, so $[A] = 1$ if $A$ is true, and $[A] = 0$ if $A$ is false. Then we also have

$$|S| \cdot |\bar{S}| = \left( \sum_{i \in V} [i \in S] \right) \left( \sum_{j \in V} [j \in \bar{S}] \right) = \sum_{i,j \in V} [i \in S, j \in \bar{S}] = \frac{1}{2} \sum_{i,j \in V} [x_i \neq x_j] = \frac{1}{4} \sum_{i<j} (x_i - x_j)^2.$$

Combining the two computations above,

$$\min_{x \in \{-1,1\}^n} \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2} = \min_{S \subseteq V} \frac{e(S)}{|S| \cdot |\bar{S}|}.$$

Now note that if $|V| = |S| + |\bar{S}| = n$, then

$$\frac{n}{2} \min(|S|, |\bar{S}|) \leq |S| \cdot |\bar{S}| \leq n \min(|S|, |\bar{S}|),$$

so we get

$$\frac{1}{n} \phi(G) = \min_{S \subseteq V} \frac{e(S)}{n \min(|S|, |\bar{S}|)} \leq \min_{x \in \{-1,1\}^n} \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2} \leq \min_{S \subseteq V} \frac{2e(S)}{n \min(|S|, |\bar{S}|)} = \frac{2}{n} \phi(G).$$

Therefore, solving the integer program

$$\min_{x \in \{-1,1\}^n} \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2}$$

allows us to approximate $\phi(G)$ within a factor of 2. The bad news is that it is NP-hard to solve this program. However, if we remove the $x \in \{-1, 1\}^n$ constraint, we can actually solve the program. Note that removing the constraint $x \in \{-1, 1\}^n$ is actually the same as saying that $x \in [-1, 1]^n$, since we can scale $x$ without changing the value of the objective function.

### 3.3 Interlude on Relaxations

The idea to drop the constraint $x \in \{-1, 1\}^n$ mentioned in the previous section is actually a recurring technique in algorithms, so it is worthwhile to give a more general explanation of this relaxation technique. A common setup in approximation algorithms is as follows: we want to solve an NP-hard question which takes the form of minimizing $f(x)$ subject to the constraint $x \in C$. Instead, we minimize $f(x)$ subject to a weaker constraint $x \in C' \supseteq C$ (see Figure 2 for an illustration). Let $p$ and $q$ be the points that minimize $f$ in $C$ and $C'$, respectively. Since $C \subseteq C'$, we know that $f(q) \leq f(p)$.

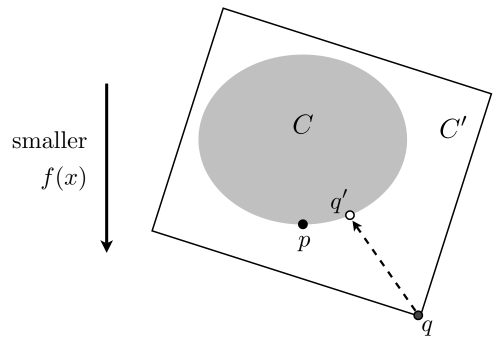



**Figure 2**: Illustration of the relaxation technique for approximation algorithms.

For this relaxation to be useful, we have to show how to "round" $q$ to a feasible point $q' \in C$, and prove $f(q') \leq \gamma f(q)$ for some constant $\gamma \geq 1$. This implies $f(q') \leq \gamma f(q) \leq \gamma f(p)$, so this process gives us a $\gamma$-approximation.

### 3.4 Solving the Relaxed Program

Going back to our integer program to find the cut of minimum ratio, now consider the following relaxed program,

$$\min_{x \in \mathbb{R}^n} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i<j}(x_i - x_j)^2}.$$

Since the value of the objective function only depends on the differences $x_i - x_j$, we can translate $x \in \mathbb{R}^n$ such that $x \perp \mathbf{1}$, i.e. $\sum_{i=1}^n x_i = 0$.

Then observe that

$$\sum_{i<j}(x_i - x_j)^2 = n \sum_{i=1}^n x_i^2,$$

which can be obtained either by expanding the summation directly, or by noting that $x$ is an eigenvector of the Laplacian of the complete graph $K_n$ with eigenvalue $n$ (as we saw in Lecture 2). Therefore, using the Rayleigh quotient,

$$\min_{x \in \mathbb{R}^n} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i<j}(x_i - x_j)^2} = \min_{\substack{x \in \mathbb{R}^n \\ x \perp \mathbf{1}}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{n \sum_{i=1}^n x_i^2} = \frac{\lambda_2}{n}.$$

Putting all the pieces together, we get

$$\begin{aligned}
\phi(G) &= \min_{S\subseteq V} \frac{e(S)}{\min(|S|,|\bar{S}|)} \\
&\geq \frac{n}{2} \min_{S\subseteq V} \frac{e(S)}{|S|\cdot|\bar{S}|} \\
&= \frac{n}{2} \min_{x\in\{-1,1\}^n} \frac{\sum_{(i,j)\in E}(x_i-x_j)^2}{\sum_{i<j}(x_i-x_j)^2} \\
&\geq \frac{n}{2} \min_{x\in\mathbb{R}^n} \frac{\sum_{(i,j)\in E}(x_i-x_j)^2}{\sum_{i<j}(x_i-x_j)^2} \\
&= \frac{n}{2} \min_{\substack{x\in\mathbb{R}^n \\ x\perp\mathbf{1}}} \frac{\sum_{(i,j)\in E}(x_i-x_j)^2}{n\sum_{i=1}^n x_i^2} \\
&= \frac{\lambda_2}{2}.
\end{aligned}$$

# 4 Cheeger's Inequality

In the previous section, we obtained the bound $\phi(G) \geq \lambda_2/2$, but what about the other direction? For that, we would need a rounding method, which is a way of getting a cut from $\lambda_2$ and $v_2$, and an upper bound on how much ithe rounding increases the cut ratio that we are trying to minimize. In the next section, we will see how to construct a cut from $\lambda_2$ and $v_2$ that gives us the following bound, which is Cheeger's Inequality.

**Theorem 4 (Cheeger's Inequality)** *Given a graph $G$,*

$$\frac{\phi(G)^2}{2d_{\max}} \leq \lambda_2 \leq 2\phi(G),$$

*where $d_{\max}$ is the maximum degree in $G$.*

As a side note, the $d_{max}$ disappears from the formula if we use the normalized Laplacian in our calculations, but the proof is messier and is not fundamentally any different from the proof using the regular Laplacian.

The lower bound of $\phi(G)^2/2d_{max}$ in Cheeger's Inequality is the best we can do to bound $\lambda_2$. The square factor $\phi(G)^2$ is unfortunate, but if it were within a constant factor of $\phi(G)$, we would be able to find a constant approximation of an NP-hard problem. Also, if we look at the examples of the path graph and the complete binary tree, their isoperimetric numbers are the same since we can cut exactly one edge in the middle of the graph and divide the graphs into two asymptotically equal-sized pieces for a value of $O(1/n)$. However, the two graphs have different upper bounds for $\lambda_2$, $O(1/n^2)$ and $O(1/n)$ respectively, which demonstrate that both the lower and upper bounds of $\lambda_2$ in Cheeger's inequality are tight (to a constant factor).

## 4.1 How to Get a Cut from $v_2$ and $\lambda_2$

Let $x \in \mathbb{R}^n$ such that $x \perp \mathbf{1}$. We will use $x$ as a map from the vertices $V$ to $\mathbb{R}$. Cutting $\mathbb{R}$ would thus give a partition of $V$ as follows: order the vertices such that $x_1 \leq x_2 \leq \ldots \leq x_n$, and the cut will be defined by the set $S = \{1, \ldots, k\}$ for some value of $k$. The value of $k$ cannot be known a priori since the best cut depends on the graph. In practice, an algorithm would have to try all values of $k$ to actually find the optimal cut after embedding the graph to the real line.

We will actually prove something slightly stronger than Cheeger's Inequality:

**Theorem 5** *For any $x \perp \mathbf{1}$, $x_1 \leq x_2 \leq \ldots \leq x_n$, there is some $i$ for which*

$$\frac{x^T L x}{x^T x} \geq \frac{\phi(\{1, \ldots, i\})^2}{2d_{max}}.$$

This is great because it not only implies Cheeger's inequality by taking $x = v_2$, but it also gives an actual cut. It also works even if we have not calculated the exact values for $\lambda_2$ and $v_2$; we just have to get a good approximation of $v_2$ and we can still get a cut.

## 4.2 Proof of Cheeger's Inequality

### 4.2.1 Step 1: Preprocessing

First, we are going to do some preprocessing. This step does not reduce the generality of the proof much, but it will make the actual proof cleaner.

- For simplicity, suppose $n$ is odd.
- Let $m = (n+1)/2$.
- Define the vector $y$ by $y_i = x_i - x_m$.

We can observe that $y_m = 0$, half of the vertices are to the left of $y_m$, and the other half are to the right of $y_m$.

**Claim 6**

$$\frac{x^T L x}{x^T x} \geq \frac{y^T L y}{y^T y}$$

**Proof** First, the numerators are equal by the operation of the Laplacian,

$$x^T L x = \sum_{(i,j) \in E} (x_i - x_j)^2 = \sum_{(i,j) \in E} \big((y_i + x_m) - (y_j + x_m)\big)^2 = \sum_{(i,j) \in E} (y_i - y_j)^2 = y^T L y.$$

Next, since $x \perp \mathbf{1}$,

$$y^T y = (x + x_m \mathbf{1})^T (x + x_m \mathbf{1}) = x^T x + 2x_m (x^T \mathbf{1}) + x_m^2 (\mathbf{1}^T \mathbf{1}) = x^T x + n x_m^2 \geq x^T x.$$

Putting together the two computations above yields the desired inequality. ■

### 4.2.2 Step 2: A Little More Preprocessing

We do not want edges crossing $y_m = 0$ (because we will later consider the positive and negative vertices separately), so we replace any edge $(i, j)$ with two edges $(i, m)$ and $(m, j)$. Call this new edge set $E'$.

**Claim 7**

$$\frac{\sum_{(i,j) \in E} (y_i - y_j)^2}{\sum_{i \in V} y_i^2} \geq \frac{\sum_{(i,j) \in E'} (y_i - y_j)^2}{\sum_{i \in V} y_i^2}.$$

**Proof** The only difference in the numerator comes from the edges $(i, j)$ that we split into $(i, m)$ and $(m, j)$. In that case, it is easy to see that (also noting that $y_m = 0$)

$$(y_j - y_i)^2 \geq (y_j - y_m)^2 + (y_m - y_i)^2.$$

■

### 4.2.3 Step 3: Breaking the Sum in Half

We would like to break the summations in half so that we do not have to deal with separate cases with positive and negative numbers. Let $E'_-$ be the edges $(i,j)$ with $i,j \le m$, and let $E'_+$ be the edges $(i,j)$ with $i,j \ge m$. We then have

$$\frac{\sum_{(i,j)\in E'}(y_i - y_j)^2}{\sum_i y_i^2} = \frac{\sum_{(i,j)\in E'_-}(y_i - y_j)^2 + \sum_{(i,j)\in E'_+}(y_i - y_j)^2}{\sum_{i=1}^m y_i^2 + \sum_{i=m}^n y_i^2}.$$

Note that $y_m$ appears twice in the summation on the denominator, which is fine since $y_m = 0$. We also know that for any $a, b, c, d > 0$,

$$\frac{a+b}{c+d} \ge \min\left(\frac{a}{c}, \frac{b}{d}\right),$$

so it is enough to bound

$$\frac{\sum_{(i,j)\in E'_-}(y_i - y_j)^2}{\sum_{i=1}^m y_i^2} \quad \text{and} \quad \frac{\sum_{(i,j)\in E'_+}(y_i - y_j)^2}{\sum_{i=m}^n y_i^2}.$$

Since the two values are essentially the same, we will focus only on the first one.

### 4.2.4 The Main Lemma

Let $C_i$ be the number of edges crossing the point $x_i$, i.e. the number of edges in the cut if we were to take $S = \{1, \ldots, i\}$. Recall that

$$\phi = \phi(G) = \min_{S\subseteq V} \frac{e(S)}{\min(|S|, |\bar{S}|)},$$

so by taking $S = \{1, \ldots, i\}$, we get $C_i \ge \phi i$ for $i \le n/2$ and $C_i \ge \phi(n-i)$ for $i \ge n/2$.

The main lemma we use to prove Cheeger's Inequality is as follows.

**Lemma 8 (Summation by Parts)** *For any $z_1 \le \ldots \le z_m = 0$,*

$$\sum_{(i,j)\in E'_-} |z_i - z_j| \ge \phi \sum_{i=1}^m |z_i|.$$

**Proof** For each $(i,j) \in E'_-$ with $i < j$, write

$$|z_i - z_j| = z_j - z_i = (z_{i+1} - z_i) + (z_{i+2} - z_{i+1}) + \cdots + (z_j - z_{j-1}) = \sum_{k=i}^{j-1}(z_{k+1} - z_k).$$

Summing over $(i,j) \in E'_-$, we observe that each term $z_{k+1} - z_k$ appears exactly $C_k$ times. Therefore,

$$\sum_{(i,j)\in E'_-} |z_i - z_j| = \sum_{k=1}^{m-1} C_k(z_{k+1} - z_k) \ge \phi \sum_{k=1}^{m-1} k(z_{k+1} - z_k).$$

Note that $z_i \le z_m = 0$, so $|z_i| = -z_i$ for $1 \le i \le m$. Then we can evaluate the last summation above as

$$\begin{aligned}\sum_{(i,j)\in E'_-} |z_i - z_j| &\ge \phi \sum_{k=1}^{m-1} k(z_{k+1} - z_k) \\ &= \phi\big((z_2 - z_1) + 2(z_3 - z_2) + 3(z_4 - z_3) + \cdots + (m-1)(z_m - z_{m-1})\big) \\ &= \phi(-z_1 - z_2 - \cdots - z_{m-1} + (m-1)z_m) \\ &= \phi \sum_{i=1}^m |z_i|.\end{aligned}$$

■

### 4.2.5 Using the Main Lemma to Prove Cheeger's Inequality

Now we can finally prove Cheeger's inequality.
**Proof of Cheeger's Inequality:** This proof has five main steps.

1. First, we normalize $y$ such that $\sum_{i=1}^{m} y_i^2 = 1$.
2. Next, this is perhaps a somewhat nonintuitive step, but we want to get squares into our expression, so we apply the main lemma (Lemma 8) to a new vector $z$ with $z_i = -y_i^2$. We now have

$$\sum_{(i,j)\in E'_-} |y_i^2 - y_j^2| \geq \phi \sum_{i=1}^{m} |y_i^2| = \phi.$$

3. Next, we want something that looks like $(y_i - y_j)^2$ instead of $y_i^2 - y_j^2$, so we are going to use the Cauchy-Schwarz inequality.

$$\sum_{(i,j)\in E'_-} |y_i^2 - y_j^2| = \sum_{(i,j)\in E'_-} |y_i - y_j| \cdot |y_i + y_j| \leq \left(\sum_{(i,j)\in E'_-} (y_i - y_j)^2\right)^{1/2} \left(\sum_{(i,j)\in E'_-} (y_i + y_j)^2\right)^{1/2}.$$

4. We want to get rid of the $(y_i + y_j)^2$ part, so we bound it and observe that the maximum number of times any $y_i^2$ can show up in the summation over the edges is the the maximum degree of any vertex.

$$\sum_{(i,j)\in E'_-} (y_i + y_j)^2 \leq 2 \sum_{(i,j)\in E'_-} (y_i^2 + y_j^2) \leq 2 \sum_{i=1}^{m} d_{max} \cdot y_i^2 \leq 2d_{max}.$$

5. Putting it all together, we get

$$\frac{\sum_{(i,j)\in E'_-} (y_i - y_j)^2}{\sum_{i=1}^{m} y_i^2} \geq \frac{\left(\sum_{(i,j)\in E'_-} |y_i^2 - y_j^2|\right)^2}{\sum_{(i,j)\in E'_-} (y_i + y_j)^2} \geq \frac{\phi^2}{2d_{max}}.$$

   Similarly, we can also show that

$$\frac{\sum_{(i,j)\in E'_+} (y_i - y_j)^2}{\sum_{i=m}^{n} y_i^2} \geq \frac{\phi^2}{2d_{max}}.$$

   Therefore,

$$\frac{x^T Lx}{x^T x} \geq \frac{y^T Ly}{y^T y} \geq \min\left\{\frac{\sum_{(i,j)\in E'_-} (y_i - y_j)^2}{\sum_{i=1}^{m} y_i^2}, \frac{\sum_{(i,j)\in E'_+} (y_i - y_j)^2}{\sum_{i=m}^{n} y_i^2}\right\} \geq \frac{\phi^2}{2d_{max}}.$$

■

### 4.2.6 So who is Cheeger anyway?

Jeff Cheeger is a differential geometer. His inequality makes a lot more sense in the continuous world, and his motivation was in differential geometry. This was part of his PhD thesis, and he was actually investigating heat kernels on smooth manifolds. A heat kernel can also be thought of as a point of heat in space, and the question is the speed at which the heat spreads. It can also be thought of as the mixing time of a random walk, which will be discussed in future lectures.

18.409 Topics in Theoretical Computer Science: An Algorithmist's Toolkit
Fall 2009