CS270 Algorithms & Data Structures **Spring 2008**

Lecture 4 Eigenvalues and cuts. : 1.31.08

Lecturer: Satish Rao *Scribe: Rao, for now*

Disclaimer: *These are rough notes, with some exercises.*

The notes have been remembered or inspired by or taken from various sources. Apologies to those who were plagiarized and/or not cited.

We are covering eigenvalues, and their connection to finding good cuts in graphs or proving there are none. The material uses pieces of S. Arora's notes (http://www.cs.princeton.edu/courses/archive/fall02/cs597D/lec6.ps), and previous (also) very rough notes (like these) notes of mine (www.cs.berkeley.edu/ satishr/cs273 on eigenvalues). It underlies some of the techniques in the currently popular "Normalized Cut" methods of Shi and Malik (search the internet for info.)

Question: Recall the relationship, we stated last time between S and $\rho(G)$?

Sure.

$$\frac{(n\ \mathrm{S}(G))^2}{d} \leq \rho(G) \leq \mathrm{S}\ (G). \tag{4.1}$$

Question: Remember how to prove the right hand side?

Start with cut, produce solution for embedding problem that gives upper bound on $\rho(G)$.

Question: How should we proceed?

Given a solution that gives $\rho(G)$, give an algorithm for finding a cut of small sparsity.

Question: What's an algorithm given the Rayleigh embedding for finding a sparse cut?

We have an embedding to a line, which induces an ordering of the vertices. Test each cut corresponding to prefix of ordering. Choose the cut with minimum sparsity.

Question: How do we proceed bounding the performance in terms of the Raleigh quotient?

Let Γ be the best Sof any cut in the procedure above. We will give an lower bound on $\rho(G)$ using Γ. Γ is at least the best sparsity, and thus the left inequality of 4.1 holds.

Question: Do you what the terrible, unintuitive short proof first or would you like to understand better?

4.1 Computing $\rho(G)$ and eigenvectors.

Exercise 1: Show that $\sum_{i,j}(x_i - x_j)^2 = 2n\sum_i(x_i - c)^2$, where $c = \sum_i x_i/n$ is the mean of the x_i?

We can thus rewrite $\rho(G) = \lambda_g/n$, where

$$\lambda_g = \min_{x:\sum_i x_i = 0} \frac{\sum_{e=(u,v)} (x_u - x_v)^2}{\sum_i x_i^2}. \tag{4.2}$$

Finding the Raleigh quotient is the same as finding the optimal value of λ_g.

Question: Recall the transition matrix for the random walk on the graph. What is it?

Say, it has 1/2 on the diagonals, and $1/2d$ on the off-diagonals. (We will assume that all nodes have the same degree, so that it is symmetric.)

Question: If you multiple a random vector by this matrix over and over again, what will happen?

You will get the vector consisting of all $1/n$.

Question: If you multiply the vector $1/n$ by this matrix, what will you get?

The same vector.

Question: What is this called?

An eigenvector. It's eigenvalue is 1.

Question: All the other eigenvalues are lower. In fact, the eigenvectors, $v_1, v_2, \ldots, v_n$ form a basis for the n-vectors. That is, we can rewrite say a random vector x, how?

$x = a_1 v_1 + a_2 v_2 \cdots a_n v_n$, where a_i are scalers.

Question: What happens when we multiply the vector by the matrix (whose eigenvalues are $\lambda_1 = 1, \ldots, \lambda_n$?

We get $x' = a_1 \lambda_1 v_1 + a_2 \lambda_2 v_2 \cdots a_n \lambda_n v_n$.

Question: What happens after t steps?

We get $x^t = a_1 v_1 + a_2 \lambda_2^t v_2 \cdots a_n \lambda_n^t v_n$.

Question: How long does it take for the remaining vector to be basically $a_1 v_1$?

Norm of second vector drops by 1/2 when $\lambda_2^t = 1/2$. Or when $t = -\log \lambda_2$ or say perhaps $1 - \lambda_2$. (We will assume $\lambda > 2$.

Question: This is called the eigenvalue gap. Why?

Don't be condescending. Well, ok, it is the gap between the first and the second eigenvalue.

Question: This also determines the mixing time. That is the time to get close to uniform. Why?

After $1/(1 - \lambda_2)$ steps the non-uniform part of the distribution halves and so on.

Question: More intuition. Mixing time of path?

At least $\Omega(n^2)$, thus $(1 - \lambda_2) = O(1/n^2)$.

Question: Let's think about the second eigenvalue for a bit. It is the vector that is orthogonal to the first that reduces by the *least* of all such vectors. What does it mean to be orthogonal

to the first?

The sum of the entries is equal to 0.

Question: How do we measure the reduction in terms of vectors?

We measure the norm.

Question: Consider a vector x, and in particular an edge (u,v). Consider x_u and x_v. What is the reduction in norm that occurs across the edge?

$x_u^2 + x_v^2 - (x_u+\delta)^2 - (x_v-\delta)^2 = -2\delta(x_u - x_v)$, where $\delta = (x_u - x_v)/2d$.

This is $-2(x_u-x_v)^2/2d$.

Question: Sum over the edges, and consider the reduction to be a fraction of the original norm, what do we get?

$$\frac{\sum_{e=(u,v)} 2(x_u - x_v)^2}{2d\sum_i x_i^2}$$

Hey, this is just λ_g scaled by $2d$! Moreover, the vector that reduces the norm by the least is the second eigenvector. Thus, we have

$$(1-\lambda_2^2) = \min_{x \text{othorgonal to } 1} \frac{\sum_{e=(u,v)} 2(x_u - x_v)^2}{2d\sum_i x_i^2}$$

which is basically $1/2d$ times λ_g. and $1/2dn$ times $\rho(G)$.

Hey, the second eigenvector gives us a vector that minimizes the Raleigh quotient.

Question: Can we compute it?

Sure the powering method above where one starts with a vector that is orthogonal to the all ones vector will converge to it. Isn't that basically the same algorithm for "drawing" the graph into the line that we discussed yesterday? (There we rescaled so the average distance was not zero. This essentially ensures that all the values don't converge to the same value, i.e., the first eigenvector.)

4.2 Some intuition for bounding $\rho(G)$.

Question: Let's examine the left hand side. Let's think of $\alpha = n\Gamma$ (recall this essentially the edge expansion of the cut within a factor of two. The left hand side (ignoring) d is basically quadratic in this quantity. Where does the quadratic behavior come from?

I don't know.

Question: Consider two embeddings; Both have (most) of the nodes embedded to two points at distance Δ. One has a single edge from left to right. One has a path of k edges from left to right, each edge being embedded to length Δ/k. What is $\rho(G)$ for each?

In both, the denominator of the raleigh quotient is $n^2\Delta^2$ The first the numerator Δ^2.. In the second is $\rho(G) = l(\Delta/l)^2 = \Delta^2/l$.

Thus, if the difference is "discharged" by a path, we divide the value of $\rho(G)$ (or $\lambda_g(G)$ by the length of the path.

Question: For a path of length l, is this the worst thing to do?

Yes, the numerator is the sum of squares of the edge lengths which is minimized when all the values are equal, i.e., $x+y=1$, x^2+y^2 is minimized when $x=y=1/2$, plus scaling and induction gives us the desired statement.

Thus, we have a lemma about $\sum_{(i,j)\in P}(x_i - x - j)^2$.

Lemma 4.1 *For a path P with at most l edges between nodes x_1 and x_l with $\Delta = (x_l - x_l)$,*

$$\sum_{(i,j} \in P)(x_i - x - j)^2 \geq \frac{\Delta^2}{l}$$

Question: Consider an embedding where at least $n/4$ nodes are embedded to 0 and at least $n/4$ nodes are embedded to 1, and the expansion is $\alpha = n \times sparsity$. How many edges leave the left set?

At least αn.

Question: Compute a max-flow from the nodes on the left to the nodes on the right. Upper bound the number of edges in is a typical path from the left to the right?

Compute a flow of size $\alpha n/4$ from left to right, which we can do by the max flow/min cut theorem, since the minimum cut in the middle is at least $\alpha n/4$. The average flow path has length bounded above by $dn/\alpha n/4 = O(d/alpha)$.

Question: What is $\rho(G)$?

The denominator is $\Theta(1/n^2)$.

For any path of length l the sum of $\sum_{e=(u,v)\in P} |x_u - x_v|^2$ is at least $1/l$. At least half of the αn paths in the flow have length $2dn/\alpha 4$. Thus the numerator of the Raleigh quotient is at least $\Omega(\alpha^2 n^2/d)$.

That is, $\rho(G) = \Omega(\alpha^2/d) = \Omega((nsparsity)^2/d)$. This is what we want to prove.

4.3 A proof based on this intuition.

Question: We will bound $\rho(G)$ (actually, $\lambda_g(G)$.), we call the numerator N and the denominator D.

That is, we will prove that

$$\lambda_g(G) = \frac{\sum_{e=(i,j)}(x_i - x_j)^2}{\sum_i x_i^2} = \Omega(\alpha^2/d),$$

where α is the expansion of a cut that is actually found in the proof.

Question: Start with an embedding. Consider that every cut has S at least Γ. Consider a median node in the embedding. We will try to bound $\sum_{i,j}(x_i - x_j)^2$ by considering the left and side separately. Can we make this simpler?

Well, for $i < n/2$ and $j > n/2$, we can approximate $(x_i - x_j)^2$ by $(x_i - x_m)^2 + (x_m - x_j)^2$ and only lose a factor of two.

We think of D' as this sum, with this approximation in place.

Question: We only consider the right side (for now). Define Δ_1 to be the distance to the right of the median node, one needs to go to leave at most $n/4$ nodes to the right. In general Δ_i is the distance one needs to go to leave at most $n/(2^{i+1})$ nodes to the right. Now, we can we upper bound. We redefine D to be the denominator for the right side where edges are cut as above. Give an upper bound on D'(portion of D' from right nodes), using these terms?

$D \leq \sum_i \Delta_{i+1}^2 n_i = \sum_i \Delta_{i+1}^2 n_i$, where $n_i = n/2^{i+1}$.

Question: Can we lower bound N_r'?

Consider the ith level, i.e., $n_i = n/2^{i+1}$ nodes to the right. We do the max-flow trick at this level. That is, we route flow from the n_i nodes on the right to the nodes to the left of level $i-1$. Notice that there are at most n_i nodes in the "middle."

Question: Can we lower bound the flow?

We know the minimum cut is at least αn_i in this flow problem. Thus, there are at least αn_i disjoint paths from right to left. (If you don't know this, you should worry a bit. But we will cover this later in the semester. It is referred to as the max-flow min-cut theorem.)

Question: What is the average number of hops in a path.

The average path hops is at most $dn_i/\alpha n_i \leq d/\alpha$.

Question: Can we use the Lake Wobegone non-theorem?

(In Lake Wobegone, all the kids are above average. Elsewhere, this doesn't happen.) Thus half the paths have length at most $2d/alpha$.

Question: Can we lower bound the numerator of the raleigh quotient?

The distance that each such path travels along the line is $\Delta_i - \Delta_{i-1}$. Thus, a typical path contributes $(\Delta_i - \Delta_{i-1})^2(2d/\alpha)$ and there are αn_i flow paths.

Thus, this level contributes $(\Delta_i - \Delta_{i-1})^2\alpha^2 n_i/d$ to the numerator.

Question: Let's assume that $\Delta_i - \Delta_{i-1}$ is approximately the same as Δ_i? Are we done?

Yes, at the ith level the contribution to the numerator $(\Delta_i - \Delta_{i-1})^2\alpha^2 n_i/d$ is within a factor of $\Theta(\alpha^2/d)$ of the contribution to the denominator $(\Delta_i^2 n_{i-1})$ at the $i-1$ level. (We didn't account for the last level of the denominator. But we can do so for the one node in that level, explicitly.)

Question: What do we do if $\Delta_i - \Delta_{i-1}$ is much less than Δ_i?

For these levels, we can "charge" the denominator value to the lower numbered group. That is, say $\Delta_{i+1} - \Delta_{i_1} \leq 1/4\Delta_{i_1}$. The contribution to the denominator at the ith level is at most $f = (5/4)^2(1/2) \leq 25/32$ times the contribution at the $i-1$ level, since Δ_i grows by at most $5/4$ and n_i drops by a factor of two.

By charging down this way, the contribution for a level to the denominator is increased by a geometric sum, $1 + 1/f$.... Thus, we can safely ignore these levels at the cost of a factor of $1/(1-f)$.

That is, we define the levels where the ratio is good to be the good levels. For the bad levels, we assume the numerator is 0, and move over the contribution of the denominator to the next smaller level (including whatever contribution it already had.)

4.4 Some exercises.

Eigen-Cut: Consider the algorithm of computing the second largest eigenvector. Sorting the vertices according to the second eigenvector. Scanning along the line. Computing α of each cut, and outputting the cut of minimum value.

Exercise 2: How does the resulting cut quality for "Eigen-Cut" compare to the optimal for the following graphs? Note: You can describe a vector whose λ_G is within a constant factor of optimal, rather than using the exact eigenvector. (E.g., for a line graph, the eigenvector is a cosine function. You can assume that it is a simple ramp function from 1 to -1 from left to right. Be off by more than a constant factor at your own risk.

Exercise 2.1: The dumbbell graph. **Exercise 2.2: The line graph.** **Exercise 2.3: A grid graph.** **Exercise 2.4: An rectangular graph.** **Exercise 2.5: A binary tree.** **Exercise 2.6: An k-dimensional grid graph.**

Exercise 3: Describe an example graph where the algorithm "Eigen-Cut" above does not give a cut with α that is within a constant factor of optimal.