CS270 Algorithms & Data Structures **Spring 2008**

Lecture 22 Approximation Algorithms: graph partitioning. : 4.10.08

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

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

Recall the sparsest cut problem from the beginning of class. How do we find a provably optimal cut?

Question: How about starting with the spectral method?

Sure, recall Cheeger's inequality. That is,

$$\Theta(n\Phi^2/d) \leq \rho \leq \Phi,$$

or

$$\Theta(\alpha^2/d) \leq n\rho \leq \alpha,$$

where the Φ on the left could be found using the "spectral" algorithm (cut along sorted line.)

In the case α is approximately $\Theta(1)$. That is an expander graph. This leads to a constant factor approximation. When the graph is far from being an expander the approximation ratio is essentially the expansion. Thus, for graphs with good cuts, the algorithm could be quite far from being optimal.

Question: Can we do better? With linear programming?

Sure. What is a linear programming relaxation. Well, we need a computable lower bound on the cut size (typically) to prove an approximation bound.

Question: Any candidates?

Recall way back in the dark ages of this course.

Question: How did we show that the grid has good expansion?

Well, we gave an embedding of the complete graph into the underlying graph and showed it had small congestion. The fact there was a congestion c embedding implied that every $k, n-k$ cut of the graph could support $k(n-k)$ edges of the complete graph with congestion c. But, this flow is supported by only the C edges in the cut. That is, $c \geq k(n-k)/C$, which in turn implied that $C/k(n-k) \geq 1/c$. This is a lower bound on sparsity.

Question: What about linear programming?

We can express the complete graph embedding problem is a linear program as follows.

$$\min c$$

$$\forall i,j; \sum_{p \in \mathcal{P}_{ij}} f_p = 1.$$

$$\forall e; \sum_{e \in p} f_p \leq 1.$$

Here, $\mathcal{P}_{ij}$ is the set of ij paths. This is exponential, but can easily be reformulated with a polynomial expression or can be solved using the path selection algorithms described in lecture 12.

We know that $1/c$ is a lower bound on the sparsity of any cut in the graph.

Question: An interpretation?

Everyone goes to a random place, the congested roads should be in "sparse" cuts.

Question: What is the dual?

This too, we saw in lecture 12. But, here it is.

$$\max \sum_{ij} d_i j.$$

$$\forall p \in \mathcal{P}_{ij}, d_{ij} \leq \sum_{e \in p} d(e)$$

$$\sum_{e} d(e) = 1$$

Question: An interpretation?

The $d(e)$ are edge weights. The d_{ij} are distances between i and j in the graph. This could be called a distance metric.

Question: Can we argue directly that $1/v$ is a lower bound on the sparsity?

Sure. Given a cut, assign $d(e) = 1/C$ on the edges in the cut. The value is $k(n-k) * 1/C$ or $1/\phi(S)$, where S is the cut.

Question: For future reference what is a lower bound on expansion?

Well, n/V. (Just by the relationship between sparsity and expansion.

Question: An interpretation?

Make it as costly as possible for everyone going to a random place by assigning a fixed amount of tolls to the roads. Assign tolls to sparse cut may be reasonable.

Question: So, we know LP lower bounds sparsity. Can we find a sparse cut?

Sure.

Question: How?

Perhaps you recall from way back that high diameter graphs have a small cut. This is trivial, due to Breadth first search. If diameter is D, then there is a cut of size $|E|/D$.

Question: Is this sparse?

No. But we can prove the following lemma.

Lemma 22.1 *If the diameter of a graph is at least D. Then there is a cut of edge expansion $\Theta(\log |E|/D)$.*

Edge expansion is the number of edges in the cut divided by the number of edges on the smaller side.

Question: Proof?

Well, do a breadth first search. Group the (at least) D levels into say $2\log n$ groups each of $D/2\log|E|$ levels.

Each group either contains more edges than all the previous groups or less, say the previous set of edges is E_p. For now, we assume that the group and its predecessors contain fewer than half the edges. (If not, we do the same except we say the group either contains more edges than all *successive* groups or not.)

If it contains fewer, then we can find a cut that has size $|E_p|/(D/2\log|E|)$. Moreover, the edge expansion is at most $2\log n/D$ since this cut cuts off at least $|E_p|$ edges.

If it contains more than $|E_p|$ edges, then the number of edges has doubled after this group. This can only occur $\log|E|$ times.

Since there are at least $2\log|E|$ groups (we get the 2 from the reducing and expanding ends), the former case must occur at least once.

Question: Discrete versus continuous? Is there a continuous statement above?

Sure. It is easy to see that the proof above, works with weights. That is, the following lemma holds.

Lemma 22.2 *If the diameter of a graph is at least D. Then there is a cut of weighted edge expansion $\Theta(\log|E|/D)$.*

Now, weighted edge expansion means simply dividing the size of the cut by the weight of the edges on the smaller size. (The $\log|E|$ would perhaps be $\log W/w_{min}$ where W is the total weight and w_{min} is the smallest. But one can do this so that one gets $\log|E|$ by rounding things a bit.)

That is, we get...

Lemma 22.3 *For a weighted graph of diameter at least D. Then there is a cut of weighted edge expansion $\Theta(\log|E|/D)$, that is cuts of W_s weight and cuts $\Theta(W_s\log n/D)$ edges.*

Question: Let's apply this lemma to the LP solution?

Recall, that the value of is V. Thus, the average distance between pairs is $D = V/n^2$. Thus, we can apply the lemma to get a cut with weighted edge expansion at most

$$\frac{n^2\log|E|}{V}.$$

Question: What is the actual expansion?

Well, in terms of the number of nodes, let n_i and W_i be the weight of the edges that are cut off.

$$\frac{n^2\log|E|}{V}\frac{W_i}{n_i},$$

If n_i/W_i is approximately $n/W = n$ (where W is the total weight of the edges which is 1.) Then, the actual expansion is

$$\approx \frac{n \log |E|}{V}.$$

Now recall that the optimal expansion is at least $n/|V|$. So, we obtain an $O(\log |E|)$ approximation.

Question: Really?

No, we assumed that $n_i/W_i \geq n/2W$. This may not be true. The bad case being that $n_i/W_i << n/W$.

Question: How to proceed?

If at first you don't succeed, try try again. So, cut the remaining graph and throw the small side in with the already removed portion.

Now, either the removed portion has ratio n_r/W_r which is at least $n/2W$ or it doesn't. If yes, the removed portion has the requisite weighted edge expansion and therefore real edge expansion.

If not, you continue. Eventually, you must remove at least the nodes, all of which removed twice as much weight as it should have. That is, you removed more than original edge weight.

Question: Huh?

Yes. That is, this is a contradiction. Thus, at some point the removed portion forms a cut of the desired ratio.

Exercise 1: Consider the problem LP relaxation for the problem of finding a cut separating pairs $(s_1, t_1), \ldots, (s_k, t_k)$.

Exercise 1.1: Give an LP relaxation of this problem. Hint: what if we assigned a weight of 1 to every node in an optimal cut, what is the minimum distance between any s_i and t_i? What is the total weight?

Exercise 1.2: Give a rounding algorithm that gives an $O(\log n)$ approximation factor. Hint: Use lemma 22.3 repeatedly.

22.1 Metric spaces.

Another, actually more general proof follows from embedding theorems of metric spaces into Euclidean metric spaces.

Question: Huh?

Ok, the $d(u, v)$ form a distance metric on the nodes in the graph.

Question: Can we map this metric into points in space where the distances are approximately preserved?

We will show an embedding that preserves all distances to within a factor of $O(\log n)$. In particular, we will not grow any distance, and shrink no distance by more than an $O(\log n)$ factor.

Question: How to proceed?

We should form coordinates which assign a value for each point. In particular, we form coordinate i, j by choosing a random subset S_{ij} of where a node is included in S_{ij} with probability $1/2^i$ of nodes and assigning $d(v, S_{ij})$ for the ij *pre*-coordinate to node v. We will have i range from 0 to $\log n$ and j range up to some value which is $\Theta(\log n)$.

Now, i influences the size of the subset we choose. The j parameter just repeats each size some number of times.

We scale down the pre-coordinates by 1/#coordinates to obtain our embedding.

Question: What is an upper bound on the ℓ_1 distance between two points?

At most $d(u,v)$. Since $|p_{ij}(u) - p_{ij}(v)| \leq d(u,v)$ by the triangle inequality. That is, the distance of u to the set S_{ij} can be no more than the distance of v to the set S_{ij} plus the distance from u to v (and vice versa.)

Question: Can we lower bound the distance that we obtain between two nodes u and v?

Let's look at a ball around u that contains 2^i nodes and one that contains 2^{i+1} nodes. The differences in radius is $\rho_i - \rho_{i-1}$. Similarly, we can define the same notion for v, and define the radius's $\pi_i - \pi_{i-1}$.

We consider the final value i, say I, where the balls do not overlap. We know that $d(u,v) \geq \rho_I + \pi_I$.

Now, consider a set S of some S_{ij} of size $n/2^i$. With constant probability the set S hits the ball around u of size 2^{i-1} and thus u has pre-coordinate less than π_{i-1}, and misses the ball around u of size 2^i and v has pre-coordinate more than ρ_i. Thus, with constant probability we get at least $\rho_i - \pi_{i-1}$ contribution. Similiarly, with constant probability we get at least $\pi_i - \rho_{i-1}$ contribution.

That is, on expectation we get a constant times

$$\sum_i |\rho_i - \pi_{i-1}| + |\pi_i - \rho_{i-1}|$$

This sum telescopes, and we get $\rho_I + \pi_I$.

Repeating each level $\Theta(\log n)$ times makes the expectatation result hold with high probability. And we get that the total pre-coordinate ℓ_1 distance is $\Theta(\log n) d(u,v)$.

Question: Scaling down, what do we get?

There were $O(\log^2 n)$ coordinates, so we get an upper bound of $d(u,v)$ on the ℓ_1 distance.

And we get a lower bound of $\Omega(d(u,v)/\log n)$. This is a version of Bourgain's embedding (though he did it for ℓ_2 distances.)

Exercise 2: Bourgain's original construction worked to embed into the Euclidean metric. What should the values of each coordinate be in order for the result to be approximately preserve distances in the ℓ_2 metric?

Question: What does this have to do with the linear program?

Now, we take the metric that is the solution to the linear program (way) above. We embed it into ℓ_1 space.

Now the ratio

$$\frac{\sum_e d(e)}{\sum_(i,j) d(ij)}$$

is preserved to within an $O(\log n)$ factor in the embedding. This is $1/V$ where V is value of the LP solution ($\sum_e d(e) = 1$.) That is

$$\frac{\sum_e h(e)}{\sum_(u,v) h(u,v)} \leq \frac{O(\log n)}{V}$$

for the ℓ_1 embedding $h(\dot{)}$.

Question: Think about coordinates?

$$\frac{\sum_c \sum_e h_c(e)}{\sum_c \sum_(u,v) h_c(ij)} \leq \frac{O(\log n)}{V}$$

Thus, one coordinate c also achieves this ratio.

$$\frac{\sum_e h_c(e)}{\sum_(u,v) h_c(ij)} \leq \frac{O(\log n)}{V}$$

Question: Cuts?

Now, we can consider all the cuts induced by scanning along this coordinate! We can rewrite $h(u,v) = \sum$cuts S that separate u and v δ_S, where δ_S is the distance associated with the empty space along this coordinate for this cut.

And, rewrite

$$\sum_{u,v} h(u,v) = \sum_{cuts S} \delta_S |S||\overline{S}|$$

and

$$\sum_e h(e) = \sum_{cuts S} \delta_S c(S, \overline{S}).$$

where $c(S, \overline{S})$ is the number of edges in the cut.

Thus, the ratio above becomes

$$\frac{\sum_{cuts S} \delta_S c(S, \overline{S})}{\sum_{cuts S} \delta_S |S||\overline{S}|} = \frac{\Theta(\log n)}{V}.$$

Again, one term for one S works and we obtain

$$\frac{c(S, \overline{S})}{|S||\overline{S}|} = \frac{\Theta(\log n)}{V}$$

a cut S whose sparsity is upper bounded by the LP lower bound on sparsity times $O(\log n)$.

Question: What generalization?

Say you only get credit for separating some subset of k pairs instead of everyone? The above argument gives an $O(\log k)$ approximation for this generalized notion of sparsity.

Exercise 3: Do one of the following.

Exercise 3.1: Argue that a constant degree expander graph has an LP solution that is a factor of $\Theta(\log n)$ different from the sparsity of the graph. Hint: an LP solution is uniform on all edges, the sparsity is $\Theta(1/n)$.

Exercise 3.2: Compare eigenvector solution to the linear programming solution for the graph with nodes consisting of pairs (i, j) ($i \in \{0, k\}$ and $j \in \{0, k^2/4\}$) with edges $((i, j), (i + 1, j))$ and edges $((i, j), (i, 2j))$ and $((i, j), (i, 2j + 1))$. This is basically a grid where the rows are lines and the columns are binary trees; the cross product of the two graphs. Which solution highlights that smallest cut better?