COMPSCI 638: Graph Algorithms November 20, 2019

Lectures 25-26

Lecturer: Debmalya Panigrahi *Scribe: Kevin Sun*

1 Overview

In the last lecture, we showed that integrality gap of the linear programming relaxation of the sparsest cut problem is $\Omega(\log n)$, so no LP-based solution can have approximation ratio better than $O(\log n)$. In this lecture, we give an overview of an SDP-based approach that allows us to obtain an $O(\sqrt{\log n})$-approximation algorithm.

2 SDP for Sparsest Cut

Recall that in the sparsest cut problem, we are given an undirected graph $G = (V, E)$ on n vertices where each edge (i, j) has capacity $c(i, j) \geq 0$. Our goal is to find the sparsest cut, and we've seen that up to a constant factor, this is equivalent to finding $S \subset V$ that minimizes $\delta(S)/|S| \cdot |\overline{S}|$ where $\delta(S)$ denotes the capacity of edges crossing S and $\overline{S} = V \setminus S$.

In Lecture 7, we gave an LP-based $O(\log n)$-approximation algorithm, and in Lecture 24, we showed that the integrality gap of this LP is $\Omega(\log n)$. Thus, in order to achieve a better performance guarantee, we introduce the following approach given by Arora, Rao, and Vazirani [ARV09].

2.1 Negative Type Metrics

Our LP formulation was obtained through a series of relaxations: finding a cut is equivalent to finding an elementary cut metric. From here, we extended our search to cut metrics, which are equivalent to ℓ_1-metrics. Finally, we extended our search to all metrics; this problem is easy to solve, and every metric can be approximated by an ℓ_1-metric with $O(\log n)$ distortion.

In this section, we tighten this final relaxation: rather than searching over all metrics, we search over a class of metrics known as *negative type metrics*. These metrics are obtained by squaring a Euclidean metric, so they are also known as *ℓ_2^2-metrics*. Optimizing over negative type metrics is captured by the following semidefinite program, where each $i \in V$ corresponds to a vector $x_i \in \mathbb{R}^n$:

$$\text{(SDP): } \min \sum_{(i,j)\in E} c(i,j)\left\|x_i - x_j\right\|^2$$

$$\sum_{i,j} \left\|x_i - x_j\right\|^2 \geq 1$$

$$\left\|x_i - x_j\right\|^2 + \left\|x_j - x_k\right\|^2 \geq \left\|x_i - x_k\right\|^2 \quad \forall i, j, k \in V$$

$$\left\|x_i\right\|^2 \leq 1 \quad \forall i \in V$$

Triangle inequality constraints: Notice that (SDP) explicitly contains the constraints for the triangle inequality, in contrast to the SDP for maximum cut that we saw in Lecture 22. The reason for this is that without them, (SDP) would have an $\Omega(n)$ integrality gap (which is even worse than the LP).

Consider the cycle on n vertices, which has sparsity $\Theta(1/n^2)$. Without the triangle inequality, we can map each vertex uniformly on the unit circle. In this case, we have

$$\sum_{(i,j)\in E} \left\|x_i - x_j\right\|^2 \approx \left(\frac{2\pi}{n}\right)^2 \cdot n = \Theta\left(\frac{1}{n}\right).$$

On the other hand, the denominator of the sparsity obtained by our SDP solution is

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

Therefore, the sparsity achieved by the SDP is $\Theta(1/n^3)$, proving that the integrality gap of (SDP) without triangle inequality constraints is $\Omega(n)$.

By enforcing the triangle inequality on our ℓ_2^2-metric, the resulting vectors cannot be too far apart. More specifically, there cannot be an obtuse angle formed between any two vectors. To maximize the number of such vectors in d-dimensional space, it can be shown that the endpoints of these vectors are the corners of the d-dimensional hypercube.

Rounding to a cut: We now describe how we round a solution of (SDP) to a sparse cut. Intuitively, if we round according to some random procedure, we want the probability edge (i,j) is cut to be roughly $\left\|x_i - x_j\right\|^2$. We also want both sides of our cut to have roughly the same size.

To formalize this intuition, we first establish some notation. Suppose we have an optimal solution $x_1, \ldots, x_n$ of (SDP). Let i be a vertex and A be a subset of vertices. Then $d(i,A) = \min_{j\in A} \left\|x_i - x_j\right\|^2$ denotes the distance from i to A, and $B(A,r)$ denotes the set of vertices whose distance from A is at most r, i.e., $B(A,r) = \{i \in V : d(i,A) \le r\}$. Ideally, we'd like to find a large subset A of vertices whose distance to other vertices is also large. In this case, we can pick a random superset of A as our cut; the following lemmma formalizes this idea.

Lemma 1. *Suppose there exists $A \subseteq V$ such that $|A| \ge \alpha n$ and $\sum_{i\in V} d(i,A) \ge \beta/n$. Let $r \in (0,1)$ be the parameter that minimizes the sparsity of the cut $B(A,r)$. Then $B(A,r)$ is an $O(1/\alpha\beta)$-approximation for the sparsest cut problem.*

Proof. We prove this as follows: choose a radius r uniformly at random in the interval $(0,1)$ and consider an edge (i,j) such that $d(i,A) \le d(j,A)$. Since r is chosen uniformly at random, the probability that (i,j) is cut is given by the following:

$$\Pr\left((i,j) \in \delta(B(A,r))\right) = \Pr(d(i,A) \le r \le d(j,A)) \le \left\|x_i - x_j\right\|^2, \tag{1}$$

where the inequality holds because our solution to (SDP) satisfies the triangle inequality. Thus, we can bound the expectation of the numerator in our sparsity:

$$\mathbb{E}\left[\delta(B(A,r))\right] \le \sum_{(i,j)\in E} c(i,j)\left\|x_i - x_j\right\|^2. \tag{2}$$

Now we bound the expectation of the denominator in the sparsity expression. This term is the sum of all pairwise distances; instead of summing over all pairs, we can sum one index over the vertices in A. In other words, we have the following:

$$\begin{aligned}\mathbb{E}\left[\sum_{i,j\in V}\left\|x_i - x_j\right\|^2\right] &\geq \sum_{i\in A}\sum_{j\in V}\Pr(j \notin B(A,r)) \\ &= |A|\sum_j \Pr(r \leq d(j,A)) \\ &= |A|\sum_j d(j,A) \\ &\geq \alpha\beta.\end{aligned}$$

where the first inequality follows from (1) and the final inequality follows from the assumptions in the lemma. By combining this with (2), we have shown that if r is chosen uniformly at random, then the ratio of the expectations is at most $1/\alpha\beta$ times the optimal (SDP) value. Thus, there must exist some radius in $(0,1)$ that achieves this sparsity, as desired. □

To turn Lemma 1 into our algorithm, we must consider two cases. The easier case is when there exists a ball known as a "dense core." In this case, this ball can act as the A in Lemma 1 and we can show that α and β are both constants, so our algorithm is actually $O(1)$-approximate. In the other case, there is no such ball; the analysis of this case is trickier and relies on deeper structural properties of our solution.

Case 1 (Dense core exists): Suppose there exists a vertex $i^* \in V$ such that the ball centered at i^* with radius $1/8n^2$ contains at least $n/4$ vertices. In this case, we set A as this ball and apply Lemma 1 so that $\alpha = 1/4$. Now for any two vertices i, j outside A, we have

$$d_{ij} - \frac{1}{4n^2} \leq d(i,A) + d(j,A)$$

by the triangle inequality and the fact that the radius of A is $1/8n^2$. Summing over all pairs $i, j \in V$ and noting $\sum_{i,j} d_{ij} \geq 1$ because our solution is feasible, we see that

$$1 - \frac{1}{8} \leq \sum_{i,j\in V}\left(d_{ij} - \frac{1}{4n^2}\right) \leq n\sum_{i\in V} d(i,A),$$

so we can set $\beta = 7/8$. Since α and β are both constants, by Lemma 1, we have a constant-approximate solution to the sparsest cut problem.

Case 2 (No dense core): Intuitively, we cannot easily find a candidate for the set A in Lemma 1, but we *can* find *two* large subsets that are far apart from each other. We then set A as one of these subsets so that α is a constant while β depends on how large the distance is between S and T.

Lemma 2. *If no dense core exists, then there exist $S, T \subseteq V$ such that $|S|, |T| = \Omega(n)$ and $d(S,T) \geq \Delta/n^2$ where $\Delta = \Theta(1/\sqrt{\log n})$.*

Before sketching the proof of this claim, we first formalize the intuition stated above. Set $A = S$ for Lemma 1, so that $\alpha = \Omega(1)$ and we have the following:

$$\sum_{i\in V} d(i,A) \geq \sum_{i\in B} d(i,A) = \Omega(n)\cdot\frac{\Delta}{n^2} = \Omega\left(\frac{\Delta}{n}\right).$$

Thus, we can set $\beta = \Delta = \Theta(1/\sqrt{\log n})$ to obtain an $O(\sqrt{\log n})$-approximate solution.

Proof sketch of Lemma 2. Intuitively, we want vectors corresponding to S to be far apart from the vectors corresponding to T. Recall that all of the vectors in the solution to (SDP) are contained in the n-dimensional unit ball. To find S and T, we will construct what is known as a *fat hyperplane*. Then S and T contain the vectors lying on the two sides of the hyperplane. (As we saw in Lecture 22, the rounding scheme for the maximum cut problem uses a thin hyperplane.)

If the hyperplane is very fat, then $d(S, T)$ will be large, but if it is too fat, then $|S|$ and $|T|$ will be too small. To strike this balance, we randomly pick a hyperplane-defining vector according to a Gaussian distribution, and with high probability, we'll have $|S|, |T| = \Omega(n)$.

However, unless we increase the width of the hyperplane, $d(S, T)$ might be too small. To remedy this, we iteratively remove violating pairs $(s, t) \in S \times T$, i.e., such pairs satisfy $d(s, t) < \Delta/n^2$ until no violating pairs remain. The remaining subsets are sufficiently separated from each other, but now they might be too small.

Notice that we can think of our pair-removal process as removing a maximal matching from a bipartite graph. The size of this matching is roughly $c_1 n$ for some constant c_1, and the closest distance between any (s, t) pair is $c_2\Delta/n$ for some constant c_2. As we decrease c_2 by iteratively removing violating pairs, the bipartite graph becomes sparser, so c_1 also decreases.

Importantly, both c_1 and c_2 are independent of S and T. Thus, at some point, we can decrease these two constants until the value of c_1 is smaller than the constants that appear in the $|S|, |T| = \Omega(n)$ guarantee. At this point, we can safely remove the remaining vertices without asymptotically decreasing $|S|$ and $|T|$, whose values remain sufficiently large.

□

3 Summary

In this lecture, we concluded our study of the sparsest cut problem. In particular, we showed how we can overcome the $\Omega(\log n)$ integrality gap of the LP relaxation via an SDP relaxation. We then sketched the algorithm of Arora, Rao and Vazirani [ARV09] that is an $O(\sqrt{\log n})$-appoximation.

References

[ARV09] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.