Lecture 7

Lecturer: Debmalya Panigrahi   *Scribe: Kevin Sun*

1 Overview

In this lecture, we continue studying the sparsest cut problem. We will give an approximation algorithm that uses an embedding theorem concerning general metrics and conclude with some discussion about the current state of the problem.

2 Sparsest Cut

Recall the (uniform) sparsest cut problem: given a graph $G = (V, E)$ with edge capacities $u(e)$, find $S \subset V$ that minimizes the sparsity $u(S)/|S| \cdot |V \setminus S|$, where $u(S)$ denotes the total capacity of edges crossing S. Before proceeding, let us define a few metrics on V that are critical for our algorithm.

1. A metric d on V is an *elementary cut metric* if there exists $S \subset V$ such that

$$d(x,y) = \begin{cases} 1 \text{ if } |S \cap \{x,y\}| = 1 \\ 0 \text{ otherwise.} \end{cases}$$

2. A metric d on V is a *cut metric* if there exist $\lambda_1, \ldots, \lambda_k \geq 0$ and elementary cut metrics $d^{(1)}, \ldots, d^{(k)}$ on V for some $k \geq 0$ such that for every $x, y \in V$,

$$d(x,y) = \sum_{i=1}^{k} \lambda_i d^{(i)}(x,y).$$

3. A metric d on V is an ℓ_1-*metric* if there exists $h \geq 1$ and $f : V \to \mathbb{R}^h$ such that, for every $x, y \in V$,

$$d(x,y) = \sum_{i=1}^{h} |f(x)_i - f(y)_i|.$$

Notice that the sparsest cut problem is equivalent to finding an elementary cut metric d that minimizes

$$\phi(d) = \frac{\sum_{x,y} u(x,y)d(x,y)}{\sum_{x,y} d(x,y)}. \tag{1}$$

And as we saw in the previous lecture, the LP relaxation of this problem is

$$\text{(LP-SC): } \min \sum_{x,y} u(x,y)d(x,y)$$

$$\sum_{x,y} d(x,y) \geq 1$$

$$d \text{ is a metric}$$

This relaxation is equivalent to minimizing (1) over *all* metrics, not just elementary cut metrics. Now suppose we extend our search space from elementary cut metrics to cut metrics—the optimal value of (1) can only decrease. However, our first lemma states that this does not actually happen.

Lemma 1. *Let ϕ_1, ϕ_2 denote the minimum values of (1) over elementary cut metrics and cut metrics, respectively. Then $\phi_1 = \phi_2$.*

Proof. The inequality $\phi_1 \geq \phi_2$ holds because the set of cut metrics includes all elementary cut metrics. Now let d be the cut metric attain the value of ϕ_2, that is,

$$\phi_2 = \phi(d) = \frac{\sum_{x,y} u(x,y) d(x,y)}{\sum_{x,y} d(x,y)}.$$

Since d is a cut metric, we know there exist $\lambda_1, \ldots, \lambda_k$ and elementary cut metrics $d^{(1)}, \ldots, d^{(k)}$ on V for some $k \geq 1$ such that

$$d(x,y) = \sum_{i=1}^{k} \lambda_i d^{(i)}(x,y)$$

for every $x, y \in V$. Applying this substitution and reordering the summations, we get

$$\phi_c = \frac{\sum_{x,y} u(x,y) \sum_i \lambda_i d^{(i)}(x,y)}{\sum_{x,y} \sum_i \lambda_i d^{(i)}(x,y)}$$

$$= \frac{\sum_i \lambda_i \sum_{x,y} u(x,y) d^{(i)}(x,y)}{\sum_i \lambda_i \sum_{x,y} d^{(i)}(x,y)}$$

$$\geq \min_i \frac{\sum_{x,y} u(x,y) d^{(i)}(x,y)}{\sum_{x,y} d^{(i)}(x,y)},$$

where the inequality holds due to a standard averaging argument. The index i that achieves this minimum ratio gives us an elementary cut metric $d^{(i)}$ such that $\phi_2 \geq \phi(d^{(i)}) \geq \phi_1$, as desired. □

So Lemma 1 tells us that when solving the sparsest cut problem, we can extend our search space from elementary cut metrics to cut metrics without changing the optimal objective value. Our next lemma allows us to convert, in both directions, between cut metrics and ℓ_1-metrics.

Lemma 2. *For every cut metric d on V, there exists an ℓ_1-metric d' on V such that all distances are exactly preserved, and vice versa. In other words, cut metrics are isometric to ℓ_1-metrics.*

Proof. Let d be a cut metric, so $d = \sum_{i=1}^{k} \lambda_i d^{(i)}$ for some $\lambda_1, \ldots, \lambda_k \geq 0$ and elementary cut metrics $d^{(1)}, \ldots, d^{(k)}$ corresponding to sets $S_1, \ldots, S_k$. Following the definition of ℓ_1-metric, let $h = k$, fix some $i \in \{1, \ldots, h\}$, and consider $f : V \to \mathbb{R}^h$ defined as

$$f(x)_i = \begin{cases} \lambda_i \text{ if } x \in S_i \\ 0 \text{ otherwise.} \end{cases}$$

Then notice $|f(x)_i - f(y)_i| = \lambda_i$ if $|S \cap \{x,y\}| = 1$ and 0 otherwise, so

$$d(x,y) = \sum_{i=1}^{k} \lambda_i d^{(i)}(x,y) = \sum_{i=1}^{h} |f(x)_i - f(y)_i|.$$

Conversely, suppose d is an ℓ_1-metric with $h \geq 1$ and $f : V \to \mathbb{R}^h$ as given in the corresponding definition. For any set $S \subseteq V$, we let δ_S denote the elementary cut metric induced by S. For simplicity, let us first assume $h = 1$ and rename the points of $V = \{x_1, \ldots, x_n\}$ so that

$$f(x_1) \leq f(x_2) \leq \cdots \leq f(x_n).$$

We also let S_i and α_i denote the following:

$$S_i = \{x_1, \ldots, x_i\} \quad \forall i \in \{1, \ldots, n-1\}$$
$$\alpha_i = f(x_{i+1}) - f(x_i) \geq 0 \quad \forall i \in \{1, \ldots, n-1\}.$$

Notice that for any $p < q$, we have

$$d(x_p, x_q) = \left| f(x_p) - f(x_q) \right| = \alpha_p + \cdots + \alpha_{q-1} = \sum_{i=1}^{n-1} \alpha_i \delta_{S_i}(x_p, x_q).$$

Thus, for $h = 1$, we have shown that d can be written as the nonnegative linear combination of elementary cut metrics, so d is a cut metric. We can extend this to higher values of h by performing the above procedure in every coordinate, and summing the results. In other words, for each of the h coordinates, we sort V to obtain $n - 1$ elementary cut metrics and the corresponding α coefficients; the final cut metric is defined as the linear combination of all $(n-1)h$ elementary cut metrics. $\square$

The isometry of cut metrics and ℓ_1-metrics is useful because, as we shall see, *any* metric can be "approximated," in some sense, by an ℓ_1-metric. Once we have an ℓ_1 metric, Lemma 2 allows us to convert it to a cut metric, and as we saw in Lemma 1, we can then convert this cut metric to an elementary cut metric. Before formalizing all of this, let us define a way to quantify how well one metric approximates another.

Definition 1. *Let d, d' be metrics on V, and suppose α, β are values that satisfy*

$$\frac{d(x,y)}{\alpha} \leq d'(x,y) \leq \beta d(x,y)$$

for every $x, y \in V$. Then we say d is embeddable *into d' with* distortion $\alpha\beta$.

We now state the final ingredient needed for our sparsest cut algorithm; this theorem allows us to obtain the ℓ_1 "approximation" metric mentioned above. The existence portion of this result is due to Bourgain [Bou85], and the construction was given by Linial et al. [LLR95].

Theorem 3 ([Bou85], [LLR95]). *Any metric on n points is embeddable into an ℓ_1-metric in $O(\log^2 n)$ dimensions with $O(\log n)$ distortion. Furthermore, this ℓ_1-metric can be found with high probability in $\text{poly}(n)$ time.*

Although the proof of Theorem 3 is not particularly difficult, we omit it for the sake of brevity. With all of its ingredients in place, we now formally state a sparsest cut algorithm.

Algorithm 1 Sparsest Cut via an ℓ_1 embedding

 1: Solve (LP-SC) to obtain a general metric d_1.
 2: Apply Theorem 3 on d_1 to obtain an ℓ_1-metric d_2.
 3: Find a cut metric d_3 that is isometric to d_2 via Lemma 2.
 4: Enumerate over the support of d_3 to find an elementary cut metric that minimizes $\phi(\cdot)$; call this elementary cut metric d_4.
 5: Return the cut corresponding to d_4.

Theorem 4. *Algorithm 1 is a polynomial-time $O(\log n)$-approximation for the sparsest cut problem.*

Proof. For any metric d, let $V(d) = \sum_{x,y} u(x,y)d(x,y)$ denote the objective attained by d in (LP-SC). Furthermore, let D^* denote the optimal value of (LP-SC) and let OPT denote the sparsity of the sparsest cut, so we have $V(d_1) = D^* \leq OPT$.

By Theorem 3, the distortion between d_1 and d_2 is $O(\log n)$, and by Lemma 2, we know that d_2 and d_3 are isometric. Finally, by Lemma 1, we know $\phi(d_4) \leq \phi(d_3)$. Putting this all together, we have the following:

$$\phi(d_4) \leq \phi(d_3) \leq V(d_3) = V(d_2) = O(\log n) \cdot V(d_1) = O(\log n) \cdot OPT,$$

so d_4 is an $O(\log n)$-approximate elementary cut metric, as desired.

Now we analyze the running time of Algorithm 1. The linear program (LP-SC) has polynomial size, so Step 1 runs in polynomial time, and the subsequent two steps run in polynomial time due to Theorem 3 and Lemma 2. In Step 4, we enumerate over $(n-1)h$ dimensions, where h is the dimension of the ℓ_1 space. By Theorem 3, we know $h = O(\log^2 n)$, so the enumeration is over $O(n \log^2 n)$ items. Since computing $\phi(d)$ can be done in polynomial time for any metric d on V, the entire algorithm runs in polynomial time. $\qquad\square$

We conclude by briefly discussing the non-uniform version of sparsest cut. In this problem, every pair $x, y \in V$ has some "demand" quantity $D(x,y)$, and the sparsity of $S \subset V$ becomes $u(S)/D(S)$ where $D(S)$ is the total demand of pairs separated by S. As we can see, in the uniform sparsest cut problem, every pair has unit demand. All of the results in this section extend in a straightforward manner to the non-uniform sparsest cut problem.

2.1 The Integrality Gap of (LP-SC)

We now briefly discuss the integrality gap of (LP-SC) and its implications. Recall that in the minimum *s-t* cut problem (see Lecture 2), the linear programming relaxation is integral, that is, there exists an optimal integral solution. Thus, there is no "gap" between the optimal solution and the optimal solution of the linear program.

In contrast, the linear programming relaxation for the sparsest cut problem is not integral. In fact, there exist instances of the problem in which optimal value of (LP-SC) is roughly a $\log n$ factor smaller than the sparsity of the sparsest cut.

This implies that for this problem, an $O(\log n)$-approximation is essentially the best algorithm this relaxation can generate. Letting ALG denote the output of any algorithm, OPT denote the optimal solution value, and LP denote the optimal value of (LP-SC), we have

$$LP \leq OPT \leq ALG.$$

As mentioned above, there are instance in which $\log n \cdot LP \leq OPT$. For these instances, if we prove that our algorithm is an α-approximation by using LP as a surrogate for OPT, then

$$\log n \leq \frac{OPT}{LP} \leq \frac{ALG}{LP} \leq \alpha.$$

In the early 2000's, a new approach to the sparsest cut problem was developed using a technique known as semidefinite programming, a generalization of linear programming. This reduced the $\log n$ integrality gap to $O(\sqrt{\log n})$. On the other hand, it has been shown that a polynomial-time constant-approximation is unlikely to exist.

3 Summary

In this lecture, we saw another algorithm for the sparsest cut problem. We showed that by applying of sequence of transformations and metric embeddings, we can obtain an $O(\log n)$-approximation in polynomial time. Finally, we discussed the LP integrality gap for this problem and its implications in the design of approximation algorithms.

References

[Bou85] Jean Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52(1-2):46–52, 1985.

[LLR95] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.