

Lecture 8

Lecturer: Debmalya Panigrahi

Scribe: Allen Xiao

## 1 Overview

In the previous lecture, we defined a series of edge parameters which we proved to lower bound connectivity. This lecture, we use connectivity to construct a graph sparsifier which preserves cuts via sampling. Implementation and time complexity will be discussed next lecture.

## 2 Cut Sparsifier

Before we state the main theorem, we remind the reader of the definition of connectivity:

**Definition 1** (Connectivity). *The connectivity  $\lambda_e$  of edge  $e$  is the size of the smallest cut containing  $e$ .*

The main theorem gives a construction for a graph sparsifier which preserves cuts.

**Theorem 1** (Fung and Harvey [FH10], Hariharan and Panigrahi [HP10]). *Consider graph  $G = (V, E)$  simple and undirected, with  $n = |V|$ . Let  $c$  be a large enough constant and  $\epsilon > 0$ . Let  $G_\epsilon$  be a weighted subgraph where every edge  $e \in E$  is sampled with probability*

$$p_e \geq \min \left( c \cdot \frac{\log^2 n}{\epsilon^2 \lambda_e}, 1 \right)$$

and included in  $G_\epsilon$  with weight  $1/p_e$  if selected. Then:

1. With high probability, all cuts  $S \subseteq V$  are preserved in weight:

$$G_\epsilon(S, \bar{S}) \in (1 \pm \epsilon)G(S, \bar{S})$$

2. The expected size of  $E(G_\epsilon)$  satisfies:

$$|E(G_\epsilon)| \leq \sum_{e \in E} c \cdot \frac{\log^2 n}{\epsilon^2 \lambda_e} = O \left( \frac{n \log^2 n}{\epsilon^2} \right)$$

The bound on expected size follows from conductance  $\leq$  connectivity. To prove the theorem, we must show that the weight of every cut in  $G_\epsilon$  concentrates to its expectation (size of the same cut in  $G$ ) under this sampling scheme.

### 3 Proving Concentration

#### 3.1 Concentration of a single cut

Consider a single cut  $S$  with  $\Delta$  cut edges in  $G$ . Sample each edge  $e \in S$  with probability:

$$p_e \geq c \cdot \frac{\log^2 n}{\varepsilon^2 \Delta} \geq c \cdot \frac{\log^2 n}{\varepsilon^2 \lambda_e}$$

The second inequality follows as  $\lambda_e \leq \Delta$ .

We rescale the weight of each  $e \in S$  (uniformly) by  $\Delta$  so that the expected weight of  $S$  in  $G_\varepsilon$  is  $\mu = 1$ .

Applying a Chernoff bound (with some constant  $\gamma$ ):

$$\begin{aligned} \Pr[G_\varepsilon(S, \bar{S}) \notin (1 \pm \varepsilon)G(S, \bar{S})] &\leq e^{-\gamma \varepsilon^2 \mu} \\ &= e^{-\gamma \varepsilon^2 \Delta p_e} \\ &= e^{-\gamma \varepsilon^2 \Delta \frac{\log^2 n}{\varepsilon^2 \Delta}} \\ &= e^{-O(\log^2 n)} \\ &= n^{-O(\log n)} \end{aligned}$$

We note that we could have achieved  $O(1/n)$  without one of the  $\log n$  factors in  $p_e$ . Regardless, this only shows concentration for a single cut will not suffice to show concentration over all the (exponentially many) cuts in  $G$  at once when we take the union bound.

The following example helps highlight some of the problems with this analysis:

**Example 1.** Consider the following graph:

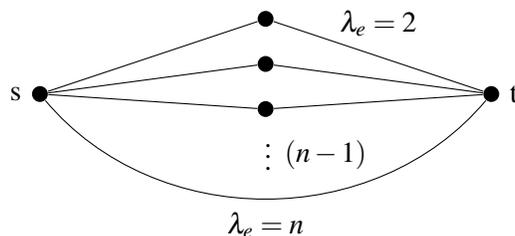


Figure 1: The degree cut of the middle vertices gives connectivity 2. All cuts including the bottom edge require all  $(n - 1)$   $s$ - $t$  paths above.

The edges in any of the 2-edge paths have connectivity  $\lambda_e = 2$  and are always sampled, as  $p_e = 1$ . The single  $s$ - $t$  edge has connectivity  $\lambda_e = n$  and has much lower sample probability ( $p_e = \log^2 n/n$ ). When that bottom edge is missing, all the  $n$ -cuts which include it lose  $O(1)$ , so we still expect this scheme to concentrate.

However, our analysis above uses

$$p_e \geq \frac{\log^2 n}{n}$$

which is much less than the actual sampling probability of the middle edges, concluding that the concentration probability is  $n^{-O(1)}$ . In essence, the analysis assumes much higher variance that we actually have so the Chernoff bound doesn't give us much.

### 3.2 Grouping edges by sampling probability

Somehow, we should use the real sampling probabilities. However, these aren't homogeneous across edges. We'll partition the edges by  $\lambda_e$  into doubling ranges, so that within each range the  $p_e$  varies by most a factor of 2.

$$R_i = \{e \mid \lambda_e \in [2^{i-1}, 2^i)\} \quad 1 \leq i \leq \log n$$

Each class  $R_i$  is relatively homogeneous in  $p_e$ , so for each class we will use a tail bound to show concentration, then take a union bound over the  $\log n$  classes.

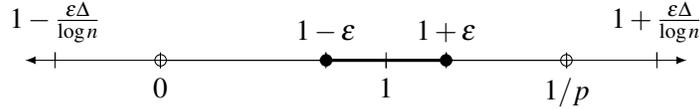


Figure 2: A singleton class will have all of its probability on the open points.

There is a subtle problem, however: If there are classes with just one edge, they have zero probability of concentrating (see Figure 2). To fix this, we relax the error allowed per class to be proportional to the cut size:

$$\pm \varepsilon \longrightarrow \pm \varepsilon \frac{\Delta}{\log n}$$

To see why we chose this, observe the following result on the concentration of sums:

**Theorem 2.** Let  $X_1, \dots, X_n$  be a set of independent random variables where

$$X_i = \begin{cases} 1/p_i & \text{w.p. } p_i \\ 0 & \text{w.p. } 1 - p_i \end{cases}$$

The for  $p_i \geq p$  and any  $N \geq n$

$$\Pr \left[ \sum_{i=1}^n X_i \notin n \pm \varepsilon N \right] \leq e^{-c\varepsilon^2 pN}$$

Applying to any class (including a singletons) with connectivity  $\lambda_e = k$  in a cut of size  $\Delta$ :

$$\begin{aligned} \Pr \left[ G_\varepsilon(\Delta, k) \notin G(\Delta, k) \pm \frac{\varepsilon \Delta}{\log n} \right] &\leq e^{-c\varepsilon^2 \cdot \frac{\log^2 n}{\varepsilon^2 k} \cdot \frac{\Delta}{\log n}} \\ &= n^{-O(\Delta/k)} \end{aligned}$$

The notation we used here to represent the edge classes was that for *cut projections*, which we will now define. These cut projections will act as an analytical grouping (in the same way as the doubling classes) inside which cuts concentrate with fixed error.

**Definition 2** ( $(\alpha, \beta)$  cut projection). An  $(\alpha, \beta)$  cut projection is a cut of size  $\alpha\beta\lambda$  where each edge has connectivity  $\lambda_e \geq \beta\lambda$ .

**Lemma 3.** The number of  $(\alpha, \beta)$  cut projections in an undirected graph is  $n^{O(\alpha)}$ .

**Corollary 4.** If (3) holds, all cuts concentrate with high probability in  $G_\varepsilon$ .

*Proof.* Consider the case where  $\alpha = \Delta/k$  and  $\beta = k/\lambda$ . Under this definition, this is a cut projection of a cut is size  $\Delta$ , where each edge has connectivity at least  $k$ .

As we showed above, each cut projection fails to concentrate with probability at most  $n^{-O(\Delta/k)}$ . Simultaneously, (3) implies that the number of such projections is  $n^{O(\Delta/k)}$ . Taking the sum, these projections (over  $\Delta, k$ ) fail with probability at most:

$$\begin{aligned} \Pr[\text{every such } \Delta, k \text{ projection fails to concentrate}] &\leq \frac{n^{c_1 \cdot \Delta/k}}{n^{c_2 \cdot \Delta/k}} \\ &= n^{c_1 - c_2} \\ &\leq n^{-3} \end{aligned}$$

The last inequality follows because we can control the constant  $c_2$  by manipulating the sampling probability  $p_e$ , whereas  $c_1$  is set by our analysis in the proof of (3).

Taking a union bound over possible choices of  $\Delta, k$ :

$$\sum_{\Delta} \sum_k n^{-3} \leq n(n(n^{-3})) = \frac{1}{n}$$

So all cuts concentrate simultaneously with high probability. □

This completes the proof of the main theorem. The next section will focus on proving (3).

## 4 Counting Projections

As a sanity check, we'll examine the case where  $\beta = 1$ .

**Example 2.** In cuts of size  $\Delta = \alpha\beta\lambda = \alpha\lambda$ , every edge is included in the projection as  $\lambda_e \geq \lambda$ . This case is essentially identical to the one in Karger's original contraction algorithm, where he provides a bound on the number of  $\alpha$ -min cuts. The analysis for that proved this was  $n^{O(\alpha)}$ , as we expect.

In order to prove the bound on the number of cut projections, we use a version of Karger's contraction algorithm. For any fixed  $\beta$ , this algorithm preserves the cut projection with probability  $\Omega(n^{-2\alpha})$ . The total probability over all possible  $\beta$  must be  $\leq 1$ , so by applying a union bound over  $\beta$  we can upper bound the number of such projections.

Some tools needed for the modified contraction algorithm follow.

**Definition 3** (splitting-off). *Let  $v$  be a vertex with even degree  $d_v$ . To split off  $v$ , we define a pairing over its neighbors and for pair  $(u_i, u_j)$ , replace edges  $(v, u_i)$  and  $(v, u_j)$  with  $(u_i, u_j)$ .*

*This can be extended to odd-degree vertices. For our purposes, it is sufficient to double every edge in the graph and make every  $d_v$  even.*

We observe that splitting off can only decrease connectivity. A theorem by Mader, stated without proof, provides existence for pairings which preserve connectivity.

**Theorem 5** (Mader [Mad82]). *When splitting off  $v$  with sufficiently high degree ( $d_v > 3$ ), there exists a pairing such that all remaining  $\lambda_e$  are preserved.*

We also define *light* and *heavy* edges/vertices.

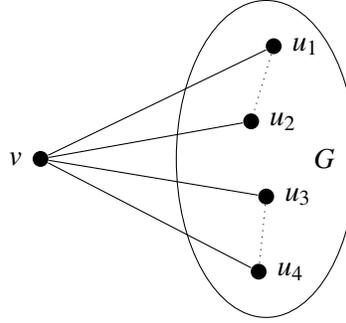


Figure 3: When we split off  $v$ , we choose a pairing of  $v$ 's neighbors, then replace all adjoining edges (solid) with edges between neighbor pairs (dotted).

**Definition 4** ( $k$ -light and  $k$ -heavy). An edge  $e$  is  $k$ -light if  $\lambda_e \leq k$ . A vertex  $v$  is  $k$ -light if all incident edges are  $k$ -light. A vertex or edge is  $k$ -heavy if it is not  $k$ -light.

The following observation follows from the degree cut of a  $k$ -heavy vertex.

**Fact 6.** Any  $k$ -heavy vertex has degree  $\geq k$ .

*Proof.* Consider the degree cut of a  $k$ -heavy vertex  $v$ . Obviously, the size of this cut is  $\geq \lambda_e$  for any adjoining edge  $e$ . If  $d_v < k$ , then every adjoining  $e$  is  $k$ -light. It follows that  $v$  is  $k$ -light by definition, contradicting our choice of  $v$ . Therefore  $v$  must have degree at least  $k$ .  $\square$

---

**Algorithm 1** modified contraction algorithm

---

- 1: **split off** all  $k$ -light vertices
  - 2: **repeat**
  - 3:     **contract** a uniform random edge, removing self loops
  - 4:     **split off** all  $k$ -light vertices
  - 5: **until**  $\alpha$  vertices are left
  - 6: **return** a random cut
- 

Splitting off controls the size of the graph at each contraction. Now, a similar analysis (which proved the cut counting lemma) gives us a proof of (3).

*Proof.* What is the probability that a fixed  $(\Delta/k, k/\lambda)$  cut projection is output by this algorithm? Once we bound this, a union bound on this tells us an upper bound on the number of possible  $(\Delta/k, k/\lambda)$  cut projections, since the sum of probabilities  $\leq 1$ .

We start with  $n = |V|$  vertices. Let the number of vertices after the  $i$ th splitting-off be  $n_i$ . (6) implies that there are at least  $(kn_i/2)$  edges remaining. The projection edges are preserved if no edge of the cut is contracted:

$$\begin{aligned}
 \Pr[\text{no cut edge is contracted on round } i] &\geq 1 - \Delta/(kn_i/2) \\
 &= 1 - \frac{2(\Delta/k)}{n_i} \\
 &= 1 - \frac{2\alpha}{n_i}
 \end{aligned}$$

Going through the whole loop (suppose we stop on some  $n_t$ :

$$\begin{aligned} \Pr[\text{no cut edge contracted by end}] &\geq \left(1 - \frac{2\alpha}{n}\right) \left(1 - \frac{2\alpha}{n_1}\right) \left(1 - \frac{2\alpha}{n_2}\right) \cdots \left(1 - \frac{2\alpha}{n_t}\right) \\ &\geq \left(1 - \frac{2\alpha}{n}\right) \left(1 - \frac{2\alpha}{n-1}\right) \left(1 - \frac{2\alpha}{n-2}\right) \cdots \left(1 - \frac{2\alpha}{2\alpha+1}\right) \\ &= \binom{n}{2\alpha}^{-1} \end{aligned}$$

In the second inequality, we fill in the missing terms between  $n_i$  and  $n_{i+1}$  to make our telescoping product complete. In general, the number of edges across the can not increase by splitting off.

There are at most  $2^{2\alpha-1}$  cuts in the final graph of  $2\alpha$  vertices.

$$\begin{aligned} &\Pr[\text{algorithm outputs the projection}] \\ &= \Pr[\text{random output cut output} = \text{the projection's cut}] \\ &\geq \binom{n}{2\alpha}^{-1} 2^{-(2\alpha-1)} \\ &\geq n^{-2\alpha} \cdot 2^{-2\alpha} \\ &\geq n^{-2\alpha} \cdot n^{-2\alpha} \\ &= n^{-O(\alpha)} \\ &= n^{-O(\Delta/k)} \end{aligned}$$

This probability bound holds for any projection of connectivity at least  $k$  of a cut of size  $\Delta$ . Applying a union bound we find a bound on the number of projections.

$$\sum_{j=1}^C n^{-O(\Delta/k)} \leq 1 \implies C \leq n^{O(\Delta/k)}$$

□

## 5 Summary

We introduced an algorithm for constructing a cut sparsifier of a simple, undirected graph using sampling and weighting.

In Section 3, we showed how a naive analysis of the sampled edges would fail to prove our convergence properties. The key technique was to group edges by sample probability, bound each of these groups, and finally apply a union-style bound across all groups.

In Section 4, we provided a bound for the number of distinct groups by constructing a randomized algorithm and bounding its success probability for each group. Since these events were all disjoint, the sum over probabilities had sum at most 1. The algorithm was a variation on Karger's edge contraction algorithm, with extra vertex pruning steps which preserved connectivity.

This sparsifier was a 2010 result by (concurrently) Fung, Harvey, Hariharan, and Panigrahi. The original paper also provides an analysis of the algorithm implementation which we in the next few lectures, giving a near-linear time algorithm. Intuitively, we cannot use connectivities  $\lambda_e$  directly in our sampling probabilities, since computing them gives us a max flow problem for every  $\lambda_e$ . Instead, we use the Nagamochi-Ibaraki indices, which lower bound  $\lambda_e$  and have a linear time calculation.

## References

- [FH10] Wai Shing Fung and Nicholas J. A. Harvey. Graph sparsification by edge-connectivity and random spanning trees. *CoRR*, abs/1005.0265, 2010.
- [HP10] Ramesh Hariharan and Debmalya Panigrahi. A linear-time algorithm for sparsification of unweighted graphs. *CoRR*, abs/1005.0670, 2010.
- [Mad82] W. Mader. Konstruktion aller n-fach kantenzusammenhngenden digraphen. *European Journal of Combinatorics*, 3(1):63 – 67, 1982.