COMPSCI 638: Graph Algorithms	September 25, 2019
Lecture 9	
Lecturer: Debmalya Panigrahi	*Scribe: Kevin Sun*

1 Overview

In this lecture, we begin studying graph sparsification. We begin with the relatively simpler approach known as uniform sampling and examine its shortcomings. We then introduce some edge parameters of an undirected graph that we will use in future sparsification algorithms.

2 Graph Sparsification

The global minimum cut algorithms from Lecture 8 gave rise to a new area of study known as graph sparsification. In this problem, we are given an undirected graph $G = (V, E)$ and we want to find a graph H on V such that the value of any cut is roughly the same in G and H. At the same time, we want H to be relatively sparse, so that we can reduce the running time of algorithms that depend on the size of the edge set.

More formally, given a graph $G = (V, E)$ and $\epsilon \in (0, 1)$, we want to construct a graph H on V such that the following holds with high probability:

$$\forall S \subset V \quad (1-\epsilon)\delta_G(S) \leq \delta_H(S) \leq (1+\epsilon)\delta_G(S), \tag{1}$$

where $\delta_G(S)$ and $\delta_H(S)$ denote the total weight of edges crossing S in G and H, respectively. Note that an event occurs *with high probability* if its probability is at least $1 - 1/n^d$ for some constant d—the exact value is not important. We say a cut *deviates* if it does not satisfy (1).

2.1 Uniform Sampling

Let us construct H from G by fixing some p, sampling each edge with probability p, and if an edge is sampled, we set its weight in H to be $1/p$. In expectation, every edge in H has weight $p \cdot 1/p = 1$, which is a good start. However, if p is too low, then the edge set of H has high variance, so H is less likely to satisfy (1). On the other hand, if p is too high, then H might contain too many edges.

One way to find the "optimal" value of p is to apply a Chernoff bound.

Lemma 1 (Chernoff Bounds)**.** *Let $X_1, X_2, \ldots, X_n$ be independent random variables in $[0, 1]$, $\mu = \mathbb{E}\left[\sum_i X_i\right]$ denote the mean of the sum, and $\epsilon \in (0, 1)$ be a parameter. Then*

$$\Pr\left(\sum_{i=1}^{n} X_i \notin [1-\epsilon, 1+\epsilon]\mu\right) \leq 2\exp\left(-\frac{\epsilon^2\mu}{3}\right).$$

Example 1: Let's apply Lemma 1 to the following graph: $V = \{s, t\}$, and E is a set of λ parallel edges between s and t. We sample each edge with probability p and set the weight of sampled edges to $1/p$. Let W_e denote the weight of e in the sampled graph, and consider the random variable $pW_e \in [0,1]$. In this example, there is only one cut to preserve, and it has value λ. Applying Lemma 1 to pW_e, we see that the probability this cut is not preserved is

$$\Pr\left(\sum_{e\in E} pW_e \notin [1-\epsilon, 1+\epsilon]p\lambda\right) \leq 2\exp\left(-\frac{\epsilon^2 p\lambda}{3}\right) \leq \frac{1}{n^d},$$

where the second equality follows by setting $p = c \cdot \frac{\log n}{\lambda\epsilon^2}$ and c and d as suitable constants.

In this example, we preserved the only cut in the graph with high probability. In general, our analysis yields a sparsification algorithm (see Algorithm 1), but it only guarantees the following: for each cut, its value is preserved with high probability. However, we want the event "every cut is preserved" (i.e., (1)) to occur with high probability, which is a much stronger guarantee.

Algorithm 1 Uniform Sampling (Karger [Kar99])

Input: An undirected graph $G = (V, E)$ on n vertices with minimum cut value λ.

```
We will construct a graph H with vertex set V and edge set E_H.
for e ∈ E do
    Add e to E_H with probability p = c · log n/(λε²) for some constant c.
    if e was added to E_H then
        Set the weight of e in H to be 1/p.
return H = (V, E_H) as a sparsifier for G
```

To analyze the output of Algorithm 1, we might try extending the analysis of our previous example via the union bound over all cuts. However, there are roughly 2^n cuts while the probability that a cut deviates is only at most $1/n^d$. So in this case, the union bound implies that the probability that some cut deviates is at most $2^n/n^d$, which is not useful.

So instead, we invoke a cut-counting lemma that we saw in Lecture 8. Recall that a cut is an *α-minimum cut* if its size is at most $\alpha\lambda$, where λ denotes the size of the global minimum cut.

Lemma 2. *The number of α-minimum cuts in an undirected graph is at most $n^{2\alpha}$.*

Intuitively, Lemma 2 says that the number of small cuts in a graph is small. This will act as a more refined version of the union bound and ultimately prove the following theorem.

Theorem 3 (Karger [Kar99])**.** *The output of Algorithm 1 satisfies* (1) *with high probability.*

Proof. Let S be a cut with $\alpha\lambda$ edges and note that Lemma 1 implies

$$\Pr(S \text{ deviates}) \leq 2\exp\left(-\frac{\epsilon^2 p\alpha\lambda}{3}\right) = \frac{1}{n^{d\alpha}}$$

for suitably chosen constants c and d. We now apply Lemma 2 and take the union bound over all cuts in the graph, parameterized by α. This gives us

$$\Pr(\text{some cut of size } \alpha\lambda \text{ deviates}) \leq \frac{1}{n^{d\alpha}} \cdot n^{2\alpha} = \frac{1}{n^{(d-2)\alpha}} = \frac{1}{n^{2\alpha}},$$

where the last equality holds by setting d large enough. Finally, we apply the union bound over all cuts in the graph, parameterized by their value of α:

$$\Pr(\text{any cut deviates}) \leq \sum_{\alpha\geq 1} \frac{1}{n^{2\alpha}} = O\left(\frac{1}{n^2}\right). \qquad \square$$

2.2 Non-Uniform Sampling

Although Theorem 1 tells us that the output of Algorithm 1 preserves all cuts with high probability, it does not imply anything about the number of edges in the sparsifier. Consider, for example, the following example known as the dumbbell graph: two cliques, each on $n/2$ vertices, with one "middle" edge between them. In this graph, we have $\lambda = 1$ so Algorithm 1 samples every edge with probability 1, and the resulting graph is very dense.

To rectify this, we shall use non-uniform sampling: we'll sample each edge depending on how "critical" it is for us to do so. For example, it is very critical to sample the middle edge in the dumbbell graph, but sampling the other edges is not as critical.

Definition 1. *The* connectivity *of an edge $e = \{u, v\}$, denoted by λ_e, is the value of the maximum flow from u to v. Equivalently, it is the value of the minimum cut separating u and v.*

Let e be an edge with connectivity λ_e, so e is in some cut S that contains λ_e edges. According to Lemma 1, in order to preserve this cut, we need to sample *every* edge of S with probability at least

$$p_e \geq c \cdot \frac{\log n}{\epsilon^2 \lambda_e}.$$

for some constant c. Sampling at this rate gives us another sparsification algorithm, and in fact, it can be shown that $\sum_{e \in E} 1/\lambda_e \leq n - 1$, so the resulting graph contains $O(n \log n/\epsilon^2)$ edges. (Observe that this inequality is tight for any tree on n vertices.)

But now the other half of sparsification needs to be addressed: does sampling this way still preserve the value of every cut with high probability? Instead of answering this question directly, we shall consider another edge connectivity parameter introduced by Benczúr and Karger [BK15].

Definition 2. *The* strength *of an edge e in a graph G, denoted by s_e, is the largest minimum cut value of any vertex-induced subgraph of G that contains e.*

We first relate the strength of an edge with its connectivity.

Lemma 4. *For any edge in an undirected graph, its strength is at most its connectivity.*

Proof. Let $e = \{u, v\}$ be an edge and let S denote a cut containing e with λ_e edges. Now consider any induced subgraph of H that contains e. Then removing the edges crossing S from H would disconnect u from v, so the minimum cut value of H is at most λ_e. □

We conclude by examining the values of s_e and λ_e in two particular graphs.

Example 1: In the dumbbell graph (defined above), the strength of the middle edge is 1, and its connectivity is also 1. If each clique has $n/2$ vertices, then we claim that the strength of any other edge $e = \{u, v\}$ is $n/2 - 1$. First note that $\lambda_e = n/2 - 1$ because $\{u\}$ is a minimum u-v cut and it cuts $n/2 - 1$ edges. Furthermore, the clique containing e is an induced subgraph with minimum cut value $n/2 - 1$, so by Lemma 4, we have $s_e = n/2 - 1 = \lambda_e$.

Example 2: Consider the following graph on $n + 2$ vertices, which we denote by $K(1, 1, n)$: we have $V = \{s, v_1, \ldots, v_n, t\}$ and E contains the edges $\{s, v_i\}$, $\{v_i, t\}$ for every i, as well as the edge $\{s, t\}$ (see Fig. 1). Then any edge that is not $\{s, t\}$ has strength and connectivity both equal to 2. However, the connectivity of $\{s, t\}$ is $n + 1$ while its strength is only 2.

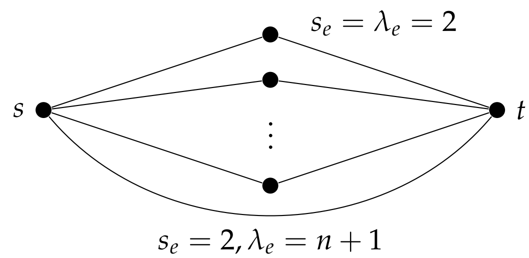

Figure 1: The graph $K(1,1,n)$: it contains n disjoint s-t paths and the edge $\{s,t\}$.

3 Summary

In this lecture, we began studying graph sparsification. We first saw how uniform sampling gives a proper sparsifier via a Chernoff bound, but the resulting graph may be rather dense. We then saw the notions of edge connectivity and strength, which will allow us to obtain a sparsifier without sampling as many edges.

References

[BK15] András A Benczúr and David R Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM Journal on Computing*, 44(2):290–319, 2015.

[Kar99] David R Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.