E0 325: Cuts, Flows and Linear Equations. Lecture 10. Feb 6 2012.
Instructor: Ramesh Hariharan Scribe: Ramesh Hariharan

Simple and Fast Graph Sparsification Maintaining Cuts

1 Introduction

In the previous class, we saw how random sampling can be used to obtain sparse graphs on which approximate min-cuts can be computed. Taking this idea one level further, Benczür and Karger [BK96] asked the following question: is there a way to obtain a sparse graph such that the size of every cut in the original graph is preserved approximately in the sampled graph. Clearly, the sparse graph so obtained must be a weighted graph. Benczur and Karger[BK96] proposed the following paradigm for creating such a sparse graph: pick each edge e independently with some carefully chosen probability p_e, and then give it weight $1/p_e$. Clearly, this scheme ensures that the expected size of any cut, post-sampling, equals its value in the original graph. However, it is unclear whether the cut-size distribution post-sampling is concentrated tightly enough around this expectation, a fact that is needed to ensure that the sizes of ALL cuts are preserved after sampling. Obtaining this tight concentration requires choosing p_e's appropriately.

So what are good values for p_e? If e is part of a large cut, p_e should be small, and vice-versa for small cuts. This suggests that p_e should be inversely proportional to either the connectivity between the endpoints of e, or to some quantity that is correlated (loosely speaking) with this connectivity. Next, consider a graph with two vertices s, t connected by several long disjoint paths. Since edge edge has constant connectivity, we would be tempted to pick each edge with some constant probability, say 1/2. Since the paths are long, it is very likely that post sampling, s and t become disconnected (which is not good because the value of any s-t cut has then changed dramatically). To address this problem, we sample at probabilities that are slightly higher, i.e., $p_e \propto \log n/k_e$, where k_e is some quantity correlated with the connectivity between the endpoints of e.

A few different k_e's have been considered in literature (connectivity, strong connectivity, electrical conductance etc). Note that sampling with probability $p_e \propto \log n/k_e$ requires knowledge of k_e for each edge e; however, in most of the above cases, computing k_e's requires very non-trivial algorithms (though doable in $\tilde{O}(m)$ time). The last section of these notes provides a brief survey of these various measures. In this lecture, we will look at a very simple and easily computable measure due to Hariharan and Panigrahi [HP10]. In particular, we will show the theorem below (for the definition of NI forests, recall the construction due to Nagamochi and Ibaraki [NI92] from previous classes, as described in Theorem 5; in a nutshell, create a family of edge-disjoint forests by processing edges in any order and inserting each edge into the very first forest that it can be inserted into without forming a cycle).

Theorem 1. *[HP10] Given an undirected, unweighted graph $G = (V, E)$, pick each edge e*

independently with probability $p_e = \min\{\frac{O(1)\log n}{\epsilon^2 NI_e}, 1\}$*, where* NI_e *is the index of the NI forest containing e (see Theorem 5), and give e weight* $1/p_e$ *if picked. Then, every cut in the resulting sampled graph has size* $(1 \pm \epsilon)$ *times its original size, with probability at least* $1 - \frac{1}{n}$*. The expected number of edges in the sampled graph is* $O(n\frac{\log^2 n}{\epsilon^2})$*. The time needed to construct this sampled graph is* $O(n + m)$*.*

Note that there are at at most n edges e with $\mathrm{NI}_e = 1$, another n edges with $\mathrm{NI}_e = 2$, and so on. It follows that the expected number of edges in the sampled graph is at most $O(n\frac{\log n}{\epsilon^2}\sum_{i=0}^{m} 1/i) = O(n\frac{\log^2 n}{\epsilon^2})$. The time needed to construct the sampled graph follows from Theorem 5. It remains to show that the size of every cut is roughly preserved by the sampling-weighting process. This proof will need two other interesting facts, each of which will be of independent interest as well; we will prove both these facts here.

The first fact *Edge Splitting* is a classic result due to Lovasz and Mader. Edge splitting refers to pairing together edges incident on a given vertex v and then taking each pair uv, vw and replacing the two edges in this pair by a single egde (u, w). The vertex v loses all its edges in the process and is then discarded from the graph. The edge splitting theorem says that there exists a pairing by which the connectivity between any pair of remaining vertices does not decrease (it is easy to note that connectivities cannot increase on account of the above operation), provided all vertices have even degree. Note that a bad pairing of the edges incident on v will not satisfy the above property (for instance, consider three vertices on path of length 2, where each path edge is actually a mutiedge with 2 parallel edges; the vertex v is the vertex in the middle; clearly, there is a pairing of edges incident on v which ensures that the terminal vertices retain their connectivity of 2, and there is a pairing where their connectivity drops to 0). The challenge is to show that there is always a good pairing. We will give a proof in this lecture. In particular, we will prove the following theorem. It is easy to see that this theorem can be applied repeatedly to pair up all edges incident on v.

Theorem 2. *Given an undirected, unweighted graph* $G = (V, E)$ *with all degrees being even, and given an edge* $e = uv$*, there exists an edge* vw *with the following property: for every pair of vertices* x, y*,* $x, y \neq v$*, the connectivity between* x, y *in* G' *equals the corresponding connectivity in G, where* $G' = (V, E - \{uv\} - \{vw\} \cup \{uw\})$

The second fact is an extension of the Karger's cut-counting theorem (see Theorem 6). Define the *k-projection* of a cut $C = (X, V - X)$ to be the subset of edges e in C satisfying the following property: the end-points of e have connectivity at least k. We prove the following theorem. Note the importance of this theorem: if the min-cut in the graph is 2, then Theorem 6 tells us that the number of cuts with size Δ is $n^{2\Delta/2}$; however, the number of k-projections of these cuts is much smaller, just $n^{O(\Delta/k)}$.

Theorem 3. *Given an integer* $\Delta \geq 1$*, the number of distinct k-projections over all cuts C in G of size* Δ *is* $n^{O(\Delta/k)}$*.*

2 Background

Recall the following from previous lectures.

Theorem 4. *Given any two cuts $(A, V-A)$ and $(B, V-B)$, let $\delta(A)$ denote the number of edges between A and $V-A$, and likewise $\delta(B)$ is the number of edges between B and $V-B$. Let $\delta(X,Y)$ denote the number of edges between vertex subsets X and Y. The following facts hold:*

- $\delta(A) + \delta(B) \geq \delta(A \cap B) + \delta(A \cup B) + 2\delta(A-B, B-A)$.
- $\delta(A) + \delta(B) \geq \delta(A-B) + \delta(B-A) + 2\delta(A \cap B, \overline{A \cup B})$.

Theorem 5. *[NI92] Given an undirected, unweighted graph G, a collection of edge disjoint forests can be constructed in $O(n+m)$ time so as the satisfy the following property: for any integer $i > 0$, consider the graph G_i obtained by unioning the first i forests; then the connectivity $C_{G_i}(x,y)$ between any two vertices x, y in G_i is at least $min\{i, C_G(x,y)\}$, where $C_G(x,y)$ is the corresponding connectivity in G.*

Theorem 6. *[Kar94] Given an undirected, unweighted graph G with min-cut size k, the number of cuts of size $h \geq k$ in G is $n^{2h/k}$.*

Theorem 7. *Suppose $X_1, X_2, \ldots, X_n$ is a set of independent random variables such that each X_i, $i \in \{1, 2, \ldots, n\}$, has value $1/p_i$ with probability p_i for some fixed $0 < p_i \leq 1$ and has value 0 with probability $1 - p_i$. For any $p \leq \min_i p_i$ and for any $\delta > 0$,*

$$Pr[\sum_i X_i > (1+\delta)n] < \begin{cases} e^{-0.3\delta^2 pn} & \text{if } 0 < \delta < 1 \\ e^{-0.3\delta pn} & \text{if } \delta \geq 1. \end{cases}$$

Theorem 8. *Suppose $X_1, X_2, \ldots, X_n$ is a set of independent random variables such that each X_i, $i \in \{1, 2, \ldots, n\}$, has value $1/p_i$ with probability p_i for some fixed $0 < p_i \leq 1$ and has value 0 with probability $1 - p_i$. For any $p \leq \min_i p_i$ and for any $\delta > 0$,*

$$Pr[\sum_i X_i < (1-\delta)n] \begin{cases} < e^{-0.5\epsilon^2 pn} & \text{if } 0 < \delta < 1 \\ = 0 & \text{if } \delta \geq 1. \end{cases}$$

3 Edge Splitting

We prove Theorem 2 here. Note that we are given an edge $e = uv$ and we are trying to find an appropriate mate vw for e. Replacing uv and vw by the edge uw will cause a reduction in the size of exactly those cuts $(C, V-C)$ for which $u, w \in C$, $v \in V-C$. These cuts will reduce in size by 2. A problem would arise if C had the following property: for some vertex pair x, y, $x, y \neq v$, $x \in C, y \in V-C$ and $\delta(C) = C_G(x,y)$ (note $\delta(C) = C_G(x,y) + 1$ could be a problem as well, but the *even degree* constraint ensures that all cuts have even size, so this case is ruled out); such critical cuts will reduce in size by 2 causing the connectivity between the associated vertices x, y to fall. Avoidance of this situation places the following constraint on the choice of vw.

Necessary Condition for vw. Consider any cut $(C, V-C)$ such that $u \in C$, $v \in V-C$, and, for some vertex pair x, y, $x, y \neq v$, $x \in C, y \in V-C$ and $\delta(C) = C_G(x,y)$; then w should be chosen to be outside C. Call such cuts C *critical*.

The big question now is: does v have a neighbor w that is outside *ALL* critical cuts. As a first observation, note the following lemma.

Lemma 9. *If $(C, V-C)$ is a critical cut then v has a neighbor outside C.*

Proof. For a contradiction, suppose all of v's neighbors are in C. Then consider $(C \cup \{v\}, V - C - \{v\})$. Since all of v's neighbors are in C and since $u \in C$, $\delta(C \cup \{v\}) < \delta(C)$. Since C is critical, for some x, y, $x, y \neq v$, $x \in C, y \in V-C$, $\delta(C) = C_G(x,y)$. It follows that $\delta(C \cup \{v\}) < C_G(x,y)$. Since $x \in C \cup \{v\}$ and $y \notin C \cup \{v\}$, $\delta(C \cup \{v\}) < C_G(x,y)$ is a contradiction. □

We're not done yet though. If there was only one critical cut then the above lemma gives us the w that we need: pick that neighbor of v that is outside C. What if there are many critical cuts? A neighbor of v outside one critical cut will satisfy that critical cut but might violate some other critical cut. We need our second lemma to bail us out here. It is immediate from this lemma (coupled with Lemma 9) that we can find a neighbor w for v that is outside all critical cuts.

Lemma 10. *If $(A, V-A)$ and $(B, V-B)$ are both critical cuts $(A \cup B, V - A - B)$ is also a critical cut.*

Proof. Let $(A, V-A)$ be critical for vertex pair x, y and $(B, V-B)$ for vertex pair x', y'. Note $u \in A \cap B$ and v in $V - A - B = \overline{A \cup B}$, so $\delta(A \cap B, \overline{A \cup B}) \geq 1$. There are three cases next (some inspection will show that these are the only cases).

Case 1: One of x, y is in $A - B$ and one of x', y' is in $B - A$. Then applying Theorem 4, we get a contradiction as follows:

$$
\begin{aligned}
C_G(x,y) + C_G(x',y') &= \delta(A) + \delta(B) \\
&\geq \delta(A-B) + \delta(B-A) + \delta(A \cap B, \overline{A \cup B}) \\
&\geq C_G(x,y) + C_G(x',y') + 1
\end{aligned}
$$

Case 2: One of x, y is in $A \cap B$ and one of x', y' is in $\overline{A \cup B}$. Then applying Theorem 4:

$$
\delta(A) + \delta(B) = C_G(x,y) + C_G(x',y') \geq \delta(A \cap B) + \delta(A \cup B) \geq C_G(x,y) + C_G(x',y')
$$

we conclude that $\delta(A \cup B) = C_G(x',y')$; since u, v and x', y' are both separated by cut $(A \cup B, V - A - B)$, this cut must be a critical cut, as required.

Case 3: $x \in A \cap B, y \in V - A - B, x' \in A - B, y' \in B - A$. Since both $(A, V - A)$ and $(B, V - B)$ split both vertex pairs x, y and $x'y'$, it must then be the case that $C_G(x, y) = C_G(x', y')$. Then applying Theorem 4:

$$\delta(A) + \delta(B) = C_G(x,y) + C_G(x',y') = 2C_G(x,y) \geq \delta(A \cap B) + \delta(A \cup B) \geq 2C_G(x,y)$$

we conclude that $\delta(A \cup B) = C_G(x, y)$; since u, v and x, y are both separated by cut $(A \cup B, V - A - B)$, this cut must be a critical cut, as required. □

4 Counting Cut Projections

We will prove Theorem 3 next. The approach is similar to the proof of Theorem 6, which uses Karger's edge compression technique [Kar94]. We pick edges uniformly at random and compress. We then lower bound the probability that a specific cut of our interest is untouched by this compression. And this gives us a upper bound on the number of cuts of interest. We mimic this approach towards counting k-projections.

Choose a particular subset of edges S that appears as a k-projection of some cut C of size Δ. We now run the random edge compression process until there are just two vertices in each connected component (note, we no longer assume that the graph is connected) or the number of vertices falls below $2\Delta/k$. Then we ask the following question: what is the probability that the cut C is preserved, i.e, two vertices on opposite sides of C are never merged in this process. Everytime we compress an edge, if there are no vertices with degree smaller than k, then we are guaranteed that there are at least $nk/2$ edges in the graph and therefore the probability of an edge from C getting compressed is at most $\frac{\Delta}{nk/2}$, which will yield the desired bound of $n^{O(\Delta/k)}$. The problem of course is that there could be vertices with degree smaller than k.

What comes to our rescue here is the fact that an edges incident on a vertex with degree $< k$ cannot be in S, since S comprises only edges whose endpoints have edge connectivity at least k. Therefore, since we are only interested in counting the number of k-projections, we can afford to *ignore* edges incident on vertices with degree $< k$. How exactly do we ignore these edges? If we just delete them, then connectivities in the graph could drop and we will no longer be able to claim that a vertex with degree $< k$ created as a result of compressions has no incident edges which belong to k-projections. Instead, we invoke Theorem 2 to split off all vertices with degrees $< k$. Splitting off ensures that connectivities of the remaining vertices are preserved, so any edge incident on a vertex of degree k at any point in this procedure is guaranteed to not be in the k-projection of any cut. Note that applying Theorem 2 does need even degrees, but that we can get by simply doubling every edge, and now counting $2k$-projections in cuts of size 2Δ instead (also note that both the splitting-off operation and the edge compression operation ensure that degrees once even stay even). For simplicity, we just assume that Theorem 2 is applicable as is.

So here is our randomized procedure. If we have a vertex of degree $< k$, we split it off. If not, we choose an edge uniformly at random and compress it. The splitting-off step is guaranteed to not affect edges which are present in any k-projection. The compression step will compress an edge crossing C with probability at most $\frac{\Delta}{n'k/2}$, where n' is the current number of vertices. We continue the splitting-off/compression steps until each connected component has two vertices each with degree $\geq k$, or the number of vertices drops below $2\Delta/k$. The probability that two vertices from opposite sides of C are never compressed in this process is at least

$$
\begin{aligned}
&(1-\frac{\Delta}{N_0 k/2})(1-\frac{\Delta}{N_1 k/2})\dots(1-\frac{\Delta}{N_r k/2}) \\
&\geq (1-\frac{\Delta}{nk/2})(1-\frac{\Delta}{(n-1)k/2})\dots(1-\frac{\Delta}{(2\Delta/k+1)*k/2}) \\
&\geq \frac{1}{n^{2\Delta/k}}
\end{aligned}
$$

where $N_j \leq n-j$ are random variables depending upon how many vertices were split-off.

In summary, for any cut C of size Δ, the probability that all edges in the k-projection of C are preserved at the end of the above procedure (i.e., not compressed, not destroyed by edge-splitting) is at least $\frac{1}{n^{2\Delta/k}}$. Can we then conclude that the number of distinct k-projections over all cuts of size Δ is at most $n^{2\Delta/k}$? Not yet, because we could have multiple connected components at the end and therefore multiple distinct k-projections could survive at the end. The number of distinct k-projections of cuts of size Δ that survive at the end is at most $\binom{n}{\Delta/k} \leq n^{\Delta/k}$ (because each component has only 2 vertices, each of degree at least k). It follows that the number of distinct k-projections of cuts of size Δ is at most $n^{3\Delta/k}$. A more careful analysis will give you $n^{2\Delta/k}$.

5 Proof of Theorem 1

Given Lemmas 2 and 3, we can now provide a short proof of Theorem 1. First, note that since edges are sampled with varied probabilities, it is challenging to employ a Chernoff bound to show tight concentration around mean for collections of edges. To make this feasible, we parition edges into categories with roughly similar sampling probabilities; in particular, category k comprises all edges e whose NI index NI_e is in the range $k \dots 2k-1$.

Next, consider all cuts C with $\Delta_k > 0$ category k edges and $\Delta_{k/2}$ category $k/2$ edges. First, we show in Lemma 11 that $\Delta_k + \Delta_{k/2} \geq k/2$ and that the number of distinct subsets S of category k edges in these cuts is $n^{O(\frac{\Delta_k+\Delta_{k/2}}{k})}$. Then, in Lemma 12, we show that each of these subsets S will have cumulative weight post-sampling in the range $\Delta_k \pm \frac{\epsilon}{2}(\Delta_k + \Delta_{k/2})$, with failure probability probability $n^{-\Theta(1)(\frac{\Delta_k+\Delta_{k/2}}{k})}$. By adjusting constants appropriately, we can conclude that ALL such subsets S have cumulative weight post-sampling in the range $\Delta_k \pm \frac{\epsilon}{2}(\Delta_k + \Delta_{k/2})$, with failure probability $n^{-\Theta(1)(\frac{\Delta_k+\Delta_{k/2}}{k})}$. Integrating this failure probability over all values of $\Delta_k, \Delta_{k/2}$ satisfying $\Delta_k + \Delta_{k/2} \geq k/2$, and then over all categories $k \geq 1$, we ensure that this combined failure probability is $\leq 1/n$. It follows that the cumulative weight post-sampling of every cut C is in the range $|C| \pm 2\frac{\epsilon}{2}|C| = |C| \pm |C|\epsilon$, as required! And we are done! Of course, it remains to show Lemma 11 and Lemma 12; short proofs follow.

Lemma 11. *Consider all cuts that have $\Delta_k > 0$ category k edges and $\Delta_{k/2}$ category $k/2$ edges. For each such cut C, consider the subset C^k of category k edges in this cut. The number of such distinct subsets C^k of category k edges over all the above cuts is $n^{O(\frac{\Delta_k+\Delta_{k/2}}{k})}$, where $\Delta_k + \Delta_{k/2} \geq k/2$.*

Proof. Recall that a category k edge has NI index in the range $k \ldots 2k$. Consider the subgraph G' of G obtained by restricting attention to edges with NI indices in the range $k/2 \ldots 2k-1$. For every cut C in G being considered, its size in G' exactly equals $\Delta_k + \Delta_{k/2}$. A key property of NI trees is that the endpoints of each category k edge in G are at least $k/2$ connected in G' (because these endpoints are in the same connected component in the NI trees $k/2 \ldots k-1$). Since $\Delta_k > 0$, it follows that $\Delta_k + \Delta_{k/2} \geq k/2$. By Lemma 3, the number of distinct subsets C^k over all cuts C being considered is at most $n^{O(\frac{\Delta_k+\Delta_{k/2}}{k})}$. □

Lemma 12. *Consider all cuts with $\Delta_k > 0$ category k edges and $\Delta_{k/2}$ category $k/2$ edges. For each such cut C, consider the subset C^k of category k edges in this cut. The cumulative weight post-sampling of this subset C^k is in the range $\Delta_k \pm \frac{\epsilon}{2}(\Delta_k + \Delta_{k/2})$, with failure probability $n^{-\Theta(1)(\frac{\Delta_k+\Delta_{k/2}}{k})}$.*

Proof. Simply use the Chernoff Bounds in Theorem 7 and 8, with $p = \frac{O(1)\log n}{\epsilon^2 2k}$, $n = \Delta_k$, and $\delta = \frac{\epsilon}{2}\frac{\Delta_k+\Delta_{k/2}}{\Delta_k}$. □

6 Survey

Benczur and Karger [BK96] were the first to show a sparsification result; they used sampling probabilities inversely proportional to a measure they introduced called *strong connectivity* to obtain an $O(n \log n)$ sized sparse graph. Spielman and Srivastava [SS11] used linear algebraic techniques to obtain an $O(n \log n)$ sized sparse graph using sampling probabilities proportional to electrical resistance. Hariharan and Panigrahi, and independently Fung and Harvey [FHHP11], showed that sampling proportional to standard connectivities works as well but yields an $O(n \log^2 n)$ sized sparse graph (it is open whether the extra log can be removed).

References

[BK96] A. A. Benczur and D. R. Karger. Approximate s-t min-cuts in $\tilde{O}(n^2)$ time. *Proc. 28th Annual ACM Symposium on Theory of Computing*, 1996.

[Gab95] H.N. Gabow. A matroid approach to finding edge connectivity and packing arborescences. *J. Comput. Syst. Sci.*, 50(2):259–273, 1995.

[HP10] R. Hariharan and D. Panigrahi. A General Framework for Graph Sparsification. http://arxiv.org/abs/1004.4080, 2010

[FHHP11] W.S. Fung, R. Hariharan, N. J. A. Harvey and D. Panigrahi. A general framework for graph sparsification. *STOC*, 2011.

[Kar94] D. Karger. Random sampling in cuts, flows and network design problems. *STOC*, 1994.

[Ma93] D. W. Matula. A Linear Time 2+epsilon Approximation Algorithm for Edge Connectivity. *SODA*, 1993.

[NI92] H. Nagamochi and T. Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(5&6):583–596, 1992.

[SS11] D.A. Spielman, N. Srivastava. Graph Sparsification by Effective Resistances. *SIAM J. Comput.* 40(6): 1913-1926 (2011). 2010