College of Computer & Information Science | Spring 2016
Northeastern University | 19 February 2016
CS 7880: Network Algorithms | Scribe: Matthew Dippel

- Small world properties
- Graph sparsifiers

1 Small world properties

Small world graphs are graphs with a small number of edges (linear in the number of nodes), small max degree (constant), and logarithmic diameter (the minimum achievable with the previous two constraints).

Consider the following problem: We have a grid graph G on n^2 vertices that we'd like to augment with additional edges. For each node, we want to add a constant number of "long range edges" such that the following holds for the new graph G':

- The diameter of G' is $O(\log n)$.
- The simplistic navigation algorithm of going to your neighbor that is closest in the grid topology to your destination results in paths that are $O(\log n)$.

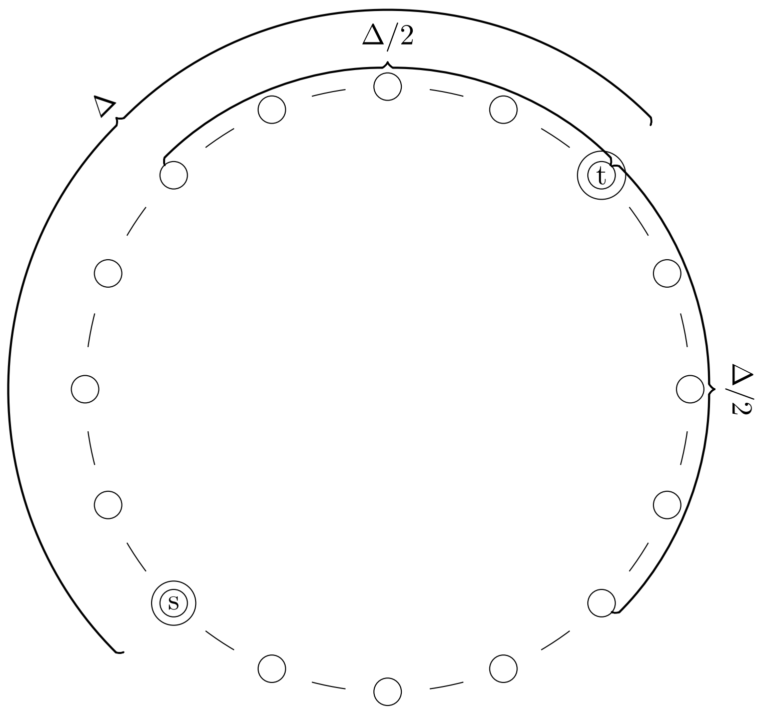
Consider the case where we take a grid graph, and each node gets 2 edges to other nodes selected uniformly at random. This can be shown to reduce the diameter to $O(\log n)$, but the resulting graph does not yield short paths under the greedy routing algorithm.

Kleinberg [3] proposed an alternate distribution for adding edges. Let $d_{grid}(u, v)$ be the distance between nodes u and v in the grid topology. He proposed that new edges be added according to the following distribution:

$$\Pr[u \to v] \propto \frac{1}{[d_{grid}(u, v)]^k}$$

1.1 Analysis on ring topology

We will not analyze this on the grid case. Rather, we will analyze a simpler case, which is the ring topology when $k = 1$. For the remaining section, $d(u, v)$ will refer to the distance in the ring topology.



Suppose we have two nodes s, t, and we want to know the expected distance between s and t. Suppose the distance in the ring topology is Δ. We will show that in expected $O(\log n)$ steps, we can get to distance $\Delta/2$ from t. Repeating this for $\Delta/4, \Delta/8, ...$ shows that we can get to t in expected $O(\log^2 n)$ time.

Let I be the set of all nodes that are distance $\Delta/2$ from t. The probability that s has a long range link to some node in I is $\frac{1}{c}\sum_{v\in I}\frac{1}{d(u,v)}$. Here c is the normalization constant $c = \sum_{v\in V}\frac{1}{d(u,v)}$

We can show that $c \geq \log(n/2)$. Since $d(s, v) \leq 3\Delta/2$, the probability above is at least $\frac{1}{c}\frac{2}{3\Delta}|I| \geq \frac{1}{3\log N}$

Hence the expected number of steps we take before we can get within the interval I around t is $3 \log N$. Thus the expected number of total steps to get to t is $O(\log^2 N)$.

2 Graph sparsifiers

A graph sparsifier for a graph G is a subgraph G' which is close to G with respect to a specific property. We will examine graph sparsifiers for cuts.

In an unweighted graph G, for any subset of vertices S, let $G(S, \overline{S})$ denote the number of edges across the cut determined by S. We would like to find a sparse subgraph H, where we have added weights across the edges, such that the weighted cut $H(S, \overline{S})$ is close to the unweighted cut $G(S, \overline{S})$.

2.1 Cut sparsifiers where Δ^* is large

Consider the following simple cut sparsifier, motivated by [1] which works by random sampling:

- For a fixed probability p, pick each edge e with probability $p_e = p$.
- Assign each selected edge a weight of $1/p$.

Theorem 1. *Generate H such that for each edge $e \in E$, e is in H with probability $p = \frac{c \log n}{\Delta^* \varepsilon^2}$. The expected size of the sparsifier is $\Omega(\frac{m \log n}{\Delta^*})$, where m is the number of edges, n is the number of vertices, and $\Delta^* >> \log n$ is the size of the min cut in G. Further, with probability $1 - \frac{1}{n^\alpha}$ for some constant α, for all sets S, $w(H(S, \overline{S})) \in (1 \pm \epsilon) G(S, \overline{S})$.*

Proof. For any cut of size Δ in G, we have $\mathbb{E}[|H(S, \overline{S})|] = p\Delta$, so then the expected weight of the cut is Δ. Since each edge being selected is an independent event, the probability that the number of edges in the cut signifiantly deviates from the expectation can be bounded:

$$\Pr[|H(S, \overline{S})| \notin (1 \pm \varepsilon)p\Delta] \leq 2\text{exp}(-\varepsilon^2 p\Delta/3)$$

If we use $p = \frac{c \log n}{\Delta^* \varepsilon^2}$, where Δ^* is the size of the min cut in the graph. This yields the above bound as:

$$\begin{aligned}\Pr[|H(S,\overline{S})| \notin (1 \pm \varepsilon)p\Delta] &\leq 2\exp(-\varepsilon^2 c \log n \Delta/(3\varepsilon^2\Delta^*) \\ &\leq 2n^{-c/3}\end{aligned}$$

However, we cannot trivially apply a union bound to this, as there are 2^{n-1} possible cuts. Instead, will make use of the following lemma:

Lemma 1. *In any graph G of n vertices, $|$the number of cuts of size $\leq \alpha \cdot$ min-cut$| \leq n^{2\alpha}$*

Proof. We will show the case where $\alpha = 1$, the argument generalizes to larger α.

Consider the following randomized algorithm, which generates a cut:

- Randomly select an edge
- Contract the edge so the two nodes joined by it become one node
- Remove all self loops
- Repeat until two nodes are left, return the parallel edges between the two vertices as the edges across the cut

We wish to calculate the probability that this achieves some particiular min cut of size Δ^*. The probability that an edge of the min cut is selected in the first round is $\Delta^*/|E|$, so the probability that the min cut "survives" the first round is $\left(1 - \frac{\Delta^*}{|E|}\right)$.

However, since the average degree is at least $2|E|/n$, and the average degree is an upper bound on the min cut (since we could always take the cut to only isolate that vertex), we have $\Delta^* \leq 2|E|/n$, or $|E| \geq \Delta^* n/2$, hence $\Delta^*/|E| \leq 2/n$, and the probability the cut survives the first round is $\geq (1 - 2/n)$.

After we contract an edge (u, v), any unaffected cut has exactly the same size in the contracted graph, so we can apply the same bound regarding the min degree.

This argument generalizes to show that the probability the min cut survives round i is at least $\left(1 - \frac{2}{n-i+1}\right)$, so the probability it survives all $n-2$ rounds is at least:

$$\left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\cdots\left(1-\frac{2}{3}\right)$$

$$=\frac{n-2}{n}\frac{n-3}{n-2}\cdots\frac{1}{3}$$

$$=\frac{2}{n(n-1)}$$

We have that the event a specific cut survives all the rounds is mutually exclusive from all other cut survival events. Thus we cannot have more than $\binom{n}{2}$ distinct min cuts, or the sum of the probabilities of picking each of them would exceed 1, which isn't possible.

This argument extends to arbitrary α by replacing Δ^* with $\alpha\Delta^*$ $\square$

Recall that we simplified our above analysis by saying $\Delta/\Delta^* \geq 1$. However, given a particular value of Δ, we now have a bound on how many cuts can achieve that value. We will split all possible cuts into different ranges, bound the sizes of the ranges, and apply the unsimplified bound across the union of the events.

Consider the set of all cuts C_i satisfying $2^{i-1}\Delta^* \leq \Delta < 2^i\Delta^*$. We have from our lemma that $|C_i| \leq n^{2^{i+1}}$. Such a cut is not preserved by our sampling with probability

$$\leq 2\exp(-\varepsilon^2 c\log n\Delta/(3\varepsilon^2\Delta^*) \leq 2n^{-2^i c/3}$$

.

Applying a union bound over all cuts in C_i yields that the probability at least one is bad is at most

$$2n^{-2^i c/3+2^{i+1}} = 2n^{-2^i(c/3-2)}$$

There are at most $\log n$ such groups, and each probability is decreasing, so we can bound the probability that any one of the groups has a single deviation as:

$$\leq 2 \log n n^{-(c/3-2)}$$

as long as c is chosen to be sufficiently large, this is an inverse polynomial in n, which goes to 0 for large n.

All that remains to analyze is the number of edges selected, which in expectation is $cm \log n/\Delta^*$. If Δ^* is sufficiently large (by definition it must be $> c \log n$) then we are removing a large fraction of the edges. However if Δ^* is small then this is not a good sparsifier. $\square$

2.2 Cut sparsifier with varying p_e

We will see that we can improve it by varying the probabilities p_e in terms of the edge e, first done in [2]. Define λ_e as the size of the smallest cut containing edge e. Then we will select an edge with probability $p_e \propto 1/\lambda_e$. Specifically, we will use $p_e = \min\{1, \frac{d \ln n}{\epsilon^2 \lambda_e}\}$, where d is a constant to be set during the analysis, and ϵ is the approximation factor we wish to achieve.

Theorem 2. *The expected size of the sparsifier with the above probabilities is* $O\left(\frac{n \ln^2 n}{\epsilon^2}\right)$. *Further, with high probability, for all sets* S, $w(H(S, \overline{S})) \in (1 \pm \epsilon) G(S, \overline{S})$.

We will not complete the proof here, but we will show a key lemma to be used in the next lecture:

Lemma 2. $\sum_{e \in E} \frac{1}{\lambda_e} \leq n$

Proof. We wish to show the sum is no more than $|V| - 1$. Induct on the number of vertices. When $n \leq 2$, it's true, since $n = 1$ yields the sum to be 0, and $n = 2$ yields the sum to be at most 1.

Suppose we have a graph of n vertices, take the min cut as described previously. Then the sum of the λ_e of these edges is exactly one. Suppose this cut partitions G into sets A and B. Consider any edge e' that is not across this cut. Let $\lambda_{e'}^A$ be the size of the smallest cut containing e' when restricted to the sub graph induced on A. We have that $\lambda_{e'}^A \leq \lambda_{e'}$, because the size of any cut within A can only be increased by adding nodes and edges to the graph. Inductively, $\sum_{e' \in A} \frac{1}{\lambda_{e'}^A} \leq |A| - 1$, and so $\sum_{e' \in A} \frac{1}{\lambda_{e'}} \leq |A| - 1$. We can also apply this to B. Combining this with the original min cut yields:

$$\sum_{e \in E} \frac{1}{\lambda_e} \leq 1 + (|A| - 1) + (|B| - 1) = |V| + 1 - 2 = |V| - 1$$

as desired. A tight case is when G is the line graph, where there are $n - 1$ edges, and each edge has $\lambda_e = 1$. $\square$

References

[1] András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *CoRR*, cs.DS/0207078, 2002.

[2] Wai Shing Fung, Ramesh Hariharan, Nicholas J.A. Harvey, and Debmalya Panigrahi. A general framework for graph sparsification. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 71–80, New York, NY, USA, 2011. ACM.

[3] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 163–170, New York, NY, USA, 2000. ACM.