CS 388R: Randomized Algorithms Fall 2017

Lecture 20 — Nov. 14, 2017

Prof. Eric Price *Scribe: Ethan Arnold, Yanyao Shen*

NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS

1 Overview

In the last lecture, we discussed spectral graph sparsification. In this lecture, we turn to the problem of computing edge connectivity in DAGs.

2 Edge Connectivity

In a directed graph, the s-t edge connectivity is the maximum number of edge-disjoint paths from s to t. This is equal to the s-t min-cut, as well as the s-t max flow. Thus, we can compute the s-t edge connectivity in a general graph in $O(m^2n)$ time using Ford-Fulkerson. We will consider the more specific problem of computing edge connectivity in a directed acyclic graph with maximum degree d. Cheung, Lau, and Leung give an $O(md^{1.38})$-time algorithm in [1]. (This is the same as the time to multiply an $m \times d$ matrix by a $d \times m$ matrix.)

For this problem, we turn to network coding.

3 Network Coding

Suppose node s (source) wants to communicate a long message to node t (target) in a communication network, where no nodes are aware of the global graph structure. For now, we consider any directed graph (i.e., the network may have cycles). Each directed edge may send one "packet" of information at each time step, and all the edges send information at the same time during a time step. We will assume that the packets are relatively large and may include some sequence number if we care about the order of packets; henceforth we only care that after some time t knows all the packets that s wanted to communicate, and not the order in which they arrived.

Concretely, s has k messages that it wants to send. Each of these messages is an element of $\mathbb{F}_q^\ell$, where q and ℓ are reasonably large. (Recall that $\mathbb{F}_q$ is the finite field with q elements, where q is prime.)

Let packet size $\ell^{'} = \ell + k$, $m_i^{'} = (e_i | m_i)$, where e_i is the unit vector in $\mathbb{F}_q^\ell$. Now, $m_1^{'}, m_2^{'}, \cdots, m_k^{'}$ are linearly independent. This guarantees that we can recover the original vectors exactly through

Gram-Shmidt decomposition. Now, let

$$z = \left(\begin{bmatrix} 1 & 0 & 0 & m_1 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & m_k \end{bmatrix} \right)$$

and the source s knows $span(z) \subset \mathbb{F}_q^{\ell+k}$. At each time step, any vertex v knows subspace $X_v \subset \mathbb{F}_q^{\ell'}$ ($X_v = \{0\}, \forall v \neq s$ to start with). At every time step, on every edge (u, v), u picks a random vector $w \in X_u$ and sends it to v. Any v that receives $(w_1, \cdots, w_d)$ from neighbors, sets $X_v \leftarrow span(X_v, w_1, \cdots, w_d)$.

Definition 1. *Let $Y_v = X_v^{\perp} = \{w \in \mathbb{F}_q^{\ell'} | \langle w, w' \rangle = 0, \forall w' \in X_v\}$. Say v is "aware" of w if $w \notin Y_v$.*

The goal of the algorithm is that, at the end, t is aware of everything s is aware of.

Claim 2. *If u is aware of w, v is not aware of w at a given time step, and u sends message to v. Then,*

$$\mathbb{P}[v \textit{ is unaware of } w \textit{ in the next round}] \leq \frac{1}{q}.$$

This result is straightforward since in order for v to be unaware of w in the next round, the coefficient for w in the random vector should be 0. For field $\mathbb{F}_q^{\ell'}$, this probability is $1/q$.

On any length L path, any w that s is aware of,

$$\begin{aligned} &\mathbb{P}[t \text{ unaware of } w \text{ after } R \text{ rounds}] \\ \leq &\mathbb{P}[\text{awareness fails to move forward} \geq r \text{ times}] \\ \leq &\binom{L+r}{r} \frac{1}{q^r} \\ \leq &\left(\frac{e(1 + \frac{L}{r})}{q} \right)^r \end{aligned}$$

where $R = L + r$. Therefore, after R rounds, if $|C_{s,t}| =$ minimum $s \to t$ cut size,

$$\mathbb{P}[\text{chance that } t \text{ is unaware of } w] \leq \left(\frac{e(1 + \frac{n}{r})}{q} \right)^{|C_{s,t}| r} \lesssim q^{-|C_{s,t}| R(1-\epsilon)}$$

if $q \geq 2^{1/\epsilon}, R \geq n/\epsilon$.

If t is aware of all $w \in X_s, w \neq 0$, then X_t must be X_s. Since $|X_t| = q^k$, $X_t = X_s$ w.h.p. after

$$q^k \cdot q^{-R|C_{s,t}|(1-O(\epsilon))} = o(1).$$

Since q is large, we only need the LHS in the above equation to be less than $\frac{1}{q}$, which gives the sufficient condition:

$$k < R|C_{s,t}|(1 - O(\epsilon)) - 1$$

On the other hand, we know that the capacity of the network requires the possible number of messages to be sent satisfies

$$k \leq R \cdot \frac{\ell'}{l} |C_{s,t}| = R \cdot |C_{s,t}|(1 + \epsilon) \triangleq \text{OPT}$$

This is saying that the randomized network coding algorithm succeeds as long as $k \leq (1 - O(\epsilon)) \cdot$ OPT.

4 Applying network coding to edge connectivity

First, we will aim for $O(md^2)$ time to find the edge connectivity in a DAG. Later, we will improve this to $O(md^{1.38})$. We will consider the nodes in topological order (this can be computed in $O(m)$ time), and we will simulate a single round of "knowledge propagation" as described in the previous section.

After simulating one round of communication (every edge sends exactly one message), we output $\dim X_t$, and we hope that this is equal to $|C_{s,t}|$ (as this is what we are trying to compute). Clearly, $\dim X_t \leq |C_{s,t}|$, because $|C_{s,t}|$ is a bound on the amount of communication that could have been transmitted from s to t. We would like to show that the probability that $\dim X_t < |C_{s,t}|$ is not too large.

To do this, we first consider the probability that the awareness of a particular vector w is transmitted from s to t along a particular path. This is at least

$$(1 - \frac{1}{q})^n \geq 1 - \frac{n}{q}.$$

Thus, the probability that awareness of w is transmitted along any path is at least

$$1 - (\frac{n}{q})^{|C_{s,t}|} = 1 - q^{-|C_{s,t}|(1-\log_q n)}.$$

Note that the number of vectors w that t is unaware of at the end is equal to

$$q^{|X_t^{\perp}|} = q^{d-\dim X_t},$$

and the expected value of this quantity is

$$q^{d-|C_{s,t}|(1-\log_q n)}.$$

Thus, by Markov's inequality, we have that the number of vectors w that t is unaware of is at most

$$q^{d-|C_{s,t}|(1-\log_q n)+1/2},$$

with high probability, and so with high probability it is also the case that

$$\dim X_t \geq |C_{s,t}|(1-\log_q n) - 1/2.$$

To learn $|C_{s,t}|$ exactly, we may set $q > n^{2d}$, or for an approximation of $|C_{s,t}|$ we may use a smaller value of q.

We have established that with high probability we obtain the correct edge connectivity value, so all that is left is to consider the time complexity. We transmit a vector m times, and each vector is a random linear combination of known vectors. We can generate the coefficients and compute the vector to transmit in $O(d^2)$ time. Upon receiving a vector from a neighbor, we simply orthogonalize it with respect to the already-known vectors of the current node, which takes $O(d^2)$ time.

Since we repeat this process for every edge, we have a total of $O(md^2)$ time to compute edge connectivity with high probability.

To achieve $O(md^{1.38})$ time, we can reduce all of these operations to matrix operations. We can batch the operations at every node, so that we orthogonalize all the incoming messages at the same time.

References

[1] Cheng, Lau, and Leung. *Graph Connectivities, Network Coding, and Expander Graphs.* 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)