CS761 Derandomization and Pseudorandomness **2022-23 Sem I**

Lecture 10: 08/09/22

Scribe: Raj Aryan Agrawal *Lecturer: Rohit Gurjar*

As we saw previously, in an expander graph, a random walk converges to uniform distribution in $O(\log n)$ steps. In particular, this means that starting from any vertex, any other vertex can be reached in $O(\log n)$ steps. Then we can check (s, t) connectivity by iterating over all $O(\log n)$ length paths from s, which requires $O(d \log n)$ space, where d is the degree. Assuming the degree of the expander graph is constant, this is a logspace algorithm.

To solve connectivity in logspace for a general graph, the idea is to convert any given graph into an expander via some operations that preserve connectivity. We will use 2 graph operations.

Expander Graph Operations

Squaring a Graph

Definition 10.1 (Square of Graph)**.** *Given a graph $G(n, d, \lambda)$, the square of the graph G^2 is the graph formed by all 2 length paths in G, that is $V(G^2) = V(G)$ and*

$$E(G^2) = \{(u, v) | \exists t \in V(G) \text{ such that } (u, t), (t, v) \in E(G)\}$$

By definition, number of vertices remains the same, n. For the degree of a vertex u, we have d neighbours of u and d neighbours from each of those neighbours, giving degree of any vertex u as d^2 (we allow multiple edges to exist between same pair).

Finally, for the parameter of expander graph, by considering all the 2 length paths only, the random walk matrix for G^2 is equivalent to M^2 where M is the random walk matrix matrix for G. Since λ is the second largest eigenvalue of M, λ^2 is the second largest eigenvalue of G^2, as long as G does not have eigenvalue -1. Thus the operation is

$$G(n, d, \lambda) \to G^2(n, d^2, \lambda^2)$$

This operation increases the degree significantly while also increasing the connectivity (since $\lambda < 1$, $\lambda^2 < \lambda$).

Zig Zag Product

In the previous operation we increased the connectivity but at the cost of increasing the degree. This second operation decreases the degree, while not hurting the connectivity too much.

Given 2 graphs $G(n, D, \lambda_1)$ and $H(D, d, \lambda_2)$, we will define their zig-zag product denoted as $G \textcircled{z} H$. We explain the construction of $G \textcircled{z} H$ in steps.

Vertices

Replace each vertex in G by all D vertices of H, that is, each vertex is duplicated D times with index (a, i) where a and i are labels for vertices in G and H respectively.

Edges

Firstly, for each vertex in G, order the edges connected to it as $1, 2, ..., D$ arbitrarily and similarly for each vertex in H, order the edges in $1, 2, \ldots, d$ arbitrarily. For edges in $G \textcircled{z} H$, we first take a random step in H, then a deterministic step in G (determined by our first random step) and finally a random step in H. The reason and intuition for this will be explained at the end of the construction. Thus, for each vertex (a, i), the neighbour is defined by the 2 random steps in graph H giving degree d^2. We can index a given edge from (a,i) as a tuple (j,k) where $j,k \in \{1,2,\ldots,d\}$. The other end point of this edge index can be determined as

- $i' \leftarrow j$th neighbour of i in H
- $b \leftarrow i'$th neighbour of a in G and
  $j' \leftarrow$ index of a in neighbour list of b
- $l \leftarrow k$th neighbour of j' in H

Then the other end point of edge with index (j,k) is (b,l).

One can imagine that for each vertex a in G we have a H-cloud in $G \textcircled{z} H$. And one step in $G \textcircled{z} H$ is equivalent to taking three steps: a H-step, a G-step, and a H-step. Intuitively, instead of taking a random step directly in G, we first take a random step in the H-cloud which tells us the edge to take in G. Then take the corresponding step in G. That is, in some sense we reduce the randomness from $\log D$ to $\log d$ in taking a random step. Also, we take a final step in the H-cloud so that the edge relation between (a,i) and (b,l) is symmetric.

Parameter λ

We now argue that the parameter λ' of $G \textcircled{z} H$ is not too much greater than λ_1, λ_2. For convenience, we will use the spectral gap defined before $\gamma_i = 1 - \lambda_i$

We first make a small claim about being able to split our matrix in 2 components

Claim 10.2. *Let A be the random walk matrix for a graph G with second eigenvalue λ. We can write A as*

$$A = \gamma J + \lambda E,$$

where J is the random walk matrix for a complete graph (with self loops) with same number of nodes. Then $\|E\| \leq 1$[1].

Proof. Since the vector $\frac{1}{n}\mathbf{1}$ is an eigenvector for both A and J, with eigenvalues 1, $\frac{1}{n}\mathbf{1}$ is also an eigenvector of E with eigenvalue 1. Now, we need to show that this is the largest eigenvalue. Consider a vector x, we write $x_\perp$ for component perpendicular to uniform vector $\frac{1}{n}\mathbf{1}$ and $x_{||}$ for component along the uniform vector $\frac{1}{n}\mathbf{1}$. Then

$$\begin{aligned}
\|Ex\| &= \|Ex_{||} + Ex_\perp\| \\
&\leq \|x_{||} + \frac{1}{\lambda}(Ax_\perp - \gamma J x_\perp)\| \\
&\leq \|x_{||} + \frac{1}{\lambda}\lambda x_\perp - 0\| && \because \text{ all other eigenvalues of complete graph are 0} \\
& && \text{for any vector orthogonal to } \mathbf{1}, A \text{ can multiply at most } \lambda \text{ in the norm} \\
&= \|x\|
\end{aligned}$$

So $\|Ex\| \leq \|x\|$ for any vector x. Thus the largest eigenvalue is at most 1.

Fun fact, by triangle inequality, we have

$$\|E\| \geq \frac{1}{\lambda}(\|A\| - \gamma\|J\|) = 1$$

giving us $\|E\| = 1$. ☐

Using this, we can prove the result

[1] $\|E\|$ stands for the spectral norm, i.e., $\max_{v \neq 0} \|Ev\|/\|v\|$. For symmetric matrices, spectral norm is equal to the largest eigenvalue.)

Claim 10.3. *We claim that*

$$\gamma(G \textcircled{z} H) \geq \gamma_1\gamma_2^2.$$

Which implies

$$\lambda' \leq \lambda_1 + 2\lambda_2$$

Proof. Let us try to construct the random walk matrix for for $G \textcircled{z} H$ by considering the three steps (H-step, G-step, H-step).

Suppose the random walk matrices for graphs G and H are A and B respectively. Then consider the matrix $\hat{B}$ which is for the H-step in the zig-zag product. For each of the n copies of a vertex of H, we need to do the same action, so we can say the matrix is

$$\hat{B} = I_n \otimes B$$

where $\otimes$ denotes the tensor product.

Now, consider the G-step in the zig-zag product. In the G-step, we go from (a, i') to (b, j'), where b is the i'th neighbor of a and a is the j'th neighbor of b. Observe that the G-step is deterministic and hence the corresponding matrix is a permutation matrix ($nd \times nd$). Let this matrix be $\hat{A}$.

Then, we can write the matrix for the zig-zag product as

$$M = \hat{B}\hat{A}\hat{B}.$$

Now using the Claim 10.2, we can write $B = \gamma_2 J + \lambda_2 E$, where $\|E\| \leq 1$. Taking the tensor product with I_n in this equation, we have

$$\hat{B} = I_n \otimes B = \gamma_2 I_n \otimes J + \lambda_2 I_n \otimes E = \gamma_2\hat{J} + \lambda_2\hat{E},$$

where we define $\hat{J}$ and $\hat{E}$ as $I_n \otimes J$ and $I_n \otimes E$. We get

$$M = \gamma_2^2\hat{J}\hat{A}\hat{J} + \lambda_2\gamma_2(\hat{J}\hat{A}\hat{E} + \hat{E}\hat{A}\hat{J}) + \lambda_2^2\hat{E}\hat{A}\hat{E}$$

The first term is simply the matrix for the zig zag product of G with a complete graph. Observe that the resultant graph from this is identical to G in behaviour since each H cloud is just a complete graph and every pair of vertices between 2 clouds have an edge. Thus a random walk in the product is same as a random walk in G, thus the second largest eigenvalue is λ_1. We will use this result below.

So, for the second largest eigenvalue of M, consider an x such that $\|x\| = 1$ and $x \cdot \mathbf{1} = 0$ which is an eigenvector for M, then for this,

$$Mx = \gamma_2^2\hat{J}\hat{A}\hat{J}x + (\lambda_2\gamma_2(\hat{J}\hat{A}\hat{E} + \hat{E}\hat{A}\hat{J}) + \lambda_2^2\hat{E}\hat{A}\hat{E})x$$

The norm of the first component $\|\hat{J}\hat{A}\hat{J}x\|$ is at most $\|\lambda_1 x\|$ by the fact that the second largest eigenvalue is λ_1. And for the remaining, the norm of each term is at most 1 (using the fact $\|AB\| \leq \|A\|\|B\|$), giving us

$$\|(\lambda_2\gamma_2(\hat{J}\hat{A}\hat{E} + \hat{E}\hat{A}\hat{J}) + \lambda_2^2\hat{E}\hat{A}\hat{E})x\| \leq \|(2\lambda_2\gamma_2 + \lambda_2^2)x\| = \|(1 - \gamma_2^2)x\|$$

Thus,

$$\|Mx\| \leq \gamma_2^2\lambda_1\|x\| + (1 - \gamma_2^2)\|x\|$$

So,

$$\lambda(M) \leq \gamma_2^2 - \gamma_2^2\gamma_1 + 1 - \gamma_2^2 = 1 - \gamma_2^2\gamma_1$$

Thus,

$$\gamma(M) \geq \gamma_2^2\gamma_1$$

Also,

$$\begin{aligned}
\lambda(M) &\leq 1 - (1 - \lambda_2)^2(1 - \lambda_1) \\
&= 1 - (1 - 2\lambda_2 + \lambda_2^2 - \lambda_1 + 2\lambda_2\lambda_1 - \lambda_2^2\lambda_1) \\
&= \lambda_1 + 2\lambda_2 - (2\lambda_2\lambda_1 - \lambda_2^2\gamma_1) \\
&\leq \lambda_1 + 2\lambda_2
\end{aligned}$$

□

Application of Expander Graph Operations

Construction of Large Expander Graphs

Start with a constant size expander graph $H(D^4, D, \frac{1}{8})$ for example. We can hope for a small λ since degree is quite large compared to number of nodes (explicit constructions exist for these, as given in assignment 1). Then we run the following steps

$$G_1 \leftarrow H^2$$
$$G_k \leftarrow G_{k-1}^2 \textcircled{z} H \ \forall k \geq 2$$

Then G_k is of the form $G_k(D^{4k}, D^2, \lambda)$

Claim 10.4. $\lambda \leq \frac{1}{2}\ \forall k \geq 1$

Proof. Base case for $k = 1$ is trivial to see as $\lambda = (1/8)^2 = 1/64$. Now for inductive case, we have

$$\lambda(G_k) \leq \lambda(G_{k-1}^2) + 2\lambda(H) \leq \left(\frac{1}{2}\right)^2 + 2\frac{1}{8} = \frac{1}{2}$$

□

Thus repeating the second step for as long as we want gives us increasing size expander graphs with the degree still being D^2.

Converting an arbitrary graph into an expander graph

Now, we describe how we can convert an arbitrary graph into an expander of polynomial size. This leads to the logspace algorithm for connectivity. Given an arbitrary graph, we can convert it to a regular graph of degree 3, which can then be squared an appropriate number of times to get a graph of form

$$G_0\left(n, D^2, 1 - \frac{1}{\mathrm{poly}(n)}\right),$$

for some large enough degree D^2. Recall that for any non-bipartite connected graph, the second eigenvalue is at least $1 - 1/\mathrm{poly}(n)$. Repeating same steps as above with the graph H ($G_k \leftarrow G_{k-1}^2 \textcircled{z} H$), we can construct G_k which is of form

$$G_k(nD^{4k}, D^2, \lambda).$$

Claim 10.5. *In $k = O(\log n)$ steps, we get $\lambda \leq \frac{1}{2}$.*

Proof. From the result of zig zag product we know that

$$\begin{aligned}\gamma_k &\geq (1 - \lambda_{k-1}^2)\left(\frac{7}{8}\right)^2 \\ &= (2\gamma_{k-1} - \gamma_{k-1}^2)\frac{49}{64}\end{aligned}$$

Assuming $\gamma_{k-1} \leq 1/2$, we have

$$\gamma_k \geq \frac{49}{64} \times \frac{3}{2}\gamma_{k-1} = \frac{147}{128}\gamma_{k-1}.$$

Thus, on each step γ_k increases by a constant multiple. Since we started with $\gamma_0 = 1/\mathrm{poly}(n)$, in $O(\log n)$ steps, we must reach $\gamma = 1/2$. □

Observe that the number of vertices at the end nD^{4k} remains polynomially bounded for $k = O(\log n)$.