Lecture 11: Constructing Expanders, Undirected Connectivity in log space

Topics in Complexity Theory and Pseudorandomness (Spring 2013)
Rutgers University
Swastik Kopparty
Scribes: Abdul Basit, Erick Chastain

1 Overview

In this lecture we will see an explicit construction of constant degree expanders. The main new tool we will see is the zig-zag product of graphs, which allows us to construct big expanders from smaller ones. Using these ideas, we will then see Reingold's beautiful deterministic log-space algorithm for the Undirected Connectivity Problem.

2 Constructing Expanders

Today we know many constructions of constant degree expander graphs. All of them are serious, and the analysis/construction always involves several interesting and nontrivial ideas.

One easily described expander family is the following.

Example 1. *For p a large prime, let $G = (V, E)$ with $V = \mathbb{F}_p$. For $x \in \mathbb{F}_p \setminus \{0\}$, the vertex x is connected to $(x+1), (x-1)$ and x^{-1}. The vertex $x = 0$ is connected to $1, (p-1)$ and 0.*

Then G_p is an 3-regular expander graph, such that $\forall S \subseteq V$ with $|S| \leq |V|/2$ we have that $|\Gamma(S)| \geq (1.001)|S|$.

This construction is due to Lubotzky-Phillips-Sarnak.

The constructions we will study in this class are based on the *zig-zag product*, which are more complicated to describe, but much simpler to analyze.

The strategy has a simple high level plan. **Plan:**

1. Start with a trivial graph (non-empty and connected).
2. Improve its expansion.
3. Repeat.

We will measure the quality of expansion so far in terms of absolute eigenvalue expansion.

We have already seen an operation that improves the expansion of a graph: powering. Given a graph G which is a d-regular λ-absolute eigenvalue expander, its tth power is a d^t-regular λ^t-absolute eigenvalue expander.

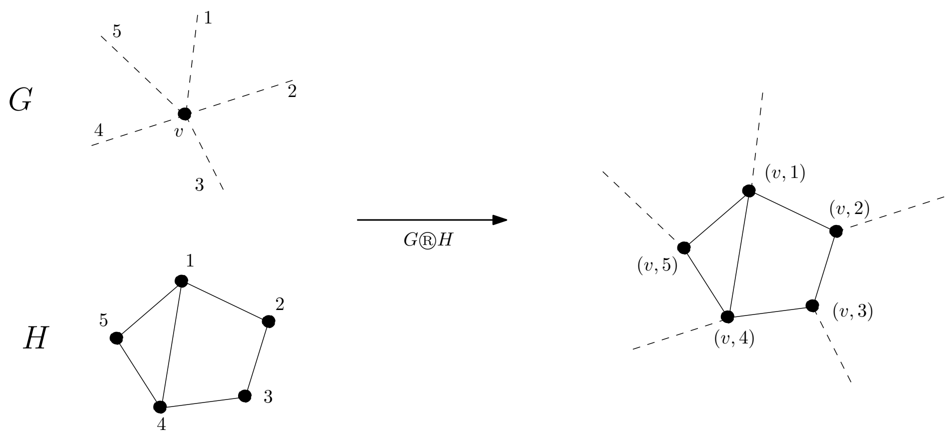

Figure 1: Replacement Product of a graph

In particular, powering decreases the quantity $\frac{\lambda}{d}$. The price that we pay is that powering increases the degree of the graph.

This is where the zig-zag product comes in: it is an operation that reduces the degree of a graph to a constant while not harming the expansion much.

2.1 Replacement product

We will first introduce the simpler *replacement product* of graphs.

To describe these products, we will use some notation. For a D-regular graph G, define the *Rotation Map* $\mathsf{Rot}_G : V \times [D] \to V \times [D]$ in the following manner:

$$\mathsf{Rot}_G(v, i) = (v', j)$$

where v' is the i^{th} neighbor or v and v is the j^{th} neighbor of v'. Also, let $v[i]$ be the i^{th} neighbor of vertex v in a graph.

Given

- G a "*large*" on D-regular graph on N vertices
- H a "*small*" on d-regular graph on D vertices

The *replacement product* $G \circledR H$ is a $(d+1)$-regular graph on ND vertices. $G \circledR H$ has vertex set $V(G) \times V(H)$ and with edge set defined by the following rotation map:

$$\mathsf{Rot}_{G \circledR H}((v, i), j) = \begin{cases} (\mathsf{Rot}_G(v, i), 0) & j = 0 \\ (v, \mathsf{Rot}_H(i, j)) & j = 1 \end{cases}$$

Put simply, each vertex in G is replaced by a *cloud* of vertices of H. The 0^{th} neighbor of vertex (v, i) is the $i^t h$ neighbor of v in G, and the j^{th} neighbor of (v, i) is the j^{th} neighbor of i in H. See Figure 2.1.

Theorem 2 (Informal)**.** *If G and H are expanders, then $G \circledR H$ is an expander.*

2.2 Zig-Zag Product

Given

- G a "*large*" on D-regular graph on N vertices.
- H a "*small*" on d-regular graph on D vertices.

The *zig-zag product* $G Ⓩ H$ is a d^2-regular graph on ND vertices. $G Ⓡ H$ has vertex set $V(G) \times V(H)$. The edges out of vertex (v, i) are indexed by elements of $[d]^2$.

Consider vertex $(v, i) \in V(G) \times V(H)$. Then the neighbor indexed by $(j_1, j_2) \in [d]^2$ is found in the following manner:

1. Set $i_2 = i[j_1] =$ the j_1^{th} neighbor of i in H.
2. Set $(v', i') = \mathsf{Rot}_G(v, i_2)$
3. Set $i_3 = i'[j_2] =$ the j_2^{th} neighbor of i' in H

The vertex (v', i_3) is the neighbor we want.

Motivation: One property that we know is associated with expanders is that random walks on them "sample well", and this is a way of saving randomness. Say we want to reduce randomness by using a random walk of length k on G, starting at a fixed vertex. We can do this easily with $O(k \log D)$ bits of randomness. Now if random walks on expanders are such a good way of generating randomness while using less randomness, perhaps we use this idea again: let us take a small expander H_0 on D vertices which is d-regular, and generate a k-step walk on H_0 using just $O(k \log d)$ random bits; then use the vertices of H_0 visited on the walk to tell us which edges to take in our random walk on G.

A random walk on $G Ⓩ H$ corresponds precisely to performing the above randomness-reduced random walk on G, using $H_0 = H^2$. Thus the zig-zag product arises naturally while trying to derandomize random walks on expanders with randomn walks on expanders.

Each step of the random walk on an edge of $G Ⓩ H$ can be regarded as a random step on a edge within a cloud, a deterministic step on an edge connecting two clouds and another random step within a cloud.

Theorem 3. *If G is a $D\lambda_G$ absolute eigenvalue expander and H is a $d\lambda_H$ absolute eigenvalue expander then $F = G Ⓩ H$ is a $d^2\lambda$ absolute eigenvalue expander with*

$$1 - \lambda \geq (1 - \lambda_H)^2 (1 - \lambda_G)$$

Proof. To prove the theorem we need the following lemma

Lemma 4. *Given G a N vertex D-regular graph, G is a $D\lambda$ absolute eigenvalue expander iff we can write*

$$A_G = (1 - \lambda) J_N + \lambda E$$

where A_G is the normalized adjacency matrix of G, J_N is the all $1/N$ matrix and $\|E\| \leq 1$ (i.e. E has max eigenvalue ≤ 1).

Proof. We can write

$$\begin{aligned}
A_G &= \sum_{i=1}^{n} \lambda_i v_i v_i^T \\
&= \lambda_1 v_1 v_1^T + \sum_{i=2}^{n} \lambda_i v_i v_i^T \\
&= J + \sum_{i=2}^{n} \lambda_i v_i v_i^T \\
&= (1-\lambda)J + \left(\lambda J + \sum_{i=2}^{n} \lambda_i v_i v_i^T\right) \\
&= (1-\lambda)J + \lambda E
\end{aligned}$$

In the last step, since the v_i are all orthogonal and $\lambda_i \leq \lambda$, it follows that E will have max eigenvalue ≤ 1.

□

We are now ready to prove Theorem 3.

Let $A_G \in \mathbb{R}^{V\times V}$, $A_H \in \mathbb{R}^{[D]\times[D]}$ and $A_F \in \mathbb{R}^{(V\times[D])\times(V\times[D])}$ be the normalized adjacency matrices of G, H and F respectively. Then we can write

$$A_F = BCB$$

where B, C are $|V|D \times |V|D$ matrices:

$$B[(u,i),(v,j)] = \begin{cases} 0 & u \neq v \\ A_H[i,j] & u = v \end{cases}$$

and

$$C[(u,i),(v,j)] = 1 \text{ iff } \mathsf{Rot}_G(u,i) = (v,j).$$

Note that $B = I \otimes A_H$ and that C is the adjacency matrix for a matching and is hence a permutation matrix. Hence we have:

$$\begin{aligned}
A_F &= BCB \\
&= (I \otimes A_H)C(I \otimes A_H)
\end{aligned}$$

From Lemma 4 we can write

$$A_H = (1-\lambda_H)J_D + \lambda_H E$$

So we get:

$$\begin{aligned}
A_F &= (I \otimes ((1-\lambda_H)J_D + \lambda_H E))C(I \otimes ((1-\lambda_H)J_D + \lambda_H E)) \\
&= (1-\lambda_H)^2(I \otimes J_D)C(I \otimes J_D) + (1-\lambda_H)\lambda_H(I \otimes J_D)C(I \otimes E) \\
&\quad +(1-\lambda_H)\lambda_H(I \otimes E)C(I \otimes J_D) + \lambda_H^2(I \otimes E)C(I \otimes E) \\
&= (1-\lambda_H)^2(I \otimes J_D)C(I \otimes J_D) + (1-(1-\lambda_H)^2)E'
\end{aligned}$$

where $||E'|| < 1$.

Observe that

$$(I \otimes J_D)C(I \otimes J_D) = (A_G \otimes J_N)$$

Using Lemma 4 again, we get:

$$\begin{aligned} A_G \otimes J_N &= ((1-\lambda_G)J_N + \lambda_G E_G) \otimes J_D \\ &= (1-\lambda_G)J_N \otimes J_D + \lambda_G(E_G \otimes J_D) \end{aligned}$$

Then

$$\begin{aligned} A_F &= (1-\lambda_H)^2(1-\lambda_G)J_N \otimes J_D + \lambda_G(E_G \otimes J_D) + (1-(1-\lambda_H)^2)E' \\ &= (1-\lambda_H)^2(1-\lambda_G)J_{ND} + (1-(1-\lambda_H)^2(1-\lambda_G))E'' \end{aligned}$$

where $||E''|| < 1$.

This implies that A_F is a λd^2 absolute eigenvalue expander, where

$$(1-\lambda) = (1-\lambda_H)^2(1-\lambda_G)$$

□

2.3 Construction

Let d, C be constants. Let H be a fixed d^{2C}-vertex d-regular graph which is a $\lambda_H d < d/2$ expander (which we find by brute force search in constant time, say).

- Start with any connected non-bipartite d^2-regular G_0 on N_0 vertices.
- We know that $\lambda_{G_0} d^2 < \left(1 - \frac{1}{N_0^3}\right) d^2$
- Now for $i = 0, 1, \ldots$, we repeat the following:
- Construct $G_{i'} = G_i^C$, which is d^{2C}-regular. Then we have that $G_{i'}$ is a $\lambda_{G_{i'}} d^{2C}$ expander, where $\lambda_{G_{i'}} = \lambda_{G_i}^C$.
- Zig-zag $G_{i'}$ with H to get G_{i+1} on $N_{i+1} = N_i d^{2C}$ vertices, which is d^2-regular.

By our analysis of the zig-zag product, and using the fact that $\lambda_H \leq 1/2$, we have that

$$1 - \lambda_{G_{i+1}} \geq \frac{1}{4}\left(1 - \lambda_{G_i}^C\right).$$

Let $\epsilon_i = 1 - \lambda_{G_i}$. We thus need to analyze the following process. Let $\epsilon_0 = \frac{1}{N_0^3}$, and $\epsilon_{i+1} \geq (1/4)(1-(1-\epsilon_i)^C)$. If $\epsilon_i < 0.01$, then we have

$$(1-\epsilon_i)^C \geq 1 - \frac{C\epsilon_i}{2}$$

. So we have $\epsilon_{i+1} \geq \frac{C\epsilon_i}{8}$, as long as $\epsilon_i < 0.01$. If we choose $C > 8$, we get that the ϵ_i grow geometrically, and so in $O(\log N_0)$ steps, ϵ_i becomes $\Omega(1)$, and we get a constant degree expander.

Note that for $i = O(\log N_0)$, we have $N_i = N_0 D^i = N_0 d^{2Ci} = poly(N_0)$, and so the final graph that we construct is polynomial in the size of the initial graph we start with.

3 Graph Connectivity

We now show that deciding Graph Connectivity can be done deterministically with small space complexity (we already saw a very simple randomized algorithm for this problem; getting a deterministic algorithm had been a very important open problem).

The problem easily reduces to testing, given G, s, t, if s is connected to t in G (this is called s-t connectivity).

Some classical observations about this problem:

1. The problem can be reduced to the case of G being constant degree.
2. If the diameter of G equals Δ, and all vertices of G have degree $\leq D$, then one can test s-t connectivity in space $\Delta \log D + O(\log n)$. This can be done by a DFS on the graph starting at s.

   In particular, on constant degree graphs this problem can be solved in $O(\Delta + \log n)$ space.
3. **Savitch's Algorithm** $conn(s, t, \Delta)$ (checks if s is connected to t by a path of length $\leq \Delta$):
   - Guess v
   - Check $conn(s, v, \Delta/2)$
   - Check $conn(t, v, \Delta/2)$
   - Return true iff both the above returned true.

   The space complexity $S(\Delta, n)$ satisfies: $S(\Delta, n) \leq \log(n) + S(\Delta/2, n)$ and thus that $S(\Delta, n) \leq \log(n)\log(\Delta)$. Thus this is a $O(\log^2(n))$ space deterministic algorithm for connectivity.

   Another way of viewing this algorithm is as follows. Given access to G we can simulate access to G^2 with $\log(n)$ extra bits of space. Given access to G^2 we can simulate access to G^4 with $\log(n)$ extra bits of space. Given access to G we can simulate access to $G^{2^{\log(n)}}$ with $\log^2(n)$ extra bits of space. Connectivity of s, t in G is simply adjacency of s, t in $G^{2^{\log n}}$. This gives us a $\log^2(n)$ space algorithm for connectivity.

Reingold's algorithm As in Savitch's algorithm, we try to improve "connectivity" of G, test s-t connectivity in new G. In Savitch, the measure of connectivity was diameter.

THE CRUX: Measure connectivity by absolute eigenvalue expansion.

How to improve eigenvalue expansion? Powering! How to bring it back to constant degree? Zig-zag!

Follow exactly the procedure for constructing expanders in the previous section, starting with G, and end up with a constant graph, each component of which is a good expander.

Good constant degree expanders have diameter $\leq O(\log n)$. We can check connectivity on such a graph in $O(\log n)$ space. This gives the connectivity algorithm.

Verifying that all this can indeed be carried out in log space requires some care (but it requires no new ideas).