Lecture 3: Randomness Efficient Error Reduction Using Expanders, Expander Construction Using Zig-Zag Product

Topics in Pseudorandomness and Complexity Theory (Spring 2018)
Rutgers University
Swastik Kopparty
Scribes: Amnon Attali, Jiyu Zhang

1 Error Reduction Algorithm

Remark. *Our goal: given an algorithm A(x,r) to compute some function* $f : \{0,1\}^n \to \{0,1\}$ *which satisfies*

$$\forall x \in \{0,1\}^n \quad \Pr_{r \in \{0,1\}^m}[A(x,r) \neq f(x)] \leq 0.1$$

we wish to produce a new algorithm A(x,r') such that*

$$\forall x \in \{0,1\}^n \quad \Pr_{r \in \{0,1\}^{m'}}[A^*(x,r') \neq f(x)] \leq 2^{-l}$$

where l is some controllable parameter that affects A's increased runtime. Note also* $m' \geq m$ *but "not by much", in the sense that this method uses less random bits than say taking the naive majority of multiple runs of the algorithm with fresh randomness.*
Our question now becomes: how small can r' be?

In this lecture we describe one such error reduction process using expander graphs.

Fix x.
Let $B = \{r \mid A(x,r) \neq f(x)\} \subseteq \{0,1\}^m \implies \frac{|B|}{s^m} \leq 0.1$
Let $N = 2^m,\ \beta = \frac{|B|}{s^m}$
Our strategy will now be to use r' to pick a set of t elements $r_1, ..., r_t \in \{0,1\}^m$ and then we will output $Maj(A(x,r_i))$ and so what we want is for the probability that more than half will be in B to be small.
Proposed solution:
Use m' = m + tlog(d), r' uniform random on $\{0,1\}^m$.
We construct a d-regular expander on $\{0,1\}^m$ and use the rest of the bits to perform a random walk of length t.
In this new setting what we want is that the probability of a random walk staying in B to be low (note that this is not true for all graphs, $t << 0.1 \times 2^m$ so for example on a grid if you start somewhere in the center of a "concentrated" B you will never leave in t steps).

We will start by considering the probability that all r_i are in B.
Let M be the normalized adjacency matrix of our d-regular λ-expander graph G.
Let P_B be the matrix which projects onto B, that is $(P_B)_{i,i} = 1\ if\ i \in B$ and 0 elsewhere. P_B

projection $\implies P_B^2 = P_B$
Let $f_0 \in \mathbb{R}^N$ be the uniform distribution
The quantity we wish to analyze involves the probability mass concentrated on B after t steps:

$$Pr[\forall i,\ r_i \in B] = \langle \vec{1}, (P_B M)^t P_B f_0 \rangle = \langle \vec{1}, (P_B M P_B)^t f_0 \rangle$$

where $P_B M$ represents the probability mass on B remaining after one step in the random walk, and the R.H.S comes from using that P_B is a projection (the reason we "insert" these extra projections is because it makes analysis easier - we have a better understanding of what distributions look like after we project onto B than simply after taking a random step in our walk).

Lemma 1.

$$\forall v \in \mathbb{R}^N \quad ||P_B M P_B v||_2 \leq (\beta + \lambda)||v||_2$$

Proof. Let $v_1, ..., v_N$ be the eigenvectors of M, forming an orthonormal basis
Let $u = P_B v$, $u = u_1 + u^* = \alpha_1 v_1 + u^*$, where $u_1 || v_1$ and $u^* \perp v_1$
$P_B M P_B v = P_B M u = P_B M(u_1 + u^*) = P_B \lambda_1 u_1 + P_B M u^* = P_B u_1 + P_B M u*$

Now since v_1 is the uniform vector $||P_B u_1||_2 = ||P_B \alpha_1 v_1||_2 \leq \alpha_1 \sqrt{\frac{|B|}{N}} = \alpha_1 \sqrt{\beta}$
And $\alpha_1 = \langle u, v_1 \rangle = \langle P_B v, v_1 \rangle = \langle v, P_B v_1 \rangle \leq ||P_B v_1||_2 ||v||_2 = \sqrt{\beta}||v||_2$ using that P_B is symmetric and then Cauchy-Schwarz So $||\alpha_1 P_B v_1||_2 \leq \beta ||v||_2$

Looking at our second term: $||P_B M u^*||_2 \leq ||M u^*||_2 \leq \lambda ||u^*||_2 \leq \lambda ||u||_2 \leq \lambda ||v||_2$ since P_B as a projection only shrinks size and the eigenvectors of M in the direction of u* all have eigenvalues at most λ in absolute value

Finally we have $||P_B u_1||_2 + ||P_B M u^*||_2 \leq (\beta + \lambda)||v||_2$

□

Now we have $||(P_B M P_B)^t f_0||_2 \leq (\beta + \lambda)^t ||f_0||_2 = (\beta + \lambda)^t \frac{1}{\sqrt{N}} \implies \langle \vec{1}, (P_B M P_B)^t f_0 \rangle \leq ||\vec{1}||_2 (\beta + \lambda)^t \frac{1}{\sqrt{N}} = \sqrt{N}(\beta + \lambda)^t \frac{1}{\sqrt{N}} = (\beta + \lambda)^t$

So the $Pr[\forall i,\ r_i \in B] \leq (\beta + \lambda)^t$

We now want the probability that more than half are in B, we do this with a simple union bound:

Fix $S \subseteq [t]$ of size $\frac{t}{2}$

Claim 2.

$$Pr[\{r_i |\ i \in S\} \subseteq B] \leq (\beta + \lambda)^{t/2}$$

Proof. First note that $P_B M^i P_B$ scales a vector by $(\beta + \lambda^i) \leq (\beta + \lambda)$, as above.
So

$$Pr[\{r_{i_1}, r_{i_2}, ..., r_{i_{t/2}}\} \subseteq B] = \langle \vec{1}, (P_B M^{i_{t/2} - i_{t/2-1}} P_B)(...)(P_B M^{i_2 - i_1} P_B) f_0 \rangle \leq$$
$$\langle \vec{1}, (P_B M P_B)^t f_0 \rangle \leq (\beta + \lambda)^{t/2}$$

□

Given the claim we now apply a union bound:

$$Pr[\exists \text{ such a bad } S] \leq (\#S's)(\beta+\lambda)^{t/2} = \binom{t}{t/2}(\beta+\lambda)^{t/2} \leq 2^t(\beta+\lambda)^{t/2}$$

So for $(\beta+\lambda) \leq 0.25$ this is $2^{\Omega(-t)}$, and we are done.

Conclusion:

Given a strongly explicit 0.1-expander (note we can reduce 0.9 to 0.1, say by squaring the graph), we can take any randomized algorithm that uses m bits of randomness, runs in time T, and has 0.1 probability of error (this too can be reduced from some higher value without using more randomness, say by repeating and taking majority), and convert it to another algorithm which uses $m + \mathbb{O}(t\log(d))$ bits and runs in time $tT + poly(m)$ time and has probability $2^{\Omega(-t)}$ of error.

This process for decreasing error without using too much randomness is optimal in the sense that we don't use any properties of the algorithm A (of the set B), and rather treat it as a black box and try to generate a t tuple of strings which contains fewer than half in the set B.

Our next goal is to construct such expander graphs as can be used for this purpose.

2 Construction of Expander

In this class we study the construction of expander based on *the zig − zag product* suggested by Reingold, Vadhan and Wigderson[RVW02], The general idea is to start with some graph and we keep improving its expansion until it is ideal, while keeping the degree of the graph fixed(roughly). Notice that one way we have already studied to improve expansion is by squaring: for an (n, d, λ)-*graph* G, the *k-th* power of G, denoted by G^k, is an (n, d^k, λ^k)-*graph*. But in this way the degree increases exponentially. Thus we introduce *the zig − zag product*, which takes a product of a large graph with a small graph, the resulting graph inherits the size of the large one and the size of the small one, and improves its expansion from both of the graphs. We first give the resulting theorem here to give a sense of the description above. Later we'll mention it again with detailed notations and proofs.

Theorem 3. *Let G be a N-vertices, D-regular, λ_G-expander graph, denoted by (N, D, λ_G)-graph. Let H be a (D, d, λ_H)-graph. Then the zig − zag product of G and H, denoted by GH, is an $(ND, d^2, 1-(1-\lambda_H)^2(1-\lambda_G))$-graph.*

The Replacement Product

We'll first describe a simpler product, *the replacement product*.
To introduce the products we first define some notations. The *Rotation Map* $Rot_G : V \times [D] \to V \times [D]$ performs as follows:

$$Rot_G(u, i) = (v, j)$$

where v is the *i-th* neighbor of u and u is the *j-th* neighbor of v.
Now given a "small" d-regular graph H and a "large" D-regular graph G, assume that for each

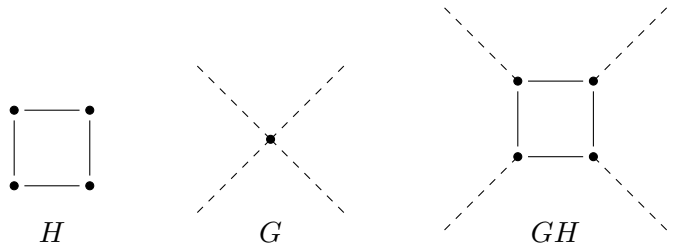

Figure 1: Replacement product of a graph

vertex of G there's some ordering on its D neighbors. The *replacement product* denoted by GH is as follows:

- Every vertex in G is replaced by vertices of H(We call it a cloud). The vertex denoted by (u, v) where $u \in V(G)$ *and* $v \in V(H)$ is the v-*th* vertex in the u- cloud.
- Let $(u, v) \in E(G)$, Then the edge$((u, i), (v, j)) \in E(GH)$ if $Rot_G(u, i) = (v, j)$ ($i.e$ if v is the i-*th* neighbor of u and u is the j-*th* neighbor of v). Also if $(i, j) \in E(H)$, then $\forall u \in V(G), ((u, i), (v, j)) \in E(GH)$

The Zig-Zag Product

Now we proceed to describe the construction of the zig-zag product. Given

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

The zig-zag product is constructed as follows:

- The vertices $V(G) \times V(H)$ of $V(GH)$ are the same as in the case of the replacement product.
- The edges are defined by a "zig-zag walk": Consider vertex $(i, j) \in V(G) \times V(H)$, the neighbor indexed by $(k_1, k_2) \in [d]^2$(where $d = N \cdot D$ in our case) is found in the following way:
  1. Let $j^{'}$ be the k_1-*th* neighbor of (i, j) in the cloud.
  2. Let $(i^*, j^*) = Rot_G(i, j^{'})$.
  3. Let $j^{''}$ be the k_2-*th* neighbor of (i^*, j^*) in the cloud(Notice that this cloud is different from that in step 1).

  The vertex $(i^*, j^{''})$ is the neighbor we want(there is an edge between (i, j) and $(i^*, j^{''})$.

Alternatively, you can interpret as such that $(i^*, j^{''})$ can be reached from (i, j) by taking a step in the first cloud, then a step between the clouds, and a step in the second cloud.(hence zig-zag)

Analysis of the Construction

Theorem 4. *GH is a d^2-regular $1-(1-\lambda_H)^2(1-\lambda_G)$-expander graph.*

We will be using the following lemma:

Lemma 5. *If G is a N-vertex, d-regular, λ-expander, then M_G(the normalized adjacency matrix of G) can be written as $(1-\lambda)J+\lambda E$.*

where J is the all $1/N$ matrix, and $\|E\| \leq 1$.
Note that $\|E\|$ is the norm of the matrix E, which is defined as:

$$\|E\| = \max_{x \in R^n, \|x\|=1} \|Ex\|$$

That is, the maximum eigenvalue of E.

Proof. Since M_G is symmetric, we can write it as

$$\begin{aligned} M_G &= \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \cdots + \lambda_n \mathbf{v}_n \mathbf{v}_n^T \\ &= J + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \cdots + \lambda_n \mathbf{v}_n \mathbf{v}_n^T \\ &= (1-\lambda)J + \lambda J + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \cdots + \lambda_n \mathbf{v}_n \mathbf{v}_n^T \\ &= (1-\lambda)J + \lambda E \end{aligned}$$

In the above, $\lambda_1 = 1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ are the eigenvalues of M_G and $\mathbf{v}_1 = \frac{1}{\sqrt{n}}, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are the corresponding system of orthogonal eigenvectors. □

Now let M be the normalized adjacency matrix of GH of size $ND \times ND$, M_H be the normalized adjacency matrix of H of size $D \times D$. M can be written as:

$$M = ABA$$

where $A = I \otimes M_H$ and B is the adjacency matrix for a permutation.
Therefore we have

$$\begin{aligned} M &= ABA \\ &= (I \otimes M_H) B (I \otimes M_H) \end{aligned}$$

Utilizing *Lemma* 4 we have $M_H = (1-\lambda_H)J + \lambda_H E$. We get:

$$\begin{aligned} M &= (I \otimes ((1-\lambda_H)J + \lambda_H E)) B (I \otimes ((1-\lambda_H)J + \lambda_H E)) \\ &= (1-\lambda_H)^2 (I \otimes J) B (I \otimes J) + (1-\lambda_H)\lambda_H (I \otimes J) B (I \otimes E) \\ &+ (1-\lambda_H)\lambda_H (I \otimes E) B (I \otimes J) + \lambda_H^2 (I \otimes E) B (I \otimes E) \\ &= (1-\lambda_H)^2 (I \otimes J) B (I \otimes J) + (1-(1-\lambda_H)^2) E' \end{aligned}$$

where $E' \leq 1$, and $I \otimes J$ is the *tensor product* of I and J.