Lecture 11: The Zig-Zag Product

Topics in Error-Correcting Codes (Fall 2022)
University of Toronto
Swastik Kopparty
Scribes: Anvith Thudi and Arkaprava Choudhury

Agenda

In today's lecture we will show how to construct explicit expanders with constant degree, i.e., $d = O(1)$. This will be done using Zig-Zag products.

To understand the zig-zag prodcut, one might consider the following "philosophy" in how to simulate random walks: instead of picking t independent random edges, we can use expanders to pick our edges ("expander walks"). This will reduce the degree of our graph, and much of what we analyze in the rest of this lecture is whether this still preserves good absolute spectral expansion.

1 The Zig-Zag Product

The high-level goal is to pick random walks (i.e., t-tuple from $[n]^t$) that don't lose our original expansion properties, but somehow use less randomness. Normal random walks pick $i_1, \cdots, i_t \in [n]$ independently. An alternative strategy is the following:

1. Pick $i \in [N]$, and $j_1, \cdots, j_{t-1} \in [D]$
2. take $i_1 = i$
3. take i_2 as j_1^{th} neighbour of i_1
4. take i_m as j_{m-1}^{th} neighbour of i_{m-1}

This approach uses less "randomness" as we are sampling from $[D]$ instead of $[N]$. We can in fact lessen this further by picking j_i according to an expander walk on $[D]$ vertices. This operation is formalized in the notion of a replacement product:

Definition 1 (Replacement Product)**.** *Consider G on N vertices with degree D, and H on D vertices with degree d. Then $G \circledR H$ is a graph on ND vertices with degree d defined as follows:*

1. *$V = [N] \times [D]$*
2. *For $l \in [d]$, l^{th} neighbour of $(i, j) \in V$ is (i'', j'') where*
3. *$j' = l^{th}$ neighbour of j in H*

4. $i'' = j'^{th}$ *neighbour of* i *in* G

5. (i, j') *corresponds to the* $j'th$ *edge coming from* i*, we take* (i'', j'') *to be the other vertex incident to that edge.*

The Zig-Zag product uses the same idea, but now simulates two steps using the expander. This conveniently makes the graph stay undirected (i.e., is symmetric)

Definition 2 (Zig-Zag Product)**.** *Consider* G *and* H *as in the previous definition. Then* $G \textcircled{Z} H$ *is the graph on* ND *vertices with degree* d^2 *defined as follows:*

1. $V = [N] \times [D]$

2. *For* $l_1, l_2 \in [d]$ *the* (l_1, l_2) *neighbour of* (i, j) *is* (i'', j''') *defined as follows*

3. $j' = j[l_1]$ *in* H

4. $i'' = i[j']$ *in* G

5. j'' *is the edge number from* i'' *leading back to* i

6. $j''' = j''[l_2]$ *in* H

As we are using expanders to simulate walks (which have good expansion properties already), our hope is that we did not lose too much of the expansion properties of G. This is formulated in the following claim.

Claim 3 (Goal)**.** $G \textcircled{Z} H$ *is a* λ*-expander where* $\lambda = (1 - (1 - \frac{\lambda_H}{d})^2(1 - \frac{\lambda_G}{D}))$*. That is, it is good expander if* H *and* G *are too.*

In proving this claim, we will make use of the following claim

Claim 4. *If* F *is a c-regular graph on* $[n]$ *vertices, then* F *is a* λ*- absolute expander iff* $\frac{1}{c}A_F = (1 - \frac{\lambda}{c})J_n + \frac{\lambda}{c}E$ *where* $||E|| \leq 1$

Proof. $A_F = \sum_{i=1}^{n} \lambda_i v_i v_i^T$ where v_i are eigenvectors of A_f. Note $\sum_{i=1}^{n} \lambda_i v_i v_i^T = cv_1 v_1^T + \sum_{j \geq 2} \lambda_j v_j v_j^T = cJ_c + \sum_{j \geq 2} \lambda_j v_j v_j^T$.

So we have $\frac{1}{c}A_F = J + \sum_{j \geq 2} \frac{\lambda_j}{c} v_j v_j^T = (1 - \frac{\lambda}{c})J + \frac{\lambda}{c} v_1 v_1^T + \sum_{j \geq 2} \frac{\lambda_j}{c} v_j v_j^T$, and now we simply define $\frac{\lambda}{c}E$ to be our last two terms. Note $E = v_1 v_1^T + \sum_{j \geq 2} \frac{\lambda_j}{\lambda} v_j v_j^T$ so $||E|| = max(1, \lambda_i/\lambda) = 1$ □

In analyzing the zig-zag product we will also use tensor products; we recall the definition below.

Definition 5 (Tensor Product)**.** *The tensor product of* M_1 *and* M_2*, where* M_1 *is* $a \times b$ *and* M_2 *is* $a' \times b'$*, is an* $aa' \times bb'$ *matrix* $M_1 \otimes M_2$ *where* $(M_1 \otimes M_2)_{(u,u'),(v,v')} = M_{1_{(u,v)}} M_{2_{(u',v')}}$

With this we now approach analyzing the spectral expansion of the zig-zag product. Letting $Y = A_{G \textcircled{Z} H}$, we will show that $\frac{1}{d^2}Y = (1 - \ small)J_{ND} + \ small \ * E$ for some $||E|| \leq 1$, which is sufficient by Claim 4. The *small* will be our absolute spectral expansion stated in Claim 3.

Claim 3. Consider the operations that defined the zig-zag product. We first stayed on the same clique (coming from replacing vertices in G with H), but moved within the clique according to A_H. We then did a permutation/relabeling corresponding to G. Then we once again moved in the new clique according to A_H. This sequence of operations is equivalent to $(A_H \otimes I_N)\Pi(A_H \otimes I_N)$, where Π is the permuation according to G, by how tensor products are defined.

Now note, by H being a λ_H-expander, we have by Claim 4 that $\frac{1}{d}A_H = (1 - \frac{\lambda_H}{d})J_D + \frac{\lambda_H}{d}E_H$. One can now use this and the linearity of tensor products to get the following:

$$\begin{aligned} \frac{1}{d^2}Y = (1-\frac{\lambda_H}{d})^2(J_D \otimes I_N)\Pi(J_D \otimes I_N) \\ +(1-\frac{\lambda_H}{d})\frac{\lambda_H}{d}(J_D \otimes I_N)\Pi(E_H \otimes I_N) \\ +(1-\frac{\lambda_H}{d})\frac{\lambda_H}{d}(E_H \otimes I_N)\Pi(J_D \otimes I_N) \\ +(\frac{\lambda_H}{d})^2(E_H \otimes I_N)\Pi(E_H \otimes I_N) \end{aligned} \tag{1}$$

While this seems slightly ugly, note in fact all the terms other than the first term are "small" when we have H is a good-expander. So now we analyze the first term. Note that in $(J_D \otimes I_N)\Pi(J_D \otimes I_N)$ the Π simply shifts the uniform probabilites according to transition matrix of G. Thus $(J_D \otimes I_N)\Pi(J_D \otimes I_N) = (J_D \otimes I_N)(I_D \otimes \frac{1}{D}A_G)(J_D \otimes I_N)$. Now we can use the fact multiplication of tensor products goes through in the natural way, and see $(J_D \otimes I_N)(I_D \otimes \frac{1}{D}A_G)(J_D \otimes I_N) = (J_D \otimes \frac{1}{D}A_G)(J_D \otimes I_N) = (J_D \otimes \frac{1}{D}A_G)$ where we used $J_D^2 = J_D$.

So the first term is $(1 - \frac{\lambda_H}{d})^2(J_D \otimes \frac{1}{D}A_G)$ and now using Claim 4 on $\frac{1}{D}A_G$ we have it equals $(1 - \frac{\lambda_H}{d})^2(1 - \frac{\lambda_G}{D})(J_D \otimes J_N) + (1 - \frac{\lambda_H}{d})^2\frac{\lambda_G}{D}(J_D \otimes E_G)$

So going back to $\frac{1}{d^2}Y$, we can express $\frac{1}{d^2}Y = (1-\frac{\lambda_H}{d})^2(1-\frac{\lambda_G}{D})(J_D \otimes J_N) + (1-(1-\frac{\lambda_H}{d})^2(1-\frac{\lambda_G}{D}))E$ for some E s.t $||E|| \leq 1$.

This completes the desired proof. □

2 Constructing Good Expanders

The Zig-Zag product gave us a way of reducing degree while still maintaining good spectral expansion. If we now have a method to improve spectral expansion (which might also increase degree), we could couple this with the zig-zag product and take bad expanders and make them good (with constant degree). This will only require some apriori good expander H to use the zig-zag product with, but we can brute force search for such expanders if they are small.

The operation will will use to boost the absolute spectral expansion is graph powering (a generalization of just squaring):

Definition 6 (Graph Powering)**.** *Given a graph G, G^i is the graph defined by A_G^i. In particular, note $\lambda_{G^i} = \lambda_G^i$.*

We now can give a simple algorithm to construct good expanders:

1. Start with $G_0 = G$ with N vertices and d^2-regular. Say $\lambda_0 :- \lambda_{G_0} = d^2(1-\epsilon_0)$ for $\epsilon_0 = \frac{1}{N}$. Note, we know such graphs exist.

2. Take $G_{i+1} = G_i^{10} ⓩ H$ where H has d^{20} vertices and degree d with $\lambda_H \leq (0.1)d$. Note we can brute-force to construct such an H.

Note that G_{i+1} is still d^2-regular, and $1 - \frac{\lambda_{G_{i+1}}}{d^2} \geq (1 - \frac{\lambda_H}{d})^2(1 - (\frac{\lambda_{G_i}}{d^2})^{10})$. So now looking at $\epsilon_i = 1 - \frac{\lambda_{G_{i+1}}}{d^2}$, we see $\epsilon_{i+1} \geq (1-(1-\epsilon_i)^{10})(0.9)^2 \geq 2\epsilon_i$.

Thus we double the absolute spectral gap with each iteration of the algorithm, and hence doing it $O(log(n))$ times we will get good epanders (i.e., absolute spectral expansion of the form $d^2(1-O(1))$ as the initial $O(1/N)$ becomes $O(1)$).