Doctoral mini-course on expander graphs
December 12, 2024 – Lecture 3

1 Preliminaries

We recall the following theorem, stated and used in the last lecture.

Theorem 1. *Let G be an (n, d, λ)-graph. Suppose we take a vertex x_0 uniformly at random in G, and then perform a random walk $x_0, \ldots, x_t$ of length t starting at x_0. Then for any subset $S \subseteq V(G)$, the probability that $x_0, x_1, \ldots, x_t$ are all in S is at most $(|S|/n + \lambda/d)^t$. Moreover, if $|S|/n \geq 6\lambda/d$, then this probability is at least $(|S|/n - 2\lambda/d)^t$.*

2 Hardness of approximation

The PCP theorem, mentioned in the previous lecture, can equivalently be stated as a result on the hardness of approximating combinatorial problems. Recall that a *clique* in a graph is a set of pairwise adjacent vertices. The *clique number* of G, denoted by $\omega(G)$, is the maximum size of a clique in G.

Theorem 2. *There exist constants $0 < a < b < 1$ such that given an n-vertex graph G for which*

(1) $\omega(G) \leq an$, or

(2) $\omega(G) \geq bn$,

there is no polynomial time algorithm distinguishing between the two cases (1) and (2), unless $\mathsf{P} = \mathsf{NP}$.

In particular, we cannot approximate the clique number in polynomial time within a factor of $b/a - \varepsilon$ unless $\mathsf{P} = \mathsf{NP}$. (We say that an algorithm $\mathcal{A}$ *approximates the clique number within a factor* $c(n) > 1$ if for any graph G on n vertices, $\omega(G)/c(n) \leq \mathcal{A}(G) \leq \omega(G)$). Indeed: if we have an algorithm $\mathcal{B}$ which approximates the clique number within a factor of $(b/a) - \varepsilon$ (i.e., such that $\omega(G)/(b/a - \varepsilon) \leq \mathcal{B}(G) \leq \omega(G)$), then we can run $\mathcal{B}$ and answer (1) if $\mathcal{B}(G) \leq an$, and (2) otherwise.

Our first observation is that we can make the constant in the inapproximability result arbitrarily large.

Assume there is no polynomial time algorithm that approximates the clique number within c_1. Let $c_2 > c_1$, and assume that some polynomial time algorithm $\mathcal{B}$ approximates the clique number within c_2.

Let G be a graph on n vertices. Consider $G * k$, the graph whose vertices are the k-tuples of vertices of G, with adjacency between two k-tuples if their union is a clique of G. It is not hard to check that $\omega(G * k) = \omega(G)^k$: in one direction, any clique of size ℓ in G produces ℓ^k k-tuples that are pairwise adjacent in $G * k$, and in the other direction, in any clique of size at least ℓ^k in $G * k$ we must see at least ℓ distinct vertices of G in the k-tuples, and the union of these ℓ elements forms a clique in G.

Take $k \geq \log c_2 / \log c_1$, and compute $\mathcal{A}(G) = (\mathcal{B}(G * k))^{1/k}$. Then

$$\omega(G)/c_1 \leq \omega(G)/c_2^{1/k} \leq \mathcal{A}(G) \leq \omega(G),$$

and thus the result approximates the clique number of G within c_1. Since $G * k$ has size polynomial in G, the new algorithm $\mathcal{A}$ is also polynomial.

Using Theorem 2, this shows that for any constant $c > 0$, there is no polynomial time algorithm approximating the clique number within factor c, unless $\mathsf{P} = \mathsf{NP}$. We will now use expanders to push the inaproximability ratio further.

Theorem 3. *There is $\varepsilon > 0$ such that there is no polynomial time algorithm approximating the clique number in n-vertex graphs within factor n^ε, unless $\mathsf{P} = \mathsf{NP}$.*

The remainder of the section is devoted to the proof of Theorem 3.

Let G be an n-vertex graph, and let F be an (n, d, λ)-graph on the same vertex set as G. For some integer t, we define a new graph H whose vertices are the $t+1$-vertex walks $x_0, \ldots, x_t$ in F, and in which two walks are adjacent if and only if their union is contained in some clique of G. Note that F has $N = nd^t$ vertices, which is polynomial in n whenever $t = O(\log n)$.

We claim that if $\omega(G) \leq an$, then $\omega(H) \leq (a + \lambda/d)^t N$.

To see this, we start by observing that if C is clique in H, the union of the vertex sets of all the corresponding walks in F corresponds to some clique C' in G. Theorem 1 says that the proportion of walks of length t of F that remain inside C' is at most $(a + \lambda/d)^t$, since $|C'| \leq an$. It follows that $|C| \leq (a + \lambda/d)^t N$, as desired.

As a set of walks of F confined in some clique of G forms a clique in H, we obtain the following immediate corollary of Theorem 1.

Corollary 4. *If $\omega(G) \geq bn$ and $\lambda \leq bd/6$, then $\omega(H) \geq (b - 2\lambda/d)^t N$.*

By taking λ/d sufficiently small (as a function of a and b) we find two constants $\alpha < \beta$ such that $\omega(G) \leq an$ implies $\omega(H) \leq \alpha^t N$, and $\omega(G) \geq bn$ implies $\omega(H) \geq \beta^t N$.

With $t = \log n$, the ratio between the two bounds is $(\beta/\alpha)^t = n^\delta = N^\varepsilon$ (for some $\delta, \varepsilon > 0$), so if we know how to approximate the clique number within a factor of (number of vertices)$^\varepsilon$, we can distinguish between (1) $\omega(G) \leq an$ and (2) $\omega(G) \geq bn$ in polynomial time. By Theorem 2, it follows that there is no polynomial time algorithm approximating the clique number in n-vertex graphs within factor n^ε, as desired.

This is not the best known result: it is known that the clique number is not approximable within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ (unless $\mathsf{P} = \mathsf{NP}$). On the other hand there is a simple n-approximation algorithm: always output 1.

3 Zig-zag product

In this section we describe a purely combinatorial (an algorithmically efficient) construction of a family of expander graphs, due to Reingold, Vadhan and Wigderson (2002). We start by defining the zig-zag product of two graphs.

We consider a graph G (a small "red" graph), which is d-regular and has D vertices, and a graph H (a large "blue" graph) which is D-regular and has n vertices.

We replace each vertex v of H by a copy G_v of G, and we use each edge uv of H to connect a vertex of G_u to a vertex of G_v (we view the edge uv as being colored blue). Note that as G has D vertices and H is D-regular, we can make sure that for every $v \in V(G)$, each vertex of G_v is incident to a single blue edge. Let Z denote the resulting (edge-colored) graph.

Now the *zig-zag product of H and G*, denoted by $H \otimes G$, is the graph on the same vertex as the graph Z described above, with an edge between $x \in G_u$ and $y \in G_v$ if and only if there is red-blue-red path on 3 edges between x and y in Z. See Figure 1 for an illustration of $K_4 \otimes K_3$. We emphasize that this construction can produce different graphs, depending on the choices of endpoints of the blue edges. Whenever we write "the graph $H \otimes G$" in the remainder, what we really mean is "any graph that can be produced by taking the zig-zag product of H and G".

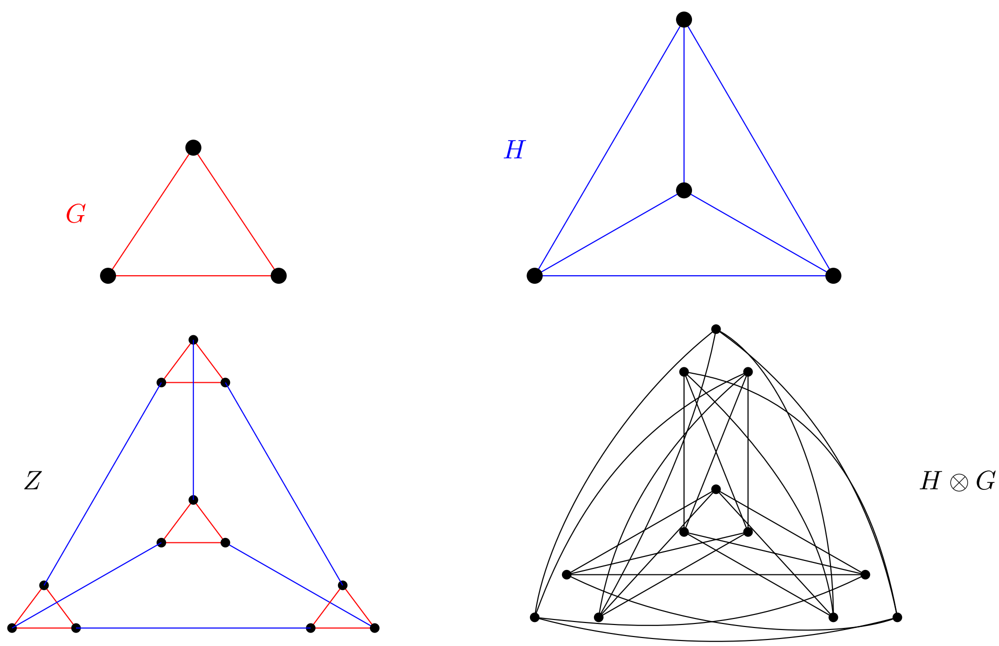

Figure 1: The zig-zag product of $H = K_4$ and $G = K_3$.

For an (n, d, λ)-graph G, we write $\lambda^*(G) = \lambda/d$.

Theorem 5. *The graph $H \otimes G$ has Dn vertices, is d^2-regular, and*

$$\lambda^*(H \otimes G) \leq \lambda^*(G) + \lambda^*(H).$$

We now explain how the zig-zag construction (and Theorem 5) can be used to construct families of expander graphs. We will need the following definition. For a graph G, G^2 stands for the *square* of G: this is the graph with the same vertex set as G in which we add an edge between two vertices for any path of length two between them (this graph will have multiple edges and loops). Note that if G is an (n, d, λ)-graph, then G^2 is an (n, d^2, λ^2)-graph, since the adjacency matrix of G^2 is the square of the adjacency matrix of G.

We start with a small d-regular graph G with $D = d^4$ vertices, and $\lambda^*(G) < 1/4$ (we find G by exhaustive search, such a graph exists as soon as d is large enough).

We set $G_1 = G^2$, and then $G_{i+1} = (G_i)^2 \otimes G$ for any $i \geq 1$. So for any $i \geq 1$, G_i has D^i vertices and degree d^2 (so after $\Omega(\log n)$ iterations we obtain a graph of size close to n). Note that G_{i+1} is well defined (the zig-zag product is legal): indeed, $(G_i)^2$ has degree $(d^2)^2 = d^4 = D$ and G has D vertices.

We claim that $\lambda^*(G_i) < 1/2$. This is due to the fact that $\lambda^*(G_i) < \lambda^*(G_{i-1}^2) + \lambda^*(G) < (1/2)^2 + 1/4 = 1/2$. The main idea here is that the squaring improves the expansion sufficiently, so that even after the loss coming the zig-zag product, λ^* remains bounded away from 1.

Observe that we can compute G_t entirely in time polynomial in the size of G_t. After a few modifications to the procedure, the computation of G_t can even be made strongly explicit (that is, given a vertex x, we can compute the list of neighbors of x in time polylogarithmic in the size of G_t).

Next week we will see how to use this construction to prove results on the complexity of s, t-*connectivity*, the problem consisting in deciding whether two vertices s, t of a graph are in the same connected component. This is a major result of Reingold (2005).