Explicit Construction of Expander Graphs

Following Arora and Barak, Chapter 21.3

Based on Arora & Barak

Rotation Maps

- For a d-regular n-vertex graph G, a *rotation map*

$$\hat{G} : [n] \times [d] \to [n] \times [d]$$

  maps $\langle v, i \rangle$ to $\langle u, j \rangle$ where u is the i-th neighbor of v and v is the j-th neighbor of u.
- $\hat{G}$ is a permutation of $[n] \times [d]$.
- Enables local computation of neighbors without global renumbering.
- Crucial for strongly explicit constructions (poly$(\log n)$ time neighbor queries).

Matrix/Path Product

Definition

For n-vertex graphs G, G' with degrees d, d' and random-walk matrices A, A', the graph $G' \cdot G$ is given by the random-walk matrix $A'A$.

- n vertices, degree dd'.
- An edge corresponds to a length-2 path: one step on G, then one on G'.
- Special case: $G^k = G \cdot G \cdots G$ (k times) is graph powering.

Expansion Guarantee (Lemma 21.16)

$$\lambda(G' \cdot G) \leq \lambda(G') \cdot \lambda(G).$$

In particular, $\lambda(G^k) = \lambda(G)^k$.

Effect: Improves expansion (reduces λ) at the cost of higher degree.

Tensor Product

Definition

For graphs G (n vertices, degree d) and G' (n' vertices, degree d'), the tensor product $G \otimes G'$ has

- nn' vertices, degree dd'.
- Rotation map:

$$\widehat{G \otimes G'}(\langle u, v\rangle, \langle i, j\rangle) = \langle u', v'\rangle, \langle i', j'\rangle$$

where $(u', i') = \hat{G}(u, i)$ and $(v', j') = \hat{G}'(v, j)$.

Expansion Guarantee (Lemma 21.17)

$$\lambda(G \otimes G') \leq \max\{\lambda(G), \lambda(G')\}.$$

Effect: Increases number of vertices while roughly preserving expansion (degree multiplies).

Replacement Product

Definition

Let G be n-vertex degree-D graph and H be D-vertex degree-d graph. The replacement product $G \circ_R H$ is the nD-vertex, degree-$2d$ graph obtained by:

1. Placing a copy of H at each vertex of G (clusters).
2. For each edge (u, v) of G with labels i, j, adding d parallel edges between the i-th vertex of u's cluster and the j-th vertex of v's cluster.

Random-walk matrix

$$A \circ_R A' = \frac{1}{2}\hat{A} + \frac{1}{2}(I_n \otimes A'),$$

where $\hat{A}$ is the permutation matrix of $\hat{G}$.

Expansion of Replacement Product

Lemma 21.18

If $\lambda(G) \leq 1 - \varepsilon$ and $\lambda(H) \leq 1 - \delta$, then

$$\lambda(G \circ_R H) \leq 1 - \frac{\varepsilon\delta^2}{24}.$$

- Dramatic degree reduction ($D \to 2d$) with only mild loss in expansion.
- Intuition: use a short expander walk on H to generate edge labels for a walk on the high-degree expander G.
- Proof idea: analyze the cube of the random-walk matrix and use the spectral decomposition of H.

High-Level Strategy

Recursive construction of a family $\{G_k\}$:

- **Tensor product:** Increase number of vertices (from $G_{\lfloor (k-1)/2 \rfloor}$ and $G_{\lceil (k-1)/2 \rceil}$).
- **Matrix product (powering):** Improve expansion (raise to the 50th power).
- **Replacement product:** Reduce degree back to constant (using a fixed small expander H).

Base cases found by brute-force search (constant size).

Base Cases and Recursion

Fix a large enough constant d so that the following exist (by probabilistic method + brute force):

- H: a $(D = (2d)^{100}, d, 0.01)$-expander.
- G_1: a $((2d)^{100}, 2d, 1/2)$-expander.
- G_2: a $((2d)^{200}, 2d, 1/2)$-expander.

Recursive definition (for $k > 2$)

$$G_k = \left(G_{\lfloor (k-1)/2 \rfloor} \otimes G_{\lceil (k-1)/2 \rceil} \right)^{50} \circ_R H.$$

The Claim

Claim

For every $k \geq 1$, G_k is a

$$\left((2d)^{100k},\, 2d,\, 1 - 1/50\right)\text{-expander.}$$

Moreover, there is a poly(k)-time algorithm that, given a vertex label i in G_k and an index $j \in [2d]$, returns the j-th neighbor of i.

- Yields a strongly explicit expander family on $n = c^k$ vertices for constant $c = (2d)^{100}$.
- Extendable to all n by mega-vertex collapsing (Exercise 21.16).

Proof: Size and Degree (Induction)

- Base cases $k = 1, 2$ hold by construction.
- Let n_k be the number of vertices of G_k.
- By induction,
$$n_{\lfloor (k-1)/2 \rfloor} = (2d)^{100 \cdot \lfloor (k-1)/2 \rfloor}.$$
- Tensor product multiplies vertex counts; replacement multiplies by $D = (2d)^{100}$:
$$n_k = n_{\lfloor (k-1)/2 \rfloor} \cdot n_{\lceil (k-1)/2 \rceil} \cdot (2d)^{100} = (2d)^{100k}.$$
- Degree: tensor gives degree $(2d)^2$; 50th power gives degree $(2d)^{100}$; replacement with H restores degree $2d$.

Proof: Expansion (Induction)

Assume $\lambda(G_m) \leq 1 - 1/50$ for $m < k$.

1. By Lemma 21.17,

$$\lambda(G_{\lfloor (k-1)/2 \rfloor} \otimes G_{\lceil (k-1)/2 \rceil}) \leq 1 - 1/50.$$

2. By Lemma 21.16 (powering),

$$\lambda\left((G \otimes G)^{50}\right) \leq \left(1 - 1/50\right)^{50} < 1/e < 1/2.$$

3. Let $\varepsilon = 1/2$ and $\delta = 0.99$ (since $\lambda(H) \leq 0.01$). Lemma 21.18 yields

$$\lambda(G_k) \leq 1 - \frac{(1/2) \cdot (0.99)^2}{24} \leq 1 - \frac{1}{50}.$$

Strong Explicitness

- The neighborhood function of G_k is computed by a constant number of recursive calls to the neighborhood functions of the two half-size graphs (plus the fixed H).
- Depth of recursion is $O(\log k)$; each level performs $O(1)$ work plus 50 matrix-product steps that themselves expand into a constant number of recursive calls.
- Overall running time is $\text{poly}(k) = \text{poly}(\log n)$, hence strongly explicit.

Theorem 21.19

There exists a strongly explicit (λ, d)-expander family for constants $d \in \mathbb{N}$ and $\lambda < 1$.

(Using further powering, any constant $\lambda > 0$ is achievable at the expense of a larger constant degree.)