CS 598: Spectral Graph Theory. Lecture 13

Expander Codes

Alexandra Kolla

Today

- Graph approximations, part 2. Bipartite expanders as approximations of the bipartite complete graph
- Quasi-random properties of bipartite expanders, expander mixing lemma
- Building expander codes
- Encoding, minimum distance, decoding

Expander Graph Refresher

- We defined expander graphs to be d-regular graphs whose adjacency matrix eigenvalues satisfy

$$|\alpha_i| \leq \epsilon d$$

for i>1, and some small ϵ.

- We saw that such graphs are very good approximations of the complete graph.

Expanders as Approximations of the Complete Graph Refresher

- Let G be a d-regular graph whose adjacency eigenvalues satisfy $|\alpha_i| \leq \epsilon d$.

- As its Laplacian eigenvalues satisfy $\lambda_i = d - \alpha_i$, all non-zero eigenvalues are between $(1-\epsilon)$d and $(1+\epsilon)d$.

- Let H=(d/n)Kn, $so\ \ x^T L_H x = d x^T x$

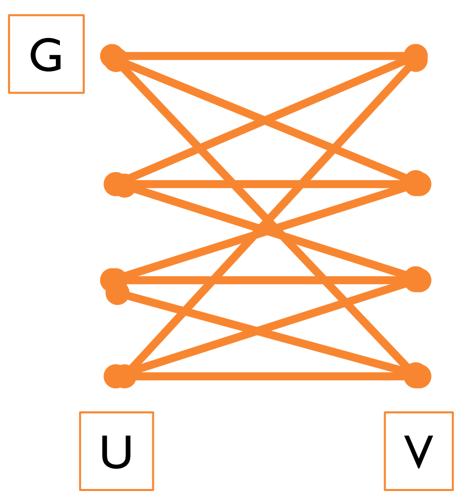
- We showed that G is an $\epsilon -$ approximation of H

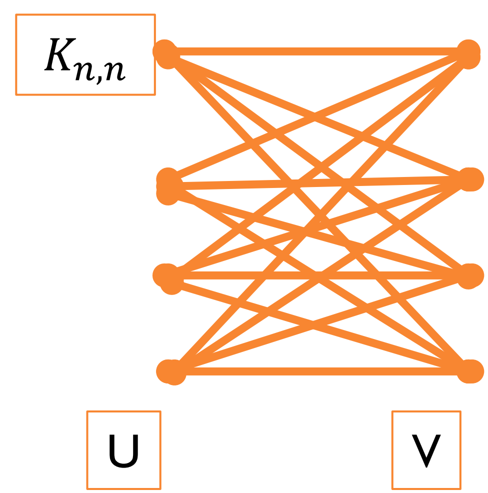
$$(1-\epsilon) x^T L_H x \preccurlyeq x^T L_G x \preccurlyeq (1+\epsilon) x^T L_H x$$

Bipartite Expander Graphs

- Define them as d-regular good approximations of (some multiple of) the complete bipartite graph H= $\frac{d}{n}K_{n,n}$:

$$(1-\epsilon)x^T L_H x \preccurlyeq x^T L_G x \preccurlyeq (1+\epsilon)x^T L_H x$$
$$\Rightarrow (1-\epsilon)H \preccurlyeq G \preccurlyeq (1+\epsilon)H$$





The Spectrum of Bipartite Expanders

- Eigenvalues of Laplacian of H= $\frac{d}{n} K_{n,n}$ are

$\lambda_1 = 0, \lambda_{2n} = 2d, \lambda_i = d, 0 < i < 2n$

- $(1 - \epsilon) x^T L_H x \preccurlyeq x^T L_G x \preccurlyeq (1 + \epsilon) x^T L_H x$

Says we want d-regular graph G that has evals

$\lambda_1 = 0, \lambda_{2n} = 2d,$

$\lambda_i \in ((1 - \epsilon)d, (1 + \epsilon)d), 0 < i < 2n$

Construction of Bipartite Expanders

- **Definition (Double Cover).** Let $G=(V,E)$ be a graph. The double-cover of G is the graph with vertex set $V\times\{0,1\}$ and edges $((u,0),(v,1))$ for (u,v) in E.

Cycle on 4 vertices and its double-cover

Construction of Bipartite Expanders

- **Proposition (Bipartite Expanders as Double Cover of Expanders).** Let G=(V,E) be a d-regular graph and H its double cover. For every eigenvalue λ_i of the Laplacian of G, H has a pair of eigenvalues λ_i, 2d-λ_i.

- We construct bipartite expander from double cover of d-regular expander

Expander Mixing Lemma, contd.

- **Theorem** Let G=($U \cup V$,E) a d-regular bipartite graph that ϵ-approximates the graph $\frac{d}{n}K_{n,n}$. Then, for every S⊆U, T⊆V

$$||E(S,T) - d\frac{|S||T|}{n}| \leq \epsilon d\sqrt{|S||T|}$$

Average Degree.

- **Theorem** Let $|S| = \sigma n, |T| = \tau n$. The average degree of vertices in G(SUT) is at most $\frac{2\mathrm{d}\sigma\tau}{\sigma+\tau} + \epsilon d$

Building Expander Codes

- Our construction of error-correcting codes will require two ingredients:
  - d-regular bipartite expander G on 2n vertices
  - A linear error-correcting code C_0 of length d (e.g. Hamming)

- We combine the two to construct code of length dn. Think of Co as small code that drives the construction (d small as n -> infinity)

Building Expander Codes

- Associate one bit with each edge of G.
- G has dn edges -> code of 2n bits $y_1, \ldots, y_{dn}$
- Describe the code by listing the linear constraints its codewords must satisfy: for each vertex, we require that the d bits corresponding to its attached d edges are a codeword in C_0.

Building Expander Codes

- If r_0 is the rate of C_0 then the space of codewords has dimension dr_0.
- Since Co is linear, its codewords satisfy some set of $d(1- r_0)$ linear equations.
- As there are 2n vertices in G, the constraints imposed by each vertex impose $2nd(1- r_0)$ linear constraints on the dn bits.
- Thus, the vector space of codewords that satisfies all of those constraints has dimension at least $dn- 2nd(1- r_0) = dn(2r_0-1)$
- Code we constructed has rate at least

$r= 2r_0-1$. Non-zero as long as $r_0>1/2$

Building Expander Codes

- Encoding? Parity Check matrix, Generator Matrix.
- Minimum Distance?
- Decoding?