Algebraic Methods in Computer Science	December 26, 2014
Lecture: 5	
Lecturer: *Amir Shpilka*	Scribe: *Orr Fischer*

Expander Graphs

We continue our discussion from last lecture and give an application of a family of (so called) "magical graphs" for constructing error correcting codes.

1 Constructing Error Correcting Codes

Definition 1.1. *A Linear Code $C \subseteq \{0,1\}^n$ is a code such that C is a linear subspace of $\mathbb{Z}_2^n$.*

We would like to construct families of linear codes that allow as high error as possible, while having a positive constant rate. To accomplish that We shall use a family of bipartite graphs that satisfy some special property: Assume we have a d-regular bipartite graph $G = (L \sqcup R, E)$, such that $|L| = n$ and $|R| = n/2$, that satisfy that for every subset $V \subseteq L$, if $|V| \leq \alpha n$ then $\Gamma(V) \geq 3d|V|/4$. I.e., every subset of L has many neighbors. Note that $|\Gamma(V)| \leq d|V|$ all such G since G is d-regular and thus we require that the size of the neighborhood of V is not that far from optimal.

Next we will see how to construct a good error correcting code from any such graph G.

Construction 1.1. *We associate each $v_i \in L$ with a variable x_i, and each $u \in R$ with a linear equation $\sum_{v_i \in \Gamma(u)} x_i = 0$. Our linear code C is the set of all vectors $x \in \mathbb{Z}_2^n$ that satisfy all of the equations determined by the vertices in R. It is clear that C is a linear code and that* $\dim(C) \geq |L| - |R| = n - \frac{n}{2} = \frac{n}{2}$.

Remark 1. *The $|R| \times |L|$ adjacency matrix of G, A_G, is also called the* parity check matrix *of C as $x \in C$ if and only if $A_G x = 0$.*

Intuitively, imagine that the entries of a word $x \in \{0,1\}^n$ label the n different vertices in L. Then $x \in C$ if and only if, each vertex in L sees an even number of 1's among its neighbors.

In the next section we prove that with this definition we get a family of good error correcting codes, and we will also give a decoding algorithm for codes in the family.

1.1 Decoding Algorithm

The intuition of the decoding algorithm agains comes from viewing the entries of a word $x \in \{0,1\}^n$ as labels for the vertices in L. The idea is that vertex $i \in L$ should flip its value if it thinks that it has a mistake. More accurately, vertex i flips its value if more than half of its neighbors in R are not "satisfied", that is, the linear equations they define are not satisfied. In this case notice that flipping the value of x_i will fix all those equations but will make the equations associated with its other neighbors unsatisfied. We repeat this process iteratively where at step ℓ every $i \in L$ makes such a decision (whether it should flip its value or not) and then all left vertices take an action simultaneously (i.e. they all follow their decision at the same time). We repeat this process until no right equation is unsatisfied. In other words, given a word $x \in \{0,1\}^n$ let $x_0 = x$. For $\ell = 1, 2, ...$ in the ℓ'th phase we have a current word $x_{\ell=1}$ and we flip the value of all x_i such that the number of unsatisfied linear equations associated with vertices in $\Gamma(i)$ is at least $\frac{d}{2}$. We denote the resulting word with x_ℓ. We describe this process in Algorithm 1.

Theorem 1.1. *If initially, the number of errors in x is at most $\frac{1}{2}(1+4\epsilon)\alpha n$ (where $\epsilon \leq \frac{1}{4}$), then the algorithm converges into the closest codeword to x after $O_{\alpha,\epsilon}(\log(n))$ iterations. That is, for $\ell = O_{\alpha,\epsilon}(\log(n))$, $x_\ell \in C$ and it is the closest word to x in C.*

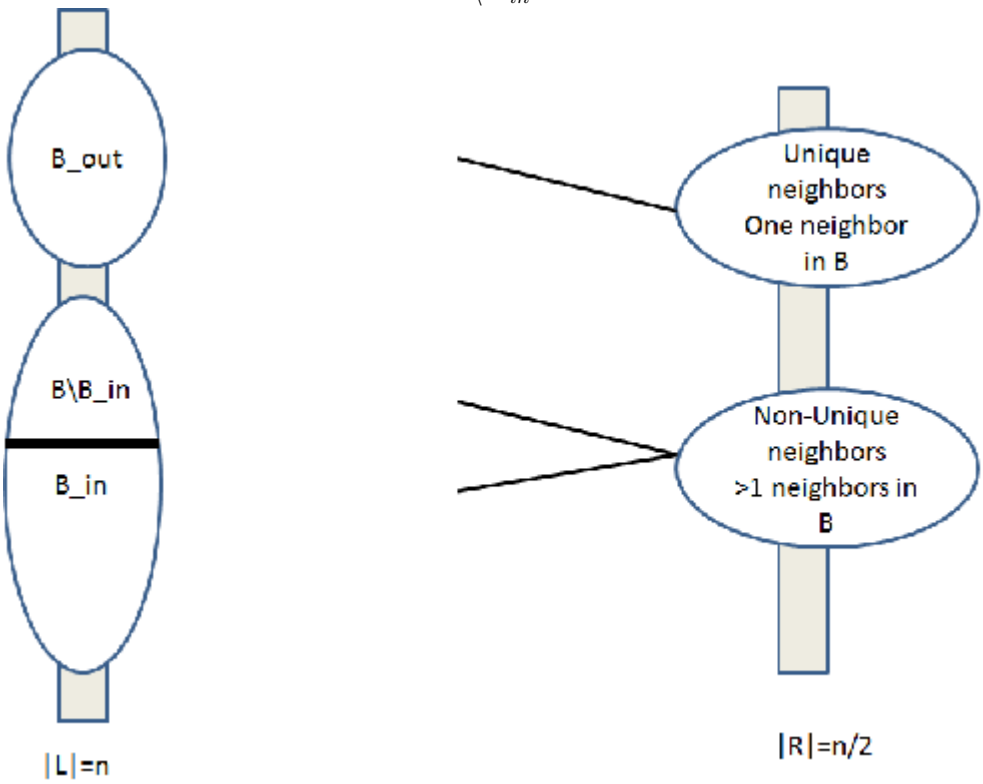
Algorithm 1: ErrorCorrect(y)

```
for ℓ = 1, 2, ... do
    if x_ℓ ∉ C then
        indexesToFlip = 0⃗
        foreach v_i ∈ L do
            if #equations not satisfied in Γ(v_i) ≥ d/2 then
                indexesToFlip[i] = 1
        x_{ℓ+1} = x_ℓ ⊕ indexesToFlip
    else
        return x_ℓ and stop
```

Proof. Given a current word y let $c \in C$ be the closest codeword to y (assuming one exists). We call a vertex v_i an error if $c_i \neq y_i$.

To prove that the decoding algorithm indeed converges to the closest codeword to x we shall prove that in each iteration the set of errors shrinks in size by a constant factor.

consider a given iteration ℓ. Denote $y = x_{\ell-1}$, the vector at the beginning of the iteration. Let $B = B_{\ell-1}$ be the set of errors of y. Our assumption guarantees that for $\ell = 1$, $|B_0| \leq \frac{1}{2}(1+4\epsilon)\alpha n$. Let $B' = B_\ell$ be the set of errors at the end of the iteration, i.e., of x_ℓ. We denote $B_{in} = B' \cap B$ and $B_{out} = B' \backslash B$. I.e., B_{in} are those errors that remained erroneous after the iteration whereas B_{out} are newly introduced errors. The set of errors that were corrected is therefore $B \setminus B_{in}$.



Lemma 1.1. $|B' \cup B| < \alpha n$

Proof. Assume for a contradiction that $|B' \cup B| \geq \alpha n$. Let $\tilde{B} \in B_{out}$ be such that $|\tilde{B} \cup B| = \alpha n$. By our

assumption on the graph (the expander property)

$$|\Gamma(B \cup \tilde{B})| \geq (\frac{3}{4} + \epsilon) \cdot d \cdot \alpha \cdot n. \tag{1}$$

On the other hand, since vertices in B_{out} were correct before the iteration, and "wrong" afterwards, each vertex there must have been flipped. Hence, every vertex in B_{out} (and therefore in $\tilde{B}$) has at least $\frac{d}{2}$ neighbors whose linear equations are unsatisfied. Since the only right vertices that have a chance of being unsatisfied must be neighbors of B we get that every vertex in $\tilde{B}$ has at least $\frac{d}{2}$ neighbors in $\Gamma(B)$. Thus, a crude upper bound on the number of neighbors of $(B \cup \tilde{B})$ is:

$$|\Gamma(B \cup \tilde{B})| = |\Gamma(B)| + |\Gamma(\tilde{B})| - |\Gamma(B) \cap \Gamma(\tilde{B})| \leq^{(*)} |\Gamma(B)| + \frac{d}{2}|\tilde{B}| \leq d \cdot |B| + \frac{d}{2} \cdot |\tilde{B}| = d \cdot |B| + \frac{d}{2}\alpha n,$$

where inequality $(*)$ holds as each vertex in $\tilde{B}$ has at most $d/2$ neighbors outside $\Gamma(B)$. Together with Equation (1) we get that

$$\begin{aligned}
&(\frac{3}{4} + \epsilon) \cdot d \cdot \alpha \cdot n \leq |\Gamma(B \cup \tilde{B})| \leq d \cdot |B| + \frac{d}{2}\alpha n \\
&(\frac{1}{4} + \epsilon) \cdot d \cdot \alpha \cdot n \leq d \cdot |B|,
\end{aligned}$$

yielding $|B| \geq \frac{1}{2}(1 + 4\epsilon)\alpha n$, in contradiction to our assumption on the number of errors. □

We continue the proof of the theorem. We would now like to bound $|\Gamma(B \cup B')|$. For this we consider *unique neighbors*. Call a vertex $u \in R$ a unique neighbor, if it has only a single neighbor in B. Notice that each linear equation associated with a unique neighbor $u \in R$ is unsatisfied, as it is only connected to one erroneous value. Hence, each vertex in B_{in} is connected to at most $\frac{d}{2}$ unique vertices. Indeed, since a vertex in B_{in} was not flipped it was connected to less than $d/2$ unsatisfied equations, and as each unique neighbor gives one unsatisfied equation it has less than $d/2$ such neighbors.

Next we consider non-unique vertices. We note that since they are not unique, each such vertex has at least two neighbors in B_{in}. We thus get

$$|\Gamma(B \cup B_{out}| \leq d \cdot |B \backslash B_{in}| + \underbrace{\frac{d}{2}|B_{in}|}_{\text{unique neighbours}} + \underbrace{\frac{1}{2} \cdot \frac{d}{2}|B_{in}|}_{\text{non-unique neigbours}} + \underbrace{\frac{d}{2}|B_{out}|}_{\text{same arg as in lemma}} =$$

$$= d(|B| - |B_{in}|) + \frac{3}{4} \cdot d|B_{in}| + \frac{d}{2}|B_{out}| = d|B| - \frac{1}{4}d|B_{in}| + \frac{d}{2}|B_{out}|.$$

From our assumption on the graph we have that

$$|\Gamma(B \cup B_{out}| \geq (\frac{3}{4} + \epsilon) \cdot d \cdot |B \cup B_{out}| \geq (\frac{3}{4} + \epsilon)d|B| + \frac{3}{4}d|B_{out}|.$$

It follows that

$$\begin{aligned}
&(\frac{3}{4} + \epsilon)d(|B| + |B_{out}|) \leq d|B| - \frac{1}{4}d|B_{in}| + \frac{d}{2}|B_{out}| \\
&\frac{1}{4}d(|B_{in}| + |B_{out}|) \leq d(\frac{1}{4} - \epsilon)|B| \\
&|B'| \leq (1 - 4\epsilon)|B|.
\end{aligned}$$

This means that the number of errors decreases by a constant factor at the ed of each iteration. This implies that the algorithm converges to the closest code word in $O(\log(n))$ iterations. □

2 Expanders

We shall now start to discuss some basic definitions of expander graphs. We shall restrict the discussion to d-regular graphs (the theory is similar to non-regular graphs but for simplicity we shall consider the regular case).

We shall use the following notation

$$E(A,B) \triangleq \{(u,v)|u \in A, v \in B\} \text{ and } e(A,B) \triangleq |E(A,B)|.$$

One way to define expander graphs, or more accurately, to say when does a family of graphs is a family of expanders is to consider the Cheeger constant that we define next.

Definition 2.1 (Cheeger's Constant)**.**

$$h = \min_{|S| \leq \frac{n}{2}} \frac{e(S,S^c)}{|S|} \leq d.$$

An (n,d,h) graph is a d-regular graph on n vertices, whose Cheeger constant is h.

Note that Cheeger's constant measures the average number of neighbors in S^c of the vertices in S, for the worst S. In other words, for every $|S| \leq n/2$, the number of edges leaving S is at least hS. We say that a graph is ϵ-expanding if $h \geq \epsilon d$. Thus, an ϵ fraction of the edges leaves every (not too large) set S.

The following theorem connected that combinatorial definition of expanding (via the Cheeger constant) to an algebraic one.

Theorem 2.1. *Let G be a d-regular graph on n vertices. Let $d \geq \lambda_2 \geq \ldots \geq \lambda_n$ be its n eigenvalues. Then,*

$$\frac{d-\lambda_2}{2} \leq h \leq \sqrt{2d(d-\lambda_2)}.$$

We will prove the first inequality in the HW and we will not give a proof of the second.

Remark 2. *These bounds are tight. When G is the n-dimensional Boolean cube its degree is n, Cheeger constant is 1 and its second eigenvalue, λ_2, is $\lambda_2 = n-2$. Thus, for Boolean cube $h = \frac{d-\lambda_2}{2}$. The other side of the inequality can be shown to be of the right order of magnitude when G is a cycle of length n. In this case the degree is 2, $h = 2/(n/2) = 4/n$ and it is not hard to prove that the second eigenvalue is $2\cos(2\pi/n) = 1 - \Theta(1/n^2)$. In this case, $\sqrt{2d(d-\lambda_2)} = O(1/n)$ which is the same order of magnitude as h (whereas $(d-\lambda_2)/2 = O(1/n^2) \approx h^2$).*

The quantity $d - \lambda_2$ is called the *spectral gap* and as explained above it is directly connected to the amount in which G expands.

3 Ramanujan graphs

Given that the spectral gap determines the expansion of G it is natural to ask how large can it be.

Theorem 3.1. $\lambda_2 \geq 2\sqrt{d-1}(1-o(1))$*, where the $o(1)$ term is (roughly) $O(1/\mathit{diameter}(G)^2)$.*

We will not prove this theorem but rather prove an easier statement. Denote $\lambda \triangleq \max(\lambda_2, |\lambda_n|)$. In words, λ is the second largest eigenvalue in absolute value.

Theorem 3.2. $\lambda \geq \sqrt{d}\sqrt{1 - \frac{d-1}{n-1}}$

Proof. As the trace of a diagonalizable matrix is the sum of its eigenvalues we get that for the adjacency matrix of G, A, it holds that $\sum \lambda_i^2 = \mathrm{tr}(A^2) = d \cdot n$. By definition of λ, $d \cdot n = \sum \lambda_i^2 \leq d^2 + (n-1)\lambda^2$. Hence, $\lambda^2 \geq d\frac{n-d}{n-1}$. □

Theorem 4.1 gives a bound on how small λ can be. A Ramanujan graph is a graph matching this bound.

Definition 3.1. *A graph is called a Ramanujan graph if* $\lambda \leq 2\sqrt{d-1}$.

We shall now describe a construction due to Lubotzky,Philips and Sarnak [LPS88].

Construction 3.1 (LPS graphs)**.** *Choose two primes* p *and* q *that are congruent to* 1 *modulo* 4*. We think of* p *as a small integer and* q *as relatively large. Let* $i \in \mathbb{F}_q$ *be such that* $i^2 = -1 \mod q$*. Such* i *exists as* $q \equiv 1 \mod 4$*. The vertices of our graph correspond to invertible* 2×2 *matrices, over* $\mathbb{F}_q$*, where we identify two matrices if they are constant non-zero multiple of each other. In other words,* $V(G) = PGL(2,q)$*. Our graph will be a Cayley graph with respect to the following set of generators:*

$$S = \left\{ \begin{pmatrix} a_0 + i \cdot a_1 & a_2 + i \cdot a_3 \\ -a_2 + i \cdot a_3 & a_0 - i \cdot a_1 \end{pmatrix} \mid a_0 > 0 \text{ odd, } a_1, a_2, a_3 \text{ even and } \sum_{i=0}^{3} a_i^2 = p \right\}.$$

Fact 3.1. *A theorem of Jacobi implies that the number of such* 4*-tuples* (a_0, a_1, a_2, a_3) *is* $|S| = p + 1$*.*

It can be checked that S *is closed under inversion. We shall connect two vertices* $A, B \in V$ *by an edge if and only if there exists* $C \in S$ *such that* $AC = B$*. As* S *is closed under inversion, this defines an undirected graph* $Cay(PGL(2,q), S)$*.*

Theorem 3.3 (LPS [LPS88])**.** *The graph* $G_{p,q}$ *is obtained by taking the connected component of the identity matrix in* $Cay(PGL(2,q), S)$*. It can be shown that* $Cay(PGL(2,q), S)$ *is either connected or has exactly two equal connected components, depending on the quadratic residue symbol* $\left(\frac{q}{p}\right)$*. In both cases,the 2nd largest eigenvalue is bounded as required.*

4 Expander Mixing Lemma

We next proof a basic property of expander graphs known as the expander mixing lemma. Roughly, the lemma shows that the density of edges between any two subsets of vertices in an expander graphs is similar to what one would expect in a random d-regular graph.

Theorem 4.1. *Let* G *be a graph, For any two subsets* $A, B \subseteq V$*, it holds that*

$$\left| e(|A|, |B|) - \frac{d|A||B|}{n} \right| \leq \lambda \sqrt{|A||B|}.$$

Proof. Let G be a d-regular graph on n vertices. We note that $1_B^t A_G 1_A = e(A, B)$

Since A_G is diagonalizable, we can find a basis of orthonormal eigenvector. Let $v_1, v_2, ..., v_n$ be such an orthonormal basis with $A_g v_i = \lambda_i v_i$. We also have that $v_1 = (\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})$ and $\lambda_1 = d$. As $v1, v2, ..., v_n$ span $\mathbb{R}^n$ there exist constants $\alpha_1, \ldots, \alpha_n$ such that $1_A = \sum \alpha_i v_i$ and similarly we can write $1_B = \sum \beta_i v_i$. Because of orthonormality, we have that

$$\alpha_1 = \langle 1_A, v_1 \rangle = \frac{|A|}{\sqrt{n}} \quad \text{and} \quad \beta_1 = \langle 1_B, v_1 \rangle = \frac{|B|}{\sqrt{n}}.$$

We therefore get that

$$\begin{aligned} e(A,B) = 1_B^t A_G 1_A &= \langle 1_B, A_G 1_A \rangle \\ &= \left\langle \sum_{i=1}^{n} \beta_i v_i, \sum_{i=1}^{n} \lambda_i \alpha_i v_i \right\rangle \\ &\underbrace{=}_{\langle v_i, v_j \rangle = 0} \sum_{i=1}^{n} \alpha_i \beta_i \lambda_i \underbrace{\|v_i\|^2}_{=1} \\ &= \frac{|A|}{\sqrt{n}} \frac{|B|}{\sqrt{n}} \cdot d + \sum_{i=2}^{n} \alpha_i \beta_i \lambda_i. \end{aligned}$$

Hence, $e(A,B) - \frac{d}{n}(|A||B|) = \sum_{i=2}^{n} \alpha_i \beta_i \lambda_i$. Applying Cauchy-Schwartz inequality we get that

$$\left| e(A,B) - \frac{d}{n}|A||B| \right| = \left| \sum_{i=2}^{n} \alpha_i \beta_i \lambda_i \right| \leq \sum_{i=2}^{n} |\alpha_i \beta_i \lambda_i| \leq \lambda \sum_{i=2}^{n} |\alpha_i||\beta_i| \underbrace{\leq}_{\text{Couchi Swartz}} \lambda \sqrt{\sum_{i=2}^{n} \alpha_i^2} \sqrt{\sum_{i=2}^{n} \beta_i^2}.$$

As $\sum_{i=1}^{n} \alpha_i^2 = \langle 1_A, 1_A \rangle = |A|$ and similarly $\sum_{i=1}^{n} \beta_i^2 = \langle 1_B, 1_B \rangle = |B|$ it follows that

$$\left| e(A,B) - \frac{d}{n}|A||B| \right| \leq \lambda \sqrt{\sum_{i=2}^{n} \alpha_i^2} \sqrt{\sum_{i=2}^{n} \beta_i^2} < \lambda \sqrt{|A||B|}.$$

□

5 Random graphs are Expanders

In this section, we will show not only there exists a d-regular bipartite expander (i.e. so called magical graph), but that most such graphs are expanders.

We start by describing a distribution on d-regular bipartite graphs. We have two sets of vertices R and L, each of size n.[1] To each vertex we connect d "half-edges". I.e., we connect to it d wires that are not connected to another vertex on the other side. Now, we choose a random permutation π on $[nd]$, and connect the open ends $(i, \pi(i))$. In words, we connect the loose ends of the edges connect to vertices in L to those of edges connected to vertices in R and we do so using a random map. Note that in the process we may get double edges, but we allow this in our graph.

Theorem 5.1. *Let $d \geq 3$. For all $\delta > 0$ there exists an $\epsilon > 0$, such that for a random d-regular bipartite graph with $|L| = |R| = n$ (sampled according to the distribution described above) with high probability it holds that for every $V \subseteq L$, if $|V| \leq \epsilon n$ then $|\Gamma(V)| \geq (d - 1 - \delta)|V|$.*

Remark 3. *We can define a similar distribution on d-regular n-vertex graphs that are not necessarily bipartite. For such graphs we will get that $|\Gamma(V)| \geq (d - 2 - \delta)|V|$ with essentially the same proof.*

For the proof we will need the following useful facts regarding binomial coefficients.

Fact 5.1. *1. $\binom{n}{k} \geq (\frac{n}{k})^k$.*

2. if $m > n$ then $\frac{\binom{n}{k}}{\binom{m}{k}} \leq (\frac{n}{m})^k$.

3. $\binom{n}{k} \leq (\frac{en}{k})^k$.

Proof. The proof of the first two items is immediate. The third item follows as $e^k \geq (1 + \frac{k}{n})^n \geq (\frac{k}{n})^k \binom{n}{k}$. □

We now prove Theorem 2.1. In the proof we shall ignore ceiling and floor notation for ease of notation.

Proof. Define a bad a event as a set $V \subseteq L$ such that $|V| \leq \epsilon n$, and $|\Gamma(V) < (d - 1 - \delta)|V|$. We would like to show that the probability of a bad event is small. For this we define the following indicator random variables: For every $T \subset R$ let $X_{V,T}$ be the indicator of the event $\Gamma(V) \subset T$. That is, $X_{V,T} = 1$ if and only if $\Gamma(V) \subset T$. We will be interested in bounding $\sum_{|V| \leq \epsilon n, |T| = (d-1-\delta)|V|} \Pr(X_{V,T} = 1)$. We first note that:

$$\Pr(X_{V,T} = 1) = \frac{\binom{|T| \cdot d}{|V| \cdot d}}{\binom{n \cdot d}{|V| \cdot d}}.$$

[1] We can also consider graphs that have a different number of vertices in each side.

Thus,

$$\begin{aligned}
\sum_{|V|\leq \epsilon n, |T|=(d-1-\delta)|V|} \Pr(X_{V,T}=1) &= \sum_{s=1}^{\epsilon n} \underbrace{\binom{n}{s}}_{\text{choosing } V} \underbrace{\binom{n}{(d-1-\delta)s}}_{\text{choosing } T} \frac{\binom{(d-1-\delta)sd}{sd}}{\binom{nd}{sd}} \\
&\leq \sum_{s=1}^{\epsilon n} \left(\frac{en}{s}\right)^s \left(\frac{en}{(d-1-\delta)s}\right)^{(d-1-\delta)s} \left(\frac{(d-1-\delta)s}{n}\right)^{sd} \\
&\leq \sum_{s=1}^{\epsilon n} e^{(d-\delta)s}(d-1-\delta)^{(1+\delta)s} \left(\frac{s}{n}\right)^{\delta s} \\
&= \sum_{s=1}^{\epsilon n} \left(\underbrace{(e^{d-\delta}(d-1-\delta)^{1+\delta})}_{\text{a constant which we denote } C_{d,\delta}} \left(\frac{s}{n}\right)^{\delta} \right)^s \\
&= \sum_{s=1}^{\epsilon n} \left(C_{d,\delta} \left(\frac{s}{n}\right)^{\delta} \right)^s =^{(*)} o(1),
\end{aligned}$$

where $(*)$ follows by first bounding its value for small values of s (say $s < \sqrt{n}$) and then for large values of s. Note that ϵ has to satisfy $C_{d,\delta} \cdot \epsilon^{\delta} < 1$ for the sum to converge (this corresponds to the region where $s \approx \epsilon n$). Thus, the probability that some bad event happened tends to 0 as n grows which concludes the proof of the theorem. □

The construction of Capalbo et al. [CRVW02] achieves expansion factor of $(1-\epsilon)d$ for a constant degree d.

Theorem 5.2 (Loseless exapnders)**.** *There exists a family of bipartite graphs as follows. $|L| = n$ and $|R| = m = n/t$. The left degree is $d = \text{poly}(\log t, 1/\epsilon)$. Sets of size $O(\epsilon n/td) = O(\epsilon m/d)$ expand by a factor $(1-\epsilon)d$.*

Notes

We used the survey of Hoory, Linial and Wigderson [HLW06] for this lecture.

References

[CRVW02] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, *Randomness conductors and constant-degree expansion beyond the degree / 2 barrier*, Proceedings of the 34th STOC, 2002, pp. 659–668.

[HLW06] S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, Bulleting of the American Mathematical Society **43** (2006), 439–561.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak, *Ramanujan graphs*, Combinatorica **8** (1988), no. 3, 261–277.