Introduction to Coding Theory **IST-UL, 2025/2026**

Week 11: Codes from graphs

Lecturer: João Ribeiro

Introduction

In previous lectures we have seen how to construct asymptotically good families of binary linear codes with efficient encoding and decoding procedures via code concatenation. Although the decoding procedure of these concatenated codes runs in polynomial time, it is not super fast. Ideally, we would like to have decoding algorithms that lend themselves to extremely efficient implementations. In these notes we study an alternative combinatorial approach towards constructing binary linear codes. More precisely, we will use graphs with certain "expansion" properties to obtain binary linear codes with decent dimension and distance. An advantage of these codes is that they have an extremely simple decoding algorithm that is fast *and highly parallelizable*.

Recommended reading: Essential Coding Theory, Chapter 11, these lecture notes by Anup Rao (lec5, lec6), and these lecture notes from John Wright's course.

1 Graphs and linear codes

Every binary linear code hides a bipartite graph. A bipartite graph is a graph G whose vertex set can be split into a left vertex set L and a right vertex set R such that all edges in the edge set E of G have an endpoint in L and an endpoint in R. We write $G = (L, R, E)$, and can represent G by a binary $|R| \times |L|$ "adjacency matrix" A_G such that $(A_G)_{i,j} = 1$ if and only if the left vertex i and the right vertex j are adjacent in G.

Let $\mathcal{C}$ be an $[n, k, d]_2$-code and let $H \in \mathbb{F}_2^{h \times n}$ be the parity-check matrix of $\mathcal{C}$. Then, we may see $\mathcal{C}$ as the bipartite graph G whose adjacency matrix is $A_G = H$. Sometimes $G_{\mathcal{C}}$ is called the *factor graph* of $\mathcal{C}$. Of course, this correspondence goes both ways. Given a bipartite graph G we can obtain a linear code $\mathcal{C}$ by taking its parity-check matrix to be $H = A_G$. In other words, if v is a right vertex of G and its neighborhood (i.e., the set of left vertices adjacent to v) is $\{i_1, \ldots, i_\ell\}$, then codewords c of the corresponding code satisfy the linear constraint

$$\sum_{j=1}^{\ell} c_{i_j} = 0 \pmod 2.$$

2 Expander graphs

In these notes we will leverage the correspondence between binary linear codes and bipartite graphs by considering codes induced by a special class of graphs, called *expander graphs*.

There is more than one notion of graph expansion. We start by defining *vertex expanders*. For a vertex v in G, we denote its neighborhood by $N(v)$. More generally, for a set of vertices S in G, we denote its neighborhood (the set of vertices adjacent to some vertex in S) by $N(S)$. We begin by giving an informal description of vertex expansion.

Suppose that the bipartite graph $G = (L, R, E)$ has the property that every left vertex has degree D. We call such graphs *D-left regular*. Then, for every subset $S \subseteq L$, it is clear that

$$|N(S)| \leq D \cdot |S|.$$

Informally, G is a good bipartite vertex expander if for every not-too-large subset $S \subseteq L$ the size of its neighborhood $N(S)$ is not much smaller than the maximum, $D \cdot |S|$. More precisely, we have the following definition.

Definition 1 (Bipartite vertex expander) *We say that a bipartite graph $G = (L, R, E)$ is an* $(n, h, D, \gamma, \alpha)$-vertex expander *if G is D-left regular with $|L| = n$, $|R| = h$, and every subset $S \subseteq L$ of size $|S| \leq \gamma n$ satisfies*

$$|N(S)| \geq \alpha |S|.$$

We begin by discussing some basic properties of vertex expanders. First, note that if $|R| > |L|$, then it is easy to construct a bipartite vertex expander with great parameters. The challenge, then, is to construct good vertex expanders with $|R| \ll |L|$ and small left degree D (so, these graphs are fairly "sparse"). This is also the setting of interest for the connection to binary linear codes, since $|R|$ controls the number of linear constraints, and hence the codimension of the resulting binary linear code (we want codes with "low" codimension $h = n - k$).

We can also obtain some basic bounds on γ and α. Since $|N(S)| \leq D \cdot |S|$ for all S, it follows that $\alpha \leq D$. Furthermore, since $|N(S)| \leq |R|$, we must have $\alpha\gamma|L| \leq |R|$, and so $\gamma \leq \frac{|R|}{\alpha|L|} = \frac{h}{\alpha n}$.

3 Expander codes

An *expander code* is simply a code constructed by taking its parity-check matrix H to be the adjacency matrix of an appropriate expander graph. A useful property of these codes is that if the graph is sparse (in the sense that there are few edges), then the resulting parity-check matrix is also a sparse matrix. These codes were first introduced by Sipser and Spielman [SS96].

We now analyze the properties of expander codes obtained from sufficiently strong vertex expanders.

3.1 Linearity, dimension, and distance

Let $G = (L, R, E)$ be an $(n, h, D, \gamma, \alpha)$-vertex expander. It is clear from construction that the code $\mathcal{C}$ we obtain by setting its parity-check matrix $H = A_G$ is a binary linear code of block length n. The following result states bounds on the dimension and distance of $\mathcal{C}$ in terms of the parameters of G.

Theorem 1 *The code $\mathcal{C}$ is a linear code of block length n and dimension at least $n-h$. Furthermore, if $\alpha > D/2$, then $\mathcal{C}$ has minimum distance at least $1 + \gamma n$.*

Proof: The claim about the dimension is easy to see, since $\mathcal{C} = \{c : Hc = 0\}$ and H has h rows.

We now analyze the minimum distance of $\mathcal{C}$. Since $\mathcal{C}$ is linear, it suffices to show that every nonzero codeword in c has Hamming weight at least $1 + \gamma n$ when $\alpha > D/2$. For the sake of a contradiction, supppose that there is a nonzero codeword $c \in \mathcal{C}$ such that $w_H(c) \leq \gamma n$. Let $S \subseteq L$ be the subset of left vertices corresponding to the support of c, i.e., the coordinates j such that $c_j = 1$. Then, since G is an $(n, h, D, \gamma, \alpha)$-vertex expander, we know that

$$|N(S)| \geq \alpha |S| > \frac{1}{2} D|S|.$$

Now, we will show that this implies that S actually has at least one *unique neighbor*. We say that a vertex $i \in R$ is a unique neighbor of S if i is adjacent to exactly one vertex in S. The reason why this notion is useful is that the existence of a unique neighbor i for S is equivalent to the existence of a row H_i such that $H_i \cdot c = 1$. In turn, this implies that $Hc \neq 0$, and so $c \notin \mathcal{C}$.

So, all that remains is to show that there exists a unique neighbor of S. We prove the following more general claim that will also be useful later on.

Claim 1 *Let $U(S)$ denote the set of unique neighbors of S. Then,*

$$|U(S)| \geq \left(\frac{2\alpha}{D} - 1 \right) D|S|.$$

Proof: To lower bound $U(S)$ we count the $D \cdot |S|$ edges emanating from S from the perspective of the right vertex set. Note that every $i \in U(S)$ contributes one edge to $D \cdot |S|$, while every $i \in N(S) \setminus U(S)$ contributes at least two edges. Therefore,

$$D \cdot |S| \geq |U(S)| + 2(|N(S)| - |U(S)|) \geq |U(S)| + 2\left(\alpha |S| - |U(S)|\right).$$

We can equivalently write this as

$$|U(S)| \geq \left(\frac{2\alpha}{D} - 1 \right) |S|.$$

■

In particular, Claim 1 implies that if $\alpha > D/2$, then $|U(S)| > 0$, and so there exists a unique neighbor of S. This concludes the proof. ■

3.2 Decoding

A great feature of expander codes is that they have a very simple decoding algorithm that can be implemented in a highly efficient manner in practice. Suppose that y is obtained by introducing at most $\gamma n/2$ errors in a codeword $c \in \mathcal{C}$. Then, the decoding algorithm proceeds as follows on input y:

> While there is a left vertex j such that the majority of its neighbors (i.e., parity checks) are not satisfied, flip y_j.

The following theorem states that this decoding algorithm succeeds whenever the underlying graph has strong vertex expansion.

Theorem 2 *If $\alpha > 3D/4$, then this algorithm returns c.*

Proof: The algorithm receives as input $y = c + e$ with $e \in \{0,1\}^n$ satisfying $w_H(e) \leq \gamma n/2$. We proceed by iteratively flipping bits of y – for the sake of simplicity, we always refer to the (updated) string by y. Note that flipping bits in y may actually *increase* the number of errors in y. We will show that:

1. While there are at most γn errors in y, the algorithm finds a coordinate of y to flip. Note that this is satisfied in the initial iteration of the algorithm. Furthermore, in every iteration the overall number of unsatisfied parity checks strictly decreases.
2. The number of errors in y never goes above γn. Therefore, the algorithm always finds a coordinate to flip (if there are still errors).

Combining the above, we conclude that the algorithm will eventually terminate and return a codeword of $\mathcal{C}$. Furthermore, the returned codeword must be c, since $\mathcal{C}$ has minimum distance at least $1 + \gamma n$ and the number of errors in y never goes above γn.

First, we argue that while there are at most γn errors in y there is always a left vertex such that the majority of its neighboring parity checks are not satisfied. Let S denote the locations of the errors in (the current version of) y. Note that S has at most $D|S|$ neighbors, and that $|S| \leq \gamma n$ by hypothesis. By Claim 1 with $\alpha > 3D/4$, it follows that $|U(S)| > \frac{1}{2}D|S|$. Therefore, more than half of the neighbors of S are actually unique neighbors of S, and so, by averaging, there must exist a vertex in S that is adjacent to more than $\frac{D}{2}$ unique neighbors of S. This vertex is a valid candidate for being flipped.

Second, we argue that the number of errors in y never grows above γn. At the beginning of the execution of the algorithm the number of errors is at most $\gamma n/2$ by hypothesis. Since G is D-left regular, the number of parity checks unsatisfied by y is thus at most $\gamma n/2 \cdot D$. By the previous paragraph there is a candidate coordinate of y to be flipped, and flipping any such candidate strictly decreases the total number of unsatisfied parity checks. If after some iterations y has at least γn errors, then the set S of errors has more than $\gamma n/2 \cdot D$ unique neighbors by Claim 1, and hence there are more than $\gamma n/2 \cdot D$ unsatisfied parity checks, a contradiction. ■

4 Existence of great vertex expanders

The codes we analyzed above are based on vertex expanders with fairly strong parameters. In order for these codes to be efficiently encodable and decodable, we would need to have efficient constructions of such graphs. Worse than that, it is not even clear at first sight whether such graphs even *exist*!

It turns out that we do have explicit constructions of sufficiently strong vertex expanders [CRVW02]. However, these are quite complicated and outside the scope of our course. Instead, here we focus on showing the existence of vertex expanders with sufficiently strong parameters for our previous analysis.

For the sake of simplicity we focus here on the setting where $\alpha = \frac{4}{5}D > \frac{3}{4}D$. However, the techniques we present generalize easily.

Theorem 3 *Fix arbitrary $\gamma \in (0,1)$. Then, for all sufficiently large n there exist $(n, h, D, \gamma, \alpha = \frac{4D}{5})$-vertex expanders with $D = O(\log(1/\gamma))$ and $h = O(\gamma \log(1/\gamma)n)$.*

Before we prove this theorem we make some observations.

Code parameters. Recall that our goal was to construct asymptotically good codes with fast decoding. Therefore, we aim for a small constant γ, since this parameter controls the minimum distance of the code. In this case, we get a right vertex set of size $h = (\gamma \log(1/\gamma)n)$, and so, by Theorem 1, the rate of the code is at least

$$1 - h/n = 1 - O(\gamma \log(1/\gamma)) > 0,$$

provided that γ is small enough. Note also that $\gamma \log(1/\gamma) \approx h(\gamma)$ when γ is small, and so the rate we achieve is not that far off the asymptotic Hamming bound.

Complexity of encoding/decoding. Assuming we are given the construction of our graph for free, then the decoding algorithm from Theorem 2 can be made to run in time[1] $O(D \cdot D_R \cdot n)$, where D_R is the maximum degree of a right vertex. From Theorem 3 we get that D is constant (for fixed γ), and it is also possible to show that D_R is constant too. Therefore, decoding runs in time $O(n)$.

The decoding algorithm we described above is inherently sequential. However, there is an alternative decoding algorithm that is friendly to parallelization. Namely, in each iteration we find all left vertices whose majority of neighbors are unsatisfied parity checks, and flip them all simultaneously. As already shown in the original work of Sipser and Spielman, this procedure converges to the correct codeword in $O(\log n)$ rounds.

Since the code is linear, encoding can be performed naively in time $O(n^2)$.

[1] This claim assumes we are working in a specific computational model, called a "RAM in the uniform cost model". The details are not too important.

Proof:[Theorem 3] We use the probabilistic method. Consider sampling a D-left regular bipartite graph G with n left vertices and h right vertices as follows: for each left vertex v, choose $N(v)$ as a uniformly random size-D subset of the right vertex set R. With some hindsight, we take

$$h = 5e^2 D\gamma n.$$

Note that G fails to be an $(n, h, D, \gamma, \alpha)$-vertex expander only if there exists a set S of left vertices of size $|S| = \ell \leq \gamma n$ such that the $D|S|$ edges coming out of S have many collisions. More precisely, if at least $D\ell/5$ of these edges point to previously selected right vertices. By a union bound, the probability that this happens for some fixed set S of size ℓ is at most

$$\begin{aligned}
\binom{n}{\ell}\binom{D\ell}{D\ell/5}\left(\frac{D\ell}{h}\right)^{D\ell/5} &\leq (en/\ell)^{\ell} \cdot (5e)^{D\ell/5} \cdot \left(\frac{D\ell}{h}\right)^{D\ell/5} \\
&= \left(\frac{en \cdot (5e)^{D/5} \cdot (D\ell)^{D/5}}{\ell h^{D/5}}\right)^{\ell} \\
&= \left(\frac{en/\ell}{(e\gamma n/\ell)^{D/5}}\right)^{\ell} \\
&= \left(\frac{1}{\gamma^{D/5}(en/\ell)^{D/5-1}}\right)^{\ell}.
\end{aligned}$$

Take $D = 5\lceil 1 + \ln(4/\gamma)\rceil = \Theta(\log(1/\gamma))$. Then, recalling that $\ell \leq \gamma n$, we get

$$\left(\frac{1}{\gamma^{D/5}(en/\ell)^{D/5-1}}\right)^{\ell} \leq \left(\frac{1}{\gamma^{D/5}(e/\gamma)^{D/5-1}}\right)^{\ell} = \left(\frac{1}{\gamma e^{D/5-1}}\right)^{\ell} \leq (1/4)^{\ell}.$$

Therefore, the probability that G fails to be a vertex expander with the desired parameters is at most

$$\sum_{\ell=1}^{\infty}(1/4)^{\ell} = 1/2 < 1.$$

■

5 Tanner codes

We will now see another way of constructing asymptotically good binary linear codes with fast decoding from graphs that only requires weaker expansion properties of the underlying graph – making this graph easier to construct explicitly. We will analyze the parameters of these codes and the weaker notion of expansion required to construct them (*spectral expansion*), but will not discuss decoding algorithms.

Given a D-regular bipartite graph $G = (L, R, E)$ with $|L| = |R| = N$ and a *base code* $\mathcal{C}_0 \subseteq \mathbb{F}_2^D$ of dimension k_0 and minimum distance d_0, define the *Tanner code* $T(G, \mathcal{C}_0)$ as[2]

$$T(G, \mathcal{C}_0) = \{c \in \mathbb{F}_2^{|E|} : c_{N(u)} \in \mathcal{C}_0 \forall u \in L \cup R\},$$

[2]We implicitly assume a pre-specified ordering of the vertices of L and R.

where c_S denotes the restriction of the vector c to the coordinates in S. In words, the Tanner code $T(G, \mathcal{C}_0)$ corresponds to assignments of values to the edges of G that yield codewords of $\mathcal{C}_0$ when restricted to the neighborhood of any vertex of G.

Linearity and dimension. It is easy to see that $T(G, \mathcal{C}_0)$ is linear whenever $\mathcal{C}_0$ is linear. Furthermore, in this case, the dimension of $T(G, \mathcal{C}_0)$ is at least

$$|E| - 2N(D - k_0),$$

since there are $2N$ vertices in G and each vertex contributes at most $D - k_0$ linear constraints (those defining $\mathcal{C}_0$ on the D edges of the neighborhood of that particular vertex). Therefore, if $\mathcal{C}_0$ has rate R_0, then $T(G, \mathcal{C}_0)$ has rate at least

$$1 - \frac{2N}{|E|}(D - k_0) = 1 - \frac{2}{D}(D - k_0) = 2R_0 - 1,$$

where we used the fact that $|E| = ND$. In particular, $T(G, \mathcal{C}_0)$ has positive rate whenever $\mathcal{C}_0$ has rate $R_0 > 1/2$.

Minimum distance. Let's try to first intuitively understand which properties we need from G to get a good guarantee on the minimum distance of $T(G, \mathcal{C}_0)$ given a linear code $\mathcal{C}_0$ with positive relative distance δ_0.

In this case $T(G, \mathcal{C}_0)$ is linear, so it suffices to lower bound the Hamming weight of an arbitrary nonzero codeword c. Since c is nonzero, it assigns value 1 to some edge $(u, v) \in G$. In turn, this means that $c_{N(u)}$ is nonzero. Since $c_{N(u)} \in \mathcal{C}_0$, we conclude that there are at least d_0 edges incident to u that are set to 1. Each of these edges is incident to some right vertex v_i. Since $c_{N(v_i)} \in \mathcal{C}_0$, this implies that at least d_0 edges incident to v_i are also set to 1. So, there seems to be a sort of "amplification process" going on here, *provided that the edges coming out of the v_i's do not all land in some small subset of left vertices (and so on).* A bit more precisely, this hints that we need to avoid the existence of small subsets $S \subseteq L$ and $T \subseteq R$ that only share edges with each other.

Let's look at what we can expect from a random graph G. For a left subset S and a right subset T, denote by $E(S, T)$ the set of edges (u, v) of G with $u \in S$ and $v \in T$. Then, over a random sampling of G, we have

$$\mathrm{E}[|E(S, T)|] = D|S| \cdot \frac{|T|}{N}.$$

We will now show formally that if G satisfies $|E(S, T)| \approx D|S| \cdot \frac{|T|}{N}$ for all subsets S and T, then we can get a good guarantee on the minimum distance of $T(G, \mathcal{C}_0)$ based on the minimum distance of $\mathcal{C}_0$.

Definition 2 (Pseudorandom graph) *We say that a D-regular bipartite graph $G = (L, R, E)$ with $|L| = |R| = N$ is* ε-pseudorandom *if for every $S \subseteq L$ and $R \subseteq R$ we have*

$$\left| |E(S, T)| - \frac{D|S| \cdot |T|}{N} \right| \leq \varepsilon D \sqrt{|S| \cdot |T|}.$$

Theorem 4 *Suppose that G is ε-pseudorandom. Then, the relative distance of $T(G, \mathcal{C}_0)$ is at least*

$$\delta_0(\delta_0 - \varepsilon).$$

In particular, this means that if G is ε-pseudorandom for $\varepsilon < \delta_0$, then we obtain positive relative distance too.

Proof:[Theorem 4] Fix a nonzero codeword $c \in T(G, \mathcal{C}_0)$. Define $S = \{u \in L : c_{N(u)} \neq 0\}$ and $T = \{v \in R : c_{N(v)} \neq 0\}$. Note that

$$|E(S,T)| \geq w_H(c) \geq \max\left(|S| \cdot d_0, |T| \cdot d_0\right) \geq \sqrt{|S| \cdot |T|} \cdot d_0.$$

We now focus on lower bounding $\sqrt{|S| \cdot |T|}$. By the chain of inequalities above and the ε-pseudorandomness of G, we get that

$$\sqrt{|S| \cdot |T|} \cdot d_0 \leq \frac{D|S| \cdot |T|}{N} + \varepsilon D \sqrt{|S| \cdot |T|}.$$

This is equivalent to

$$\sqrt{|S| \cdot |T|} \geq (\delta_0 - \varepsilon)N.$$

Combining this with the above, we get that

$$w_H(c) \geq (\delta_0 - \varepsilon) d_0 N = (\delta_0 - \varepsilon)\delta_0 N D.$$

■

5.1 Spectral expanders and pseudorandomness

We have seen that pseudorandom graphs suffice to get asymptotically good Tanner codes. We now show how to construct pseudorandom graphs via *spectral expanders*.

Let G be a D-regular graph with N vertices, and let A be its adjacency matrix. Note that A is an $N \times N$ symmetric matrix. Therefore, A has N real eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, and we can write

$$A = \sum_{i=1}^{N} \lambda_i u_i u_i^\top$$

with u_i an eigenvector with eigenvalue λ_i. Note that $-D \leq \lambda_i \leq D$, since each row of A has exactly D ones. Furthermore, the unit vector $(1/\sqrt{N}, \ldots, 1/\sqrt{N})$ is an eigenvector of A with eigenvalue D. This means that D-regular graphs always have $\lambda_1 = D$. Informally, G is a spectral expander if its second largest eigenvalue (in absolute value) is small. More formally, we have the following definition.

Definition 3 (Spectral expander) *We say that G is an* (N, D, λ)-spectral expander *if G is a D-regular graph on N vertices and $\lambda = \max(\lambda_2, |\lambda_N|)$.*

Spectral expanders are very important objects. They are also weaker than the vertex expanders we required for the previous construction. In fact, it is possible to show that, generically, spectral expanders can only yield vertex expanders with $\alpha \leq 1/2$, while our analysis above required $\alpha > 1/2$.

We now show that a natural way of obtaining a bipartite graph from a spectral expander yields a pseudorandom graph. More precisely, given a graph $G = (V, E)$ on N vertices, we define its *double cover* as the bipartite graph $H = (L, R, E')$ with $L = R = V$ and such that if $(u, v) \in E$, then $(u, v), (v, u) \in E'$.

Theorem 5 (Expander mixing lemma) *If $G = (V, E)$ is an (N, D, λ)-spectral expander, then its double cover $H = (L, R, E')$ is (λ/D)-pseudorandom.*

Proof: Fix arbitrary subsets $S \subseteq L$ and $R \subseteq R$, and let $E(S, T)$ denote the number of edges between S and T in H. For a set U, write 1_U for the binary vector satisfying $(1_U)_i = 1$ if and only if $i \in U$.

Recalling that $A = \sum_{i=1}^{N} \lambda_i u_i u_i^\top$, we have

$$|E(S,T)| = 1_S^\top A 1_T = \sum_{i=1}^{N} \lambda_i \langle 1_S, u_i\rangle\langle u_i, 1_T\rangle = \lambda_1 \langle 1_S, u_1\rangle\langle u_1, 1_T\rangle + \sum_{i=2}^{N} \lambda_i \langle 1_S, u_i\rangle\langle u_i, 1_T\rangle. \quad (1)$$

Note that

$$\langle 1_S, u_1\rangle = \langle 1_S, (1/\sqrt{N}, \ldots, 1/\sqrt{N})\rangle = \frac{|S|}{\sqrt{N}}.$$

Likewise, $\langle u_1, 1_T\rangle = \frac{|T|}{\sqrt{N}}$. Combining this with Equation (1) and recalling that $\lambda_1 = D$ yields

$$|E(S,T)| = \frac{D|S| \cdot |T|}{N} + \sum_{i=2}^{N} \lambda_i \langle 1_S, u_i\rangle\langle u_i, 1_T\rangle. \quad (2)$$

Furthermore, we can write 1_S and 1_T according to the orthonormal basis of eigenvectors of A as

$$1_S = \sum_{i=1}^{N} \alpha_i u_i$$

and

$$1_T = \sum_{i=1}^{N} \beta_i u_i$$

for some α_i's and β_i's. Using this notation, we get

$$\sum_{i=2}^{N} \lambda_i \langle 1_S, u_i\rangle\langle u_i, 1_T\rangle = \sum_{i=2}^{N} \lambda_i \alpha_i \beta_i$$

Therefore,

$$\begin{aligned}\left||E(S,T)| - \frac{D|S|\cdot|T|}{N}\right| &= \left|\sum_{i=2}^{N}\lambda_i\alpha_i\beta_i\right| \\ &\le \lambda\left|\sum_{i=2}^{N}\alpha_i\beta_i\right| \\ &\le \lambda\sqrt{\sum_{i=2}^{N}\alpha_i^2}\cdot\sqrt{\sum_{i=2}^{N}\beta_i^2} \\ &\le \lambda\|1_S\|_2\cdot\|1_T\| \\ &= \lambda\sqrt{|S|\cdot|T|}.\end{aligned}$$

The first inequality uses the fact that G is an (N, D, λ)-spectral expander. The second inequality uses Cauchy-Schwarz.

It follows that H is $(\varepsilon = \lambda/D)$-pseudorandom. ∎

Explicit spectral expanders. We know strongly explicit constructions of spectral expanders with constant degree D and $\lambda \le 2\sqrt{D-1}$, which is essentially optimal (graphs with this property are called *Ramanujan*). This goes back to works of Lubotzky-Phillips-Sarnak and Margulis in the late 1980s, but is still an active research area.

6 Further reading

If you are interested in learning more about expander graphs and their applications, see the excellente survey of Hoory, Linial, and Wigderson [HLW06]. For more on spectral graph theory and its applications, see these great lecture notes by Luca Trevisan.

References

[CRVW02] Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, pages 659–668. Association for Computing Machinery, 2002.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.