An Invitation To Pseuodrandomness

Gil Cohen

Contents

I Pseudorandomness - a bird's eye view

1 Randomness in proofs and a glance at pseudorandom objects . . 7

1.1 Randomness in proofs: the probabilistic method . . . 7
1.1.1 Ramsey graphs . . . 8
1.1.2 The probabilistic method vs. lower bounds . . . 9
1.1.3 The probabilistic method as a benchmark . . . 10
1.1.4 Randomness in mathematics . . . 10

1.2 Error correcting codes . . . 10
1.2.1 A quick and dirty introduction to coding theory . . . 10
1.2.2 The Reed-Solomon code . . . 11
1.2.3 The Gilbert-Varshamov bound . . . 12
1.2.4 Reed-Muller codes . . . 12
1.2.5 Algebraic-Geometric codes . . . 13

1.3 Expander graphs . . . 14

1.4 Combining structure and randomness: tree codes as a case study . . . 18
1.4.1 Tree code existence proof via the Lovász Local Lemma . . . 19
1.4.2 Four-color tree codes . . . 20
1.4.3 Explicit tree codes . . . 23

2 Randomness in computation . . . 25

2.1 Randomness in algorithms . . . 25
2.1.1 Does randomness add computational power? . . . 25

2.2 Does randomness exist? . . . 25

2.3 Randomness in algorithms . . . 26
2.3.1 Does randomness add computational power? . . . 26

2.4 Does randomness exist? . . . 26

2.5 What is it like to be a pseudorandomnessist? . . . 26

II Basic Pseudorandom Primitives

3 Bounded independence 29

3.1 Constructing bounded independence distributions 29

3.1.1 Constructing bounded-independent distributions 30

3.1.2 Pairwise independent hash functions 30

3.1.3 A construction of a family of pairwise independent hash functions 30

3.1.4 k-independent hash functions 31

3.2 PRGs for decision trees 31

3.3 PRGs for combinatorial rectangles 33

3.4 Bounded independence fools conjunctions 36

3.5 Bounded independent fools product of low variance variables 38

3.6 Improved PRGs for combinatorial rectangles 38

3.7 Constructing Ramsey graphs 38

eudorandomness - a bird’s eye view

1 **Randomness in proofs and a glance at pseudorandom objects** 7

1.1 Randomness in proofs: the probabilistic method 7

1.2 Error correcting codes 10

1.3 Expander graphs 14

1.4 Combining structure and randomness: tree codes as a case study 18

2 **Randomness in computation** 25

2.1 Randomness in algorithms 25

2.2 Does randomness exist? 25

2.3 Randomness in algorithms 26

2.4 Does randomness exist? 26

2.5 What is it like to be a pseudorandomnessist? . 26

1. Randomness in proofs and a glance at pseudorandom objects

"Anyone who considers arithmetical methods of producing random digits is, of course, in the state of sin" [John von Neumann]

Before delving into the primary focus of this text, which is pseudorandomness, it is essential first to explore and understand the concept of randomness. Randomness, to say the least, is a fascinating concept that has captured the attention of philosophers and scientists for centuries. Although it's not entirely clear how randomness should be defined or whether it truly exists in reality—as opposed to merely being a theoretical concept—it is extensively utilized by physicists, statisticians, and, quite surprisingly, in mathematical proofs decades before computer science entered the picture.

In this introductory chapter, we will discuss the role of randomness in mathematical proofs, in computation and in coding theory. We will do so by considering some fundamental problems in these fields and use this opportunity to introduce key ideas, objects, and results from these respective fields.

Interestingly, historically, it was around the same time that randomness was used in mathematical proofs and in coding theory. Indeed, around the same time, both Erdős (1947) and Shannon (1948) employed what is now known as *the probabilistic method* in their proofs. Erdős explored Ramsey graphs, which have proven to be pivotal in the field of pseudorandomness and are highly useful in theoretical computer science. Meanwhile, Shannon used the probabilistic method to demonstrate the existence of asymptotically good error correcting codes. It goes without saying that coding theory is a vast area of research. Interestingly, it is heavily applied in the realm of pseudorandomness. Furthermore, ideas and results from pseudorandomness play a crucial role in the development of coding theories.

There is some historial context to these works. About a decade before, in 1933, Kolmogorov put probability theory on an axiomatic footing. There was also a growing influence of randomness in other fields such as in Physics in the preceding decades.

We start this chapter by considering the role of randomness in proofs, a technique which goes under the name the probabilistic method.

1.1 Randomness in proofs: the probabilistic method

We illustrate the probabilistic method with the following example: Consider a unit circle colored alternately in blue and red, where the total arc length of the blue segments is strictly greater than that of the red (for simplicity, let's assume that each connected segment of color has a nonzero length). Now, take a square centered at the origin, with its four vertices on the unit circle. Our objective is to demonstrate that we can rotate the square in such a way that at least three of its vertices touch the blue segments.

To do this, we consider a sample space where the angle $\theta \in [0, 2\pi)$ determines the square's rotation, chosen uniformly at random. Define four random variables $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4$, each corresponding to one of the square's vertices, where X_i indicates whether vertex i touches a blue segment. It is evident that for each i, $\mathbf{E}[\mathbf{X}_i] > \frac{1}{2}$. Hence, by the linearity of expectation, the expected number of vertices touching blue is given by

$$\mathbf{E}[\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 + \mathbf{X}_4] = \mathbf{E}[\mathbf{X}_1] + \mathbf{E}[\mathbf{X}_2] + \mathbf{E}[\mathbf{X}_3] + \mathbf{E}[\mathbf{X}_4] > 4 \cdot \frac{1}{2} = 2.$$

The crucial, albeit straightforward, insight is that there must be a specific angle θ in our sample space where the actual number of vertices touching blue meets or exceeds this expectation, specifically being at least 2. Since the number of vertices touching blue is an integer, rotating the square to this particular θ ensures that at least 3 vertices will touch blue.

This proof is elegant, conclusively establishing the proposition. The application of randomness in this context does not suggest a margin for error, which is often associated with probabilistic algorithms. However, the proof leaves a practical question unanswered: it doesn't suggest a method to find the specific angle θ for a given coloring. Notably, the proof bypasses the intricate structure of the coloring by leveraging the powerful yet simple principle of linearity of expectation. While $\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 + \mathbf{X}_4$ encapsulates all complex dependencies dictated by the circle's coloring, considering their expectation allows us to examine these variables independently.

1.1.1 Ramsey graphs

On a conceptual level, Ramsey theory explores the phenomenon wherein any sufficiently large system, regardless of its apparent chaos, inevitably exhibits some form of local structure. Although Ramsey's original paper (1930) focused on first-order logic, the following combinatorial definition bears his name.

Definition 1.1 An undirected graph $G = (V, E)$ with $|V| = n$ vertices is termed *k-Ramsey* if it contains neither a clique nor independent sets of size k.

Ramsey demonstrated that no graph with n vertices can be $\frac{1}{2}\log_2 n$ Ramsey. Complementing this, Erdős established that most graphs with n vertices are $3\log_2 n$ Ramsey. To understand this, consider the sample space which is uniform over all n-vertex graphs. In this distribution, a graph is generated by independently deciding whether to connect each pair of distinct vertices with an edge, with a probability of $\frac{1}{2}$.

Now, consider a fixed subset $T \subseteq V$ of size k. The probability that T induces a clique in the graph is precisely $2^{-\binom{k}{2}}$, and the same applies to the probability of T inducing an independent set. By applying the union bound over all such subsets T, we infer that the probability of the sampled graph not being k-Ramsey is at most

$$\binom{n}{k} \cdot 2 \cdot 2^{-\binom{k}{2}}.$$

The key, albeit straightforward, observation is that if this bound is strictly less than 1, there must exist a graph within the sample space, in this case, that is k-Ramsey. By setting $k = 3\log_2 n$, this probability becomes negligible, indicating that almost all graphs are $3\log_2 n$ Ramsey. Further optimization of the constant reveals that $(2 + o(1))\log_2 n$ is also sufficient.

1.1.1.1 Constructing Ramsey graphs

The probabilistic proof, once again, raises the question of how to provide a concrete example of a Ramsey graph. More specifically, for every n, the challenge is to "construct" a k-Ramsey graph where $k = (2 + o(1))\log n$, or even $k = O(\log n)$. In fact, Erdős, known for offering prizes for solving his favorite open problems, offered \$100 for achieving this task.

But what exactly does it mean to "construct" a graph? Mathematicians implicitly mean that the graph can be succinctly communicated or described by a formula of sorts. For instance, using the pigeonhole principle, it can be easily demonstrated that the union of $\sqrt{n}$ disjoint cliques, each of size $\sqrt{n}$, forms a k-Ramsey graph for $k = \sqrt{n}+1$. Although this offers a relatively weak bound, the graph itself is undeniably simple to describe. This construction was further refined through a recursive approach by Abbott in 1972, resulting in an $O(k)$-Ramsey graph, where $k = n^{\log_5 2} \approx n^{0.43}$.

In 1975, Nagy introduced a remarkable "construction" of a graph that further improves upon these bounds. Assume $n = \binom{m}{3}$ and identify the vertices with 3-element subsets of the set $[m]$. Two vertices, corresponding to sets A and B, are connected by an edge precisely when $|A \cap B| = 1$. This clear and concise description certainly qualifies as "explicit" by any standard. Nagy proved that this ingenious construction is k-Ramsey for $k = O(n^{1/3})$.

In a landmark paper, Frankl and Wilson in 1981 advanced Nagy's approach to construct the first $k = n^{o(1)}$ Ramsey graph. The specific bound they achieved was $k = 2^{O(\sqrt{\log n \cdot \log\log n})}$. Their approach also employed restricted intersection sizes of sets as the basis for their graph's structure.

All these graphs are considered explicit by any standard. However, formally defining what makes a graph "explicit" seems to invariably involve complexity theory. Indeed, we aim to exclude approaches like the following "construction": Examine every graph on n vertices and identify the first one that is $3\log n$-Ramsey. While this method is straightforward to describe – requiring only 15 words – it's practically infeasible due to its $2^{O(n^2)}$ time complexity. Complexity theory provides the appropriate framework for evaluating the efficiency of such algorithms.

We consider a graph on n vertices to be explicit if an algorithm exists that, given n as input, can generate the graph in time $n^{O(1)}$. Unfortunately, such a definition omits any subjective sense of elegance or simplicity. An algorithm that constructs a graph need not be a straightforward formula; it can be significantly more complex. In a fascinating development, following a series of advancements, computer scientists have substantially improved upon the Frankl-Wilson construction. Currently, we possess an explicit construction of k-Ramsey graphs for $k = (\log n)^{O(1)}$.

In fact, the construction of the Ramsey graph is explicit in a more robust sense: it allows for local queries in the following manner. Specifically, there exists an algorithm that, when provided with two vertices u and v, determines in polynomial time whether these vertices are connected by an edge. Considering that the input length is $2\log n$, the algorithm's runtime is $(\log n)^{O(1)}$. However, I sense that many mathematicians remain uncomfortable with these constructions due to their complexity. While I empathize with these sentiments as a mathematician, I regard such objections as mathematically unfounded from a complexity theory standpoint.

But why do computer scientists cared about Ramsey graphs in the first place? It turns out that these are related to the problem of "extracting" randomness from defective random sources. We will touch upon this in Section 2.4.

1.1.2 The probabilistic method vs. lower bounds

The strength of the probabilistic method resides in its simplicity. It offers a unique form of 'cheating' — proving the existence of a sought-after object without necessarily gaining substantial, or any, insight into its nature. Therefore, as a pseudorandomnessist, one utilize the probabilistic method to establish a benchmark, which then guides the search for explicit constructions.

But to what extent should we trust the probabilistic method as a benchmark? This inquiry encompasses two distinct aspects: a quantifiable one, and the issue of explicitness which we consider in Section 1.1.3. As for the first issue, despite significant focus, there remains a persistent gap between the upper and lower bounds proved by Erdős and Ramsey, respectively. The critical question is: which of these bounds is correct, or is the actual value somewhere in between? Remarkably, in a recent groundbreaking development [**CGMS23**], the lower bound has been improved to $(\frac{1}{2}+\varepsilon)\log n$ for some universal constant $\varepsilon > 0$, marking the first major progress in approximately 80 years. However, the question remains wide open.

The general rule of thumb is that (good) lower bounds are far harder to prove than upper bounds using the probabilistic method. The main source for that is that proving a lower bound requires a deep understanding of the problem at hand, arguably even more so than required for constructing an upper bound explicitly. In almost all cases studied so far, the lower bounds proved are matched by the probabilistic method. There are two remarkable instances in which this is not the case: Ramanujan graphs and Algebraic-geometric codes. We delve into these in Sections 1.2 and 1.3.

1.1.3 The probabilistic method as a benchmark

Here is a frightening thought: it is possible that no *explicit* k-Ramsey graph exists that matches the outcomes achieved through the probabilistic method, not to mention the lower bound. Namely, it could be the case that efficient computation inherently limit the quantitative bound achievable of Ramsey graphs, and any problem in general. If this were true, then the benchmark we are striving for would be unachievable, and we might not even realize this limitation. I often liken this scenario to an analogy of Gödel's First Incompleteness Theorem, a parallel I sincerely hope does not apply in this context. Our inability to know, for a given instance, whether this is the case or not is an analogy to Gödel's Second Incompleteness Theorem.

1.1.4 Randomness in mathematics

Randomness is not only useful in proofs of mathematical theorems but rather in mathematics itself. Consider for example the drunken walk. Starting at 0, at each time step a drunken person move either a step left or a step right with equal probability. It is well-known that in such case, after n steps, with high probability, it will be at distance $O(\sqrt{n})$ from the origin.

Now consider a drunken mathematician. Obviously, at step k, she will move either left or right, or step put, according to the parity of the distinct number of primes dividing k, staying put if a prime appears with multiplicity (this is the Mobious function). Will the same phenomena occurs? It turns out that this is equivalent to the Riemann Hypothesis! A weaker statement is equivalent to the prime number theorem.

What about things we do know? Well, one such instance is Weil's bound. Let $f(x) \in \mathbb{F}_q[x]$ which is not a square root, and let $\chi : \mathbb{F}_q \to \{0,1,-1\}$ indicating if x is a quadratic residue (and χ vanishes only at 0). Then,

$$|\sum_{x\in\mathbb{F}_q} \chi(f(x))| \le O(d\sqrt{q}).$$

1.2 Error correcting codes

In order to tell you the amazing story about algebraic-geometric codes, we have to start at the beginning, defining a code and recall the mother of all codes - the Reed-Solomon code.

1.2.1 A quick and dirty introduction to coding theory

Coding theory addresses the problem of communicating over an imperfect channel. Classically, the setting is as follows. Alice wishes to communicate a message m to Bob over a channel that can be tampered by an adversary. How should Alice encode m so that if the amount of errors is not excessive, Bob would be able to recover m? To this end, error correcting codes were first introduced by Shannon (1948).

In formal terms, an *error correcting code* over an alphabet Σ is defined as any mapping $\mathsf{C} : \Sigma^k \to \Sigma^n$. However, our interest primarily lies in those mappings where the points in the image are well separated. To quantify this, we introduce the concept of the *distance* of C, defined as:

$$\mathsf{dist}(\mathsf{C}) = \min_{\substack{x,y\in\Sigma^k \\ x\neq y}} \mathsf{dist}(\mathsf{C}(x),\mathsf{C}(y)),$$

where the distance on the right-hand side refers to the Hamming distance. The *relative distance* of C is then derived by normalizing $\mathsf{dist}(\mathsf{C})$, which is given by $\delta(\mathsf{C}) = \frac{1}{n}\mathsf{dist}(\mathsf{C}) \in [0,1]$. The *rate* of the code, denoted as $\rho(\mathsf{C}) = \frac{k}{n}$, reflects the "density" of C's image within its range. Elements within this image are known as *codewords*.

It is important to note that the distance and rate of a code are solely dependent on the image of C, not on the mapping itself. Therefore, in discussions about the distance and rate of a code, one typically refers to C by its image, setting aside the details of the actual mapping.

There is a discernible tension between the two parameters of distance and rate. Indeed, embedding an increasing number of points in the range Σ^n — thereby elevating the rate ρ — inherently complicates the task of maintaining ample spacing between these points. This challenge typically leads to a reduction in the distance δ. Consequently, the pertinent question arises: what is the trade-off between these two parameters? Specifically, if we accord equal significance to both parameters, it becomes intriguing to investigate the maximum achievable sum of $\rho + \delta$.

The Singleton bound, established in 1964 (though previously proven by Joshi in 1958 and even earlier by Komamiya in 1953), asserts that for any code $\mathsf{C} : \Sigma^k \to \Sigma^n$, the following holds: $\rho + \delta \le 1 + \frac{1}{n}$. The proof of this theorem is straightforward and relies on the pigeonhole principle: Consider the first $n-d+1$ coordinates, where $d = \mathsf{dist}(\mathsf{C})$. If $k > n-d+1$, then it becomes inevitable for two distinct codewords to match in these coordinates, consequently reducing the distance to less than d.

1.2.2 The Reed-Solomon code

Thus, the Singleton bound establishes a limit of feasibility. This naturally leads to the question: Is it possible to achieve the Singleton bound? Remarkably, the answer is yes. To demonstrate this, let $q \ge n$ be a prime power and let $\Sigma = \mathbb{F}_q$. We can then define the code $\mathsf{RS} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ in the following manner. First, we represent an input $a = (a_0, a_1, \dots, a_{k-1}) \in \mathbb{F}_q^k$ for RS as a polynomial $f_a(x) \in \mathbb{F}_q[x]$. This polynomial is defined by $f_a(x) = \sum_{i=0}^{k-1} a_i x^i$. The corresponding codeword for the message a is derived by evaluating the polynomial f_a at n distinct points. To formalize this, assume $P_1, \dots, P_n$ are distinct points in $\mathbb{F}_q$ (this is why we require $q \ge n$). We then define:

$$\mathsf{RS}(a) = (f_a(P_1), \dots, f_a(P_n)).$$

It's important to recognize that RS is not just any mapping; it is, in fact, a linear map. A code defined by a linear map is known as a *linear code*. In the case of linear codes, the calculation of the distance becomes more straightforward. The distance of the code can be expressed as:

$$\mathsf{dist}(\mathsf{C}) = \min_{x \ne 0} \mathsf{dist}(\mathsf{C}(x), 0).$$

Therefore, to analyze the distance of RS, our task is to establish an upper bound on the number of zeros that any polynomial f_a can have, for every a. Since the degree of f_a is at most $k-1$, as per the fundamental theorem of algebra, the number of zeros of f_a is limited to $k-1$. Consequently, the distance of the code is at least $d \ge n-(k-1)$, thereby reaching the Singleton bound.

If this is your first encounter with such concepts, it might be quite surprising to discover that the code is constructed by interpreting the message as a polynomial, and it hinges on nothing less than the fundamental theorem of algebra. It turns out that polynomials over finite fields are incredibly potent tools in coding theory and in the study of pseudorandomness. This is because, in these fields, we seek to construct mappings that can be rigorously analyzed, necessitating a certain level of structure. This is in stark contrast to the probabilistic method, which tends to overlook structural aspects. The need for structure has led to the widespread use of polynomials over finite fields. Their utility largely stems from the fundamental theorem of algebra. The code mentioned above, renowned and highly influential in the field, is known as the Reed-Solomon code, introduced in 1960.

1.2.3 The Gilbert-Varshamov bound

A downside of the Reed-Solomon code is that it requires the alphabet size, Σ, to grow with the block length n. This can be limiting - if we want to send a book in English, we want to keep the alphabet to the English letters and some additional characters. It makes little sense to increase the alphabet size as the book length increases. This becomes even more apparent when thinking of binary codes, namely, when we send bits, $\Sigma = \{0,1\}$. The question emerges: For a given alphabet size $|\Sigma| = q$, what is the best distance-rate tradeoff that can be gained?

In case it is not yet clear, the first thing we should do is to try and use the probabilistic method. Doing so leads to the famous Gilbert-Varshamov Bound (1952, 1957) which, informally, states that for every ρ, δ satisfying

$$\rho + \delta \geq 1 - \Omega\left(\frac{1}{\log q}\right)$$

there exists a code with relative distance δ and rate ρ.

In this context, the quantity $\frac{1}{\log q}$ emerges as particularly significant. It represents the compromise one must accept when opting to work with a smaller alphabet, even with the full leverage of the probabilistic method at one's disposal. The probabilistic method seems to be exceedingly well-suited for this scenario. Conceptually, the task of designing a code can be likened to a packing problem. For instance, if our goal is to create a code with distance d, we can visualize each codeword as a ball centered at the codeword and having a radius of d. This visualization helps ensure the required distance between different codewords. Our objective then becomes to pack as many of these balls as possible into the space of Σ^n. Approaching this task randomly appears to be a very rational strategy.

Can we match the Gilbert-Varshamov bound explicitly? Can we do better, explicitly or otherwise?

1.2.4 Reed-Muller codes

The first natural attempt at reducing the alphabet size from n down to something of significance, leads to the famous Reed-Muller code. Let q be a prime power, and let k, n be integers such that $k \leq n = q^2$. Assume that r is an integer such that $k = \binom{r+2}{2}$ holds. Let $P_1, \dots, P_n$ be distinct elements in $\mathbb{F}_q \times \mathbb{F}_q$. Similarly to the Reed-Solomon code, we identify a message a with a polynomial, this time though, with a bivariate polynomial $f_a(x,y) \in \mathbb{F}_q[x,y]$. Observe that, as $k = \binom{r+2}{2}$, this identification can be done using polynomial of total degree at most r. More explicitly, under a suitable indexing of the the message, we can write $a = (a_{i,j})_{0 \leq i+j \leq r}$, and then define

$$f_a(x,y) = \sum_{i,j} a_{i,j} x^i y^j.$$

With this, we define the Reed-Muller code by $\mathsf{RM} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ by $\mathsf{RM}(a) = (f_a(P_1), \dots, f_a(P_n))$.

The rate of RM is easy to analyze: we have that $k = \binom{r+2}{2} \geq \frac{r^2}{2}$, and so $\rho \geq \frac{r^2}{2n}$. What about the distance? We need some two-dimensional analog of the fundamental theorem of algebra. This is precisely the ever so useful Schwartz-Zippel lemma.

Lemma 1.1 — The Schwartz-Zippel Lemma. Let $f[x,y] \in \mathbb{F}_q[x,y]$ be a polynomial of total degree at most d. Then, the number of zeros of f in $\mathbb{F}_q^2$ is bounded by dq.

From this it follows that the relative distance $\delta \geq 1 - \frac{r}{\sqrt{n}}$. Hence, $\rho \geq \frac{1}{2}(1-\delta)^2$. To recap, the Reed Muller code can support smaller field size, $q = \sqrt{n}$, compared to Reed-Solomon. However, the rate is bounded above by $\frac{1}{2}$, even if the distance approaches 0. When increasing the dimension from 2 to some parameter m, the requirement on the field size deteriorates as expected to $q \leq n^{1/m}$. Unfortunately, however, the rate deteriorates very rapidly, at a pace of roughly $\frac{1}{m!}$.

This rate barrier was there for a while until a clever way of breaking it has been found. In addition to outputting the evaluations of the polynomial, also output the evaluation of the directional derivatives. To illustrate this in the two-dimensional case, we again identify the message a with a bivariate polynomial of total degree r, where as before $k = \binom{r+2}{2}$, and consider the code $\mathsf{RM}' : \mathbb{F}^k \to (\mathbb{F}_q^3)^n$ defined by

$$\mathsf{RM}'(a) = \left(\left((f_a(P_1), \frac{\partial f_a}{\partial x}(P_1), \frac{\partial f_a}{\partial y}(P_1) \right), \ldots, \left((f_a(P_n), \frac{\partial f_a}{\partial x}(P_n), \frac{\partial f_a}{\partial y}(P_n) \right) \right).$$

This suggestion seems futile as now we output symbols of size that are 3 times larger than the original code. This deteriorates the rate which, in the case of output alphabet that may not necessarily equal the input alphabet, scales in the natural way. That is, we have that

$$\rho = \frac{k}{3n} \geq \frac{1}{6}\left(\frac{r}{q}\right)^2. \tag{1.1}$$

The point of using derivatives is that now, for a symbol, say one corresponding to a point P, to vanish, it holds that $f_a(P) = \frac{\partial f_a}{\partial x}(P) = \frac{\partial f_a}{\partial y}(P) = 0$. Put differently, f_a vanishes at P with multiplicity two. The appropriate generalization of the Schwartz-Zippel lemma (Lemma 1.1) that takes into account the multiplicity is as follows.

Lemma 1.2 Let $f[x,y] \in \mathbb{F}_q[x,y]$ be a polynomial of total degree at most d. Then,

$$\sum_{P \in \mathbb{F}_q^2} \mathsf{mult}(f,P) \leq dq,$$

where $\mathsf{mult}(f,P)$ is the multiplicity of P as a zero of f.

With this, the number of zeros in $\mathsf{RM}''(a)$ is bounded above by $\frac{dq}{2}$ as each zero contributes a multiplicity of 2 to the LHS of the summation appearing in Lemma 1.2. Thus, $\delta \geq 1 - \frac{r}{2q}$. Plugging this to the rate calculation we did before (see Equation (1.1)), we get that

$$\rho \geq \frac{2}{3}(1-\delta)^2.$$

Using higher-order derivatives we can increase the rate as close as we wish to 1 albeit, the symbol sizes of the output will increase which can be viewed as some sort of cheating. After all, the code that outputs one huge symbol which consists of the entire message has distance $\delta = 1$. Nonetheless, the growth of symbol size is not that terrible and a known technique, dubbed code concatenation, can reduce the symbol size back to $\mathbb{F}_q$ with low cost in terms of rate and distance.

1.2.5 Algebraic-Geometric codes

Reed-Muller codes and their extension - multiplicity codes - are great. They have some further properties you would want from a code. But, as a way of achieving great rate with a small field size, their utility is limited. Remarkably, with a far deeper mathematics, it is possible not only to match, but in fact outperform the probabilistic construction, namely, the Gilbert-Varshamov bound. Indeed, the revelation of the following theorem was a very surprising development in coding theory.

Theorem 1.1 — Ihara (1981); Tsfasman-Vladut-Zink (1982); Garcia-Stichtenoth (1995). For every q which is an even power of a prime and all ρ, δ satisfying

$$\rho + \delta \geq 1 - \frac{1}{\sqrt{q}-1}$$

there exist *explicit* arbitrary long codes with distance δ and rate ρ.

These codes, known as Algebraic-Geometric codes (AG codes) or Goppa codes, named after Goppa who proposed them in 1981, surpass the probabilistic method exponentially in their dependence in q. This enhanced performance is attributed to their ability to exploit underlying structural properties, enabling them to achieve a packing density far beyond what random placement of elements could accomplish.

The construction of these codes is remarkably elegant, and their analysis draws upon profoundly deep results from algebraic geometry (or, more precisely, arithmetic geometry as the finite field structure comes into play). A key element in this analysis is the utilization of the analog of the Riemann Hypothesis for curves over finite fields, initially proposed by Artin in 1924. This monumental question was subsequently addressed by Hasse, who successfully proved it for the genus 1 case, and by Weil in 1949, who provided a general proof. Despite the considerable mathematical depth required for analyzing these codes, we can still appreciate and discuss the essence of their construction. This aspect of the codes, independent of the intricate analysis, stands out for its conceptual innovation and the elegance of its design.

Like Reed-Solomon codes, Algebraic-Geometric codes (AG codes), also known as Goppa codes after their proposer Goppa in 1981, are a type of *evaluation codes*. This means that the message is interpreted as a function, and the corresponding codeword is generated by evaluating this function at a set of points. In the case of Reed-Solomon codes, the evaluation points are selected arbitrarily in $\mathbb{F}_q$. However, the choice of evaluation points in AG codes is much more subtle. If we fix the field size q, the question arises: where should we choose numerous evaluation points from? The answer lies in selecting points on an algebraic curve over $\mathbb{F}_q$.

To illustrate this, we will consider a celebrated AG code suggested by Garcia and Stichtenoth (1995). Consider an even prime power q and denote $p = \sqrt{q}$. We can look at the set of points $(x,y) \in \mathbb{F}_q^2$ that satisfy $\mathsf{GS}(x,y) = 0$, where

$$\mathsf{GS}(x,y) = y^p - y - \frac{x^p}{1-x^{p-1}}.$$

It can be shown that the number of such points is $\Omega(p^3)$, out of the p^4 points in the plane $\mathbb{F}_q \times \mathbb{F}_q$. To increase the number of points, we might consider curves of higher dimensions defined similarly. For example, in three dimensions, we could use the curve in $\mathbb{F}_q^3$ consisting of all (x,y,z) such that $\mathsf{GS}(x,y) = \mathsf{GS}(y,z) = 0$, and so on. Despite the alphabet size being fixed to q, it is shown that these curves contain a significant number of points, which will serve as the evaluation points for the code.

The next question is: what functions should we evaluate? In Reed-Solomon codes, we utilized low-degree polynomials, as we had a bound on the number of zeros due to the fundamental theorem of algebra. However, the situation is much more intricate for AG codes. Different polynomials can yield identical functions when restricted to the curve $\mathsf{GS}(x,y)$, and we also need to consider how to bound the number of zeros a polynomial has when restricted to a curve. The functions to be evaluated in AG codes are, in fact, rational functions, not just polynomials, derived from a space known as the Riemann-Roch space. The profound Riemann-Roch theorem addresses the dimension of this space, which, in conjunction with the number of points on the curve, determines the code's rate. Notably, the curve's property, called its *genus*, influences the distance-rate tradeoff. In this regard, the Garcia-Stichtenoth curve is known for offering the best tradeoff.

1.3 Expander graphs

Spectral graph theory is the study of graphs by examining the eigenvalues and eigenvectors of matrices that are associated with the graph. Given a graph $G = (V,E)$, one can consider the *adjacency matrix* $\mathbf{M}_G$ whose rows and columns are indexed by V, and the (u,v) entry of $\mathbf{M}_G$ is 1 if $(u,v) \in E$ and 0 otherwise. We are mostly (but not solely) interested in undirected graphs, and so from hereon we assume a graph is undirected. A matrix related to $\mathbf{M}_G$ is the *random walk matrix* $\mathbf{W}_G = \mathbf{M}_G \mathbf{D}_G^{-1}$ where $\mathbf{D}_G$ is the diagonal matrix whose (v,v) entry equals to the degree of v. Both $\mathbf{M}_G$ and its "normalized" variant $\mathbf{W}_G$ are typically used when viewed as operators. A third important matrix that is associated with a graph G is its *Laplacian* $\mathbf{L}_G = \mathbf{D}_G - \mathbf{M}_G$. The Laplacian is used when considering the quadratic form $\mathbf{x}^\top \mathbf{L}_G \mathbf{x}$. Indeed, as $\mathbf{x}^\top \mathbf{L}_G \mathbf{x} = \sum_{uv \in E} (\mathbf{x}(u) - \mathbf{x}(v))^2$, the Laplacian measures how "smooth" is $\mathbf{x}$ with respect to G. However, our focus in this section will be on $\mathbf{M}_G$ and $\mathbf{W}_G$.

A basic fact in linear algebra asserts that an $n \times n$ real symmetric matrix has n real eigenvalues when counted with multiplicities. Moreover, there is an orthonormal basis of $\mathbb{R}^n$ that is composed of the matrix eigenvectors. We denote the eigenvalues of $\mathbf{M}_G$ by $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$, and the eigenvalues of $\mathbf{W}_G$ by $\omega_1 \geq \cdots \geq \omega_n$. The eigenvalues of $\mathbf{L}_G$ are denoted in reversed-order $\lambda_1 \leq \cdots \leq \lambda_n$. For d-regular graphs, which will be our focus in this section, one has that $\mu_i = d\omega_i = d - \lambda_i$ for all $i \in [n]$.

1.3.0.1 What does the first few eigenvalues tell us?

A general theme in spectral graph theory is that of extracting combinatorial properties of graphs from the spectrum (i.e., the multi-set of eigenvalues) of one of the above matrices. The first eigenvalue (in the above notation) encodes little information about the graph. Indeed, for every graph, $\omega_1 = 1$ and $\lambda_1 = 0$ which encodes no information about the graph (besides it being finite). The largest eigenvalue of the adjacency matrix, μ_1, does encode some information about the graph's degrees, e.g., μ_1 is bounded between the largest degree and the average degree of G.

The second eigenvalue tells us much more about the graph and, in particular, in a sense, to what extent the graph is connected. It is an easy fact that $\lambda_2 = 0$ if and only if the graph is disconnected. This result can be extended in two directions. First, the multiplicity of the zero eigenvalue (namely, the dimension of the kernel of $\mathbf{L}_G$) is precisely the number of connected components in G. The second, more interesting generalization gives a quantitative relation between λ_2 and the "connectedness" of G. Formally, for a set $S \subseteq V$ let $\partial(S) = \{uv \in E \mid u \in S, v \notin S\}$, and define the isoperimetric ratio of G by $\theta_G = \min_{|S| \leq n/2} \frac{|\partial(S)|}{|S|}$. With this definition, the larger θ_G is, the "more connected" is G. A graph is said to be θ*-edge expander* if $\theta_G \geq \theta$. An important theorem due to Cheeger relates the spectral quantity $\lambda_2(G)$ with the combinatorial quantity θ_G. Namely, for a d-regular graph G,

$$\frac{\lambda_2}{2} \leq \theta_G \leq \sqrt{2\lambda_2 d}. \tag{1.2}$$

1.3.0.2 Imitating independent samples

A good spectral expander has the remarkable property of allowing us to imitate truly independent samples. To formalize this, let $G = (V,E)$ be an undirected graph consisting of n vertices. Given that G is undirected, its adjacency matrix $\mathbf{M}$ is symmetric. Thus, according to the spectral decomposition theorem, $\mathbf{M}$ can be represented using an orthonormal basis of eigenvectors $\psi_1, \dots, \psi_n \in \mathbb{R}^n$, each ψ_i corresponding to the eigenvalue μ_i. This allows us to express $\mathbf{M}$ as

$$\mathbf{M} = \sum_{i=1}^{n} \mu_i \psi_i \psi_i^\top.$$

For the sake of simplicity, we will assume henceforth that G is a d-regular graph. This assumption simplifies our presentation. In such case, it is evident that the largest eigenvalue μ_1 is equal to d. Furthermore, the corresponding eigenvector ψ_1 can be identified as the vector with all entries equal to $\frac{1}{\sqrt{n}}$, which is commonly represented by $\mathbf{1}$. Since $\mathbf{1}\mathbf{1}^\top = \mathbf{J}$ - the matrix all of whose entries are $\frac{1}{n}$, we see that

$$\frac{1}{d}\mathbf{M} = \mathbf{J} + \sum_{i=2}^{n} \frac{\lambda_i}{d} \psi_i \psi_i^\top.$$

Note that the RHS is precisely $\mathbf{W}$, the random walk matrix of G. In this context, if $\mathbf{p} \in \mathbb{R}^V$ represents a distribution over the vertices V, then $\mathbf{Wp}$ describes the new distribution achieved by first selecting a vertex according to $\mathbf{p}$ and then moving to a neighboring vertex at random in G. Similarly, $\mathbf{J}$ acts as the random walk matrix for a complete graph with self-loops, where a random step equates to choosing any vertex uniformly at random, independent of the starting vertex. This latter process epitomizes pure randomness, essential in randomness studies. This leads us to write

$$\mathbf{W} - \mathbf{J} = \mathbf{E},$$

where $\mathbf{E} = \sum_{i=2}^{n} \omega_i \psi_i \psi_i^\top$ quantifies the "distance" between the random walk matrix of G and the ideal random walk matrix (i.e., for a complete graph with self-loops). Ideally, we aim for a graph G where $\mathbf{E}$ is minimal. A practical measure of $\mathbf{E}$'s "smallness" is its spectral norm $\|\mathbf{E}\|_2$, which is also equivalent to $\max(\omega_2, |\omega_n|)$. This value, denoted as $\omega(G)$, is referred to as the *normalized spectral expansion* of G. The *spectral expansion* is defined naturally as $\mu(G) = d\omega(G) = \max(\mu_2, |\mu_n|)$. The question then arises: How low can $\omega(G)$ be for a d-regular graph?

A graph G is called an *α-spectral expander* if $\mu(G) \leq \alpha d$. When referring to an expander, without explicitly referring to the parameter α, one typically means that G is α-spectral expander for some constant $\alpha < 1$. We usually consider not a single graph but a family of d-regular graphs, of a growing number of vertices, all of which are α-spectral expanders for some constant $\alpha < 1$. Cheeger's inequality, Equation (1.2), gives an equivalence (up to some quantitative loss) between edge expansion and spectral expansion.

1.3.0.3 Ramanujan graphs

Given the above discussion on random walks, and the relation to edge expansion, it is natural to ask what are the "best" spectral expanders. Namely, what is the least possible value of $\mu(G)$ for a d-regular graph G. On top of being a fundamental question, it is well-motivated by applications. Indeed, while some applications only require $\mu \leq \alpha d$ for some constant $\alpha < 1$ others heavily deteriorate as μ increases. It is a well-known pheonmena in random matrix theory that eigenvalues tend to "repel" each other and so one does not expect all of them to lie in a small interval around zero. To see this, consider G with no self-loops, and let us calculate the average square distance between the eigenvalues of the corresponding adjacency matrix,

$$\mathop{\mathbf{E}}_{i,j\sim[n]} (\mu_i - \mu_j)^2.$$

To this end, first note that

$$0 = \operatorname{Tr}(\mathbf{M}) = \sum_{i=1}^{n} \mu_i,$$

and so

$$0 = \operatorname{Tr}(\mathbf{M})^2 = \sum_{i,j=1}^{n} \mu_i \mu_j.$$

Moreover, we observe that

$$dn = \operatorname{Tr}(\mathbf{M}^2) = \sum_{i=1}^{n} \mu_i^2.$$

With this, we have that

$$\sum_{i,j=1}^{n} (\mu_i - \mu_j)^2 = \sum_{i,j=1}^{n} \mu_i^2 + \mu_j^2 - 2\sum_{i,j=1}^{n} \mu_i \mu_j = 2dn^2.$$

Thus,

$$\mathop{\mathbf{E}}_{i,j\sim[n]} (\mu_i - \mu_j)^2 = 2d.$$

This tells us that, in terms of the Euclidean distance (rather than in terms of absolute values), the average distance between eigenvalues is $2d$. In particular, $|\mu_1 - \mu_n| \geq \sqrt{2d}$, which immediately implies that $\mu(G) \geq \sqrt{d/2}$.

The Alon-Boppana bound [**Nilli91**, **Friedman93**] gives the essentially optimal quantitative bound

$$\mu(G) \geq 2\sqrt{d-1}\left(1 - \frac{2}{k+1}\right),$$

where k is the diameter of G. In particular, as $k \geq \log_d n$, thinking of d is fixed and $n \to \infty$, $\mu(G) \geq 2\sqrt{d-1} - o(1)$. A graph G is called *Ramanujan* if it (essentially) meets this bound, namely, $\mu(G) \leq 2\sqrt{d-1}$. The quantity $2\sqrt{d-1}$ appears naturally as the spectral radius of the infinite d-ari tree which is, in a sense, the best spectral expander provided infinite graphs are allowed. Indeed, Ramanujan graphs, and spectral expanders in general, can be thought of as a finite approximation of the latter.

Answering a conjecture posed by Alon, Friedman [**Friedman08**] proved that a random d-regular graph (under some natural distribution) is very close to Ramanujan, namely, it attains $\mu(G) \leq 2\sqrt{d-1} + \varepsilon(n)$ for some $\varepsilon(n)$ that vanishes as $n \to 0$. It is conjectured that a d-regular graph is Ramanujan with some constant probability [**MNS08**]. Proving that is a fundamental open problem which essentially asks whether Ramanujan graphs are "special". Recently, Mohanty, O'Donnell and Paredes [**MOP20**] gave a weakly explicit construction of infinite families of graphs G with $\mu(G) \leq 2\sqrt{d-1} + \varepsilon(n)$ of every degree d. By "weakly explicit" we mean that the graph can be produced in time polynomial in the number of its vertices.

As for Ramanujan graphs, already at the get go, Margulis [**Margulis88**] and Lubotzky, Phillips and Sarnak [**LPS88**] gave the first constructions of infinite families of Ramanujan graphs. These have degrees $p+1$, for a prime p. These construction are *strongly explicit*, meaning, given a vertex v and a number $k \in [d]$, the k^{th} neighbor of v (under some fixed, yet arbitrary order) can be computed in polynomial time in the input length, namely, in time $O(\log n)$. Some applications of expanders heavily rely on strongly explicitness. There have been several other explicit constructions all of which produce graphs of degree $q+1$ for a prime power q. These constructions are extremely elegant. However, the analysis is based on heavy mathematical machinery from number theory. As a consequence, it only provides Ramanujan graphs of degrees as above. No less important, the constructions are hard to "tweak" as required by some applications which build on certain variants of expander graphs. This latter flexibility is key in several important applications.

1.3.0.4 Explicit constructions of expander graphs

The seminal work of Reingold, Vadhan and Wigderson [**RVW02**] introduced the Zig-Zag product of graphs and used it to give strongly explicit constructions of expanders of *every* degree d having $\mu = O(d^{2/3})$. While far from the $2\sqrt{d-1}$ bound of Ramanujan graphs, the bound is much better than the vanilla αd bound for $\alpha < 1$ a constant. Moreover, the ideas underlying the construction were key to the breakthrough result of Reingold, **SL** = **L** [**Reingold08**].

The Zig-Zag product of graphs motivated the work of Ben-Aroya and Ta-Shma [**BATS11**] who obtained a strongly explicit construction with $\mu = \sqrt{d} \cdot 2^{O(\sqrt{\log d})} = d^{\frac{1}{2}+o(1)}$. While still quite far from Ramanujan, the ideas that went into the construction, and its improved parameters, were the basis for Ta-Shma's breakthrough work [**TaShma17**] on the construction of binary codes that nearly match the Gilbert-Varshamov bound. Equivalently, Ta-Shma constructed near-optimal small-bias sets.

Independently of the above line of work, Bilu and Linial [**BL06**] gave a weakly explicit construction of expander graphs with a stronger bound $\mu \leq \sqrt{d} \cdot O((\log d)^{3/2})$. Moreover, they proposed a conjecture that, if holds, yields Ramanujan graphs of every degree. In a tour de force result, Marcus, Spielman and Srivastava [**MSS15**] proved the Bilu-Linial conjecture for *bipartite* graphs. Their proof consists of two independent parts, both of which are novel and exciting. The first is to employ (or, more precisely, adapt) results from *free probability theory*, and the second is to exploit a property of families of real polynomials called interlacing.

1.3.0.5 Ramanujan graphs via free probability and interlacing families

To achieve their results, Marcus, Spielman, and Srivastava [**MSS15**] examined the union of d independent perfect matchings to form their graph. Specifically, they took the adjacency matrix $\mathbf{M}$ of a single fixed matching and combined it with d independent permutation matrices $\mathbf{P}_1,\dots,\mathbf{P}_d$. The adjacency matrix of the graph they proposed is represented as $\mathbf{P}_1^\top \mathbf{M}\mathbf{P}_1 + \cdots + \mathbf{P}_d^\top \mathbf{M}\mathbf{P}_d$. The first step of their proof involved analyzing the expected characteristic polynomial corresponding to these graphs. Marcus *et al.* observed that free probability theory "predicts" that the second largest root of the expected characteristic polynomial is bounded above by $2\sqrt{d-1}$. Subsequently, they formulated a finite version of the relevant aspects of free probability theory, which allowed them to establish a similar bound in a finite setting. Essentially, finite free probability theory offers a way to predict asymptotic behavior by examining limit behavior, aiming to achieve the desired bounds in the asymptotic context.

However, knowledge about the roots of the expected polynomial's behavior does not necessarily provide information about the roots of each individual polynomial in the distribution. In simpler terms, the bound on the spectrum of the expected characteristic polynomial doesn't automatically translate to a bound on a specific characteristic polynomial within the distribution. Generally, the probabilistic method is not effective in this scenario. Nonetheless, the second part of their proof ingeniously utilized a particular structure related to root interlacing in the polynomials. This structure allowed them to infer a bound on an individual polynomial in the distribution based on the established bound for the expected polynomial. Therefore, in a broad sense, MSS managed to adapt the principles of the probabilistic method to eigenvalue analysis, despite the process being based on coefficient space expectations.

1.4 Combining structure and randomness: tree codes as a case study

There are instances where the probabilistic method is applied in a more subtle manner than merely considering a uniform distribution over a set of objects. Sometimes, it becomes necessary to infuse some structure into the randomness to better address the problem at hand. In other situations, more advanced tools from probability theory are utilized, moving beyond basic concepts like the union bound and linearity of expectation. Interestingly, both of these approaches can be exemplified through one of my favorite problems: the construction of tree codes.

While, as discussed in Section 1.2.1, error-correcting codes can be used to solve the problem of sending a single message from Alice to Bob over an imperfect channel, in some settings, the two parties interact with each other, sending multiple messages where a message depends on previous messages that were exchanged. Interactive coding addresses the subtler problem of enabling such dynamic interaction over an imperfect channel. In this far more challenging setting, standard codes do not offer a satisfactory solution.

Tree codes are powerful combinatorial structures, defined and studied in a celebrated sequence works by Schulman [**Schulman93**, **Schulman96**] as key ingredients for achieving interactive coding schemes. They play a role analogous to the one error-correcting codes take in the single message setting. Tree codes, as their name suggests, are trees with certain distance properties. To give the formal definition, we set some notation. Let T be the infinite complete rooted binary tree that is endowed with an edge coloring from some ambient color set, or alphabet, Σ. For vertices u,v of equal depth let w be their least common ancestor and denote the distance, in edges, from u to w by ℓ. Let $p_u, p_v \in \Sigma^\ell$ be the sequences of colors on the path from w to u and to v, respectively. We define $h(u,v)$ to be the relative Hamming distance between p_u and p_v. Informally, $h(u,v)$ measures the distance between the two color sequences obtained by following the paths from the root to each of u and v, excluding the “non-interesting” common prefix. A tree code is any coloring that has a lower bound on this quantity. Formally,

Definition 1.2 — Tree codes. Let T be the complete infinite rooted binary tree. The tree T, together with an edge-coloring of T by a color set Σ is called a *tree code with distance* δ if for every pair of vertices u, v with equal depth, it holds that $h(u,v) \geq \delta$.

At first glance, it’s not immediately apparent that a tree code with a distance $\delta > 0$ actually exists. Let us consider the naive attempt at invoking the probabilistic method where we sample a color to any given edge uniformly and independently across edges. First thing first, this description does not make sense as there is no distribution which is uniform over an infinite support. Even ignoring that, say by analyzing a tree code with a finite depth of n, by using c colors, the likelihood of any two sibling nodes being connected to their parent by edges of the same color is $\frac{1}{c}$. Therefore, unless c is on the order of $\Omega(2^n)$, it’s highly probable that such sibling pairs exist, which would consequently reduce the distance to zero.

In fact, the challenge of devising a tree code with finite depth can, in some sense, be seen as a general case. It’s known that if we can devise a series of tree codes, each corresponding to a different depth n, then there exists a method to "glue" these together into a single, infinite tree code without significantly compromising the tree’s parameters. However, the question of how to prove the existence of such a finite tree code still remains. We will present two proofs to address this. The first one (see Section 1.4.1) employs the Lovász Local Lemma, showcasing the application of a more advanced probabilistic tool. The second proof, which is the content of Section 1.4.2, eschewing complex tools, will demonstrate how to incorporate the inherent structure of the problem into the distribution under consideration.

1.4.1 Tree code existence proof via the Lovász Local Lemma

The Lovász Local Lemma (LLL) is a powerful tool in probability theory and combinatorics, particularly useful in situations where you have a large set of events and you want to show that none of these events occur, even though they might be dependent on each other to some extent. This lemma is particularly useful in computer science and combinatorics, for problems where you have to prove that a certain configuration or outcome is possible, despite the presence of potential conflicts or dependencies among components of the system.

Lemma 1.3 — General Lovász Local Lemma (LLL). Let $A_1, A_2, \dots, A_n$ be events in a probability space. Suppose that for each i, A_i is mutually independent of a set of all other events except for a subset $\Gamma(A_i)$, and suppose there exist real numbers $x_1, x_2, \dots, x_n$ in the interval $[0,1)$ such that for each i,

$$\mathbf{Pr}(A_i) \leq x_i \prod_{A_j \in \Gamma(A_i)} (1 - x_j). \tag{1.3}$$

Then,

$$\mathbf{Pr}\left(\bigcap_{i=1}^{n} \overline{A_i}\right) > 0. \tag{1.4}$$

Using LLL, we turn to prove the existence of a tree code, for any desired distance $\delta > 0$, having $2^{\Omega\left(\frac{1}{1-\delta}\right)}$ colors. Consider again the uniform coloring of a depth-n tree code using c colors. Focus on the following set of events: For every pair of paths with equal lengths that originate from the same vertex, define an event where the two paths match in at least a τ-fraction of their corresponding edge pairs. If none of these events occur, a tree code with distance $\delta = 1 - \tau$ is successfully obtained.

To further analyze, these events are categorized based on the depths of the paths. Let A_i represent the set of events associated with pairs of paths having depth i.

By the union bound, for an event A in A_i,

$$\mathbf{Pr}[A] \leq \binom{i}{\tau i} \cdot \left(\frac{1}{c}\right)^{\tau i} \leq 2^{H(\tau)i-\delta i \log c} \leq 2^{\tau \ln(\frac{1}{\tau})i - \tau i \log_2 c}.$$

Thus, as long as $\log_2 c \geq 2 \ln \frac{1}{\tau}$, we obtain the bound

$$\mathbf{Pr}[A] \leq c^{-\frac{\tau i}{2}}. \tag{1.5}$$

Consider a fixed event A. Note that the only events in $\Gamma(A)$ are those corresponding to path pairs that intersect the paths underlying A in edges. Thus, for every $A \in A_i$ and every $j \in [n]$, we can bound $|\Gamma(A) \cap A_j| \leq 2i \cdot 4^j$. Thus,

$$x(A) \prod_{B \in \Gamma(A)} (1 - x(B)) = x(A) \cdot \prod_{j=1}^{n} \prod_{B \in \Gamma(A) \cap A_j} (1 - x(B)).$$

For every $i \in [n]$ and for $A \in A_i$, we set with hindsight, $x(A) = 32^{-i}$. Using that $1 - x \geq e^{-2x}$ for all $x < 0.79$, we have that for $B \in A_j$, $1 - x(B) \geq e^{-2 \cdot 32^{-j}}$, and so

$$\prod_{B \in \Gamma(A) \cap A_j} (1 - x(B)) \geq \left(e^{-2 \cdot 32^{-j}}\right)^{|\Gamma(A) \cap A_j|} = e^{-4i \cdot 8^{-j}}.$$

Thus,

$$x(A) \prod_{B \in \Gamma(A)} (1 - x(B)) \geq 32^{-i} e^{-4i \sum_{j=1}^{n} 8^{-j}} \geq (32e)^{-i}.$$

Thus, together with Equation (1.5), the hypothesis of the LLL is satisfied provided that $32e \leq c^{\frac{\tau}{2}}$, which indeed yields a tree code with number of colors $c = e^{\frac{64e}{\tau}} = 2^{\Omega\left(\frac{1}{1-\delta}\right)}$.

1.4.2 Four-color tree codes

In this section, we adopt a different strategy. Instead of employing the Lovász Local Lemma (LLL), we utilize more fundamental probabilistic methods, customizing the distribution specifically for our problem. We will demonstrate the existence of a tree code that has four colors!

Before proceeding, let's first establish why a 3-color tree code is not feasible. Consider any hypothetical 3-color tree code. Initially, we can assume that each pair of sibling vertices is connected to their parent by edges of distinct colors; otherwise, the tree code's distance would be zero. Let u and v be any two vertices. Emanating from u and v, there are four edges. According to the pigeonhole principle, in any 3-color scheme, at least two of these edges must share the same color. Following our initial assumption, one of these edges must extend from u, and the other from v. Consequently, starting from the two children of the root, it is possible to construct two paths of any desired length $\ell \geq 1$ that share the same color pattern. This observation confirms that the tree code has a distance of zero.

We start by setting some notation. Let T be the infinite complete rooted binary tree. We identify length-n paths in T that starts at the root with length-n binary strings in the natural way. Namely, we identify left son and right son with 0 and 1, respectively. Given a node v at depth $n \geq 1$ we define $p_v \in \{0,1\}^n$ to be the string that encodes the (unique) path from the root to v.

An edge-coloring of T by a color set Σ is given by a function, which for ease of readability, we slightly abuse notation and also denote by $T : \{0,1\}^{\mathbb{N}_1} \to \Sigma^{\mathbb{N}_1}$, where the color of an edge $e = u \to v$ is $T(p_v)_{\mathsf{depth}(v)}$. Here, $\mathbb{N}_1$ consists of all positive natural numbers. Note that T is an online function, namely, for every $x \in \{0,1\}^{\mathbb{N}_1}$ and $i \in \mathbb{N}_1$, the value $T(x)_i$ is determined by $x_1, \ldots, x_i$.

The (probabilistic) construction

Let $\{R_i\}_{i \in \mathbb{N}_1}$ be a sequence of independent random variables, each is uniformly distributed over $\mathbb{F}_4$–the field of 4 elements. Let $\mathbb{F}_2$ be the (unique) subfield of $\mathbb{F}_4$ of size 2. Define the (random) coloring function $T : \mathbb{F}_2^{\mathbb{N}_1} \to \mathbb{F}_4^{\mathbb{N}_1}$ (where we identify $\mathbb{F}_2$ and $\{0,1\}$ in the natural way) as follows: for every $t \in \mathbb{N}_1$

$$T(x)_t = \sum_{i=1}^{t} R_{t+1-i} x_i. \tag{1.6}$$

Let v be a depth-n vertex in T. Let $\ell \geq 1$ and $x, y \in \mathbb{F}_2^{\ell-1}$. For $k = 1, \dots, \ell$ we define the random variable

$$a_v(x,y,k) = T(p_v \circ 1 \circ y)_{n+k} - T(p_v \circ 0 \circ x)_{n+k}.$$

Note that $a_v(x,y,k)$ is a (random) element in $\mathbb{F}_4$. We define the integral random variable

$$h_v(x,y) = \sum_{k=1}^{\ell} I_k,$$

where I_k is the indicator random variable that equals 1 when $a_v(x,y,k) \neq 0$. Note that $h_v(x,y) \in \{0,1,\dots,\ell\}$ is the Hamming distance between $T(p_v \circ 0 \circ x)_{[n+1,n+\ell]}$ and $T(p_v \circ 1 \circ y)_{[n+1,n+\ell]}$.

Claim 1.1 Let v be a vertex in T. Let $\ell \geq 1$ and $x, y \in \mathbb{F}_2^{\ell-1}$. Then, for every $k \in \{1,\dots,\ell\}$ it holds that

$$a_v(x,y,k) = R_k + \sum_{i=1}^{k-1} R_{k-i}(y-x)_i.$$

Proof. Denote the depth of v by n. Fix $k \in \{1,\dots,\ell\}$. By Equation (1.6),

$$T(p_v \circ 0 \circ x)_{n+k} = \sum_{i=1}^{n+k} R_{n+k+1-i}(p_v \circ 0 \circ x)_i = \sum_{i=1}^{n} R_{n+k+1-i}(p_v)_i + \sum_{i=1}^{k} R_{k+1-i}(0 \circ x)_i.$$

Similarly

$$T(p_v \circ 1 \circ y)_{n+k} = \sum_{i=1}^{n} R_{n+k+1-i}(p_v)_i + \sum_{i=1}^{k} R_{k+1-i}(1 \circ y)_i.$$

Thus,

$$a_v(x,y,k) = \sum_{i=1}^{k} R_{k+1-i}(1 \circ y)_i - \sum_{i=1}^{k} R_{k+1-i}(0 \circ x)_i = R_k + \sum_{i=1}^{k-1} R_{k-i}(y-x)_i.$$

■

Claim 1.2 Let v be a vertex in T. Let $\ell \geq 1$ and $x, y \in \mathbb{F}_2^{\ell-1}$. Then, the random variables $a_v(x,y,1),\dots,a_v(x,y,\ell)$ are independent and each is uniformly distributed over $\mathbb{F}_4$.

Proof. By Claim 1.1, $a_v(x,y,k) = R_k + L_k$ where L_k is some $\mathbb{F}_4$-linear combination of $R_1,\dots,R_{k-1}$. Therefore, $a_v(x,y,k)$ is independent of the joint distribution of $a_v(x,y,1), \dots, a_v(x,y,k-1)$. As this holds for every k we have that $a_v(x,y,1),\dots,a_v(x,y,\ell)$ are independent. To conclude the proof, note that for every fixing of $R_1,\dots,R_{k-1}$, $a_v(x,y,k) = R_k + \ell_k$ for some fixed $\ell_k \in \mathbb{F}_4$ and so $a_v(x,y,k)$ is uniform over $\mathbb{F}_4$. ■

Claim 1.3 For every two vertices u, v in T and every $x, y \in \mathbb{F}_2^{\ell-1}$,

$$h_v(x,y) = h_u(x,y),$$
$$h_v(x,y) = h_v(0^{\ell-1}, y-x).$$

Proof. The first equality follows immediately by Claim 1.1 as, for every $k \in \{1, \dots, \ell\}$, the expression obtained for $a_v(x,y,k)$ is independent of the choice of v. As for the second asserted equality, again by Claim 1.1,

$$a_v(x,y,k) = R_k + \sum_{i=1}^{k-1} R_{k-i}(y-x)_i = R_k + \sum_{i=1}^{k-1} R_{k-i}((y-x)-0)_i = a_v(0^{\ell-1}, y-x, k),$$

where observe that for the last equality we are using the fact that $\mathbb{F}_2$ is a subfield of $\mathbb{F}_4$ and so $y - x \in \mathbb{F}_2^{\ell-1}$. Indeed, recall that a_v's second argument is a binary string and so the equality above would have been meaningless otherwise. The above equation implies $h_v(x,y) = h_v(0^{\ell-1}, y-x)$, proving the claim. ■

Given Claim 1.3 we can simplify our notation as follows. Let r denote the root of T. For $x \in \{0,1\}^{\ell-1}$ and $k \in \{1, \dots, \ell\}$ we define the random variables

$$a(x,k) = a_r(0^\ell, 1 \circ x, k),$$
$$h(x) = h_r(0^{\ell-1}, x).$$

Note that $h(x) = \sum_{k=1}^{\ell} a(x,k)$.

Proposition 1.1 There exists a fixing of the sequence $\{R_i\}_i$ such that the function T is a tree code with distance 0.05.

Proof. First note that for every fixing of the sequence $\{R_i\}_i$, T is an online function. Observe that, for a fixing of $\{R_i\}_i$, T is a tree code with distance δ if and only if for every $\ell \geq 1$ and $x \in \{0,1\}^{\ell-1}$ it holds that $h(x) \geq \delta\ell$. Indeed, recall that by definition, T is a tree code with distance δ if and only if for every vertex v in T, $\ell \geq 1$, and for every $x, y \in \{0,1\}^{\ell-1}$ it holds that $h_v(x,y) \geq \delta\ell$. However, by Claim 1.3, $h_v(x,y) = h(y-x)$.

For $x \in \{0,1\}^{\ell-1}$ denote by $E(x)$ the event $h(x) < \delta\ell$. By the above discussion, it suffices to prove, for $\delta = 0.05$, that

$$\mathbf{Pr}\left[\bigcup_{x \in \{0,1\}^{\mathbb{N}}} E(x)\right] < 1.$$

To this end, by the union bound, it suffices to prove that

$$\sum_{x \in \{0,1\}^{\mathbb{N}}} \mathbf{Pr}[E(x)] < 1.$$

Consider any $x \in \{0,1\}^{\ell-1}$ with $\ell \geq 1$. Note that the event $E(x)$ holds if and only if there exists a set $T \subseteq \{1, \dots, \ell\}$ of size $|T| \geq \lceil (1-\delta)\ell \rceil$ such that for every $k \in T$, $a(x,k) = 0$. By taking the union bound over all such sets T, and using that $a(x,1), \dots, a(x,\ell)$ are independent and each is uniformly distributed over $\mathbb{F}_4$ (Claim 1.2), we get

$$\mathbf{Pr}[E(x)] \leq \binom{\ell}{\lceil (1-\delta)\ell \rceil} 4^{-\lceil (1-\delta)\ell \rceil}. \tag{1.7}$$

We thus have that

$$\frac{1}{\ell} \cdot \log_2 \binom{\ell}{\lceil (1-\delta)\ell \rceil} \leq H\left(\frac{\lceil (1-\delta)\ell \rceil}{\ell}\right).$$

As $\delta < \frac{1}{2}$ and since the entropy function H decreases in $[\frac{1}{2}, 1]$ we have that

$$H\left(\frac{\lceil (1-\delta)\ell \rceil}{\ell}\right) \leq H(1-\delta) = H(\delta).$$

Substitute to Equation (1.7), we get that $\mathbf{Pr}[E(x)] \le 2^{(H(\delta)-2(1-\delta))\ell}$. Thus,

$$\sum_{x\in\{0,1\}^{\mathbb{N}}} \mathbf{Pr}[E(x)] \le \sum_{\ell=1}^{\infty} 2^{\ell-1} \cdot 2^{(H(\delta)-2(1-\delta))\ell} = \frac{1}{2}\sum_{\ell=1}^{\infty} 2^{(H(\delta)+2\delta-1)\ell}.$$

One can verify that for $\delta = 0.05$ the above geometric sum is strictly smaller than 1, and the theorem follows. ■

We remark that there is a way of improving the above construction by increasing the distance from 0.05 to 0.136 while using four colors still. It is easy to see that the distance cannot be higher than 0.25 using only four colors.

1.4.3 Explicit tree codes

2. Randomness in computation

2.1 Randomness in algorithms

2.1.0.1 The Polynomial Identity Testing problem (PIT)

Suppose we wish to prove the identity between two polynomial equations we believe are equal, such as the Vandermonde identity

$$\det(x_i^{j-1})_{i,j} = \prod_{i<j}(x_i - x_j).$$

Equivalently, we are interested in figuring out whether a polynomial equation $f(x) \equiv 0$ (namely, $f(x) = 0$ as an element of $\mathbb{R}[x_1,\dots,x_n]$). For a degree d polynomial, expanding all monomials may take $n^{O(d)}$ and is typically infeasible.

Randomness comes to our rescue here because the distinction between $f \equiv 0$ and $f \not\equiv 0$ becomes very apparent geometrically: In the first case all points in $\mathbb{R}^n$ are zeros of f whereas in the second case, the (loosely speaking) algebraic variety f defines has measure 0. We can make this precise using the Schwartz-Zippel lemma, this time, stated over the reals.

Theorem 2.1 Let $f(x_1,\dots,x_n) \in \mathbb{R}[x_1,\dots,x_n]$ be nonzero. Assume that f has degree at most d in each of the variables. Then, if s_i is sampled uniformly at random from $[3d]$, independently for each $i \in [n]$, then

$$\mathbf{Pr}[f(s_1,\dots,s_n) = 0] \leq \frac{1}{3}.$$

Since evaluating f at a point is easy (or as easy as the formula describing it is), we got ourselves a probabilistic algorithm for PIT.

2.1.0.2 Primality and factorization

2.1.1 Does randomness add computational power?

****Todo:** Do space and time & PRGs**

2.2 Does randomness exist?

2.3 Randomness in algorithms

2.3.0.1 The Polynomial Identity Testing problem (PIT)

Suppose we wish to prove the identity between two polynomial equations we believe are equal, such as the Vandermonde identity

$$\det(x_i^{j-1})_{i,j} = \prod_{i<j}(x_i - x_j).$$

Equivalently, we are interested in figuring out whether a polynomial equation $f(x) \equiv 0$ (namely, $f(x) = 0$ as an element of $\mathbb{R}[x_1,\ldots,x_n]$). For a degree d polynomial, expanding all monomials may take $n^{O(d)}$ and is typically infeasible.

Randomness comes to our rescue here because the distinction between $f \equiv 0$ and $f \not\equiv 0$ becomes very apparent geometrically: In the first case all points in $\mathbb{R}^n$ are zeros of f whereas in the second case, the (loosely speaking) algebraic variety f defines has measure 0. We can make this precise using the Schwartz-Zippel lemma, this time, stated over the reals.

Theorem 2.2 Let $f(x_1,\ldots,x_n) \in \mathbb{R}[x_1,\ldots,x_n]$ be nonzero. Assume that f has degree at most d in each of the variables. Then, if s_i is sampled uniformly at random from $[3d]$, independently for each $i \in [n]$, then

$$\mathbf{Pr}[f(s_1,\ldots,s_n) = 0] \le \frac{1}{3}.$$

Since evaluating f at a point is easy (or as easy as the formula describing it is), we got ourselves a probabilistic algorithm for PIT.

2.3.0.2 Primality and factorization

2.3.1 Does randomness add computational power?

****Todo:** Do space and time & PRGs**

2.4 Does randomness exist?

2.5 What is it like to be a pseudorandomnessist?

II Basic Pseudorandom Primitives

3 Bounded independence 29

3.1 Constructing bounded independence distributions 29

3.2 PRGs for decision trees 31

3.3 PRGs for combinatorial rectangles 33

3.4 Bounded independence fools conjunctions . . 36

3.5 Bounded independent fools product of low variance variables 38

3.6 Improved PRGs for combinatorial rectangles . 38

3.7 Constructing Ramsey graphs 38

3. Bounded independence

3.1 Constructing bounded independence distributions

We begin with an algorithmic motivating example.

Definition 3.1 Let $G=(V,E)$ be an undirected graph. Given $S,T \subseteq V$ define

$$\mathrm{cut}(S,T)=\{\{u,v\}\in E \mid u\in S, v\in T\}.$$

Further, let $\mathrm{cut}(S)=\mathrm{cut}(S,V\setminus S)$.

The MaxCut **Problem.** Given a graph $G=(V,E)$ find the largest cut $\mathrm{cut}(S)$.
MaxCut is **NP**-hard (in contrast to MinCut which is in **P**).

Proposition 3.1 Let $G=(V,E)$ be an undirected graph with no self-loops (these cannot be cut). Then, $\mathsf{MaxCut}(G)\geq \frac{|E|}{2}$

Proof. We employ the probabilistic method, where we assign every vertex v to S with probability $\frac{1}{2}$ uniformly and independently across V. More precisely, for $v\in V$ define $\mathbf{X}_v$ to be a random variable which attain the values $0,1$ with equal probability, where the $\mathbf{X}_v$-s are independent. Define the random cut as

$$\mathbf{S}=\{v\in V \mid \mathbf{X}_v=1\}.$$

Then, the cut size is a random variable which we can write as

$$|\mathrm{cut}(\mathbf{S})|=\sum_{uv\in E}\mathbf{1}(\mathbf{X}_v\neq\mathbf{X}_u).$$

Thus, by linearity of expectation,

$$\mathbf{E}[|\mathrm{cut}(\mathbf{S})|]=\sum_{\{u,v\}\in E}\mathbf{E}\left[\mathbf{1}(\mathbf{X}_u\neq\mathbf{X}_v)\right]=\sum_{\{u,v\}\in E}\mathbf{Pr}[\mathbf{X}_u\neq\mathbf{X}_v]=\frac{|E|}{2}.$$

Thus, and this is the key observation underlying the probabilistic method, there exists $S\subseteq V$ such that $|\mathrm{cut}(S)|\geq\frac{|E|}{2}$. ■

The above probabilistic proof suggests an efficient randomized $\frac{1}{2}$-approximation for MaxCut. Can we devise a deterministic algorithm? The unsatisfactory answer is yes in the sense we can go over all points in the probability space underlying **S** and for each check what is the respective cut size. 3.1 guarantees that the average result will be as desired, and thus there is such a desired point. The problem is that the number of points is 2^n, where $n = |V|$ so, really, we will go over all possible cuts.

The key observation is that there is (an efficiently computable) probability space which is much smaller, namely it has a smaller support, and that the proof of 3.1 works just as well if we use that more economical space. To see this, note that the proof of 3.1 goes through for any probability space satisfying: (1) $\mathbf{X}_v$ is unbiased for every $v \in V$; and (2) the random variables $\{\mathbf{X}_v\}_{v\in V}$ are pairwise independent.

3.1.1 Constructing bounded-independent distributions

Here is a simple construction of such a distribution. Let $s = \lceil \log_2(n+1) \rceil$ and let $\mathbf{Y}_1, \dots, \mathbf{Y}_s$ be uniformly and independent random bits. For every $\emptyset \neq T \subseteq [s]$, define

$$\mathbf{Y}_T = \sum_{t \in T} \mathbf{Y}_t,$$

where the summation is over $\mathbb{F}_2$. It is easy to see that $\mathbf{Y}_T$ is unbiased and that that $\mathbf{Y}_T$ and $\mathbf{Y}_R$ are independent for every $T \neq R$. Indeed, it suffices to check that their parity is unbiased (as both of them are). This follows since $\mathbf{Y}_T + \mathbf{Y}_R = \mathbf{Y}_{T \triangle R}$ and $T \triangle R \neq \emptyset$.

Now, if we wish to have n random variables that are pairwise independent, we take n of the $2^s - 1$ variables $\{\mathbf{Y}_T\}_T$, which can be done by our choice of s. Thus, we generated n bits that are pairwise independent using only s "truly" random bits.

****Gil:** relate to inner product**

Going back to the MaxCut problem, our deterministic algorithm will iterate over the sample space (which, note, can be done efficiently) that is composed of $O(n)$ points and for each check if the respective cut is of size at least $\frac{|E|}{2}$.

3.1.2 Pairwise independent hash functions

Many applications requires a sequence of random variables $\mathbf{X}_1, \dots, \mathbf{X}_n$ that are pairwise independent where each $\mathbf{X}_i$ is uniformly distributed on an alphabet $[m]$. The naive reduction to the previous construction of pairwise independent bits will require $O(\log n \cdot \log m)$ random bits. We can do better - $O(\log n + \log m)$.

Before describing the construction, note that the sequence $\mathbf{X}_1, \dots, \mathbf{X}_n$ is supported on $[m]^n$. It is sometimes more convenient to consider the sequence as a function $f : [n] \to [m]$. This alternative view is captured by the following definition.

Definition 3.2 A distribution $\mathscr{H}$ over functions $[n] \to [m]$ is called *pairwise independent* if for every $x \in [n]$, when sampling $h \sim \mathscr{H}$, $h(x)$ is uniform over $[m]$ and for every distinct x, y, $(h(x), h(y))$ is uniform over $[m]^2$.

Another way of saying this is saying that $\mathscr{H}(x)$ is uniform over $[m]$ for every x and that $\mathscr{H}(x)$ and $\mathscr{H}(y)$ are independent for every $x \neq y$.

A word regarding notation: It is typically the case that $\mathscr{H}$ is uniform over a set, also denoted as $\mathscr{H}$, and then one writes $\mathscr{H} = \{h : [n] \to [m]\}$. Even in the more general case, one can think of $\mathscr{H}$ as a multi-set. We refer to the multi-set as a *family of pairwise independent hash functions*.

3.1.3 A construction of a family of pairwise independent hash functions

We will start by constructing a family for the case $n = m$ and where n is a prime (or a prime power). In this case, we can identify $[m]$ with a finite field $\mathbb{F}$. The family is then given by

$$\mathscr{H} = \{h_{a,b} : \mathbb{F} \to \mathbb{F} \mid a, b \in \mathbb{F}\},$$

where $h_{a,b}(x) = ax + b$. To see that this is a family of pairwise independent hash functions, note first that for a fixed $x \in \mathbb{F}$, the random variable $ax + b$, where $a, b \sim \mathbb{F}$ are uniform, is uniform over

$\mathbb{F}$. More interestingly, for $x,y \in \mathbb{F}$ distinct, we claim that $(ax+b, ay+b)$ is uniform over $\mathbb{F}^2$. One way to see this is to note that this pair is given by

$$\begin{pmatrix} x & 1 \\ y & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}.$$

Now, the matrix above has full rank, and so it is a bijection. Thus, when iterating over all pairs (a,b) the output pair is uniform over $\mathbb{F}^2$.

This constructions requires $2\log n$ random bits as apposed to $\log^2 n$ required by the naive use of pairwise independent bits, and to $n \log n$ required for sampling a truly random function.

I leave it as a simple exercise to handle any values of n,m. If these are powers of 2, which conveniently correspond to functions $h : \{0,1\}^{\log_2 n} \to \{0,1\}^{\log_n m}$, then the field arithmetic can be made efficient, then it can be shown that $2 \cdot \max(m,n)$ random bits suffices. This is close to being tight as it can be shown that $\max(m,n)+m$ bits are necessary.

3.1.4 k-independent hash functions

The above construction can be easily extended to give k-wise independent distributions, defined in the natural way. Indeed, the corresponding family is given by

$$\mathscr{H} = \{h_{a_0,\ldots,a_{k-1}} : \mathbb{F} \to \mathbb{F} \mid a_i \in \mathbb{F}\},$$

where

$$h_{a_0,\ldots,a_{k-1}}(x) = \sum_{i=0}^{k-1} a_i x^i.$$

Using the invertability of the Vandermonde matrix, it can be shown that this yields a family of k-wise independent hash functions. The randomness required to sample an element of the family is $k \log \mathbb{F}$, which in general can be made to equal to $k \cdot \max(n,m)$. ****Gil:** write the correct bound**

****Gil:** Discuss k-wise over general dist**

3.2 PRGs for decision trees

Definition 3.3 A function $f : \{0,1\}^n \to \mathbb{R}$ is called a *k-junta* if it depends on at most k of its input variables. That is,

$$f(x_1,\ldots,x_n) = g(x_{i_1},\ldots,g_{i_k}),$$

for some function $g : \{0,1\}^n \to \mathbb{R}$ and $i_1,\ldots,i_k \in [n]$.

A k-wise independent distribution (over bits in this case) can be thought of as a distribution that "looks completely random" to any k-junta. This idea of measuring the randomness of a distribution with respect to a certain class of functions is key in complexity theory.

****Gil:** this should appear before anyhow**

Definition 3.4 Let $f : \{0,1\}^n \to \mathbb{R}$ and $\mathbf{X}$ a probability distribution over $\{0,1\}^n$. We say that $\mathbf{X}$ fools f with error ε, or that ε-fools f if

$$|\mathbf{E}[f(\mathbf{X})] - \mathbf{E}[f(\mathscr{U}_n)]| \le \varepsilon.$$

We say that $\mathbf{X}$ ε-fools a class (a set) of functions if it ε-fools all functions in the class.

As a remark regarding notation, we use $\mathbf{E}[f]$ as a shorthand for $\mathbf{E}[f(\mathscr{U})]$ throughout. With this, we can say that any k-wise independent distribution fools the class of k-juntas. It is very useful to consider the function that produces the distribution fooling a certain class given a (hopefully short) string of truly random bits.

Definition 3.5 Let $f : \{0,1\}^n \to \mathbb{R}$. A function $\mathsf{PRG} : \{0,1\}^s \to \{0,1\}^n$ is called an ε-pseudorandom generator (PRG) for f if $\mathsf{PRG}(\mathcal{U}_s)$ ε-fools f. We naturally extend the definition to a class of functions.

The input to PRG is called a *seed*, and thus the parameter s - the number of truly random bits required by PRG to produce the n pseudorandom bits, is the *seed length*.

k-wise independent distributions are key in the construction of PRGs for stronger computational models than k-juntas. But, in fact, they themselves fool more than just k-juntas.

Definition 3.6 A *decision tree* over $\{0,1\}^n$ is a binary tree where each internal tree is labeled with a variable x_i where its two children correspond to the value taken by x_i. The leaves are labeled by $\{0,1\}$. A decision tree computes a function $f : \{0,1\}^n \to \{0,1\}$ by walking from the root to a leaf according to the values of the variables queried.

Although a depth-k is not necessarily a k-junta, any k-wise independent distributions perfectly fools it. To see this, note that we can write the function f computable by a depth-k decision tree as

$$f(x) = \sum_{\ell} f_\ell(x),$$

where ℓ is a leaf of the tree, labeled by 1, and $f_\ell(x)$ is the k-junta indicating whether we reached ℓ from the root. If $\mathbf{X}$ is a k-wise independent distribution then, by linearity of expectation,

$$\mathbf{E}[f(\mathbf{X})] = \sum_{\ell} \mathbf{E}[f_\ell(\mathbf{X})] = \sum_{\ell} \mathbf{E}[f_\ell] = \mathbf{E}[f].$$

In fact, and this may seem surprising at first, k-independence fools functions that are computable by decision trees of any depth, where the error depends on the size of the tree. To prove this we will use the so-called sandwiching technique.

We say that a function $f : \{0,1\}^n \to \mathbb{R}$ is δ-sandwiched between $f_\ell : \{0,1\}^n \to \mathbb{R}$ and $f_u : \{0,1\}^n \to \mathbb{R}$ if

1. $f_\ell(x) \le f(x) \le f_u(x)$ holds for all $x \in \{0,1\}^n$; and
2. $\mathbf{E}[f_u - f_l] \le \delta$.

The key, easy, observation is that fooling the sandwiching functions f_ℓ, f_u suffices to fool f.

Claim 3.1 Suppose f is δ-sandwiched between f_ℓ, f_u, and assume that $\mathbf{X}$ ε-fools f_ℓ and f_u. Then, $\mathbf{X}$ $(\varepsilon + \delta)$-fools f.

Proof. On the one hand,

$$\mathbf{E}[f(\mathbf{X})] \le \mathbf{E}[f_u(\mathbf{X})] \le \mathbf{E}[f_u] + \varepsilon \le \mathbf{E}[f] + \delta + \varepsilon.$$

On the other hand,

$$\mathbf{E}[f(\mathbf{X})] \ge \mathbf{E}[f_\ell(\mathbf{X})] \ge \mathbf{E}[f_\ell] - \varepsilon \ge \mathbf{E}[f] - \delta - \varepsilon.$$

■

With this we can prove the following.

Proposition 3.2 Every k-wise independent distribution fools the class of size-m decision trees with error $m \cdot 2^{-k}$.

Proof. Given a size-m decision tree computing f, define a pair of decision trees which computes f_ℓ and f_u as follows: For every node in the tree of depth exactly k, transform this node to a leaf by removing its descendants and label the leaf with 0 in f_ℓ and with 1 in f_u.

Clearly, f_ℓ, f_u are depth-k decision trees which are perfectly fooled by any k-wise independent distribution. Thus, to complete the proof it suffices to prove that these $m \cdot 2^{-k}$ sandwich f. Clearly, $f_\ell \le f \le f_u$ holds point-wise. Now, every leaf we introduced to f_ℓ, f_u is reached by the uniform distribution with probability 2^{-k}. The proof follows as there are at most m such new leaves. ■

3.2 gives as an ε-PRG for size-m decision trees with seed length $O(\log n \cdot \log \frac{m}{\varepsilon})$. But, in fact, it shows much more - *every* bounded-independence distribution fools the class of decision trees. There are scenarios in which we cannot choose precisely which distribution is applied, only that it comes from a certain class, and a result such as 3.2 is useful in such cases. Anyhow, we will soon obtain a PRG with an improved seed length of $O(\log \frac{nm}{\varepsilon})$ using small-bias sets.

3.3 PRGs for combinatorial rectangles

Definition 3.7 An *n-dimensional combinatorial rectangle over* $[m]$ is a function $f : [m]^n \to \{0,1\}$ of the form

$$f(x_1,\dots,x_n) = \prod_{i\in[n]} f_i(x_i),$$

where each $f_i \in [m] \to \{0,1\}$.

Note that a combinatorial rectangle gets its name as it servers as the indicator for the set $f_1^{-1}(1) \times \cdots \times f_n^{-1}(1)$. A PRG for combinatorial rectangles can therefore be thought of as a, hopefully, small set of points in $[m]^n$ such that the fraction of points in this set within any combinatorial rectangle approximates the density, or volume, of that combinatorial rectangle.

We turn to construct a PRG for combinatorial rectangles where our strategy will be based on dimension reduction, namely, we will invest some amount of randomness to obtain a combinatorial rectangle in dimension $\frac{n}{2}$ that has roughly the same density as the original combinatorial rectangle which lives in dimension n. We will repeat this process until we get dimension 1.

Dimension reduction via pairwise independence. The error reduction makes use of a family of pairwise independent hash functions for sampling a matching in $[m] \times [m]$. Let $\mathcal{H} : \{h : [m] \to [m]\}$ be a family of pairwise independent hash functions.

The following lemma, which is extremely useful when dealing with pairwise independent hash functions states that for every fixed choice of a dimension-2 combinatorial rectangle, $A \times B \subseteq [m]^2$ it holds that for most choices of h, the fraction of edges of the matching that falls in $A \times B$ is a good approximation for the density of $A \times B$.

Lemma 3.1 — Hash mixing lemma. Let $\mathcal{H} : \{h : [m] \to [m]\}$ be a family of pairwise independent hash functions. Let $A, B \subseteq [m]$, with densities $\alpha = \frac{|A|}{m}$ and $\beta = \frac{|B|}{m}$. Then,

$$\Pr_{h\sim\mathcal{H}}\left[\left|\Pr_{x\in[m]}[x \in A \text{ and } h(x) \in B] - \alpha\beta\right| \geq \varepsilon\right] \leq \frac{\alpha\beta}{\varepsilon^2 m}.$$

For the proof of 3.1 we prove the following.

Lemma 3.2 Let $\mathbf{X}_1,\dots,\mathbf{X}_n$ be a sequence of pairwise independent random variables supported on $\{0,1\}$, each with expectation p. Then,

$$\sigma^2 = \mathrm{Var}(\mathbf{X}) = np(1-p).$$

Moreover, for every $c \geq 1$,

$$\mathbf{Pr}[|\mathbf{X} - \mathbf{E}[\mathbf{X}]| \geq c\sigma] \leq \frac{1}{c^2}.$$

Proof of 3.2. We start by computing the expectation,

$$\mathbf{E}[\mathbf{X}] = \sum_{i=1}^{n} \mathbf{E}[\mathbf{X}_i] = pn.$$

Now, using that $\mathbf{X}_i \in \{0,1\}$ and that the $\mathbf{X}_i$-s are pairwise independent,

$$\mathbf{E}[\mathbf{X}^2] = \sum_{i,j=1}^{n} \mathbf{E}[X_iX_j] = \sum_{i=1}^{n} \mathbf{E}[X_i] + \sum_{i\neq j} \mathbf{E}[\mathbf{X}_i]\,\mathbf{E}[X_j] = pn + n(n-1)p^2.$$

Thus,

$$\mathrm{Var}(\mathbf{X}) = \mathbf{E}[\mathbf{X}^2] - (\mathbf{E}[X])^2 = np(1-p).$$

The moreover part, readily follows by Chebyshev's inequality. ■

Proof of 3.1. For $h \in \mathscr{H}$ let

$$S_h = \{x \in A \mid h(x) \in B\},$$

and let $s_h = \frac{|S_h|}{m}$. With this notation, we want to prove that

$$\Pr_{h \sim \mathscr{H}}[|s_h - \alpha\beta| \geq \varepsilon] \leq \frac{\alpha\beta}{\varepsilon^2 m}.$$

For $x \in A$, define $I_h(x)$ to be the indicator to the event $h(x) \in B$. Then,

$$S_h = \sum_{x \in A} I_h(x).$$

As the random variables $\{I_h(x) \mid x \in A\}$ are pairwise independent bits with a common marginal expectation of $p = \mathbf{E}[I_h(x)] = \beta$, we have by 3.2 that

$$\mathrm{Var}(S_h) = |A|\beta(1-\beta)$$

and that for every $c \geq 1$,

$$\Pr_h[|S_h - |A|\beta| > c\sigma] \leq \frac{1}{c^2}.$$

Dividing by m, we get

$$\Pr_h\left[|s_h - \alpha\beta| > \frac{c\sigma}{m}\right] \leq \frac{1}{c^2}.$$

Hence, by choosing c so that $\varepsilon = \frac{c\sigma}{m}$, we get that

$$\Pr_h[|s_h - \alpha\beta| > \varepsilon] \leq \frac{\alpha\beta(1-\beta)}{\varepsilon^2 m} \leq \frac{\alpha\beta}{\varepsilon^2 m}.$$

■

The construction. Let $\mathscr{H} : \{h : [m] \to [m]\}$ be a family of pairwise independent hash functions. We define a sequence of PRGs: for $j \in \mathbb{N}$,

$$\mathsf{PRG}_j : \mathscr{H}^j \times [m] \to [m]^{2^j}.$$

To fool an n-dimensional combinatorial rectangle we can take $j = \lceil \log_2 n \rceil$ and discards to last $2^j - n$ symbols. The base PRG, $\mathsf{PRG}_0 : [m] \to [m]$, is defined to be the identity function which is clearly a 0-error PRG for dimension 1 combinatorial rectangles. For $j \geq 1$, we define PRG_j recursively by

$$\mathsf{PRG}_j(h_1,\ldots,h_j,x) = \mathsf{PRG}_{j-1}(h_1,\ldots,h_{j-1},x) \circ \mathsf{PRG}_{j-1}(h_1,\ldots,h_{j-1},h_j(x)).$$

So, for example, if we open the recursion we see that

$$\begin{aligned} \mathsf{PRG}_1(h_1,x) &= (x, h_1(x)), \\ \mathsf{PRG}_2(h_1,h_2,x) &= (x, h_1(x), h_2(x), h_1(h_2(x))). \end{aligned}$$

Claim 3.2 For every $\varepsilon > 0$ and for every combinatorial rectangle $A \subseteq [m]^{2^j}$,

$$\Pr_{h_1,\ldots,h_j}\left[\left|\Pr_{x \sim [m]}[\mathsf{PRG}_j(h_1,\ldots,h_j,x) \in A] - \mu(A)\right| \geq 2^{j+1}\varepsilon\right] \leq \frac{2^j}{\varepsilon^2 m}.$$

Proof. Fix j.

For every odd $i \in [n]$ and for $h \in \mathscr{H}$ define

$$A_{i,i+1}(h) = \{x \in [m] \mid (x, h(x)) \in A_i \times A_{i+1}\}.$$

Fix $\varepsilon > 0$. 3.1 readily implies that for every odd $i \in [2^j]$,

$$\mathop{\mathbf{Pr}}_{h\sim\mathscr{H}}[|\mu(A_{i,i+1}(h)) - \mu(A_i \times A_{i-1})| \geq \varepsilon] \leq \frac{1}{\varepsilon^2 m}.$$

We say that h is *ε-good* for i if the above event does not hold, namely,

$$|\mu(A_{i,i+1}(h)) - \mu(A_i \times A_{i-1})| < \varepsilon$$

Define the 2^{j-1} combinatorial rectangle

$$A(h) = A_{1,2}(h) \times A_{3,4}(h) \times \cdots \times A_{2^j-1,2^j}(h)$$

We claim that

$$\mathop{\mathbf{Pr}}_{h\sim\mathscr{H}}\left[|\mu(A(h)) - \mu(A)| \geq 2^j\varepsilon\right] \leq \frac{2^{j-1}}{\varepsilon^2 m}.$$

To see this, consider the event that h is simultaneously good for all odd $i \in [2^j]$. By the union bound, This event occurs except with probability $\frac{2^{j-1}}{\varepsilon^2 m}$ over $h \sim \mathscr{H}$. Conditioned on that event, we wish to prove that

$$|\mu(A(h)) - \mu(A)| \leq 2^j\varepsilon.$$

This follows by a straightforward induction, using that fact that if a, b are two numbers in $[0,1]$ where $|a-\alpha| \leq \delta$ and $|b-\beta| \leq \delta$ then $|ab-\alpha\beta| \leq 2\delta$. Indeed,

$$\begin{aligned}|ab-\alpha\beta| &\leq |ab-\alpha b| + |\alpha b - \alpha\beta| \\ &= b|a-\alpha| + \alpha|b-\beta| \\ &\leq 2\delta.\end{aligned}$$

The observation to be made here is that the sequence of PRGs were defined to facilitate dimension reduction so to speak. Namely, h_1 is used to reduce the 2^j-dimensional rectangle A to the 2^{j-1}-dimensional rectangle $A(h_1)$. We proved above that with high probability over $h_1 \sim \mathscr{H}$, $A(h_1)$ is a good proxy for A in terms of density.

Similarly, using h_2 we obtain except with probability $\frac{2^{j-2}}{\varepsilon^2 m}$ a 2^{j-2}-dimensional rectangle $A(h_1, h_2)$ which is a good proxy for $A(h_1)$, hence for A. More precisely, we first condition on the event that h_1 is ε-good for A. Now, for each such ε-good h_1 we condition on the event that h_2 is ε-good for $A(h_1)$. If both these events hold then

$$\begin{aligned}|\mu(A(h_1,h_2)) - \mu(A)| &\leq |\mu(A(h_1,h_2)) - \mu(A(h_1))| + |\mu(A(h_1)) - \mu(A)| \\ &\leq (2^j + 2^{j-1})\varepsilon.\end{aligned}$$

Extending this argument, by induction, we get that except with probability

Repeating this, we see that except with probability

$$\sum_{i=0}^{j-1} \frac{2^i}{\varepsilon^2 m} \leq \frac{2^j}{\varepsilon^2 m},$$

over $h_1, \ldots, h_j$, we have that

$$|\mu(A(h_1,\ldots,h_j)) - \mu(A)| \leq \sum_{i=1}^{j} 2^i\varepsilon \leq 2^{j+1}\varepsilon.$$

But

$$\mu(A(\vec{h}_j)) = \mathop{\mathbf{E}}_{x\sim[m]} \mathsf{PRG}_j(\vec{h}_j, x),$$

which completes the proof. ■

Corollary 3.1 For every n, ε and m, provided that

$$m \geq \frac{256n^3}{\varepsilon^3},$$

$\mathsf{PRG}_{\lceil \log_2 n \rceil}$, restricted to its first n output symbols, is an ε-PRG for combinatorial rectangles over $[m]$ in dimension n. Its seed length is

$$2 \log m \cdot \log n.$$

The lower bound required for m is not really an issue. Fooling combinatorial rectangles over a smaller m is easier. Having said that, regardless of the value of m, the seed length will remain

$$O\left(\log n \cdot \log \frac{n}{\varepsilon}\right).$$

3.3.0.1 A probabilistic proof for the existence of a PRG

Can we do better? The answer we will give now can be best characterized as "probably yes; just try harder". The fantastic fact is that we can show that a PRG with shorter seed exists, but the proof will give no clue as to how to find that PRG. More precisely, the proof will not say anything about the efficiency of the PRG it guarantees exists.

Let $r_1, \dots, r_{2^s}$ be 2^s uniformly and independent strings in $[m]^n$. For a fixed choice of such sequence r, you can think of the $\mathsf{PRG}_r : \{0,1\}^s \to [m]^n$ which is defined by $\mathsf{PRG}_r(i) = r_i$, where we identify $[2^s]$ with $\{0,1\}^s$. So, really, we are considering the uniform distribution over functions of the form $\{0,1\}^s \to [m]^n$.

Fix a combinatorial rectangle $A \subseteq [m]^n$ and denote its density by α. Let I_i be the indicator for the event $r_i \in A$. Then, $\mathbf{E}[I_i] = \alpha$. As the r_i-s are independent, by the Chernoff bound,

$$\mathbf{Pr}\left[\left|\frac{1}{2^r}\sum_i I_i - \alpha\right| > \varepsilon\right] \leq 2^{-c\varepsilon^2 2^r},$$

for some universal constant $c > 0$ which I can never seem to remember. But note that $\frac{1}{2^r}\sum_i I_i$ is precisely $\mathbf{Pr}[\mathsf{PRG}_r(\mathscr{U}) \in A]$. Thus, we conclude that expect with probability $2^{-c\varepsilon^2 2^r}$, a random function sampled from our distribution is an ε-PRG for a particular choice of A. Now, taking the union bound over all A, we see that

$$\mathop{\mathbf{Pr}}_r[\mathsf{PRG}_r \text{ is not an } \varepsilon\text{-PRG for combinatorial rectangles}] \leq 2^{mn} \cdot 2^{-c\varepsilon^2 2^r}.$$

Thus, if we take r so that the about bound is strictly smaller than 1, namely,

$$r = \log n + \log m + 2\log\frac{1}{\varepsilon} + O(1),$$

then there must exists an ε-PRG for combinatorial rectangles.

In fact, at the same asymptotic price we prove that only 2^{-mn} fraction of functions are not ε-PRGs.

3.4 Bounded independence fools conjunctions

Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be random variables on the set $\{0,1\}$. This section investigates the properties of the conjunction of the events $\bigwedge_i(\mathbf{X}_i = 1)$ under two different sampling scenarios: first, when the $\mathbf{X}_i$-s are sampled independently, and second, when they are sampled in a k-wise independent manner. We prove

Lemma 3.3 Let $\mathbf{X}_1,\dots,\mathbf{X}_n$ be random variables on the set $\{0,1\}$, and denote $p_i = \mathbf{E}[\mathbf{X}_i]$. Let $\mathscr{D}$ be a k-wise independent distribution. Then,

$$\left| \Pr_{\mathscr{D}}\left[\bigwedge_{i=1}^{n}(\mathbf{X}_i = 1)\right] - \Pr_{\mathscr{U}}\left[\bigwedge_{i=1}^{n}(\mathbf{X}_i = 1)\right]\right| \le 2^{-\Omega(k)}.$$

Proof. For simplicity, we consider the case where all marginal expectations $\mathbf{E}[\mathbf{X}_i]$ are equal. Their joint value is denoted $1-p$. There is a reason for this perhaps weird looking choice of notation, and it will come in the proof. However, at an informal level, in the case of the uniform distribution, the probability of the conjunction of the events $\bigwedge_i(\mathbf{X}_i = 1)$ to hold is tiny unless the common value $\mathbf{E}[\mathbf{X}_i] \approx 1$. Quantitatively, things are getting interesting only when $\mathbf{E}[\mathbf{X}_i] \approx 1 - O(\frac{1}{n})$.

The proof can be adjusted to handle the general case. The proof makes use of the inclusion-exclusion principle. For $j \in \{1,\dots,n\}$ let

$$T_j = \sum_{\substack{I\subseteq[n]\\|I|=j}} \Pr[\forall i \in I \quad \mathbf{X}_i = 0].$$

Then, by the inclusion-exclusion principle,

$$\Pr_{\mathscr{U}}\left[\bigwedge_{i=1}^{n}(\mathbf{X}_i = 1)\right] = \sum_{j=0}^{n}(-1)^j T_j,$$

where we set $T_0 = 1$.

Furthermore, if we denote $S_j = \sum_{i=0}^{j}(-1)^i T_i$, it is well-known that for an even j,

$$\Pr_{\mathscr{U}}\left[\bigwedge_{i=1}^{n}(\mathbf{X}_i = 1)\right] \le S_j,$$

whereas for an odd j,

$$\Pr_{\mathscr{U}}\left[\bigwedge_{i=1}^{n}(\mathbf{X}_i = 1)\right] \ge S_j.$$

This, together with the fact that for all $j \le k$, S_j remains unchanged if we replace $\mathscr{U}$ with $\mathscr{D}$, we have that

$$\left| \Pr_{\mathscr{D}}\left[\bigwedge_{i=1}^{n}(\mathbf{X}_i = 1)\right] - \Pr_{\mathscr{U}}\left[\bigwedge_{i=1}^{n}(\mathbf{X}_i = 1)\right]\right| \le T_k$$

if k is even (for an odd k, the bound is T_{k-1} but we ignore this technical detail).

Now,

$$T_k \le \binom{n}{k} p^k \le \left(\frac{nep}{k}\right)^k.$$

Thus, if $np \le \frac{k}{2e}$ we obtain a bound of 2^{-k} and we are done. Assume than that $np > \frac{k}{2e}$ and let n' be an integer such that

$$\frac{k}{2e} - 1 \le n'p \le \frac{k}{2e}.$$

Now,

$$\Pr_{\mathscr{U}}\left[\bigwedge_{i=1}^{n'}(\mathbf{X}_i = 1)\right] = \prod_{i=1}^{n'} \Pr_{\mathscr{U}}[\mathbf{X}_i = 1] = (1-p)^{n'} \le e^{-pn'} = e^{-\frac{k}{2e}+1}.$$

Moreover, by the same argument used above, and since $n'p \le \frac{k}{2e}$,

$$\left| \Pr_{\mathscr{D}}\left[\bigwedge_{i=1}^{n'}(\mathbf{X}_i = 1)\right] - \Pr_{\mathscr{U}}\left[\bigwedge_{i=1}^{n'}(\mathbf{X}_i = 1)\right]\right| \le 2^{-k}.$$

Thus,

$$\Pr_{\mathscr{D}}\left[\bigwedge_{i=1}^{n}(\mathbf{X}_i = 1)\right] \leq \Pr_{\mathscr{D}}\left[\bigwedge_{i=1}^{n'}(\mathbf{X}_i = 1)\right] \leq e^{-\frac{k}{2e}+1} + 2^{-k} = 2^{-\Omega(k)}.$$

■

3.5 Bounded independent fools product of low variance variables

3.6 Improved PRGs for combinatorial rectangles

3.7 Constructing Ramsey graphs