18.619 — Discrete Probability and Stochastic Processes

Class by Kuikui Liu

Notes by Sanjana Das

Spring 2025

Lecture notes from the MIT class **18.619** (Discrete Probability and Stochastic Processes), taught by Kuikui Liu. All errors are my own.

Contents

1 February 3, 2025 — Introduction **7**
- 1.1 Logistics . . . 7
- 1.2 Overview . . . 7
- 1.3 Percolation . . . 8
- 1.4 Triangles . . . 8
- 1.5 Connectivity . . . 10
  - 1.5.1 The approach . . . 10
  - 1.5.2 Some intuition . . . 11
  - 1.5.3 The case $c > 1$ . . . 12
  - 1.5.4 The case $c < 1$ . . . 13

2 February 5, 2025 — Second moment method and branching processes **14**
- 2.1 The Paley–Zygmund inequality . . . 15
- 2.2 Galton–Watson branching processes . . . 16
  - 2.2.1 First and second moments . . . 16
  - 2.2.2 Extinction . . . 17
- 2.3 Percolation on a tree . . . 20
- 2.4 Erdős–Rényi and branching processes . . . 22

3 February 10, 2025 — Hypothesis testing and the second moment method **22**
- 3.1 Hypothesis testing . . . 23
- 3.2 Broadcast processes . . . 23
- 3.3 Phase transition for reconstruction . . . 24
  - 3.3.1 Branching processes perspective . . . 25
  - 3.3.2 Reinterpreting b^* . . . 27
  - 3.3.3 Proof of the reconstructible case . . . 28

4 February 12, 2025 — Lovász Local Lemma **30**
- 4.1 Boolean satisfiability . . . 30
- 4.2 Approximate assignments for SAT . . . 32

4.3 SAT with additional structure . . . 32
4.4 A proof attempt . . . 33
4.5 The Lovász local lemma . . . 33
4.6 Application to k-uniform CNFs . . . 34
4.7 Proof of the Lovász local lemma . . . 35
4.8 Second moment bound on total variation distance . . . 37

5 February 18, 2025 — Shannon's noisy coding theorem 39
5.1 Error-correcting codes . . . 39
5.2 Shannon's theorem . . . 40
5.3 Preliminaries . . . 41
5.4 The large-rate case . . . 41
5.5 The low-rate case . . . 44

6 February 19, 2025 — Chernoff bounds 47
6.1 A motivataing example — MaxCut . . . 47
6.2 A heuristic — sums of i.i.d. random variables . . . 48
6.3 The Chernoff–Hoeffding bound . . . 49
6.4 Back to MaxCut . . . 50
6.5 Moments and the moment generating function . . . 50
6.6 Proof of Chernoff–Hoeffding . . . 51
6.7 Proof of Hoeffding's lemma . . . 52
6.8 The Berry–Esseen theorem and sharpness of Chernoff bounds . . . 54
6.9 Proof of Claim 6.9 . . . 55

7 February 24, 2025 — Sub-Gaussian random variables 55
7.1 Sub-Gaussian random variables . . . 55
7.2 Concentratio bounds for sub-Gaussians . . . 56
7.3 Random matrices . . . 57
7.4 Operator norm of random matrices . . . 59
7.4.1 Discretizing the sphere . . . 59
7.5 Proof of Claim 7.15 . . . 60
7.6 Proof of Claim 7.16 — constructing a δ-net . . . 61

8 February 26, 2025 — Sub-Exponential random variables 63
8.1 Examples of non-sub-Gaussian random variables . . . 63
8.2 Sub-exponential random variables . . . 64
8.3 Bernstein's inequality . . . 65
8.4 Concentration for norms of random vectors . . . 67
8.5 Poisson limit of sparse binomial distributions . . . 68
8.5.1 Coupling . . . 68
8.5.2 Coupling and total variation distance . . . 70
8.5.3 Proof of Theorem 8.13 . . . 71
8.5.4 Equality case for the coupling lemma . . . 71

9 March 3, 2025 72
9.1 Martingales . . . 72
9.2 Some examples . . . 72
9.2.1 Doob martingales . . . 73
9.3 Expectation and variance of martingales . . . 75
9.4 The Azuma–Hoeffding inequality . . . 76
9.5 The bounded differences inequality . . . 77

9.6 Chromatic number of a random graph . . . 78
9.6.1 A lower bound on the expectation . . . 79
9.6.2 Concentration of the chromatic number . . . 80

10 March 5, 2025 **82**
10.1 Stopping times . . . 82
10.2 The optional stopping theorem . . . 83
10.2.1 Proof of the optional stopping theorem . . . 84
10.3 Random walk . . . 86
10.4 Supermartingales and submartingales . . . 88
10.5 A local search algorithm for 2SAT . . . 89

11 March 10, 2025 — Euclidean traveling salesperson problem **92**
11.1 The Euclidean TSP problem . . . 92
11.2 Estimating the expectation . . . 93
11.2.1 The lower bound . . . 94
11.2.2 The upper bound . . . 95
11.2.3 Proof of Proposition 11.5 . . . 97
11.3 Concentration — a first attempt . . . 98
11.4 A more refined bound . . . 99
11.4.1 Better bounds on the increments . . . 100

12 March 12, 2025 — Optimization over Erdős–Rényi **101**
12.1 Clique and independent set . . . 101
12.2 Concentration of the clique number . . . 102
12.3 A greedy algorithm . . . 102
12.3.1 The expected behavior . . . 103
12.3.2 Formalizing the intuition . . . 104
12.3.3 The upper bound . . . 105
12.4 Independent sets in sparse graphs . . . 105
12.5 Proof of Theorem 12.9 . . . 106
12.5.1 First moments . . . 106
12.5.2 Second moments . . . 107
12.5.3 Frieze's idea . . . 108
12.6 Analysis of the greedy algorithm . . . 110

13 March 17, 2025 — Janson's lower tail inequality and positive correlations **111**
13.1 Motivating problem . . . 111
13.2 Positive correlations . . . 111
13.3 Janson's inequality . . . 112
13.3.1 Application — triangle counts . . . 113
13.3.2 Proof of Janson's inequality . . . 114
13.4 Stochastic domination . . . 117

14 March 19, 2025 — Giant component in Erdős–Rényi **119**
14.1 Stochastic domination . . . 119
14.2 The phase transition . . . 120
14.3 Branching process intuition . . . 121
14.4 Exploring a component . . . 121
14.5 Some observations . . . 123
14.6 Comparing the exploration processes . . . 123
14.7 The subcritical case . . . 124

14.8 The supercritical case . . . 124
14.8.1 First moment for Proposition 14.13 . . . 125
14.8.2 Existence and uniqueness of the giant component . . . 126
14.8.3 Proof sketch of Proposition 14.12 . . . 126
14.8.4 Proof of Lemma 14.15 . . . 127

15 March 31, 2025 — Contiguous families of distributions 127
15.1 Contiguity . . . 127
15.2 Average-case computational complexity . . . 129
15.3 Typical proper colorings . . . 130
15.4 Random d-regular graphs and the configuration model . . . 132
15.5 Cycles in the configuration model . . . 134
15.5.1 Proof sketch of Theorem 15.17 . . . 135

16 April 2, 2025 — Gibbs distributions of graphical models 136
16.1 Definitions . . . 137
16.2 Some examples . . . 137
16.3 Marginal and conditional distributions . . . 139
16.4 Phase transitions . . . 140
16.5 Correlation decay and spatial mixing . . . 142
16.6 Spatial mixing for the hardcore model . . . 143
16.6.1 Further results . . . 144
16.6.2 Proof sketch of Theorem 16.16 . . . 144

17 April 7, 2025 145
17.1 Markov chains . . . 146
17.2 Some examples . . . 146
17.3 Stationary distributions . . . 147
17.3.1 Existence . . . 148
17.3.2 Uniqueness and convergence . . . 149
17.4 Markov Chain Monte Carlo . . . 150
17.5 Reversibility . . . 151
17.6 Examples of big Markov chains . . . 152
17.6.1 The swap chain . . . 152
17.6.2 Glauber dynamics . . . 153

18 April 9, 2025 — The Markov Chain Monte Carlo paradigm 154
18.1 Metropolis–Hastings algorithm . . . 155
18.1.1 Example: Hard spheres model . . . 156
18.2 Markov chains via statistical inference . . . 157
18.2.1 Example: Glauber dynamics . . . 158
18.2.2 The Swendsen–Wang Markov chain . . . 159
18.3 Proof of the fundamental theorem of Markov chains . . . 162
18.3.1 Couplings of Markov chains . . . 162

19 April 14, 2025 — Markov chain mixing times 163
19.1 Markovian couplings . . . 163
19.1.1 An example — walks on the hypercube . . . 164
19.2 Proof of the fundamental theorem . . . 164
19.3 Mixing times . . . 167
19.4 Mixing times for the random walk on the hypercube . . . 168

19.5 Debrushin influence . . . 169
19.5.1 Some examples . . . 170

20 April 16, 2025 — Spectral gaps and the conductance method **172**
20.1 Issues with total variation distance . . . 172
20.2 The χ^2 divergence . . . 173
20.3 Linear algebraic setup . . . 174
20.4 Mixing times and the spectral gap . . . 175
20.5 The Dirichlet form . . . 176
20.6 Conductance . . . 177
20.7 Slow mixing for the Curie–Weiss model . . . 179

21 April 23, 2025 **181**
21.1 More about the spectral gap . . . 181
21.2 Intuition for spectral gap vs. mixing . . . 183
21.3 Proof of Theorem 21.2 . . . 183
21.4 Concentration via mixing/isoperimetry . . . 185
21.5 Gausian concentration inequality . . . 186
21.6 Dobrushin's condition to concentration . . . 189

22 April 28, 2025 — Probability and the geometry of polynomials **190**
22.1 Real-rootedness . . . 190
22.2 An application — the monomer-dimer model . . . 192
22.3 Lack of phase transitions . . . 193
22.4 Proof of the Heilmann–Lieb theorem . . . 194
22.4.1 Some simplifications . . . 194
22.4.2 Decomposing the matching polynomial . . . 195
22.4.3 Interlacing . . . 195
22.4.4 The induction . . . 197

23 April 30, 2025 — Random spanning trees **198**
23.1 Kirchoff's matrix tree theorem . . . 199
23.2 Negative correlation for spanning trees . . . 203
23.3 Real stability . . . 204

24 May 5, 2025 **207**
24.1 Introduction to percolation . . . 207
24.2 The model . . . 208
24.3 Infinite components . . . 208
24.4 The lower bound . . . 209
24.5 Closed circuits in the dual . . . 210
24.6 Proof of Theorem 24.10 . . . 213
24.6.1 Step 1 — reduction to $2n \times 3n$ . . . 213
24.6.2 Step 2 — proof for $2n \times 3n$ . . . 214
24.7 Forecast . . . 215

25 May 7, 2025 **216**
25.1 Setup . . . 216
25.2 The finite size criterion . . . 216
25.3 The plan . . . 218
25.4 Influence . . . 219
25.5 Edge isoperimetry . . . 221

25.6 Bounding the derivative . . . 222

26 May 12, 2025 — Local sampling algorithms **223**
26.1 Local sampling . . . 224
26.2 Reduction to a single vertex . . . 225
26.3 Approximate sampling . . . 225
26.4 Perfect sampling . . . 227
26.4.1 A recursive approach . . . 228
26.4.2 A first attempt . . . 228
26.4.3 Stopping the recursion . . . 228
26.4.4 Runtime . . . 230
26.4.5 Fake argument of correctness . . . 231
26.4.6 The actual argument of correctness . . . 231
26.5 Conclusion . . . 232

§1 February 3, 2025 — Introduction

§1.1 Logistics

The syllabus is on Canvas, and will be updated with OH times and locations. There's also a Piazza and pset partners. If we have any questions or trouble hearing or seeing the board, we can shout them out.

§1.2 Overview

This course is about stochastic processes, which is really a fancy name for a large collection of random variables $\mathcal{X} = \{X_i\}_{i\in\mathcal{I}}$ (for some index set $\mathcal{I}$). We'll discuss tools for how to analyze such processes. This is a foundations course, and these fundamentals are going to rest upon three main pillars.

The first is *models* — we'll see many different examples of random processes that we'd like to study. Some classic examples include a large collection of independently flipped coins. We'll also look at much more interesting classic examples, maybe ones where the index set has some kind of meaning — the edges of a graph or the entries of a matrix — and the choice of index set is going to influence the types of questions we'll ask. We'll also look at situations where we don't have independence.

The second pillar is about *questions*, lots of which are motivated by various important fields of study. We'll have questions originating from statistics. For example:

> **Question 1.1.** Suppose we have *samples* of $\mathcal{X}$ (so we have samples of our random variables from some high-dimensional distribution μ). What properties of this distribution μ can we infer from these samples?

This falls under the umbrella of statistical inference.

We'll also see questions motivated by computer science.

> **Question 1.2.** Imagine the index set corresponds to the edges of a graph. Can we compute a Hamiltonian cycle of this graph, or a covering? More generally, can we solve various algorithmic tasks on *random* instances better than the worst case?

A third class of questions is motivated by physics.

> **Question 1.3.** Suppose we have a parametrized family of stochastic processes $\mathcal{X}_p$ (indexed by some parameter p). How do the properties of our processes change as we vary p?

We'll see lots of interesting phenomena — for instance, the presence of *phase transitions*. In lots of the models we'll look at, we'll find there's a critical value of p such that if we vary p around this critical value, the properties of the processes can change very drastically.

Finally, the third pillar we'll look at are *techniques*, as to how we try to answer these questions. They'll sort of fall into four main themes.

- The probabilistic method and moment methods.
- Concentration of measure (which we may have seen a little of before).
- Ways to *compare* different stochastic processes.
- Algorithmic methods.

This is a very brief overview of what we'll look at in this course.

As a brief disclaimer, in lots of the models we'll look at, most of the time we won't have the luxury of closed-form formulas — lots of our results will be phrased as estimates or asymptotics.

§1.3 Percolation

With this overview, let's begin with perhaps one of the most fundamental examples of a stochastic process, called *percolation*.

Definition 1.4. Let $G = (V, E)$ be a (finite, undirected, and simple) graph, and let $p \in [0, 1]$. The percolation on G is a random subgraph $H = (V, F)$ where $\mathbb{P}[F] = p^{|F|}(1-p)^{|E \setminus F|}$ for all $F \subseteq E$.

So given G and p, a percolation on G is a random subgraph on the same vertex set where we take each edge of G, and include it in H independently with probability p.

Definition 1.5. The Erdős–Rényi random graph, denoted $\mathcal{G}(n, p)$, is the case $G = K_n$.

So here we have all possible edges in G (and we include each in our random graph with probability p). (Here n is the number of vertices.)

As a warmup:

Fact 1.6 — We have $\mathbb{E}[\#\text{edges in } \mathcal{G}(n,p)] = p\binom{n}{2}$.

This is because we have $\binom{n}{2}$ possible edges to include, and each is included with probability p (so you can use linearity of expectation).

In particular, if p is a constant independent of n, then we expect to see a very dense graph. We'll be interested in various regimes of p:

- When p is a constant independent of n — this is called the *dense* case.
- We'll also be interested in sparser regimes, where p depends on n; for example, we might have $p_n = \Theta(n^{-\alpha})$ for some $\alpha \in (0, 1)$.
- We might also have $p_n = (\log n)/n$.
- We might also have $p_n = \Theta(1/n)$.

§1.4 Triangles

We'll start by looking at local structures in these graphs; specifically, we'll look at triangles.

Definition 1.7. A triangle is a set of three distinct vertices $\{u, v, w\}$ such that $uv, vw, uw \in E$.

Lemma 1.8

Let T_G denote the number of triangles. Then $\mathbb{E}[T_G] = p^3\binom{n}{3} = (1 + o(1))p^3n^3/6$.

This is because there's $\binom{n}{3}$ possible triples, and we need to ensure every one of these three edges is in the graph (and each has probability p).

Now let's look at the probability of *deviating* from this count.

Proposition 1.9

For all fixed $0 < p \leq 1$ (i.e., p is a constant independent of n), we have

$$\mathbb{P}\left[\left|\frac{T_G}{p^3n^3/6} - 1\right| > \varepsilon\right] \to 0 \quad \text{as } n \to \infty.$$

So here we're looking at the probability that the number of triangles deviates from its expectation. Not only is the *expected* number of triangles roughly $p^3n^3/6$, but actually with very high probability we'll see close to that number of triangles.

Proof. We used the first moment before when computing the expectation. For this deviation bound, we'll use the *second* moment — we'll bound the variance of this random variable and apply Chebyshev's inequality. So the plan is to compute $\mathrm{Var}[T_G]$, and then use Chebyshev.

To do so, we set up some indicator variables — let $\mathcal{I}_{uvw}$ be the indicator that uvw forms a triangle. In particular, we have $T_G = \sum_{uvw} \mathcal{I}_{uvw}$. We want to compute the variance of this guy, so let's start by computing $\mathbb{E}[T_G^2]$. For this, we can plug this sum in and expand everything out, and we get

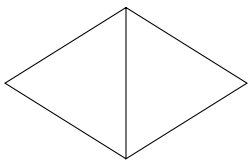
$$\mathbb{E}[T_G^2] = \sum_{uvw,xyz} \mathbb{E}[\mathcal{I}_{uvw}\mathcal{I}_{xyz}] = \sum_{uvw,xyz} \mathbb{P}[uvw \text{ and } xyz \text{ are both triangles}].$$

Now crucially, the variables in this large collection are not at all independent — they depend on how many vertices are in common between these two triples. So let's classify all such cases. For convenience, we'll call

$$p(uvw, xyz) = \mathbb{P}[uvw \text{ and } xyz \text{ are both triangles}].$$

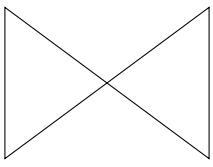
Case 1 ($uvw = xyz$). Then these two random variables $\mathcal{I}_{uvw}$ and $\mathcal{I}_{xyz}$ are actually the same, and $p(uvw, xyz) = p^3$. And as before, we have $O(n^3)$ such pairs of triples. (We choose the first triple, and then the second one is determined.)

Case 2 (uvw and xyz share two vertices, i.e., $|uvw \cap xyz| = 2$). As a picture, the kind of structures we're looking for are like this:



From this picture, it's pretty clear that $p(uvw, xyz) = p^5$ — we have five edges. We can also count the number of such structures; it looks something like $O(n^4)$. (You can also write an explicit formula, but we're not going to do that.)

Case 3 ($|uvw \cap xyz| = 1$). Then the picture looks like this:



This structure has 6 edges, so $p(uvw, xyz) = p^6$. And there's going to be $O(n^5)$ such structures.

Case 4 ($uvw \cap xyz = \emptyset$). In this case, the picture looks like two disjoint triangles. The corresponding probability will be $p(uvw, xyz) = p^6$; and now we have $\binom{n}{3}\binom{n-3}{3}$ such things. This will be the dominant term in our computation for $\mathbb{E}[T_G^2]$ — we're thinking of p as a constant. (That's why we need the actual formula for this.)

Now we have

$$\mathrm{Var}[T_G] = \mathbb{E}[T_G^2] - \mathbb{E}[T_G]^2 = p^6 \binom{n}{3}\binom{n-3}{3} + O(n^5) - p^6 \binom{n}{3}^2.$$

And the dominant n^6 terms are going to cancel, so this whole thing is going to be $O(n^5)$ (the constant may depend on p, but it's some constant independent of n).

So now we've computed our variance, and we can use our beloved Chebyshev inequality — this says

$$\mathbb{P}\left[|T_G - \mathbb{E}[T_G]| > t\right] \leq \frac{\mathrm{Var}[T_G]}{t^2}.$$

Now setting $t = \varepsilon \mathbb{E}[T_G] = \varepsilon \cdot O(n^3)$, we get

$$\mathbb{P}\left[\left|\frac{T_G}{p^3 n^3/6} - 1\right| > \varepsilon\right] \leq \frac{O(n^5)}{\varepsilon^2 \cdot O(n^6)} \leq O_{p,e}(1/n) \to 0$$

(as $n \to \infty$; the constants may depend on p and ε). □

§1.5 Connectivity

Now let's switch to something more interesting. If you have a graph and you think about it as a network, one main event you care about is whether or not the network is connected. So now let's look at connectivity, which is a more *global* property of the graph.

There's actually a rather remarkable phase transition. Of course, if p is a constant, you expect the graph to not just be connected but very dense. So we'll look at a much sparser regime.

Theorem 1.10

Let $p_n = (c \log n)/n$ (where c is some constant). Then

$$\lim_{n\to\infty} \mathbb{P}[\mathcal{G}(n, p_n) \text{ is connected}] = \begin{cases} 1 & \text{if } c > 1 \\ 0 & \text{if } c < 1. \end{cases}$$

So the asymptotics for this probability depend on c. This probability goes to 1 if c is *any* constant greater than 1 (e.g., $c = 1.001$) and 0 if c is any constant less than 1 (e.g., 0.999). So this is a *phase transition* — if you vary c around 1, you get drastically different behavior on large-scale networks.

Student Question. *What happens when $c = 1$?*

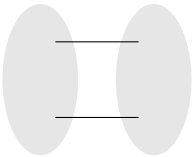
Answer. Actually, there's a fairly explicit function for this; it's a constant strictly between 0 and 1.

§1.5.1 The approach

To refresh our memory, connectivity means that if I look at any pair of vertices, there's some way of navigating from the first to the second — there's some path in the graph going from u to v. There's actually another definition of connectivity that'll be more amenable for our analysis.

Fact 1.11 — A graph G is connected if and only if for all $\emptyset \subsetneq S \subsetneq V$, there is an edge $uv \in E$ such that $u \in S$ and $v \in \overline{S}$.

In other words, for every nontrivial cut S, there's an edge crossing that cut (we use $\overline{S}$ to denote the complement of S). So the picture looks like we break the graph into two pieces, and require that there's at least *some* edge that crosses this cut. If I have this property for all proper subsets S, then my whole graph is connected.



Definition 1.12. We call a partition $(S, \overline{S})$ a cut, and we use $E(S, \overline{S})$ denote the set of edges crossing the cut.

Now with this definition of connectivity, we can see that

$$\mathbb{P}[G \text{ is disconnected}] = \mathbb{P}[\text{exists } S \text{ s.t. } E(S, \overline{S}) = \emptyset].$$

In other words, the probability our graph is disconnected is the probability there exists some cut that has no edges crossing it. And this is nice because we have a probability of *existence*; so what's the natural thing to use? If we just want upper bounds, we can just use a union bound, and we get

$$\mathbb{P}[\text{disconnected}] \leq \sum_S \mathbb{P}[E(S, \overline{S}) = \emptyset].$$

Because S and its complement induce the same cut, we'll just look at all sets of size at most $n/2$.

This is a good thing because these probabilities are extremely explicit — now we just have a whole bunch of edges, we look at all the possible pairs crossing this cut, and we just need to ensure none actually become edges in this graph. So using the independence of how our graph is generated, this becomes

$$\sum_{k=1}^{n/2} \binom{n}{k} (1-p_n)^{k(n-k)} \tag{1.1}$$

(where $\binom{n}{k}$ is the number of subsets of size k, and $k(n-k)$ is the number of pairs crossing this cut).

So the plan is to show that for $c > 1$, even this upper bound (1.1) decays to 0 (since we're looking at the probability the graph is disconnected). For the other side, this is an upper bound, so it doesn't prove the other side. But it'll *suggest* what we should do to prove the other side. As a preview, we'll examine the $k = 1$ case, which will turn out to be sort of the highest-order contribution to the sum.

§1.5.2 Some intuition

Before we jump into computations, let's see why we're looking at this specific constant — why $c = 1$?

Here's a very useful asymptotic that you can use to get at least a preliminary handle on a quantity like $\binom{n}{k}$: very informally speaking, we have $\binom{n}{k} \approx \exp(\mathsf{H}_e(k/n) \cdot n)$, where $\mathsf{H}_e(x)$ is the *entropy*

$$\mathsf{H}_e(x) = x \log x - (1-x) \log(1-x).$$

This function looks something like an upside-down parabola hitting the x-axis at 0 and 1. (We use the e subscript to mean that we're using base-e logarithms.)

Now let's look at this other term $(1-p_n)^{k(n-k)}$. Using the very useful inequality $1 - x \leq e^{-x}$ (which we'll use many times throughout this course; it's a good approximation for small x), we get

$$(1-p_n)^{k(n-k)} \leq \exp\left(-p_n \cdot k(n-k)\right) = \exp\left(-c \log n \cdot \frac{k}{n}\left(1 - \frac{k}{n}\right) \cdot n\right).$$

The p_n already has a $1/n$ in it; we have $c \log n$ from the rest of p_n; and everything else is the same.

And we want to basically ensure that this term is much smaller than however large the $\binom{n}{k}$ term is. The convenient thing is that this is also a quadratic-looking function in terms of k/n. So we'll have the following picture: we can imagine plotting $\mathsf{H}_e(q)$ where $q = k/n$; and then we'll have a much larger parabola correspondign to $c\log n \cdot q(1-q)$. Basically, when $c > 1$, we'll have something like this where the second parabola dominates the first (at least, in the vast majority of our interval). This will tell us that the $(1-p_n)^{k(n-k)}$ term is much, much smaller than however large the $\binom{n}{k}$ term is.

So roughly speaking, we would expect from this heuristic that $\exp(-c\log n \cdot \frac{k}{n}(1-\frac{k}{n})\cdot n)$ is much smaller than $\exp(\mathsf{H}_e(\frac{k}{n})\cdot n)$. And here, the $\log n$ is going to be helping us.

This is really informal intuition, but now let's try to formalize this.

§1.5.3 The case $c > 1$

(This is the regime where we want our disconnectivity probability to decay to 0.)

We're going to use something called Stirling's formula, which is how you actually get the earlier approximation on $\binom{n}{k}$:

> **Fact 1.13** (Stirling) — We have $\lim_{k\to\infty} \frac{k!}{k^{(k+1)/2}e^{-k}} = 1$.

Basically we're trying to bound $k!$ by something that's a bit more tractable (nicer for our calculations).

This in particular implies that

$$\log k! \geq \left(k+\frac{1}{2}\right)\log k - k - A$$

for some constant A, which we're basically going to ignore (it's not going to be important for us).

Now let's try to keep upper-bounding this guy. We'll also use the fact that

$$\binom{n}{k} \leq \frac{n^k}{k!}.$$

We'll split into two regimes, because technically speaking, this approximation only holds if $k = \Theta(n)$ (e.g., $k \geq n/100$).

So let's fix some constant $\varepsilon > 0$. For technical reasons, we need $(1-\varepsilon)\cdot c > 1$. (This always exists because we assumed $c > 1$.) Then we'll have two cases, depending on whether k is small (at most εn) or big (at least εn). We'll split the sum into these two cases, and consider them separately. We'll call the right-hand side of (1.1) RHS, so that

$$\text{RHS} \leq e^A \sum_{k=1}^{n/2} \exp\left(k\log n - k(n-k)p_n - \left(k+\frac{1}{2}\right)\log k + k\right).$$

(The second term is from $(1-p_n)^{k(n-k)}$, and the rest comes from our use of Stirling.)

Case 1 ($1 \leq k \leq \varepsilon n$). Here we're basically going to pick out some of these terms that are not relevant for us, and sort of simplify our expression. In particular, observe that if we look at the $-(k+1/2)\log k + k$ term, that's going to be upper-bounded by some constant, just because $k\log k$ grows faster than something linear in k. So we can say

$$\exp\left(-\left(k+\frac{1}{2}\right)\log k + k\right) \leq O(1).$$

Now let's look at the first two terms; plugging in our choice of p_n, we have

$$k\log n - \frac{c\log n}{n}k(n-k).$$

We can bound $k(1-k) > (1-\varepsilon)n$, so this is at most

$$-k((1-\varepsilon)c - 1)\log n$$

(we can kind of combine the k's and $\log n$'s; we assumed k is small, so $n-k$ is large — at least $(1-\varepsilon)n$ — and then if you simplify, you get this expression).

In particular, if we now look at just the first εn terms, their sum is upper-bounded (up to constants) by

$$\sum_{k=1}^{\varepsilon n} n^{-\widetilde{c}k}$$

where $\widetilde{c} = (1-\varepsilon)c - 1 > 0$. This is good — it's $O(n^{-\widetilde{c}})$, which decays to 0.

So that's the first case; now let's do the second.

Case 2 ($\varepsilon n \le k \le n/2$). Then the observation is that

$$k\log n - \left(k + \frac{1}{2}\right)\log k + k \le k\log n - k\log(\varepsilon n) = k(1 + \log(1/\varepsilon)).$$

So this scales as some constant times k. At the same time, we know that

$$k(n-k)p_n \ge \frac{c}{2}k\log n$$

(because $k \le n/2$, so $n - k \ge n/2$, and the n cancels with the $1/n$ in p_n). In particular, if we look at the remaining $(1/2 - \varepsilon)n$ terms, they're upper-bounded by

$$\sum_{k=\varepsilon n}^{n/2} \left(\frac{e}{\varepsilon n^{c/2}}\right)^k = O(n^{-c/2})$$

(these two terms combine, so we get some exponent with k, and the base of the exponent depends on these things). This also goes to 0 as $n \to \infty$.

(If you didn't follow too many of the calculations that's fine; we're just trying to formalize some of the approximations we said earlier.)

So to summarize, we've used this union bound to produce this upper bound (1.1) on the probability the graph is disconnected. To formalize Stirling's approximation and everything, we broke into two regimes — when k is at most or at least εn — and we saw both of those terms are upper-bounded by something decaying to 0 as $n \to \infty$. So that's it for the first case; now we'll move on to the second case.

§1.5.4 The case $c < 1$

The idea is that in (1.1), this quantity is no longer going to decay to 0 as $n \to \infty$, and the reason that's the case is sort of because of the $k = 1$ terms — we have

$$\sum_{k=1}^{n/2} \binom{n}{k}(1-p_n)^{k(n-k)} \ge n \cdot (1-p_n)^{n-1} \approx n\exp(-c\log n).$$

And if $c < 1$, this is at least n^{1-c}, which is actually growing with n.

What this suggests is that what we should look at are *isolated vertices*, which correspond to the case $k = 1$ (i.e., $k = 1$ corresponds to the presence of isolated vertices). Certainly, if you have an isolated vertex, your

graph cannot be connected. So what we'll actually do now is show that with probability going to 1, there will exist an isolated vertex (when $c < 1$).

So let $\mathcal{I}_v$ be the indicator that v is isolated, meaning that it has no neighbors in the graph. Then we have

$$\mathbb{E}\left[\sum_v \mathcal{I}_v\right] = n(1-p_n)^{n-1} \gtrsim n^{1-c}.$$

(This grows with n.) We want to deduce that

$$\mathbb{P}[\text{exists an isolated vertex}] \to 1.$$

So this is our goal.

We have control on the expectation, and we want to show the probability it's positive goes to 1. Control on the expectation is not enough to get a lower bound on the probability that the random variable is positive. For that, we need a bound on the *variance*; and then we can use Chebyshev — this is the same strategy as what we did for triangles.

Let's let $\mathcal{I} = \sum_v \mathcal{I}_v$ be the number of isolated vertices; we want to show that $\mathcal{I}$ is something positive. So let's look at the variance, or the second moment; we have

$$\mathbb{E}[\mathcal{I}^2] = \sum_v \mathbb{E}[\mathcal{I}_v^2] + \sum_{u \neq v} \mathbb{E}[\mathcal{I}_u \mathcal{I}_v]$$

(by linearity of expectation). The first term is easy to compute. For the second, let's look at two vertices u and v; there's a bunch of possible edges out of each of them. And we just need to ensure that none of these $(n-2)+(n-2)+1$ edges occur in the graph (where the $+1$ is for the edge uv); so this is

$$\mathbb{E}[\mathcal{I}^2] = n(1-p_n)^{n-1} + n(n-1)(1-p_n)^{2(n-2)+1}.$$

Now this is just a calculation; we have a random variable with very large expectation, so it being 0 means it's deviating very far from its expectation. So by Chebyshev we get

$$\mathbb{P}[\mathcal{I} = 0] \leq \mathbb{P}\left[\mathcal{I} \leq \frac{1}{2}\mathbb{E}[\mathcal{I}]\right] \leq \frac{4\operatorname{Var}[\mathcal{I}]}{\mathbb{E}[\mathcal{I}]^2}.$$

If you calculate everything out, you get something like

$$\frac{1}{n^{1-c}} + p_n \to 0 \quad \text{as } n \to \infty.$$

So to summarize, we saw that if c exceeds the threshold 1, then with high probability our graph will be connected. And if $c < 1$, then not only is the graph disconnected, but there must exist an isolated vertex (and actually there will exist *many* isolated vertices).

§2 February 5, 2025 — Second moment method and branching processes

Last time, we introduced percolation, and a very fundamental model of random graphs called the Erdős–Rényi random graph. We saw that just by computing expectations and variances, you could deduce very interesting phenomena, like the phase transition in connectivity. Today we'll develop these more (particularly the first and second moments), and use it to study a stochastic process where we no longer have full independence. The next few lectures will be about using the first and second moment methods to deduce very interesting results. (The first problem set will be posted after class and due the Monday after next one.)

§2.1 The Paley–Zygmund inequality

Last time, we saw how we can use bounds on the variance combined with Chebyshev to upper-bound the probability that some random variable deviates from its expectation. But if our goal is just to lower-bound the probability a random variable is *positive* (if we're trying to use the probabilistic method to certify some object exists), there's a slightly better result that has a wider range of applicability.

Theorem 2.1 (Paley–Zygmund inequality)

Let X be a nonnegative random variable. Then for any $0 \leq \theta < 1$, we have

$$\mathbb{P}[X > \theta \cdot \mathbb{E}[X]] \geq (1-\theta)^2 \cdot \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}.$$

For instance, if $\theta = 0$, then you're just lower-bounding the probability that X is positive.

Before we prove this, maybe it's helpful to first try to compare this with what you would get if you just applied Chebyshev's inequality out of the box. We have $0 < \theta < 1$, so Chebyshev will give you an upper bound on the probability that X is very far from its expectation, in particular less than $\theta\mathbb{E}[X]$; so

$$\mathbb{P}[X \leq \theta\mathbb{E}[X]] \leq \mathbb{P}[|X - \mathbb{E}[X]| \geq (1-\theta)\mathbb{E}[X]] \leq \frac{\mathrm{Var}[X]}{(1-\theta)^2\mathbb{E}[X]^2}.$$

This implies that

$$\mathbb{P}[X > \theta \cdot \mathbb{E}[X]] \geq 1 - \frac{\mathrm{Var}[X]}{(1-\theta)^2\mathbb{E}[X]^2}.$$

Now let's think about when this bound actually tells us anything at all. Certainly you ned $(1-\theta)\mathbb{E}[X]^2 \leq \mathrm{Var}[X]$, or else this just tells us the probability is nonnegative, which is absolutely useless. For convenience, assume $\theta = 0$; then Chebyshev is meaningful if and only if

$$\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \leq \mathbb{E}[X]^2,$$

in other words if the second moment is bounded by *twice* the square of the first moment.

Now let's compare with what happens if you use Paley–Zygmund. There all you need is that the second moment is bounded by *any* constant times the square of the first — even 100 would be okay. So Paley–Zygmund is meaningful even if $\mathbb{E}[X^2] = 100\mathbb{E}[X]^2$. In this sense, it has a much wider range of applicability, especially if you're trying to use the second moment method to certify that some property holds or some object exists, typically in conjunction with the *probabilistic method*.

Fortunately for us, the proof is also not too difficult.

Proof. We're going to use a very standard trick when analyzing random variables — if you want to bound the expectation of some random variable, you can try splitting it into two regimes, one where the variable is large and one where it's small. You can do this by introducing indicator random variables, sort of artificially. In particular, we can write

$$\mathbb{E}[X] = \mathbb{E}[X \cdot \mathbf{1}_{X \leq \theta\mathbb{E}[X]}] + \mathbb{E}[X \cdot \mathbf{1}_{X > \theta\mathbb{E}[X]}].$$

(So we've artificially inserted some indicator random variables here.) Now we can use some inequalities. For instance, for the first random variable, it's going to give me a 0 if $X > \theta\mathbb{E}[X]$, and otherwise it's less than $\theta\mathbb{E}[X]$. So for free we get

$$\mathbb{E}[X \cdot \mathbf{1}_{X \leq \theta\mathbb{E}[X]}] \leq \theta\mathbb{E}[X].$$

(You can sort of see where this is going — we're going to move this to the other side and do some rearranging and get what we want.)

To handle the second term, we use Cauchy–Schwarz, which is how we get the second moment out — we have

$$\mathbb{E}[X \cdot \mathbf{1}_{X>\theta\mathbb{E}[X]}] \leq \sqrt{\mathbb{E}[X^2]\mathbb{P}[X > \theta\mathbb{E}[X]]}$$

by Cauchy–Schwarz. And now that we have this, we can just rearrange — we have

$$\mathbb{E}[X] \leq \theta\mathbb{E}[X] + \sqrt{\mathbb{E}[X]^2\mathbb{P}[X > \theta\mathbb{E}[X]]},$$

which rearranges to the desired bound. □

§2.2 Galton–Watson branching processes

Now let's actually use this for something. We'll now introduce a class of stochastic processes called *branching processes* — these will be random variables that are not all jointly independent (though they'll have some independence baked into them). One reason we'll look at them is they're very useful for making heuristic predictions about more complex random variables, like the local structure of Erdős–Rényi random graphs (we'll talk more about this at the end of the lecture).

To define a branching process, we need to first start with an ffspring distribution: Let ξ be a distribution on $\mathbb{N} = \{0, 1, \ldots\}$ (for example, you could take a binomial distribution $\text{Bin}(d, p)$). We'll call this the *offspring distribution*. (The reason for the terminology is that you can imagine a population of organisms dividing, and this is going to model the evolution of the population as we increase the number of generations.)

Definition 2.2. A Galton–Watson branching process is a sequence of random variables $\{Z_\ell\}_{\ell\in\mathbb{N}}$ which we generate inductively as follows:

- $Z_0 = 1$ with probability 1.
- For each ℓ, we generate Z_ℓ by adding $Z_{\ell-1}$ many independent samples drawn from our offspring distribution ξ.

The kind of picture you should have in your mind — and the reason for the name — is you can imagine this process generates a possibly infinite random tree. We start with a root r (since $Z_0 = 1$). At the first step we generate a random integer from the offspring distribution; this gives us a set of children (let's say $Z_1 = 3$). Then each child independently spawns its own set of children, again sampled from this offspring distribution (maybe they have 2, 1, and 4 children, respectively, so Z_2 would be $2 + 1 + 4 = 7$), and so on. Maybe some of them die out, and others continue on forever, and so on. That's kind of why it's called a branching process.

Student Question. *Does it matter which parent a child process has? Like, in Z_1 you have 3 different nodes, and in Z_2 you have 7 items; are those 7 identical?*

Answer. Yes — you could track additional information if you wanted, but for this lecture we'll only track the number of vertices at each generation, and not track which vertex is in which subtree.

There's a lot of independence, but certainly Z_1 depends on Z_0, Z_2 on Z_1, and so on. So between the counts there's lots of complex dependencies, but there's also some amount of independence baked into how we're generating these random variables.

§2.2.1 First and second moments

We have a bunch of random variables, so let's compute the mean and variance and so on, because that's the first thing you do when you want to understand a new randmo variable.

Lemma 2.3

Suppose the offspring distribution ξ has mean μ and variance $\sigma^2 < \infty$. Then in the Galton–Watson branching process, we have

$$\mathbb{E}[Z_\ell] = \mu^\ell \quad \text{and} \quad \operatorname{Var}[Z_\ell] = \sigma^2 \mu^{\ell-1} \sum_{k=0}^{\ell-1} \mu^k.$$

For the expectation, this makes sense — you can kind of see there's going to be some exponential growth (or decay). The variance is harder to guess, but turns out to be quite nice as well. It's of course related to the variance of the original distribution, and then there's extra stuff capturing the exponential growth or decay.

In the interest of time, we're just going to prove the first claim. The second claim is actually not any more difficult; it's the same proof, just with slightly more calculations, and it's in the notes.

Proof for the mean. Let's do induction, because these random variables are inductively defined. The base cases $\ell = 0$ and 1 are true by definition. So now let's assume it holds for some ℓ, and try to compute $\mathbb{E}[Z_{\ell+1}]$. How would you expand $\mathbb{E}[Z_{\ell+1}]$? We can use the law of total expectation to write this as

$$\mathbb{E}[Z_{\ell+1}] = \sum_{k=0}^{\infty} \mathbb{E}[Z_{\ell+1} \mid Z_\ell = k] \mathbb{P}[Z_\ell = k].$$

Now this expectation is nice — it says I have exactly k vertices, each of which generates some number of nodes according to our offspring distribution; so we have

$$\mathbb{E}[Z_{\ell+1} \mid Z_\ell = k] = k \cdot \mu.$$

And this means

$$\mathbb{E}[Z_{\ell+1}] = \mu \sum_{k=0}^{\infty} k \mathbb{P}[Z_\ell = k] = k \mathbb{E}[Z_\ell]. \qquad \square$$

The proof for the variance is the same idea — you expand out $\mathbb{E}[Z_{\ell+1}^2]$ and so on.

§2.2.2 Extinction

We're going to use this to do something a little more interesting. If you think of this branching process as modelling the population of a collection of organisms, one thing you might be interested in is whether this collection of organisms eventually dies out, or goes extinct. So we're going to study this probability. (This also turns out to be intimately connected to the behavior of many more complex processes.)

Definition 2.4. We define $E_\ell = \{Z_\ell = 0\}$; the extinction event is $E = \bigcup_\ell E_\ell$.

So this is the event that at some point (possibly far in the future), my petri dish of organisms just dies out.

First, can we give *some* lower bound on the probability of extinction as a function of the offspring distribution? One lower bound is just the probability of the offspring distribution giving 0 (since if you die in the first step, you're done):

Fact 2.5 — We have $\mathbb{P}[E] \geq \xi(0)$.

But we want to get more information — maybe even a way to *compute* this probability. We're crucially going to take advantage of this recursive way in which you can generate a Galton–Watson branching process.

Lemma 2.6

If we define $\psi_\xi(s) = \mathbb{E}_{X\sim\xi}[s^X]$, then $\mathbb{P}[E]$ must be a fixed point of ψ_ξ, i.e.,

$$\mathbb{P}[E] = \psi_\xi(\mathbb{P}[E]).$$

If you've seen moment generating functions before, this function ψ_ξ is basically a reparametrization of that. And this function is going to allow us to get a handle on what the actual extinction probability is.

Proof. The key idea for why you would define this function ψ_ξ, or how you'd go about proving this, is similar to how we proved the mean — to take advantage of the recursive nature of these random processes. The recursive definition of the process is as follows:

- We first generate $Z_1 \sim \xi$.
- Then we independently recursively generate Z_1-many Galton–Watson branching processes. Let's call our Galton–Watson branching process $\mathcal{Z} = \{Z_\ell\}_{\ell\in\mathbb{N}}$; then in the second step, we're generating $\mathcal{Z}^{(1)}$, $\ldots$, $\mathcal{Z}^{(Z_1)}$.
- Then we set Z_ℓ as the sum of the $(\ell-1)$th terms of each of these subprocesses (for all ℓ), i.e.,

$$Z_\ell = \sum_{i=1}^{Z_1} \mathcal{Z}^{(i)}_{\ell-1}.$$

(Each $\mathcal{Z}^{(i)}$ is an entire sequence of random variables geneated from this process, and we combine all of these together to get a new random process with the same law.)

Why do we want to do this? A fixed point equation is kind of a recursively defined thing. The way we'll think about this now is that the only way for the process $\mathcal{Z}$ to go extinct is if every single one of these guys $\mathcal{Z}^{(i)}$ goes extinct. (Ultimately the whole process is a sum of these random variables; so if one $\mathcal{Z}^{(i)}_{\ell-1}$ is positive, then Z_ℓ is also positive. Intuitively, if one of my children has infinitely long lineage, then I have an infinitely long lineage. So as long as one of my children spawns an infinite subtree, I also contain one.) So the observation is that

$$\begin{aligned}\mathbb{P}[\mathcal{Z} \text{ goes extinct}] &= \sum_{k=0}^{\infty} \mathbb{P}[\mathcal{Z} \text{ goes extinct} \mid Z_1 = k]\xi(k)\\ &= \mathbb{P}[\mathcal{Z}^{(i)} \text{ goes extinct for all } i \mid Z_1 = k]\xi(k).\end{aligned}$$

(Here we're splitting up by the value of Z_1 — the number of subprocesses generated — using the law of total probability as before.)

Now we can use independence — we have a $\forall$ statement and a bunch of independent processes, so this becomes

$$\sum_{k=0}^{\infty} \xi(k) \cdot \prod_{i=1}^{k} \mathbb{P}[\mathcal{Z}^{(i)} \text{ goes extinct}].$$

And each of these probabilities is just the probability of the extinctino event (since the $\mathcal{Z}^{(i)}$'s all have the same distribution as $\mathcal{Z}$), so we get

$$\mathbb{P}[E] = \sum_{k=0}^{\infty} \xi(k)\mathbb{P}[E]^k = \psi_\xi(\mathbb{P}[E]).$$ □

Now here's the major thing, which classifies *when* these processes go extinct. You can already guess there should be a phase transition around $\mu = 1$ — if $\mu > 1$ we'd expect exponential growth, and if $\mu < 1$ we'd expect exponential decay.

Theorem 2.7

- (Subcritical case) Suppose $\mu < 1$. Then $\mathbb{P}[E] = 1$.
- (Supercritical case) Suppose $\mu > 1$. Then there exists $p^* \in [\xi(0), 1)$ such that $\mathbb{P}[E] = p^*$.
- (Critical case) Suppose $\mu = 1$ and $\sigma^2 > 0$. Then $\mathbb{P}[E] = 1$.
- Otherwise, if $\mu = 1$ and $\sigma^2 = 0$ (meaning that ξ always spits out 1), then $\mathbb{P}[E] = 0$.

(In fact, in the subcritical case, 1 is the *only* fixed point of ψ_ξ.)

We're just going to prove the first two cases; the main emphasis is going to be on the supercritical case.

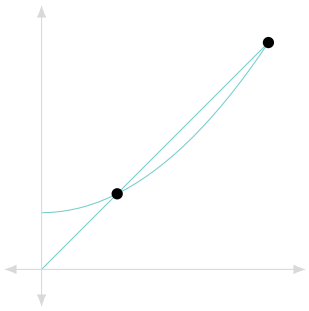
Proof sketch for the subcritical case. When $\mu < 1$, $\mathbb{E}[Z_\ell] = \mu^\ell$ is exponentially decaying. So in particular

$$\sum_{\ell=0}^{\infty} \mathbb{E}[Z_\ell] = \sum_{\ell=0}^{\infty} \mu^\ell = \frac{1}{1-\mu} < \infty.$$

But if there was some positive probability that the process never goes extinct, then there's no way this could be finite — it would just be ∞ times whatever that positive probability is. So this means $\mathbb{P}[E] = 1$ (i.e., the process must go extinct.) □

Proof for the supercritical case. First, we saw that $\mathbb{P}[E]$ is a fixed point of ψ_ξ. We always have 1 as a fixed point. The first thing is we're going to show there's just *two* fixed points — one will be 1, and the other will be p^*. Once we have that, all we need to show is that the extinction probability is strictly less than 1; then it's not that fixed point, so it has to be the other one p^*.

Let's first do step (1) — proving there are only two fixed points. For this, we'll need one basic and nice property of ψ_ξ, which is that it's *convex*. If you were to plot what this function looks like, you could imagine drawing the line $y = x$; there's a fixed point at 1, and we're going to show that what ψ looks like is it's going to dip down, cross the line, and look something like this (where the y-intercept is $\psi(0)$, and the intersection with $y = x$ is our p^*).



You can sort of prove this convexity just by differentiating the thing, so we won't go through that. Also, the key is that the derivative of the tangent at 1 has slope strictly steeper than the line $y = x$, so that we get below the line just before $y = 1$.

So we observe that ψ_ξ is convex and that $\psi'_\xi(1) = \mu > 1$.

So this is a convex function that dips below 1, and that means there'll have to be at least two intersection points; we know there's a second one because at some point it becomes bigger than the line $y = x$.

Now we're just going to rule out our extinction probability being equal to 1 — we just need to say it's less than 1, and then we're done.

So now let's do step (2); this is where we're going to use our second moment, since we computed the variance and mean of our random process. We just need to show $\mathbb{P}[E] < 1$; equivalently, we want to show

$$\mathbb{P}\left[\bigcap_{\ell=0}^{\infty}\{Z_\ell > 0\}\right] > 0.$$

Now, this is an intersection of infinitely many events; we can write this as

$$\lim_{\ell\to\infty}\mathbb{P}\left[\bigcap_{k=0}^{\ell}\{Z_k > 0\}\right].$$

And in this thing, if $Z_\ell > 0$, then all the previous Z_k's are also positive; so this is the same thing as

$$\lim_{\ell\to\infty}\mathbb{P}[Z_\ell > 0].$$

So we just need to show this thing is lower-bounded by some positive constant *independent* of ℓ.

So our goal is to show that $\mathbb{P}[Z_\ell > 0] \geq C$ for some $C > 0$ independent of ℓ.

Now we can finally use Paley–Zygmund from earlier — it says that

$$\mathbb{P}[X > 0] \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}.$$

Applying this and using our formulas for the mean and variance, we get that

$$\mathbb{P}[Z_\ell > 0] \geq \frac{\mathbb{E}[Z_\ell]^2}{\mathbb{E}[Z_\ell^2]} = \frac{\mu^{2\ell}}{\mu^{2\ell} + \sigma^2\mu^{\ell-1}\sum_{k=0}^{\ell-1}\mu^k}.$$

The μ^k grow exponentialy fast, so this is approximately μ^ℓ (up to constants). And after some simplifications, we get

$$\mathbb{P}[Z_\ell > 0] \geq \frac{1}{1 + c\sigma^2/\mu^2}$$

(for some constant c). The crucial thing is that this is independent of ℓ! So we're done. □

This completes the proof of the most interesting case; the last cases are kind of not so interesting, but if you want to see them, they're in the notes.

You might wonder how you compute this extinction probability; the fact that it's a fixed point means you can use fixed-point iteration to at least numerically estimate what the extinction probability is.

> **Student Question.** *Where does the second part break if $\mu < 1$?*
>
> **Answer.** If $\mu < 1$, then the sum $\sum_k \mu^k$ becomes some constant; now you have $\mu^{2\ell}$ in the numerator and μ^ℓ in the denominator, so the numerator decays much faster.

§2.3 Percolation on a tree

Now we'll go back to percolation; except we won't do percolation with the complete graph anymore, but rather percolation on a tree.

Definition 2.8. We use $\widehat{\mathbb{T}}_d$ to denote the infinite d-ary tree.

(The hat is to distinguish it from the infinite d-regular tree, which is slightly different.)

In percolation, we have a graph, and we generate a random subgraph by including each edge independently with some probability p. One reason this thing is called percolation to begin with is it was invented by mathematical physicists, who wanted to know, if I want to model some porous material and I imagine dumping water (e.g., through a sponge), what's the probabiltiy I see water flowing through the material? To model this, you imagine the edges of the graph being pipes which water can flow through — if I have an edge (with some probability) then water can flow through. And I want to know the probability there's a path from some vertex to another. Typically this is modelled on infinite graphs, because lots of material has billions of atoms — for example, integer lattices. But analyzing percolation on lattices can be hard, so people usually start with an easier infinite graph, like the infinite tree.

Definition 2.9. We define the percolation event E_∞ as the event that there exists an infinite connected component in a random subgraph drawn from the p-percolation of $\widehat{\mathbb{T}}_d$.

So p is our parameter, for the probability we include each edge. For convenience, we'll use $\mathbb{P}_p$ to indicate our edge probability — so we'll write $\mathbb{P}_p[E_\infty]$ for the event that there exists an infinite component.

Definition 2.10. The critical probability p_c is defined as

$$p_c = \sup\{p \mid \mathbb{P}_p[E_\infty] = 0\}.$$

The larger p becomes, the more edges you have, so you'd expect to see larger components (in particular, an infinite one). So the critical probability is the largest probability for which you don't.

Theorem 2.11

We have $p_c = 1/d$.

This is maybe not too surprising — you can use what we just saw about branching processes. I start from my root, and I first look at how many of my children are included in my random subgraph, and then how many of *theirs* are included, and so on. This is going to generate a branching process where the offspring distribution is binomial, with d trials and probability p. And the expectation of this is pd; and we saw that the threshold is when the expectation is 1, meaning that $pd = 1$.

As a corollary, there's a phase transition even in the value of the percolation probability as well.

Corollary 2.12

We have

$$\mathbb{P}_p[E_\infty] = \begin{cases} 1 & \text{if } p > 1/d \\ 0 & \text{if } p \leq 1/d. \end{cases}$$

This is actually not too difficult to see. We're not going to prove the theorem, because it follows fairly immediately from what we derived about branching processes. We're also not going to prove this corollary. But it's fairly intuitive — once there's *some* positive probability of generating some infinite component according to the branching process, the picture looks like this. Imagine we have our root; if $p > 1/d$, there's some positive probability that the root vertex itself already generates an infinite component. Now, if it doesn't, let's say it gets chopped off at some finite depth. But then the vertices at this depth start their own subtrees. So we're trying an infinite number of times, where at each time there's some positive probability

of success; and this means at some point someone is going to succeed. That's why we get 1. (There's a way to formalize this that has a name, called the Kolgomorov 0–1 law.)

§2.4 Erdős–Rényi and branching processes

Now in the last 10 or so minutes, we'll briefly discuss how to connect this back to Erdős–Rényi random graphs.

Last lecture, we saw that in the Erdős–Rényi random graph, we have

$$\mathbb{P}[\mathcal{G}(n, c(\log n)/n) \text{ connected}] \to \begin{cases} 1 & c > 1 \\ 0 & c < 1. \end{cases}$$

But there's an arguably even more interesting transition that occurs well before this, which we'll see later on. We'll consider the regime where p scales as $1/n$. Then the graph is going to be disconnected. But in fact, if $p_n = c/n$ for $c > 1$, then with probability $1 - o(1)$, $\mathcal{G}(n, p_n)$ has a *unique* connected component of linear size $\Omega(n)$ (a constant fraction of the vertices in the graph). So the graph is not globally connected, but it's going to have a very large community. And there's actually a phase transition — if $c < 1$, then with high probability *all* your components are tiny, specifically of size size $O(\log n)$. We'll prove this in a future lecture; but we'll briefly talk about how you can use branching processes to at least *predict* such a phenomenon would occur. (Formalizing this is a whole different issue, but one first starts off with heuristics.)

Heuristically, imagine I start from some arbitrarily picked vertex, and I want to know, how large is the connected component containing this specific vertex v? So I'm going to start exploring my randomly sampled Erdős–Rényi graph by a BFS — I'll first reveal the vertices in the depth-1 neighborhood, then at level 2, then 3, and so on.

Let's look at how many neighbors we'd expect to see. If we call this number of neighbors N_1, it's distributed as $N_1 \sim \text{Bin}(n-1, p_n)$ (I have $n-1$ different vertices I could possibly connect to). So we can see if $p_n = c/n$ for $c > 1$, then we expect a constant (larger than 1) number of neighbors.

Now let's look at the second level. For each u in the level-1 neighborhood, the number of vertices left that we haven't already explored is roughly n minus some constant ($n - 1$ minus however many vertices were in the level-1 neighborhood, which we expect to be some constant). So we have $n - o(n)$ many vertices here. And so the number of new vertices we gain when we take our BFS one further step is going to be again distributed approximately according to $\text{Binom}(n, p_n)$. This is not rigorous by any means — it's not exactly the right number of vertices, and there may be some collisions (where two u's have the same neighbor on this level), but that's a pretty rare event.

So the heuristic picture is that if N_ℓ is the number of vertices at distance ℓ from v, and I look at the sequence $\{N_\ell\}_\ell$, then this is *approximately* a Galton–Watson branching process with offspring distribution $\text{Bin}(n, p_n)$ (we're being very loose here). (For large ℓ this cannot hold; but you have exponential growth, so presumably by the time you've hit large ℓ, e.g. $\ell \approx \log n$, you're already at linearly many vertices. Similarly, if $c < 1$, we'd expect this process is going to die off pretty quickly — in fact, we'd expect the size of this component to typically be constant.)

So this is a heuristic about how we can use branching processes to make predictions about the behavior of something much richer.

In the next lecture, we're going to use some of these ideas to study a statistical inference problem.

§3 February 10, 2025 — Hypothesis testing and the second moment method

Hopefully we've taken a look on the first pset. A a quick remark, if you want to use a result that is not stated or proven in the lecture notes and is not found in one of the prerequisite courses, then you should

prove that result. So if you found it on Wikipedia and it's not a result you've already seen in a previous course or the lecture notes, please prove it.

§3.1 Hypothesis testing

Today we're going to talk about hypothesis testing. This is one of those fundamental statistical inference-type problems. Here you're given a *sample* (or an *observation*) X from one of two distributions — either it's from distribution μ or from distribution ν (these are sometimes called the *hypotheses* — they're hypotheses for how this random variable was generated). And your goal is to tell which distribution it came from — to tell whether $X \sim \mu$ or $X \sim \nu$.

So this is the basic setup behind hypothesis testing. You can consider extensions of this problem where you have multiple samples and so on, but for simplicity we'll consider the setting where you have one.

§3.2 Broadcast processes

We'll also look at a *specific* hypothesis testing problem, called the *reconstruction problem*; this is a classical statistical model in evolutionary genetics.

To set this up, we'll define a specific class of stochastic processes called *broadcast processes*. You can consider this on any graph, but here for simplicity we'll just consider trees, in particular the infinite d-ary tree $\widehat{\mathbb{T}}_d$ (where every vertex has exactly d children).

Then the broadcast process consists of a bunch of correlated coin flips.

Definition 3.1 (Broadcast process)**.** Fix $d \in \mathbb{N}$ and an error parameter $0 \le \varepsilon \le \frac{1}{2}$. Sample a labelling $\sigma\colon \{\text{vertices of } \widehat{\mathbb{T}}_d\} \to \{\pm 1\}$ as follows:

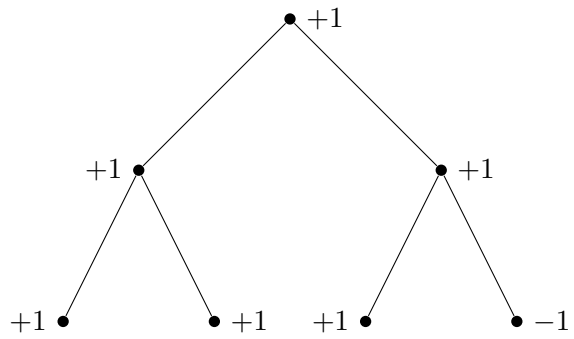
(1) For the root vertex r, we sample its assignment $\sigma_r \sim \mathsf{Unif}\{\pm 1\}$.

(2) Recursively, for each child u of r, we sample its assignment based on the assignment already given to its parent — we set

$$\sigma_u = \begin{cases} \sigma_r & \text{with probability } 1-\varepsilon \\ -\sigma_r & \text{with probability } \varepsilon. \end{cases}$$

(3) This process continues recursively in each subtree.

To visualize what's happening here, we have our tree: let's maybe look at the case $d = 2$, so every vertex has exactly two children. I generate an assignment of ± 1 to *all* the vertices of this tree. First I assign the root uniformly. Then for each vertex, I copy its value to its children with some large probability; and there's some small probability ε of there being a mutation.



Eventually, we're going to see the vertices at a distance n from the root, and our goal will be to reconstruct the assignment for the root — so we'll have two hypotheses, one where the root was $+1$ and one where the root was -1.

(You can also imagine this as a noisy broadcasting process where people are on the nodes of the network and you're playing a big game of Telephone, where people whisper the secret to their neighbors and errors accumulate over time.)

Problem 3.2 (Reconstruction)

For $n \in \mathbb{N}$, let $L(n) = \{\text{vertices at distance } n \text{ from the root}\}$.

Goal: Given an assignment $\tau : L(n) \to \{\pm 1\}$, design an estimator for the root.

More concretely, a *estimator* is a function on asignments — a function $\widehat{\sigma}_r : \{\pm 1\}^{L(n)} \to \{\pm 1\}$. This says, if I see a certain assignment for the vertices at level n, I should output whether my root was 1 or -1. More generally, we can allow our estimator to be randomized: then we consider a function $\widehat{p} : \{\pm 1\}^{L(n)} \to [0, 1]$ (where given $L(n)$, our guess for σ_r is the outcome of a coin toss with bias $\widehat{p}(\sigma_{L(n)})$).

So I only tell you information for the vertices at distance n, and I want you to recover the original assignment for the root vertex.

Definition 3.3. For an estimator $\widehat{\sigma}_r$, we define its probability of success as

$$\mathbb{P}_\sigma[\widehat{\sigma}_r(\sigma_{L(n)}) = \sigma_r].$$

In other words, this is the probability our estimator is correct over a random assignment with respect to this process, as well as with respect to the randomness of the estimator itself.

(In general, for a set of vertices S, we write σ_S for the restriction of σ to S.)

Definition 3.4. We define the advantage of the estimator as the quantity $b(\widehat{\sigma}_r)$ where

$$\mathbb{P}[\widehat{\sigma}_r(\sigma_{L(n)}) = \sigma_r] = \frac{1 + b(\widehat{\sigma}_r)}{2}.$$

(There's always a trivial estimator which guesses completely randomly, and achieves success probability $\frac{1}{2}$; so b quantifies the *amount* by which we beat $\frac{1}{2}$.)

Definition 3.5. We define $b^* = \sup_{\widehat{\sigma}_r} b(\widehat{\sigma}_r)$.

So this is the best possible advantage you could achieve, where we're taking a sup over all possible estimators. (We allow computationally unbounded estimators — so you can use exponential time algorithms or unbounded computations or whatever you want.)

Remark 3.6. You can equivalently think of this as a sup of $b(\widehat{p})$ over all functions $\widehat{p} : \{\pm 1\}^{L(n)} \to [0, 1]$ (where your estimator outputs a number and then tosses the coin with that number).

Our goal will be to study this quantity, as a function of the level of noise ε. (Of course b^* depends on n, so we might write b_n^*.)

§3.3 Phase transition for reconstruction

The theorem we'll discuss establishes a sharp phase transition for this problem.

Theorem 3.7

Fix $d \in \mathbb{N}$ and define $\theta = 1 - 2\varepsilon$ (for $0 \le \varepsilon \le \frac{1}{2}$). Let θ_c satisfy $d\theta_c^2 = 1$ (i.e., $\theta_c = d^{-1/2}$ and $\varepsilon_c = \frac{1}{2} - \frac{1}{2}d^{-1/2}$). Then we ehave

$$\lim_{n\to\infty} b_n^* \begin{cases} = 0 & \text{if } \varepsilon > \varepsilon_c \\ > 0 & \text{if } \varepsilon < \varepsilon_c. \end{cases}$$

Let's start by getting a qualitative understanding of what's happening. Of course, if $\varepsilon = \frac{1}{2}$, then I'm totally ignoring what my parent is — I'm not broadcasting anything, just every vertex gets a uniformly random assignment — so you shouldn't be able to get anything (the vertices don't tell you any information about what the root was). This says that if we're any larger than something a bit less than $\frac{1}{2}$, then there's no estimator that significantly beats the trivial one. But the amazing thing is that once you go below this threshold, you can beat the trivial estimator by a nontrivial amount.

This dependence on d kind of makes sense, because if I have more vertices to broadcast to, then there's more vertices to pick up the signal — so the larger d is, the easier I expect the problem to be, which is why the threshold becomes closer to $\frac{1}{2}$.

§3.3.1 Branching processes perspective

Let's try to understand why this phase transition comes about. The intuition goes back to the notion of branching processes from the previous lecture — we can actually predict that this phase transition is going to occur using branching processes (this perspective will also explain why we parametrize things in terms of $\theta = 1 - 2\varepsilon$).

We're going to switch how we think about how this process is generated. An equivalent formulation is as follows:

(1) For the root, we still generate $\sigma_r \sim \mathsf{Unif}\{\pm 1\}$.

(2) For the recursive steps, rather than preserving vs. flipping the assignment of the parent, we'll think of preserving vs. *erasing* information. So for each child u of r independently, we set $\sigma_u = \sigma_r$ with probability $1 - 2\varepsilon$; otherwise we set $\sigma_u \sim \mathsf{Unif}\{\pm 1\}$ with probability 2ε. (Then there's a $\frac{1}{2}$ probability that I get lucky and get the same assignment as my parent.)

So either I'm preserving information — directly copying the assignment of my parent — or completely erasing the information, totally ignoring the parent and sampling a fresh bit.

If you have this perspective, now we can augment our above diagram slightly; or we can think of it as follows. Now we can imagine not just labelling the vertices of the graph, but also labelling the edges, where we label an edge 1 if we're in the copying case (where we directly copy the assignment of our parent) and a 0 if we erase information (where we erase the assignment of the parent).

Once you have this perspective, you can see there's a branching process happening here, for the edges that are labelled 1. Let

$$Z_n = \#(\text{vertices at distance } n \text{ with a path of 1-edges to the root}).$$

These are the set of vertices that directly copy the root assignment — along the entire path, we're just copying and copying and copying. So Z_n is the number of vertices in $L(n)$ which 'receive' the signal.

In particular, $\{Z_n\}_{n\in\mathbb{N}}$ is a Galton–Watson branching process with offspring distribution $\mathsf{Binom}(d, \theta)$ (where $\theta = 1 - 2\varepsilon$). We saw in the previous lecture that there's a phase transition for the extinction probability at $\theta = 1/d$. If the process goes extinct, that means after a certain point there's no signal left — none of the vertices at that point directly copied the assignment of the root. On the other hand, if this branching process doesn't go extinct, at least one vertex at level n received the signal.

But this doesn't quite match the statement of our theorem, where we have $d^{-1/2}$ instead of d^{-1}. There *is* a phase transition that happens here, but it's of a very different kind (and we'll discuss it later). This is for good reason — at distance n from the root, the total number of vertices is d^n, which is huge. So even if you have one vertex that directly copied from the root (if your process didn't go extinct, all you can say is at least one vertex got the signal), that could be drowned out from all the noise of the $d^n - 1$ other vertices that got independent random bits.

So what you really want is for there to be *many* vertices that got the signal.

Now let's do a heuristic calculation for how many vertices you would need. To do this thought experiment, imagine two worlds. In the first world (World 1, sometimes called the *null hypothesis*), imagine all vertices at level n completely ignore the root — everyone gets a freshly randomly generated bit, so we have IID $\sigma_u \sim \mathsf{Unif}\{\pm 1\}$ for all $u \in L(n)$.

And the second world (World 2) is our world, where $\sigma_{L(n)}$ is generated from our process.

The intuition is that there's going to be some correlations — if the root is $+1$, I expect to see more $+1$'s than -1's in the vertices of $L(n)$, and if the root is -1, I expect to see more -1's than $+1$'s. So let's now compare the difference in expectations between these two things.

In World 1, the expected number of vertices in $L(n)$ with the same assignment as the root is

$$\mathbb{E}[\#\{u \in L(n) \mid \sigma_u = \sigma_r\}] = \frac{1}{2} d^n.$$

On the other hand, if we look at this number in World 2, we have $\mathbb{E}[Z_n] = d^n \theta^n$ (where Z_n is the number of vertices that receive the signal); equivalently,

$$\mathbb{E}[\#\{u \in L(n) \mid \sigma_u = \sigma_r\}] = d^n \theta^n + \frac{1}{2} d^n (1 - \theta^n)$$

(θ^n is the number of vertices that directly copy from the root, and the other $d^n(1-\theta^n)$ vertices get uniformly random assignments, so they have $\frac{1}{2}$ probability of matching the root).

If I want to distinguish between these worlds, not only do I want these expectations to be different, but I want them to be *significantly* different — I want the difference to be larger than the typical fluctuations of the World 1 random variable from its expectation (because otherwise I could have reasonably explained the size of Z_n using the first world, rather than the second world).

So we want the difference in expectations to exceed the *standard deviation*, or the typical size of the fluctuations, under the null hypothesis.

The point is when you're comparing two distributions, it's natural to compare their expectations; but to distinguish, you want the difference to not just be positive, but to in fact be *large* with respect to the typical deviations of one of them.

If I look at the standard deviation in World 1, it's on the order of $d^{n/2}$ (I have a bunch of uniformly random ± 1's, so the typical deviation is on the order of the square root of the number of them). This means to be able to distinguish, I want

$$d^n \theta^n > d^{n/2},$$

or equivalently $(d\theta^2)^{n/2} > 1$ or $d\theta^2 > 1$. So roughly, this is the kind of heuristic.

Student Question. *Should we also consider the standard deviation of the branching process?*

Answer. Ultimately we will have to do something like this (we'll have to consider one distribution where the root is $+1$ and another where the root is -1); this is just a back-of-the-envelope to get some sense of what behavior we'd expect.

§3.3.2 Reinterpreting b^*

We'll start by reinterpreting b^*. In the interest of time, we'll focus on the second part of the theorem — that for $\varepsilon > \varepsilon_c$ we can get a nontrivial advantage. (The proof of the first part is in the notes, but we won't have time to discuss it.)

For convenience, we'll define two distributions — μ_n^+ is the distribution of $\sigma_{L(n)}$ conditioned on $\sigma_r = +1$, and μ_n^- is the distribution of $\sigma_{L(n)}$ conditioned on $\sigma_r = -1$, i.e.,

$$\mu_n^+ = \mathsf{Law}(\sigma_{L(n)} \mid \sigma_r = +1),$$
$$\mu_n^- = \mathsf{Law}(\sigma_{L(n)} \mid \sigma_r = -1).$$

(We use Law to denote the distribution, or law, of a random variable.)

So we consider these two distributions, and our goal is to distinguish between them. The *total variation (TV) distance* is defined by

$$\left\|\mu_n^+ - \mu_n^-\right\|_{\mathsf{TV}} = \frac{1}{2}\sum_{\tau} \left|\mu_n^+(\tau) - \mu_n^-(\tau)\right|.$$

(This measures how close the two distributions are; τ ranges over all possible assignments, i.e., $\tau \in \{\pm 1\}^{L(n)}$.)

Proposition 3.8

The quantity b_n^* is exactly equal to $\|\mu_n^+ - \mu_n^-\|_{\mathsf{TV}}$.

So we have two perspectives on b_n^* — one as a L^1 distance between two distributions, and one as an optimization problem about the best possible estimator.

We'll actually prove a slightly more general fact:

Lemma 3.9

If μ and ν are two distributions on a common state space Ω and we define

$$\|\mu - \nu\|_{\mathsf{TV}} = \frac{1}{2}\sum_{\omega \in \Omega} |\mu(\omega) - \nu(\omega)|,$$

then we have

$$\|\mu - \nu\|_{\mathsf{TV}} = \sup_{f:\Omega \to [0,1]} |\mathbb{E}_{\omega\sim\mu}[f(\omega)] - \mathbb{E}_{\omega\sim\nu}[f(\omega)]| = \sup_{A \subseteq \Omega} |\mathbb{P}_\mu[A] - \mathbb{P}_\nu[A]|.$$

So in the second term, we're looking at the best possible distinguishing function f, and taking its difference in expectations under the two distributions.

We're just going to prove this lemma; and this lemma immediately implies the previous proposition (the best test function f is your best estimator for distinguishing between the two distributions).

Proof. For simplicity, let's assume Ω is finite. (This is okay for our application.) Now we can think of μ, ν, and f as being some big vectors in $\mathbb{R}^\Omega$ (or $\mathbb{R}^n$ where $n = |\Omega|$). If we look at this difference of expectations, we can write it linear-algebraically as an inner product — we have

$$|\mathbb{E}_{\omega\sim\mu}[f(\omega)] - \mathbb{E}_{\omega\sim\nu}[f(\omega)]| = |\langle \mu - \nu, f\rangle|.$$

And now we're optimizing this functional over the cube — i.e., over all $f \in [0,1]^\Omega$.

So I have a linear functional (or the absolute value of one, which is something convex) and I'm maximizing it over a polytope (here, a cube). And the optimizers are attained at an extreme point (i.e., a vertex); so this is going to be equal to

$$\sup_{f\in\{0,1\}^\Omega} |\langle u-v, f\rangle|.$$

(In general, if you're maximizing a convex function over a polytope — a subset of $\mathbb{R}^n$ defined by some finite collection of linear inequalities — it achieves optimality at a vertex. In our case, the function is just linear, so you can imagine just running gradient descent or something until you hit a vertex.)

And any such function $f \in \{0,1\}^\Omega$ is basically just an indicator of a set. So this is equal to

$$\sup_{A\subseteq\Omega} |\mathbb{P}_\mu[A] - \mathbb{P}_\nu[A]|,$$

which proves the second equality.

Now we're going to show that this is equal to $\|\mu-\nu\|_{\mathsf{TV}}$. Let's fix some event $A \subseteq \Omega$. Now we can write

$$|\mathbb{P}_\mu[A] - \mathbb{P}_\nu[A]| = \frac{1}{2}\left|\sum_{\omega\in A}(\mu(\omega)-\nu(\omega))\right| + \frac{1}{2}\left|\sum_{\omega\in\Omega\setminus A}(\mu(\omega)-\nu(\omega))\right|$$

(since if I replace A with $\Omega\setminus A$, I get the same value for $|\mathbb{P}_\mu[A]-\mathbb{P}_\nu[A]|$ — both get replaced with 1 minus their original value). And by the triangle inequality, this is at most

$$\frac{1}{2}\sum_{\omega\in\Omega}|\mu(\omega)-\nu(\omega)| = \|\mu-\nu\|_{\mathsf{TV}}.$$

To prove equality, I just need to exhibit some choice of A that achieves equality. The natural choice of A is the one which makes this inequality into an equality — we just want to arrange it so that all these signs go in the right direction. So we take

$$A = \{\omega\in\Omega \mid \mu(\omega) > \nu(\omega)\}.$$

On this set, all these triangle inequalities are tight, so we get equality. □

§3.3.3 Proof of the reconstructible case

Now here's the plan for how we're going to prove the theorem. We know that $b_n^* = \|\mu_n^+ - \mu_n^-\|_{\mathsf{TV}}$. This lemma in particular tells us a very good estimator for our task — the *optimal* estimator looks like

$$\widehat{\sigma}_r = \begin{cases} +1 & \text{if } \mu_n^+(\sigma_{L(n)}) > \mu_n^-(\sigma_{L(n)}) \\ -1 & \text{otherwise.}\end{cases}$$

In other words, I look at the assignment I see, and compare the probabilities of seeing it under the two possible assignments to the root. This is called the *maximum likelihood estimator*; its advantage is exactly the total variation distance.

This estimator can be a bit difficult to analyze, so we'll analyze an easier estimator which is suboptimal, but sufficient for lower bounds. The estimator we're going to use is actually something quite natural — if the root vertex was assigned +1, then we'd naturally expect to see more +1's on vertices at distance n. So a natural estimator to use is just the *majority vote estimator* (this is actually different from the maximum likelihood estimator), defined by

$$\widehat{\sigma}_r^{\mathsf{MAJ}} = \operatorname{sgn}\left(\sum_{u\in L(n)}\sigma_u\right).$$

So I just add up all the assignments for the vertices at distance n, and take their sign — if I see more $+1$'s than -1's I'll output $+1$, and if I see more -1's than $+1$'s I'll output -1.

In particular, we can lower bound b_n^* by the performance of *this* estimator, and this is the one we're going to analyze — we have $b_n^* \geq b(\widehat{\sigma}_r^{\mathsf{MAJ}})$.

Exercise 3.10. Letting $S_n = \sum_{u \in L(n)} \sigma_u$, we have

$$b(\widehat{\sigma}_r^{\mathsf{MAJ}}) = \|\mathsf{Law}(S_n \mid \sigma_r = +1) - \mathsf{Law}(S_n \mid \sigma_r = -1)\|_{\mathsf{TV}}.$$

So our goal is now to lower-boudn the right-hand side of this. For this, we're going to use the second moment method. So we need one more lemma.

Lemma 3.11

Let X and Y be $\mathbb{R}$-valued random variables. Let W be the new random variable with law

$$\mathsf{Law}(W) = \frac{1}{2}\mathsf{Law}(X) + \frac{1}{2}\mathsf{Law}(Y).$$

Then we have

$$\|\mathsf{Law}(X) - \mathsf{Law}(Y)\|_{\mathsf{TV}} \geq \frac{1}{4}\frac{(\mathbb{E}[X] - \mathbb{E}[Y])^2}{\mathrm{Var}[W]}.$$

So I start with two random variables X and Y, and produce a new random variable W where I toss a coin, and if it lands heads I take X, and if it lands tails I take Y.

First we'll show how to use this lemma to lower bound our total variation distance by a constant as $n \to \infty$; if we have time (which we probably won't) then we'll prove this lemma.

Let $X_n = (S_n \mid \sigma_r = +1)$ and $Y_n = (S_n \mid \sigma_r = -1)$ be our signed sums conditioned on the root being $+1$ and -1 (respectively), and $W_n = S_n$. We're going to use this lemma, so we need to be able to compute these quantities, i.e., $\mathbb{E}[X]$, $\mathbb{E}[Y]$, and $\mathrm{Var}[W]$.

First, we have

$$\mathbb{E}[X_n] = \mathbb{E}[\#(\text{vertices directly copying the root})] = d^n\theta^n$$

(since all the other variables get a fresh uniformly random bit, which will have expectation 0). And by symmetry, $\mathbb{E}[Y_n] = -d^n\theta^n$. Now all we have to do is compute $\mathrm{Var}[S_n]$. And for this, we're going to directly calculate it — we have

$$\mathrm{Var}[S_n] = \mathbb{E}[S_n^2] = \sum_{u,v \in L(n)} \mathbb{E}[\sigma_u\sigma_v].$$

Now, how do you calculate this thing? You have to look at how correlated these two assignments are. And how correlated they are depends on the *least common ancestor* between u and v — if we take u and v, where's the first vertex in their shared family tree? That's sort of the main source of the correlations between the two assignments. So if we want to compute this thing, we should stratify it by how far the least common ancestor is. So then this is equal to

$$\sum_u \mathbb{E}[\sigma_u^2] + \sum_{u \in L(n)} \sum_{\ell=0}^{n-1} \sum_{v : \mathsf{LCA}(u,v) \in L(\ell)} \mathbb{E}[\sigma_u\sigma_v]$$

(where $\mathsf{LCA}(u,v)$ is the least common ancestor, and ℓ denotes its distance from the root). And once I fix how far away this common ancestor is, I can directly compute this — then we end up having

$$\mathbb{E}[\sigma_u\sigma_v] = \theta^{2(n-\ell)}.$$

The way to see this is that in order for these two assignments to be the same, I want these two vertices to copy every step along the way from the least common ancestor; the distance from each is $n - \ell$, and I have two sides (for u and v), which is why there's $2(n - \ell)$ in the exponent.

This is a sketch of the key calculations for how you'd go about lower-bounding b_n^*. As a quick word on how to prove this lemma, it's not very difficult; it basically boils down to a very nice Cauchy–Schwarz. The precise calculations are in the notes.

§4 February 12, 2025 — Lovász Local Lemma

Today we're going to switch gears slightly. In particular, we won't look at models based on graphs anymore, but something else. Last lecture was a bit rushed, so if we have time at the end today, we'll go over the proof of the second moment lower bound for total variation distance that we didn't do last lecture.

Today's lecture is about a really cool tool called the Lovász local lemma. The high-level theme behind this tool is *beating the union bound*. This is the rough setup for when this tool is particularly useful: Imagine you have a whole bunch of events $A_1, \dots, A_m$, and you want to certify that the probability none of the complements occur is positive, i.e., $\mathbb{P}[\bigcap_i \overline{A_i}] > 0$. (This goes hand in hand with probabilistic method applications.)

§4.1 Boolean satisfiability

To make our discussion concrete, we'll focus on a specific model that comes up particularly in computer science applications, which is Boolean satisfiability. So for the bulk of this lecture, we'll be studying this problem and illustrating how to use the Lovász local lemma (which we'll abbreviate LLL) there.

We've probably seen what SAT is before, but we'll refresh our memory on what it involves. It's sort of the problem that underpins the entire theory of NP-completeness, which is a useful way of classifying problems as 'easy' or 'hard.'

The basic setup behind SAT is the following: You have a collection of Boolean-valued variables $x_1, \dots, x_n$ (Boolean-valued means that it takes values in $\{\mathsf{T}, \mathsf{F}\}$). We'll define a couple of pieces of notation:

Notation 4.1. For a variable x, its *negation*, written $\neg x$, is its opposite.

So if $x = \mathsf{T}$ then $\neg x = \mathsf{F}$, and vice versa.

Definition 4.2. A *literal* is a symbol of the form x or $\neg x$.

So a literal is a variable or its negation.

Definition 4.3. A *clause* is an OR of literals.

Example 4.4

We might have a clause that looks like $x_1 \vee \neg x_7 \vee x_{10}$.

So when you plug in T-F assignments to the variables of this clause, you'll get out T or F.

Definition 4.5. A *CNF-formula* $\Phi = (\mathcal{V}, \mathcal{C})$ is a collection of variables $\mathcal{V}$ and an AND of clauses in $\mathcal{C}$.

Example 4.6

An example of a CNF is $\Phi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_1)$.

Definition 4.7. We say an assignment $x \in \{\mathsf{T}, \mathsf{F}\}^{\mathcal{V}}$ *satisfies* a clause C (or a formula Φ) if C (or every clause in Φ) evaluates to T under the assignment x.

So x satisfies a CNF-formula if it satisfies *every* clause in the formula.

Definition 4.8. A formula is *satisfiable* if it has at least one satisfying assignment.

Example 4.9

The formula Φ in Example 4.6 has the all-true satisfying assignment — we just need one literal in each clause to be true, and this works. In fact, the all-false assignment also satisfies Φ.

Example 4.10

A clause with k literals has $2^k - 1$ satisfying assignments — there's only one way to make it *not* satisfied, which is to ensure every literal evaluates to F.

In the SAT problem, you're given such a formula Φ, and you want to find a satisfying assignment (or even weaker, just decide whether or not the assignment is satisfiable).

Problem 4.11 (SAT)

Given Φ, find a satisfying assignment (or decide it's unsatisfiable).

This is one of the classic very hard problems in computer science; as we mentioned earlier, it underpins the entire theory of NP-completeness and so on. So let's study easier cases.

Student Question. *What does CNF stand for?*

Answer. Conjunctive normal form.

The complexity of this problem really depends on how you represent this formula. For example, if you have a OR of a bunch of ANDs instead, that's called DNF form, and there's actually a polynomial-time algorithm to find a satisfying assignment. But if it's given in CNF form, it's actually quite hard (or we have good evidence that it's hard). So given that it's hard, we're going to look at slightly easier special cases to see where exactly hardness sets in.

Let's look at a slightly restricted version of this problem:

Definition 4.12. We say a formula Φ is *k-uniform* if every clause has exactly k variables.

For example, the formula in Example 4.6 is 2-uniform, because every clause has exactly 2 variables.

(We have 'variables' instead of 'literals' to avoid things like $x_1 \vee x_1 \vee x_1$.)

As it turns out, even if you restrict the formula to be k-uniform for any $k \geq 3$, the SAT problem remains just as hard as the full problem (without this uniformity constraint).

If you have a hard problem, there are a few natural questions.

Question 4.13. Can we efficiently find an assignment satisfying *many* clauses?

Maybe you don't get to satisfy all of them, but you try to satisfy as many as you can.

Question 4.14. What additional structure can we impose to make the problem tractable?

§4.2 Approximate assignments for SAT

Proposition 4.15

For any 3-uniform CNF formula Φ, there is an assignment satisfying at least $\frac{7}{8}m$ clauses.

(Here we're not assuming Φ is satisfiable or not.) We always use m to denote the number of clauses.

You can also generalize this to larger k.

Proof. This is a classic application of the probabilistic method — we have a problem, so rather than being clever and trying to find a solution deterministically, we just pick a random one and show with positive probability it satisfies this many constraints. And we can do that just by computing the *expected* number of satisfied clauses.

So we pick a uniformly random assignment $x \sim \mathsf{Unif}\{\mathsf{T},\mathsf{F}\}^{\mathcal{V}}$. Every clause has 2^3 possible assignments, and only one out of those 2^3 will not satisfy the clause; so each clause has a $\frac{1}{8}$ probability of *not* being satisfied, or a $\frac{7}{8}$ probability of being satisfied. In particular, the expected number of satisfied clauses is $\frac{7}{8}m$, so there must be at least one assignment that satisfies that many clauses. □

There also turns out to be an efficient algorithm to find such an assignment (even deterministically; this is not unrelated to one of our homework problems).

Remark 4.16. In fact, it's a very cool result in TCS that says $\frac{7}{8}$ is actually optimal in some sense — there's a kind of computational phase transition that occurs in the approximability of this problem. Specifically, it's known that for every $\varepsilon > 0$, finding an assignment satisfying at least $(\frac{7}{8}+\varepsilon)m$ clauses is as hard as just solving SAT in general (i.e., it's NP-hard). (This is a result of Hastad from the 1990s, building on a lot of beautiful work in TCS, in particular a result called the PCP theorem.) So there's a very cool result that sort of matches this.

§4.3 SAT with additional structure

That's all we're going to say about Question 4.13; now let's turn to Question 4.14. And this is where the new tool, the Lovász local lemma, is going to come in. Here's the theorem we want to prove.

Theorem 4.17

Let Φ be a k-uniform CNF-formula such that for every variable x, there are at most $2^k/4k$ clauses containing x. Then Φ is satisfiable.

So we're imposing the additional constraint that no variable participates in too many clauses — no variable appears too many times. Already if k is something reasonable, like 10, this is just saying no variable occurs in more than 25 clauses or something like that. But in particular, you can allow any number of clauses, as long as no variable occurs too many times.

And the claim is that if we have this additional restriction, then Φ is satisfiable.

Remark 4.18. In fact, it's also known that there is an efficient 'stochastic local search' algorithm to find such an assignment. This is a big breakthrough from around 15 years ago by Moser and Tardos. The algorithm is really simple to state, but the analysis is novel and was a big breakthrough. The algorithm is to start with a uniformly random assignment; and while there's some unsatisfied clause, you pick one of them and re-randomize all the variables inside it. You just run this until you find a satisfying assignment. So the algorithm is simple; the big breakthrough was showing this thing terminates in expected polynomial time. (We won't discuss this.)

§4.4 A proof attempt

Let's try the same strategy as we did for proving the earlier proposition — let's start by picking a uniformly random assignment. So we pick $x \sim \mathsf{Unif}\{\mathsf{T},\mathsf{F}\}^{\mathcal{V}}$. What tools do we have at our disposal? Imagine I want to look at the event that a given clause is satisfied or not; so let A_i be the event that clause C_i is not satisfied. We think of this as a *bad* event that you want to avoid. And our goal is to show that the probability we avoid all the bad events is positive, i.e., that $\mathbb{P}[\bigcap_i \overline{A_i}] > 0$ (where $\overline{A_i}$ means the clause is satisfied, and we want all the clauses to be satisfied).

In particular, for this kind of thing, it's not going to be enough to compute an expectation — saying the expected number of satisfied clauses is some fraction of m wouldn't be useful, since we want *every* clause to be satisfied.

Besides that, what do we have at our disposal?

If $A_1, \dots, A_m$ were independent, then of course this probability would just be $\prod_{i=1}^m \mathbb{P}[\overline{A_i}] > 0$, in which case that would be quite easy (we would just need to ensure each individual clause is satisfiable, which is true). But of course, this is a collection of events which are *not* independent — if two clauses share some variables, then if I tell you one of the clauses is satisfied or not, it'll definitely influence whether or not the other one is satisfied. So unfortunately, we don't have independence.

On the other hand, what we *could* try to do is a union bound type of argument — a union bound doesn't require *anything* about how these events are related to each other. So we can try to union bound; and then we need to show that $\sum_{i=1}^m \mathbb{P}[A_i] < 1$. (The complement of $\bigcap_i \overline{A_i}$ is $\bigcup_i A_i$, and to show that $\mathbb{P}[\bigcap_i \overline{A_i}]$ is positive, we need the complement to have probability strictly less than 1.)

And we know each of these probabilities — it's 2^{-k}. So we'd need $k > \log_2 m$ for this to work — we'd need the size of the clauses to be sufficiently large. This is very different from the kind of assumption we started with — where the size of each clause could still be some constant. So this is also no good.

The new tool, the Lovász local lemma, is what'll allow us to get around this.

§4.5 The Lovász local lemma

Basically the key feature of this assumption we want to capture is that if each variable doesn't appear in too many clauses, not too many clauses can depend on each other. So even though there are some dependencies, the amount of dependency is not too much.

To formalize this:

Definition 4.19. Given a collection of events $A_1, \dots, A_m$, we say that A_i is *mutually independent* of $\{A_j \mid j \in J\}$ (for some index set $J \subseteq [m]$) if for all $J' \subseteq J$, we have

$$\mathbb{P}\left[A_i \cap \bigcap_{j \in J'} A_j\right] = \mathbb{P}[A_i] \cdot \mathbb{P}\left[\bigcap_{j \in J'} A_j\right].$$

What we're eventually going to use is that if two clauses don't share any variables whatsoever, they're going to be independent of each other.

Definition 4.20. We say a graph $G = ([m], E)$ is a *dependency graph* for a collection of events $\{A_i\}_{i=1}^m$ if A_i is mutually independent of $\{A_j \mid j \notin N[i]\}$.

Notation 4.21. We write $N(i) = \{j \mid ij \in E\}$ to denote the *neighborhood* of i and $N[i] = N(i) \cup \{i\}$ to denote the *closed neighborhood* of i.

Notation 4.22. We write $[m] = \{1, \dots, m\}$.

In words, this means A_i is mutually independent of all the events not in its neighborhood.

For example, the complete graph is always a dependency graph for any collection of events. But it could be that there's a much sparser dependency graph for some collection of events; and the sparsity of the dependency graph is going to be what's relevant for us.

Theorem 4.23 (Lovász local lemma)

Suppose a collection of events $A_1, \dots, A_m$ has a dependency graph of maximum degree d, and assume that $\mathbb{P}[A_i] \le 1/e(d+1)$ for all i. Then $\mathbb{P}[\bigcap_i \overline{A_i}] > 0$.

So we have a collection of events with a *sparse* dependency graph. And we assume that the probabilities of the A_i's aren't too large; but the bound only depends on d. And this is enough to get what we want — that the probability of avoiding all of them is positive.

Remark 4.24. This is the *symmetric* version of the LLL; there are lots of more general versions, but this is the most user-friendly.

Student Question. *What is e?*

Answer. It's the Euler constant (i.e., $e \approx 2.71$).

Sometimes it's useful to say $1/e(d+1) \le 1/4d$. But the key takeaway is that you have an upper bound that only depends on d, not the number of vertices m (which is much, much larger).

§4.6 Application to k-uniform CNFs

Before we prove this statement, let's very quickly apply it to our k-uniform CNF setup.

As before, we're pickign $x \sim \mathsf{Unif}\{\mathsf{T}, \mathsf{F}\}^{\mathcal{V}}$, and A_i is the event that the clause C_i is not satisfied. What's a good dependency graph? Let's connect $A_i \sim A_j$ if C_i and C_j share a variable. (This is the only way they can be dependent on each other.)

Now, I've assumed that every variable is contained in at most $2^k/4k$ clauses. So what kind of degree bound can I get on this dependency graph? We get

$$\max \deg \le \frac{2^k}{4k} \cdot k = \frac{2^k}{4},$$

which we'll call d. Basically, we're looking at some clause, and all the other clauses that intersect me. If they intersect me, they must intersect at one variable; there are k variables to choose from, and each leads to at most $2^k/4k$ other clauses.

On the other hand, we also know that the probability of each of these events is $\mathbb{P}[A_i] = 2^{-k}$, which is exactly $1/4d$. So we can invoke LLL as a black box and get that our formula is satisfiable.

Student Question. *Why is the maximum degree $2^k/4$?*

Answer. I'm at some clause, and I want to upper-bound the number of clauses that intersect me. Imagine that some other clause intersects me. There must be some variable responsible for the intersection; there are k variables inside me, and each variable can be responsible for at most $2^k/4k$ clauses.

§4.7 Proof of the Lovász local lemma

Now let's prove the LLL. We don't have independence, but we can always sort of factorize the probability we want as a product of *conditional* probabilities — we can write

$$\mathbb{P}\left[\bigcap_{i=1}^m \overline{A_i}\right] = \prod_{i=1}^m \mathbb{P}\left[\overline{A_i} \,\middle|\, \bigcap_{j=1}^{i-1} \overline{A_j}\right] = \prod_{i=1}^m \left(1 - \mathbb{P}\left[A_i \,\middle|\, \bigcap_{j=1}^{i-1} \overline{A_j}\right]\right).$$

So basically, we've assumed the *unconditional* probabilities of the A_i's are bounded by some constant. And our goal is to show that if we have a bounded-degree dependency graph, then this probability doesn't degrade too much when we introduce this conditioning.

Student Question. *Why are the things in the conditioning all the ones before i? What if A_i is dependent on A_{i+5}?*

Answer. This expansion isn't about which is dependent on what (it's definitely possible A_i depends on something after). It's just that I can order the events arbitrarily and expand the probability in this way. This is kind of just the definition of conditional probabilities — if I have three events, I can always write

$$\mathbb{P}[A_1 \wedge A_2 \wedge A_3] = \mathbb{P}[A_1]\mathbb{P}[A_2 \wedge A_3 \mid A_1] = \cdots.$$

In particular, the key lemma is the following:

Lemma 4.25

For all i and all $J \subseteq [m] \setminus \{i\}$, we have

$$\mathbb{P}\left[A_i \,\middle|\, \bigcap_{j \in J} \overline{A_j}\right] \leq \frac{1}{d+1}.$$

So really this is all we have to prove — that the fact I have bounded dependence means it doesn't degrade my probability bound by too much, even if I insert some conditioning.

Proof. We'll go by (increasing) induction on $|J|$, the size of the set we're conditioning on. The base case is when $|J| = 0$, i.e., $J = \emptyset$; here this is true by assumption.

So now let's do the inductive step. Now I have a set J, and I have some events that I'm dependent on and some events that I'm not dependent on; so let's split J into these two pieces. So we let

$$J_{\text{indep}} = J \setminus N[i] \quad \text{and} \quad J_{\text{dep}} = J \cap N(i).$$

The second is all the events in J that are dependent with A_i, and the first is all the remaining.

We can always assume J_{dep} is nonempty — if it's empty, then we're just conditioning on a bunch of independent stuff, and we can just pull it out (the conditional probability is the same as the unconditional one). So we can assume without loss of generality that $J_{\text{dep}} \neq \emptyset$.

Now let's look at our probability

$$\mathbb{P}\left[A_i \,\middle|\, \bigcap_{j\in J} \overline{A_j}\right].$$

We want to now write this in terms of conditional probabilities where I've reduced the size of J, so that I can invoke the inductive hypothesis. So let's use Bayes's rule; this is equal to

$$\frac{\mathbb{P}\left[A_i \mid \bigcap_{j\in J_{\text{dep}}} \overline{A_j} \mid \bigcap_{j\in J_{\text{indep}}} \overline{A_j}\right]}{\mathbb{P}\left[\bigcap_{j\in J_{\text{dep}}} \overline{A_j} \mid \bigcap_{j\in J_{\text{indep}}} \overline{A_j}\right]}$$

(this is just Bayes's rule; we haven't done anything yet). For the numerator, we do something a bit silly: we have an intersection of a bunch of events, so we can just upper-bound it by one of them. So the numerator is *always* upper-bounded by

$$\mathbb{P}\left[A_i \,\middle|\, \bigcap_{j\in J_{\text{indep}}} \overline{A_j}\right].$$

And now I'm only conditioning on a bunch of events that are *independent* of me, so this is just equal to

$$\mathbb{P}[A_i] \le \frac{1}{e(d+1)}.$$

The key part of the argument, where we'll invoke our inductive hypothesis, is to lower-bound the denominator.

So what about the denominator? Now let's do the same thing we did with the expansion as a product — let's just enumerate the dependent events in some order, and expand out. So let's let $j_1, \ldots, j_t$ be the elements of J_{dep}; then the denominator is

$$\prod_{s=1}^{t}\left(1 - \mathbb{P}\left[A_{j_s} \,\middle|\, \bigcap_{j\in J_{\text{indep}}} A_j \cap \bigcap_{p=1}^{s-1} A_{j_p}\right]\right)$$

(we're conditioning on all the independent events, along with all the previous events).

And now the crucial thing is that this is an intersection of a bunch of events where the number of events is *strictly* less than $|J|$, so I get to apply my induction hypothesis; so each of these terms is at most $1/(d+1)$ by the inductive hypothesis, which means this entire thing is lower-bounded by

$$\left(1 - \frac{1}{d+1}\right)^t.$$

And for the last step of the proof, we have $t \le d$ (since t is some enumeration of the dependent events, and we assumed there's always at most d of them); so we can replace t by d, and by numerics

$$\left(1 - \frac{1}{d+1}\right)^d \ge \frac{1}{e}.$$

Now if I push this back, it cancels with the $1/e$ in the numerator, and that concludes the inductive step. □

§4.8 Second moment bound on total variation distance

That's all we'll say about LLL. Now we'll come back to something we didn't get to do in the previous lecture.

Recall that in the previous lecture, we had the following setup (from hypothesis testing): We have two distributions μ and ν on a common state space Ω. And the goal is, if you're given a sample from one of them but not told which one it came from, can you tell from the sample alone which distribution it was drawn from?

We saw the fundamental quantity related to how well you'll be able to distinguish is the *total variation distance* between them. One of the many equivalent definitions for it is

$$\|\mu - \nu\|_{\mathrm{TV}} = \frac{1}{2}\sum_{\omega\in\Omega} |\mu(\omega) - \nu(\omega)|\,.$$

We also saw, more relevantly for hypothesis testing, that this is equal to

$$\sup_{A\subseteq\Omega} |\mathbb{P}_\mu[A] - \mathbb{P}_\nu[A]| = \sup_{f:\Omega\to[0,1]} |\mathbb{E}_\mu[f] - \mathbb{E}_\nu[f]|\,.$$

The last one has a particularly nice interpretation in hypothesis testing — you're picking a function that will help you distinguish between the two distributions by looking at how different their expectations are, and you're sort of picking the *best* possible distinguisher for these distributions.

Student Question. *Can you construct f from A?*

Answer. Yes — we can take A and let f be the indicator function for that event. The point is that subsets correspond to functions with values in $\{0,1\}$ instead of $[0,1]$. And also, we saw that the optimal choice of A is the set of all elements which are more likely under μ than ν, i.e., $A = \{\omega \mid \mu(\omega) > \nu(\omega)\}$.

The lemma we didn't prove last time was a way to lower bound this quantity using the second moment method. We used this for the reconstruction problem for the broadcast process.

Lemma 4.26

Let X and Y be real-valued random variables, and let W be a random variable with

$$\mathsf{Law}(W) = \frac{1}{2}\mathsf{Law}(X) + \frac{1}{2}\mathsf{Law}(Y).$$

Then we have

$$\|\mathsf{Law}(X) - \mathsf{Law}(Y)\|_{\mathrm{TV}} \geq \frac{1}{4}\cdot\frac{(\mathbb{E}[X]-\mathbb{E}[Y])^2}{\mathrm{Var}(W)}.$$

Think of X and Y as the random numbers that pop out from applying f to a random variable drawn from either distribution (μ or ν). And we generate a new random variable W by tossing a fair coin; if it comes up heads I output X, and if it comes up tails I output Y. And then we can lower-bound the total variation distance based on the difference in expectations, and the variance of this new random variable W.

How we finished last lecture is for the reconstruction problem, we computed these quantities for the majority-vote estimator.

Now we'll show how to prove this lemma. There's a kind of converse to this lemma that's also in the notes.

Proof. For this proof, we'll start by using the L^1 version of total variation distance. (Later in the course we'll see a third equivalent definition for total variation distance. This is a feature, not a bug; one nice thing

about total variation distance is it admits many equivalent definitions and you can pick one that's more convenient for your application.) This tells us

$$\|\mathsf{Law}(X) - \mathsf{Law}(Y)\|_{\mathrm{TV}} = \frac{1}{2}\sum_w |\mathbb{P}[X = w] - \mathbb{P}[Y = w]|\,.$$

(For convenience we'll assume the random variables are discrete; otherwise you can replace this by an integral.)

Now let's sort of insert our random variable W. We can write this as

$$\frac{1}{2}\sum_{w\in\mathrm{supp}(W)} \frac{|\mathbb{P}[X = w] - \mathbb{P}[Y = w]|}{\mathbb{P}[W = w]}\mathbb{P}[W = w]$$

(if we want a lower bound, we can just restrict the sum to the support of W, the set of possible outputs for W).

So we've inserted this random variable W. And $\mathbb{P}[W = w]$ has a nice interpretation — this is

$$\frac{1}{2}\mathbb{P}[X = w] + \frac{1}{2}\mathbb{P}[Y = w].$$

So in particular, we can write

$$\|\mathsf{Law}(X) - \mathsf{Law}(Y)\|_{\mathrm{TV}} \geq \mathbb{E}[|f(W)|],$$

where we define the function

$$f(w) = \frac{\mathbb{P}[X = w] - \mathbb{P}[Y = w]}{\mathbb{P}[X = w] + \mathbb{P}[Y = w]}.$$

Now let's do some Cauchy–Schwarz, so we get the appearance of a second-moment-type quantity. One observation is that this function f automatically takes values in $[-1, 1]$, because we're taking the difference and dividing by the sum. So we can always lower-bound this by the expectation of the *square* of the function, meaning that

$$\mathbb{E}[|f(W)|] \geq \mathbb{E}[f(W)^2].$$

Now let's do some Cauchy–Schwarz — Cauchy–Schwarz says that

$$\mathbb{E}[W \cdot f(W)]^2 \leq \mathbb{E}[W^2]\mathbb{E}[f(W)^2].$$

Now let's move things to the other side; that gives a lower bound for $\mathbb{E}[f(W)^2]$, namely

$$\mathbb{E}[f(W)^2] \geq \frac{\mathbb{E}[W \cdot f(W)]^2}{\mathbb{E}[W^2]}.$$

The bottom is essentially the variance — what we should have done first is we can shift X, Y, and W by a constant (specifically, $\mathbb{E}[W]$) so that $\mathbb{E}[W] = 0$; that won't change the total variation distance between X and Y. So then $\mathbb{E}[W^2] = \mathrm{Var}[W]$. (We should have done this shifting at the very beginning of the proof.)

Now we just need to compute the numerator $\mathbb{E}[W \cdot f(W)]^2$. We have

$$\mathbb{E}[W \cdot f(W)] = \sum_w w \cdot \mathbb{P}[W = w] \cdot \frac{\mathbb{P}[X = w] - \mathbb{P}[Y = w]}{\mathbb{P}[X = w] + \mathbb{P}[Y = w]}.$$

The $\mathbb{P}[W = w]$ cancels with the denominator, up to a factor of 2 — the denominator is $2\mathbb{P}[W = w]$, so these are going to cancel, and we get

$$\frac{1}{2}\sum_w (w\mathbb{P}[X = w] - w\mathbb{P}[Y = w]) = \frac{1}{2}\mathbb{E}[X] - \frac{1}{2}\mathbb{E}[Y].$$

Now if I square, I get the $1/4$ and the $(\mathbb{E}[X] - \mathbb{E}[Y])^2$; and in the denominator I get $\mathrm{Var}[W]$. □

Student Question. *Could we use Chebyshev twice to prove something like this — if we used Chebyshev to bound* $\mathbb{P}[X \geq \mathbb{E}[X+Y]]$*?*

Answer. Kuikui thinks you might be able to do this, though he hasn't tried.

§5 February 18, 2025 — Shannon's noisy coding theorem

The first pset is due tonight; Kuikui will host OH immediately after this class in this room. The second pset is also posted on Canvas.

Today's lecture will be about an absolutely beautiful theorem called Shannon's noisy coding theorem. This is another really cool application of simple bare-hands methods, specifically the first and second moment methods. (Starting next lecture we'll talk about stronger concentration inequalities, but it's amazing you can prove such a theorem using just the second moment theorem.)

§5.1 Error-correcting codes

Before we state the result, we'll give some context. This result really comes from information theory and the theory of error-correcting codes. For the purposes of this lecture, an *error-correcting code* will be the following:

Definition 5.1. Fix $k, n \in \mathbb{N}$ with $k \leq n$. A *$[n,k]$-code* is a pair of functions $\mathsf{Enc}\colon \{0,1\}^k \to \{0,1\}^n$ and $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^k$.

We call these two functions the *encoder* and the *decoder*; we call elements of $\{0,1\}^n$ *codewords*, and elements of $\{0,1\}^k$ *messages*. In applications, you typically want additional assumptions (e.g., being easily computable, or linearity); but here we won't impose any additional assumptions.

The picture you should have is that there's a person with a message $m \in \{0,1\}^k$; they'll pass m through a box Enc and it'll produce a codeword. Then they'll send this codeword through a noisy channel, which may corrupt some of the bits you sent. Then on the other side, someone receives what you sent, and they're going to decode it to a message m'. And what you want is that $m' = m$ — so they're able to recover the message you wanted to send in the first place, despite there having been some errors in the transmission process.

Shannon's theorem is about the limits on this — how good an error-correcting code we can hope for under various models of the noise channel. For this lecture, we'll fix a simple and nice model which we've already seen — the binary symmetric channel.

Definition 5.2. Fix a noise parameter $0 \leq p < \frac{1}{2}$. The *binary symmetric channel* BSC_p sends each codeword $x \in \{0,1\}^n$ to $y \in \{0,1\}^n$ where for each coordinate independently,

$$y_i = \begin{cases} x_i & \text{with probability } 1-p \\ \neg x_i & \text{with probability } p. \end{cases}$$

For instance, when we looked at the broadcast process, we essentially put this channel on each edge of the tree — we introduced an error for each coordinate independently with probability p. (You can also consider richer channels, but for simplicity we'll focus on this one.)

Now we want to look at what it means for an error-correcting code to 'succeed' in some sense. Suppose we fix a message $m \in \{0,1\}^k$ that you want to send. We're looking at the following probabilities:

Definition 5.3. The *success probability* for m is

$$\mathbb{P}[\mathsf{Dec}(\mathsf{BSC}_p(\mathsf{Enc}(m))) = m].$$

The *failure probability* is $\mathbb{P}[\mathsf{Dec}(\mathsf{BSC}_p(\mathsf{Enc}(m))) \neq m]$.

So I first encode the message, then I send it through my noisy channel BSC_p, and then I look at what the probability is that if I decode this resulting noisy codeword, I get back the original message. (This probability is with respect to the noise of the channel; we're looking at a fixed error-correcting code.) We want the success probability to be as close to 1 as possible for every message.

§5.2 Shannon's theorem

There'll be a natural tradeoff between the *efficiency* of the codeword (how large n is with respect to k) and the success and failure probabilities. And Shannon's theorem gives a precise characterization of this tradeoff.

Definition 5.4. The *binary entropy function* is defined by $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$.

Definition 5.5. We define the *rate* of the code as $r = k/n$.

This quantifies how efficient the code is; the amount of redundancy you're adding to make your code more robust to errors.

Then Shannon's theorem says we have the following phase transition. (We parametrize things in terms of r and n, and k is a function of r and n.)

Theorem 5.6

- Suppose $r > 1 - H(p)$. Then for all n and every $[n,k]$-code, we have

$$\max_{m\in\{0,1\}^k} \mathbb{P}[\mathsf{Dec}(\mathsf{BSC}_p(\mathsf{Enc}(m))) \neq m] = 1 - o(1).$$

- Suppose $r < 1 - H(p)$. Then for every n, there exists a $[n,k]$-code such that

$$\max_{m\in\{0,1\}^k} (\text{failure probability of } m) \leq o(1).$$

So in the first case, we have a really efficient code, and the result says that there's no way we can guarantee reliable transmission on every message — the worst message has failure probability close to 1. And in the second case, we *can* guarantee reliable transmission.

Student Question. *What does r mean in terms of message transmission?*

Answer. For instance, imagine the error-correcting code where I take my message and duplicate every bit, in the hope that by adding redundancy I increase robustness to errors; then the rate is $\frac{1}{2}$. The more redundancy I add, the smaller r is, so the less efficient my code is; and I can hope to make the code more robust. And this theorem says when I can hope to get reliable transmission for every single message (in terms of the rate).

§5.3 Preliminaries

The amazing thing is this theorem uses really simple tools — the first and second moment methods and the probabilistic method — that we've already discussed.

Here's the first thing we'll use:

Theorem 5.7 (Weak law of large numbers)
Let $X_1, \dots, X_n$ be IID random variables with mean μ and variance $\sigma^2 < \infty$. Then

$$\lim_{n\to\infty} \mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| < \delta\right] = 1 \quad \text{for all } \delta > 0.$$

So we have IID random variables with finite variance, and we look at the probability that the empirical mean deviates from its expectation by more than δ; and that goes to 0. This is a very weak concentration statement for sums of independent random variables (we'll see much stronger ones next lecture), but this is enough for our purposes.

Proof. The proof is just Chebyshev's inequality — if we have a sum of independent random variables, the variance of their empirical mean is going to decay as $1/N$. □

Remark 5.8. The proof actually shows something slightly stronger — we get the conclusion even for $\delta \gg 1/\sqrt{N}$. So δ doesn't have to be a constant — it can be something decaying with N, as long as it decays slower than $1/\sqrt{N}$.

There's one more piece of notation we'll define — we need a notion of distance between codewords.

Definition 5.9. The *Hamming distance* between two codewords $x, y \in \{0,1\}^n$ is the number of coordinates at which they disagree, i.e.,

$$d_H(x,y) = \#\{i \mid x_i \neq y_i\}.$$

§5.4 The large-rate case

We have two sides of the theorem, so we'll do one at a time. We'll start with the case $r > 1 - H(p)$; then we want to show there always exists some message m whose failure probability is very close to 1. Actually what we'll do is go by the contrapositive — we'll say that *if* the failure probabilities of all messages are bounded away from 1, then we must actually have $r \leq 1 - H(p)$. So we assume that

$$\max_{m\in\{0,1\}^k} (\text{failure probability of } m) \leq 1 - \varepsilon$$

(for some constant ε), and we want to conclude that $r \leq 1 - H(p)$.

The key proposition is the following. Basically what we want to claim is, let's fix some particular message m. What I want to claim is that if the failure probability is bounded away from 1, then actually there must be *many* codewords that decode to this message.

Proposition 5.10
If a message m has failure probability at most $1 - \varepsilon$, then

$$\#\{x \in \{0,1\}^n \mid \mathsf{Dec}(x) = m\} \geq 2^{(H(p)-o(1))n}.$$

So an exponential number of codewords must decode to m. To see why this is relevant for us, let's first prove this case of the theorem assuming this proposition, and we'll then come back to proving this proposition later.

Basically, the point is going to be that the set of all codewords that decode to m — if we look at the collection of preimages, they partition the set of codewords (every codeword is mapped to only one message). So if all these preimages are large, there cannot be too many of them. (This is basically a volume argument — you can think of this proposition as a lower bound on the 'volume' of codewords mapping to a given message.)

Proof of Theorem assuming Proposition 5.10. We'll go by contrapositive. There are 2^n codewords, so

$$2^n = \sum_{m \in \{0,1\}^k} \#\{x \in \{0,1\}^n \mid \mathsf{Dec}(x) = m\} \geq 2^k \cdot 2^{(H(p)-o(1))n}.$$

And if we rearrange, this implies that $r = k/n \leq 1 - H(p)$. □

This is a classic type of argument called a *volume argument*, and the key that makes this argument work is this lower bound on the number of codewords that decode to a given message.

Student Question. *With the first equality, are we saying the encoding is surjective?*

Answer. No. This isn't about the encoder at all, just the decoder — it's just saying that the decoder is a function, so each codeword can decode to only one message.

Now let's prove the proposition. This is where we're going to use the weak law of large numbers, and also a very basic bound on the size of Hamming balls in the space of all codewords.

Proof of Proposition 5.10. Our assumption tells us that

$$\mathbb{P}[\mathsf{BSC}_p(\mathsf{Enc}(m)) \in \mathsf{Dec}^{-1}(m)] \geq \varepsilon$$

(i.e., that if I apply my noisy channel to the encoding of m, the probability this lands into the pre-image of the decoding function is at least ε). And our goal is to deduce that the size of the pre-image is large, i.e.,

$$\left|\mathsf{Dec}^{-1}(m)\right| \geq 2^{(H(p)-o(1))n}.$$

Here's a really generic recipe for how you can lower-bound the size of a set: The rough strategy is we're going to construct a large set $A \in \{0,1\}^n$, and we're going to lower-bound the intersection of the pre-image with this set — we'll construct A such that

$$\mathbb{P}_{x \sim \mathsf{Unif}(A)}[x \in \mathsf{Dec}^{-1}(m)]$$

is lower-bounded. If we can do this, then we have

$$\left|\mathsf{Dec}^{-1}(m)\right| \geq \left|A \cap \mathsf{Dec}^{-1}(m)\right| = \mathbb{P}_{x \in \mathsf{Unif}(A)}[x \in \mathsf{Dec}^{-1}(m)] \cdot |A|.$$

So if we can lower-bound each of these two terms, we'll win.

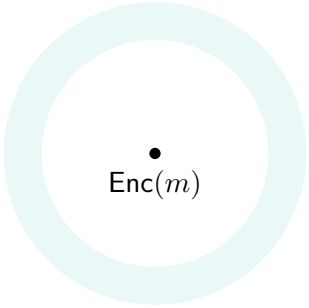
For the first term, you can sort of pattern-match that we want to take advantage of the assumption that under some process (the binary symmetric noise channel), we have a lower bound on the probability that the resulting point lies in the pre-image. This is not exactly what we're going to do, but it's morally what we want to try. The main challenge is handling the fact that the distribution over codewords is a non-uniform one — if p is very small, this distribution is going to be more concentrated on points close to the encoding of m than points further away. So we need to be able to handle this somehow.

So here our assumption is very roughly speaking that for a typical codeword drawn from the distribution $\mathsf{BSC}_p(\mathsf{Enc}(m))$, it's going to land in the pre-image of m (with decent probability). So we're going to take our set A to be the set of 'typical' points that you get out of this noise process.

So for $\delta > 0$, we define

$$A_\delta = \{x \in \{0,1\}^n \mid d_H(x, \mathsf{Enc}(m)) \in [(p-\delta)n, (p+\delta)n]\}.$$

In expectation, when I apply the binary symmetric channel to the encoding of m, I'll get a point pn away from it. So I'm enlarging this slightly.



The idea behind looking at this annulus is that our original distribution is not uniform; we want to get something that does look somewhat uniform. The idea is that all points in A_δ should be roughly the same mass under this distribution, and by concentration (the weak law of large numbers), the mass of A_δ should be large.

(We'll let δ depend on n — so it may be decaying with n, but sufficiently slowly that the weak law of large numbers kicks in.)

Call $\mathsf{Enc}(m) = y$. By the weak law of large numbers, we know that

$$\mathbb{P}[\mathsf{BSC}_p(y) \in A_\delta] \geq 1 - \frac{\varepsilon}{2}$$

(for large enough n, as long as δ doesn't shrink too quickly).

Now we're going to intersect this annulus A_δ with our pre-image — we start with the assumption

$$\varepsilon \leq \mathbb{P}[\mathsf{BSC}_p(y) \in \mathsf{Dec}^{-1}(m)].$$

Now we'll break this up by conditioning on the event that our codeword $\mathsf{BSC}_p(y)$ lies in the annulus A_δ — this occurs with very high probability, so we don't lose much. Specifically, we get that this is equal to

$$\mathbb{P}[\mathsf{BSC}_p(y) \in \mathsf{Dec}^{-1}(m) \cap A_\delta] + \mathbb{P}[\mathsf{BSC}_p(y) \in \mathsf{Dec}^{-1}(m) \mid \mathsf{BSC}_p(y) \notin A_\delta]\mathbb{P}[\mathsf{BSC}_p(y) \notin A_\delta].$$

For the second term, the conditional probability is at most 1, and $\mathbb{P}[\mathsf{BSC}_p(y) \notin A_\delta]$ is small (at most $\varepsilon/2$) by what we said earlier. So rearranging this, we get that

$$\mathbb{P}[\mathsf{BSC}_p(y) \in \mathsf{Dec}^{-1}(m) \cap A_\delta] \geq \frac{\varepsilon}{2}.$$

And now we're almost done — because on this annulus A_δ, the law of $\mathsf{BSC}_p(y)$ is roughly uniform — it looks like roughly $p^{pn}(1-p)^{(1-p)n} = 2^{-H(p)n}$. To be a bit more formal about this, the left-hand side is equal to

$$\sum_{x \in \mathsf{Dec}^{-1}(m) \cap A_\delta} \mathbb{P}[\mathsf{BSC}_p(y) = x] \leq p^{(p-\delta)n}(1-p)^{(1-p+\delta)n} \left|\mathsf{Dec}^{-1}(m)\right|.$$

And now we can rearrange — we're saying this thing is lower-bounded by some constant, so this implies

$$\left|\mathsf{Dec}^{-1}(m)\right| \geq \frac{\varepsilon}{2} \cdot 2^{(H(p)-O(\delta))n}.$$

And $\delta \to 0$ as $n \to \infty$ (i.e., $\delta = o(1)$), and ε is a constant, so we can absorb it into the $o(1)$; so then this is $2^{(H(p)-o(1))n}$. □

Student Question. *How do we go from the assumption to* $\mathbb{P}[\mathsf{BSC}_p(y) \in A_\delta] \geq 1 - \varepsilon/2$*?*

Answer. We don't use the assumption here, just the weak law of large numbers. If we consider the random variable X_i which is 1 if an error occurs in the ith coordinate, then the Hamming distance is just going to be $\sum_i X_i$; and the weak law of large numbers says that it's going to concentrate around its mean, which is pn, with very high probability. (The specific quantity $\varepsilon/2$ is not important — you can make it as small as you want — it's just useful for the analysis.)

So from an extremely weak concentration statement, you can already do something very interesting about the limits of error-correcting codes.

Student Question. *Why is it weak?*

Answer. Next lecture we'll see much stronger concentration inequalities that say your deviation will typically be around $\sqrt{n}$, and past that you'll get *exponential* decay — so we get really strong guarantees on how fast it goes to 1.

It's also weak because it doesn't use full independence — all you needed to use was Chebyshev, which just needed a bound on variance of the empirical mean. That's much weaker than full joint variance.

§5.5 The low-rate case

Now we'll move to the other side of the implication, which says that if I set r to be smaller than $1 - H(p)$, then there *exists* a good error-correcting codes. Constructing good error-correcting codes is in general not an easy problem; but here we just want existence, so we'll use our best friend, the probabilistic method — we'll pick a *random* error-correcting code, and show that it's good with very high probability. (When you're faced with something very hard, it's good to do something random first and analyze what happens.)

So we pick a random function $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n$. This means for every choice of message m, we pick an independent uniformly random n-bit string — so $\mathsf{Enc}(m) \sim \mathsf{Unif}\{0,1\}^n$ (chosen independently for all messages m).

And for the decoder, we'll use something called a *maximum likelihood decoding* — we'll define

$$\mathsf{Dec}(x) = \arg\min_m d_H(x, \mathsf{Enc}(m))$$

(the message which has the closest encoding to x). So the picture is that we have our small space $\{0,1\}^k$; we're mapping (via Enc) to a much larger space, so now we have just a few points scattered around here. Now in comes some candidate codeword x. It might not immediately be the image of a message under the encoding scheme; but I'm just going to find the closest encoding of a message to me, and that's going to be what I decode to.

Remark 5.11. There are a whole bunch of issues with whether you can efficiently compute this, but we won't worry about that (e.g., you can brute-force compute this decoder in exponential time).

Student Question. *Isn't the encoder random — how does that work with the definition of the decoder?*

Answer. The encoder is random; but once I fix the choice of encoder, I have a fixed choice of decoder.

And we're going to show that this totally simple error-correcting code is actually extremely good. We're actually going to show it's good with very good probability.

So our goal is to show that with positive probability over the random choice of encoder,

$$\max_{m \in \{0,1\}^k} \mathbb{P}[\mathsf{BSC}_p(\mathsf{Enc}(m)) \notin \mathsf{Dec}^{-1}(m)] \leq o(1).$$

(So with positive probability our random construction gives us a choice of encoder where the success probability for *every* single message is lower-bounded by $1 - o(1)$.)

This is a very hard statement, so let's start by proving something a little easier — a little more 'average-case.' As a stepping stone, we'll prove the following key proposition.

Proposition 5.12

Suppose $r < 1 - H(p)$. Then for every message m,

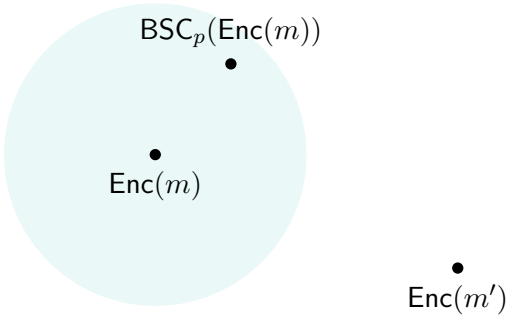
$$\mathbb{E}_{\mathsf{Enc}}[\mathbb{P}_{\mathsf{BSC}_p}[\mathsf{Dec}(\mathsf{BSC}_p(\mathsf{Enc}(m))) \neq m]] \leq o(1).$$

This is a weaker statement — I'm guaranteeing something in expectation, and the $\forall$ is on the outside. (This is a large sequence of symbols, but it's saying that if I look at the *expected* failure probability for this specific message, this is decaying in n. There's two choices of randomness here — in the random encoding function, and in the errors being introduced to a codeword.)

In a moment we'll see how to use this weaker average-case statement to get what we want (there's a very clever trick that goes into that), but sort of the key is establishing this average-case statement.

Proof of Proposition 5.12. What's the intuition behind this? Imagine we have our point $\mathsf{Enc}(m)$, and we draw around it a ball of radius $(p + \delta)n$. We know that after I pass this codeword through the binary symmetric channel, I'm going to get a point that's still within this ball with high probability.

The hope is going to be that none of the other messages have encodings close to this point — because every one of these messages has an encoding drawn independently and uniformly from $\{0, 1\}^n$. So the hope is that if I look at any other m', its encoding $\mathsf{Enc}(m')$ is going to be somehwere far away from this perturbation of $\mathsf{Enc}(m)$.



In particular, what we want is that $\mathsf{Enc}(m)$ is going to be the closest encoding of a message to our noised-up codeword — that $\mathsf{BSC}_p(\mathsf{Enc}(m))$ is closer to $\mathsf{Enc}(m)$ than to any other $\mathsf{Enc}(m')$.

So let's look at our failure probability and break it into two events. For convenience, let's define $y = \mathsf{Enc}(m)$ again. The only way to fail is if the above statement fails; so we can write

$$\mathbb{P}[\mathsf{Dec}(\mathsf{BSC}_p(y)) \neq m] \leq \mathbb{P}[d_H(\mathsf{BSC}_p(y), y) > (p + \delta)n] + \sum_{m' \neq m} \mathbb{P}[d_H(\mathsf{BSC}_p(y), \mathsf{Enc}(m')) \leq (p + \delta)n].$$

This is just a union bound — if I fail, either the above picture is violated (my perturbation was farther than $(p + \delta)n$), or there's some other message m' whose encoding is closer to my noised-up codeword $\mathsf{BSC}_p(y)$ than $(p + \delta)n$ (and we're union-bounding over m').

Now we're going to bound each one of these separately. Basically the point now is for the first term, we can again use the weak law of large numbers to say that the probability I land outside this ball is very small (so

for this, we'll again just use weak concentration). And for the second part, we'll use the fact that we just drew random codewords — we'll take advantage of the randomness of the encoding.

By the weak law of large numbers, the first term is $o(1)$.

To deal with the second term, let's fix $m' \neq m$. (So far we haven't even taken advantage of the fact that we're taking an expectation over the randomness of the encoder; we'll do that here.) We want to say that

$$\mathbb{E}_{\mathsf{Enc}}[\mathbb{P}[d_H(\mathsf{BSC}(y), \mathsf{Enc}(m')) \leq (p+\delta)n]] \leq o(2^{-k})$$

(because then when we sum over all 2^k messages m', we'll get an $o(1)$ upper bound).

Now we'l take advantage of the fact that we have a uniform random encoding — so $\mathsf{Enc}(m')$ is a uniform random point in $\{0,1\}^n$, independent of whatever $\mathsf{BSC}_p(y)$ is. To highlight this, we can rewrite the above thing as

$$\mathbb{E}_{\mathsf{BSC}_p}[\mathbb{P}_{\mathsf{Enc}}[d_H(\mathsf{BSC}_p(y), \mathsf{Enc}(m')) \leq (p+\delta)n]]$$

(we're switching the roles of the randomness — this is because we can imagine writing down an indicator for this event, and then we're taking an expectation over both sources of randomness; and we can switch them by independence). And now this is much easier to think about — we have some fixed point $\mathsf{BSC}_p(y)$, and we're essentially bounding the volume of a ball around it (under the uniform distribution), since $\mathsf{Enc}(m')$ is just a uniform random point. So this is

$$\mathbb{P}_{z \sim \mathsf{Unif}\{0,1\}^n}[z \text{ lies in a ball of radins } (p+\delta)n \text{ around } \mathsf{BSC}_p(y)]$$

(where $\mathsf{BSC}_p(y)$ is some arbitrary point; this probability doesn't actually depend on what thsi point is).

Now we're almost done, and we just need to plug in a couple of calculations.

Claim 5.13 — For every $x \in \{0,1\}^n$, we have

$$\#\{y \mid d_H(x,y) \leq pn\} \leq 2^{H(p)n}.$$

This is actually something we've already seen before, in the first lecture.

Proof sketch. The left-hand side is $\sum_{k=0}^{pn} \binom{n}{k}$. And we saw in the first lecture that if you use Stirling's approximation, you can approximate each of these binomials by something that looks like an exponential of an entropy of something (times n). □

In particular, now we see that the probability we're interested in is upper-bounded by

$$2^{(H(p)+\delta-1)n}$$

(the -1 is because we're drawing a uniformly random point, so we divide by 2^n; and the $+\delta$ is because we're taking a ball of radius $p+\delta$, where we think of δ as decaying slowly with n).

Now going back to our failure probability split into two terms, the first term is $o(1)$ by the weak law of large numbers, so we get

$$o(1) + 2^k \cdot 2^{(H(p)+\delta-1)n}.$$

And this second term can be rearranged to

$$2^{(r+H(p)+\delta-1)n}.$$

If $r < 1 - H(p)$, this is also going to decay to 0 as $n \to \infty$ (in fact, it'll decay to 0 exponentially fast). □

Now we have an average-case statement — for every m, in expectation over the random encoder, the failure probability is $o(1)$. Now we want to boost this to a much stronger statement, that every message *simultaneously* has failure probability $o(1)$. This requires just one final trick.

Proposition 5.12 implies that if I take a random encoding and a random message, the probability of failure is low — we have

$$\mathbb{E}_{\mathsf{Enc}}[\mathbb{E}_{m\in\{0,1\}^k}[\mathbb{P}[\text{failure for } m]]] \leq o(1).$$

In particular, there exists some choice of encoding such that for a random message, the failure probability is small — i.e., there exists Enc with the average-case guarantee that

$$\mathbb{E}_{m\in\{0,1\}^k}[\mathbb{P}[\text{failure for } m]] \leq o(1).$$

This is not quite what we wanted. But the key is that now this means at least *half* of the points m must be good — there exists an encoding and a subset of messages $S \subseteq \{0,1\}^k$ with $|S| \geq 2^{k-1}$ such that

$$\max_{m\in S} \mathbb{P}[\text{failure for } m] \leq o(1)$$

(to be more precise, if our original probability bound was $\eta(n)$ (which decays to 0), then here we get an upper bound of $2\eta(n)$ — because if all messages in S had failure probability exceeding $2\eta(n)$, then that would exceed our bound with an expectation over m).

And essentially that's it — we've constructed a good encoder for this subset S of messages. And that basically means we've constructed a good $[n, k-1]$-encoder — we can encode bitstrings of length $k-1$ instead of k — which has essentially the same rate.

§6 February 19, 2025 — Chernoff bounds

In the past few lectures, we saw lots of applications of fairly basic techniques like first and second moments. These required very little assumptions about the stochastic process we're looking at — linearity of expectation doesn't require *anything*, for instance — so these are widely applicable, and we saw applications to statistical inference and the limits of error-correcting codes and analyzing random networks (e.g., connectivity) and solving NP-hard problems (like SAT).

Now we'll turn to settings where these techniques won't be enough; and it'll motivate us to develop stronger concentration inequalities.

§6.1 A motivataing example — MaxCut

As a motivating example, consider a problem we saw on the first pset, MaxCut. Here we're given a graph G as input, and our goal is to partition the vertices into two nonempty sets to maximize the number of edges *cut* (i.e,. going between the two parts).

We devised an approximation algorithm for this. But now we want to look at the average case — if I sample a *random* graph from the Erdős–Rényi distribution, can we understand very tightly what its MaxCut value is?

> **Question 6.1.** Can we understand the value of MaxCut in the average case, i.e., for $\mathcal{G}(n, \frac{1}{2})$?

(You can do this analysis for any edge probability you want.)

So we want to understand the value of the max-cut, so let's look at our random variables — we can have a random variable for the number of edges cut when I look at a specific partition. So for a subset $S \subsetneq V$ of vertices, I can look at the random variable counting the number of edges cut by this partition, i.e., $\left|E(S, \overline{S})\right|$.

And we can do all sorts of things like computing its expectation and variance and so on. For instance, we know its expectation is the number of possible pairs of vertices — i.e., $|S| \cdot |\overline{S}|$ — times the edge probability, which is

$$\mathbb{E}[E(S, \overline{S})] = \frac{|S| \, |\overline{S}|}{2}.$$

And you can do a similar calculation for the variance. So then if you apply Chebyshev, it implies that for each individual partition S,

$$\mathbb{P}\left[\left|E(S, \overline{S})\right| \text{ close to expectatio}\right] \geq 99.9\%$$

(for instance — or you can get even closer to 1 if you like). This is the type of guarantee you'd get from a first and second moment type analysis.

But now, here we're looking at the *maximum* cut — we're looking at the maximum of these random variables over all exponentially many choices of the cut S. These random variables are highly correlated (if S and T overlap, the corresponding cut values will be highly correlated). And if you don't have any kind of independence, the only tool at your disposal is a union bound. But if you try applying a union bound with this kind of guarantee, you have *exponentially* many random variables; and for such a bound to not be useless, you need this probability guarantee to be *inverse exponentially* close to 1 — 99.9% is not going to be good enough for us.

So to control something like MaxCut, we need the probability guarantee to be more like $1 - \exp(-\Theta(n))$, because there are exponentially many cuts that we're union-bounding over. And this is simply not possible with something like Chebyshev.

So this is one motivation for why we're going to develop some stronger concentration inequalities. (This motivation is much broader — if you want to look at the maximum of a large collection of random variables, you're going to need these kinds of concentration bounds.)

§6.2 A heuristic — sums of i.i.d. random variables

Let's say we have a bunch of *independent* random variables $X_1, \ldots, X_n$, and let's focus on their average

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

For now, let's assume they're i.i.d. with mean μ and variance $\sigma^2 < \infty$. So in particular, the variance of this average $\overline{X}$ is going to be $\frac{\sigma^2}{n}$.

In the previous lecture, we used the weak law of large numbers, which is a weak kind of concentration inequality:

Theorem 6.2 (Weak law of large numbers)
For every $\varepsilon > 0$, we have $\lim_{n \to \infty} \mathbb{P}[\left|\overline{X} - \mu\right| < \varepsilon] = 1$.

If you've taken a probability course before, you may have heard of e.g. the central limit theorem, which says somethin g much stronger — not only does this thing go to 0, but we can actually understand the *distribution* of this thing. It'll be approximately Gaussian, and that also suggests a more quantitative bound on this probability. Specifically, we know that if we look at the cumulative distribution function of this empirical mean, it's going to look like a Gaussian.

Theorem 6.3 (Central limit theorem)

For all $t \in \mathbb{R}$, we have

$$\mathbb{P}\left[\frac{\sqrt{n}}{\sigma}(\overline{X} - \mu) \le t\right] \to \mathbb{P}_{g\sim\mathcal{N}(0,1)}[g \le t].$$

This is going to suggest the *type* of bound we're going to shoot for. It's not too difficult to prove that the CDF of a Gaussian satisfies

$$\mathbb{P}_{g\sim\mathcal{N}(0,1)}[g \le t] \approx \frac{1}{t} \cdot \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \quad \text{for all } t > 0.$$

So using this approximation, we can kind of get a sense of what kind of concentration inequality we can hope for. If we set $t \approx \sqrt{n}/\sigma \cdot \mu$ (to cancel with the factor of $\sqrt{n}/\sigma$ in our statement of the CLT), then we'd *expect* an inequality of the form

$$\mathbb{P}\left[\left|\overline{X} - \mu\right| \ge \varepsilon\mu\right] \lesssim \exp\left(-\frac{n\mu^2}{\sigma^2}\right)$$

(this inequality is not legit at the moment, but the CLT suggests an inequality of this form, so it's something we're going to shoot for).

This is already quite useful — if we have a random variable whose variance isn't too much larger than the square of the variance, then this is a bound that decays exponentially in n as $n \to \infty$. And we can hope to apply this in a situation like the earlier one, where we have exponentially many random variables.

So our goal is to actually prove something like this, so that it's no longer just an 'inequality' in quotes.

Remark 6.4. The reason this inequality isn't legit is mainly that it's not taking into account the error coming from the convergence in CLT.

§6.3 The Chernoff–Hoeffding bound

Here's the main theorem we'll prove today. For this lecture we'll work with just bounded random variables (e.g., Bernoullis); next lecture we'll start to extend these inequalities to larger and larger classes of random variables, but this lecture we'll just focus on bounded ones.

Theorem 6.5 (Chernoff–Hoeffding)

Let $X_1, \ldots, X_n$ be independent random variables with $X_i \in [a_i, b_i]$ for all i. Then letting $\overline{X} = \frac{1}{n}\sum_{i=1}^n X_i$, we have

$$\mathbb{P}[\overline{X} - \mathbb{E}[\overline{X}] \ge t] \le \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Maybe one way to interpret this bound — this denominator might look slightly strange. But it's essentially just an upper bound on the variance of this empirical mean. Specifically, note that

$$\mathrm{Var}[\overline{X}] \le \frac{1}{n^2}\sum_{i=1}^n (b_i - a_i)^2,$$

because each individual random variable X_i has variance upper-bounded by $(b_i - a_i)^2$ (since it's in $[a_i, b_i]$), and the variance of a sum of independent random variables is the sum of their variances (and we scale down by n^2 because we're averaging). So we should interpret this kind of like a Gaussian tail, where we have t^2 divided by the variance of our random variable.

Student Question. *Is there a reason there's no absolute value?*

Answer. We can also get the lower inequality by negating everything; so we could also say

$$\mathbb{P}[\left|\overline{X} - \mathbb{E}[\overline{X}]\right| \geq t] \leq 2\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

§6.4 Back to MaxCut

Maybe before we prove this, let's come back to the motivating example of MaxCut. (In the next few lectures, we'll see more applications of these types of inequalities.)

Theorem 6.6

We have $\mathbb{P}[\mathsf{MaxCut}(\mathcal{G}(n, \frac{1}{2})) \geq (1 + \frac{1}{2\sqrt{n}})\frac{n^2}{8}] \leq 2^{-n}$.

(There's also a matching lower bound, though we won't go into that — so we can very tightly characterize what the MaxCut value is if I sample a random graph.)

Proof. Let's set up our random variables; so first, let's fix some $\emptyset \subsetneq S \subsetneq V$, and look at our random variables. As we said earlier, we have

$$\mathbb{E}[\left|E(S, \overline{S})\right|] = \frac{|S|\left|\overline{S}\right|}{2} \leq \frac{n^2}{8}$$

(the worst case for this quantity is the case where $|S|$ is exactly half the total number of vertices).

And we can decompose $\left|E(S, \overline{S})\right|$ as a sum of indicator random variables — as

$$\left|E(S, \overline{S})\right| = \sum_{u \in S, v \in \overline{S}} \mathbf{1}[uv \text{ is an edge}].$$

So we can directly apply Chernoff–Hoeffding to upper-bound the probability that this thing exceeds its expectation. In particular,

$$\mathbb{P}\left[\left|E(S, \overline{S})\right| \geq \left(1 + \frac{1}{2\sqrt{n}}\right)\frac{n^2}{8}\right] \leq \exp(-2n)$$

(if you calculate it out). And now we can union-bound over all 2^n possible cuts. □

Student Question. *Does Chernoff–Hoeffding only require pairwise independence?*

Answer. It requires full independence; we'll see this in the proof.

Student Question. *Are the indicator variables all independent?*

Answer. Yes — the edges are sampled independently. (We've fixed a cut first and we're bounding its tail, and then we union-bound over all cuts.)

§6.5 Moments and the moment generating function

Now we'll discuss how we prove this theorem. We're really going to take advantage of the joint independence of all these variables. For the weak law of large numbers, you didn't need full joint independence — you just

needed pairwise independence, since all we used was that the variance of the sum was the sum of variances. (That's one of the reasons Chebyshev is kind of limited in the kinds of tail bounds it gives.)

So to use the full joint independence more, we can look at *higher* moments. Recall that if X is a nonnegative random variable, then Markov says

$$\mathbb{P}[X \geq t] \leq \frac{\mathbb{E}[X]}{t}.$$

But if we have more information about something like the moments of X — i.e., the expectations of X to some power — then we can get a better dependence on t. Specifically, we have

$$\mathbb{P}[X \geq t] = \mathbb{P}[X^k \geq t^k] \leq \frac{\mathbb{E}[X^k]}{t^k},$$

where the dependence on t (the deviation from the expectation) now decays much faster than what you'd get from Chebyshev. (This is essentially just generalizing the proof of Chebyshev's inequality to leverage information about higher moments of the random variable.)

Now, if you have information about *all* these moments, you can kind of wrap them all up into one nice function called the moment generating function.

Definition 6.7. The *moment generating function* of a random variable X is the function

$$s \mapsto \mathbb{E}[\exp(sX)].$$

If we expand this as a Taylor series, it's the power series whose coefficients have all the moments of X — we have

$$\mathbb{E}[\exp(sX)] = \sum_{k=0}^{\infty} \frac{s^k \mathbb{E}[X^k]}{k!}$$

(assuming all these moments exist and the series converges and so on).

And we're going to use this to prove Chernoff–Hoeffding.

§6.6 Proof of Chernoff–Hoeffding

The key lemma is a bound on the moment generating function of a *bounded* random variable.

Lemma 6.8 (Hoeffding)
If X is a mean-0 random variable taking values in $[a, b]$, then for all $s > 0$, we have

$$\mathbb{E}[\exp(sX)] \leq \exp\left(\frac{s^2}{2} \cdot \left(\frac{b-a}{2}\right)^2\right).$$

We again see the length of this interval appearing; it's kind of a proxy for the variance of X (it might not be the true variance, but it's certainly an upper bound); and we have a quadratic function.

Before we prove this key lemma, we'll see how to deduce the Chernoff bound from this; and this is where we'll see the use of independence.

Proof of Chernoff–Hoeffding. We're basically going to use the idea of using Markov on moments, but we're going to replace X^k with an exponential. Specifically, we can write

$$\mathbb{P}[\overline{X} - \mathbb{E}[\overline{X}] \geq t] = \mathbb{P}[\exp(s(\overline{X} - \mathbb{E}[\overline{X}])) \geq \exp(st)]$$

(where we exponentiate both sides; s is going to be some parameter we'll optimize later, similarly to how with moments, you can imagine picking the best possible k that gives the best bound). Now we'll use Markov; we took an exponential, so we have a nonnegative random variable, and this is upper-bounded by

$$\frac{\mathbb{E}[\exp(s(\overline{X} - \mathbb{E}[\overline{X}]))]}{\exp(st)}.$$

And now in the numerator, we have a sum of random variables which are independent; the exponential of a sum is equal to the product of the exponentials of each terms, and by independence we can factor out the expectations. So by independence, this is equal to

$$\frac{\prod_{i=1}^n \mathbb{E}[\exp(\frac{s}{n}(X_i - \mathbb{E}[X_i]))]}{\exp(st)}.$$

And now each one of these random variables is bounded, so we can use the Hoeffding lemma — by Hoeffding, this is upper-bounded by

$$\exp\left(\frac{s^2}{2n^2}\sum_{i=1}^n \left(\frac{b_i - a_i}{2}\right)^2 - st\right).$$

(Shifting X_i by its mean doesn't affect the size of its interval.)

And now on the inside we have some ugly-looking expression, but it's really just some quadratic function $as^2 - bs$. So we can imagine graphing this function. And we want to choose the best possible s, the one that gives the best upper bound. So we want to minimize whatever's inside this exponential here — we basically want to find the minimum of this quadratic function, and that'll happen at $s = b/2a$. So if you just plug in the best choice of s here, you'll get exactly the conclusion. □

§6.7 Proof of Hoeffding's lemma

Now let's prove the Hoeffding lemma. If you Wikipedia how to prove Hoeffding's lemma, it ultimately boils down to some calculus, and that was the proof Kuikui was taught; but he'll show us a different, very slick proof that uses some convex analysis and really minimal calculation.

If we take the log of both sides, the inequality we want is equivalent to

$$\log \mathbb{E}[\exp(sX)] \le \frac{s^2}{2}\left(\frac{b-a}{2}\right)^2.$$

The function on the left is also a very fundamental function — it's something called the *cumulant generating function*, and it'll make an appearance later in the course as well (when we'll look at generalizations of this object). For now, we'll just give it a name; we'll call it $\psi_X(s)$. So we just want to bound this function by a quadratic function.

We're going to break this into two claims — one about this function ψ_X, and another about how you compare the two sides. The idea is a quadratic is a very nice convex function; we're going to show ψ_X is also a convex function, and it's sort of dominated by this one.

The first claim is just a generic lemma from convex analysis.

Claim 6.9 — Let $f, g: \mathbb{R} \to \mathbb{R}$ be smooth. Suppose that:

- $f - g$ is convex.
- There exists some x^* such that $f(x^*) \ge g(x^*)$ and $f'(x^*) = g'(x^*)$.

Then $f \ge g$ on all of $\mathbb{R}$.

This is a generic lemma in convex analysis; if we have time we'll prove it, and if we don't the proof is in the notes. But basically the point is, imagine we have one convex function f, and maybe we have another convex function g; and let's say there's some point where their derivatives coincide, and at which f is larger than g. Because f is 'more' convex than g, we'd expect f to grow faster as we move away from this point; and that's kind of the intuition behind this claim.

The second claim is going to be about this cumulant generating function and the quadratic on the right-hand side — it'll basically let us verify the two conditions of this claim.

Claim 6.10 — Fix some $s \in \mathbb{R}$, and define the new random variable Y_s whose distribution is given by

$$\mathbb{P}[Y_s = z] = \frac{\mathbb{P}[X = z] \cdot \exp(sz)}{\mathbb{E}[\exp(sX)]}.$$

Then we have $\psi_X'(s) = \mathbb{E}[Y_s]$ and $\psi_X''(s) = \mathrm{Var}[Y_s]$.

So we're defining a new random variable Y_s which is related to X by $\mathbb{P}[Y_s = z] \propto \mathbb{P}[X = z]\exp(sz)$ (the denominator is just the factor we need to normalize by). And the claim is that the moments of this random variable are related to the derivatives of the cumulant generating function.

Remark 6.11. This definition is just for discrete random variables; but we can do something similar for continuous ones (with the density function).

The key is that because we have $\mathbb{P}[X = z]$ on the right and $\mathrm{supp}(X) \subseteq [a, b]$, we also have $\mathrm{supp}(Y_s) \subseteq [a, b]$. So in particular we get

$$\psi_X''(s) = \mathrm{Var}[Y_s] \le \left(\frac{b-a}{2}\right)^2.$$

In a moment we'll see how you could have come up with such a random variable — when we differentiate ψ_X, we'll see quantities that can be interpreted like this.

So this is in some sense the key lemma.

Proof of Claim 6.10. Let's just sort of do the usual calculus — we're looking at ψ_X', so if we use the chain rule we get

$$\psi_X'(s) = \frac{\frac{d}{ds}\mathbb{E}[\exp(sX)]}{\mathbb{E}[\exp(sX)]}.$$

Now the key observation is that differentiation is a linear operator on functions, and expectation is also linear. So we can actually push the differentiation inside the expectation. (It's easiest to see this if we imagine X is discrete; then we're just averaging a finite number of functions, and we're just pushing the derivative into each term.) So this is the same as if we put the derivative inside the expectation, giving

$$\psi_X'(s) = \frac{\mathbb{E}[\frac{d}{ds}\exp(sX)]}{\mathbb{E}[\exp(sX)]} = \frac{\mathbb{E}[X \cdot \exp(sX)]}{\mathbb{E}[\exp(sX)]}.$$

And by the definition of our random variable Y, this is exactly $\mathbb{E}[Y_s]$.

Now if we differentiate again, it's really the same type of calculation — we get

$$\psi''(s) = \frac{\mathbb{E}[X^2 \cdot \exp(sX)]}{\mathbb{E}[\exp(sX)]} - \frac{\mathbb{E}[X \cdot \exp(sX)]^2}{\mathbb{E}[\exp(sX)]^2} = \mathrm{Var}[Y_s]$$

(the first time is $\mathbb{E}[Y_s^2]$, and the second is $\mathbb{E}[Y_s]^2$).

That's it for this claim. □

And now you can kind of see how to use this to prove what we want. We have these two functions, and we know the second derivative of the cumulant generating function is upper-bounded by the second derivative of our quadratic function; so we know their difference is a convex function. And if we evaluate these functions at 0, we get 0 for both of them. So both our criteria are satisfied, and we're done.

Proof of Hoeffding lemma. Let $f(s) = \frac{s^2}{2}(\frac{b-a}{2})^2$. Then we know that $\psi_X''(s) \le f''(s)$ by the boundedness of Y_s. And we also have that

$$\psi_X(0) = f(0) = \psi_X'(0) = f'(0) = 0.$$

So now we can just use Claim 6.9. □

§6.8 The Berry–Esseen theorem and sharpness of Chernoff bounds

So that's essentially the proof of Chernoff–Hoeffding. Now we'll briefly discuss the sharpness of this inequality. There's a very nice theorem called Berry–Esseen, which is a quantitative form of the central limit theorem. We don't have time to prove this, but we'll give the statement and talk about how it kind of shows Chernoff–Hoeffding is sharp.

Theorem 6.12 (Berry–Esseen)

Let $X_1, \dots, X_n$ be i.i.d. with mean μ nad variance $\sigma^2 < \infty$, and let

$$\gamma = \mathbb{E}\left[\frac{|X_i - \mu|^3}{\sigma^3}\right] < \infty.$$

Then we have

$$\sup_{t \in \mathbb{R}} \left| \mathbb{P}\left[\frac{\sqrt{n}}{\sigma}\left(\frac{1}{n}\sum_{i=1}^n X_i - \mu\right) \le t\right] - \mathbb{P}_{g \sim \mathcal{N}(0,1)}[g \le t] \right| \le O(\gamma/\sqrt{n}).$$

(So we're comparing the CDFs of our normalized mean and a standard normal.)

So you have $1/\sqrt{n}$ convergence, uniformly in all thresholds t, for the tail probabilities.

What does this mean for us? What this tells us is not just for t large that the tail probabilities are exponentially decaying, but it also actually tells us *lower* bounds on these tail probabilities. In particular, Berry–Esseen implies that

$$\mathbb{P}\left[|\overline{X} - \mu| \ge \frac{t\sigma}{\sqrt{n}}\right] \ge 2\mathbb{P}_{g \sim \mathcal{N}(0,1)}[g \ge t] - O(1/\sqrt{n}).$$

Now imagine we set t to be of constant order, so that $t\sigma/\sqrt{n}$ is basically of the order of the standard deviation of $\overline{X}$. Then we're getting some absolute constant lower bound on this deviation probability.

So really this is telling us that the picture for the distribution of $\overline{X}$ is going to have a bump in the middle and it's going to decay at the tails; and when you should start seeing exponential decay is once you've gone at least on the order of a standard deviation away from the mean. So we have an interval of length on the order of the standard deviation, and it's only *after* this that you get your exponential decay — everything inside this interval still constitutes some constant fraction of the probability mass of this random variable. (This is not true for general random variables, but at least for averages of sums of independent variables, you get something like this.)

§6.9 Proof of Claim 6.9

We still have ten minutes left, so we'll prove Claim 6.9 and then wrap up.

Proof. Really all that matters for us is $f - g$; so let's just replace f with their difference $f - g$ and g with the 0 function (we can do this without loss of generality). Then the assumptions of this lemma just say that f is convex, and there exists a special point x^* such that $f(x^*) \geq 0$ and $f'(x^*) = 0$.

And this essentially already tells you everything — if I have a convex function, and it has a point where the derivative is 0, then that's the minimmizer for this function. And I'm telling you that the minimizer is nonnegative, so it had better be nonnegative everywhere.

So we note that x^* is the minimizer of f; this means $f(x) \geq f(x^*) \geq 0$ for all $x \in \mathbb{R}$, and we're done. □

§7 February 24, 2025 — Sub-Gaussian random variables

Today we'll continue our discussion of Chernoff-type bounds; this lecture and the next, we'll derive more and more such bounds that apply to more general setups.

Last lecture, we saw that if we have a large collection of *bounded* random variables (e.g., Bernoullis), then if I take a sum of a bunch of independent copies of these random variabes, I get a random variable that's very well-concentrated.

This lecture, we'll look at a more general class of random variables which satisfy the types of concentration inequalities we saw last lecture; these are called sub-Gaussian.

§7.1 Sub-Gaussian random variables

The name already suggests the definition. There's at least three equivalent ways of defining what a sub-Gaussian random variable is (with more provided in the notes).

Proposition 7.1

For a mean-0 random variable X, the following are equivalent:

(i) There exists a constant $k > 0$ such that

$$\mathbb{P}[|X| \geq t] \leq 2\exp(-t^2/k^2) \quad \text{for all } t \geq 0.$$

(ii) There exists a constant $k > 0$ such that

$$\mathbb{P}[|X|^p]^{1/p} \leq k_2\sqrt{p} \quad \text{for all } p \in \mathbb{N}.$$

(iii) There exists a constant $k > 0$ such that

$$\mathbb{E}[\exp(sX)] \leq \exp(k^2 s^2) \quad \text{for all } s \in \mathbb{R}.$$

We say X is *sub-Gaussian* if it satisfies any one of these.

These properties are all ways in which a random variable can resemble a Gaussian. The first way in (i) is if its *tails* are sub-Gaussian. (For example, if I have a bounded random variable, I can take k to be on the order of the length of the interval on which it's bounded.) This is perhaps the most natural way — it immediately says the random variable has strong concentration.

The second way, in (ii), is through *moments* — if its moments resemble those of a Gaussian, i.e., for all p, its pth moments are bounded by those of a Gaussian.

And the third way is to look at the moment generating function, which sort of encapsulates all the moments in a single function. For example, we saw that if we have a bounded random variable, then by Hoeffding's inequality (iii) is satisfied, where k_3 corresponds to the length of that interval.

We're not going to go through the proof of why these are equivalent. But you can see that (iii) implies (i) using Markov's inequality, and also implies (ii) if we look at Taylor series of the exponential (then all the coefficients correspond to moments of X).

We'll also have a definition that quantifies *how* sub-Gaussian a random variable is.

Definition 7.2. We define the *variance proxy* of X as the smallest k such that (iii) holds, i.e.,

$$\|X\|_{vp} = \inf\{k > 0 \mid \mathbb{E}[\exp(sX)] \le \exp(k^2 s^2) \text{ for all } s \in \mathbb{R}\}.$$

Example 7.3

By Hoeffding's lemma, if X is bounded in the interval $[-a, a]$, then $\|X\|_{vp} \lesssim a$.

The smaller this variance proxy is, the better concentrated my random variable is.

The reason for the name is the following (which is a nice exercise):

Fact 7.4 — We have $\|X\|_{vp}^2 \ge \mathrm{Var}(X)$.

For instance, if we have a random variable bounded in $[-a, a]$, its variance is always at most $\Theta(a^2)$.

§7.2 Concentratio bounds for sub-Gaussians

Now we'll try to generalize some of what we saw in the previous lecture to sub-Gaussian random variables. First, we'll look at a generalization of Chernoff–Hoeffding.

Theorem 7.5

Let $X_1, \dots, X_n$ be independent sub-Gaussian random variables, and fix $v \in \mathbb{R}^n$. Then

$$\left\| \sum_{i=1}^n v_i X_i \right\|_{vp}^2 \le \sum_{i=1}^n v_i^2 \|X_i\|_{vp}^2 .$$

This tells us that a linear combination of independent sub-Gaussians is also sub-Gaussian, and the variance proxy is in some sense a *norm* on the space of all sub-Gaussian random variables.

In particular, this gives us the following tail bound.

Corollary 7.6

Letting $Y = \sum_{i=1}^n v_i X_i$, we have

$$\mathbb{P}[|Y - \mathbb{E}[Y]| \ge t] \le 2\exp\left(-\frac{t^2}{\sum_{i=1}^n v_i^2 \|X_i\|_{vp}^2}\right) \quad \text{for all } t \ge 0.$$

The proof is essentially the same as what we saw last lecture — we use independence to say that the moment generating function for Y factors as the product of the moment generating functions of the individual random variables $X_1, \dots, X_n$, and then apply the usual Markov's inequality and so on.

So that deals with *linear* combinations; now let's look at other functionals.

Lemma 7.7

Let $X_1, \dots, X_n$ be independent random variables with mean 0 and $\|X_i\|_{vp}^2 \le \widehat{\sigma}^2/n$, and let $X = (X_1, \dots, X_n)$. Then

$$\mathbb{P}[\|X\| \ge \alpha\widehat{\sigma}] \le \exp(-C\alpha^2 n) \quad \text{for all } \alpha \ge \alpha^*$$

(where C and α^* are absolute constants).

We'll discuss the concentration of the Euclidean norm some more in the next lecture — the appearance of α^* will actually be that the Euclidean norm concentrates very sharply around some positive constant (so you can't go too far below that constant).

The proof is again analogous to the above one.

As one last one, let's look at the *maximum* of a bunch of sub-Gaussian random variables.

Theorem 7.8

Let $X_1, \dots, X_n$ satisfy $\|X_i\|_{vp}^2 \le 1$ for all n. Then

$$\mathbb{P}\left[\max_i X_i \ge t\right] \le n \cdot \exp\left(-\frac{t^2}{2\widehat{\sigma}^2}\right) \quad \text{and} \quad \mathbb{E}[\max_i X_i] \lesssim \widehat{\sigma}^2\sqrt{\log n}.$$

Note that here, we're *not* necessarily assuming that $X_1, \dots, X_n$ are independent.

The proof of this — since we're not assuming independence, it'll have to rely on something simple, like the union bound. And that's basically the proof — in order for the maximum to exceed some threshold t, there must exist some X_i exceeding this threshold t, and I can bound this using a union bound (that explains the factor of n). And then for the expectation, I can just integrate this tail bound.

§7.3 Random matrices

This lecture, we'll give a brief introduction on the theory of random matrices — this is an extremely rich and interesting subject, and we'll only scratch the surface. These objects aren't just because matrices are interesting so let's add randomness; they actually have some very surprising applications. The original reaso these were studied was to applications in mathematical physics. Eugene Wigner, sometime around the 1950s, realized that if you wanted to understand the energy levels of the nucleus of an atom, you can actually make very good predictions based on taking a random matrix and looking at its eigenvalues, which is a very surprising connection; so that was one of the main motivations for the study of random matrices. This lecture, we'll see some initial bounds on the *operator norm* of a random matrix.

Throughout, we'll let $G \in \mathbb{R}^{n\times n}$ be a random matrix with independent $\mathcal{N}(0,1)$ entries. To symmetrize or normalize things, we'll let

$$J = \frac{G + G^\intercal}{\sqrt{2n}}$$

(we'll see in a moment why we want this normalization).

Definition 7.9. This distribution is called the *Gaussian orthogonal ensemble*, denoted $\mathsf{GOE}(n)$.

Definition 7.10. We define H as the quadratic form associated to J — i.e., $H(x) = x^\intercal Jx$.

For example, $\max_{\|x\|=1} H(x)$ gives back the largest eigenvalue, or essentially the spectral norm, of H.

Before we look at the norm of this matrix, we'll look at the maximum of this quadratic form over just the Boolean cube.

Proposition 7.11

For $n \in \mathbb{N}$, we have

$$\mathbb{E}_{J\sim \mathsf{GOE}(n)} \left[\max_{\varphi \in \{\pm 1\}^n} H(\varphi) \right] \lesssim n\sqrt{\log 2}.$$

(We can also get a corresponding tail bound.)

This object $\max_\varphi H(\varphi)$ is also a very important object in mathematical physics — these are called the *ground states* for the *Sherrington–Kirkpatrick model*. The very high-level story behind this is roughly that statistical physicists were trying to model the behavior of glasses using physical ideas, such as the Ising model. And they came up with a model that was a little too hard to analyze, so they simplified it to essentially look at this particular function H — this turns out to be the energy function or Hamiltonian that governs this model. And it's an important area of study to understand the structure of the optimizers, and so on. This is actually the subject of two very recent results — in 2021, Jordan Piresi won the Nobel Prize in Physics, and one of the reasons why was a (non-rigorous) heuristic analysis of this model. And a couple of years later, Michel Talagrand won the Abel Prize for a rigorous analysis of the model.

So there's been a lot of interest in this type of thing recently. Here we'll give a first bound on the expected maximum for this energy function.

Proof. We're bounding the maximum of a bunch of random variables, and we have a Gaussian matrix, so we want to invoke Theorem 7.8 for an appropriate choice of random variables. So for each $\varphi \in \{\pm 1\}^n$, let $X_\varphi = \varphi^\intercal J\varphi$. These are a bunch of random variables; they're not independent by any means (they're correlated through the choice of J), but fortunately Theorem 7.8 doesn't make any assumptions about independence.

We know each X_φ is distributed as a mean-zero Gaussian random variable, because the entries of G are independent Gaussians; so we just need to bound the variance proxies for each one of these. We have

$$\|X_\varphi\|_{vp}^2 \lesssim \mathrm{Var}[X_\varphi] = \frac{1}{2n}\mathrm{Var}(\varphi^\intercal G\varphi) = \frac{1}{2n}\mathbb{E}[(\varphi^\intercal G\varphi)^2].$$

And now we can compute this thing explicitly, using the independence of the entries of G — this is going to be

$$\frac{1}{2n}\mathbb{E}\left(\sum_{ij}\varphi_i\varphi_j G_{ij}\right)^2 = \frac{1}{2n}\sum_{ijkl}\varphi_i\varphi_j\varphi_k\varphi_\ell \mathbb{E}[G_{ij}G_{k\ell}].$$

And the G_{ij} and $G_{k\ell}$ are independent Gaussians with mean 0 unless $(i,j) = (k,\ell)$. So up to constants, this is just going to be equal to

$$\frac{1}{2n}\sum_{ij}\varphi_i^2\varphi_j^2.$$

And now because we're assuming φ is all ± 1's, these are all 1's; so this is on the order of n.

And now we're basically done — $\widehat{\sigma}$ is going to be on the order of $\sqrt{n}$, and we have 2^n many such random variables; so we get $\sqrt{n}\cdot\sqrt{\log 2^n} \lesssim n$. □

§7.4 Operator norm of random matrices

So this is just a bound for the quadratic form of our random Gaussian matrix when φ comes from $\{\pm 1\}^n$. Now we'll use similar ideas to do something much better — to actually bound the operator norm of this matrix, which is essentially the supremum of H over all unit vectors v. This is an uncountably large set, so a naive union bound isn't going to work; but we'll see how to deal with that.

For notational convenience, we'll drop the symmetry condition now (it's not really going to affect anything).

Definition 7.12. The *operator norm* of J is defined as $\sup_{x\in S^{n-1}} \|Jx\|_2$.

Theorem 7.13

Let J be a random matrix with independent mean-0 entries, such that $\|J_i\|_{vp} \le \widehat{\sigma}^2/n$ for all i and j. Then there exists $\gamma^* > 0$ such that for all $\gamma > \gamma^*$, we have

$$\mathbb{P}\left[\|J\|_{\text{op}} \ge \gamma\widehat{\sigma}\right] \le 2\exp(-C\gamma^2 n).$$

So here we're working in a more general setting where the entries of J don't have to be $\mathcal{N}(0,1)$; they just have to be sub-Gaussian. So the probability our operator norm substantially exceeds the variance proxy is exponentially small in n.

You can already kind of see why it's of this order — the vectors φ on the Boolean cube have norm $\sqrt{n}$, so if I square them down by $\sqrt{n}$, I'll remove the factor of n in the previous result, and should get a constant bound. So this is consistent with that statement.

The challenge is that the operator norm is the supremum of H-values, but over the *entire* unit sphere; this is uncountable, so we can't just do a union bound, because we have infinitely many test values.

But we do want to union bound. So to do that, we're first going to *discretize* the sphere $\mathbb{S}^{n-1}$ — we're going to approximate the sphere with a discrete set of points $\mathcal{N}$, and then union-bound over those. We're going to want $\mathcal{N}$ to satisfy two competing properties. On one hand, we want $\mathcal{N}$ to be kind of dense in the sphere — we want it to be a very good approximation. But at the same time, we also want it to not contain too many points, because we'll eventually want to union-bound over it. So we need to resolve the tension between these two criteria.

§7.4.1 Discretizing the sphere

Here's a definition for how we'll construct our approximating set.

Definition 7.14. Fix $\delta > 0$. We say a set $\mathcal{N} \subseteq \mathbb{S}^{n-1}$ is a *δ-net* if for all $y \in \mathbb{S}^{n-1}$, there exists $x \in \mathcal{N}$ such that $\|x - y\|_2 \le \delta$.

In other words, if I imagine drawing my sphere, I'm pickign my discrete set of points so that I can cover my sphere with balls of radius δ centered at these points. So I'm covering the sphere with a net, and sort of trapping all the points in some sense — every point of the sphere has some point in $\mathcal{N}$ that's close-by.

Student Question. *What does the subscript* 2 *mean?*

Answer. Euclidean norm.

We need two claims. The first is that this is a sufficiently good notion of approximation that it'll let us bound the operator norm; and the second is that we can construct a δ-net which does not have too many points.

Claim 7.15 — We have $\max_{\varphi\in\mathcal{N}} \|J\varphi\|_2 \le \|J\|_{\text{op}} \le \frac{1}{1-\delta}\max_{\varphi\in\mathcal{N}}\|J\varphi\|_2$

This says that maximizing over just our δ-net is a very good approximation to maximizing over everything.

Claim 7.16 — There exists a δ-net of size at most $(1+2/\delta)^n$.

As we shrink δ, we'll have more points. But the point is that this has a good dependence on δ. Specifically, we want something that decays at most exponentially in n — because our tail bound decays exponentially in n. And this will be good enough (if we set δ properly).

Let's very quickly see how if we have these two claims, then we'll have our theorem. This is essentially the same proof of the earlier proposition (for $\{\pm 1\}^n$).

Proof of theorem. Let's fix a δ-net $\mathcal{N}$; then we want to bound $\mathbb{P}[\|J\|_{\text{op}} \ge \gamma\widehat{\sigma}]$. By the first claim, we can replace the operator norm by the maximum of this objective function only over this discrete set of points in the δ-net — so

$$\mathbb{P}[\|J\|_{\text{op}}] \le \mathbb{P}\left[\bigcup_{x\in\mathcal{N}}\{\|Jx\|_2 \ge (1-\delta)\gamma\widehat{\sigma}\}\right].$$

And now I only have a finite number of points, so I can do a union bound — this is upper-bounded by

$$\sum_{x\in\mathcal{N}}\mathbb{P}[\|Jx\|_2 \ge (1-\delta)\gamma\widehat{\sigma}].$$

(Think of δ as fixed, e.g., $\delta = 1/10$; it'll just be some constant.)

Now we just want to show that each of these probabilities is expnentially decaying in n, specifically decayin gfaster than th egrowth of the size of the δ-net.

The observation is that if we look at the entries of the vector Jx, they'lre also Gaussian. To make this more transparent, we'll write v instead of x in the previous bounds. Then each entry of Jv is a linear combination of independent sub-Gaussian random variables; so it's going to be sub-Gaussian with variance

$$\|(Jv)_i\|_{vp}^2 \le \widehat{\sigma}^2/n.$$

And now I'm taking a Euclidean norm, so I can use the earlier lemma (the tail bound on the Euclidean norm of a sub-Gaussian vector) to get

$$\mathbb{P}[\|Ju\|_2 \ge (1-\delta)\gamma\widehat{\sigma}] \le \exp(-C(1-\delta)^2\gamma^2 n).$$

(We're basically just plugging in the earlier lemma, assuming $(1-\delta)\gamma$ exceeds some universal constant.)

Now we just need to set things appropriately — we have a constant of $(1-\delta)^2\gamma^2$ in the exponent, and we want the constant to be bigger than the growth rate of our δ-net. And the δ-net grows at rate at most 20^n (for example); so as long as γ^* is substantially bigger than $\log 20$ we're okay. □

Now all we need to do is prove these last two key claims — we need to construct a net of the above size, and show that bounding the objective function on the net is enough for us.

§7.5 Proof of Claim 7.15

The first bound is immediate (since I'm only enlarging the space of vectors on which I'm optimizing); it's really the second inequality that's the point. So let $\psi \in \mathbb{S}^{n-1}$ maximize $\|J\psi\|_2$. The key is just that we need to be able to approximate the value of this objective function at the true optimizer, so let's fix such

an optimizer. Because $\mathcal{N}$ is a δ-net, there exists some $\varphi \in \mathcal{N}$ which is basically close to this optimizer, specifically $\|\varphi - \psi\|_2 \le \delta$. And the claim is basically that the objective function at φ is going to be a good approximation for the objectivee function at the true optimizer.

To see this, by the triangle inequality we have

$$\|J\|_{\mathrm{op}} = \|J\psi\|_2 = \|J(\psi - \varphi) + J\varphi\|_2 \le \|J\|_{\mathrm{op}} \|\psi - \varphi\|_2 + \|J\varphi\|_2 .$$

And now if we rearrange things, we're done.

Remark 7.17. Here we're using the fact that we can rewrite

$$\|J\|_{\mathrm{op}} = \sup_{x \neq 0} \frac{\|Jx\|_2}{\|x\|_2}$$

(where now we're considering all vectors, not just ones on the unit sphere).

§7.6 Proof of Claim 7.16 — constructing a δ-net

This tells us if we have a discrete set approximating the sphere in this sense (meaning it's a δ-net), then it really does suffice to look at the maximizer of the objective function over just this δ-net — we don't have to look over the entire continuous sphere. So now we just need to construct a good δ-net.

We're going to prove this essentially by duality — we'll define a sort of dual notion.

Definition 7.18. A *δ-packing* is a maximal collection of points $\mathcal{N} \subseteq \mathbb{S}^{n-1}$ such that the balls of radius δ around the points in $\mathcal{N}$ are disjoint, i.e.,

$$\mathbb{B}_2(\varphi, \delta/2) \cap \mathbb{B}_2(\psi, \delta/2) = \emptyset \quad \text{for all } \varphi \neq \psi \text{ in } \mathcal{N}.$$

Notation 7.19. We use $\mathbb{B}_2(\varphi, \delta/2)$ to denote the Euclidean ball of radius $\delta/2$ around φ, i.e., $\{x \mid \|x - \varphi\|_2 \le \delta/2\}$.

So rather than covering my sphere, I'm trying to pack as many balls inside the sphere as I can. Another way to phrase this is that this is a large set of points such that the distance between any two points is large (at least δ) — it's equivalent to saying that $\|\varphi - \psi\|_2 \ge \delta$.

Here we're really going to want this packing to be *maximal*, meaning that I cannot add any more points to create a larger set while preserving this property (that pairwise these points are far from each other, or that their $(\delta/2)$-neighborhoods are disjoint).

Claim 7.20 — Any maximal δ-packing is also a δ-net.

This is because if there was some point that is not covered — meaning it's far away from all the points in my set — then I could have added it, while preserving the fact that it's a packing (which would contradict its maximality as a packing).

Now we're going to construct this using a greedy algorithm — we imagine we have a sphere, and we pick some point; this cuts out a ball of radius δ, and all those points are no longer alive (I can no longer add any of them to $\mathcal{N}$). Then I pick another point; this also carves out a ball of disallowed points. And I continue this process (repeatedly adding points one at a time, and cutting out a ball) until I must stop.

Student Question. *Can we always guarantee that there is a maximal δ-packing?*

Answer. Yes, by this construction — I can keep adding points until I can no longer add any more.

So this is going to be how we construct this packing; now we want to argue that its cardinality is not too large.

Student Question. *Once you take out enough chunks, won't there be isolated points around?*

Answer. Yes, there might be some highly irregular shapes in the middle. But the balls I'm carving out are of radius δ, not of $\delta/2$ — these are the points that are *disallowed*, using the condition $\|\psi - \varphi\|_2 \geq \delta$. If there's some central point that isn't covered, then it's fine if its radius-δ ball overlaps some of the previously cut out regions; we just need that the radius-$\delta/2$ balls are disjoint.

So we now want to argue that this greedy construction doesn't produce too many points. For this, we'll actually use an argument from before when discussing error-correcting codes — a *volume argument*.

This is where it'll be convenient to switch to the first definition of a packing, with disjoint balls — every point we choose carves out some ball from the surface area of the sphere. And the surface area of a sphere is some finite number; so if I add too many points I'll eventually exceed this surface area, which can't happen. So this gives an upper bound on the number of points we can add.

This is similar to what we did with codes — if I have a good code then many codewords must decode to the same message, and that gave me limits on how good the code could be.

Claim 7.21 — For $\mathcal{N}$ constructed greedily, we have $|\mathcal{N}| \leq (1 + 2/\delta)^n$.

Proof. This is essentially a volume argument. By the definition of a packing, we have a bunch of disjoint balls — we know the balls $\mathbb{B}_2(\varphi, \delta/2)$ are all disjoint. At the same time, we also know they're all contained in some larger ball — we know this is always contained in $\mathbb{B}_2(0, 1 + \delta/2)$. (Imagine I have my sphere, and a ball on the sphere; this is contained in some slightly larger ball, where the radius gets increased by $\delta/2$.)

So because our balls are disjoint, we have

$$\mathrm{Vol}\left(\bigcup_{\varphi \in \mathcal{N}} \mathbb{B}_2(\varphi, \delta/2)\right) = |\mathcal{N}| \,\mathrm{Vol}(\mathbb{B}_2(0, \delta/2)) \leq \mathrm{Vol}(\mathbb{B}_2(0, 1 + \delta/2)).$$

This implies

$$|\mathcal{N}| \leq \frac{\mathrm{Vol}(\mathbb{B}_2(0, 1 + \delta/2))}{\mathrm{Vol}(\mathbb{B}_2(0, \delta/2))} = \left(1 + \frac{2}{\delta}\right)^n. \qquad \square$$

Student Question. *Why is the ball around φ of radius $\delta/2$ contained in a ball centered at 0?*

Answer. We know φ lives within a ball of radius 1, and we're drawing a ball of radius $\delta/2$ around it; so by the triangle inequality, any point in this ball is contained in the slightly bigger ball around the origin.

Remark 7.22. As one quick remark, now we actually know much more about not just the operator norm of a random matrix, but even its entire spectrum — and we even know a sharp constant for what this operator norm should be.

§8 February 26, 2025 — Sub-Exponential random variables

Next week Kuikui will be out of town, so lectures will be over Zoom; a link will be posted on Canvas later today. The second pset is due next Monday.

§8.1 Examples of non-sub-Gaussian random variables

Today we'll continue deriving ever more general classes of Chernoff-type bounds that apply to more and more general situations. Last time, we defined a class of random variables called sub-Gaussian — these were random variables whose tails behave like Gaussians. You can define them by saying their tail is bounded by the tail of a Gaussian, or equivalently by saying that their moment generating function is bounded by that of a Gaussian. There are many equivalent definitions, but these are two of the most important.

Unfortunately, there are lots of random variables we'd like to analyze which are not sub-Gaussian.

Example 8.1

Consider the Laplace distribution, whose density is given by

$$p(x) = \frac{\lambda}{2} e^{-\lambda|x|}.$$

So the decay of the tails is only exponential in $|x|$, rather than in x^2.

Example 8.2

The Poisson distribution is a discrete distribution on the nonnegative integers with probability mass function

$$k \mapsto \frac{\lambda^k e^{-k}}{k!}.$$

These are distributions whose tails are definitely much heavier than the tails of a Gaussian. But we'd still like to have general-purpose concentration inequalities for such random variables; and that's going to be the topic of today's lecture.

We'll start off with another example of a distribution we'd like to analyze, but can't using the previous inequalities:

Example 8.3

Consider a sparse binomial distribution $\mathsf{Bin}(n, \frac{d}{n})$, where d is constant.

This is a setting where the central limit theorem doesn't apply, because the random variables you're adding up have distribution depending on n. In fact, it's known that this distribution converges to a Poisson distribution with parameter d.

Let's see what would happen if you tried applying Chernoff–Hoeffding to this thing. What would it give you? We're looking at a sum of independent random variables, each with success probability $\frac{d}{n}$; so Chernoff–Hoeffding gives that

$$\mathbb{P}_{S \sim \mathsf{Bin}(n, d/n)}[S - d \geq t] \leq \exp(-2t^2/n).$$

This is what you'd get if you just applied Chernoff–Hoeffding (for generic bounded random variables).

If you look at this bound, it's only meaningful if the deviation t is of order at least $\sqrt{n}$ — that's the regime where you get an improvement over the trivial bound of 1. But our expectation is of *constant* order. So

saying the probability it deviates by a $\sqrt{n}$ factor is small is telling you nothing. For instance, if you just computed the variance, you could already get something much stronger — the variance is something like d, which doesn't depend on n.

In some sense, the reason why this occurs is if you look at how we proved Chernoff–Hoeffding, all we used was the fact that our random variables are bounded; and we used some crude proxy for their variance, namely the size of the bounding interval. We never actually used the variance itself. That's why applying some of the general-purpose inequalities we saw previously would not work for a distribution like this.

Another thing to point out is for these distributions with heavier tails, their moment generating function can also be poorly behaved. For instance, the moment generating function of the Laplace distribution is going to look like

$$\mathbb{E}_{X\sim\mathsf{Lap}(\lambda)}[\exp(sX)] = \frac{1}{1-(s/\lambda)^2}.$$

So it'll have degeneracies at the points $\pm\lambda$, where it'll blow up to ∞. In particular, for such random variables, we *really* cannot hope to have a bound on the MGF like we had with sub-Gaussians.

So for heavy-tailed distributions, the moment generating function can also be rather poorly behaved.

§8.2 Sub-exponential random variables

Now we'll define a different class of random variables, which will capture these examples. Similarly to the sub-Gaussian case, it has many equivalent definitions.

Proposition 8.4

For a mean-zero random variable X, the following conditions are equivalent; we say such a random variable X is *subexponential*.

(1) There exists $k > 0$ such that $\mathbb{P}[|X| \geq t] \leq 2\exp(-t/k)$.

(2) There exists $k > 0$ such that $\mathbb{E}[\exp(sX)] \leq \exp(ks^2)$ for all $s \in [-1/k, 1/k]$.

(3) There exists $k > 0$ such that $\mathbb{E}[\exp(|X|/k)] \leq 2$.

Condition (1) says the tails decay *exponentially* fast — so I have t instead of t^2.

Condition (2) says that if I look at the moment generating function, it's bounded by the moment generating function of a Gaussian, but only for s restricted to some interval.

Similarly to the previous lecture, we can use some of these criteria to quantify *how* sub-exponential a random variable is.

Definition 8.5. For a sub-exponential random variable, we define its *sub-exponential norm* as

$$\|X\|_{\psi_1} = \inf\{k > 0 \mid \mathbb{E}[\exp(|X|/k)] \leq 2\}.$$

This quantity basically controls just how sub-exponential my random variable is.

There's also something called a sub-Gaussian norm, defined similarly but with a square:

Definition 8.6. For a sub-Gaussian random variable, we define its *sub-Gaussian norm* as

$$\|X\|_{\psi_2} = \inf\{k > 0 \mid \mathbb{E}[\exp(X^2/k^2)] \leq 2\}.$$

Similarly, this is a measure of how sub-Gaussian X is. Last time we worked with a different measure, the variance proxy (but this one will be more convenient for today); we have

$$\|X\|_{\psi_2}^2 \asymp \|X\|_{vp}^2$$

for sub-Gaussian X (for the same reason as the equivalences of the various definitions of sub-Gaussianity).

There's a relationship between sub-exponential and sub-Gaussian random variables:

Fact 8.7 — If X is sub-Gaussian, then X^2 is sub-exponential, with

$$\|X\|_{\psi_2}^2 = \left\|X^2\right\|_{\psi_1}.$$

So if I have a sub-Gaussian random variable and I square it, that'll give me a heavier tail; and I'll actually get a sub-exponential random variable.

§8.3 Bernstein's inequality

Now let's state a general concentration inequality.

Theorem 8.8 (Bernstein's inequality)

Let $X_1, \dots, X_n$ be independent sub-exponential random variables, and fix $v \in \mathbb{R}^n$. Let $Y = \sum_{i=1}^n v_i X_i$. Then we have

$$\mathbb{P}[Y - \mathbb{E}[Y] \geq t] \leq \exp\left(-\frac{1}{C} \cdot \min\left\{\frac{t^2}{\sum_{i=1}^n v_i^2 \|X_i\|_{\psi_1}^2}, \frac{t}{\max_i |v_i| \|X_i\|_{\psi_1}}\right\}\right).$$

Last lecture we looked at linear combinations of sub-Gaussian random variables; here we're doing the same for sub-exponential random variables.

We get two terms in the exponent. The first looks kind of like a sub-Gaussian, where we have t^2 decay. But in the second term, we only have t decay.

This looks kind of nasty, but let's parse it. We have a sum of a bunch of independent random variables; and there's kind of two ways this random variable can go way past its expectation. Imagine $X_1, \dots, X_n$ are centered. One way Y could exceed its expectation by a lot is if somehow $X_1, \dots, X_n$ all had the same sign, so they all pointed in the same direction and added up to something large. This is what the first term is capturing — you don't expect there to be too much collusion between these random variables (you expect lots of cancellations).

So that's one possibility for how Y could deviate very far from its expectation. Another is if somehow, some individual X_i was extremely large. This occurs with only exponentially small probability, not sub-Gaussian-like probability — because I've only assumed these random variables are sub-exponential, so they could have heavy tails.

So you can think of the first term as capturing 'cancellations' due to independence, and the second term as capturing the case where one X_i is unusually large. This is kind of the intuitive explanation for why we have a minimum of these two terms, and why we have something slightly more complicated in the exponent.

Student Question. *Are there absolute value bars?*

Answer. Sure, you can put $|Y - \mathbb{E}[Y]| \geq t$ on the left and multiply by a factor of 2 on the right.

We'll also mention one very related inequality, which also often goes under the same name; this is specifically for bounded random variables.

Theorem 8.9

Let $X_1, \dots, X_n$ be mean-0 independent random variables bounded in some interval $[-k, k]$. Let $S = \sum_{i=1}^n X_i$ and $\sigma^2 = \mathrm{Var}[S]$. Then we have

$$\mathbb{P}[S \geq t] \leq \exp\left(-\frac{t^2/2}{\sigma^2 t + kt/3}\right).$$

So this is the regime where Chernoff–Hoeffding is applicable; but this inequality takes advantage of not just the boundedness, but also the variance of these random variables.

We have a factor of t in the denominator; that's kind of the same thing as the appearnace of the second term in the original inequality. So you get a Gaussian-like tail fairly close to the expectation, but very far away you only get an exponential tail.

We'll spend the next 20 minutes proving the first theorem; we won't prove the second, but it's of a similar flavor.

Student Question. *Is the second theorem a consequence of the first?*

Answer. Probably not, because the first theorem doesn't reference the variance; but it's kind of the same proof.

As a quick comment, coming back to the example with the sparse binomial, now if you shift these random variables by d/n (so it's centered) and plug in $t = O(\sqrt{d})$, this bound is actually meaningful — so it's much better than applying Chernoff–Hoeffding, precisely because there's the appearance of this variance term. Notably there's no n in the denominator, unlike Chernoff.

Proof of Theorem 8.8. For convenience, we'll define some constants — define

$$k = \max |v_i| \, \|X_i\|_{\psi_1} \quad \text{and} \quad \widehat{\sigma}^2 = \sum_{i=1}^n v_i^2 \, \|X_i\|_{\psi_1}^2$$

(these are the denominators appearing in the bound). We'll do the usual thing of trying to bound the moment generating function of Y and then optimizing our choice of s. We'll consider

$$\mathbb{E}[\exp(s(Y - \mathbb{E}[Y]))].$$

Here s can no longer be arbitrary; instead it'll have to lie in the range $[-1/Ck, 1/Ck]$ — this is because we want to use the bound on $\mathbb{E}[\exp(sX)]$ from the definition of sub-exponential random variables, and this bound only holds on such a range.

Proceeding as usual, we're going to take advantage of independence — this is a sum of random variables, and you can use independence to pull the product out. So by the usual argument, we eventually get that this is upper-bounded by

$$\mathbb{E}[\exp(s(Y - \mathbb{E}[Y]))] \leq \exp\left(C\widehat{\sigma}^2 \cdot s^2\right) \quad \text{for } s \in \left[-\frac{1}{Ck}, \frac{1}{Ck}\right].$$

(This is using independence plus criteria (2) for sub-exponential random variables — we've skipped some steps, but it's the same type of thing we did for Chernoff–Hoeffding.)

And now we can just do Markov — we have

$$\mathbb{P}[Y - \mathbb{E}[Y] \geq t] \leq \inf_{s \in [-1/Ck, 1/Ck]} \exp\left(C\widehat{\sigma}^2 s^2 - st\right)$$

(we want to pick the best choice of s inside this interval for which this upper bound is valid). So we have the same situation where we have some quadratic function inside the exponential that we want to optimize, subject to this range. The minimum comes from the fact that either the best choice of s lies inside the range, in which case you get a Gaussian-like tail; or it lies outside the range, in which case you pick one of the endpoints (and that's where you get the second term).

We won't do all the calculations; they're in the lecture notes. But that's the main idea. □

§8.4 Concentration for norms of random vectors

Last lecture, we proved an upper bound on the probability that the norm of a sub-Gaussian random vector *exceeds* some threshold. But this lecture, we'll see something more surprising — that we actually have really strong concentration around the expectation.

Proposition 8.10

Let X be a random vector with independent $O(1)$-sub-Gaussian entries, such that $\mathbb{E}[X_i] = 0$ and $\mathbb{E}[X_i^2] = 1$ for all i. Then we have

$$\left\| \|X\|_2 - \sqrt{n} \right\|_{\psi_2} \leq O(1).$$

(By $O(1)$-sub-Gaussian, we mean that the sub-Gaussian norm or variance proxy is $O(1)$.)

In this case, we have $\mathbb{E}[\|X\|^2] = n$; so we'd expect that $\mathbb{E}[\|X\|]$ is around $\sqrt{n}$. And we're looking at the random variable that represents the deviation from that; and the claim is that this is $O(1)$-sub-Gaussian.

What does this mean? Imagine we draw the origin, and we draw a ball of radius $\sqrt{n}$, and I look at a *constant*-width band around it — so a very thin annulus or shell around the origin. Then even though this shell is only of constant radius, it actually captures the bulk of the probability mass — it actually contains e.g. 99% of the probability mass, in that if you pick a random variable, it'll lie somewhere inside this $\sqrt{n}$-distance shell. What's most surprising is that the thickness of this shell is of *constant* order — independent of the dimension.

Proof. The idea here is that I've assumed that my random variables X_i are sub-Gaussian, so their square is going to be sub-exponential. So I'll first derive a concentration inequality for the *square* of the Euclidean norm, and then deduce one for just the norm itself (without the square).

First, by Bernstein, we know that

$$\mathbb{P}\left[\left|\frac{1}{n}\|X\|_2^2 - 1\right| \geq t\right] \leq 2 \cdot \exp\left(-\Theta(1) \cdot n \cdot \min\{t^2, t\}\right).$$

(I have a whole bunch of sub-exponential random variables X_i^2, so I want to use Bernstein.) Now I need to massage this inequality to get an inequality for when I replace n with $\sqrt{n}$ and $\|X\|_2^2$ by $\|X\|_2$.

Claim 8.11 — If $x \geq 0$ satisfies $|x - 1| \geq s$, then $\left|x^2 - 1\right| \geq \max\{s, s^2\}$.

This is just going to be convenient for when we turn this into a bound on just $\|X\|_2$ (without the square). It's a very short exercise to prove this.

Now using this claim, we get that

$$\mathbb{P}\left[\left|\frac{1}{\sqrt{n}}\|X\|_2 - 1\right| \geq s\right] \leq \mathbb{P}\left[\left|\frac{1}{n}\|X\|_2^2 - 1\right| \geq \max\{s, s^2\}\right].$$

And now we can use the above inequality on this; so this is upper-bounded by

$$2\exp(-\Theta(1)\cdot n\cdot\min\{\max\{s^2,s^4\},\max\{s,s^2\}\}).$$

This looks like some horrible thing, but it turns out to be extremely nice — it's exactly equal to s^2 (you can check the cases $s\le 1$ and $s\ge 1$ separately). And that's essentially it — this gives you the sub-Gaussian tail. (You should probably replace s by $s/\sqrt{n}$ first.) □

§8.5 Poisson limit of sparse binomial distributions

Now we'll turn to a fact we mentioned at the beginning of the lecture — that if we take a binomial distribution with success probability scaling as $1/n$, then as $n\to\infty$, that converges to not a Gaussian, but instead a Poisson distribution.

One reason we're looking at this is for instance, if you look at the Erdős–Rényi random graph in the sparse regime where the expected degree of every vertex is constant, then this binomial distribution is the distribution of the degree. So we're understanding the distribution of degrees of a sparse Erdős–Rényi random graph as $n\to\infty$.

Definition 8.12. The *total variation distance* between two distributiosn μ and ν on a common state space Ω is defined by

$$\|\mu-\nu\|_{tv}=\frac{1}{2}\sum_{\omega\in\Omega}|\mu(\omega)-\nu(\omega)|.$$

We're going to quantify convergence in distributions using total variation. Previously we proved one alternative characterization for TV distance is if you look at the best possible distinguishing function, how well can you tell whether a sample came from μ vs. ν? Soon we'll see another useful equivalent characterization.

Here's the theorem we want to prove.

Theorem 8.13

For any $d\ge 0$, we have

$$\left\|\mathsf{Bin}\left(n,\frac{d}{n}\right)-\mathsf{Poi}(d)\right\|_{tv}\le\frac{d^2}{n}.$$

§8.5.1 Coupling

To prove this, we'll first introduce a third way of thinking of TV distance. One is as the regular L^1 distance between vectors; we saw previously you can also write it as picking the best possible test function of distinguishing these two distributions, and looking at the advantage of using that to solve a hypothesis testing problem.

Here's a third way, based on the notion of *coupling*.

Definition 8.14. A *coupling* ξ between two distributions μ and ν is a distribution on $\Omega\times\Omega$ such that

$$\mu(x)=\sum_y\xi(x,y)\text{ for all }x\in\Omega$$
$$\nu(y)=\sum_x\xi(x,y)\text{ for all }y\in\Omega.$$

A coupling is really just a correlated way of sampling from these distributions — so it's a distribution on *pairs* of states with marginals μ and ν. One way to think about this is I'm writing down a matrix ξ, with the conditions that if I sum along rows I get μ, and if I sum along columns I get ν.

So one way to think about a coupling is if I want to sample an element according to μ or one according to ν, one way I can do so is sample according to a coupling — so there's a possibly correlated pair — and I throw away one of them, then that gives me a sample for either μ or ν.

First, we claim that couplings *exist*.

Example 8.15

One example of a coupling is

$$\xi(x,y) = \mu(x)\nu(y),$$

which corresponds to sampling independently from μ and ν (so there's no correlation whatsoever).

Here's a few more examples, before we use couplings to prove the theorem.

Example 8.16

If $\mu = \nu$, then we can couple these distributions perfectly — we can define

$$\xi(x,y) = \begin{cases} \mu(x) & \text{if } x = y \\ 0 & \text{if } x \neq y. \end{cases}$$

Here's one more example, which is very cute: It's a way to couple two coins with possibly different biases.

Example 8.17

Let $\mu = \mathsf{Ber}(p)$ and $\nu = \mathsf{Ber}(q)$, where $p \leq q$. Then we can define a correlated way of sampling the coins:

(1) Sample a uniform real $U \sim \mathsf{Unif}[0,1]$.

(2) We'll then use this random variable as a threshold, and output heads or tails based on which range it falls in — we output

$$(X,Y) = \begin{cases} (\mathrm{H},\mathrm{H}) & \text{if } 0 \leq U \leq p \\ (\mathrm{T},\mathrm{H}) & \text{if } p < U \leq q \\ (\mathrm{T},\mathrm{T}) & \text{if } q < U \leq 1. \end{cases}$$

So I'm defining a correlated way of sampling these two coins (where the first coin has probability p, and the second q).

Let's track what happens with the first coin. Then I'm sampling a random number between 0 and 1, and I output heads if it lands in the range $[0,p]$, which occurs with probability p; and you can check the same thing for the other coin.

These are definitely correlated coins — for example, if $p = q$ then you'll only be in the first or last cases (you'll always output the same thing). In general, we have

$$\mathbb{P}[X \neq Y] = |p - q|.$$

So the probability we get different outcomes for the two coins is the difference between the parameters of the distributions.

§8.5.2 Coupling and total variation distance

One reason to show this example is it suggests a connection between couplings and TV distance — you can use the probability that under the coupling the two outcomes are not equal as some measure of distance. And in fact, it turns out to be *equal* to the total variation distance.

Lemma 8.18 (Coupling lemma)

For every coupling ξ of two distributions μ and ν, we have

$$\mathbb{P}_{(X,Y)\sim\xi}[X \neq Y] \geq \|\mu - \nu\|_{tv}.$$

Moreover, there exists a coupling ξ achieving equality.

So in other words, a third way of interpreting TV distance is that I take the best possible coupling between them (the one that minimizes the chance the two outcomes disagree).

In the interest of time, we'll just prove the upper bound; if we have time we'll also do the equality case later. But for now we'll just prove the upper bound and use that to prove Theorem 8.13.

Proof. Really, the key observation is that because I'm looking at the probability X and Y disagree (or equivalently, the probability they do agree), I'm really looking at the diagonal entries of the matrix ξ. And these diagonal entries are bounded — for every x, we have

$$\xi(x,x) \leq \min\{\mu(x), \nu(x)\}.$$

This is just because whatever entry I put here, it's certainly upper-bounded by the sum of the entries in that row, which is $\mu(x)$; and it's also upper-bounded by the sum of entries in that column, which is $\nu(x)$; so it's certainly upper-bounded by the minimum of the two.

In particular, this implies

$$\mathbb{P}[X = Y] \leq \sum_x \min\{\mu(x), \nu(x)\}.$$

Now we just need to show that if I take 1 minus these two things, the right-hand side gives me the total variation distance ($1 - \mathbb{P}[X = Y]$ is definitely the thing on the left-hand side of the lemma).

So this implies

$$\mathbb{P}_\xi[X \neq Y] \geq 1 - \sum_x \min\{\mu(x), \nu(x)\}.$$

And I can replace 1 by $\sum_x \mu(x)$, because μ is a distribution; so then this becomes

$$\sum_x (\mu(x) - \min\{\mu(x), \nu(x)\}).$$

And now we're essentially done — all the terms with $\mu(x) \geq \nu(x)$ give 0, so this becomes

$$\sum_{x:\nu(x)<\mu(x)} (\mu(x) - \nu(x)).$$

And this is exactly equal to $\|\mu - \nu\|_{tv}$ (since we saw earlier that the indicator function for whether $\nu(x) < \mu(x)$ is actually the optimal distinguisher between these two distributions). □

It turns out you can also design a coupling that makes all these inequalities into equalities, but we probably don't have time to do that.

§8.5.3 Proof of Theorem 8.13

Now we'll use this inequality to prove our convergence. I want to bound the total variation distance betwen $\mathsf{Bin}(n, d/n)$ and $\mathsf{Poi}(d)$; and one way to do this is to construct a coupling.

Before we do this, similarly to how a binomial distribution can be decomposed as a sum of Bernoulli random variables, we'd like to also decompose the Poisson distribution.

Claim 8.19 — If $X \sim \mathsf{Poi}(d_1)$ and $Y \sim \mathsf{Poi}(d_2)$ are independent, then $X + Y \sim \mathsf{Poi}(d_1 + d_2)$.

This is a simple calculation. But what it tells us is that $\mathsf{Poi}(d)$ can be decomposed as the sum of n independent $\mathsf{Poi}(d/n)$ random variables. And similarly, $\mathsf{Bin}(n, d/n)$ is of course the sum of n independent $\mathsf{Ber}(d/n)$ random variables.

So if I want to couple these two distributions, it kind of suffices to couple all these little guys and the corresponding sums.

So let's say we have i.i.d. $X_1, \dots, X_n \sim \mathsf{Ber}(d/n)$ and i.i.d. $Y_1, \dots, Y_n \sim \mathsf{Poi}(d/n)$, so that $X = \sum_{i=1}^n X_i \sim \mathsf{Bin}(n, d/n)$ and $Y = \sum_{i=1}^n Y_i \sim \mathsf{Poi}(d)$. Nwo we want to bound

$$\left\| \mathsf{Bin}\left(n, \frac{d}{n}\right) - \mathsf{Poi}(d) \right\|_{tv}.$$

This is certainly always upper-bounded by the probability that *one* fo the pairs of X_i's and Y_i's are different — because if $X_i = Y_i$ for all i, then definitely their sums are equal. So we get

$$\left\| \mathsf{Bin}\left(n, \frac{d}{n}\right) - \mathsf{Poi}(d) \right\|_{tv} \le \sum_{i=1}^n \mathbb{P}[X_i \neq Y_i],$$

under any coupling. (This is just using the coupling interpretation of total variation distance.)

And now this is something that's much easier to analyze — if I take the best possible coupling, I'm looking at just the total variation distance between a Bernoulli and a Poisson, and this is much simpler to calculate. So we get

$$\|\mathsf{Bin}(n, d/n) - \mathsf{Poi}(d)\|_{tv} \le n \cdot \|\mathsf{Ber}(d/n) - \mathsf{Poi}(d/n)\|_{tv}.$$

And now this thing can be computed very explicitly — the Bernoulli is a random variable just supported on 0 and 1, so we can compute the total variation distance just looking there (or you could explicitly construct a coupling). And this turns out to be

$$n \cdot \frac{d}{n}\left(1 - e^{-d/n}\right).$$

Now the n's cancel, and usijng the inequality $1 - e^{-x} \le x$, we get that this is upper-bounded by d^2/n.

This is why it's very convenient to have multiple equivalent definitions for the same thing (here, TV distance) — you can use whichever one is most convenient. So we used coupling to translate an upper bound on the TV to just upper-bounding the TV between *components*; and then we could do an explicit calculation.

This notion of coupling will come up again later in the course. This gives a very nice way to sort of 'compare' distributions; so when we look at Markov chains, this concept is going to come back.

§8.5.4 Equality case for the coupling lemma

In the last few minutes, we'll sketch a proof of the equality case of the coupling lemma — how to construct a coupling that actually achieves equality.

Basically, I want a coupling ξ that satisfies

$$\xi(x,x) = \min\{\mu(x), \nu(x)\}.$$

This is the only inequality we used in the proof of the bound, so if I make this an equality, then everything's an equality. So I just need to construct a coupling where along the diagonal, I actually put the corresponding row and column.

Let's think about what this means — let $A = \{x \mid \mu(x) > \nu(x)\}$, $B = \{x \mid \mu(x) < \nu(x)\}$, and $C = \{x \mid \mu(x) = \nu(x)\}$. And let's think about what this matrix ξ must look like. Let's break it up into three blocks, corresponding to these sets A, B, and C.

On the diagonal, I'm putting the minimums $\xi(x,x) = \min\{\mu(x), \nu(x)\}$. And I know that every row must sum to μ, and every column must sum to ν. So in particular, all the blocks in the C rows and columns must be a whole bunch of 0's — because the diagonal entries saturate both the rows and columns (I put the maximum amount of probability mass on the diagonal, so everything else must be 0).

Similarly, in the columns corresponding to A, the non-diagonal entries must be 0. And the same happens with B and rows. That really just leaves the $A \times B$ block — that's the only place we have freedom when we construct our coupling.

So now you can imagine there's various constraints that these entries must satisfy — the rows should sum to $\mu(x) - \nu(x)$, and similarly with the columns. You can imagine an iterative algorithm to fill out the entries of the matrix; or there's also an explicit formula for what you can put in these entries. (The formula for the optimal coupling is in the notes.)

§9 March 3, 2025

§9.1 Martingales

Definition 9.1. We say a sequence of random variables $\{Y_n\}_{n\geq 0}$ is a *martingale* with respect to another sequence of random variables $\{X_n\}_{n\geq 0}$ if:

- $\mathbb{E}[|Y_n|] < \infty$ for all n.
- Y_n is a function of $X_0, \ldots, X_n$.
- $\mathbb{E}[Y_{n+1} \mid X_0, \ldots, X_n] = Y_n$ for all n.

We say $\{Y_n\}$ is a martingale to mean that it's a martingale with respect to itself.

§9.2 Some examples

Example 9.2

Consider the Galton-Watson branching process $\{Z_\ell\}_{\ell\in\mathbb{N}}$ where to sample Z_ℓ, we sample $Z_{\ell-1}$ many independent copies from some offspring distribution ξ on $\mathbb{N}$, and add them together. Then $\{Z_\ell/\mu^\ell\}_{\ell\in\mathbb{N}}$ is a martingale, where μ is the mean of ξ.

Proof. We'll just check the third condition (which is the important one) — this means we want to consider

$$\mathbb{E}\left[\frac{Z_{\ell+1}}{\mu^{\ell+1}} \,\middle|\, \frac{Z_0}{\mu^0}, \ldots, \frac{Z_\ell}{\mu^\ell}\right].$$

This sequence of random variables is Markovian, so once we know the previous value Z_ℓ, the ones before that don't matter; so this is

$$\mathbb{E}\left[\frac{Z_{\ell+1}}{\mu^{\ell+1}} \,\middle|\, \frac{Z_\ell}{\mu^\ell}\right].$$

And by the way our process was generated, this is exactly

$$\frac{Z_\ell \cdot \mu}{\mu^{\ell+1}} = \frac{Z_\ell}{\mu^\ell}.$$

□

Let's see the example of gambling, which was perhaps one of the original reasons for developing this notion.

Example 9.3

Suppose I'm at a casino where the game is the following: With probability 1/2 you win what you bet, and with probability 1/2 you lose what you bet.

Suppose you have a gambling strategy — this means you have a bunch of functions f_n, where $f_n : \{\mathsf{Win}, \mathsf{Loss}\}^n \to \mathbb{R}$ is a function of your history of wins and losses up to time n and tells you how much you'll bet on the nth round, based on your history of outcomes.

Consider the random variable W_n, representing your winnings — so

$$W_n = W_{n-1} + f_n(X_0, \dots, X_{n-1}) \cdot X_n$$

(where $X_n \sim \mathsf{Unif}\{\pm 1\}$ tells you whether or not you won or lost in the nth round).

Then the sequence $\{W_n\}_{n \geq 0}$ is a martingale.

Clearly the amount of money you win is *some* function of your history of wins and losses. And just by the fact that X_n has mean 0, you can also show that this third martingale property is satisfied. So really, this follows from the fact that $\mathbb{E}[X_n] = 0$.

One thing that's interesting about this example is perhaps we've seen a related class of stochastic processes called *Markov chains*. This is an example of a martingale which is not a Markov chain (and there are many Markov chains you can construct that are not martingales). These are another class of stochastic processes that have a sort of similar flavor to martingales; but one reason to show this example is that they're different (here the amount of money you bet on the next round can depend on the *entire* history of wins and losses).

§9.2.1 Doob martingales

Now we'll talk about a third class of martingales, which is going to come up quite frequently, called *Doob martingales*. These martingales sort of capture the following intuition — you want to imagine $X_1, \dots, X_n$ as the 'information' you have about a given function.

Consider a game between two players, Alice and Bob. The rules of the game are as follows: Before the game starts, Alice and Bob agree on:

- A function $f : \mathbb{R}^n \to \mathbb{R}$.
- A distribution μ on $\mathbb{R}^n$ (such that $\mathbb{E}_\mu[f] < \infty$).

Alice will sample $X \sim \mu$ and compute $Y = f(X)$ (she doesn't reveal X or Y to Bob). Bob's goal is to estimate Y.

For instance, you can imagine μ as a Gaussian, and f as the sum of the coordinates. (So f is some deterministic function we've fixed before; there's only randomness in the choice of *input* to f.)

Bob has no information about what X is, so one very natural strategy is to compute the mean of f under μ. This makes intuitive sense, and it turns out that it's optimal in a sense, in that it's the estimator that achieves the best squared error — we have

$$\mathbb{E}_{X\sim\mu}[f(X)] = \arg\min_{s\in\mathbb{R}} \mathbb{E}_Y[(s-Y)^2].$$

This is the case if Bob has no information whatsoever. But what if Alice decides to be generous, and reveals *some* information about the coordinates? If Alice reveals $X_1, \ldots, X_k$, then Bob has an analogous strategy — he can refine his estimate based on what Alice has revealed. So he can output the conditional expectation

$$Y_k = \mathbb{E}_\mu[f(X) \mid X_1, \ldots, X_k].$$

This is some function of $X_1, \ldots, X_k$; and the value of this random variable only depends on the randomness of $X_1, \ldots, X_k$. For example, if you look at the extreme case $k = n$, if Alice reveals all the coordinates, then Bob is just directly computing Y. And if $k = 0$, this recovers the original strategy, where you have no information. So this interpolates smoothly between these two extremes.

Lemma 9.4

The sequence $\{Y_k\}$ is a martingale with respect to $\{X_k\}$.

Proof. Suppose I want to consider

$$\mathbb{E}[Y_{k+1} \mid X_1, \ldots, X_{k+1}].$$

Unwinding the definition of Y_{k+1}, I can write this as

$$\mathbb{E}[\mathbb{E}[f(X) \mid X_1, \ldots, X_{k+1}] \mid X_1, \ldots, X_k].$$

(This looks a bit messy, but it is just directly using the definition.)

So I have these two expectations. You should think of the inner expectation as being with respect to X_{k+2}, $\ldots$, X_n (so we're averaging out the contribution from all the later coordinates); and the outer expectation is with respect to X_{k+1}.

And by the law of total expectation, or the tower property of conditional expectations, this is exactly

$$\mathbb{E}[f(X) \mid X_1, \ldots, X_k]$$

(these two expectations average out over all the randomness in $X_{k+1}, \ldots, X_n$, and what's left is $X_1, \ldots, X_k$). And by definition this is exactly Y_k. □

The crucial step here is the use of this law of total expectation.

Very concretely, you can see there's an information game being played here — the random variables X_k sort of reveal progressively more and more information about some input into your deterministic function f, and the Y_k's are the best estimators you could use for the value of f.

Student Question. *Do martingales have applications in machine learning?*

Answer. Kuikui expects they do; certainly the concentration inequalities we'll derive through martingales have applications.

Let's see one example of a Doob martingale.

Example 9.5 (Balls and bins)

Suppose you have m balls and n bins, and each ball is thrown independently into a uniformly random bin. Let f be the function on bin assignments which counts the number of empty bins, and let Y be the number of empty bins under this random process. Then we can define

$$Y_k = \mathbb{E}[\#\text{empty bins} \mid \text{where the first } k \text{ balls land}].$$

Then $\{Y_k\}$ is a Doob martingale (with respect to the sequence of where each ball lands).

This kind of balls-and-bins process is often used when we want to analyze the theoretic properties of load-balancing data structures and similar things. One statistic you might want to track about this process is the distribution of the set of *empty* bins — if you imagine the bins as being processors that you want to assign tasks to, the empty ones are the ones which are wasted potential in some sense. So we'd like to study the random variable associated to this function. And we can study this using the Doob martingale; and we can use some of the concentration inequalities we'll prove later in this lecture to study Y.

§9.3 Expectation and variance of martingales

Now let's start studying concentration for martingales. Let's first get a handle on the expectation and variance — the usual stuff.

Lemma 9.6

Let $\{Y_n\}$ be a martingale with respect to $\{X_n\}$. Then

$$\mathbb{E}[Y_n] = \mathbb{E}[Y_0],$$
$$\mathbb{E}[(Y_n - Y_0)^2] = \sum_{k=1}^{n} \mathbb{E}[(Y_k - Y_{k-1})^2].$$

You should think of $\mathbb{E}[(Y_n - Y_0)^2]$ as essentially being the variance of Y_n (you can think of the case where Y_0 is fixed, so it's some deterministic number; then this is $\mathbb{E}[(Y_n - \mathbb{E}[Y_n])^2]$).

Proof. The first claim is essentially just a direct consequence of the tower property of conditional expectations, so we won't go through that. But we'll spend more time on the second claim.

The second claim is equivalent to saying that

$$\operatorname{Var}[Y_n - Y_0] = \sum_{k=1}^{n} \operatorname{Var}[Y_k - Y_{k-1}]$$

(because the expectations of all these random variables are 0 — we have $\mathbb{E}[Y_n - Y_0] = 0$ and $\mathbb{E}[Y_k - Y_{k-1}] = 0$ for all k). And in order for the variance of a sum to decompose, we just need the terms to be pairwise uncorrelated — so we just need to prove that for all $k < \ell$, we have

$$\mathbb{E}[(Y_\ell - Y_{\ell-1})(Y_k - Y_{k-1})] = 0.$$

If we can do this, then the variance will decompose as a sum. And for this, we just want to use the definition of a martingale. In particular, we can take an expectation where we first condition on $X_0, \ldots, X_k$, and then take another expectation with respect to the randomness of $X_0, \ldots, X_k$ — so this is

$$\mathbb{E}[\mathbb{E}[(Y_\ell - Y_{\ell-1})(Y_k - Y_{k-1}) \mid X_0, \ldots, X_k]].$$

And $Y_k - Y_{k-1}$ is some deterministic function of $X_0, \dots, X_k$; so once I've fixed the values of $X_0, \dots, X_k$, this just becomes some number. This means I can pull it out of the inner expectation, so this becomes

$$\mathbb{E}[(Y_k - Y_{k-1})\mathbb{E}[Y_\ell - Y_{\ell-1} \mid X_0, \dots, X_k]].$$

(This is just because once I've fixed $X_0, \dots, X_k$, the values of Y_k and Y_{k-1} are completely determined, so there's no longer any randomness.) And

$$\mathbb{E}[Y_\ell - Y_{\ell-1} \mid X_0, \dots, X_k] = 0$$

by the martingale property; so this entire thing is equal to 0.

So in other words, these increments $Y_k - Y_{k-1}$ are pairwise uncorrelated. □

§9.4 The Azuma–Hoeffding inequality

So we've computed the expectation and variance; now let's try to prove sub-Gaussian-type concentration inequalities. This is basically an analog of Chernoff–Hoeffding from before — we'll consider the case where these increments $Y_k - Y_{k-1}$ are bounded random variables.

Theorem 9.7 (Azuma–Hoeffding)

Let $\{Y_n\}$ be a martingale with respect to $\{X_n\}$, and suppose there exists a sequence of positive constants $\{c_n\}$ such that $|Y_n - Y_{n-1}| \le c_n$ for all n (with probability 1). Then for all n and all $t \ge 0$, we have

$$\mathbb{P}[Y_n - Y_0 \ge t] \le \exp\left(-\frac{t^2}{2\sum_{k=1}^n c_k^2}\right).$$

So we have bounded increments. Again think of Y_0 as being some deterministic thing, so that $Y_0 = \mathbb{E}[Y_n]$.

You can also generalize this statement to the case where these increments are not necessarily bounded, but sub-Gaussian or sub-exponential; but for simplicity we'll focus on the case where they're bounded (almost surely). This really is just the analog of Chernoff–Hoeffding, but now for martingales as opposed to independent random variables.

Proof. The proof is the exact same style of argument as we did for the previous Chernoff–Hoeffding type inequalities — we're going to bound the moment generating function of $Y_n - Y_0$, and then we're going to use the usual Markov inequality plus an optimization over some parameter.

So let's look at the moment generating function. And now we're going to really use the martingale property to bound it inductively.

So we'll fix $s > 0$ (this is the parameter we'll optimize over when we apply Markov); we're going to look at $\mathbb{E}[\exp(s \cdot (Y_n - Y_0))]$. I'm going to first bound this in the case where we condition over $X_0, \dots, X_{n-1}$, and then take an additional expectation on the outside — so we consider

$$\mathbb{E}[\exp(s \cdot (Y_n - Y_0)) \mid X_0, \dots, X_{n-1}].$$

We have $Y_n - Y_0$, so let's decompose this as

$$(Y_n - Y_{n-1}) + (Y_{n-1} - Y_0)$$

(we're trying to do an inductive argument). Then we can write this as

$$\mathbb{E}[\exp(s \cdot (Y_{n-1} - Y_0)) \cdot \exp(s \cdot (Y_n - Y_{n-1})) \mid X_0, \dots, X_{n-1}].$$

And now again using the fact that the Y's are functions of the X's, the quantity $Y_{n-1}, \dots, Y_0$ is fixed once I've fixed the values of $X_0, \dots, X_{n-1}$. So I can push the expectation inside to just the second term; and this becomes

$$\exp(s \cdot (Y_{n-1} - Y_0)) \cdot \mathbb{E}[\exp(s \cdot (Y_n - Y_{n-1})) \mid X_0, \dots, X_{n-1}].$$

And now let's use the fact that these increments are bounded almost surely — by Hoeffding's lemma, we have

$$\mathbb{E}[\exp(s \cdot (Y_n - Y_{n-1})) \mid X_0, \dots, X_{n-1}] \le \exp\left(-\frac{c_n^2 s^2}{2}\right).$$

Now if I take a big expectation over $X_0, \dots, X_{n-1}$, this implies

$$\mathbb{E}[\exp(s \cdot (Y_n - Y_0))] \le \exp\left(\frac{c_n^2 s^2}{2}\right) \cdot \mathbb{E}[\exp(s \cdot (Y_{n-1} - Y_0))].$$

And now I can induct. In particular, if I unravel the induction, this is upper-bounded by

$$\exp\left(\frac{s^2 \sum_{k=1}^n c_k^2}{2}\right).$$

And this holds for all $s \in \mathbb{R}$. So this is the key bound we get on the moment generating function of $Y_n - Y_0$. And now we do the usual Markov inequality plus optimizing over the best choice of s. (So really, the only new ingredient was the inductive bound on the moment generating function, which we got from the definition of a martingale; other than that, the argument is the same as what we've seen before — bounding the moment generating function, applying Markov, and optimizing over the parameter s we've introduced.) □

§9.5 The bounded differences inequality

Now we want to use this to say something about concentration for more general classes of functions of independent random variables — this is going to generalize much farther than just sums of independent random variables.

Corollary 9.8 (McDiarmid's inequality)

Let $X_1, \dots, X_n$ be *independent* random variables, and let f be a function of $X = (X_1, \dots, X_n)$. Assume that f is L-Lipschitz with respect to the Hamming distance, i.e.,

$$|f(x) - f(y)| \le L \cdot \#\{i \in [n] \mid x_i \ne y_i\}.$$

Then for all $t \ge 0$, we have

$$\mathbb{P}[f(X) - \mathbb{E}[f(X)] \ge t] \le \exp\left(-\frac{t^2}{2L^2 n}\right).$$

(This is also sometimes called the bounded differences inequality.)

So we're putting a Lipschitz condition on how much f fluctuates — by perturbing a single coordinate, you can perturb the value of f by at most L. (The Hamming distance between two vectors is just the number of coordinates at which they differ.)

This is a very general-purpose concentration inequality, for a very general class of functions; it gets used in all sorts of places, and we'll see a very nice application towards the end of the lecture.

Proof. We're just going to invoke Azuma–Hoeffding for an appropriate martingale — namely, the Doob martingale that we defined earlier.

So we consider the Doob martingale

$$Y_k = \mathbb{E}[f(X_1, \dots, X_n) \mid X_1, \dots, X_k].$$

This is a random variable whose randomness depends on the choice of the first k coordinates $X_1, \dots, X_k$ (and we're taking an expectation with respect to all the remaining coordinates).

Now let's look at the increments $Y_k - Y_{k-1}$. Observe that the Lipschitz condition of f implies that

$$|Y_k - Y_{k-1}| \le L$$

(with probability 1) for all k, just by using the definition of the Y_k's. So now we can conclude the proof by using Azuma–Hoeffding, and we're done. □

Student Question. *Where did we use the independence of the input?*

Answer. When we said that $|Y_k - Y_{k-1}| \le L$. (We maybe handwaved this a bit, but it's worth thinking about.)

As a remark, there are many concentration inequalities in the literature of this flavor — if you have nice regularity properties of your function f, then you get concentration (at least with respect to a nice distribution, like the one where the coordinates are independent). In particular, there are related concentration inequalities for Lipschitzness with respect to other metrics. As one example:

Theorem 9.9 (Gaussian concentration inequality)

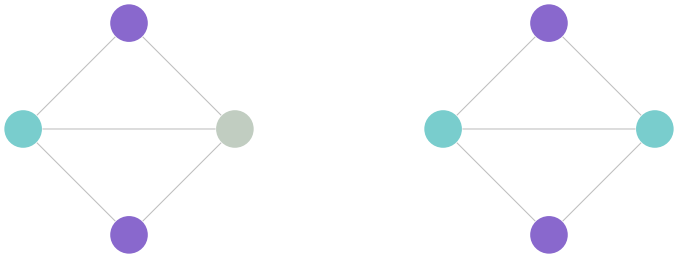
If $X \sim \mathcal{N}(0, I)$ and $f : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz with respect to the Euclidean norm $\|\bullet\|_2$ on $\mathbb{R}^n$, then

$$\mathbb{P}[|f(X) - \mathbb{E}[f(X)]| \ge t] \le 2\exp\left(-\frac{t^2}{2L^2}\right).$$

§9.6 Chromatic number of a random graph

Now we'll turn to discussing a very neat application of these concentration inequalities to the chromatic number of a graph.

Definition 9.10. For a graph $G = (V, E)$ and $q \in \mathbb{N}$, a *proper q-coloring* of G is an assignment $\sigma : V \to \{1, \dots, q\}$ such that neighboring vertices do not receive the same color — i.e., for all $uv \in E$, we have $\sigma(u) \ne \sigma(v)$.



(Here the left-hand side is a proper coloring; the right-hand side isn't, because the diagonal edge violates the constraint on neighboring vertices not having the same color.)

Definition 9.11. The *chromatic number* of a graph G, denoted $\chi(G)$, is the smallest $q \in \mathbb{N}$ such that G admits a proper q-coloring.

We want to use McDiarmid's inequality to study the chromatic number of a random graph.

This is some extremely complicated function of the graph — just from a complexity-theoretic perspective, even *approximating* the chromatic number of a graph up to a $n^{1/2}$ multiplicative factor is a very hard problem (as hard as solving SAT — more precisely, it's NP-hard).

So this is an insanely complex function — it's NP-hard even to approximate. But for random graphs, we'll be able to say something very strong about the chromatic number — we'll be able to say something very nice about the behavior of this function for an *average*-case input.

So let's consider the Erdős–Rényi random graph $\mathcal{G}(n,p)$ — recall that this is a random graph formed by adding edges independently with probability p (for every pair of vertices, I add the edge between them with probability p). We're working in the dense regime, where p is constant.

§9.6.1 A lower bound on the expectation

We'll claim two facts. The first is that we can give a lower bound on the order of magnitude where this chromatic number lies.

Fact 9.12 — For any (fixed) $p > 0$, we have

$$\mathbb{E}[\chi(\mathcal{G}(n,p))] \gtrsim \frac{n}{\log n}.$$

So this is roughly how large we expect the chromatic number to be — the number of colors we'd expect to need in order to properly color a dense Erdős–Rényi graph.

This is actually not too difficult to prove; we'll briefly say in words how you'd prove it. It goes by the following observation: imagine that I look at my graph, and let's say I've properly colored it. The key observation is that if I look at a color class — e.g., all the vertices colored orange — then this is an independent set (and similarly for the blue vertices, and the red vertices, and so on).

Definition 9.13. An *independent set* is a subset of vertices such that no pair of vertices are connected by an edge.

This is directly from the definition of a coloring — all the vertices assigned the same color can't have any edges between them.

In particular, this implies that the chromatic number is always lower-bounded by

$$\chi(G) \geq \frac{n}{|\alpha(G)|},$$

where $\alpha(G)$ is the size of the maximum independent set. (If I can color the graph with very few colors, then it must have a large independent set.) In particular, if I can upper-bound the size of the largest independent set, that'll give a lower bound on the number of colors.

And it turns out you can show, using just the first moment method, that

$$|\alpha(\mathcal{G}(n,p))| \lesssim \log n$$

(with high probability).

Remark 9.14. We will revisit the maximum independent set in Erdős–Rényi in a future lecture; but for now we'll take this fact as a given. We actually know even much sharper results — it's known that

$$\mathbb{E}[\chi(\mathcal{G}(n,p))] = (1 \pm o(1)) \cdot \frac{n}{2\log_{1/(1-p)} n}.$$

But we're not going to need this; all we'll really need is that its expectation is on the order of $n/\log n$.

§9.6.2 Concentration of the chromatic number

Now here's our theorem — that the chromatic number is very well-concentrated.

Theorem 9.15

For any $p > 0$ and $0 < \varepsilon < 1$, we have

$$\mathbb{P}\left[\chi(\mathcal{G}(n,p)) \notin (1 \pm \varepsilon)\mathbb{E}[\chi(\mathcal{G}(n,p))]\right] \leq 2\exp\left(-\Omega\left(\frac{\varepsilon^2 n}{\log^2 n}\right)\right).$$

(The constant may depend on p.) So the probability that the chromatic number deviates from its expectation by more than a $(1+\varepsilon)$-multiplicative factor is exponentially small.

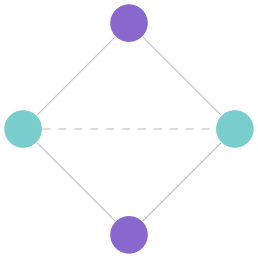
This is a really cool result because the chromatic number of a graph is a hopelessly complicated function of a graph — even computing this function is really hard. But using fairly generic tools (concentration bounds and so on), we can actually show that at least for a randomly chosen graph, its chromatic number is very strongly concentrated around its expectation (and we also understand what order its expectation is). So we have a good understanding of the chromatic number in the *average* case, in some sense.

Proof. I want to use McDiarmid's inequality. One very natural approach to do this is to view χ as a function of the graph. And I want this function to be 1-Lipschitz with respect to perturbing the graph (or really L-Lipschitz for any constant L, but we'll show we can take $L = 1$).

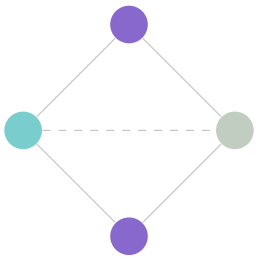
Perhaps a very natural way to perturb the graph is by adding or deleting edges — every edge is included with some probability, so I have a collection of independent random variables of size n^2.

Claim 9.16 — The chromatic number χ is 1-Lipschitz with respect to adding or deleting single edges.

We're not going to spend time proving this, but it's not too difficult to see — imagine I take a graph, and it has some chromatic number; and now I add a new edge to this graph. Then we claim that the chromatic number of the new graph is at most the chromatic number of the old graph plus one. One way to see this is that I can take a coloring from the old graph, and I can assign those colors to the vertices of the new graph. There might be *one* violation of the constraints — the constraint on this new edge I added. And to fix this new violation, I can introduce just one new color, and assign that color to one of the endpoints of this new edge. As a picture, let's say I have the following graph (which is 2-colorable). And say I introduce the dashed edge.



Because of the new edge, now there's going to be a violation (on the two blue vertices). But I can introduce a new color, and color one of the endpoints with that new color.



So I introduce a *single* new color (let's say green); and this implies $\chi(G') \le \chi(G)+1$ (and certainly we have $\chi(G') \ge \chi(G)$, since I only have more constraints in G'). And you can argue the exact same way if you delete an edge instead of adding an edge — it'll change the chromatic number by at most one.

Now combining this claim with McDiarmid's inequality, we get that

$$\mathbb{P}\left[\chi(\mathcal{G}(n,p)) \notin (1\pm\varepsilon)\mathbb{E}[\chi(\mathcal{G}(n,p))]\right] \le 2\cdot\exp\left(-\frac{\varepsilon^2\mathbb{E}[\chi(\mathcal{G}(n,p))]^2}{2\cdot\binom{n}{2}}\right)$$

(here L is 1, and $\binom{n}{2}$ is the number of random variables). And now if we use the fact that $\mathbb{E}[\chi(\mathcal{G}(n,p))] \gtrsim n/\log n$, this is at most

$$2\cdot\exp\left(-\Omega\left(\frac{\varepsilon^2\cdot n^2}{(\log n)^2\cdot n^2}\right)\right).$$

So we got something that's exponential in $1/(\log n)^2$. And this is kind of useless — this goes to 1 as $n\to\infty$, and we wanted something that goes to 0.

So this is not going to work. But we can still use the same strategy; we just need to tweak things a bit.

The problem with this analysis was that there's too many random variables here — there were $\Theta(n^2)$ random variables, one for each pair of vertices. We'll instead do a more efficient way to use McDiarmid — more precisely, we'll consider a more efficient representation of the chromatic number.

What we want is to have n random variables, instead of n^2. But these random variables are going to be more complex — instead of being $\{0,1\}$-valued, they'll be *vector*-valued (or given by subsets). But we'll still want 1-Lipschitzness with respect to an arbitrary perturbation of one of these coordinates.

Let's say I order the vertices arbitrarily; call them $v_1,\dots,v_n$. Then we'll define

$$E_t = \{v_tv_i \in E \mid 1\le i\le t-1\}.$$

So this is a random subset of edges — all the edges that connect v_t to the vertices before it.

And now I'm going to view my chromatic number as a function of $E_1,\dots,E_n$ — because once I've fixed the neighborhoods of all the vertices, that uniquely determines the graph.

But now I have n random variables, and because these correspond to disjoint sets of edges (i.e., $E_t \subseteq \{v_tv_i \mid 1\le i\le t-1\}$, which means the E_t's are determined by disjoint sets of vertex pairs), $E_1,\dots,E_n$ are jointly independent.

This seems much better, because now I only have n random variables, as opposed to n^2. Now I just need to verify that the chromatic number is still 1-Lipschitz.

> **Claim 9.17 —** The chromatic number χ is still 1-Lipschitz with respect to arbitrary perturbation of a single E_t.

So I can take a single vertex and arbitrarily mess around with the incident edges to this vertex; and no matter how I mess around with its incident edges, I'm only going to change the chromatic number by at most 1. We'll leave this as a quick exercise; it follows by the same reasoning we used earlier when we argued it was Lipschitz with respect to adding or removing edges.

Now combining this with McDiarmid proves the theorem. □

The next couple of lectures, we're going to see more applications of martingales. As a reminder, the second pset is due; the third pset will be released later this evening.

§10 March 5, 2025

Today we'll continue talking about martingales. We'll add an extra twist on the theory of martingales, and look at an additional variable helpful for analyzing them, called stopping times.

§10.1 Stopping times

As a refresher of what a martingale is:

Definition 10.1. A sequence of random variables $\{Y_n\}$ is a *martingale* with respect to another sequence of random variables $\{X_n\}$ if:

- $\mathbb{E}[|Y_n|] < \infty$.
- Y_n is a function of $X_0, \ldots, X_n$.
- $\mathbb{E}[Y_n \mid X_0, \ldots, X_{n-1}] = Y_{n-1}$.

The third condition essentially says that they're unbiased.

Now we'll define a new random variable, called a *stopping time*:

Definition 10.2. We say a random variable T taking values in $\mathbb{N} \cup \{+\infty\}$ is a *stopping time* with respect to a sequence of random variables $\{X_n\}$ if for every $n \in \mathbb{N}$, the event $\{T = n\}$ is completely determined by $X_0, \ldots, X_n$.

The intuition is that I have a sequence of random variables, which you can think of as the evolution of some stochastic process in time (or information revealed over time). And the idea is that T is determined by the information we've revealed so far. It's called a stopping time because we can think of using it to stop the process.

Another way to think about this is in the analogy of gambling. Imagine you go to a casino and want to determine when you're going to leave — maybe you came in with a dollars, and you want some criterion for when to stop gambling. There are various natural criteria — for example, maybe I'll leave the moment I lose all my money, or the moment I double my money. These are valid stopping times, because you're looking at the first time you meet a certain criterion.

Example 10.3

I can stop gambling (i.e., leave the casino) at:

- The first time you double your money (or you lose all your money).
- After playing 773 rounds of blackjack.
- The third time you win.

These are all examples of stopping times, because to determine whether or not you should leave at this moment, all you have to do is look at your history of outcomes (wins or losses, or the history of how much money's in your pocket).

Example 10.4 (A non-example)

As a non-example, consider the *last* time you win. This is a non-example because you have to be able to see into the future and know you're never going to win from this time forwards in order to know you should leave.

(This is not valid because it's based on the future and not just the past.)

Student Question. *Is this a generalization of a geometric random variable?*

Answer. You could think of it that way — depending on the X_n's and your criterion for when to stop, you could make the stopping time distributed according to a geometric random variable.

One of the professors at the institution Kuikui called his PhD liked to call stopping times 'you know it when you see it.' The key thing is you can determine when to stop only based off your history.

Notation 10.5. We write $x \wedge y = \min\{x, y\}$.

Lemma 10.6

If T is a stopping time (with respect to some sequence $\{X_n\}$), then so is $T \wedge n$ for any fixed $n \in \mathbb{N}$.

This means I'm stopping either when the criterion for T is met or when I reach some fixed number of steps n.

Kind of the purpose of this lecture is to see how stopping times give us really powerful tools to analyze martingales.

Remark 10.7. There's a more appropriate level of generality to define these things (beyond just martingales) using measure theory. We're not going to do that, but if you're interested, it's in the notes.

§10.2 The optional stopping theorem

We'll now state one of the most important theorems about martingales and stopping times. We'll consider a martingale $\{Y_n\}$, and we want to study Y_T — for example, if we think of Y as how much money you have and T as the time you step out of the casino, then Y_T is the amount of money you have when you leave. There's two layers of randomness here — both in Y and in the time at which you stop.

We're going to assume some regularity conditions on these random variables. We can assume any one of the conditions here — in different situations you'll want to use different ones, but they all give the same conclusion.

Theorem 10.8 (Doob's optional stopping theorem)

Let $\{Y_n\}$ be a martingale with respect to $\{X_n\}$, and let T be a stopping time.

Assume that *any* one of the following conditions hold:

(1) The stopping time is bounded — i.e., there exists (finite) $L > 0$ such that $\mathbb{P}[T \leq L] = 1$.

(2) The martingale is bounded — there exists (finite) $B > 0$ such that $|Y_{T \wedge n}| \leq B$ with probability 1 (uniformly for all n).

(3) $\mathbb{E}[T] < \infty$, and there exists a finite $C > 0$ such that for all n, we have

$$\mathbb{E}[|Y_n - Y_{n-1}| \mid X_0, \ldots, X_{n-1}] \leq C$$

(with probability 1).

Then $\mathbb{E}[Y_T] = \mathbb{E}[Y_0]$.

The first condition requires just that your stopping time is bounded almost surely; the second requires a uniform bound on these martingale random variables. The third is slightly weaker — we just assume that the *expectation* of your stopping time is finite and that the *increments* of your martingale are bounded. The expectation in (3) is still random because it depends on $X_0, \ldots, X_{n-1}$; but I want that for any possible fixing of $X_0, \ldots, X_{n-1}$, the expected magnitude of the increment is bounded (and this is uniformly for all n).

And under any of these conditions, we get the same conclusion, which is kind of what you'd expect. Last lecture we saw that if T is some *fixed* number, then this holds — this follows just from the definition of a martingale. This theorem says that under some regularity conditions, the same is true even if the time T you look at is random.

We'll see how this can be used to analyze various stochastic processes that would otherwise be much more complicated to analyze.

§10.2.1 Proof of the optional stopping theorem

The plan is in the next 15 or so minutes, we're going to prove this theorem. Then we'll see some nice applications — to gambling and also to some algorithmic problems.

First, we should say Kuikui is going to try not to use any amount of measure theory, but this might require us to handwave a couple of the limiting arguments needed. These can be formalized using various convergence theorem (e.g., the monotone convergence theorem, the bounded convergence theorem, or the dominated convergence theorem). These details can all be found in the notes, but we'll handwave them here.

To prove this theorem, there's one key lemma, which really takes advantage of these *bounded* stopping times $T \wedge n$ (where I take my stopping time, and take the minimum with a fixed deterministic time).

Lemma 10.9

Let $\{Y_n\}$ be a martingale and T a stopping time (with respect to $\{X_n\}$). If we define a new sequence of random variables $Z_n = Y_{T \wedge n}$, then $\{Z_n\}$ is also a martingale with respect to $\{X_n\}$.

This directly gives at least the first case of the optional stopping theorem (we'll talk about that in more detail in a moment); this is sort of the key lemma.

Proof. We have these random variables Z_n, and we want to show they satisfy the key unbiasedness property of martingales. To do that, I'm going to first decompose Z_n into Y_n's, because we know those are a martingale.

So we can observe that we can decompose Z_n into Y_n's by inserting indicator random variables for the value of the stopping time — we can write

$$Z_n = Y_n \cdot \mathbf{1}_{T \geq n} + \sum_{k=0}^{n-1} Y_k \mathbf{1}_{T=k}.$$

This is a very similar trick to what we did when we e.g. proved Paley–Zygmund during the second moment method — when you want to analyze a random variable, you can artifically insert indicator random variables to decompose the original random variable you were analyzing.

So this is one way of writing Z_n. Now because T is a stopping time (and $\{Y_n\}$ is a martingale), we know that the random variables $\mathbf{1}_{T=0}$, $\dots$, $\mathbf{1}_{T=n-1}$, and also Y_0, $\dots$, Y_{n-1} — and even the random variable $\mathbf{1}_{T \geq 1} = 1 - \mathbf{1}_{T \leq n-1}$ — are all completely determined by X_0, $\dots$, X_{n-1}. What this means is once I fix the values of X_0, $\dots$, X_{n-1}, this fixes the values for all these random variables. The only one that is *not* fixed when I tell you the values of X_0, $\dots$, X_{n-1} is Y_n. And this is going to be crucial for us.

So now let's look at

$$\mathbb{E}[Z_n \mid X_0, \dots, X_{n-1}].$$

I need to show that this is equal to Z_{n-1} (in order for $\{Z_n\}$ to be a martingale).

Using the decomposition and the observation that all these random variables are determined once I fix X_0, $\dots$, X_{n-1}, by linearity of expectation I can write this as

$$\sum_{k=0}^{n-1} Y_k \mathbf{1}_{T=k} + \mathbb{E}[Y_n \mid X_0, \dots, X_{n-1}] \mathbf{1}_{T \geq n}.$$

(This just directly uses linearity of expectation plus the fact that once I fix X_0, $\dots$, X_{n-1}, all of Y_k, $\mathbf{1}_{T=k}$, and $\mathbf{1}_{T \geq n}$ are fixed quantities — so the expectation is only acting on Y_n.)

And now I finally want to use the fact that $\{Y_n\}$ is a martingale — that's the only assumption we haven't used yet. So this is equal to

$$\sum_{k=0}^{n-1} Y_k \mathbf{1}_{T=k} + Y_{n-1} \mathbf{1}_{T \geq n}$$

(using the martingale assumption, which says $\mathbb{E}[Y_n \mid X_0, \dots, X_{n-1}] = Y_{n-1}$). And now we're essentially done — because in the sum I also have a Y_{n-1} term, where $k = n-1$. And I can combine it with this one, so I get

$$\sum_{k=0}^{n-2} Y_k \mathbf{1}_{T=k} + Y_{n-1} \mathbf{1}_{T=n-1} + Y_{n-1} \mathbf{1}_{T \geq n} = \sum_{k=0}^{n-2} Y_k \mathbf{1}_{T=k} + Y_{n-1} \mathbf{1}_{T \geq n-1}.$$

And again using the decomposition from above, this is exactly equal to Z_{n-1}. □

> **Student Question.** *You said at the beginning that all the random variables were completely determined by X_0, $\dots$, X_{n-1}. Is that a rephrasing of the condition $\mathbb{E}[Y_n \mid X_0, \dots, X_{n-1}] = Y_{n-1}$?*
>
> **Answer.** It's the second statement — that Y_n is a function of X_0, $\dots$, X_n (and similarly $\{T = n-1\}$ is determined by X_0, $\dots$, X_{n-1} from the definition of a stopping time).

Now we want to use this to prove the optional stopping theorem. We're going to handwave some of the arguments; but we'll at least be able to do the first condition directly.

Proof of optional stopping theorem. Recall that condition (1) says that $\mathbb{P}[T \leq L] = 1$ (so T is almost surely uniformly bounded). Now using the key lemma, we know that $\{Z_n\} = \{Y_{T \wedge n}\}$ is a martingale as well (for

any n). So in particular, from what we showed in the previous lecture (i.e., the unbiasedness of martingales), we have

$$\mathbb{E}[Z_n] = \mathbb{E}[Z_0]$$

(just because the Z_n's form a martingale).

Now, what is Z_0? This is the minimum of our stopping time with 0, so it's just Y_0.

And now, because I know $T \leq L$ with probability 1, if I set $n = L$, then Z_L is just Y_T (because you're taking the minimum of T and a number that's definitely larger than T). This implies

$$\mathbb{E}[Y_T] = \mathbb{E}[Z_L] = \mathbb{E}[Z_0] = \mathbb{E}[Y_0].$$

Now what about the other conditions? For conditions (2) and (3), we're going to use condition (1) for the stopping times $T \wedge n$. We know from condition (1) that

$$\mathbb{E}[Y_{T\wedge n}] = \mathbb{E}[Y_0]$$

(just because $T \wedge n$ is always bounded by n); and this holds for all n.

We also know that if I look at the sequence of stopping times $T \wedge n$, this is 'converging' to T as $n \to \infty$ (whatever this means). And we want to deduce that $\mathbb{E}[Y_{T\wedge n}] \to \mathbb{E}[Y_T]$. In order to make these limits actually legit, you need to take advantage of these regularity assumptions (the conditions (2) and (3)). And you can use various theorems in measure theory (like the dominated convergence theorem or monotone convergence theorem) to make these precise. But we won't go through the details, because it's not conceptually important for what we're going to do with stopping times. (If you're interested in this kind of thing, it's in the lecture notes.) So really in the proof of this thing, there's kind of one key case, which is the case where the stopping time is bounded by some finite constant (almost surely). □

§10.3 Random walk

Now we're going to use this to analyze some martingales. We'll use this to think about gambling; or another way to think about it is as a random walk on the integers.

Example 10.10

Let $\{X_n\}$ be a sequence of i.i.d. $\mathsf{Unif}\{\pm 1\}$ random variables, and let $S_n = \sum_{k=1}^n X_k$.

So the picture here is I have the number line $\mathbb{Z}$, and I start from 0. Thn at each step, I pick one side to jump to. Maybe on the first step I jump to $+1$, then I jump back to 0, then -1, then -2, then -3, then back to -2, and so on. So this is a random walk on the integers where I increment or decrement by 1.

You can also think of this as a very basic model for gambling — imagine you start with some amount of money (think of 0 as having net 0 gain or loss); and at each round of playing, you either win a dollar or lose a dollar (with equal probability).

We'll be interested in the following stopping times:

Notation 10.11. For $x \in \mathbb{Z}$, let $T_x = \min\{n \mid S_n = x\}$.

For example, if we start with A dollars, then T_{-A} is the first time you lose all your money; and T_B is the first time you gain B dollars (if $B > 0$). So these are natural stopping times you might care about.

We'll consider the following questions — imagine you're going to stay at the casino until you either lose all your money or gain B dollars.

Question 10.12. What's the probability that $\mathbb{P}[T_{-A} < T_B]$ (i.e., that you lose all your money before you gain B dollars)?

Question 10.13. What's $\mathbb{E}[\min\{T_{-A}, T_B\}]$?

Lemma 10.14

We have $\mathbb{P}[T_{-A} < T_B] = \frac{B}{A+B}$ and $\mathbb{E}[\min\{T_{-A}, T_B\}] = A \cdot B$.

The first expression makes at least some intuitive sense. Qualitatively, if I set my target B to be extremely large, what you'd expect is it's much more likely for you to lose A dollars than to win B extra dollars (if $B \gg A$). And this exactly captures this.

And $A \cdot B$ also makes some kind of intuitive sense. If you imagine renormalizing things so that $A = N$ and $B = N$, so you're just doing a simple random walk on the integers and you're stopping the moment you hit $-N$ or N; then if you look at the distribution of where the simple random walk is, the distribution of the kth step of this random walk should look like a binomial with k trials and success probability $\frac{1}{2}$ (shifted down so it's centered at 0). And we know from the concentration of the binomial distribution that after k steps, you're going to expect to be at distance roughly $\sqrt{k}$ from 0. So if you want to hit N with high probability, you want to look at N^2 trials. And that's roughly what's captured here.

So these expressions for this lemma sort of match what you might qualitatively expect.

Proof. We're going to prove these using stopping times. For convenience, let $T = \min\{T_{-A}, T_B\}$. We know by definition that

$$|S_{T \wedge n}| \leq \max\{A, B\}$$

with probability 1 (for all n). So I can use the optional stopping theorem (the second condition is satisfied).

We have $\mathbb{E}[S_0] = 0$ (since I start at net 0 dollars); so this implies

$$0 = \mathbb{E}[S_0] = \mathbb{E}[S_T].$$

And what happens when I hit T? Either I've gained B dollars or I've lost A; so we have

$$\mathbb{E}[S_T] = \mathbb{P}[T = T_B] \cdot B - \mathbb{P}[T = T_{-A}] \cdot A.$$

(By the definition of the stopping time, when I hit this stopping time T the amount of money I've lost or gained is either that I've gained B dollars or I've lost A dollars.)

And now you can write these probabilities in terms of $\mathbb{P}[T_{-A} < T_B]$ — note that

$$\mathbb{P}[T = T_{-A}] = \mathbb{P}[T_{-A} < T_B].$$

Then if you rearrange things, this will give you the first claim.

What about the second one? Now let's look at the following sequence of random variables — we'll look at

$$Y_n = S_n^2 - n.$$

We'll see in a moment why you'd want to do this. But basically then we'll have

$$Y_T = S_T^2 - T,$$

and when I take an expectation, I'll get $\mathbb{E}[T]$.

This class of random variables, where you take the square of a martingale and normalize by the variance, is also a martingale, called a *second moment martingale*:

Claim 10.15 — $\{Y_n\}$ is also a martingale with respect to $\{X_n\}$.

This is a fairly direct calculation, so we're not going to go through it.

Also, by the same thing as above, we know that

$$|Y_{T\wedge n}| \le \max\{A^2 - A, B^2 - B\}$$

with probability 1, for all n (because the first time I hit the stopping time T, I'm either going to be at A or B, and all throughout in between, S_n is going to be between $-A$ and B).

So in particular, once again we can apply the optional stopping theorem, because it satisfies at least the second condition.

Student Question. *Why do we need the 'with probability 1' caveat here? Isn't everything finite here, so the events have positive probability?*

Answer. This is a random walk on the full integers, so I'm not assuming the S_n's are bounded with probability 1. But once I hit the stopping time, by the definition of the stopping time, I know I'm definitely in a bounded region.

But yes, here the bounds do hold always, not just with probability 1 — there are no weird measure theory edge cases here (there's no event in which it exceeds these bounds).

So we know this is a martingale and satisfies the conditions of the optional stopping theorem, so we can use it and get

$$0 = \mathbb{E}[Y_0] = \mathbb{E}[Y_T] = \mathbb{E}[S_T^2] - \mathbb{E}[T].$$

By rearranging, this implies that

$$\mathbb{E}[T] = \mathbb{E}[S_T^2].$$

And once again, when I hit the stopping time T, then S_T is going to be either $-A$ or B; so this is

$$\mathbb{E}[T] = \mathbb{P}[T = T_{-A}] \cdot A^2 + \mathbb{P}[T = T_B] \cdot B^2.$$

And now we can use the calculation we saw earlier; so this is

$$\frac{B}{A+B} \cdot A^2 + \frac{A}{A+B} \cdot B^2 = A \cdot B. \qquad \square$$

So already just by using this theory of stopping times, you can give a fairly short and quick analysis of at least a basic model for gambling, where you win or lose a dollar with probability $\frac{1}{2}$.

Remark 10.16. If you're interested in the setting where there's a bias (a slightly higher chance you lose money than win money), that can also be analyzed using similar techniques; the calculations are slightly more complicated, but they're in the notes.

§10.4 Supermartingales and submartingales

One thing that's slightly dissatisfactory is that you could also have proved this lemma using more direct calculations, because this sequence of random variables S_n also forms what's called a *Markov chain* — so there's also a recursive or inductive way to prove these identities. But now we'll turn to some examples where the theory of optional stopping and martingales is really powerful in analyzing processes that don't have this Markovian structure.

To do this, we'll extend the theory of martingales just slightly — we'll define a related class of stochastic processes that are basically martingales, but without the exact unbiasedness condiiton.

Definition 10.17. We say a sequence of random variables $\{Y_n\}$ is a *supermartingale* (respectively, *submartingale*) with respect to another sequence $\{X_n\}$ if:

- $\mathbb{E}[|Y_n|] < \infty$ for all n.
- Y_n is a deterministic function of $X_0, \ldots, X_n$.
- $\mathbb{E}[Y_{n+1} \mid X_0, \ldots, X_n] \leq Y_n$ for supermartingales, or $\mathbb{E}[Y_{n+1} \mid X_0, \ldots, X_n] \geq Y_n$ for submartingales.

So these are the same conditions, except that we replace the unbiasedness condition with an inequality.

Remark 10.18. It's perhaps slightly confusing terminology that for supermartingales, we expect them to *decay*; and for submartingales, we expect them to *grow* (at least, compared with a standard martingale).

We'll see a lemma, which will connect the standard theory of martingales with convexity.

Lemma 10.19

Let $\{Y_n\}$ be a martingale with respect to $\{X_n\}$. If $f : \mathbb{R} \to \mathbb{R}$ is a convex function (respectively, concave), then the new sequence $\{f(Y_n)\}$ is a submartingale (respectively, supermartingale).

So from a martingale, I can produce many natural submartingales or supermartingales, just by applying a function which is convex or concave.

And there are analogous tail bounds to Azuma–Hoeffding for submartingales and supermartingales (though you have to be slightly careful, because you only get a one-sided rather than two-sided bound).

Student Question. *What boundedness conditions on f are sufficient for this lemma?*

Answer. The proof is by using Jensen's inequality. So you need $\mathbb{E}[|f(Y_n)|] < \infty$, but aside from this, there shouldn't be anything else.

And for optional stopping, if you open up the proof for standard martingales, we also get that

$$\mathbb{E}[Y_T] \leq \mathbb{E}[Y_0]$$

for supermartingales, and analogously

$$\mathbb{E}[Y_T] \geq \mathbb{E}[Y_0]$$

for submartingales.

§10.5 A local search algorithm for 2SAT

Now we want to use the notion of a submartingale and supermartingale to analyze a nice algorithm for solving the computational problem 2SAT.

Problem 10.20 (2SAT)

- **Given:** A Boolean formula in conjunctive normal form, where every clause has at most 2 variables.
- **Output:** A satisfying assignment (if one exists).

Recall that your formula looks like an AND of a bunch of ORs. There are nice deterministic formulas for this, but we'll use this machinery to analyze a very nice and simple *stochastic local search* algorithm for solving 2SAT.

Algorithm 10.21 (Stochastic local search for 2SAT)

Suppose we're given a 2-CNF formula Φ.

(1) Initialize with an arbitrary assignment $x_0 : \{\text{variables}\} \to \{\mathsf{T}, \mathsf{F}\}$.

(2) While x_t doesn't satisfy Φ, we do the following:

- Pick an arbitrary unsatisfied clause C.
- Pick a uniformly random variable $v \in C$.
- Resample the assignment for this variable — so we set $x_{t+1}(u) = x_t(u)$ for all $u \neq v$, and

$$x_{t+1}(v) \sim \mathsf{Unif}\{\mathsf{T}, \mathsf{F}\}.$$

Recall that because we're looking at 2SAT, the clause C involves at most two variables.

Theorem 10.22

For any satisfiable 2-CNF formula Φ, the above algorithm terminates in $O(n^2)$ rounds in expectation.

(If you want to boost 'in expectation' to 'with high probability,' you can use Markov.)

And each round of this algorithm takes only $O(m)$ time, where m is the number of clauses. So in total, the algorithm takes $O(n^2 m)$ time for n variables and m clauses.

This algorithm is very related to an algorithm we briefly mentioned when discussing the Lovász local lemma. We used the LLL to show satisfiability for a large class of very natural k-CNF formulas. And it turns out that result is *constructive* in a way — there is a simple local search algorithm that actually finds a satisfying assignment with high probability. This is sort of very related to that.

Before we prove this theorem, we'll state the key technical lemma we're going to use, which is going to be about submartingales or supermartingales. The high-level approach to this theorem is we'll build a supermartingale (or maybe submartingale) that sort of tracks the amount of progress our algorithm is making towards finding a satisfying assignment. But before we do that, we'll state and prove the technical lemma that's needed. (This lemma will also be useful for your homework.)

Lemma 10.23

Let $\{Y_t\}$ be a sequence of random variables taking values in $\{0, \dots, n\}$, and assume that $\{Y_t\}$ forms a supermartingale with respect to some sequence $\{X_t\}$ and that

$$\mathbb{E}[(Y_t - Y_{t-1})^2 \mid X_0, \dots, X_{t-1}] \geq \sigma^2$$

with probability 1 for all t. Then if T denotes the stopping time for when Y_T hits 0, then we have

$$\mathbb{E}[T] \leq \frac{n^2}{\sigma^2}.$$

You should think of the condition as being a lower bound on the variance of the increments. Here n^2 is the size of the domain we're looking at, and we divide by this variance term.

What does this mean? The picture you should have is I have my number line 0 through n, and I have a whole bunch of possible values that my Y_t's can take. And let's suppose Y_0 is in the middle (it doesn't really matter). The way to interpret the supermartingale condition is basically that there's some slight drift towards 0 (the drift might be 0, if the Y_t's are an actual martingale; but we just don't want it to drift towards n). So there's a bias towards getting smaller and smaller, which is good for us. But of course the

martingale that just sits at Y_0 forever is also a valid martingale. So there has to actually be some variance telling you that you actually make a move as time progresses — the variance lower bound means that you actually move. So basically these two conditions say informally that if you're more likely to move towards 0 and you're reasonably likely to move at all, then within a reasonable amount of time you will hit 0.

Student Question. *How do I know a supermartingale would move towards* 0 *and not towards* n*?*

Answer. In the definition, you see that if I'm currently at Y_n, then I'm 'more likely' to decrease (to be smaller than Y_n) in the next step, at least in expectation.

Before we prove this technical lemma, we'll use it to go back to proving this theorem, that the stochastic local search algorithm actually finds a satisfying assignment (assuming one exists).

Proof of Theorem 10.22. We assumed our formula has some satisfying assignment — we don't know what it is, but such a thing exists. So let x^* be any satisfying assignment.

Now let's consider the potential function

$$f(x) = d_H(x, x^*)$$

(the Hamming distance to the satisfying assignment — i.e., the number of coordinates where they differ). Now let $Y_t = f(x_t)$. Recall that x_t is the current state of the algorithm — the Boolean assignment that my local search algorith mis currently at. This certainly takes values in $\{0, \dots, n\}$ — if I'm at 0 then I'm at a satisfying assignment (specifically, x^*), and n is the number of variables.

Now let's look at what happens with this stochastic process.

Claim 10.24 — $\{Y_t\}$ is a supermartingale with respect to $\{x_t\}$.

Proof. Think about what the algorithm does at each step. Suppose I'm at x_t, and it isn't already a satisfying assignment. Then it must differ from x^* at some variable. More specifically, if x_t doesn't satisfy a specific clause C — recall that the algorithm was I picked some unsatisfied clause C and randomized a uniformly random variable in it — then for at least one of the variables $v \in C$, we know that $x^*(v) \neq x_t(v)$. (This is because x^* is a satisfying assignment, so x^* satisfies C, while x_t does not.)

Now what happens in the algorithm? I pick a uniformly random variable in C, and I randomize the assignment for that variable. And from how the algorithm works, one can show that

$$\mathbb{P}[Y_t - Y_{t-1} = -1 \mid x_0, \dots, x_{t-1}] \geq \mathbb{P}[Y_t - Y_{t-1} = +1 \mid x_0, \dots, x_{t-1}].$$

So I make a step that gets closer to x^*. And this implies the claim. □

So now we have a supermartingale, and we just need to lower bound the variance. But this is also easy — each time we're picking a uniformly random variable in an unsatisfied clause and rerandomizing it. This means we immediately get $\sigma^2 \geq \Omega(1)$.

Now we can apply the lemma; and we get

$$\mathbb{E}[\text{time to hit } d_H(x_t, x^*) = 0] \leq O(n^2).$$

So actually, in n^2 time you'll hit x^* in expectation (unless you hit some other satisfying assignment first). □

Unfortunately we're out of time, so we'll wrap up and say in 30 seconds roughly how you prove the key lemma. The proof of the key lemma is basically a slightly fancier version of the analysis we did for gambling, when we analyzed the expected stopping time. You define a new sequence of random variables which look

like quadratics (like the second moment martingale from earlier); but there'll be some parameters set, and it won't be a bona fide martingale but rather a submartingale. And you use the optional stopping theorem in the same way as before to study this submartingale, and use that to deduce a bound for $\mathbb{E}[T]$. (We don't have time to go through the proof, but it's in the notes, and the ideas are just slightly fancier versions of the ideas we saw earlier when we analyzed the simple random walk.)

§11 March 10, 2025 — Euclidean traveling salesperson problem

Today we'll discuss a very neat application of the martingale techniques we've been discussing. We'll use it to discuss one of the quintessential optimization algorithms that occurs all over operations research, the traveling salesperson problem. We'll look specifically at the Euclidean case.

§11.1 The Euclidean TSP problem

Problem 11.1 (Euclidean TSP)

- **Given:** n points $p_1, \dots, p_n \in \mathbb{R}^d$.
- **Goal:** Find a tour of the points, i.e., a sequence $p^{(1)}, \dots, p^{(m)}$ where we visit all the points, minimizing

$$\mathsf{Cost} = \sum_{j=1}^{m-1} \|p^{(j)} - p^{(j+1)}\|_2.$$

So I want to visit all the points at least once, and I want to minimize the total distance travelled.

As a picture, we have a whole bunch of points; and a tour might look something like this:

We allowed a tour to visit a point more than once. We can observe that the optimal tour will visit each point *only* once by the triangle inequality; but it'll be more convenient for us to allow tours that revisit a point.

This is a very classical problem you'll see in an undergraduate CS course, where you'll design approximation algorithms for worst-case versions of this problem.

But here we want to use martingales to study *average-case* versions — we'll look at the case where the points are drawn randomly.

Theorem 11.2 (Beardwood–Halton–Hammersley 1959)

Suppose $p_1, \dots, p_n \sim \mathsf{Unif}[0,1]^d$ are i.i.d. Then there exists a constant $\beta(d) > 0$ such that

$$\frac{\mathsf{OPT}}{n^{1-1/d}} \to \beta(d).$$

(We write OPT for the cost of the optimal tour.)

You can generalize to other distributions, but for simplicity we'll focus on this one.

In other words, the scaling for the optimal tour is roughly of the order of $n^{1-1/d}$. We're not going to be super precise about what this convergence means; you can imagine saying for every ε, the probability the LHS deviates from $\beta(d)$ by ε goes to 0 as $n \to \infty$.

In fact, it's even known what the right constant $\beta(d)$ is, in some sense: Rhee (1992) showed that

$$\lim_{d\to\infty} \frac{\beta(d)}{\sqrt{d}} = \frac{1}{\sqrt{2\pi e}}.$$

We're not going to prove this theorem exactly; we'll show something weaker — we'll show that $\mathbb{E}[\mathsf{OPT}]$ is of the order $n^{1-1/d}$, and we'll show sharp concentration bounds for this random variable. Specifically:

Theorem 11.3

We have $\mathbb{E}[\mathsf{OPT}] = \Theta_d(n^{1-1/d})$.

Theorem 11.4 (Rhee–Talagrand 1987)

For all n and d, there is a constant $C(n,d)$ such that

$$\mathbb{P}[|\mathsf{OPT} - \mathbb{E}[\mathsf{OPT}]| \geq t] \leq 2\exp\left(-\frac{t^2}{C(n,d)}\right)$$

for all $t \geq 0$. Furthermore, we have

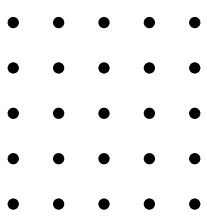
$$C(n,d) = \begin{cases} O(\log n) & \text{if } d = 2 \\ O_d(n^{1-2/d}) & \text{if } d > 2. \end{cases}$$

This says we have sub-Gaussian tails. And when $d = 2$, for example, the expectation is on the order of $\sqrt{n}$; but the deviations are only on the order of $\sqrt{\log n}$, which is *much* sharper.

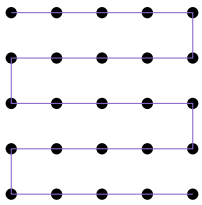
Here we're thinking of d as some fixed dimension (like 2 or 3 or 10), and the number of points n as going to ∞.

§11.2 Estimating the expectation

First, why do we have $n^{1-1/d}$ scaling for the expectation? One way to intuit this is imagine rather than having randomly placed points, you had a very nice and evenly spaced sequence of points — so we have $\sqrt{n}$ rows each with $\sqrt{n}$ points (in the case $d = 2$).



If your points look like this, there's a very natural tour, which traverses each row in a snakelike fashion. And if you compute the total distance travelled, it's $O(\sqrt{n})$ — for each row you pay a cost of at most 1 (the side length of the square), and there are $\sqrt{n}$ rows.



You can generalize this to higher dimensions — if your points look like evenly spaced grid points, you'll have order $n^{1-1/d}$ cost.

And Theorem 11.3 says that if we pick random points, then they might not look like evenly spaced grid points, but the cost will actually be on the same order as if they were evenly spaced grid points. So that's how one might at least guess that the expectation scales this way.

Now let's prove Theorem 11.3. We'll need one key proposition, which will also play a role when we prove the concentration inequality.

Proposition 11.5

Let $p \in [0,1]^d$ be an arbitrary point. Then we have

$$\mathbb{E}[\min_i \operatorname{dist}(p, p_i)] \asymp n^{-1/d}.$$

So if I take some arbitrary point p and look at the closest point p_i (where the p_i's are drawn uniformly and independently from the cube $[0,1]^d$), then this is on the order of $n^{-1/d}$.

So the picture is I pick some arbitrary point p and start plopping down random points in the square; and the claim is that the closest point to p has distance of order $n^{-1/d}$. This kind of makes sense — the more points you plop down, the closer you'd expect the closest point to be.

We'll prove this proposition in a moment, but we'll first use this to prove Theorem 11.3. We have to establish two inequalities — an upper bound and a lower bound, both of order $n^{1-1/d}$.

§11.2.1 The lower bound

The key observation behind the lower bound is that thinking about the constraints on our tour, one key constraint is that we must visit every point at least once. In particular, the cost of any tour is always lower-bounded by

$$\mathsf{OPT} \geq \frac{1}{2}\sum_{i=1}^{n} \operatorname{dist}(p_j, \mathcal{P} \setminus \{p_j\})$$

(where we write $\mathcal{P} = \{p_1, \dots, p_n\}$, and we write $\operatorname{dist}(p, S) = \inf_{q \in S} \|p - q\|_2$).

(The $\frac{1}{2}$ is just there because we have a path and not a cycle; it's not important.)

Now if I take expectations on both sides, I can use linearity of expectation to say

$$\mathbb{E}[\mathsf{OPT}] \geq \frac{1}{2}\sum_{j=1}^{n} \mathbb{E}[\operatorname{dist}(p_j, \mathcal{P} \setminus \{p_j\})].$$

And each of these terms is lower-bounded by $n^{-1/d}$ by Proposition 5.10, and there's n of them; so we get a lower bound of $n^{1-1/d}$.

§11.2.2 The upper bound

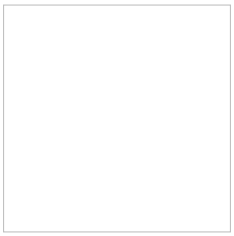
For the upper bound, we'll prove something stronger — we'll actually show this holds not just in expectation, but with probability 1. We'll show that no matter how you place your points, there'll be some tour with this cost.

> **Claim 11.6 —** For every set of points, there exists a tour of cost $O(n^{1-1/d})$.

So it doesn't just hold in expectation; it holds for every single set of points.

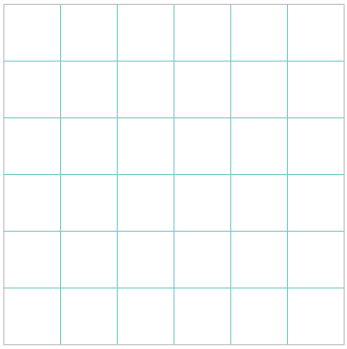
The idea is really going to come back to the picture in the nicely evenly spaced case.

The proof is essentially going to be by picture. Imagine we take our box, and we plop down some points (these points can be arbitrary, so we can have some weird clustering in the top-right and a void with no points in the bottom-right).

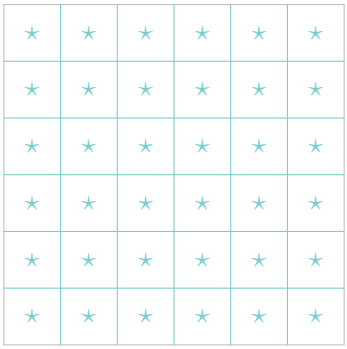


(We're imagining $d = 2$; this illustrates all the ideas for higher dimensions.)

Now imagine I cut this box into a bunch of smaller boxes — imagine that we have $\sqrt{n}$ boxes in each row, and all of them have side length $1/\sqrt{n}$. So in total there's n boxes, each of which has side length $1/\sqrt{n}$. (In higher dimensions, you cut into boxes of side length $n^{-1/d}$; so you maintain that you have n boxes in total.)

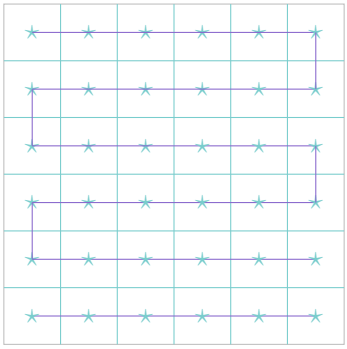


Now we just need to construct *some* tour. So what we'll do is place a new point (which we denote by a $\star$) at the center of each of these boxes. So I'm just adding *more* points to my input. And what I'm going to do is construct a tour that goes through all $2n$ points; then you can use the triangle inequality to get a tour for the original points. So I'm going to construct a tour on all the dots and stars, and show its cost is at most $O(n^{1-1/d})$.



To construct this tour, we'll essentially use the snakelike picture above. That picture gives a tour between stars; so now I just need to construct a tour within the boxes. So I construct an arbitrary tour within the boxes, and jump between stars. So for our tour, we alternate between:

- An arbitrary tour within each subcube.
- Moving between the stars (i.e., the centers of the subcubes), according to the snakelike tour between regular grid-points.



Claim 11.7 — This tour has cost $O(n^{1-1/d})$.

Proof. There are two types of terms that contribute to the cost — transitions when I'm moving between points inside a subcube, and transitions when I'm moving between two subcubes. So let the subcubes be $\mathcal{C}_1, \dots, \mathcal{C}_n$, with centers $q_1, \dots, q_n$, and let $k_i = |\mathcal{P} \cap \mathcal{C}_i|$ be the number of points in the ith subcube. Then

$$\mathsf{Cost} = \sum_{i=1}^{n} \mathsf{Cost}(\text{tour inside } \mathcal{C}_i) + \sum_{i=1}^{n} \|q_i - q_{i+1}\|_2$$

(where the second term is from moving between the cubes following the snakelike tour).

The first term $\mathsf{Cost}(\text{tour inside } \mathcal{C}_i)$ is a sum of k_i terms, and each of those terms is bounded by the diameter of the subcube, which is something like $\sqrt{d} \cdot n^{-1/d}$. And the cost of the snakelike tour is also of order $n^{1-1/d}$. So we get

$$\mathsf{Cost} \lesssim \sqrt{d} \cdot n \cdot n^{-1/d} + n^{1-1/d} \lesssim n^{1-1/d}$$

(using the fact that $\sum k_i$ is the total number of points, which is n). □

§11.2.3 Proof of Proposition 11.5

So up to the key proposition, we've shown that the expected optimal tour cost is of order $n^{1-1/d}$. Now let's prove this proposition, which we'll also use when we look at concentration.

Recall that we're writing $\mathcal{P} = \{p_1, \dots, p_n\}$, and $\operatorname{dist}(p, \mathcal{P}) = \inf_{q \in \mathcal{P}} \|p - q\|_2$. Our goal is to bound $\mathbb{E}[\operatorname{dist}(p, \mathcal{P})]$, where $\mathcal{P}$ consists of independent and uniformly generated points in the cube.

How are we going to control this? We'll use the standard trick where we can always convert an expectation into integrating a *tail* of a random variable — this is equal to

$$\mathbb{E}[\operatorname{dist}(p, \mathcal{P})] = \int_0^{\sqrt{d}} \mathbb{P}[\operatorname{dist}(p, \mathcal{P}) \geq R] \, dR$$

(we cap the integral at $\sqrt{d}$ because that's the diameter of the space). The points in $\mathcal{P}$ are generated independently of each other, so we can simplify this — it becomes

$$\int_0^{\sqrt{d}} \mathbb{P}_{q \sim \mathsf{Unif}[0,1]^d}[\|p - q\|_2 \geq R]^n \, dR$$

(since the event $\operatorname{dist}(p, \mathcal{P}) \geq R$ holds if and only if every single point $q \in \mathcal{P}$ is far from our reference point p). So now we just need to estimate $\mathbb{P}[\|p - q\|_2 \geq R]$ (the thing inside the exponent of n).

Now let's draw another picture; I want to look at the probability

$$\mathbb{P}_{q \sim \mathsf{Unif}[0,1]^d}[\|p - q\|_2 \geq R].$$

So the picture is I have some point p, and I'm going to draw a ball of radius R. This event occurs if and only if my point q lands outside this ball. So to understand this probability, I really just have to estimate the volume of the cube intersected with this ball; and that's going to give me control on this quantity.

So this thing is equal to

$$1 - \operatorname{Vol}(\mathbb{B}_2(p, R) \cap [0,1]^d),$$

where $\mathbb{B}_2(p, R) = \{x \mid \|x - p\|_2 \leq R\}$.

Now what do we expect? If I take a ball in dimension d and scale up the radius, I expect the volume to scale as R^d. So this volume is of the order R^d (up to constants that depend on d).

(You can imagine you have a slightly pathological case where p is really close to a corner; but that's okay, because it's still growing at the same rate as we increase the radius R.) So we have

$$\mathbb{P}_q[\|p - q\|_2 \geq R] \asymp R^d$$

(this means it's lower-bounded and upper-bounded by $c(d)R^d$ and $C(d)R^d$, for all $0 \leq R \leq \sqrt{d}$; certainly if I go beyond $\sqrt{d}$ the volume is no longer going to grow).

Based on this, now let's sort of control this integral. For the lower bound, one thing to note is that if I view this tail as a function of R, it's a decreasing function. So if I want to lower bound this integral, I can just choose some threshold for R, and throw away a whole bunch of terms from this integral. In particular, this integral is always lower-bounded by

$$\int_0^{R_0} \mathbb{P}_q[\|p - q\|_2 \geq R_0]^n \, dR.$$

And this function doesn't depend on R (I just fixed some constant R_0), soit's really lower bounded by

$$R_0 \cdot \mathbb{P}_q[\|p - q\|_2 \geq R_0]^n,$$

which is lower-bounded by

$$R_0(1 - c(d) \cdot R_0^d)^n.$$

Now we want to take R_0 satisfying $c(d) \cdot R_0^d = \Theta(1/n)$, so that this exponential becomes constant. If I rearrange, then $R_0 \asymp n^{-1/d}$. And that's it for the lower bound.

The upper bound is actually quite similar, though the calculation is slightly more complicated. Now you can't just throw away terms in this integral. But essentially to get a bound, we're also going to use some threshold R_0 in the following way. Let's again set

$$R_0 = \left(\frac{1}{c(d) \cdot n}\right)^{1/d}.$$

And we'll also let

$$T = \left\lceil \frac{\sqrt{d}}{R_0} \right\rceil.$$

Basically, we're going to break up this integral into a bunch of scales dpeending on R_0 — we have

$$\mathbb{E}[\mathrm{dist}(p, \mathcal{P})] \le \sum_{t=0}^{T-1} \int_{tR_0}^{(t+1)R_0} \left(1 - c(d) \cdot R^d\right)^n \, dR.$$

Now this thing $(1 - c(d)R^d)^n$ is a decreasing function of R, so I can upper-bound it by plugging in the lower endpoint of the interval; so this is at most

$$R_0 \cdot \sum_{t=0}^{T-1} (1 - c(d) \cdot t^d R_0^d)^n.$$

Now using our favorite identity $1 - x \le e^{-x}$, we can replace this by

$$R_0 \cdot \sum_{t=0}^{T-1} \exp(-c(d) \cdot t^d R_0^d n).$$

And now this is very nice because our choice of R_0 is going to cancel out the n and $c(d)$; so we get something upper-bounded by

$$R_0 \sum_{t=0}^{\infty} e^{-t^d}.$$

And this sum is an absolute constant, so that's it; and because $R_0 \asymp n^{-1/d}$, we get a bound of $n^{-1/d}$.

So really all the action behind this bound is by looking at this radius $R_0 \asymp n^{-1/d}$.

§11.3 Concentration — a first attempt

This completes the proof of Theorem 11.3; now we want to discuss concentration around the expectation.

We'll view OPT as being a function of the n input points $p_1, \dots, p_n$. And we can try to use some general-purpose concentration inequalities like McDiarmid. In order to use McDiarmid, I need to verify that this function is Lipschitz in the coordinates.

Claim 11.8 — We have $|\mathsf{OPT}(p_1, \dots, p_k, \dots, p_n) - \mathsf{OPT}(p_1, \dots, p_k', \dots, p_n)| \le 2\sqrt{d}$.

This is clear — if I remove the point p_k and add a new point p_k', then I can change the tour just by adding two edges to p_k' and back, each of which has cost at most $\sqrt{d}$ (the $\sqrt{d}$ comes from the diameter of the space).

Student Question. *Isn't it better to choose the closest point to detour to p_k' from, so we could get $n^{-1/d}$ instead?*

Answer. If I want to use McDiarmid, I need a worst case statement — it could very well be that all my points were in one corner, and I added one really bad point in the opposite corner.

So if we use McDiarmid, we get a concentration inequality

$$\mathbb{P}[|\mathsf{OPT} - \mathbb{E}[\mathsf{OPT}]| \geq t] \leq 2 \cdot \exp\left(-\frac{t^2}{8dn}\right).$$

In particular, McDiarmid already tells you that you're expecting deviations of the order of $\sqrt{n}$. This is actually already kind of close if you think about what $C(n,d) = O_d(n^{1-2/d})$ gives you for large d. But this is way off if you look at low dimensions, e.g., $d = 2$ — there, according to Theorem 11.4, you only expect the deviations to be of constant order (or really $\sqrt{\log n}$), which is much much smaller than the expectation.

So this is good for large d, but way off for small d.

What we're going to do, in order to get a much sharper concentration inequality, is to open up the black box of McDiarmid. For that, we used a martingale concentration inequality (Azuma–Hoeffding), which requires bounded increments. And we're instead going to use a non-uniform choice of bounds for those increments; bounding those increments is going to use Proposition 11.5.

§11.4 A more refined bound

For convenience, let $\mathcal{P}_{\leq k} = \{p_1, \dots, p_k\}$ and $\mathcal{P}_{>k} = \mathcal{P} \setminus \mathcal{P}_k$, and $\mathcal{P}_{-k} = \mathcal{P} \setminus \{p_k\}$. We'll look at the Doob martingale

$$Y_k = \mathbb{E}[\mathsf{OPT} \mid \mathcal{P}_{\leq k}]$$

(the expectation of OPT conditioned on the first k points). This is the exact martingale we'd use to prove McDiarmid; but now we want to open up the black box of McDiarmid and use something stronger to prove concentration.

Proposition 11.9

For all k, we (almost surely) have

$$|Y_k - Y_{k-1}| \leq \min\left\{2\sqrt{d}, \frac{O_d(1)}{(n-k)^{1/d}}\right\}.$$

So we get a bound for the increments that's much better than what the Lipschitz argument gives you.

Before we prove this, let's see how it implies Theorem 11.4.

Proof of Theorem 11.4. We just want to use Azuma–Hoeffding. Azuma–Hoeffding yields that

$$\mathbb{P}[|\mathsf{OPT} - \mathbb{E}[\mathsf{OPT}]| \geq t] \leq 2 \cdot \exp\left(-\frac{t^2}{\sum_{k=1}^n c_k^2}\right),$$

where the c_k's are such that $|Y_k - Y_{k-1}| \leq c_k$ (with probability 1). So if I have some bounds on just how large these increments are, then Azuma–Hoeffding gives me a concentration bound depending on that.

And these c_k's are exactly what Proposition 11.9 gives us — i.e.,

$$c_k = \min\left\{2\sqrt{d}, \frac{C_d}{(}n-k)^{1/d}\right\}.$$

So now I just need to evaluate the sum of squares of these guys; and we get

$$\sum_{k=1}^{n} c_k^2 \lesssim_d \sum_{k=1}^{n} \frac{1}{(n-k)^{2/d}}.$$

This looks like a slightly ugly sum, but now we can use integratino; so this is at most

$$\int_1^n \frac{1}{x^{2/d}}\,dx \asymp \begin{cases} \log n & \text{if } d = 2 \\ n^{1-2/d} & \text{if } d > 2. \end{cases}$$

□

So the key point is if you open up the black box of McDiarmid, you can prove a stronger bound on the increments which is not uniform in k.

Student Question. *Where did we ever use the fact that $c_k \le 2\sqrt{d}$?*

Answer. You don't really need this; you can just absorb it into the C_d.

§11.4.1 Better bounds on the increments

Now all that's left to do is to prove this proposition; and we're going to use Proposition 11.5.

To prove this, we'll need one short lemma. It says that if we start with any set of points $\mathcal{P}$ and add another point p, then the optimum doesn't move too much — specifically, it moves by at most $2 \cdot \operatorname{dist}(p, \mathcal{P})$. (This is related to the question asked earlier.)

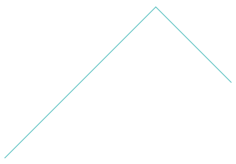
Lemma 11.10

For any $\mathcal{P} \subseteq [0,1]^d$ and $p \in [0,1]^d$, we have

$$\mathsf{OPT}(\mathcal{P}) \le \mathsf{OPT}(\mathcal{P} \cup \{p\}) \le \mathsf{OPT}(\mathcal{P}) + 2 \cdot \operatorname{dist}(p, \mathcal{P}).$$

In the worst case this can be $\sqrt{d}$. But now we have expectations, since $Y_k = \mathbb{E}[\mathsf{OPT} \mid \mathcal{P}_{\le k}]$. And when we take the expectation of this, that's exactly the kind of quantity arising in Proposition 11.5.

Proof of Lemma 11.10. We'll also do a proof by picture. Imagine I have my points $\mathcal{P}$, and I add my new point p. The first inequality is essentially obvious. For the second, imagine I take an optimal tour for $\mathcal{P}$ — I already have a tour for the points that don't include p. One way I can construct a new tour including p is if I take this tour; and when I hit the point q that's closest to p, I add a little excursion to this new point p (so I go from q to p and come back, and then resume the original tour). And certainly the optimal tour is no worse than the tour I just constructed this way. □



Now we'll take Lemma 11.10 and combine it with Proposition 11.5.

Proof of Proposition 11.9. I want to boud $|Y_k - Y_{k-1}|$, and they're both given by conditional expectations. So now let's open up the definition — we want to eventually massage this into something that just has quantities like $2 \cdot \operatorname{dist}(p, \mathcal{P})$ in them. We'll be explicit about what these expectations are over. So we fix p_1,

$\dots$, p_k (Y_k and Y_{k-1} are random variables, which depend on the first k points; so I need to show that for every choice of these first k, the conditional expectations over the choice of the remaining points satisfies this bound). Then we have

$$|Y_k - Y_{k-1}| = \left|\mathbb{E}_{p_{k+1},\dots,p_n}[\mathsf{OPT}(p_1,\dots,p_n)] - \mathbb{E}_{p'_k,p_{k+1},\dots,p_n}[\mathsf{OPT}(p_1,\dots,p'_k,\dots,p_n)]\right|.$$

(In Y_k I've fixed the first k points; in Y_{k-1} I've picked the first $k-1$, and I do a bit more averaging over p'_k.)

The only difference between these two distributions is the kth input (where we have p'_k vs. p_k), so we'll focus on this difference. (If you want, we can couple the two distributions.)

This is upper-bounded by what happens if I take the worst choice of p'_k and pull the expectation out — so it's upper-bounded by

$$\sup_{p_k} \mathbb{E}_{p_{k+1},\dots,p_n}[\left|\mathsf{OPT}(p_1,\dots,p_k,\dots,p_n) - \mathsf{OPT}(p_1,\dots,p'_k,\dots,p_n)\right|]$$

(I can use Jensen to pull out the expectation, and rather than taking a random p'_k, I take the worst one).

Now this is quite nice, because I can use Lemma 11.10. These are on the same set of points except for p_k vs. p'_k, so I can compare the optima with what would happen if I just deleted the two points — i.e., I can compare with $\mathsf{OPT}(\mathcal{P}_{-k}) = \mathsf{OPT}(p_1,\dots,p_{k-1},p_{k+1},\dots,p_n)$. So this is always upper-bounded by

$$2\sup_{p'_k} \mathbb{E}_{p_{k+1},\dots,p_n}[\mathrm{dist}(p_k,\mathcal{P}_{-k}) + \mathrm{dist}(p'_k,\mathcal{P}_{-k})]$$

by Lemma 11.10. Now I'm really using the fact that the sets of points I'm computing on agree on everyone except the kth point, but I have a random choice over p_{k+1}, $\dots$, p_n.

And I'm only going to blow up this distance if I shrink the set, so I can replace it with the set of points I'm taking an expectation over; this is upper-bounded by

$$2\sup_{p'_k} \mathbb{E}_{p_{k+1},\dots,p_n}[\mathrm{dist}(p_k,\mathcal{P}_{>k}) + \mathrm{dist}(p'_k,\mathcal{P}_{>k})].$$

And now we're essentially done — we have arbitrary points p_k and p'_k, and we're measuring their distance to $n-k$ *randomly* chosen points and looking at the expected minimum distance, so we can use Proposition 11.5 — we have

$$2\sup_{p'_k} \mathbb{E}_{p_{k+1},\dots,p_n}[\mathrm{dist}(p_k,\mathcal{P}_{>k}) + \mathrm{dist}(p'_k,\mathcal{P}_{>k})] \lesssim \frac{1}{(n-k)^{1/d}}$$

by Proposition 11.5. (The first points p_1, $\dots$, p_k were really arbitrary, and they didn't play a role in the analysis in the end.) □

§12 March 12, 2025 — Optimization over Erdős–Rényi

The project proposal deadline has been moved to next Friday. The entire goal of the initial proposal is so that we've thought about what we want to do for the project.

§12.1 Clique and independent set

Today we'll continue discussing optimization problems in average-case instances. Last time we saw how to analyze TSP given a random set of points in Euclidean space. This lecture, we'll discuss optimization problems in random graphs. In particular, we'll look at the following problems.

Problem 12.1 (Clique)

Find $S \subseteq V$ maximizing $|S|$ such that all pairs of vertices in S are connected by an edge.

We'll also look at the complementary problem.

Problem 12.2 (Independent set)

Find $S \subseteq V$ maximizing $|S|$ such that *no* pair of vertices in S is connected by an edge.

These are classic combinatorial optimization problems. They're extremely hard — not only are they NP-hard, but they're NP-hard to even approximate up to any reasonable multiplicative problem. So these problems are really hard in the worst case. But today we'll look at the average-case setting, where graphs are drawn according to the Erdős–Rényi distribution (where each edge is included independently with probability p).

Notation 12.3. Let $\alpha(G)$ be the size of the largest independent set (the *independence number*), and $\omega(G)$ the size of the largest clique (the *clique number*).

§12.2 Concentration of the clique number

We want to analyze average-case performance, so first let's get a sense of what we're comparing against. For convenience, we'll consider $p = 1/2$.

Theorem 12.4

For every $\varepsilon > 0$, we have $\mathbb{P}[\omega(\mathcal{G}(n, 1/2)) \in (1 \pm \varepsilon) \cdot 2\log_2 n] \geq 1 - o(1)$.

So we can very tightly understand, at least for most graphs, what the size of the largest clique is going to be.

Proof idea. The proof is just by first and second moments. We pick some k and look at the expected number of size-k cliques in the graph, and compute its expectation; and we find that around this threshold, it goes from growing with n exponentially fast, to going to 0 exponentially fast. And you can also compute the variance, and show that you actually do get a clique of this size. □

§12.3 A greedy algorithm

Now that we have our benchmark, we'll discuss an *algorithm* for actually finding a large clique. We'll look at a greedy algorithm, which is perhaps the first algorithm one might try. We'll just order the vertices arbitrarily and loop through the vertices, and we'll add a vertex to the set if it's connected to everything before it.

Algorithm 12.5

Order the vertices arbitrarily as $v_1, \ldots, v_n$, and initialize $S = \emptyset$.

Loop through the vertices, and add v_i to S if v_i has an edge to each vertex of S.

So we're just going to greedily add vertices — if I can add a vertex, I'll add it.

Now we want to understand the performance of this algorithm, at least for a random instance.

Theorem 12.6

For every $\varepsilon > 0$, we have $\mathbb{P}_{G \sim \mathcal{G}(n,1/2)}[\text{size of clique produced by greedy} \in (1 \pm \varepsilon) \log_2 n] \geq 1 - o(1)$.

So it's still of order $\log n$, but without the factor of 2. In particular, if you compare these two things, this says that on a typical graph you'll find a reasonably large clique — one that approximates the optimum up to a factor of 2.

Remark 12.7. This is definitely not true in the worst case — in the worst-case you can make the greedy algorithm find a clique only of size 2, even if the maximum clique is much larger (e.g., of size n). For example, for independent set (which is equivalent to Clique), consider a complete bipartite graph where we remove a perfect matching. If I greedily pick one vertex on the left and its partner on the right, then that's it — I can no longer add any more vertices. But there's clearly a very large independent set, just by taking one side of the bipartition. So the optimum is of order n, but the greedy algorithm only finds something of size 2.

So this theorem means that Greedy is pretty good on average — for typical instances (for some natural meaning of 'typical').

It's easiest to prove this by reframing the algorithm just slightly — so we'll consider a slightly different implementation, but it's functionally equivalent to the formulation above.

Algorithm 12.8 (Reformulation of Greedy)

Initialize $S_0 = \emptyset$ and $R_0 = V$.

While $R_t \neq \emptyset$:

- Pick an arbitrary vertex $v \in R_t$ and add it to S_t.
- Delete all vertices in R_t which don't have an edge to v.

We'll think of R as being the set of 'candidate' vertices that we *can* add to the clique.

So this is really just the same algorithm, but we're formulating it slightly differently, where now we're tracking a set of candidate vertices, or possible vertices I can add to the clique. The moment I add a vertex, that removes all the vertices that *don't* have an edge to it from the candidate pool (since if I added one of those, I wouldn't get a clique).

§12.3.1 The expected behavior

With this notation, now let's think about what we expect this algorithm to do. Because I'm adding edges between vertices independently with probability $\frac{1}{2}$, the expected behavior of this algorithm looks like the following. Let's look at how fast the size of this candidate set is decreasing. What we'd expect is that

$$|R_{t+1}| \approx \frac{1}{2} |R_t| ,$$

because I add a vertex, and out of all the other vertices in R_t, there's going to be an edge between that vertex and the vertex I added independently with probability $\frac{1}{2}$. So in expectation, once I add that vertex, I'm going to kill off half of my remaining vertices.

And now if it really behaved like the expected behavior, that's why you have $\log_2 n$ — if the size of the candidate set decays by a factor of 2 every time I add a vertex, then after $\log_2 n$ steps I'm going to hit the empty set.

So really, all we have to do is prove that the algorithm really tracks this expected behavior. And to do that, we're going to use our favorite tools — Chernoff-type bounds and so on.

§12.3.2 Formalizing the intuition

To formalize this, let $\delta(n) \to 0$ be some sufficiently slowly decaying function (it's basically going to be how far we're deviating from this expected behavior). Consider the following events: Let A_k be the event

$$\left(\frac{1}{2} - \delta(n)\right)|R_{k-1}| \le |R_k| \le \left(\frac{1}{2} + \delta(n)\right)|R_{k-1}|$$

(this is the event that my candidate set shrinks by roughly a factor of 2). What we want to say is that these events occur with overwhelmingly high probability.

The proof is going to go about how you think, but there's one slight technicality — to handle the case where the size of the candidate set has shrunk a lot (when it's no longer of order n, but maybe n^ε or maybe even smaller) — then the concentration bounds won't be strong enough, so there'll be some issues there. But other than that, the proof will work how you'd expect.

Let $T = (1-\varepsilon)\log_2 n$. For the lower bound, we want to say that after this many iterations, the number of candidate vertices is still positive — there's still some vertices we haven't checked. We'll handle the upper bound in a moment.

First let's look at the probability that all these events A_k occur up to time T — i.e., $\mathbb{P}[\bigcap_{k=1}^T A_k]$. (If I can control this, then I have very strong control on the number of candidate vertices.)

These events are definitely not independent — A_{k+1} definitely depends on A_k, because it depends on the size of the candidate sets. So we can't just factor it as a product. But we can factor it as a product if we introduce additional conditioning — we can write it as

$$\mathbb{P}_{k=1}^T \mathbb{P}\left[A_k \,\middle|\, \bigcap_{i=1}^{k-1} A_i\right].$$

Now, if all of $A_1, \dots, A_{k-1}$ have occurred, then I have very good control on the size of my candidate set, so I can use a Chernoff bound to bound this probability — this is lower-bounded by

$$\prod_{k=1}^T \left(1 - 2\exp\left(-\frac{n_k \delta(n)^2}{2}\right)\right),$$

where n_k satisfies

$$\left(\frac{1}{2} - \delta(n)\right)^k n \le n_k \le \left(\frac{1}{2} + \delta(n)\right)^k n.$$

In particular, because $T = (1-\varepsilon)\log_2 n$, these n_k's are all at least roughly n^ε — specifically, $n_k \gtrsim n^{\varepsilon - o(1)}$. In particular, these are nontrivially large, so this bound is meaningful — these are exponentially small quantities. (We're letting $\delta(n)$ go to 0 very slowly, e.g., as $1/(\log n)$.)

This essentially already proves the lower bound — it says with very high probability (this is at least $1-o(1)$), all these events occur. So I've run up to this number of iterations, and I still have roughly n^ε remaining vertices to look at. In particular, I haven't terminated by this point, so I've definitely constructed a clique of at least this size. In other words, teh fact that $\mathbb{P}[\bigcap_{k=1}^T A_k] \ge 1 - o(1)$ means that

$$\mathbb{P}[|\mathsf{greedy}| \ge (1-\varepsilon)\log_2 n] \ge 1 - o(1).$$

This also implies that $|R_T| \approx n^{\varepsilon \pm o(1)}$.

§12.3.3 The upper bound

So we already have our lower bound; now all we have to do is prove our upper bound.

For our upper bound, basically what we need to analyze now is how many more of the vertices in R_T we can add to our clique. Certainly, the vertices I add from this set must *themselves* form a clique. So for the upper bound, I always have

$$|\mathsf{greedy}| \le T + \omega(G[R_T]).$$

(The remaining vertices in R_T that I add must themselves form a clique, so I certainly have this kind of upper bound.)

We also know from the first theorem that $\omega(G[R_T])$ is small — now I only have n^ε many vertices, so replacing n with n^ε in that theorem, we get that

$$\omega(G[R_T]) \le O(\varepsilon \cdot \log_2 n)$$

with probability $1 - o(1)$ (because on that induced subgraph, it's again $\mathcal{G}(n, 1/2)$).

This implies $\mathbb{P}[|\mathsf{greedy}| \le (1 + O(\varepsilon)) \log_2 n] \ge 1 - o(1)$.

§12.4 Independent sets in sparse graphs

So this is all very nice, and just uses bare-handed methods.

Now we'll discuss a different regime for this problem — we'll look at independent sets, but when the graph is sparse.

Theorem 12.9 (Frieze 1990)

Fix $d \in \mathbb{N}$. For all $\varepsilon > 0$, we have

$$\mathbb{P}\left[\alpha(\mathcal{G}(n, d/n)) \in (1 \pm \varepsilon) \cdot \frac{2 \log d}{d} \cdot n\right] \ge 1 - o(1).$$

We think of d as a constant independent of n, so we have a sparse graph. So I have a very sparse graph with few edges, which means I expect the size of the independent set to be much larger — of linear size. And we can actually get even the right constant.

For the algorithmic part, we'll run the same greedy algorithm, where I add a vertex if it doesn't connect to any of the previous vertices I've added.

Theorem 12.10

For all $\varepsilon > 0$ and $d \in \mathbb{N}$, we have

$$\mathbb{P}_{G \sim \mathcal{G}(n, d/n)}\left[|\mathsf{greedy}| \in (1 \pm \varepsilon) \cdot \frac{\log d}{d} \cdot n\right] \ge 1 - o(1).$$

So again we've lost the factor of 2, but we get a constant-factor multiplicative approximation.

Remark 12.11. There's this gap between 2 and 1 here, and there's been lots of recent work studying whether or not that gap is inherent. It's even been posed as a conjecture that there are no polynomial-time algorithms achieving better than this threshold $\frac{\log d}{d} \cdot n$ with high probability. (Of course you can find an independent set of size $\frac{2 \log d}{d} \cdot n$ by brute force by Theorem 12.9, but that requires exponential time.)

The plan for the rest of the lecture is we'll sketch the proof of Theorem 12.9, or some of the interesting ideas. It involves some heavy calculation, so we'll try to minimize that, but we'll see how it goes. And then we'll see a different way of analyzing the greedy algorithm, which is also very elegant.

§12.5 Proof of Theorem 12.9

§12.5.1 First moments

First let's get some intuition for why we have this threshold $2 \cdot \frac{\log d}{d} \cdot n$.

For the upper bound, we'll again use the moment method. Let's fix some constant $\eta \in (0,1)$ (which we'll determine in a moment). Let $k = \eta \cdot n$, and let

$$N_k = \#(\text{size-}k \text{ independent sets}).$$

Now I want to compute a first moment — I want to compute $\mathbb{E}[N_k]$ and see for what values of η this thing grows or decays with n, and that's how we're at least going to predict this threshold.

Let's consider $\mathbb{E}[N_k]$. I have $\binom{n}{k}$ possible sets to consider, and I need to ensure that the set is independent, meaning that no pair of vertices is connected by an edge; so I have

$$\mathbb{E}[N_k] = \binom{n}{k}\left(1 - \frac{d}{n}\right)^{\binom{k}{2}}.$$

Now this is where some approximations and calculations will begin. I'm looking at k which is of order n. So I'm going to approximate the binomial coefficient using Stirling's formula. Here's a generic useful approximation (which we used even in the first lecture):

Fact 12.12 — We have $\binom{n}{\eta n} = \exp(n \cdot H(\eta) \pm o(n))$, where $H(\eta) = -\eta \log \eta - (1-\eta)\log(1-\eta)$.

We can also approximate $1 - \frac{d}{n}$ using $1 - x \approx e^{-x}$ and $\binom{k}{2} \approx \frac{\eta^2 n^2}{2}$, so we get

$$\mathbb{E}[N_k] = \exp\left(n\left(H(\eta) - \frac{d}{2}\eta^2\right) \pm o(n)\right).$$

So basically, whether $H(\eta) - \frac{d}{2}\eta^2$ is positive or negative will determine whether this expectation blows up or decays to 0 as $n \to \infty$ (this is the leading-order term; everything else is much smaller).

So now let's think about when this thing is positive or negative. We've seen that if $H(\eta) - \frac{d}{2}\eta^2 < 0$ then $\mathbb{E}[N_k] \to 0$, and if it's positive then $\mathbb{E}[N_k] \to \infty$. So now let's look at when this thing is 0, i.e.,

$$H(\eta) - \frac{d}{2}\eta^2 = 0.$$

We can rearrange this to

$$\frac{-\eta \log \eta - (1-\eta)\log(1-\eta)}{\eta^2} = \frac{d}{2}.$$

And we claim this occurs around $\eta = (1 \pm o_d(1)) \cdot \frac{2}{d} \cdot \log d$ (where the error depends on d). This explains where this threshold comes from.

§12.5.2 Second moments

Now let's do a second moment — I want to show that $\mathbb{E}[N_k^2]$ is not too much larger than $\mathbb{E}[N_k]^2$. If we skip a few steps, we have

$$\mathbb{E}[N_k^2] = \sum_{j=0}^{k} \binom{n}{k} \cdot \binom{k}{j} \cdot \binom{n-k}{k-j} \cdot \left(1 - \frac{d}{n}\right)^{2\binom{k}{2} - \binom{j}{2}}.$$

This looks a bit complicated, but the point is we break N_k as a sum of indicator random variables $\mathcal{I}_A$ (the indicator for whether A is a clique). Then we think of j as $|A \cap B|$ (where A and B are the two cliques in our term $\mathcal{I}_A \mathcal{I}_B$); the $\binom{n}{k}$ corresponds to choosing A, the $\binom{k}{j}$ to choosing the intersection $A \cap B$, and the $\binom{n-k}{k-j}$ to choosing the rest of B.

Now we want to ask, what's the value of j that maximizes this thing? We'll use one more useful approximation, which is sort of a generalization of the above:

Fact 12.13 — If μ is a distribution on $\{1, \dots, t\}$, then

$$\binom{n}{\mu_1 n, \dots, \mu_t n} = \exp(n \cdot H(\mu) \pm o(n)).$$

(This is a generalization of the earlier approximation for a binomial coefficient, where we just had a distribution on a support of size 2.)

So this is is going to be roughly

$$\mathbb{E}[N_k^2] \approx \max_{0 \le j \le k} \exp\left(n \cdot \left(H\left(\frac{j}{n}, \frac{k-j}{n}, \frac{k-j}{n}, 1 - \frac{2k-j}{n}\right) - d \cdot \frac{k^2}{n} + \frac{d}{2} \cdot \frac{j^2}{n}\right)\right).$$

This looks absolutely horrendous. But one point we want to get across is actually counter to what one might expect, the second moment fails us here. Skipping a lot of steps, we have

$$\frac{\mathbb{E}[N_k^2]}{\mathbb{E}[N_k]^2} \approx \exp\left(n \cdot (G(\eta) - 2H(\eta))\right)$$

(the left-hand side is the thing I need to be bounded for second moments to work), where

$$G(\eta) = \sup_{0 \le \zeta \le \eta} \left\{ H(\zeta, \eta - \zeta, \eta - \zeta, 1 - 2\eta + \zeta) + \frac{d}{2}\zeta^2 \right\}.$$

(Recall $k = \eta n$ and $\eta \in (0, 1)$ is a constant.)

This is a very complicated function, but the key point is that this is actually exponentially large — we have $G(\eta) > 2H(\eta)$. This means if I look at this second-moment type quantity, it's too large — we have

$$\frac{\mathbb{E}[N_k^2]}{\mathbb{E}[N_k]^2} \approx \exp(\Omega(n)).$$

So the second moment fails us — it's going to imply at best that $\mathbb{P}[N_k \ge 1] \ge \exp(-\Omega(n))$, which is not good.

Student Question. *Should we have the intuition this is because the supremum is not at $\zeta = 0$? So essentially that means in that sum, the main contribution doesn't come from disjoint guys?*

Answer. Yes — it's not going to be maximized at $\zeta = 0$, it'll be something in between.

So there are a bunch of calculations here, but the point to highlight is that the second moment fails us, unfortunately. But there'll be a nice trick using concentration that'll allow us to recover.

§12.5.3 Frieze's idea

The second idea we'll add on top of this was — here the variance was too big to deduce anything useful. So we want to look at a sparser class of independent sets in some sense, where the variance will be slightly better behaved. For this class, it'll turn out the second moment also doesn't work, but there'll be a neat trick to add on top of it to make it work. This was Frieze's idea: what we're going to do is look at a special class of independent sets.

Fix a partition $\mathcal{P}$ of the vertex set V into blocks of size

$$L = \frac{d}{(\log d)^2}$$

(one can imagine first setting L as a parameter and then optimizing the choice of L in the end). We're going to look at independent sets $S \subseteq V$ with the additional restriction that $|S \cap P| \leq 1$ for all $P \in \mathcal{P}$. So maybe we've partitioned the vertex set into a bunch of pieces; I pick at most one vertex from each piece, and I have to enforce the independent set constraint.

If you sort of calculate things very carefully using a similar type of thing (which we won't do), what you'll find is that letting

$$M_k = \#(\text{size-}k \text{ independent sets with this property}),$$

if we look at the second moment-type quantity, it'll still be bounded by something exponentially large, but that constant is going to be small, and this will matter for us — we'll have

$$\frac{\mathbb{E}[M_k^2]}{\mathbb{E}[M_k]^2} \leq \exp\left(\frac{n}{d^{4/3}}\right).$$

The specifics of how this computation goes isn't going to be important for us; what will be important is how we take the failure of the second moment method and fix it.

Remark 12.14. The important thing about this is that $4/3 > 1$ (in the dependence on d).

So what's the idea? Now let's think about the following function of graphs (we're going to show some kind of concentration inequality). Let $\beta(G)$ be the size of the largest independent set with that constraint — so

$$\beta(G) = \max\{k \mid \text{exists size-}k \text{ independent set with the additional constraints}\}.$$

Of course we always have $\beta(G) \leq \alpha(G)$. And what we're going to do is lower-bound this guy.

The first claim is that this guy has nice concentration; and we're going to use that to show that its expectation is large, by comparing that with what we get from the second moment bound.

Claim 12.15 — The function $\beta(G)$ is 1-Lipschitz in the following sense: Fix some $P \in \mathcal{P}$. If we let G and G' be graphs which differ on the set of edges incident to vertices in P (but nowhere else), then

$$\left|\beta(G) - \beta(G')\right| \leq 1.$$

And from this Lipschitzness-type property, we'll be able to conclude some type of concentration for $\beta(G)$ when G is drawn randomly.

In the interest of time we won't prove this in full, but we'll sketch it briefly.

Proof sketch. If S is an independent set in G with this partition constraint, then if we look at S and remove whatever vertices it has in P, the resulting set $S \setminus P$ is independent in G'. And since I've imposed that my independent set contains at most one vertex in any part $P \in \mathcal{P}$, removing P from S loses at most one vertex. □

Now we want to combine this with the second moment bound.

Lipschitzness (using McDiarmid, or Azuma–Hoeffding on the Doob martingale) implies that

$$\mathbb{P}\left[|\beta(G) - \mathbb{E}[\beta(G)]| \geq t \cdot \frac{\log d}{d}\right] \leq 2\exp\left(-\frac{t^2 n}{2d}\right).$$

(In this calculation, we use the fact that our choice of the size L and so on.)

First observe that if $\mathbb{E}[\beta(G)]$ is large, then just by this concentration inequality, we're going to get what we want — so we're done if we can show that

$$\mathbb{E}[\beta(G)] \geq (1-\varepsilon) \cdot \frac{2\log d}{d} \cdot n.$$

If we can do this, then by concentration we'll get that with high probability there will exist some independent set achieving the expectation (up to a small error).

So now let's prove this. We'll do this by contradiction — suppose not. Our second moment calculation

$$\frac{\mathbb{E}[M_k^2]}{\mathbb{E}[M_k]^2} \leq \exp\left(\frac{n}{d^{4/3}}\right)$$

implies, by Paley–Zygmund, that

$$\mathbb{P}[M_k \geq 1] \geq \exp\left(-\frac{n}{d^{4/3}}\right).$$

Then we'll combine this with the above concentration bound, which says that

$$\mathbb{P}\left[\beta(G) \geq (1-\varepsilon) \cdot \frac{2\log d}{n}\right] \leq \exp\left(-\Omega_\varepsilon\left(\frac{n}{d}\right)\right)$$

(the point is that this is exponential in n, where the constant is roughly $1/d$). (This is just saying that if the expectation is small and the probability I deviate from the expectation is also small, then the probability the thing is large is also small.)

But now we can compare this to the second moment bound (where $k = (1-\varepsilon) \cdot \frac{2\log d}{d} \cdot n$). Here we have a lower bound on the probability we have an independent set of size k, and we just saw an upper bound. And these things are scaling differently — they both scale linearly in n, but with different dependences on d. In particular, the lower bound is going to be larger than the upper bound. So the above thing contradicts

$$\mathbb{P}[M_k \geq 1] \geq \exp\left(-\frac{n}{d^{4/3}}\right).$$

This means it must be the case that $\mathbb{E}[\beta(G)]$ is actually quite large, specifically

$$\mathbb{E}[\beta(G)] \geq (1-\varepsilon) \cdot \frac{2\log d}{d} \cdot n.$$

And then you can invoke concentration, and you're done.

Remark 12.16. This seems like a very roundabout way to reason about the expected value of $\beta(G)$, but it's really clever to combine these concentration inequalities in this way.

Student Question. *Are we assuming d is large with respect to ε or something?*

Answer. Yes — think of $\varepsilon \approx \frac{1}{\log\log d}$ (something decaying to 0, but very slowly in d). (There are also more refined results in Frieze's paper; there's a link in the notes.)

§12.6 Analysis of the greedy algorithm

Finally, we'll discuss a different way to think about the greedy algorithm, which is conceptually quite nice.

What we'll imagine for this greedy algorithm is — we'll imagine running this algorithm not on a *finite* graph, but on an *infinite* graph. So we consider an infinite random graph $\mathcal{G}(\infty, \frac{d}{n})$, where we connect two vertices (marked by positive integers) with the usual Erdős–Rényi type of protocol.

Student Question. *What is n?*

Answer. We're fixing d and n at the start. Eventually I'm going to truncate this graph to a finite thing, but it's useful to think about the graph as first being infinite.

So we connect $i \sim j$ independently with probability $p = \frac{d}{n}$.

Now let's think about running this whole algorithm on the infinite graph.

Fact 12.17 — We have $\mathcal{G}(\infty, p)[\{1, \dots, n\}] = \mathcal{G}(n, p)$.

(In other words, if I take this infinite graph and restrict it to just the first n vertices, this is equal in law to $\mathcal{G}(n, p)$.)

Now let's look at the number of steps it takes to add each next vertex — let T_k be the number of steps it takes to add the $(k+1)$th vertex, after adding the kth vertex. You can think of these as marking the timestamps for when I add a vertex to my independent set.

Then another way to formulate the size of the independent set we get is that

$$|\mathsf{greedy}| = \max\left\{k \,\middle|\, \sum_{j=1}^{k} T_j \le n\right\}.$$

(I just need to ensure that I don't exhaust all the vertices of my graph.)

Now the key point behind this construction is that these times T_j are actually very nice to analyze. If you've seen the coupon collector problem, for example, it's exactly an analysis in that flavor. The key observation is that these T_ks are independent, and we have exact formulas for how they're distributed.

Claim 12.18 — The collection $\{T_k\}$ is independent, and $T_{k+1} \sim \mathsf{Geom}(q_k k)$, where $q_k = (1-p)^k$.

In particular, this means for all t, we have

$$\mathbb{P}[T_{k+1} = t] = (1 - q_k)^{t-1} q_k.$$

(The geometric distribution is, if I'm tossing a coin repeatedly, the number of times I need to toss it to get heads.)

We'll talk about why this is true; and then if you analyze these random variables and prove concentration using Chebyshev or something, you can analyze the greedy algorithm.

Proof of claim. Suppose I've already added k vertices to S, and I want to look at when I add the $(k+1)$th vertex. For all vertices v visited after this point, there's a $(1-p)^k$ probability that v has no neighbors in S (because I'm sampling edges independently). □

So these random variables have a geometric distribution; and they're nicely concentrated (you can compute their expectation and variance explicitly), and that lets you analyze this sum. Where you get the expected size of the independent set found by the greedy algorithm is by looking at the largest k for which $\mathbb{E}[\sum_{j=1}^{k} T_j] \le n$.

§13 March 17, 2025 — Janson's lower tail inequality and positive correlations

In the next couple of lectures, we'll come back to random graphs. We're going to see some really fundamental properties about random graphs — next lecture we'll talk about the appearance of the 'giant component,' one of the most famous phase transitions. But today we'll talk about *local* substructures, which we'll analyze using fairly general methods.

§13.1 Motivating problem

For simplicity, we're just going to look at counting triangles in Erdős–Rényi, but you can generalize this machinery substantially.

Question 13.1. For a graph G, let T_G be the number of triangles in G. What is

$$\mathbb{P}[T_{\mathcal{G}(n,p_n)} \leq (1-\delta)\mathbb{E}[T_{\mathcal{G}(n,p_n)}]]?$$

In words, we're looking at the number of triangles in the Erdős–Rényi graph, and we want to upper-bound the probability that it's small (substantially below its expectation).

§13.2 Positive correlations

We want to study this probability through some fairly general-purpose tool. For this, we'll introduce the concept of *positive correlations*, which we'll discuss in a generic setting: Let ν be a product measure on $\{0,1\}^N$ — a *product measure* means if I look at a random sample from ν, then the coordinates of that random sample are going to be jointly independent. Equivalently, this means there exist $p_1, \ldots, p_N$ such that $\nu(x) = \prod_{i:x_i=0} p_i \prod_{i:x_i=1}(1-p_i)$. (The Erdős–Rényi graph is an example of this.)

We'll be looking at a special class of functions, of which triangle counts will be an important example.

Definition 13.2. We say a function $f : \{0,1\}^N \to \mathbb{R}$ is *increasing* if $f(x) \leq f(y)$ for all x and y such that $x_i \leq y_i$ for all i.

In other words, we're looking at a partial order on the entire set of N-bit strings, where we say one bit-string is larger than another if it's larger coordinatewise. And now we look at this partial order, and we say a function is increasing if it satisfies the obvious inequality.

Similarly, we say an event is increasing if its indicator function is increasing.

Here's a theorem, which studies the correlations between increasing functions with respect to *any* product measure.

Theorem 13.3 (Harris, Kleitman)
For any product measure ν on $\{0,1\}^N$ and any pair of increasing functions $f, g : \{0,1\}^N \to \mathbb{R}$, we have

$$\mathbb{E}_\nu[fg] \geq \mathbb{E}_\nu[f]\mathbb{E}_\nu[g].$$

We use $\mathbb{E}_\nu[g]$ as shorthand notation for $\mathbb{E}_{x\sim\nu}[g(x)]$; and with fg, we're looking at the entry-wise multiplication of functions.

Let's step back and look at some examples of functions and product measures: we'll look at various properties one might be interested in for Erdős–Rényi.

Example 13.4

Let ν be the distribution $\mathcal{G}(n, p_n)$ (where $N = \binom{n}{2}$). Let

$$f(G) = \mathbf{1}[G \text{ is connected}];$$

this is an increasing function, because if I add more edges, I'm only going to preserve connectivity. Let $g(G) = \chi(G)$ (this is the chromatic number — the minimum number of colors needed to color the vertices so that no pair of adjacent vertices have the same color); by adding edges, I can only increase the chromatic number of the graph. Then Theorem 13.3 says, as a black box, that these two properties are correlated in the sense that

$$\mathbb{E}[\chi(\mathcal{G}(n, p_n)) \mid \mathcal{G}(n, p_n) \text{ connected}] \geq \mathbb{E}[\chi(\mathcal{G}(n, p_n))].$$

So I have some very complicated random variable (like the chromatic number or connectivity). Without using any of its properties (other than that it's increasing), I can already say that if I know the graph is connected, it's more likely to have a larger chromatic number. You can replace these with various other things — the indicator that the graph has a perfect matching or contains some local substructure or contains a Hamilton cycle or so on. So this is an extremely general theorem which applies to a wide class of functions.

We're not going to prove Theorem 13.3; it's a half-page induction argument (where you induct on N). But we're going to use it to prove lower tail bounds.

§13.3 Janson's inequality

We're going to state Janson's inequality in a very general framework. We'll imagine drawing $x \sim \nu$, and for $A \subseteq [N]$, let $\mathcal{I}_A = \mathbf{1}[x_i = 1 \text{ for all } i \in A]$. We can think of $x \in \{0,1\}^N$ as representing a random subset of $[N]$; and all this is saying is we want this subset to contain A.

Let $\mathcal{F} \subseteq 2^{[N]}$, and let $X = \sum_{A \in \mathcal{F}} \mathbf{1}_A$. Let $\Delta = \sum_{A \in \mathcal{F}} \sum_{B \in \mathcal{F} : A \cap B \neq \emptyset} \mathbb{E}_\nu[\mathcal{I}_A \mathcal{I}_B]$.

For example, you can think of $\mathcal{F}$ as consisting of all triples of vertices which would form a triangle; and X counts the number of occurrences of these substructures in our graph. We think of Δ as quantifying how independent these events are — we're summing the probabilities that the set I sampled contains both A and B.

Example 13.5

We can take $\mathcal{F} = \{\{uv, vw, uw\} \mid u, v, w \in [n]\}$ — so this is the set of possible triangles in $\mathcal{G}(n, p)$.

Here's the theorem, which is about lower tail probabilities:

Theorem 13.6 (Janson)

Let $\mu = \mathbb{E}_\nu[X]$. Then

$$\mathbb{P}_\nu[X \leq \mu - t] \leq \exp\left(-\frac{t^2}{2\Delta}\right) \quad \text{for all } 0 \leq t \leq \mu.$$

For instance, if we're looking at triangles, then μ is the expected number of triangles in Erdős–Rényi, and we're looking at the probability the number of triangles is substantially less than its expectation. And we get that it's exponentially small, where the exponent depends on how correlated these sets are in your random graph.

§13.3.1 Application — triangle counts

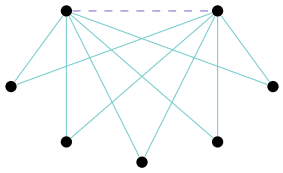
Example 13.7

Let's see what happens for triangle counts (where $X = T_{\mathcal{G}(n,p_n)}$).

To see why this theorem is interesting, let's see how it compares to what would happen if you tried using something like McDiarmid's inequality. We have $\mu = \mathbb{E}[T] = \binom{n}{3}p_n^3 \asymp n^3p_n^3$. We also know that if I add an edge to a graph, it can at most create roughly n triangles (if I want to use McDiarmid, I need to bound the Lipschitzness of this function; in this case the Lipschitz constant is n).

Fact 13.8 — By adding or removing an edge, I can create or destroy at most n triangles.

The picture for Lipschitzness is if we imagine you have two vertices u and v and there is no edge connecting them, but there's a whole bunch of other vertices which connect to both u and v, then if you add this edge you'll increase the number of triangles by roughly n. And this is kind of the worst case.



If you combine these two claims, then McDiarmid implies

$$\mathbb{P}[X \le \mu - t] \le \exp\left(-\frac{t^2}{2\cdot\binom{n}{2}\cdot n^2}\right)$$

(here $\binom{n}{2}$ is the number of independent random variables involved, and n^2 comes from the square of the Lipschitz constant).

Now let's think about what we want t to be. Remember that $\mu \asymp n^3p_n^3$; and now say $t = \delta\mu$. Then this tells us that

$$\mathbb{P}[X \le (1-\delta)\mu] \le \exp\left(-\Theta(\delta^2p_n^6n^2)\right).$$

You can see that this bound is meaningful (i.e., not vacuous) if and only if p_n is not too small — e.g., $p_n \gg n^{-1/3}$.

Now let's compare this with what we'd get if we used Janson's inequality. If we want to use Janson's inequality, the key thing we need to bound is Δ — this measure of dependency between our events. Basically, we want to look at pairs of triangles which share at least 1 edge, meaning that they share at least two vertices. There's really only two cases — we have the case where the two triangles are literally equal to each other, and we have the case where they form a diamond (we have two vertices u and v that they share, and then two other vertices w and w' forming the triangles). So we get

$$\Delta = \binom{n}{3}p_n^3 + O(n^4)p_n^5.$$

We can't directly simplify this because depending on the regime of p_n, one of the terms might dominate the other. But we can directly plug this in, and Janson implies

$$\mathbb{P}_\nu[X \le (1-\delta)\mu] \le \exp\left(-\frac{c\delta^2p_n^6n^6}{n^3p_n^3 + n^4p_n^5}\right).$$

(The constant c absorbs all the various constants in the expression for Δ.) Now if we do a bit of case analysis depending on what the scaling is for the denominator — we can immediately cancel some factors of n and p_n, and then simplify everything, to get

$$\exp\left(-\frac{c\delta^2 p_n^3 n^3}{1+np_n^2}\right) = \begin{cases} \exp(-\Theta(\delta^2 p_n^3 n^3)) & \text{if } p_n \lesssim n^{-1/2} \\ \exp(-\Theta(\delta^2 p_n n^2)) & \text{if } p_n \gtrsim n^{-1/2}. \end{cases}$$

The key point is that these are meaningful for basically any regime (unlike the one from McDiarmid).

The very vague intuition for why we're able to get a better bound here is that counting triangles are all monotone functions. And kind of by positive correlations between these triangles, if you see one, you expect to see many more — in some sense, it's the rich get richer phenomenon happening with triangle counts. So the idea is to use positive correlations between triangle counts, or more generally the various sets in $\mathcal{F}$.

Student Question. *Why is it p_n^5 and not p_n^4?*

Answer. I have 4 vertices, but 5 edges, and I need to ensure all of them are included.

Before we prove Janson's inequality, we'll make a couple of remarks about triangle counting.

Remark 13.9. It turns out this upper bound for the lower tail is essentially sharp. For instance, in the regime $p_n \gtrsim n^{-1/2}$, you can always lower-bound this thing by the probability the graph is empty — you have

$$\mathbb{P}[X \le (1-\delta)\mu] \ge \mathbb{P}[\mathcal{G}(n,p_n) \text{ empty}] = (1-p_n)^{\binom{n}{2}} \gtrsim \exp(-\Theta(p_n n^2)).$$

And that matches the bound that we get in the second case here.

Remark 13.10. One might ask if such a bound also holds for the *upper* tail — if I look at the probability that the number of triangles *exceeds* its expectation by a constant factor. This turns out not to be the case — the behavior is different for the upper tail. For precise quantitative dependencies, you can see the note. But intuitively, you can see that the presence of positive correlations certainly isn't going to help you bound the upper tail.

§13.3.2 Proof of Janson's inequality

Let's now prove Janson's inequality. As usual, when we're bounding tails of events, we're going to be looking at the moment generating function. For our purposes, it's going to be slightly more convenient to work with the cumulant generating function (which we already saw when looking at Hoeffding's inequality). We'll consider

$$\psi(s) = -\log \mathbb{E}_\nu[\exp(-s \cdot X)]$$

(we've put minus signs because we're interested in lower tails).

The key claim is an upper bound on this function, or rather its derivative.

Claim 13.11 — For every $s > 0$, we have $\psi'(x) \ge \mu \cdot \exp(-\frac{s\cdot\Delta}{\mu})$.

Of course, if you have a bound on the derivative of your function, you can integrate this bound to get a bound for the function itself.

Maybe before we prove this claim, let's use it to get our tail bound. To prove Janson's inequality from this, we use the standard trick

$$\mathbb{P}_\nu[X \le \mu - t] = \mathbb{P}_\nu[-\exp(-s\cdot X) \ge \exp(-s(\mu-t))] \le \exp\left(s(\mu-t) - \psi(s)\right)$$

(exponentiating both sides, and then applying Markov as usual). Picking the best choice of s, this means

$$\mathbb{P}_\nu[X \le \mu - t] \le \exp\left(\inf_{s>0}\{s(\mu - t) - \psi(s)\}\right).$$

So if I have a lower bound on the cumulant generating function, then I get an upper bound on this entire quantity.

And we can write the thing inside the exponential as

$$s(\mu - t) - \int_0^s \psi'(y)\, dy.$$

Then you can directly plug in the bound from the key claim and get the result — we always have

$$\int_0^s \psi'(y)\, dy \ge \frac{\mu^2}{\Delta}\left(1 - \exp\left(-\frac{s\Delta}{\mu}\right)\right),$$

and then you can optimize over s (there'll be some very explicit form for s) and get the inequality — we get an upper bound of $\exp(-t^2/2\Delta)$ by setting

$$s = -\frac{\mu}{\Delta}\ln\left(1 - \frac{t}{\mu}\right).$$

We won't go through the details of the calculation; but the key point is that we want to lower-bound the moment generating function, and to do that, it suffices to lower-bound its derivative.

So now let's prove the claim.

Proof of Claim 13.11. Maybe let's first compute this derivative — if we differentiate ψ, in the denominator we get whatever's inside the log, and in the numerator we get the derivative of that thing; and because expectation and differentiating are both linear, we can swap them to get

$$\psi'(s) = \frac{\mathbb{E}_\nu[X \cdot \exp(-s \cdot X)]}{\mathbb{E}_\nu[\exp(-s \cdot X)]}.$$

We can already see why Harris–Kleitman might be useful: In the numerator, we have an expectation of a product of functions — X and $\exp(-s \cdot X)$. That's in general nasty to deal with, but if you can decorrelate things and relate it to the expectation of the *individual* functions, then we'll be in good shape. We want to show that this is lower-bounded by

$$\mu \cdot \exp\left(-\frac{s\Delta}{\mu}\right).$$

Let's start by massaging the numerator — we can write

$$\text{numerator} = \sum_{A \in \mathcal{F}} \mathbb{E}[\mathcal{I}_A \cdot \exp(-s \cdot X)].$$

(Right now we haven't done anything yet — we've just used the definition.) Now I want to decompose X into terms that I can relate to A. So we're going to define the following. For each $A \in \mathcal{F}$, we let ν_A be the distribution of $x \sim \nu$ conditional on $\mathcal{I}_A(x) = 1$ — so I'm going to define a distribution where I condition on A happening. I'm also going to decompose X — I define

$$Y_A = \sum_{B \in \mathcal{F} : B \cap A \neq \emptyset} \mathcal{I}_B \quad \text{and} \quad Z_A = \sum_{B \in \mathcal{F} : B \cap A = \emptyset} \mathcal{I}_B.$$

So in particular, I can always decompose $X = Y_A + Z_A$.

Now let's do some inequalities. We have

$$\sum_{A\in\mathcal{F}} \mathbb{E}[\mathcal{I}_A \cdot \exp(-s\cdot X)].$$

The first thing we can do is condition on A happening — we can replace this by

$$\sum_{A\in\mathcal{F}} \mathbb{P}[\mathcal{I}_A = 1]\cdot \mathbb{E}_{\nu_A}[\exp(-s\cdot X)]$$

(just by using the law of total expectation). Also, from this decomposition, we can break X into a sum of two random variables, which are in general correlated, but we'll handle them using Harris–Kleitman — we can write this as

$$\sum_{A\in\mathcal{F}} \mathbb{P}[\mathcal{I}_A = 1]\mathbb{E}_{\nu_A}[\exp(-sY_A)\exp(-sZ_A)].$$

Now the observation is that individually, Y_A and Z_A are both increasing; that means $\exp(-sY_A)$ and $\exp(-sZ_A)$ are decreasing. And Harris–Kleitman also works if both functions are decreasing (I can replace f and g by $-f$ and $-g$, and the same inequality is going to hold). And ν was a product measure itself, so ν_A is also a product measure. So Harris–Kleitman gives that this is lower-bounded by

$$\sum_{A\in\mathcal{F}} \mathbb{P}[\mathcal{I}_A = 1]\cdot \mathbb{E}_{\nu_A}[\exp(-sY_A)]\mathbb{E}_{\nu_A}[\exp(-sZ_A)]$$

(just by Harris–Kleitman). One nice thing already pops out — because all the sets inside Z_A don't intersect my set A, if I condition on the values of the coordinates in A, that has no effect on what Z_A is. That means I can just get rid of the conditionining in the second expectation — we have

$$\mathbb{E}_{\nu_A}[\exp(-sZ_A)] = \mathbb{E}_{\nu}[\exp(-sZ_A)].$$

And this is lower-bounded by the same quantity if we replaced Z_A with *everybody* — this is lower-bounded just by $\mathbb{E}_\nu[\exp(-sX)]$ (because X is always bigger than Z_A).

And now I can move this to the other side — this in particular implies that

$$\psi'(x) \geq \sum_{A\in\mathcal{F}} \mathbb{P}[\mathcal{I}_A = 1]\mathbb{E}_{\nu_A}[\exp(-sY_A)].$$

(All we did was move $\mathbb{E}_\nu[\exp(-sX)]$ to the other side.)

So now we have a slightly simpler random variable Y_A capturing all the sets that intersect A; that's how we're going to get the dependency-like quantity Δ.

The rest of the proof is just massaging the right-hand side — we're going to apply Jensen's inequality over and over again, and then we'll be done. So let's do that. Here I have the expectation of an exponential; an exponential is a nice convex function, so I can push the expectation inside the exponential. (If f is a convex function, then the average of the function at two points is always going to be bigger than the function evaluated at the average of the two points.) So Jensen gives that

$$\text{RHS} \geq \sum_{A\in\mathcal{F}} \mathbb{P}[\mathcal{I}_A = 1]\cdot \exp\left(-s\cdot \mathbb{E}_{\nu_A}[Y_A]\right).$$

Now if you look at this quantity, the coefficients are a bunch of probabilities, so we're tempted to use Jensen one more time. These probabilities don't sum to 1 — they're not probabilities of individual outcomes, but marginals of various subsets of variables. But they have a nice sum — they sum to the mean μ. So if I insert μ on the inside and outside to get

$$\mu \sum_{A\in\mathcal{F}} \frac{\mathbb{P}[\mathcal{I}_A = 1]}{\mu}\cdot \exp\left(-s\cdot \mathbb{E}_{\nu_A}[Y_A]\right),$$

then the coefficients do sum to 1 and I can apply Jensen again — this gives a lower bound of

$$\mu \cdot \exp\left(-s \cdot \sum_{A \in \mathcal{F}} \frac{\mathbb{P}[\mathcal{I}_A = 1]}{\mu} \cdot \mathbb{E}_{\nu_A}[Y_A]\right)$$

(this is just because $\sum_A \mathbb{P}[\mathcal{I}_A = 1] = \mu$ by definition). And now we're essentially done — we just need to show that this thing inside the exponential is what's written in the key claim. This is equal to

$$\mu \cdot \exp\left(-\frac{s}{\mu} \cdot \sum_{A \in \mathcal{F}} \mathbb{E}[\mathcal{I}_A \cdot Y_A]\right)$$

(we can push $\mathbb{E}_{\nu_A}[Y_A]$ back into the expectation). And this is exactly equal to

$$\mu \cdot \exp\left(-\frac{s \cdot \Delta}{\mu}\right),$$

by the definition of Y_A (which captures all the contributions to X from sets B which intersect A). □

Student Question. *Can you explain how you went from* $\sum_{A \in \mathcal{F}} \mathbb{P}[\mathcal{I}_A = 1]\mathbb{E}_{\nu_A}[Y_A]$ *to* $\sum_{A \in \mathcal{F}} \mathbb{E}[\mathcal{I}_A \cdot Y_A]$*?*

Answer. If I look at $\mathbb{E}[\mathcal{I}_A Y_A]$, this is a sum of a bunch of things; the inside is only going to be nonzero if $\mathcal{I}_A = 1$. So I can sort of for free condition on $\mathcal{I}_A = 1$ and write this as $\mathbb{E}[\mathcal{I}_A Y_A \mid \mathcal{I}_A = 1]\mathbb{P}[\mathcal{I}_A = 1]$. And in the first expectation, now $\mathcal{I}_A$ is identically 1 so this just becomes $\mathbb{E}_{\nu_A}[Y_a]$.

§13.4 Stochastic domination

In the remaining 20 or so minutes, we'll switch gears slightly to prepare for the next lecture — in the next lecture we're going to start discussing this very famous phase transition in Erdős–Rényi, the transition from the graph having all its connected components tiny (of logarithmic size) to having one very large component (of linear size). This transition happens well before the phase transition for connectivity.

To prepare for that, we'll introduce one concept (which we already saw in some sense when we used the bound $\mathbb{E}_\nu[\exp(-s \cdot Z_A)] \geq \mathbb{E}_\nu[\exp(-s \cdot X)]$). This is the concept of *stochastic domination*, which is a way of comparing random variables.

To formalize this, let's recall the notion of coupling.

Definition 13.12. Fix two distributions μ and ν. A *coupling* is a joint distribution ξ (on pairs of outcomes) such that for all x we have $\sum_y \xi(x, y) = \mu(x)$, and for all y we have $\sum_x \xi(x, y) = \nu(y)$.

This basically says that the marginals of ξ are μ and ν (respectively).

Previously, when we defined couplings, that was in the context of proving the Poisson approximation theorem (if we have a binomial distribution where the success probability scales as $\frac{1}{n}$, that goes to not a normal distribution but a Poisson distribution). The notion of coupling turns out to be super powerful (we'll see it another time in the course later). But for now, we'll use the notion of coupling as a way of comparing random variables rigorously.

Definition 13.13. Let μ and ν be distributions on $\mathbb{R}$. We say ν *stochastically dominates* μ (written as $\mu \preceq \nu$) if there exists a coupling ξ of μ and ν such that

$$\mathbb{P}_{(X,Y)\sim\xi}[X \leq Y] = 1.$$

This definition extends in a direct way if we replace $\mathbb{R}$ with any partially ordered set (e.g., the Boolean cube), but for simplicity we'll just work with $\mathbb{R}$.

So this says that if I have a random variable distributed according to μ and another distributed according to ν, then I can couple them such that one is always smaller than the other.

Let's revisit the example of coupling biased coins.

Example 13.14

Let $0 < p < q < 1$. Then $\mathsf{Ber}(p) \preceq \mathsf{Ber}(q)$.

This is intuitive — you have a larger probability of seeing a 1 in the second distribution than in the first.

Proof. We can just use the coupling we discussed previously: to define a coupling ξ, we first sample $U \sim \mathsf{Unif}[0,1]$. Then we're just going to threshold this random variable to determine whether or not our coin comes up heads or tails — we set

$$(X,Y) = \begin{cases} (1,1) & \text{if } U \le p \\ (0,1) & \text{if } p < U \le q \\ (0,0) & \text{if } U > q. \end{cases}$$

We see that if I sample two coins from this coupling and throw away the second coin, the first coin is going to look like $\mathsf{Ber}(p)$ (because I set it to 1 if and only if this uniform variable U was less than p). Similarly, if I sample a pair of coins from this coupling and throw away the first (and only look at the second), that coin is distributed as $\mathsf{Ber}(q)$ — it comes up 1 if and only if $U \le q$. So marginally X is distributed according to $\mathsf{Ber}(p)$, and Y according to $\mathsf{Ber}(q)$. And in this coupling, we always have $X \le Y$. So this is a coupling that certifies $\mathsf{Ber}(p)$ is stochastically dominated by $\mathsf{Ber}(q)$. □

Now we'll look at a few more examples.

Example 13.15

Let $n \le m$, and fix $p \in [0,1]$. Then $\mathsf{Bin}(n,p) \preceq \mathsf{Bin}(m,p)$.

This is again somewhat intuitive — if I give you more trials, I should expect to see more successes.

Proof. Here's the coupling. We have a binomial distribution, which is a sum of independent coin flips — we have

$$\mathsf{Bin}(n,p) = \mathsf{Law}(b_1 + \cdots + b_n) \quad \text{for i.i.d. } b_1, \dots, b_n \sim \mathsf{Ber}(p).$$

So here's the coupling — I take the first n coin flips to be the same when I sample $\mathsf{Bin}(n,p)$ and $\mathsf{Bin}(m,p)$ (and for the second, for the remaining $m-n$ trials, I do them independently).

More explicitly, we sample i.i.d. $b_1, \dots, b_m \sim \mathsf{Ber}(p)$, and we set

$$X = b_1 + \cdots + b_n \quad \text{and} \quad Y = b_1 + \cdots + b_m.$$

Then $X \sim \mathsf{Bin}(n,p)$ and $Y \sim \mathsf{Bin}(m,p)$. But for sure we always have $Y \ge X$. (The coupling ensures that the first n trials are literally equal to each other, which is what makes this work.) □

Let's do one last example, and then we'll be done. This example is going to be about coupling Erdős–Rényi random graphs.

Example 13.16

Fix $n \in \mathbb{N}$ and $0 < p < q < 1$. Then $\mathcal{G}(n,p) \preceq \mathcal{G}(n,q)$.

(We haven't defined stochastic domination for random variables that aren't real-valued, but we use the natural ordering on graph — we say one graph is larger than another if it has all the edges of the smaller graph, i.e., $G \leq H$ if and only if $E(G) \subseteq E(H)$.)

Proof. The coupling for this is really just an extension of the coupling for pairs of biased coin flips — for every edge, I'm going to independently run one of these biased coin couplings from Example 13.14.

So for every pair of vertices $uv \in \binom{[n]}{2}$, I sample $\mathcal{U}_{uv} \sim \mathsf{Unif}[0,1]$ (independently). Then we include uv in $\mathcal{G}(n,p)$ if $\mathcal{U}_{uv} \leq p$, and similarly we include it in $\mathcal{G}(n,q)$ if $\mathcal{U}_{uv} \leq q$. This is going to result in a pair of correlated random graphs, each of which is marginally distributed according to Erdős–Rényi with the correct edge probability; and I'm always guaranteed that the second graph contains all the edges of the first (so it's larger in some sense, or denser). □

This notion of stochastic domination will be very useful in the next lecture, when we discuss the giant component.

Finally, we'll say one more thing, which is an alternative way to think about this last coupling (the one in Example 13.16). We can imagine that I first sample $G \sim \mathcal{G}(n,p)$. Then if I want to generate a second graph, what I can do is give every non-edge a second chance — for every $uv \notin E(G)$, with probability

$$\frac{q-p}{1-p}$$

(independently) I add it to G to produce H. This is another way of thinking about this coupling — G is distributed according to $\mathcal{G}(n,p)$, and the graph H I get out of this has edge probability q (because with probability p I got it in the first case, and there's an extra $q-p$ probability that I didn't get it in the first case but did in the second).

Student Question. *Could you explain again why the second graph has marginal $\mathcal{G}(n,q)$?*

Answer. Let's look at some pair of vertices uv; I want to look at the probability there's an edge between them after this whole process. There's a probability p I included it in the first chance to begin with. Then there's a $(1-p)$ probability I don't do that, times probability $(q-p)/(1-p)$ that I do add it on the second chance.

§14 March 19, 2025 — Giant component in Erdős–Rényi

Today we'll discuss one of the most important phase transitions that occurs in the Erdős–Rényi random graph, the emergence of a giant component.

§14.1 Stochastic domination

First, let's review the notion of stochastic domination from last time, which is a way of comparing two random variables.

Definition 14.1. Given two probability measures μ and ν on $\mathbb{R}$, we say ν *stochastically dominates* μ (written $\mu \preceq \nu$) if there exists a coupling ξ such that $\mathbb{P}_{(X,Y)\sim\xi}[X \leq Y] = 1$.

Example 14.2 (Biased coins)

If $q > p$, then $\mathsf{Ber}(q) \succeq \mathsf{Ber}(p)$. We can define a coupling by choosing a random variable $U \sim \mathsf{Unif}[0,1]$ and thresholding based on how its values compare to p and q.

Example 14.3

If $n \leq m$, then $\mathsf{Bin}(n,p) \preceq \mathsf{Bin}(n,q)$.

This is the main tool we'll use today, combined with the usual tools (concentration inequalities, first and second moments, and so on).

§14.2 The phase transition

We'll fix a constant d, which we think of as the average degree; and we'll think of p as being d/n (this is well below the connectivity threshold, which we saw is of order $(\log n)/n$).

Theorem 14.4 (Erdős–Rényi)

Fix $d > 0$, and let $p = d/n$.

- Suppose $d < 1$. Then with probability $1 - o(1)$, all connected components in $\mathcal{G}(n,p)$ have size $O(\log n)$.
- Suppose $d > 1$. Then with probability $1 - o(1)$, there exists a *unique* component of size $\Omega(n)$, and all other components have size $O(\log n)$.

In the second case, we think of the $\Omega(n)$-sized component as 'the giant.'

The picture to have in your mind is, imagine that I slowly increase the edge density in the random graph, and let's look at various regimes, where we start with $o(1/n)$, then d/n for $d < 1$, then d/n for $d > 1$, and all the way on the right we have $c(\log n)/n$ (for $c > 1$). T

he picture for how the graph evolves is that when it's very sparse, you have a bunch of tiny components (how small these components are will depend on how the $o(1/n)$ goes), and inside these components they'll be trees — there are actually no cycles whatsoever. Once you get into the roughly $1/n$ range, you're going to start to see some cycles, but the components remain quite small; and all the components are going to be trees or they might have one cycle. (We actually proved this on the homework.)

And then there's another jump at $1/n$ where now all of a sudden you have a huge component, occupying at least a constant fraction of the graph; and then you have a whole bunch of tiny satellites outside (of logarithmic size).

And finally, once you reach $(\log n)/n$, the whole graph becomes fully connected. (This is also the regime where you no longer have isolated vertices — below this threshold you'll have some lonely vertices on the outside.)

So this is what some of the most interesting transitions look like. The argument is somewhat complex, but we'll at least go through the main ingredients in the proof of this theorem.

§14.3 Branching process intuition

To start, we'll first discuss the intuition for why the transition happens at $1/n$. This comes again from a branching process perspective. Imagine that I fix some arbitrary vertex v, and I want to look at what the connected component containing v looks like, or how large it is. You can imagine you first reveal the vertices in its immediate neighborhood; and let's count them as N_1 (and say $N_0 = 1$). Then those neighbors reveal their neighbors. Since the graph is so sparse, we're going to have very few collisions, so this essentially behaves as a tree, and we get N_2. And we keep growing this process.

The hope is that if we look at this sequence of numbers $\{N_\ell\}$ — where N_ℓ is the number of vertices at distance exactly ℓ from v — then this sequence of random variables should be distributed approximately as the Galton–Watson branching process $\{Z_\ell\}$ with offspring distribution $\mathsf{Bin}(n, d/n)$. This is certainly the degree distribution of v (it's distributed as $\mathsf{Bin}(n-1, d/n)$, to be more precise); and you might hope that roughly speaking, all the vertices afterwards also have (approximately) that offspring distribution.

This is an approximation because we've already imposed a limitation on the size of the graph (e.g., the number of vertices is fixed).

If you believe this approximation, then the branching process gives you at least a prediction for what the threshold should be. The branching process is *subcritical* (respectively, *supercritical*) if and only if $d < 1$ (respectively, $d > 1$) — recall that *subcritical* means the expectation of the offspring distribution is less than 1, so the expected number of individuals in your population is decaying exponentially fast (and supercritical means it's growing exponentially fast). This has consequences for the probability of extinction, and so on.

If your process is subcritical, you might expect this process to die off very quickly (maybe even after a constant number of steps). And when it's supercritical, you might hope it goes on for a long time, at least until it covers a constant fraction of the graph.

So that's the very handwavy intuition behind this.

> **Student Question.** *Why is it only a constant number of steps by which it dies, rather than* $\log n$*?*
>
> **Answer.** The *expected* size of the component is going to be of constant size, because you're decaying exponentially fast. But you could imagine the reason we have $\log n$ instead of a constant is because we need to union-bound over n vertices, so you need a 'high probability' bound.

§14.4 Exploring a component

Of course, the challenge behind formalizing this is the fact that e.g. if you've already seen εn vertices, then you're no longer distributed as $\mathsf{Bin}(n, \frac{d}{n})$, because you only have $(1-\varepsilon)n$ vertices left. But we'll be able to write down something more precise.

To make this more precise, we're going to more precisely describe how we grow a component; you can think of this as an exploration process. And we're going to look at two pictures — one where we're exploring Erdős–Rényi, and another where we're exploring the GW process — and we're going to couple these two pictures.

Let $\mathcal{C}$ be the connected component containing v. I want to iteratively reveal this component (you can imagine in a BFS sort of manner).

We'll do this as follows: We'll initialize three sets $\mathcal{D}^{\mathrm{ER}}$, $\mathcal{F}^{\mathrm{ER}}$, and $\mathcal{U}^{\mathrm{ER}}$. Think of $\mathcal{D}$ as the vertices we've completely explored (we've completely determined what their neighborhood looks like, and we're 'done' exploring them). Think of $\mathcal{F}$ as the 'frontier' — we haven't finished exploring their neighborhoods, but we've revealed some information. And $\mathcal{U}$ is the 'unexplored' vertices.

What we'll do is, we initialize $\mathcal{D}^{\mathrm{ER}} = \emptyset$, $\mathcal{F}^{\mathrm{ER}} = \{v\}$, and $\mathcal{U}^{\mathrm{ER}} = V \setminus \{e\}$. Then while the frontier $\mathcal{F}^{\mathrm{ER}}$ is nonempty:

- Pick $u \in \mathcal{F}^{\mathrm{ER}}$ arbitrarily.
- Now we're going to reveal all its neighbors which are still unexplored: so for all $w \in \mathcal{U}^{\mathrm{ER}}$, we reveal whether or not $u \sim w$. For all such w, add w to $\mathcal{F}^{\mathrm{ER}}$.
- Remove u from $\mathcal{F}^{\mathrm{ER}}$, and add u to $\mathcal{D}^{\mathrm{ER}}$. (At this point, we've revealed all of u's immediate neighbors — all the neighbors that came before it, and all the ones that come after.)

We'll also describe a parallel of this in the world of branching processes. We initialize a root vertex r. We're now going to initialize essentially the same data structure (except we won't have an unexplored set, because in a branching process there's no limit to what set you have).

So we initialize $\mathcal{D}^{\xi} = \emptyset$ and $\mathcal{F}^{\xi} = \{r\}$ (we use ξ to denote the offspring distribution). Then the algorithm is essentially the same. While $\mathcal{F}^{\xi}$ is nonempty:

- Pick $u \in \mathcal{F}^{\xi}$ arbitrarily.
- Then we're going to generate its offspring — so we sample $Y \sim \xi$ from the offspring distribution, and add Y new vertices to $\mathcal{F}^{\xi}$.
- Then we remove u from $\mathcal{F}^{\xi}$ and add u to $\mathcal{D}^{\xi}$.

Here's roughly what the picture looks like. We're going to look at each step of this while loop as one time-step. As a picture, suppose we have four vertices, with v on the left. Initially, I haven't done anything. We circle v in yellow to denote that it's the current frontier; and x, y, and z are in red to indicate that I haven't touched them yet.

Then on the first step, I'm going to reveal the neighborhood of v — let's say v has neighbors x and y. Then the frontier is going to move to $\mathcal{F}^{\mathrm{ER}} = \{x, y\}$. And now the unexplored set is just $\mathcal{U}^{\mathrm{ER}} = \{z\}$. And I move v into the set of done vertices, so $\mathcal{D}^{\mathrm{ER}} = \{v\}$.

And then this process continues. I look at x and determine its neighborhood. Let's say it has neighbors y and z. Now the frontier is going to be $\mathcal{F}^{\mathrm{ER}} = \{y, z\}$, and the done vertices are $\mathcal{D}^{\mathrm{ER}} = \{v, x\}$. This process continues until nothing more is added.

And there's going to be an analogous picture for the GW process. We start with a root vertex r. Then on the first step, we're going to spawn some children (we draw 3 children); now r is done, and the frontier is all its children.

In the second step, I pick any one of these children and decide how many hanging children *it* has, and now the frontier is all these leaf vertices in the tree I'm building (and the root and the one we just chose are done).

And so on.

> **Student Question.** *In the first step of the iteration, we pick $u \in \mathcal{F}^{\xi}$ and sample Y; isn't the distribution going to change as we perform the steps because there's less undiscovered vertices left?*
>
> **Answer.** This is for the Galton–Watson tree, where there's no fixed number of vertices. The hope is that we're going to couple these two pictures together; but absolutely this coupling is going to break down after a certain point.

So we're going to try to couple these two processes (at least, up to some time, which we'll see in a moment).

§14.5 Some observations

First, here's a couple of observations about this. We'll collect a few facts we need that relate the various pieces of data in this process to quantities we care about.

Definition 14.5. Let $T^{\mathrm{ER}} = \min\{t \geq 0 \mid \mathcal{F}_t^{\mathrm{ER}} = \emptyset\}$, and analogously $T^{\xi} = \min\{t \geq 0 \mid \mathcal{F}_t^{\xi} = \emptyset\}$.

(This is just when the process stops.)

Fact 14.6 — For all $0 \leq t \leq T^{\mathrm{ER}}$:

- We have $|\mathcal{C}_v| \geq \left|\mathcal{D}_t^{\mathrm{ER}}\right| + \left|\mathcal{F}_t^{\mathrm{ER}}\right|$.
- We have $|\mathcal{D}_t^{\mathrm{ER}}| = t$.
- $|\mathcal{F}_t| = \left|\mathcal{F}_{t-1}^{\mathrm{ER}}\right| - 1 + X_t$, where $X_t \sim \mathsf{Bin}(|\mathcal{U}_t|, d/n)$.

(These should all have ER superscripts, but we omit them.)

We use $\mathcal{C}_v$ to denote the connected component of v. The second statement is because we add one vertex at every step; the third is because we mremove one vertex, and add each unexplored vertex with probability d/n (the probability it has an edge to the chosen u).

Fact 14.7 — We have $\mathcal{D}_T = \mathcal{C}_v$, and $T = |\mathcal{C}_v|$.

Fact 14.8 — The sets $\mathcal{D}^{\mathrm{ER}}$, $\mathcal{F}^{\mathrm{ER}}$, and $\mathcal{U}^{\mathrm{ER}}$ partition the vertices.

We also have analogous statements for the Galton–Watson process, which we'll denote by $\{Z_\ell\}$.

Fact 14.9 — For all $0 \leq t \leq T^{\mathrm{GW}}$:

- We have $\sum_{\ell=0}^{\infty} Z_\ell = T^{\xi} \geq |\mathcal{D}_t^{\xi}| + |\mathcal{F}_t^{\xi}|$.
- We have $|\mathcal{D}_t| = t$.
- We have $|\mathcal{F}_t| = |\mathcal{F}_{t-1}| - 1 + Y_t$ for $Y_t \sim \xi$.

§14.6 Comparing the exploration processes

To prove our theorem, we want to compare these two processes. Fix $v \in V$, and let $s_n \leq O(\log n)$. We're going to try to couple these two processes for roughly this number of steps (and that'll be enough). We're actually going to look at *two* branching processes — let $\{Z_\ell^{\mathrm{lower}}\}$ and $\{Z_\ell^{\mathrm{upper}}\}$ be GW branching processes with distributions $\mathsf{Bin}(n - s_n, d/n)$ and $\mathsf{Bin}(n-1, d/n)$, respectively. (Basically what you shoudl imagine is that this is roughly tracking the number of unexplored vertices so far — $n-1$ is always an upper bound, and if we're tracking up to s_n steps, we'll always have a lower bound like $n - s_n$.)

We'll also define $Z^{\mathrm{lower}} = \sum_{\ell=0}^{\infty} Z^{\mathrm{lower}}$ and $Z^{\mathrm{upper}} = \sum_{\ell=0}^{\infty} Z^{\mathrm{upper}}$.

Proposition 14.10

We have $\mathbb{P}[Z^{\mathrm{lower}} \geq s_n] \leq \mathbb{P}[|\mathcal{C}_v| \geq s_n] \leq \mathbb{P}[Z^{\mathrm{upper}} \geq s_n]$.

We're not going to get into the proof of this at the moment; it essentially follows from the fact that $\mathsf{Bin}(n,p) \preceq \mathsf{Bin}(m,p)$ if $n \leq m$ (these aren't binomials, but they're sums of binomials).

Student Question. *How can we be sure that $n - s_n$ is a lower bound?*

Answer. It's definitely not a lower bound with probability 1, but we'll see in the proof that this suffices.

Student Question. *Why do we only go up to $s_n \le O(\log n)$?*

Answer. You don't need it for the upper bound. But for the lower bound, past this, you have too many explored vertices, and things break down.

We'll also need one natural concentration bound:

Theorem 14.11

For any n and any $0 \le p \le 1$, letting $\mu = pn$, we have

$$\mathbb{P}_{Y \sim \mathsf{Bin}(n,p)}[Y \ge t] \le \exp\left((t-\mu) - t\log\frac{t}{\mu}\right) \quad \text{for all } t \ge \mu,$$
$$\mathbb{P}_{Y \sim \mathsf{Bin}(n,p)}[Y \le t] \le \exp\left((t-\mu) - t\log\frac{t}{\mu}\right) \quad \text{for } 0 \le t \le \mu.$$

If we have time, we'll get into the proof of this proposition. But for now we'll use it to prove the theorem.

§14.7 The subcritical case

Let's start with the subcritical case, because that's easier. Now if we set $s_n = O(\log n)$, we want to bound these tail probabilities, which boils down to invoking this theorem — it's a standard concentration argument. We can say

$$\mathbb{P}[|\mathcal{C}_v| \ge s_n] \le \mathbb{P}[Z^{\text{upper}} \ge s_n] \le \mathbb{P}[Z^{\text{upper}} \ge s_n].$$

And every single time I'm removing a vertex from the frontier, so in order for the process to keep going, I need to ensure the frontier is nonempty; in particular, in order for $Z^{\text{upper}} \ge s_n$, I need

$$\sum_{i=0}^{s_n - 1} Y_i^{\text{upper}} \ge s_n$$

(because I'm removing s_n vertices after s_n many steps).

And I know the distribution of each of these Y_i^{upper} random variables — so this is just a binomial distribution with $(n-1)s_n$ trials and success probability d/n, which means this becomes

$$\mathbb{P}_{Y \sim \mathsf{Bin}((n-1)s_n, d/n)}[Y \ge s_n].$$

Here we have $\mu = d \cdot s_n$, where $d < 1$. So we're really looking at an upper tail bound, and we can use the theorem to say that this is at most

$$\exp((1 - o(1))(1 - d + \log d)s_n)$$

(where the coefficient $1 - d + \log d$ is negative). So if I set the constant $s_n = C \log n$ to be sufficiently large, then this is at most $1/n^2$ (for example). And then I can just union bound over all the vertices.

§14.8 The supercritical case

Now let's do the supercritical case — let's assume $d > 1$. For this part, we're going to break it down into two steps (or really, three). There's two key propositions.

The first is just that — notice in the statement of the theorem, we say there's a *unique* component of size linear in n, and all the others have size $\log n$. So in particular, there's a gap — e.g., there's no components of size $\sqrt{n}$. This is what the first proposition says.

Proposition 14.12

Let $\zeta \in (0,1)$ be the unique solution to $1-\zeta = e^{-d\zeta}$. Then for all $b < \zeta$ and all a sufficiently large (as a function of b), there exists δ such that

$$\mathbb{P}[\text{exists } v \text{ with } a\log n \le |\mathcal{C}_v| \le bn] \le O(n^{-\delta}).$$

This specific equation for ζ is maybe not so important for us, but you'll see why it comes up; it's some constant in $(0,1)$ depending on d.

So there are no intermediate components — every component is either of $\log n$ size or linear size. Another way to interpret this is that the moment a component reaches size $a\log n$, it'll achieve 'escape velocity' — it'll grow very quickly to linear size.

In particular, this proposition allows us to reduce showing existence of the giant to showing existence of *some* component which achieves size $a\log n$, and that's what the next proposition says.

For convenience, we'll define one more random variable: let $\mathcal{G}_k = \#\{v \mid |\mathcal{C}_v| \ge k\}$.

The second key proposition is that when $d > 1$, there'll be many vertices (a large constant fraction of n) which participate in some logarithmically sized component.

Proposition 14.13

For every constant $a > 0$, there exists $\varepsilon_n = o(1)$ and $\delta > 0$ such that

$$\mathbb{P}[|\mathcal{G}_{a\log n} - \zeta n| \ge \varepsilon_n n] = O(n^{-\delta}).$$

So in particular, what this says is certainly there must be at least *some* connected component which achieves escape velocity — it reaches this threshold of $a\log n$, and after that it's going to grow all the way to linear size.

As one quick remark, this random variable $\mathcal{G}_k$ is counting the number of vertices that participate in a large component. What it *doesn't* count is the *number* of large components itself. (For example, this does not contradict the uniqueness of the giant component.)

Student Question. *This also implies that there can be at most one giant component, right?*

Answer. Yes, it turns out this also implies the uniqueness of the giant, and we'll see that in a moment. But already by combining these two propositions, you see that at least existence holds — at least there exists a giant.

§14.8.1 First moment for Proposition 14.13

First, we'll very briefly sketch the proof of Proposition 14.13. You can imagine this is going to rely on standard techniques where you want to first compute $\mathbb{E}[\mathcal{G}_k]$, and then bound its variance. In the interest of time, we'll just do the first moment bound; the second moment bound is in the notes.

We're looking at this random variable, which is a sum of a bunch of indicators — the sum over a bunch of vertices of the indicator that its component is large.

But then we can directly use Proposition 14.10 — we can say

$$\mathbb{P}[|\mathcal{C}_v| \ge a\log n] \ge \mathbb{P}[Z^{\text{lower}} \ge s_n].$$

And this is a branching process which is supercritical. In particular, that means this lower bound is always lower-bounded by the probability of non-extinction (if it doesn't go extinct, then this thing goes to ∞). So

this is at least

$$\mathbb{P}[Z^{\text{lower}} \text{ does not go extinct}].$$

Because it's supercritical, this is some positive constant; and if you recall our analysis of branching processes from earlier, you can *compute* that constant as a fixed point of some equation (this is where $1-\zeta = e^{-d\zeta}$ comes from). In particular, this is at least $(1-o(1))\zeta$ (the $o(1)$ is because we have some error in the number of vertices in the binomial distribution).

Student Question. *How do you get the upper bound?*

Answer. The first inequality is not lossy because you can do the same calculation with Z^{upper}. And the non-extinction probability bound is actually not lossy either. So you can actually show this is an equality.

And then you can bound the variance and get a concentration bound.

§14.8.2 Existence and uniqueness of the giant component

Now we'll quickly say how to combine these propositions to get the existence and uniqueness of the giant component.

We want to look at the probability that the conclusion of the theorem holds, and we want to express this in terms of the $\mathcal{G}_k$ random variables.

Claim 14.14 — We have $\mathbb{P}[\text{conclusion holds}] \geq \mathbb{P}[\mathcal{G}_{a\log n} = \mathcal{G}_{bn} \text{ and } \mathcal{G}_{a\log n} \in (1\pm o(1))\zeta n]$.

What this first part is saying is that all the vertices that participate in a component of size $a\log n$ really participate in a component of size (at least) bn — so the first statement says that all vertices with component size at least $a\log n$ are really in a giant component, and the second part really gets the size of the giant.

Here we're fixing b with $\zeta/2 < b < \zeta$. These two claims already imply existence. For uniqueness, the idea is suppose there were two giant components. If you had two giant components, then both of them had better have size at least bn. That means you have at least $2bn$ vertices counted in $\mathcal{G}_{a\log n}$; and this is way larger than ζn.

So actually combining these two events not only gives you existence, it also gives you uniqueness.

And now if you want to lower-bound this by $1-o(1)$, you just take the complement and do a union bound and so on.

§14.8.3 Proof sketch of Proposition 14.12

In the last ten minutes, we'll sketch the proof of the first key proposition.

Lemma 14.15

Let $Y_t \sim \mathsf{Bin}(n-1, 1-(1-d/n)^t)$. Then $\mathbb{P}[|\mathcal{C}_v| = t] \leq \mathbb{P}[Y_t = t]$.

This is similar in spirit to the key proposition above; but now rather than changing the number of trials, I'm changing the success probability for each individual trial.

We'll assume this first, and then if we have time we'll prove this lemma.

Now I want to look at $\mathbb{P}[a\log n \leq |\mathcal{C}_v| \leq bn]$. We can just sum over all possible sizes, so this is bounded

$$\sum_{t=a\log n}^{bn} \mathbb{P}[Y_t = t].$$

(We can even just use $Y_t \le t$.) And now we can use the tail bounds — by concentration, this is going to be upper-bounded by

$$\sum_{t=a\log n}^{bn} \exp\left(-t\left(\frac{\mu_t}{t} - 1 - \log\frac{\mu_t}{t}\right)\right),$$

where $\mu_t = (n-1)(1-(1-d/n)^t)$ is the mean of this binomial distribution. Notably we'll have $\mu_t \ge t$ if and only if $t \le (1-o(1))\zeta n$ — this is another place where the constant ζ comes up.

So now if we assume $b < \zeta$, then we get a little gap in this ratio (we'll have that μ_t/t is some constant strictly larger than 1); so this is going to be upper-bounded by

$$\sum_{t=a\log n}^{bn} \exp(-\Omega(t)).$$

And then the dominant term is going to be the first term, where you set $t = a\log n$ — if the constant is α, then we get a bound something like $n^{-\alpha a}$.

Now we use the usual bounds — we union-bound over all vertices and set a sufficiently large (relative to α), and we'll be good to go.

§14.8.4 Proof of Lemma 14.15

Now in the last five minutes we'll briefly sketch this lemma. The idea is very similar to how you would prove Proposition 14.10, and this is where you use stochastic domination.

First, where does this quantity come from? The idea is that if I look at $n-1-Y_t$, now this is distributed as $\mathsf{Bin}(n-1,(1-d/n)^t)$. And we want to say this is roughly tracking the number of *unexplored* vertices, i.e., $|\mathcal{U}_t|$.

To be more formal, we have the following: If I look at the number of unexplored vertices, we have

$$|\mathcal{U}_t| = |\mathcal{U}_{t-1}| - X_t \quad \text{for } X_t \sim \mathsf{Bin}(|\mathcal{U}_t|, p_n)$$

for all $t < T^{\mathrm{ER}}$, then this is literally tracking the number of unexplored vertices. And the idea is that you can couple this to basically the same process where you set $U_t = U_{t-1} - X_t$ for all t (not just up to your stopping time) and show some sort of stochastic domination.

(This is a bit rushed, but the more precise argument is in the notes.)

§15 March 31, 2025 — Contiguous families of distributions

Today's lecture will be about contiguity, a way of comparing classes of distributions with each other. A major theme in this class is often we have a large system, and we'd like to say that with very high probability, some property of the system holds (as its size goes to ∞). There are many examples where proving this statement directly for the system of interest can be very challenging. But contiguity will give us a tool to reduce proving that statement to proving it for a *simpler* class of distributions.

§15.1 Contiguity

First let's define what we mean by contiguity.

Definition 15.1. Let $\vec{\mu} = \{\mu_n\}_{n\in\mathbb{N}}$ and $\vec{\nu} = \{\nu_n\}_{n\in\mathbb{N}}$ be two sequences of distributions (where μ_n and ν_n live on the same state space Ω_n, where $|\Omega_n| \to \infty$ as $n \to \infty$). We say $\vec{\nu}$ is *contiguous* with respect to $\vec{\mu}$ if for every sequence of events $\{A_n \subseteq \Omega\}_{n\in\mathbb{N}}$, we have that if $\mu_n(A_n) \to 0$, then $\nu_n(A_n) \to 0$.

You can imagine $\{\nu_n\}$ is some natural class of distributions you want to study (like Erdős–Rényi, or as we'll see later in this lecture, a uniform random d-regular graph). And $\{\mu_n\}$ will be a different class of models which is easier to analyze. So you'll try to first show your desired events hold with high probability under this simpler model μ_n; and if you can show the two models are contiguous, you immediately get the same claim for ν_n for free.

We'll do a quick example; this is actually a very generic example that we'll use later on.

Example 15.2

Suppose we have a sequence of events $\{E_n \subseteq \Omega_n\}_{n\in\mathbb{N}}$ such that $\mu_n(E_n) \geq c$ for all n (where $c > 0$ is some constant). Then the sequence of conditional distributions $\{\mu_n \mid E_n\}_{n\in\mathbb{N}}$ is contiguous with respect to $\{\mu_n\}_{n\in\mathbb{N}}$.

So this is saying that if I have my domain Ω_n (which is everything), and I condition on a large subset of the domain (with mass at least some constant c), then if I look at any sequence of events E_n which hold with vanishing probability, it'll also hold with vanishing probability when I restrict my attention to this constant fraction of the state space. If I have this little A_n whose mass goes to 0 under the entire distribution μ_n, it'll also have mass going to 0 when I condition on E_n.

So you always get to sort of condition on large probability events, and you always get contiguous classes of distribution.

Later in this lecture, we'll see a nice application of this example.

Here's a lemma, which gives another way to certify contiguity between two distributions.

Definition 15.3. We define the *χ^2-divergence* between two distributions μ and ν as

$$\chi^2(\nu \parallel \mu) = \mathbb{E}_{x\sim\mu}\left[\left(1 - \frac{\nu(x)}{\mu(x)}\right)^2\right].$$

This looks very similar to the total variation distance, but it's kind of a quadratic version — the total variation distance would be what we'd get if we replaced the square with an absolute value.

Lemma 15.4

If $\chi^2(\nu_n \parallel \mu_n) \leq O(1)$, then $\{\nu_n\}$ is contiguous with respect to $\{\mu_n\}$.

This is a connection between contiguity and something like the second moment method — here we're looking at a second moment-type quantity. This is also often how contiguity results are proven.

Remark 15.5. You can sort of refine all these definitions and notions to include rates of convergence or decay and so on, but for simplicity we won't look at that.

Student Question. *How do you intuitively think of the χ^2-divergence?*

Answer. You can sort of think of it as the variance — you can rewrite it as

$$\mathrm{Var}_\mu\left(\frac{d\nu}{d\mu}\right).$$

So if I compare how different μ and ν are entry-wise that gives me some function, and I look at the variance of that.

Proof. Similarly to the total variation distance, χ^2-divergence also admits a very useful variational characterization:

Fact 15.6 — We can write

$$\chi^2(\nu \,\|\, \mu) = \sup_{f:\Omega\to\mathbb{R}} \frac{(\mathbb{E}_\mu[f] - \mathbb{E}_\nu[f])^2}{\mathrm{Var}_\mu[f]}.$$

We won't prove this, but the proof is actually in the notes of Lecture 3 (when we looked at statistical inference and used the second moment method to study the broadcast process on the tree).

Now if we want to show one sequence is contiguous to another, we should just plug in whatever low-probability events we're looking at — suppose we have a sequence of events $\{A_n \subseteq \Omega_n\}$ with $\mu_n(A_n) \to 0$. Then plugging in $f = \mathbf{1}_{A_n}$ and using the given constant upper bound, we get that

$$O(1) \geq \frac{(\mathbb{P}_{\mu_n}[A_n] - \mathbb{P}_{\nu_n}[A_n])^2}{\mathbb{P}_{\mu_n}[A_n](1 - \mathbb{P}_{\mu_n}[A_n])}.$$

Now the claim si that this inequality can only hold if $\mathbb{P}_{\nu_n}[A_n] \to 0$ as well — the denominator is going to 0, so that should be blowing up this whole ratio, and the only way it *doesn't* blow up is if the numerator is *also* going to 0, which requires $\mathbb{P}_{\nu_n}[A_n]$ to also be going to 0. □

Student Question. *So the function we take is just the indicator function?*

Answer. Yes, the indicator function of your events.

§15.2 Average-case computational complexity

Now we'll discuss, or at least give handwavy sketches of, how this is used to study average-case computational problems. Here, for the next 20 or so minutes, there will be no theorems or real proofs; this is just to give us a flavor of why the concept is useful.

Imagine we have some classic computational problem — e.g., finding a solution to a SAT formula, or finding large independent sets. And now you want to study the complexity of this problem on an *average-case* instance. (We saw some of this in previous lectures, like finding independent sets in sparse Erdős–Rényi, but here we want to discuss a more systematic theory for this type of things.)

For a concrete example for this lecture, we'll look at *colorings*.

Example 15.7

Suppose we want to find q-colorings of the sparse Erdős–Rényi $\mathcal{G}(n, d/n)$.

This means we want to assign colors $1, \ldots, q$ to the vertices of the graph so that no edge is monochromatic. In the worst case, this is known to be a very hard problem (e.g., finding the smallest number of colors required is hard to even approximate). But in an average-case setting, the following is known. First, similarly to

the case of independent sets, if I run a very simple greedy algorithm, I can do pretty well: Simple greedy heuristics find a q-coloring with high probability whenever $d \leq q \log q$ (up to $o(1)$-factors). Kind of similarly to the case for independent sets, we also have the right threshold for when a q-coloring exists at all, and there's going to be a slight gap — for $d \leq 2q \log q$, with high probability a q-coloring *exists*. This factor of 2 is very similar to the factor of 2 you see for finding large independent sets (in sparse Erdős–Rényi, we saw the greedy algorithm finds independent sets of size $n(\log d)/d$, but there exist independent sets of size $2n(\log d)/d$).

The interesting thing is this doesn't seem to just be an artifact of us looking at a greedy algorithm; *any* algorithm we've tried coming up with that we can analyze gets stuck at this threshold. So one might wonder if there's something more inherent to why there is this gap. One way you can *try* to understand this is to look at the following: We can investigate what the space of all possible q-colorings 'looks' like. Suppose you have d which is larger than the *algorithmic threshold* $q \log q$ (we'll call this the algorithmic threshold, at least for current technology), but still smaller than the *satisfiability threshold* (of $2q \log q$). This means there exists a coloring, but somehow it's difficult to *find* one.

Question 15.8. What does the space of colorings 'look like'?

This was investigated in a long line of works. There's a nice paper by Achlioptas–Coja–Oghlan which showed the following: They showed that when $q \log q < d \leq 2q \log q$, the space of colorings looks *extremely* disconnected, a phenomenon they call *shattering*. This means if we look at the proper q-colorings (drawn in a box representing all possible colorings), there'll be a bunch of small clusters (whose diameter with respect to the Hamming distance is $o(n)$), and the distance between these clusters is very large ($\Omega(n)$ in Hamming distance). And there will be exponentially many of these clusters.

One can imagine this is a reason why algorithms are failing. Most algorithms one would try to design are 'local' in some sense, but here you'd have to traverse a huge gap of colorings which are not proper at all to find one of these tiny islands of good colorings.

And this 'shattering' actually has been turned into a rigorous algorithmic barrier. There's been lots of recent work on this; one citation is Garmanik–Sudan.

§15.3 Typical proper colorings

Actually, the landscape is much richer: You can ask what the solution space looks like for various values of d, not just between the algorithmic and satisfiability thresholds.

Imagine we draw a plot with d on the x-axis. We're going to have various phase transitions in what the solution space looks like. There's a first threshold below which everything is contained in one giant cluster (you can think of this sort of as 'connectivity' in Erdős–Rényi). Then when we go above some threshold, you'll get one large cluster, and maybe a bunch of smaller clusters which are not connected to it (under local perturbations of the coloring). (You can think of this as a 'giant component.') Then you have *shattering*, where you have a bunch of tiny clusters far away from each other. And beyond this, there's just nothing (you're beyond the satisfiability threshold).

And you can actually make these kinds of predictions using various heuristics from statistical physics. But one major endeavor is to actually formalize some of these predictions. And that's what some of these works are able to do. They crucially take advantage of this notion of contiguity.

Here's how one might try to establish something like this (we'll be extremely non-rigorous in this description).

Imagine you want to understand what a 'typical' coloring looks like. So imagine you do the following process:

(1) Sample a graph $G \sim \mathcal{G}(n, d/n)$.

(2) Sample a uniformly random proper coloring $\chi : V \to [q]$.

(3) Output your sample (G, χ).

You can imagine if you do this, you can compute some statistics — take a test function and estimate its expectation — to get a sense of what such a coloring 'looks like.'

This is very natural, but it's very hard — sampling colorings is a notoriously difficult problem. Especially if you're not in the connectivity phase of this diagram, for instance you're not going to have Markov chains — lots of natural sampling strategies to implement this are not going to work.

But there's a very nice, cute strategy for getting access to this distribution, which goes by contiguity. We're going to consider a slightly different distribution, called the *planted trick*. The process above outputs a pair of a graph and coloring. Here we're also going to do that, but we're going to generate it slightly differently.

(1) First sample a coloring $\chi^{\mathsf{PL}} : V \to [q]$ uniformly at random (with no constraints whatsoever — this is very easy to do).

(2) Sample the 'constraints' G^{PL} — i.e., for every bichromatic pair of vertices $uv \in \binom{V}{2}$, add an edge independently with probability $p(d, q)$, where $p(d, q)$ is some expression chosen so that the average degree is d (similarly to Erdős–Rényi, but we don't try adding edges between monochromatic pairs of vertices).

(3) Output $(G^{\mathsf{PL}}, \chi^{\mathsf{PL}})$.

So you're sampling a graph ensuring that it has at least one proper coloring, namely the one that you've forced into it.

Student Question. *On the left, if the graph has no proper colorings, what do you do?*

Answer. We're below the satisfiability threshold, so with high probability there'll be a proper coloring. So we can condition on being q-colorable (which is a valid thing to do, in that it creates a contiguous class of distributions).

Basically, what we want to do is compare these two processes. Why would we want to do this? Basically, the hope is that the planted distributions are easier to analyze.

First, let's convince ourselves of that fact. We claim that at least if you want to estimate various statistics (e.g., computing the expectation of some function), doing so in the planted world is much easier than in the original world that you cared about.

Example 15.9

Suppose you have some function $f : [q]^V \to \mathbb{R}$ (a function on the space of all colorings — e.g., you're counting the number of times blue occurs), and you want to compute $\mathbb{E}_G[\mathbb{E}_\chi[f(\chi)]]$.

We claim that if you push through all the calculations, doing it in the planted world is much, much easier. In the planted world, it turns out that this literally equals

$$\mathbb{E}_{G^{\mathsf{PL}}}[\mathbb{E}_{\chi^{\mathsf{PL}}}[f(\chi)]] = \frac{\mathbb{E}_{G\sim\mathcal{G}(n,d/n)}[\sum_\chi f(\chi)]}{\mathbb{E}_{G\sim\mathcal{G}(n,d/n)}[\#\text{proper } q\text{-colorings}]}.$$

Meanwhile, in the original world, we'd have

$$\mathbb{E}_{G\sim\mathcal{G}(n,d/n)}\mathbb{E}_\chi[f(\chi)] = \mathbb{E}_{G\sim\mathcal{G}(n,d/n)}\left[\frac{\sum_\chi f(\chi)}{\#\text{proper } q\text{-colorings}}\right].$$

(Both sums are over proper q-colorings χ.) Assuming you believe these calculations, the first is much easier to compute with — ultimately we want to compute the expectation of a ratio of two things, but it's much

easier to compute the expectations of the numerators and denominators individually. These two things are definitely not equal, but the first is easier to compute.

So that tells us the planted world is much easier to work with. And the hope is that if we have contiguity, we can first prove whatever we want in the planted world (e.g., statements like shattering), and then lift those results to the original model we cared about.

Goal 15.10. Get contiguity, and then lift the analysis of the planted model to the original.

It turns out that if you want to prove this contiguity, you use something like the χ^2-divergence. Then this boils down to understanding how well the number of proper q-colorings of a randomly chosen graph concentrates.

Student Question. *In the planted model, this isn't a conditional distribution of Erdős–Rényi given that it admits that coloring, right? It's slightly different?*

Answer. Maybe it's easier not to work with Erdős–Rényi, but the random graph where I fix the number of edges (so that I don't have this business with changing the probability). But it's supposed to be a conditional distribution.

So the picture is you want to prove something about the geometry of the space of solutions in the first picture; and the hope is that it's easier to prove it in the second picture, and if you can prove a contiguity result then you can get everything you want for free (i.e., you can get a sort of transference).

This is a very high-level sketch — we won't go into precise results here. But we will illustrate an easier application of this rigorously.

§15.4 Random d-regular graphs and the configuration model

We'll look at the following random graph model: We want to look at the uniform distribution over all simple d-regular graphs (graphs where every vertex has exactly d neighbors; 'simple' means there are no parallel edges or self-loops).

The space of all simple d-regular graphs — maybe at first glance, it might seem that the regularity constraint is somewhat rigid. So what we're going to do is we'll sort of study another class of distributions, where we now relax the simplicity constraint. That class of distributions is going to be easier to analyze, and then we'll show contiguity between those two classes of distributions.

First, to get us started, one way you could try to attack this is:

Example 15.11

Let's consider the case $d = 1$.

This means we're looking at a uniformly random perfect matching on n vertices (we'll always assume dn is even, so that there exist n-vertex d-regular graphs).

In the case $d = 1$, we're just looking for a uniformly random perfect matching. Here, we have very explicit formulas — I can generate a random perfect matching by, for instance, partitioning my vertex set into two pieces of size $n/2$, and then drawing a random permutation of the vertices on the top, and then matching them. So one sampling process is:

- Partition the vertices into two sets of size $n/2$, uniformly at random.
- Pick a uniformly random permutation of the vertices.
- Match them up.

This is a very simple nice way to sample a uniformly random perfect matching on all n vertices.

We might hope that by an analogous process for larger values of d, we could get a uniformly random d-regular graph. This is the basic idea behind the process called the *configuration model*. (It's called this because some people call perfect matchings 'configurations.')

Definition 15.12. The *configuration model* (for n-vertex d-regular graphs) is defined as follows:

(1) Initialize dn vertices, partitioned into n 'clouds' of d vertices each.

(2) Sample a uniformly random perfect matching on these dn vertices.

(3) Contract the clouds.

Example 15.13

Suppose $d = 3$ and $n = 4$.

Then we'll have four clouds of vertices, each cloud having exactly 3. Then we'll sample a uniformly random perfect matching on these vertices. So maybe I'll have a couple of edges inside the clouds, and a couple of edges between them; so all vertices are paired up and they have degree 1.

And now I'm going to contract these clouds, meaning I shrink each cloud into a single vertex. If I have an edge inside the cloud, I'm going to add a self-loop; and if I have an edge across two clouds, I add an edge between them. In general, this can create multiple edges (if there are multiple edges between two clouds) and self-loops; but every vertex is going to have degree exactly d.

Theorem 15.14

For every $d \in \mathbb{N}$, as $n \to \infty$ we have

$$\mathbb{P}_{\text{config model}}[G \text{ is simple}] \to \exp\left(\frac{1-d^2}{4}\right).$$

In particular, this means this probability is bounded below by a constant. So if I condition on G being a simple graph, I'm conditioning on a constant-probability event. We need to say one more thing:

Lemma 15.15

If we condition on the graph G (produced by the configuration model) being simple, then it is distributed as a uniformly random (simple) d-regular graph.

With the above theorem, that implies the distribution of a uniformly random (simple) d-regular graph is contiguous to the configuration model; and then you can just work with the configuration model for all your calculations.

Corollary 15.16

The uniformly random simple d-regular graph model is contiguous to the configuration model.

From this process, you can also count the total number of d-regular multigraphs; so Theorem 15.14 also gives you an asymptotic formula for the number of d-regular simple graphs.

§15.5 Cycles in the configuration model

Both of these results are special cases of more general results. Theorem 15.14 is actually a special case of a more general result about the number of cycles in the configuration model.

Theorem 15.17

For each $k \geq 1$, let X_{kn} be the number of length-k cycles in a sample from the configuration model. Define $\lambda_k = \frac{(d-1)^k}{2k}$. Then for every fixed constant $\ell \geq 1$, we have

$$\mathsf{Law}(X_{1n}, \dots, X_{\ell n}) \to \mathsf{Poi}(\lambda_1) \otimes \cdots \otimes \mathsf{Poi}(\lambda_\ell)$$

in distribution as $n \to \infty$.

In other words, if I look at the number of cycles of lengths $1, \dots, \ell$, then this converges to a bunch of independent Poissons ($\otimes$ denotes that the Poissons are independent). So asymptotically, all the different cycle counts are independent and jointly Poisson.

And one more proposition — one of the other very fundamental properties of random d-regular graphs is that they are 'locally tree-like.'

Proposition 15.18

Let H be any fixed-size graph with r vertices and s edges, where $s > r$. Then with probability $1 - o(1)$, a sample from the configuration model contains no copy of H as a subgraph.

Here $s > r$ means that H is more than a tree or forest by at least two edges. So all local neighborhoods must be very sparse — either a tree or a tree plus one edge.

Remark 15.19. One quick comment is that Theorem 15.17 implies Theorem 15.14 — you can write

$$\mathbb{P}[G \text{ is simple}] = \mathbb{P}[X_{1n} = 0 \text{ and } X_{2n} = 0].$$

And now because X_{1n} and X_{2n} are asymptotically independent and they're Poisson with some very explicit mean, you can just calculate this (in the limit) — as $n \to \infty$, this tends to

$$\mathbb{P}_{Z_1 \sim \mathsf{Poi}(\lambda_1)}[Z_1 = 0] \cdot \mathbb{P}_{Z_2 \sim \mathsf{Poi}(\lambda_2)}[Z_2 = 0],$$

which by explicit calculation is the expression we stated.

So now we'll sketch the proofs of Theorem 15.17 and Proposition 15.18, and we'll be done. This will only be a sketch because the full proof requires quite a bit of calculations, though it's still reasonably tractable; we'll just do some of the most illustrative ones.

§15.5.1 Proof sketch of Theorem 15.17

Kind of the basic idea if you want to prove a convergence theorem like this is to show that the joint moments of your random variables X_{kn} converge to the joint moments of your independent Poissons. For convenience, we'll work with a slight change of variables — for $X \in \mathbb{R}$ and $r \in \mathbb{N}$, define

$$(X)_r = X \cdot (X-1) \cdots (X-r+1)$$

(this is called the *falling factorial*; it's some strange quantity you can define for any real number). We're going to be looking at the expectations of these guys.

Example 15.20

We have $\mathbb{E}_{X \sim \mathsf{Poi}(\lambda)}[(X)_r] = \lambda^r$ for all $r \in \mathbb{N}$.

What we're going to show is that if we look at the analogous thing for the X_{kn} — the factorial moments of these cycle counts — they converge to these very nice values (the corresponding powers of λ).

We're going to use the following tool (a very useful tool in general if you want to prove a sequence of random variables converges to Poissons):

Theorem 15.21 (Multivariate Brun's Sieve)

Suppose there exist $\lambda_1, \ldots, \lambda_\ell$ such that

$$\lim_{n \to \infty} \mathbb{E}[(X_{1n})_{x_1} \cdots (X_{\ell n})_{x_\ell}] = \prod_{k=1}^{\ell} \lambda_k^{x_k}$$

for all $x_1, \ldots, x_\ell \in \mathbb{N}$. Then $(X_{1n}, \ldots, X_{\ell n}) \to \bigotimes_{k=1}^{\ell} \mathsf{Poi}(\lambda_k)$.

(The notation in the end means we're looking at independent $\mathsf{Poi}(\lambda_k)$ distributions.)

You can basically think of this as convergence of moments (or rather, factorial moments) — if all the factorial moments converge to the right thing (what we would have for independent Poissons), then we get convergence in distribution.

We're not going to prove the multivariate Brun's sieve; maybe it's believable that if the moments converge then you're going to get convergence in distribution. But we're going to use it — we'll at least sketch how you can go about computing these moments.

So here's a handwavy idea for how the argument's going to go. First you can show that if I look at the factorial moment for just a single one of these random variables X_{kn}, then it converges to the right Poisson. Then you have to handle the fact that these random variables are also asymptotically independent of each other. Basically the idea behind that is to think about how correlations can arise. Imagine that I break down these random variables as a sum of indicators (for each subset of vertices of a given size, I look at the indicator that it forms a cycle). Then you only get correlations between these random variables if these cycles intersect. And kind of the point is that because I'm only looking at constant-sized cycles, very few of them actually intersect (relative to the total number of all cycles). So the dependencies decay as $n \to \infty$, since they are 'caused' by intersections between constant-sized cycles. Formalizing this is a lot of hairy calculations, but this is basically what you want to say.

And now we just have to show that each individual random variable converges to the right Poisson distribution. For that, we'll sketch the simplest possible case for convenience, where $x_k = 1$:

Claim 15.22 — We have $\mathbb{E}[X_{kn}] \to \lambda_k$.

The same kind of calculations will also lead you to prove Proposition 15.18 (you can prove this using something like the first moment method).

Proof. Suppose I'm looking at a length-3 cycle in the graph G. This length-3 cycle came from a structure that looked like the following: I had a cloud for each of these three vertices. And I must have picked two vertices in the top two sets and matched them up, two vertices in the top-left and bottom sets and matched them up, and two vertices between the top-right and bottom sets and matched them up. What we'll do is count the number of such configurations, and also the probability of seeing any one of them.

So we'll call this kind of structure a 'cycle' of length k in the cloud graph (we put 'cycle' in quotes because it's not literally a cycle, but it will form a cycle once you've contracted the vertices in each cloud).

Then we can write

$$\mathbb{E}[X_{kn}] = \#\{\text{cycles of length-}k \text{ in the cloud graph}\} \cdot \mathbb{P}[\text{seeing a particular length-}k \text{ cycle}].$$

And now we're just going to compute each one of these.

For the first term, let's count ordered cycles first. Imagine I first fix $i_1, \dots, i_k \in [n]$ — the identities of the clouds that I want to look at. Then I'm going to get a contribution of $d^k(d-1)^k$ — for each cloud, I'm looking at the number of ways of forming a pairing of vertices that leads to a cycle on these clouds. So I have d choices for the first vertex in the first part, d choices for the vertex it connects to; then I only have $d-1$ choices for the other vertex in that cloud, then d choices for the vertex that connects to; then $d-1$ choices for the other vertex in that cloud, and so on. So each cloud contributes a factor of $d(d-1)$.

Student Question. *Are we only counting simple cycles?*

Answer. Yes (X_{kn} is the number of simple cycles — meaning we don't repeat vertices).

Now for the remaining count, now I have to choose the clouds that I'm going to form my cycle with. So I'm going to get

$$\frac{1}{2k} n(n-1)\cdots(n-k+1) \cdot d^k(d-1)^k$$

where the $1/2k$ comes from the fact that my cycle doesn't change if I change the choice of the starting vertex in the cycle, and it also doesn't change if I change the order in which I traverse my cycle. So that's the answer for the first term (the number of cycles of length-k in the cloud graph).

For the second term (the probability of seeing a specific cycle), I've already determined some perfect matching on $2k$ vertices, and I want to compare that with the *total* number of perfect matchings that are there. So this is

$$\frac{\#\{\text{perfect matchings on } dn - 2k \text{ vertices}\}}{\#\{\text{perfect matchings on } dn \text{ vertices}\}}$$

(since to get this cycle, we've already fixed what happens with these $2k$ vertices, so we only get to match the remaining $dn - 2k$). And these have relatively standard formulas based on factorials. So you can compute what these are and take the product, and you'll find that they converge to the right number. (The calculations are fleshed out in greater detail in the notes.) □

§16 April 2, 2025 — Gibbs distributions of graphical models

Today we'll introduce a new large class of distributions, and sort of go far beyond the settings where our random variables are independent. Lots of the distributions we'd looked at so far have lots of joint

independence (e.g., Erdős–Rényi random graphs). Now we'll define distributions that go far beyond this and are useful for modelling many situations.

These will again be distributions built on top of graphs (or more generally, hypergraphs); we'll be able to give a compact description of these distributions. But they'll exhibit many highly complex and global phenomena.

§16.1 Definitions

For simplicity, in this lecture we'll focus on a special class of graphical models, which are discrete and undirected (but there are also continuous and directed analogs of these models).

Definition 16.1. A (pairwise) *Markov random field* (or a *spin system*) is specified by two pieces of data: a graph $G = (V, E)$, and an *interaction matrix* $B \in \mathbb{R}_{\geq 0}^{q \times q}$, which is a $q \times q$ nonnegative matrix (where $q \geq 2$). The corresponding *Gibbs distribution* is the distribution

$$\mu(\sigma) \propto \prod_{uv \in E} B(\sigma(u), \sigma(v)) \quad \text{for } \sigma \in [q]^V.$$

Think of $q \geq 2$ here as the number of 'colors' available to assign to the vertices of the graph; we'll refer to elements of $\{1, \dots, q\}$ as either colors or spins.

We'll see many examples in a moment.

Definition 16.2. The *partition function* Z is the normalizing constant for the Gibbs measure, i.e.,

$$Z = \sum_{\sigma \in [q]^V} \prod_{uv \in E} B(\sigma(u), \sigma(v)).$$

Historically, this class of distributions was first considered in a statistical physics context: Imagine G is some discretization of space, e.g., the integer lattice, and imagine that vertices of this lattice are locations of particles in some material. For simplicity, assume particles only directly interact with neighboring particles; you can think of the interaction matrix B as specifying your laws of physics (for how particles interact). You want to understand what kinds of global properties of the entire system emerge out of just specifying these local interactions.

So these are really fundamental in statistical mechanics. And there's recent examples to machine learning and other things — graphical models provide a really flexible and intuitive framework for modelling dependencies between random variables, so you can study things like causality and so on.

§16.2 Some examples

We'll now see some examples (several of which we've already looked at, but now we'll put them all under one unifying framework).

Example 16.3 (Ising model)

Fix $\beta \in \mathbb{R}$. Suppose $q = 2$, and let

$$B = \begin{bmatrix} e^{\beta} & 1 \\ 1 & e^{\beta} \end{bmatrix}.$$

What this says is that if I have an edge and the endpoints of the edge receive the same color, then I'm going to get an e^β in the probability mass; and if I see a disagreement, I'm just going to get a 1. If you look at this distribution, one way to write it is that

$$\mu(\sigma) \propto \exp\left(\frac{\beta}{2} \cdot \sigma^\intercal A_G \sigma\right)$$

(where A_G is the adjacency matrix of the graph, and we imagine our spin assignments as $\sigma \in \{\pm 1\}^V$). Equivalently, a more combinatorial way of phrasing this is that we can think of it as a distribution of *subsets* of vertices in the graph, where

$$\mu(S) \propto \exp(-\beta \cdot |E(S, S^{\mathsf{c}})|).$$

(Think of S as the set of vertices that receive $+1$, for instance.)

This distribution has a name; it's the famous *Ising model* of magnetism from statistical physics. Here, what you imagine is that I have a graph (let's imagine it's the lattice again), and this lattice is sort of representing a big block of metal. And the vertices of this lattice are your metal atoms. Each individual vertex can be imagined like a small magnet pointing north or south. And we can wonder whether the entire chunk of metal acts like a global magnet, which will happen if all our little particles align.

We see in this distribution that if I have a large positive β, then the distribution is putting larger probability mass on configurations which have few cut edges — you want to maximize *agreements* across edges of this graph.

We can view the parameter β as an 'inverse temperature' — its magnitude tells you how much you prefer configurations with more or less disagreements. If $\beta = 0$, then this distribution is just uniform over all configurations, and there's no interactions whatsoever between the particles (you have joint independence). Meanwhile, if β is really large, e.g., $\beta = \infty$, then you'll only be looking at configurations where everyone agrees.

Student Question. *Is β positive?*

Answer. You can let β be any real number. If $\beta > 0$ you're encouraging agreements; if $\beta < 0$ you're encouraging disagreements. For example, if you send $\beta \to -\infty$, then the distribution is going to concentrate mass on the maximum cuts of the graph.

We can also generalize this to a larger number of colors:

Example 16.4

For $q \geq 2$, we can take B to be the matrix consisting of e^β on the diagonals, and 1's everywhere else.

Again if $\beta > 0$ we'll be encouraging neighboring vertices to receive the same color, and if $\beta < 0$ we'll be discouraging them to receive the same color. In particular, if $\beta = -\infty$, then μ will be uniform over proper q-colorings of G.

Let's do one more example:

Example 16.5 (Hardcore gas model)

Let $q = 2$, and let

$$B = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

The way you should think about this is that we have one special color, where we're not allowed to have both endpoints of an edge be assigned that special color. But otherwise, everything else is the same. In

particular, the corresponding distribution μ is going to be uniform over *independent sets* of the graph, where we use this special color to mark which vertices we're including in the independent set. This is a model sometimes known as the *hardcore gas model*. The picture for why physicists might look at this is imagine you have a box, and inside the box you have a bunch of gas particles. You could imagine these gas particles have a little bit of volume. Then gas particles might collide, but you want to ensure they're not allowed to actually overlap with each other. That's essentially modelled by this independent set constraint. (This is actually a discretization of a continuous model called the *hard spheres model*. But you can understand it purely combinatorially with independent sets in a graph.)

For this model, you typically generalize it slightly to allow varying the density of the independent sets:

Example 16.6

More generally, for some fixed parameter $\lambda \geq 0$, we can define a distribution by

$$\mu(I) \propto \lambda^{|I|} \quad \text{for independent sets } I \subseteq V.$$

In this case, the corresponding partition function $Z_G(\lambda) = \sum_I \lambda^{|I|}$ is a very nice and well-studied polynomial in combinatorics, called the *independence polynomial*. And actually, the properties of this partition function as a polynomial turn out to be intimately related to e.g. the Lovász local lemma that we saw in the third or fourth lecture (though we won't discuss that).

§16.3 Marginal and conditional distributions

In this lecture, we're going to not really state any theorems or proofs; it'll be more about setting up a language to talk about these distributions, and sort of preparing for the following lectures.

If I have a distribution and I want to understand its properties, I can do two things to it (I can do many, but two are very natural). One is that I can marginalize out many variables and focus my attention on a few vertices. Or I could condition on the colors that several vertices receive.

Let's make more precise what we mean by this:

Definition 16.7. For $\Lambda \subseteq V$, we define the *marginal distribution* on partial colorings $[q]^\Lambda$ by

$$\mu_\Lambda(\eta) = \sum_{\substack{\sigma \in [q]^V \\ \sigma|_\Lambda = \eta}} \mu(\sigma).$$

So we're taking a partial coloring η (coloring only the vertices in Λ); and the marginal is of course given by summing over all completions σ of η (of their Gibbs measure).

Definition 16.8. For $\Lambda \subseteq V$ and $\eta : \Lambda \to [q]$, we define the *conditional distribution* μ^η by

$$\mu^\eta(\sigma) = \frac{\mu(\sigma)}{\mu_\Lambda(\eta)} \quad \text{for all } \sigma \in [q]^V \text{ such that } \sigma|_\Lambda = \eta.$$

(For this to be well-defined, we of course need to assume that there exists some full configuration σ that is consistent with η.) We typically refer to η as a *pinning* — I'm forcing the colors assigned to some subset of vertices Λ.

Here's a nice property of these kinds of distributions. They're definitely not jointly independent in general, but they do satisfy some 'weak' kind of independence.

Lemma 16.9

Let μ be a Gibbs distribution of a spin system. Then for every partition of V into three sets $V = R \sqcup S \sqcup T$ such that S is a *separator* between R and T, and every pinning $\eta : S \to [q]$, we have

$$\mu^\eta = \mu_R^\eta \otimes \mu_T^\eta.$$

What it means for S to be a separator is that for every vertex in R and every vertex in T, if I look at any path between these two vertices, then it must go through some vertex in S along the way. (In other words, for every $u \in R$ and $v \in T$ and every path γ from u to v, we have $\gamma \cap S \neq \emptyset$ — any path from u to v must go through S.)

And what this says is if I fix the values of the colors assigned to the vertices in the separator, once I condition on those values, if I draw a random sample from the Gibbs distribution (satisfying this conditioning), then the assignment on R is going to be independent from the assignment on T.

The way to think about this is that once I've pinned the vertices in S, I'm effectively deleting them from the graph, and then I have two components which are disconnected from each other and have no path between them. That's why you get independence once you condition on these vertices.

So these distributions are definitely not jointly independent — they'll have strong correlations in some sense — but at least you can factorize them, after you've pinned the values of some vertices.

§16.4 Phase transitions

Now we've defined this very large class of distributions; what kinds of things can we study? One thing we can look at is, as usual, *phase transition* phenomena. Typically, we'll be able to detect phase transitions through how the properties of the correlations between vertices change as we vary some of the parameters, like β (the strength of the interactions).

Informally, as we said earlier, if I look at the Ising model (or more generally, the version for q-colorings) where I have a very large β, then the distribution is putting more probability mass on configurations that have many agreements. So you should think of large β as representing a configuration with very large correlations. Conversely, if β is small in magnitude, all the correlations are small (and if $\beta = 0$, every vertex will get an independent color).

What turns out to happen in many classes of graphs (almost all families of graphs you might care about) is that there's a sharp threshold between these two regimes. So we'll have the following diagram: can imagine plotting β on the x-axis. There'll be a specific critical point β_c below which we have the 'high-temperature phase,' and the way you should think about this is that there's very weak correlations (we'll be more precise soon what we mean by this). And above β_c is the 'low-temperature phase,' where you have strong or *long-range* correlations.

Let's illustrate this with an example.

Maybe before we get there, this diagram is very heuristic; there are typically many transition points, where you can get different kinds of strengths of correlation and so on, but we won't get into the details of that.

These phase transitions, apart from being interesting from a physics perspective, also have very important consequences for the complexity of various algorithmic tasks, which we'll discuss later.

Example 16.10

Consider the ferromagnetic Ising model on the complete graph K_n. (Ferromagnetic means $\beta > 0$, so we're encouraging agreements.) Consider the random variable

$$X = \langle \sigma, \mathbf{1} \rangle = k - (n-k)$$

(where we think of $\sigma \in \{\pm 1\}^n$, and k is the number of $+1$'s under σ).

This is equivalent to just counting the number of $+1$'s or -1's. We're expecting that if β is very large, this random variable is going to concentrate on one of two possible values — either nearly everyone is -1 (in which case this is small, or close to $-n$), or nearly everyone is $+1$ (in which case this random variable is very large close to n). And of course if β is very small — if $\beta = 0$ then this is just a sum of independent random variables, which has things like the central limit theorem.

If we think about what this random variable looks like, we'll have

$$\mathbb{P}[X = n - 2k] \approx \frac{\binom{n}{k} \exp(\frac{\beta}{2}(n-2k)^2)}{\sum_{\ell=0}^{n} \binom{n}{\ell} \exp(\frac{\beta}{2}(n-2\ell)^2)}$$

(up to shifting, this is really just the quadratic form of σ with respect to the adjacency term of the complete graph). We think of the second factor $\exp(\bullet)$ as an *energy* term (in physics language), and the $\binom{n}{k}$ as an *entropy* term (the number of configurations at that energy level; the denominator is just the normalization).

The phase transition comes in because of the competition between these two effects — the entropy and energy. Using the approximation

$$\binom{n}{k} \approx \exp\left(n \cdot H\left(\frac{k}{n}\right)\right),$$

where $H(p) = -p \log p - (1-p)\log(1-p)$, the numerator is going to be approximately

$$\exp\left(n \cdot H(\frac{k}{n}) + \frac{\beta}{2} \cdot (n-2k)^2\right).$$

And in this expression, we have two terms competing with each other. We have an entropy term, which is maximized at the middle — when $k = \frac{n}{2}$. On the other hand, we have an energy term that prefers configurations where everyone is $-$ or everyone is $+$. And basically, there's going to be a sharp transition for when one of these effects wins out. One way to see this is that if we change the normalization slightly — if we rescale β so that we have $\frac{\beta}{2n}$ instead of $\frac{\beta}{2}$ — then this is really

$$\exp\left(n \cdot \left(H\left(\frac{k}{n}\right) + \frac{\beta}{2}\left(1 - \frac{2k}{n}\right)^2\right)\right).$$

So I can define a function

$$\psi_\beta(p) = H(p) + \frac{\beta}{2}(1-2p)^2.$$

When $\beta < 1$, you'll find that this is a nice, unimodal, concave function. At $\beta = 1$, a transition happens where it's still concave, but it's very flat at the top. And when $\beta > 1$, you get a bimodal function, with two distinct global maximizers.

So you can already sort of see the phase transition in the properties of the system, by thinking about how this univariate function changes as I change β. When β is small (the high-temperature regime), you expect a distribution that behaves basically like a sum of independent random variables. At the transition point we get some still unimodal distribution, but it's more anticoncentrated. And above this, you'll really have a bimodal distribution concentrating mass on two very distinct points — one configuration where most vertices are assigned -1, and one where most are $+1$.

We'll return to this example again in a future lecture and make some of these statements more precise. But this is roughly what the picture looks like. And you can see this also in many other models, not just the complete graph.

So the theme here is that phase transitions come from the 'competition between energy and entropy.'

§16.5 Correlation decay and spatial mixing

Now we'll illustrate one very important property of these distributions that you can study when your graph isn't like the complete graph — when your graph is sparse (and maybe even has geometry, and so on).

It turns out one very important property, important for designing algorithms, is the notion of *correlation decay*. Above, we characterized the notion of high temperature as saying correlations are small. When your graph has geometry or is sparse, we'll look at a slightly different property, that says that correlations are decaying with distance.

Here's one way of making correlation decay precise, called *spatial mixing*.

Definition 16.11. We say μ exhibits *weak spatial mixing* if there exist $0 < \delta < 1$ and $C > 0$ such that the following holds: For every $r \in V$, every subset $\Lambda \subseteq V \setminus \{r\}$, and every pair of pinnings $\tau, \sigma : \Lambda \to [q]$, we have

$$\|\mu_r^\tau - \mu_r^\sigma\|_{\mathsf{TV}} \leq C(1-\delta)^{\mathrm{dist}(r,\Lambda)}.$$

What we want to say here is that the influence of the subset of vertices Λ on our vertex r is small — specifically, that it's decaying based on the distance between r and this set of vertices. To formalize this, we look at the marginal distribution of the color assigned to r conditioned on one of these pinnings (or boundary conditions). And if I compare that to what happens when I change the boundary condition (from τ to σ), the total variation distance between those two marginals should be exponentially decaying in $\mathrm{dist}(r, \Lambda)$.

This is a mouthful, so let's draw an illustration to show what this means. Basically, what the definition is trying to capture is the following. Imagine I have two pictures of a lattice; and let's suppose r is the origin. And let's take Λ to be some vertices at a faraway distance (in both pictures, we draw Λ as a large box around r). But now in these two pictures, in the first one I pin all the vertices in Λ to some configuration τ — in the Ising model, for example, maybe everyone here is $+1$. And then I look at the same picture but with a different pinning, where I pin Λ to σ (e.g., maybe σ is all -1's). And I'm comparing how different the marginal distribution of the root r is between these two pictures. What we want to say is that they're very similar — in fact they're exponentially close to each other (in $\mathrm{dist}(r, \Lambda)$). What we're trying to formalize is the fact that 'faraway vertices cannot collectively influence the vertex r.'

This is some property of the distribution you can ask for. It definitely doesn't hold in general, but it's a nice property if you do have it, and it turns out you can use this property to design algorithms and so on.

Student Question. *Does the distance have to be the same along all points of the boundary?*

Answer. Here we only look at the closest vertex to the boundary when defining the distance — more formally, $\mathrm{dist}(r, \Lambda) = \min_{u \in \Lambda} \mathrm{dist}(r, u)$.

Remark 16.12. One thing to mention is you can of course relax this notion of correlation decay — for example, instead of exponential decay, you can ask for polynomial decay. You can also ask for more average-case versions, where instead of picking a worst-case choice of σ and τ, we take random ones. These weaker notions are also useful. For example, they're intimately related to the broadcast process we saw a long time ago. We saw there's a threshold for signal-to-noise where above this threshold you can recover the assignment of the root, and below it you can't. And it turns out one way to phrase non-reconstructibility is to show a type of correlation decay — not as strong as weak spatial mixing, but an average-case version where τ and σ are randomly generated.

Remark 16.13. This definition is also called *weak* spatial mixing. There's also a strong version, but we won't talk about it here.

§16.6 Spatial mixing for the hardcore model

In the last 20 minutes, we'll talk about spatial mixing for a very concrete model — independent sets, but on a tree, where things are more convenient to analyze.

For convenience, we'll work with the following very special case:

Example 16.14

Consider the d-ary tree of height ℓ rooted at r, which we denote by $\mathbb{T}_{d,\ell}$.

We're going to prove some sort of correlation decay in the special case where we compare the following two pictures. We draw two pictures of the tree (as a triangle) with r at the top. We're going to think of our boundary set of vertices Λ as all the vertices at distance ℓ from r in this tree — i.e., all the leaves of this tree. And I want to look at two different pinnings, which are the ones you should intuitively think of as maximizing or minimizing the probability that the root is in the independent set. So we'll consider one pinning where σ is 'in,' and the other pinning where τ is 'out.' So I'm pinning all the leaves of the tree to either everyone being in the independent set, or everyone being out of it. And I want to compare how the marginal distribution of the root differs under the two pictures; we'll show that they're very close below some threshold. (In some sense, you can think of this as being the worst-case pair of pinnings, so this would give you some sort of spatial mixing; but we'll just focus on this special case.)

Goal 16.15. Compare μ_r^τ and μ_r^σ.

Student Question. *Didn't we do something like this a month ago? This looks very familiar...*

Answer. There was a very related picture when we looked at something called the *broadcast process*, where I first sample a random assignment of the root and then do some sort of noisy thing.

In the broadcast process, the assignments received at the leaves were randomly generated. But here, we're looking at a *worst-case* pair of pinnings. So we're looking at two extremal pinnings — one where everyone is in and one where everyone is out. This is related to the comment from earlier — you can ask for a version of correlation decay that holds in expectation for randomly generated τ and σ (instead of worst-case ones), and that's very much related to what we did with the broadcast process.

What we want to prove is the following theorem:

Theorem 16.16 (Kelly 1985)

Fix $d \geq 1$, and define

$$\lambda_c(d) = \frac{d^d}{(d-1)^{d+1}} \approx \frac{e}{d-1}.$$

Then the following hold:

- If $\lambda < \lambda_c(d)$, then $\|\mu_r^\tau - \mu_r^\sigma\|_{\mathsf{TV}} \leq \exp(-\Omega(\ell))$.
- If $\lambda > \lambda_c(d)$, then $\|\mu_r^\tau - \mu_r^\sigma\|_{\mathsf{TV}} \geq \Omega(1)$.

So if λ is below the critical value, then we get exponentially small total variation distance; this is basically weak spatial mixing. Meanwhile, if λ is above this threshold, then actually these things are lower-bounded

by some constant independent of ℓ.

> **Student Question.** *What's λ?*
>
> **Answer.** It's the parameter from the description of the hardcore model, which describes how much you care about the density of the independent set drawn from your Gibbs distribution.

§16.6.1 Further results

Before we get into a proof sketch of this theorem, we'll talk about some of the more interesting algorithmic consequences of this, which was a long line of work in statistical physics and TCS. Building on this, you can prove the following result, which says that this threshold isn't just a threshold for infinite d-ary trees, but also for all graphs of bounded degree.

> **Theorem 16.17** (Weitz 2006, Salas–Sokal 2005)
> If $\lambda < \lambda_c(d)$, then the hardcore model on *any* graph of maximum degree $d+1$ has spatial mixing.

And another theorem (by many people — a combination of many results in the field) also characterizes the computational complexity of some very natural algorithmic tasks.

> **Theorem 16.18**
> - If $\lambda < \lambda_c(d)$, then there exists an $O(n \log n)$-time algorithm for sampling from the hardcore distribution.
> - If $\lambda > \lambda_c(d)$, then sampling is NP-hard.

Sampling is one of these very fundamental algorithmic tasks in high-dimensional statistics and machine learning, where you want to basically draw samples from your distribution — it has many downstream applications (once you can sample, you can compute or estimate various statistics — you can use Monte Carlo to compute expectations of various functions, use it to perform statistical inference, and so on). It turns out these phase transitions have striking consequences for the complexity of this task.

> **Student Question.** *Is this exact or approximate sampling?*
>
> **Answer.** It's approximate — I give you an arbitrary parameter ε, and I want you to be able to sample from the distribution up to total variation distance ε.

So basically, these phase transitions are interesting from a physics kind of perspective, but they also have very important consequences for algorithms.

§16.6.2 Proof sketch of Theorem 16.16

Now we'll spend the last ten minutes or so sketching the proof of Theorem 16.16.

To start, we'll use one very convenient fact about the hardcore model, which basically says that if I pin a bunch of vertices to being in the independent set, that's basically equivalent to deleting all those vertices and their neighbors; similarly, if I pin them to be out of the independent set, that's equivalent to deleting just those vertices.

Fact 16.19 — If we fix $v \in V$ and consider $\mu_G^{v \leftarrow \text{out}}$ (the hardcore Gibbs distribution conditioned on v not being in the graph), this is the same as μ_{G-v} (the hardcore distribution where I delete v from the graph). Similarly, we have

$$\mu_G^{v \leftarrow \text{in}} = \mu_{G - N[v]},$$

where $N[v] = \{v\} \cup \{u \in V \mid u \sim v\}$ is the closed neighorhood of v.

The reason this is useful is that our two pictures then become much simpler — we have a tree of depth $\ell - 1$, and we're comparing it with a tree of depth $\ell - 2$. So I'm really looking at two trees where I don't pin anything, but I'm changing the depth of the tree — I have one with depth $\ell - 1$, and I'm comparing it with a tree of depth $\ell - 2$ (because if I pin all these vertices to be out, that's just deleting them; if I pin them to be in, then I'm deleting them along with their neighbors, which are their parents).

Why is this convenient? Now I basically want to write down what the probability of the root is, as a function of the depth of this tree. So we define

$$p(\ell) = \mathbb{P}_{\mathbb{T}_{d,\ell}}[r \leftarrow \text{out}]$$

(the marginal probability with respect to the height-ℓ tree that the root is out). So we're looking at $p(\ell - 2)$ and $p(\ell - 1)$ — we want to study how these marginals evolve as I increase the depth of this tree.

Now here's the claim: we're going to write down a sort of recursion for these marginals.

Claim 16.20 — Define $f(p) = \frac{1}{1+\lambda p^d}$. Then $p(\ell) = f(p(\ell - 1))$ for all $\ell \geq 1$.

We'll explain in a moment where this function comes from. But this essentially states that iterating the function gives me all my marginals – in other words, $p(\ell) = f^{\circ \ell}(\frac{1}{1+\lambda})$ (where this thing is $p(0)$).

This seems very abstract, so why is it relevant? Now we've reduced this to the problem of understanding the fixed points of this single univariate function f. And it turns out that this function is going to have only one fixed point. But if I look at the function composed with itself once, then I'm going to have two extra fixed points. So here's the picture of f: f is in general a decreasing function (we graph f and the line $g(p) = p$), so it's going to have one fixed point. But now if I look at $f \circ f$, it's going to be an increasing function, and depending on the value of λ, we may have multiple fixed points. For $\lambda < \lambda_c(d)$, we only have one (we draw $g(p) = p$ again, and we plot $f(f(p))$ as starting above and being concave, then going above and becoming convex). And for $\lambda > \lambda_c(d)$ you're going to get multiple fixed points (we draw this as a sort of s-shape going around the line $g(p) = p$). These fixed points are going to tell you what the marginals will be when you send the height of the tree to ∞.

So when $\lambda > \lambda_c(d)$, what'll happen is that the marginals of the root oscillate between these points as you increase the height of the tree. This means the correlations won't decay (your constant will be the difference between these points). Otherwise, when $\lambda < \lambda_c(d)$, you're contracting towards the unique fixed point, and that's going to be your exponential decay in the first statement.

Unfortunately we're out of time. The calculations for proving this claim are in the notes. It's not very difficult; we just use the conditional independence or the global Markov property we saw earlier, and also the self-similarity of the trees. And for these plots, you can just try plotting them in Mathematica or something and you'll see these pictures.

§17 April 7, 2025

Last class we started discussing probabilistic graphical models, which are a general class of high-dimensional probability distributions. Given one of these, one problem you might be interested in is drawing a *sample* from that distribution. In this class, we'll introduce one algorithm for doing so, using Markov chains.

§17.1 Markov chains

Definition 17.1. A *Markov chain* is a stochastic process $\{X_t\}_{t\in\mathbb{N}}$ on a state space Ω which satisfy the *Markov property*, meaning that for every $t \geq 0$ and every $x_0, \dots, x_t \in \Omega$, we have

$$\mathbb{P}[X_t = x_t \mid X_0 = x_0, \dots, X_{t-1} = x_{t-1}] = \mathbb{P}[X_t = x_t \mid X_{t-1} = x_{t-1}].$$

You should think of t as time, and Ω as being an enormous state space (e.g., $\mathbb{R}^n$ or the space of colorings of a graph). The Markov property says that the probability distribution of X_t conditioned on the *entire* history is the same as if we condition on *only* the previous states. In other words, the process is *memoryless* — to transition to the next state X_t, you only have to look at where you are currently (i.e., X_{t-1}), and you can safely ignore everything else in the history.

One way you can think about a Markov chain is that it's a graphical model on a directed graph. You can imagine I have a graphical model with a vertex for X_0, and another vertex for X_1, all the way up to X_t (extending to ∞). And then you have directed edges that point forwards in time. Each X_i is taking values in this gigantic state space Ω. So you can think of a Markov chain as being on a graphical model on a very simple graph — one that's just a path of vertices — but where the space of 'colors' (possible states that these random variables can lie in) is extremely large.

Another way you can think about Markov chains is as *random walks*, or as directed graphs. Imagine you have a very large state space Ω. Now for every pair of states $x, y \in \Omega$, you'll add an edge between them if there's a chance of transitioning from x to y.

Before we get to this, in this course we're only going to be considering the *time-homogeneous* case:

Definition 17.2. A *time-homogeneous Markov chain* is a Markov chain where $\mathbb{P}[X_t = y \mid X_{t-1} = x]$ does not depend on t.

In particular, this means we can encode all transition probabilities into a matrix $P \in \mathbb{R}_{\geq 0}^{\Omega\times\Omega}$ (a nonnegative-entry matrix indexed by pairs of states), where

$$P(x \to y) = \mathbb{P}[X_t = y \mid X_{t-1} = x] \quad \text{for all } t.$$

(This is called the *Markov kernel* or the *transition probability matrix*.)

If I have a time-homogeneous Markov chain, I can think of the corresponding graph where I have a vertex for every state, and a directed edge between two states if there's a nonzero chance of transitioning from one to the other with respect to this transition probability matrix, i.e., $P(x \to y) = 0$. Then we can think of this sequence of random variables as taking an (infinite) random walk in this directed graph.

§17.2 Some examples

Example 17.3 (Simple random walk)

Consider a simple random walk on an undirected graph $G = (V, E)$. We can think of this process algorithmically as follows:

- Start with an arbitrary vertex $u_0 \in V$.
- For every $t \geq 1$, Let u_t be a uniformly random neighbor of u_{t-1}.

So we have our graph (we draw it as having edges $\{12, 23, 34, 42\}$). If at time $t = 0$ I'm currently at vertex 3, then at time 1 I'll be at either 2 or 4 — I pick one with probability $\frac{1}{2}$. Here, the corresponding matrix

would be

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}$$

I look at each vertex v, look at its set of neighbors, and put $\frac{1}{\deg(v)}$ in that corresponding entry. In general, the corresponding transition matrix will be that I take the adjacency matrix of this graph and normalize it so that all the rows sum to 1 — so $P_G = D_G^{-1} A_G$, where D_G is the diagonal matrix whose diagonal consists of $\deg(v)$ for all v.

Example 17.4 (Birth-death process)

We have state space $\Omega = \{0, \dots, n\}$, and $\mathbb{P}[i \to i+1] = b_i$, $\mathbb{P}[i \to i-1] = d_i$, and $\mathbb{P}[i \to i] = r_i$.

This is often used to model the evolution of a population of organisms. We have vertices $0, \dots, n$; and from each, we have some probability of increasing the population by 1 (a *birth*) and some probability of *decreasing* the population by 1 (a *death*). You can also have self-loops — the probability that the population doesn't change at all.

The transition matrix will be a 'tri-diagonal' matrix — we'll have the r's on the diagonal, the b's above it, and the d's below it (and 0's everywhere else). Of course, we have the constraint that $b_k + r_k + d_k = 1$ (and these are all nonnegative, and so on).

§17.3 Stationary distributions

Now we'll discuss an algorithmic application of Markov chains to *sampling*. To do this, we need to define stationary distributions.

Definition 17.5. We say a distribution μ on Ω is *stationary* with respect to a Markov chain P if $\mu = \mu P$.

What does this mean? Here's a fact:

Fact 17.6 — If we view $\mu_t = \mathsf{Law}(X_t)$ as a row vector in $\mathbb{R}^\Omega$, then

$$\mu_t = \mu_{t-1} P = \cdots = \mu_0 \cdot P^t.$$

So I can always express the marginal distribution of each of these random variables X_t linear-algebraically using this Markov kernel. And we say our distribution is stationary if it's sort of invariant under hitting it with this Markov kernel. In other worlds, if I imagine two worlds — one where I just sample from μ, and another where I sample from μ and take a step according to this chain — then these two worlds will have identical distributions if μ is stationary.

Eventually we'll be discussing convergence — we want the marginal distributions of these X_t's to converge to this stationary distribution, and that's going to be our connection to sampling.

Example 17.7

For the simple random walk, let

$$\mu_G(v) = \frac{\deg(v)}{2\,|E|}.$$

Then μ_G is stationary with respect to the random walk on G.

Intuitively, you can imagine that the vertices that will get visited most frequently are the vertices that have the most neighbors. You can already kind of see how you would prove this — I have this distribution μ, and if I think of it as a vector, then up to scaling by a constant, the entries of these vectors are the degrees of the vertices. These will cancel with the inverse of the degree matrix normalizing the adjacency matrix of the graph; and if you calculate things, you see that this equality holds.

For birth-death processes we'll have a more complicated formula; you can imagine proving it by induction, though we'll prove it in a different way.

Example 17.8

For a birth-death process, if we define

$$w_k = \prod_{j=1}^{k} \frac{b_{k-1}}{d_k} \quad \text{and } \pi_k = \frac{w_k}{\sum_{j=0}^{n} w_j}$$

(where $w_0 = 1$), then π is stationary.

§17.3.1 Existence

So we've made this definition. One very natural question is, do these things exist in general?

Lemma 17.9

Every (time-homogeneous) Markov chain has at least one stationary distribution.

There may be many, and we'll discuss conditions under which the stationary distribution is unique. But this is a generic lemma saying that no matter what, you will always have at least one.

Proof. There's many different ways you can prove this. One way you can try to prove this is to think about this purely linear-algebraically — you can think about the equation $\mu = \mu P$ as an eigenvalue problem. There are a couple facts from linear algebra: We know the eigenvalues of P and $P^\intercal$ are the same. And we also know that P itself, if we look at *right* eigenvectors, it always has all-1's as an eigenvector (just because we imposed that every row of P is a distribution, so it sums to 1) — i.e., $P\mathbf{1} = \mathbf{1}$. This implies there exists some $v \in \mathbb{R}^n$ (this doesn't *a priori* need to have nonnegative entries) such that $v = vP$.

Now I want to convert this eigenvector (which may have negative entries) into a distribution which satisfies the same equation.

We'll do the one thing you might try right off the bat — if I have negative entries, I just take their absolute value (to get a vector with positive entries), and then normalize everything to be a distribution. S we let

$$\mu_i = \frac{v_i}{\|v\|_1}$$

for all i (where $\|v\|_1 = \sum_i |v_i|$). And the claim is that μ is stationary.

Now let's look at this μ. What I want to show is that

$$|v_i| = \sum_j |v_j| \, P(j \to i).$$

If we can do this, then I'm done (since I'm just normalizing everything by the same constant).

A first observation is that we always have an *inequality*: because v is an eigenvector, we have

$$|v_i| = \left|\sum_j v_j P(j \to i)\right| \le \sum_j |v_j| \, P(j \to i)$$

(where the second inequality is the triangle inequality).

Now I just need to show that this inequality is actually an *equality*. Now we're going to use the fact that v is an eigenvector again. Suppose for contradiction that there exists i for which the inequality is strict. Then if I sum over all i, it'll still be a strict inequality. But both sides are going to sum to $\|v\|_1$, so that's a contradiction — the inequality cannot possibly be strict.

In other words, we're using the fact that

$$\|v\|_1 = \sum_i |v_i| = \sum_i \sum_j |v_j| \, P(j \to i).$$

So every inequality must be tight.

And that's it — that certifies that if I take this eigenvector v, take the absolute value of everything, and normalize to make it a distribution, then that gives a stationary distribution for P. □

§17.3.2 Uniqueness and convergence

So we have the *existence* of stationary distributions; now we want to talk about uniqueness and convergence. Here, there are essentially two barriers to uniqueness or convergence — there's basically two kinds of 'bad' examples.

One kind of bad example is if you think about a Markov chain as a random walk on a graph, this graph is *disconnected*.

Example 17.10

For example, imagine the extreme case where all vertices just had self-loops. Then P would just be the identity matrix, and *every* distribution would be stationary.

Example 17.11

In general, if the graph of your Markov chain has more than one connected component — say $\mathcal{C}_1$ and $\mathcal{C}_2$ — then there'll be one stationary distribution supported just on $\mathcal{C}_1$ and another supported only on $\mathcal{C}_2$, and every convex combination of them is going to be stationary. So if your Markov chain is disconnected, then you can have many stationary distributions.

That's going to be a barrier to uniqueness of the stationary distribution. A barrier to convergence is if the Markov chain has directed cycles:

Example 17.12

Imagine we have a Markov chain where every vertex only has a single directed edge out of it, and those form a directed cycle.

This will have a stationary distribution, the uniform distribution over vertices. But if I initialize it deterministically at some vertex and track the trajectory of my Markov chain, then these random variables X_t are always going to be deterministic, depending on what t is modulo the length of the cycle.

So here $\mathsf{Unif}\{0,1,2,3\}$ is stationary (in our example with 4 vertices); but if $\mu_0 = \delta_0$ (meaning I start at vertex 0), then $\mu_t = \delta_{t \bmod 4}$. You're always going to keep going around this cycle, and you'll never actually converge.

This example is actually fairly generic:

Example 17.13

Imagine I have a graph which is bipartite. If I initialize the chain on one side, then on the next step I'll be on the other side, and then back on this side — I'll keep bouncing between the two sides, and this will be a barrier to convergence.

Perhaps kind of surprisingly, these are the *only* barriers!

Definition 17.14. We say a Markov chain P is *irreducible* if for every pair of states $x, y \in \Omega$, there exists some t such that $P^t(x,y) > 0$.

In other words, there exists some number of steps t such that you can always walk from x to y in that number of steps.

Remark 17.15. This is the same as saying that the graph associated to P is strongly connected.

Definition 17.16. We say P is *aperiodic* if $\gcd(\{t \geq 1 \mid P^t(x,x) > 0\}) = 1$ for all $x \in \Omega$.

This is maybe a weird definition; what does it mean? If I look at $P^t(x,x)$, this is looking at whether or not there is a directed cycle of length t starting at x (not necessarily a simple one). So in other words, I'm asking that the lengths of all directed cycles that x participates in should have greatest common divisor 1.

Definition 17.17. We say P is *ergodic* if it is irreducible and aperiodic.

Then we have the fundamental theorem of Markov chains, which says that these are the only barriers to uniqueness and convergence.

Theorem 17.18 (Fundamental theorem of Markov chains)

If P is ergodic, then P has a unique stationary distribution μ. Moreover, for every choice of initial distribution μ_0, we have that $\mu_t = \mu_0 P^t \to \mu$ pointwise as $t \to \infty$.

So ergodicity guarantees uniqueness and convergence from an arbitrary initial state.

§17.4 Markov Chain Monte Carlo

The fact that this theorem guarantees convergence from an *arbitrary* initial states is onen reason Markov chains are useful for sampling. Imagine μ is some complicated distribution. Then I can start with an arbitrary distribution μ_0; and as long as I have a Markov chain with stationary distribution μ and I can implement this chain, by running the chain for a long time I can at least approximately sample from μ.

This is called the *Markov Chain Monte Carlo paradigm*: Suppose that μ is a complicated distribution you want to sample from. Then what you can do is you can design a Markov chain P such that μ is stationary and its transitions are efficiently implementable. (We also want this Markov chain to be ergodic.) And then we pick an arbitrary starting state μ_0, and then simulate the chain up to a large time T. Then we know that $\mathsf{Law}(X_T) \approx \mu$. (It is an important question how you should set T, and we'll address that in a future lecture,

but this is the rough scheme behind one of the most popular approaches for sampling from high-dimensional distributions.)

> **Student Question.** *If you have two strongly connected components and an edge from $\mathcal{C}_1$ to $\mathcal{C}_2$ and $\mathcal{C}_2$ is aperiodic, is the Markov chain still ergodic?*
>
> **Answer.** We don't call it ergodic, but it's enough for the conclusion of the distribution — if I have this edge, then I'll be leaking mass from $\mathcal{C}_1$ to $\mathcal{C}_2$, so any stationary distribution would have 0 mass on $\mathcal{C}_2$. So there is a more refined version of Theorem 17.18, but we won't discuss it.

One reason MCMC is so powerful is that guaranteeing ergodicity is actually extremely easy — ergodicity is actually an extremely weak condition. Guaranteeing connectivity of your Markov chain is quite easy. Guaranteeing aperiodicity is also quite easy, because you can always manually force the Markov chain to be aperiodic. You can do this by just adding self-loops, or making the Markov chain *lazy* — you can replace P with $\frac{1}{2}(I + P)$. So for every state, we just don't move with probability $\frac{1}{2}$ (this is why it's called lazy), and with probability $\frac{1}{2}$ we follow the usual transition of the Markov chain.

The lazification is always going to be aperiodic — every state participates in a length-1 cycle corresponding to that self-loop, so the gcd will always have $t = 1$ in that set. So this is always aperiodic.

And moreover, it essentially preserves all the properties you care about:

> **Fact 17.19 —** If P is irreducible, so is its lazification; and if μ is stationary with respect to P, then it's also stationary with respect to its lazification.

So essentially once you have an irreducible Markov chain (which is in general very easy to construct), you can also get an aperiodic one essentially for free. So in general ergodicity is very easy to satisfy.

As one more comment:

> **Fact 17.20 —** The simple random walk on an undirected graph G is aperiodic if and only if the graph G is not bipartite.

This is just because in an undirected graph, the set of cycle lengths at x always contains $t = 2$.

We'll prove the fundamental theorem in the next lecture, since we need to introduce a few more tools to prove it (we'll see a proof using coupling). For the rest of this lecture, we'll see some examples for Markov chains for sampling; and we'll see how you can build Markov chains that certifiably have the correct stationary distribution that we want.

§17.5 Reversibility

In general, if you're handed a Markov chain in the wild, determining its stationary distribution is not a simple task. But for a class of Markov chains called *reversible* chains, it's actually quite easy; and we'll see a general scheme of constructing such chains.

> **Definition 17.21.** We say P is *reversible* with respect to μ if it satisfies the *detailed balance equations*:
>
> $$\mu(x)P(x \to y) = \mu(y)P(y \to x) \quad \text{for all } x, y \in \Omega.$$

There's a simple lemma that this is a sufficient condition for stationarity.

> **Lemma 17.22**
>
> If P is reversible with respect to μ, then μ is stationary.

Let's see some examples.

Example 17.23

Suppose P is a symmetric matrix. Then it's reversible with respect to the uniform distribution (over all your states).

In general, P won't be symmetric; but suppose that somehow it is. This is just because $\mu(x) = \mu(y)$, so then the detailed balance equations just recover the fact that your Markov chain is symmetric.

For instance, if I have a simple random walk on a d-regular graph, then the uniform distribution will be stationary (and it'll be reversible).

Now let's return to the birth-death process.

Example 17.24

The birth-death process is reversible with respect to π (as defined earlier).

Basically, the weights w_k were chosen so that the detailed balance equations are satisfied. We just need to check that

$$w_{k-1} \cdot \mathbb{P}[(k-1) \to k] = w_k \cdot \mathbb{P}[k \to k-1].$$

By definition, this says $w_{-1}b_{k-1} = w_k d_k$. In particular, the w_k's we defined satisfy that relation (by an inductive argument), so we see that this is the right stationary distribution.

Really, the way you should think about reversible Markov chains is that they're essentially equivalent to a simple random walk on an undirected graph, where the edges have weights attached to them (and you pick a neighbor with probability proportional to the edge weight connecting the two vertices). (This essentially describes *all* reversible Markov chains.)

§17.6 Examples of big Markov chains

Now let's see some more nontrivial or bigger Markov chains that come up in sampling-type applications.

§17.6.1 The swap chain

Example 17.25 (Swap chain)

Fix a bipartite graph $G = (L \cup R, E)$, where $|L| = |R| = n$. We'll define a Markov chain for sampling perfect matchings in this bipartite graph, which we call the *swap chain* on perfect matchings: Suppose the current matching is $M \subseteq E$. In a move, I'm going to pick two distinct edges uniformly at random from this matching, and try to do a swap if that is legal:

- Pick $u_1v_1, u_2v_2 \in M$ uniformly at random (without replacement).
- Let $M' = M \cup \{u_1v_2, u_2v_1\} \setminus \{u_1v_1, u_2v_2\}$. If M' is still a valid perfect matching, then transition to M'. (Otherwise, stay at M.)

So I take these two edges u_1v_1 and u_2v_2 and exchange partners, if I can do so (and otherwise we do nothing).

The picture to have in your head is something like the following: Suppose my bipartite graph has L and R, and my matching looks like:

Then I pick two special edges u_1v_1 and u_2v_2 (drawn in red). And suppose that there are also edges u_1v_2 and u_2v_1 (in the graph, but of course not in the matching). Then my transition is going to go to the following matching: all the other edges are going to stay the same, except that I'm going to remove these two red edges, and put in u_1v_2 and u_2v_1.

So that would be one move in this chain.

It's clear that if I look at any pair of distinct perfect matchings, then if they differ in a swap, the probability of transitioning from one to the other is going to be $\frac{1}{\binom{n}{2}}$ (the probability of choosing that pair of edges for the swap). This is independent of whether I'm going $M \to M'$ or $M' \to M$. In particular, the transition matrix for this Markov chain is going to be symmetric, so the uniform distribution over all perfect matchings will be stationary.

Example 17.26

For instance, if I have the complete bipartite graph, then I can think of perfect matchings as really being permutations on $\{1, \dots, n\}$. Then this Markov chain is just a random walk on all permutations of $\{1, \dots, n\}$ given by applying random transitions to the permutation.

In general, this Markov chain is not irreducible — for example, a cycle of even length $G = C_{2n}$ only has two perfect matchings, which are completely disjoint from each other.

So if I have a cycle of even length, I only have two perfect matchings; and they're completely disjoint, so I can't move from one to the other using local swaps like this.

§17.6.2 Glauber dynamics

That's one class of examples; here's another.

Example 17.27 (Glauber dynamics)

Consider the ferromagnetic Ising model on a graph G — this means we're assigning ± 1's to the vertices of this graph, and

$$\mu(\sigma) \propto \exp\left(\frac{\beta}{2}\sigma^\intercal A_G \sigma\right) \quad \text{for all } \sigma \in \{\pm 1\}^V.$$

The Glauber dynamics is a Markov chain that samples from this distribution: At each step it picks a uniform random vertex from this graph, and then resamples the distribution of that vertex conditioned on everything else in the graph.

- Pick $v \in V$ uniformly at random.
- Update $\sigma_v \leftarrow -\sigma_v$ with probability
$$\frac{\mu(\sigma^{\oplus v})}{\mu(\sigma) + \mu(\sigma^{\oplus v})},$$
where $\sigma^{\oplus v}$ is the configuration which agrees with σ everywhere else but flips the assignment on v.

(This distribution places more mass on configurations with many agreements along edges; the parameter β (the *inverse temperature*) sort of scales the strength of these interactions.)

Again, you can check that this Markov chain has μ as its stationary distribution — you can even check reversibility — we have

$$\mu(\sigma)\mathbb{P}[\sigma \to \sigma^{\oplus v}] = \frac{\mu(\sigma)\mu(\sigma^{\oplus v})}{\mu(\sigma) + \mu(\sigma^{\oplus v})},$$

which doesn't depend on whether I'm going $\sigma \to \sigma^{\oplus v}$ or the other way around.

Also, since this distribution is supported on everything in the hypercube, this chain will be irreducible; and it's also aperiodic, because there's some probability of staying at the current state.

So Theorem 17.18 says that if I run Glauber dynamics for a very long time (from an arbitrary initial configuration), I'll always reach my desired distribution.

Next class we'll discuss the fundamental theorem, and also start discussing mixing times.

§18 April 9, 2025 — The Markov Chain Monte Carlo paradigm

Today we'll continue our discussion of Markov chains and we'll see some general schemes for constructing Markov chains for sampling from high-dimensional probability distributions.

A few key definitions from last class:

Definition 18.1. A Markov chain P is *irreducible* if for all $x, y \in \Omega$, there exists $t > 0$ such that $P^t(x \to y) > 0$.

In other words, the underlying graph the MC is walking on is strongly connected.

Definition 18.2. A Markov chain P is *aperiodic* if for all $x \in \Omega$, we have

$$\gcd\{t \geq 1 \mid P^t(x \to x) > 0\} = 1.$$

This in some sense means all the directed cycles in the graph don't have some nontrivial common denominator.

Definition 18.3. A Markov chain is *ergodic* if it is both irreducible and aperiodic.

Definition 18.4. A Markov chain is *reversible* with respect to μ if for all $x, y \in \Omega$, we have

$$\mu(x)P(x \to y) = \mu(y)P(y \to x).$$

This essentially means the MC is equivalent to a walk on an undirected (weighted) graph.

Last time, we stated but didn't prove the following theorem.

Theorem 18.5 (Fundamental theorem)
If P is ergodic, then there exists a unique stationary distribution μ. Furthermore, for every distribution μ_0, we have $\mu_0 P^t \to \mu$ as $t \to \infty$.

The MCMC paradigm is: If you have some high-dimensional probability distribution you want to sample from, you design some MC which is ergodic and has this distribution as its stationary distribution. And then

you simulate the MC for a sufficiently large number of steps t, and output the final state as your sample; and the hope is that sample will be approximately distributed according to the target distribution.

Last class we saw a few basic examples of this paradigm. In this lecture we'll see two very general schemes for constructing Markov chains, both based on reversibility.

§18.1 Metropolis–Hastings algorithm

The first scheme we'll present is a very famous algorithm, called the Metropolis–Hasting algorithm. The way to think of it is as a general purpose scheme for taking any arbitrary Markov chain, and converting it into a new Markov chain which has the correct stationary distribution.

The setting is that μ is your target distribution (e.g., think of it as the Gibbs distribution of some graphical model). And Q is *any* Markov chain on the right state space, which might have nothing to do with μ. The idea is Q proposes the next state we possibly transition to; and then with some probability, we either accept or reject this proposal. We think of μ as being given by

$$\mu(x) = \frac{w(x)}{\sum_{y\in\Omega} w(y)}.$$

So I have nonnegative weights $w(x)$ on my state space which I *can* compute (e.g., for the Ising model, this would be the exponential of the corresponding quadratic form); while the normalizing constant (the denominator) may be difficult to compute.

Algorithm 18.6 (Metropolis–Hastings)

Let μ be your target distribution, and let $Q \in \mathbb{R}_{\geq 0}^{\Omega\times\Omega}$ be *any* Markov chain on Ω, which we call the *proposal* chain.

Suppose the current state of the chain is $x \in \Omega$.

(1) Sample $y \sim Q(x \to \bullet)$.

(2) Go to y with probability

$$\min\left\{1, \frac{w(y)Q(y\to x)}{w(x)Q(x\to y)}\right\}.$$

Otherwise stay at x.

So we're at some state x, and we want to say how to transition to the next state of the chain. We first sample y according to Q (every row of Q is some probability distribution, and we assume we can sample from it). And then we accept this step with some probability, and stay at x otherwise. This probability is sometimes called the *acceptance probability* or the *Metropolis filter*.

Assuming that we can implement Q, we actually can implement this algorithm. Note that if P_μ is the new chain, then

$$P_\mu(x\to y) = Q(x\to y)\cdot\min\left\{1, \frac{w(y)Q(y\to x)}{w(x)Q(x\to y)}\right\}$$

(for all $x \neq y$).

Lemma 18.7

For any Q and μ, the new Markov chain P_μ is reversible with respect to μ.

In particular, this means μ will be a stationary distribution; and if Q is ergodic, then P_μ will also be ergodic, so μ will be the *unique* stationary distribution for this Markov chain.

(We'll see an example in a moment.)

Proof. The proof is really just by verifying the detailed balance (or reversibility) equations. We can write down

$$w(x) \cdot P_\mu(x \to y) = w(x)Q(x \to y) \cdot \min\left\{1, \frac{w(y)Q(y \to x)}{w(x)Q(x \to y)}\right\}.$$

And now we can push $w(x)Q(x \to y)$ inside the minimum, so this becomes

$$\min\{w(x)Q(x \to y), w(y)Q(y \to x)\}.$$

And in particular, this expression is symmetric with respect to x and y — so I could reverse the roles of y and x and obtain the same thing. So in particular, this is also equal to $w(y)P_\mu(y \to x)$. And of course, I can normalize everything to change the w's to μ's. □

So we really just took one very simple criterion for guaranteeing μ is a stationary distribution, and we turned it into a very generic algorithm. The Markov chain Q could be completely arbitrary, but you can always 'correct' its stationary distribution to the one you want.

§18.1.1 Example: Hard spheres model

Let's see one nice example, perhaps one of the first applications (or reasons why this algorithm was invented to begin with).

Example 18.8 (Hard spheres model)

Imagine you want to simulate gas particles in a box. So we have a box with width R, and a bunch of little particles; we think of these particles as little spheres, each of radius $r \ll R$.

We define μ to be uniform over possible configurations of these gas particles — collections of points $\{x_1, \dots, x_n\}$ such that $\mathbb{B}(x_i, r) \cap \mathbb{B}(x_j, r) = \emptyset$ for $i \neq j$, and that $\mathbb{B}(x_i, r) \subseteq [0, R]^3$.

In words, the conditions say that the gas particles are disjoint from each other, and each of them is fully contained in the box (we write $\mathbb{B}(x, r) = \{y \mid \|y - x\|_2 \leq r\}$).

So I have a bunch of particles bouncing around in the box, and I want to understand what a typical configuration looks like (subject to the constraint that they're not allowed to overlap).

This is a very complicated distribution, but here's an algorithm that lets you sample from it (at least if we have a very long time). At each step, I'll pick one of these particles uniformly at random, and I'll try to move it somewhere. For instance, maybe I'll pick the particle on the middle-left. And what I'll do is, say I draw a slightly bigger ball around it (in red) — this is the region of points where I'm going to propose to move this point to. So maybe I'll propose to move it to a certain red point in this ball; this is the proposal. And I'll accept this move if the resulting configuration of points still satisfies all the constraints — namely, that it doesn't overlap with any of the other particles, and it's still fully contained within the box.

Algorithm 18.9

(1) Pick a random point (i.e., a random $i \in [n]$).

(2) Pick $x_i' \sim \mathbb{B}(x_i, L)$ (where L is some parameter).

(3) Move x_i to x_i' if this would still result in a legal configuration.

In some sense, if you squint, you can imagine this is kind of what each particle is trying to do — it's going to move around locally in this way.

§18.2 Markov chains via statistical inference

Now we'll consider another general-purpose scheme for designing these Markov chains, which connects this theory to the theory of statistical inference.

The basic idea behind this way of constructing MCs comes from the following thought experiment: imagine you have some friends (some very rich friends, or God, or something) with unbounded computational resources, and they do all the work for you — they sample from the distribution μ you care about. But rather than giving you the sample, as a good friend would do, they're a little mischievous, and they add a bit of noise to the sample. If your distribution lives in $\mathbb{R}^n$, for example, maybe they first sample from the distribution you care about, but then they add a bit of noise and give that corrupted sample to you.

You can kind of think of the corrupted sample they give you as a 'hint' for what the true sample should be, and your goal is to recover what that true sample is.

You can imagine if they only added a little bit of noise, then this recovery process is easy; and if they added a ton of noise, it becomes very difficult (as hard as the original sampling problem where you're not given anything).

A natural thing for you to do, given this corrupted sample, is to try to sample from the *posterior sample* — the distribution of x conditioned on what you have observed.

To make this precise, let Σ denote some other state space. (Right now we'll present things in a fairly abstract way, but this level of abstraction will actually be necessary for us.) We'll also let $\mathcal{N} \in \mathbb{R}_{\geq 0}^{\Omega \times \Sigma}$ be a matrix indexed by pairs of states — one from Ω (the state space you care about), and one from Σ — where all rows are distributions over Σ. You should think of $\mathcal{N}$ as capturing the noise that your mischievous friend is adding to the sample that they drew. So you imagine that the mischievous friend samples $X \sim \mu$ and gives you $Y \sim \mathcal{N}(X \to \bullet)$. (You can think of $\mathcal{N}$ as adding Gaussian noise, or something like that.)

One very natural thing for you to do is, now you've been given Y, and you can think of this as a statistical inference problem where you're trying to recover the original sample X (because that's the one that's actually distributed according to your target distribution). One natural approach to do this is to sample from the distribution of X *conditioned* on Y — i.e., we sample from the *posterior distribution* to estimate X. More precisely, this is given by Bayes' rule — we have

$$R_\mu(Y \to X) = \mathbb{P}[X \mid Y] = \frac{\mathbb{P}[Y \mid X]\mathbb{P}[X]}{\text{normalization}}.$$

And now we can write this in terms of μ and the noise process; we get

$$\mathcal{R}_\mu(Y \to X) \propto \mu(X)\mathcal{N}(X \to Y).$$

(The noise process $\mathcal{N}$ doesn't have to have anything to do with μ, but the recovery procedure $\mathcal{R}_\mu$ does.)

So far we haven't defined a Markov chain yet; but this recovery process is something that you can do.

You can imagine a dialogue between you and the mischievous friend. You ask them to sample, and they instead give you Y, and now you draw from this posterior distribution. Now, this dialogue is also giving you a Markov chain as follows.

Definition 18.10. We define $P_\mu = \mathcal{N} \cdot \mathcal{R}_\mu$.

In other words, we're first applying the noise process, and then applying this recovery map. This is a matrix whose rows and columns are indexed by the original state space Ω.

In other words, we take a state, apply some noise to it, and then apply a statistical inference procedure to sample from the posterior distribution.

Lemma 18.11

For every $\mathcal{N}$, we have that P_μ is reversible with respect to μ.

This noise process $\mathcal{N}$ could have been anything, so this gives you a very generic way to construct Markov chains with the correct stationary distribution.

The proof is similar to the one for Metropolis–Hastings — we can very directly verify the detailed balance equations.

§18.2.1 Example: Glauber dynamics

Let's see some examples of this kind of a scheme.

Example 18.12

Let μ be a distribution over the hypercube $\{\pm 1\}^n$. Suppose our noise process is defined by deleting a random coordinate — in other words, let $\Sigma = \bigcup_{i\in[n]}\{\pm1\}^{[n]\setminus\{i\}}$, and let $\sigma_{-i} \in \{\pm1\}^{[n]\setminus\{i\}}$ be given by erasing the ith coordinate of $\sigma \in \{\pm1\}^n$. Define

$$\mathcal{N}(\sigma \mapsto \sigma_{-i}) = \frac{1}{n} \quad \text{for all } i \in [n].$$

So we have a distribution over the hypercube (e.g., think of the Ising model). I start with the full distribution, and then I add a bit of noise by randomly deleting one of the coordinates.

Then we claim the corresponding recovery map is going to have the following form. It's going to take our partial assignment σ_{-i} and map it to one of the two possible full assignments which agree with σ_{-i} — so it'll complete σ_{-i} to either $\sigma^{i\leftarrow+1}$ or $\sigma^{i\leftarrow-1}$ (the extensions of σ_{-i} where we force the ith coordinate to be $+1$ or -1). And we'll ahve

$$\mathcal{R}_\mu(\sigma_{-i} \to \sigma^{i\leftarrow+1}) = \frac{\mu(\sigma^{i\leftarrow+1})}{\mu(\sigma^{i\leftarrow+1}) + \mu(\sigma^{i\leftarrow-1})}.$$

In other words, the Markov chain does the following: If I'm currently at some configuration, I pick a uniformly random coordinate, I delete that coordinate, and then I resample the assignment for that coordinate conditioned on the values of everyone else.

Put together, if I look at the corresponding Markov chain, if I look at the probability that σ transitions to the configuration where I flip the coordinate (which we denote by $\sigma^{\oplus i}$), we have

$$P_\mu(\sigma \to \sigma^{\oplus i}) = \frac{1}{n} \cdot \frac{\mu(\sigma^{\oplus i})}{\mu(\sigma) + \mu(\sigma^{\oplus i})}.$$

So this Markov chain just recovers Glauber dynamics, which we mentioned in the previous lecture.

One other comment about this is that maybe we can think briefly about how this process looks on something like the ferromagnetic Ising model.

Example 18.13

Consider the ferromagnetic Ising model on a graph $G = (V, E)$, where

$$\mu_G(\sigma) \propto \exp\left(\beta \sum_{uv\in E} \sigma(u)\sigma(v)\right) \quad \text{for all } \sigma \in \{\pm1\}^V.$$

The way to think about what Glauber dynamics is doing is that at each step, I pick a uniformly random vertex. Then I look at all my neighbors and see what their assignments are. And I take a weighted majority vote in some sense. So I pick a random vertex v. Then I have a bunch of neighbors; some of them are $+1$ and some are -1. And then I need to decide what I'm going to update my vote for v to. And what I'm going to do is set $v \leftarrow +1$ with probability

$$\frac{\exp(\beta \sum_{u\sim v} \sigma(u))}{\exp(\beta \sum_{u\sim v} \sigma(u)) + \exp(-\beta \sum_{u\sim v} \sigma(u))}.$$

So we're essentially taking a majority vote (corresponding to computing $\sum_{u\sim v} \sigma(u)$), then tossing a coin whose bias depends on how much of a majority there is.

§18.2.2 The Swendsen–Wang Markov chain

Now we'll see another example of this scheme that's specially tailored to the ferromagnetic Ising model, which will lead to very different behavior than the Glauber dynamics.

To define this, we'll first rewrite the Ising model slightly. Another way to write the Ising model is as

$$\mu(\sigma) \propto \exp\left(2\beta \cdot |E_\sigma|\right),$$

where E_σ is the set of monochromatic edges — i.e.,

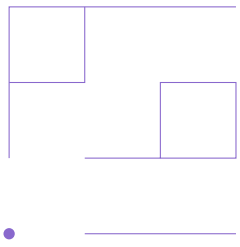
$$E_\sigma = \{uv \in E \mid \sigma(u) = \sigma(v)\}.$$

Now I can define an alternative noise process which sort of switches between two perspectives — one being assignments of ± 1's to vertices, and the other being subgraphs of my graph. So I define this alternative state space as $\Sigma = 2^E$ (the collection of all subsets of *edges* in this graph). For simplicity, to define $\mathcal{N}$, we'll start by sending each set σ to its unique set E_σ — so we define $\mathcal{N}$ to be given by sending σ to $E_\sigma \subseteq E$.

The picture is, suppose I have a graph (we draw a lattice). And let's say I have some set of assignments

$$\begin{bmatrix} + & + & + & + \\ + & + & - & - \\ + & - & - & - \\ - & + & + & + \end{bmatrix}.$$

We look at the edges that have the same assignments to both endpoints:



So these red edges are E_σ; and I'm mapping σ to this set of edges, and forgetting what the vertex assignments were.

Now let's think about what the recovery map looks like. This is now a subgraph of my graph, and I have a bunch of connected components (in this picture, I have four). And I have the constraint that all vertices within a component must receive the same assignment, but across components they may receive different assignments.

Claim 18.14 — For every $F \subseteq E$, we have

$$\mathcal{R}_\mu(F \to \infty) \propto \exp\left(\beta \sum_{uv \in E \setminus F} \sigma(u)\sigma(v)\right)$$

for all σ such that $F \subseteq F_\sigma$.

In other words, this becomes an Ising model on a new graph (with repeated edges) where I contract all of the vertices inside a component to a single vertex.

This process by itself isn't a particularly good one, because if the subset of edges F turns out to be empty (you got unlucky), then this is essentially the original Ising model you started out with, so you'd have to solve a problem that's at least as hard as sampling from the original distribution.

What'll turn out to be a better way to do this is not to literally spit out the set of edges E_σ, but to instead run *percolation*.

So we'll define a new noise process $\mathcal{N}$ given by *bond percolation* on (V, E_σ) (the set of monochromatic edges) with parameter p. In other words,

$$\mathcal{N}(\sigma \to F) = p^{|F|}(1-p)^{|E_\sigma \setminus F|}.$$

So I'm going to look at all the monochromatic edges, but then I'm going to throw away some of them (and I throw away an edge independently of the other edges).

Why would this be better? Now let's look at what the recovery map (or the posterior distribution) looks like. We'll have

$$\mathcal{R}_\mu(F \to \sigma) \propto \mu(\sigma) \cdot p^{|F|}(1-p)^{|E_\sigma \setminus F|}.$$

And now let's massage this a bit. Everything is up to some constant of proportionality, so up to some slight normalizations we get

$$\mathcal{R}_\mu(F \to \sigma) \propto \exp(2\beta \cdot |E_\sigma|) \cdot (1-p)^{|E_\sigma|}$$

(the first factor comes from the definition of μ; and F is fixed, so $p^{|F|}$ and $(1-p)^{|F|}$ are both constants). And now we see that if we set

$$p = 1 - e^{-2\beta},$$

then this expression is not going to depend on σ anymore! In particular, this is going to be independent of σ if we set $p = 1 - e^{-2\beta}$. In particular, what that tells us is that the recovery map $\mathcal{R}_\mu$ is just the *uniform* distribution over all distributions σ consistent with F (i.e., such that $E_\sigma \supseteq F$).

This gives us a very different Markov chain. And this Markov chain in general is going to have extremely global moves.

Algorithm 18.15 (Swendsen–Wang chain)

Suppose the current configuration is $\sigma \in \{\pm 1\}^V$.

(1) Compute the set of monochromatic edges E_σ. For every monochromatic edge $e \in E_\sigma$, keep it independently with probability $1 - e^{-2\beta}$ to get a random subset $F \subseteq E_\sigma$.

(2) Compute the connected components of (V, F); call them $\mathcal{C}_1, \ldots, \mathcal{C}_k$.

(3) Sample $\tau \sim \mathsf{Unif}\{\pm 1\}^k$, and assign τ_i to every vertex of the ith component $\mathcal{C}_i$.

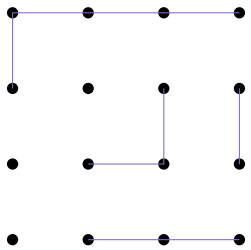
So each connected component corresponds to a set of vertices where we'll enforce all their assignments are the same; but otherwise we'll have complete independence across the components.

This is an extremely global Markov chain — for instance, it's possible to go from the all-$+1$'s configuration to the all--1's in a single step. This is very different from Glauber dynamics, which is an extremely local chain (it only looks at one vertex at a time).

Example 18.16

Suppose we start with the example from before.

Then when I sample, I might get the following.



So I have four big components, and three isolated vertices. Each of these components gets a uniformly random assignment, and that assignment goes to every one of its vertices (e.g., maybe the big component on the top becomes -1, and so on). Note that many assignments could have changed in this single step.

One reason we consider this Markov chain is that it has an advantage over something simple like Glauber dynamics: We're looking at the ferromagnetic Ising model, and we have this parameter β, which scales the strength of our interactions. If I have really large β I'm really preferring all vertices to have the same assignment; if I have really small β I have no interaction between the vertices, and each essentially gets a uniform independent coin flip.

What happens in the large β case? We haven't precisely discussed mixing times yet, so we'll only say things at an informal level (we'll discuss them more formally in a future lecture). But for large β, Glauber dynamics will have a very hard time going from a configuration where everyone is -1 to a configuration where everyone is $+1$ — more generally, it'll take $\exp(\Omega(n))$ time to go from predominantly $+1$ to predominantly -1. If we just think about how Glauber dynamics evolve, suppose I start from a configuration where everyone is $+1$. Then I pick a random vertex and look at all its neighbors. All my neighbors are voting $+1$, so I'm very likely to vote $+1$ as well. And this is expected to persist for exponentially long before I move to -1's.

On the other hand, because Swendsen–Wang is global and we can freely move between $+1$'s and -1's, we can actually get fast convergence *always*.

Theorem 18.17 (Guo–Jerram 2014)

For any G and any β, any initial distribution μ_0, and any $\varepsilon > 0$, we have

$$\left\|\mu_0 P^t - \mu\right\|_{\mathsf{TV}} \leq \varepsilon$$

with $t \leq \mathsf{poly}(|V|, \log(1/\varepsilon))$ (where P denotes the Swendsen–Wang chain).

(It turns out proving these kinds of statements is a very hard thing, so this was only proved recently.)

Conjecture 18.18 (Peres) — Taking $t = O(n^{1/4}\log(1/\varepsilon))$ is sufficient.

The polynomial in the theorem is quite bad — n^{10} or something — but at least in the realm of polynomial-time algorithms, this already separates Swendsen–Wang from a naive chain like Glauber dynamics.

What we really want to emphasize here is that this other scheme of constructing Markov chains, by viewing things through the lens of statistical inference, is very powerful; and it can allow you to design algorithms that overcome bottlenecks like this (the one between $+1$'s and -1's).

Remark 18.19. one advantage of something like Glauber dynamics is that it's totally generic — it doesn't use anything specific about the Ising model, and it works for any distribution on the cube — while something like this really makes use of the specific structure of the ferromagnetic Ising model.

§18.3 Proof of the fundamental theorem of Markov chains

In the last 20 or so minutes, we'll make good on the promise from last lecture to show us a proof of the fundamental theorem — that if P is ergodic, then it has a unique stationary distribution μ, and for every initial distribution μ_0, we have $\mu_0 P^t \to \mu$ as $t \to \infty$. We'll measure things in total variation, so we'll show that

$$\left\|\mu_0 P^t - \mu\right\|_{\mathsf{TV}} \to 0.$$

§18.3.1 Couplings of Markov chains

We'll need one tool, which is *coupling* (which we'll also use in the next lectures). We've seen coupling before in the context of coupling distributions, but now we want to discuss coupling Markov chains.

Definition 18.20. Let P_X and P_Y be two Markov chains on Ω. A *coupling* of P_X and P_Y is a stochastic process $\{(X_t, Y_t)\}_{t=0}^{\infty}$ such that for all t and all $a, b \in \Omega$, we have

$$\mathbb{P}[X_{t+1} = b \mid X_t = a] = P_X(a \to b) \quad \text{and} \quad \mathbb{P}[Y_{t+1} = b \mid Y_t = a] = \mathbb{P}_Y[a \to b].$$

So this says marginally if I only look at X_t, it's going to evolve according to P_X; and if I only look at Y_t, it's going to evolve according to P_Y.

Basically what we're going to try to do is use the notion of coupling to certify a bound on this total variation distance.

Before we get there, we'll give a different way to think about the notion of coupling: a *Markovian* coupling is a collection of couplings $Q((x, y) \to \bullet, \bullet)$ between $P_X(x \to \bullet)$ and $P_Y(y \to \bullet)$ (with one for each $x, y \in \Omega$). In other words, one way to design a coupling is to just say, for every pair of possible states, I design a coupling between the corresponding transitions — each row of P_X and P_Y is a probability distribution, so I can couple them in the usual sense of couplings between distributions.

Here's a lemma (it's really just a corollary of the coupling lemma for distributions). We want to consider a coupling between two Markov chains, which are both evolving according to the same transition matrix P but start with different initial distributions.

Lemma 18.21

If $\{(X_t, Y_t)\}$ is a coupling between two Markov chains $\{X_t\}_{t=0}^{\infty}$ and $\{Y_t\}_{t=0}^{\infty}$ both evolving according to P and where $X_0 \sim \mu_0$ anad $Y_0 \sim \mu$ (where μ is stationary), then

$$\left\|\mu_0 P^t - \mu\right\|_{\mathsf{TV}} \le \mathbb{P}[X_t \neq Y_t].$$

So I can upper-bound the distance from stationary by the collision probability under *any* coupling. These two chains are both marginally evolving according to P, except that I've initialized them at different distributions — one at this initial distribution μ_0, and the other is already initialized at stationary (in particular, we always have $Y_t \sim \mu$, and $X_t \sim \mu_0 P^t$, for all t).

This lemma is really just a direct consequence of the usual coupling lemma for distributions. But now we can reinterpret our task for bounding the total variation distance as designing a coupling between the two runs of the chain, such that we maximize the probability that these two chains collide. And that's what we're going to do to prove the fundamental theorem; and we're also going to use this as a tool to bound *rates* of convergence.

Remark 18.22. We only have 5 minutes left, and we don't want to rush the proof of the fundamental theorem, so we'll leave that for next lecture and adjourn for the day.

§19 April 14, 2025 — Markov chain mixing times

PS4 is due tonight, and PS5 should be out now on Canvas and will be due in 2 weeks.

Today we'll continue discussing Markov chains; in particular, this lecture we'll look at rates of convergence to stationary. We'll also complete the proof of the fundamental theorem, using the notion of couplings.

§19.1 Markovian couplings

We'll define couplings of Markov chains in a special case that will be enough for our purposes.

Definition 19.1. Let P be the transition matrix for a Markov chain on state space Ω. A *Markovian coupling* for P is a collection of couplings $\{\xi_{uv}\}_{u,v\in\Omega}$ (in the standard distribution sense) where the marginals of ξ_{uv} are $P(u \to \bullet)$ and $P(v \to \bullet)$, respectively.

In other words, the way you should think about what a Markovian coupling is as a way to simulate a *pair* of Markov chains, each individually distributed according to the standard chain P, but which are correlated or coupled in some way.

So maybe I have two views; both are Markov chains, which we'll call $\{X_t\}_{t\geq 0}$ and $\{Y_t\}_{t\geq 0}$. Both of these are going to be evolving according to this Markov chain P — so maybe in the first view I start at X_0 and walk around in this big box Ω, and similarly in the second view for Y. And if I look at any one of these boxes individually, it looks just like a usual run of the Markov chain (given by P). But whewn I look at both overlaid on top of each other, I realize these two trajectories are evolving in a correlated fashion — maybe these two pictures somehow resemble each other.

So a coupling of the Markov chain gives rise to $\{(X_t, Y_t)\}_{t\geq 0}$ where marginally, $\{X_t\}_{t\geq 0}$ and $\{Y_t\}_{t\geq 0}$ both evolve according to P.

Another way to think about a Markovian coupling is that it's actually defining a Markov chain whose state space is given by $\Omega\times\Omega$; to evolve this chain, if my current *pair* of states is at (X_t, Y_t), then to evolve to the next state (X_{t+1}, Y_{t+1}), I sample from the corresponding coupling at (X_t, Y_t).

Recall the following lemma, which we stated at the end of last lecture:

Lemma 19.2

If μ_0 and ν_0 are any pair of initial distributions and we let $\{(X_t, Y_t)\}_{t\geq 0}$ evolve according to a Markovian coupling with the initial conditions $X_0 \sim \mu_0$ and $Y_0 \sim \nu_0$, then we have

$$\left\|\mu_0 P^t - \nu_0 P^t\right\|_{\mathsf{TV}} \leq \mathbb{P}[X_t \neq Y_t].$$

Proof. The proof is really just by the standard coupling lemma, applied to couplings between distributions — we have $X_t \sim \mu_0 P^t$ and $Y_t \sim \nu_0 P^t$, so they have the correct marginal distributions, and we can just apply the coupling lemma. (You can interpret the total variation distance as picking the best possible coupling between the laws of X_t and Y_t, and asking for the probabilities that they're not equal.) □

This is the connection between couplings and studying rates of convergence to stationary; we'll be more precise about this in a moment.

§19.1.1 An example — walks on the hypercube

Before we get there, let's see an example of what a Markovian coupling might look like.

Example 19.3

Let $\Omega = \{\pm 1\}^n$ be the hypercube, and let $\mu = \mathsf{Unif}(\Omega)$ and let P be the Glauber dynamics. So P evolves by the following:

- Pick a uniformly random coordinate i.
- Pick a uniform random $s \in \{\pm 1\}$.
- Update $\sigma_i \leftarrow s$.

So P is a simple (lazy) random walk on the hypercube — at each step, I pick a random coordinate, and take a step by updating that coordinate randomly.

To get a Markovian coupling, I want to evolve the two chains so that marginally they're evolving according to Glauber dynamics, but the evolution between them may be correlated.

Of course, one way to do this is to let them evolve independently — I can say X_{t+1} picks its own uniform random coordinate and update, and Y_{t+1} independently does the same thing. So I can always evolve Markov chains independently, and that will give me *a* coupling, though not a very interesting one.

One that's more interesting, and will be more useful, is to instead pick the *same* coordinate and the *same* choice of update.

Example 19.4

To get a coupling for P, suppose we're currently at (X_t, Y_t). Then:

- Pick a single $i \sim \mathsf{Unif}[n]$.
- Pick a single $s \sim \mathsf{Unif}\{\pm 1\}$.
- Update $X_{t+1}(i) = Y_{t+1}(i) = s$, and leave all other coordinates unchanged.

So I just pick a uniformly random coordinate, and use that *same* coordinate for both copies of the chain; and I pick a uniformly random update, and also use that same update for both copies of the chain.

If you look at things from just the perspective of the X_t's and totally ignore the Y_t's, I'm just picking a random coordinate and update, so I'm evolving marginally according to the Glauber dynamics; and the same is true for the Y_t's. But if I look at both simultaneously, I see that something interesting is happening — namely, the number of disagreements between X_t and Y_t is shrinking over time.

§19.2 Proof of the fundamental theorem

Now we'll use the concept of coupling to prove the fundamental theorem.

Theorem 19.5

If P is ergodic, then P has a unique stationary distribution μ, and we have $\left\|\mu_0 P^t - \mu\right\|_{\mathsf{TV}} \to 0$ as $t \to \infty$ (for all μ_0).

The most important part of this is the convergence — if you have the convergence, then you automatically get uniqueness of the stationary distribution, for free.

The key lemma will be the following, which gives an alternative way of thinking about ergodicity.

Lemma 19.6

If P is ergodic, then there exists $t^* > 0$ such that $P^{t^*}(x \to y) > 0$ for all $x, y \in \Omega$.

We'll prove this key lemma in a moment (it really just follows from the definition of ergodicity, plus maybe a bit of elementary number theory). But now we'll use it to prove the theorem. To do this, we'll design a coupling.

Another way to think about this key lemma is that if I iterate the Markov chain enough times, then the underlying graph of the iterated Markov chain eventually becomes a complete graph — where all pairs of states can reach each other in a single step.

Proof. To show that Lemma 19.6 implies the theorem, we'll show an even stronger inequality. For convenience, define $Q = P^{t^*}$ — so Q is going to be a new Markov chain, satisfying the nice property that all entries of Q are strictly positive. Then Q is also going to have μ as its stationary distribution. If under iterations of Q we have convergence to μ, then we also have convergence under iterations of P itself; so it suffices to show that $\mu_0 Q^t \to \mu$ as $t \to \infty$, for all μ_0.

Now we're going to use this key lemma. We're going to show something even stronger — we'll show that there exists $\varepsilon > 0$ such that for all pairs of distributions μ_0 and ν_0, we have

$$\|\mu_0 Q - \nu_0 Q\|_{\mathsf{TV}} \leq (1 - \varepsilon) \|\mu_0 - \nu_0\|_{\mathsf{TV}}.$$

In other words, under every single step of my chain Q, I'm making substantial progress towards stationary — the distance decays by a constant multiplicative factor in every step (i.e., it decays exponentially fast).

To prove this, let's look at the following coupling: I want to construct a Markovian coupling for this new Markov chain Q which certifies this inequality. What we'll do is the following: Say X_t and Y_t are the current states. What we'll do is almost essentially the trivial (independent) coupling, but there's a slight twist — we'll make these two Markov chains stick to each other for all time, once they've collided.

- If $X_t = Y_t$, then sample $X_{t+1} \sim Q(X_t \to \bullet)$ (as usual), and set $Y_{t+1} = X_{t+1}$.

  So once the two copies of the chain have collided, they'll stay stuck to each other until the end of time.
- Otherwise, we'll just do the trivial thing and let them evolve independently of each other — if $X_t \neq Y_t$, then we evolve them independently according to Q.

Another way to describe this is that if I look at a pair of states u and v which are equal, then I use the identity coupling (which guarantees that the pair of outcomes sampled from this coupling are always equal). And if $u \neq v$, then I use the independent coupling between those two transition distributions.

Now let's prove that contraction inequality. Let ξ_0 be a TV-optimal coupling between μ_0 and ν_0 (here when we say TV-optimal, we mean this is the coupling for which the probability that when I sample a pair of random variables they're not equal is exactly equal to the total variation distance — i.e., $\mathbb{P}_{(X,Y)\sim\xi}[X \neq Y] = \|\mu_0 - \nu_0\|_{\mathsf{TV}}$; such a coupling is guaranteed to exist by the coupling lemma).

Now let's look at $\|\mu_0 Q - \nu_0 Q\|_{\mathsf{TV}}$. I want to bound this thing by $(1-\varepsilon)\|\mu_0 - \nu_0\|_{\mathsf{TV}}$. So now let's use our coupling — by the coupling lemma, this is at most

$$\|\mu_0 Q - \nu_0 Q\|_{\mathsf{TV}} \le \mathbb{P}[X_1 \neq Y_1]$$

(with respect to the coupling we just constructed). And we can write this as

$$\mathbb{P}[X_1 \neq Y_1 \mid X_0 \neq Y_0] \cdot \mathbb{P}[X_0 \neq Y_0].$$

The second term here is very nice — it's exactly $\|\mu_0 - \nu_0\|_{\mathsf{TV}}$, by the fact that we chose ξ_0 to be a TV-optimal coupling.

Now we just need to say that the conditional proability $\mathbb{P}[X_1 \neq Y_1 \mid X_0 \neq Y_0]$ is at most $1-\varepsilon$. And this is essentially just a consequence of the key lemma — no matter what I choose X_1 to be (sampled from the transition distribution under Q), Y_1 always has at least ε probability of hitting X_1, if we take ε to be the smallest entry of Q, i.e.,

$$\varepsilon = \min_{x,y} P^{t^*}(x \to y) > 0.$$

So the probability that they're equal is at least ε, which means the probability they're not equal is at most $1-\varepsilon$. And that's it. □

Student Question. *We're assuming there's a finite number of states?*

Answer. Yes.

Now we just need to prove the key lemma, and then we really have the fundamental theorem.

Proof of Lemma 19.6. The basic idea behind the key lemma is to reduce this statement to a statement just for the diagonal entries of P. So the key sublemma is the following:

Claim 19.7 — There is $s^* > 0$ such that for every $s \ge s^*$ and every $x \in \Omega$, we have $P^s(x \to x) > 0$.

So there is some threshold s^* such that for all time after that threshold, I always have some positive probability of staying at the current state (after that number of steps).

Let's first see why this is sufficient: If I look at $P^t(x \to y)$, I can always lower-bound this by

$$P^t(x \to y) \ge P^s(x \to x) P^{t-s}(x \to y).$$

(If I want to look at the probability of going from x to y after t steps, I can lower-bound it by the probability of staying at x after s steps, and then going to y in the remaining $t-s$ steps.) And I can guarantee that for all $s \ge s^*$ the first term is positive, and I know that at some point $P^{t-s}(x \to y)$ is also going to be positive by irreducibility — i.e., $P^{t-s}(x \to y)$ is positive for some $t-s$, and $P^s(x \to x)$ is positive for all $s \ge s^*$. This will mean there exists $t^*(x,y)$ depending on x and y such that $P^t(x \to y) > 0$ for all $t \ge t^*(x,y)$; and then I can take the maximum of all these $t^*(x,y)$'s to get a t^* that doesn't depend on x or y.

So we've reduced the lemma to just a claim about the *diagonal* entries of the powers of our matrix; the key that makes all this work is that we need a statement for *every* $s \ge s^*$.

And now this is where we're going to use aperiodicity. By irreducibility, there is some $j > 0$ such that $P^j(x \to x) > 0$. And by aperiodicity, there must exist $j, k > 0$ such that both $P^j(x \to x)$ and $P^k(x \to x)$ are positive, with $\gcd(j,k) = 1$. Now this essentially boils down to some number theory, which we're just going to appeal to as a black box.

In particular, what this means is that $P^s(x,x) > 0$ if we can write s as some nonnegative integer combination of j and k — i.e., if we can write $s = n \cdot j + m \cdot k$, where $m, n \in \mathbb{N}$. This is kind of essentially by the same claim from before — we have

$$P^s(x \to x) \ge (P^j(x \to x))^n (P^k(x \to x))^m.$$

So from this, we get that for all integers which can be written as some nonnegative integer combination of j and k, we have positivity.

And it turns out, by some number theory, that this will be true for all integers larger than some s^*:

Fact 19.8 (Schur's theorem) — If $\gcd\{j,k\} = 1$, then there exists s^* such that for all $s \geq s^*$, we can write $s = nj + mk$ for $m, n \in \mathbb{N}$.

We won't get into too much details on the number theory; the smallest s^* for which this is true is called the *Frobenius number*, and it's known that you can take $s = jk$. If you're interested in details, there's a very nice textbook on Markov chain mixing times which discusses this in greater detail; but we'll leave it at this. □

Student Question. *In the fundamental theorem, can we split the two conditions of ergodicity to show uniqueness vs. convergence?*

Answer. Yes. You can say that if the Markov chain is irreducible then the stationary distribution is unique; if it's aperiodic then you'll always have convergence to *some* stationary distribution. (This probably can't be seen from this proof, though, since we used both crucially.)

§19.3 Mixing times

Now we'll start discussing rates of convergence. We've seen many examples of Markov chains now. Generally speaking, ergodicity is a very weak property — you just need connectivity, and you can lazify to make it aperiodic. We also saw schemes to construct Markov chains which were ergodic and had the correct stationary distribution. But now we want to look at:

Question 19.9. How fast do we get to stationary?

The starting point is the following inequality, which says that under each application of the Markov chain, you're never going to get further away from stationary.

Lemma 19.10 (Data processing inequality)

For all μ and ν, we have

$$\|\mu P - \nu P\|_{\mathsf{TV}} \leq \|\mu - \nu\|_{\mathsf{TV}}.$$

This is essentially the same as the key lemma, but without the $1 - \varepsilon$ factor (and sometimes equality can hold, even for ergodic chains). But what this says is that Markov chains cannot 'unmix' a distribution.

With this lemma in hand, it makes sense to define a *mixing time* as follows:

Definition 19.11. For a parameter $\varepsilon > 0$, we define the *mixing time*

$$T_{\mathrm{mix}}(\varepsilon; P, \mu_0) = \min\{t \geq 0 \mid \left\|\mu_0 P^t - \mu\right\|_{\mathsf{TV}} \leq \varepsilon\}$$

(where μ is the stationary distribution).

Here $\varepsilon > 0$ is a parameter essentially quantifying how close to stationary you want the chain to be.

The above lemma says once you've reached a distribution which is within ε of stationary, you'll remain within ε of stationary for all following times; so it makes sense to look at the *first* time when this happens.

Typically we're going to drop the dependence on the initial distribution, and just look at the *worst* choice:

Definition 19.12. We define $T_{\text{mix}}(\varepsilon) = \sup_{\mu_0} T_{\text{mix}}(\varepsilon; P, \mu_0)$. We also define the *mixing time* as $T_{\text{mix}}(1/4)$.

The reason for this latter definition is that it turns out once you hit a distance $1/4$ from stationary, then to get within an ε distance of stationary, you just have to pay an additional multiplicative factor of $\log(1/\varepsilon)$. You'll show this on the homework:

Exercise 19.13. We have $T_{\text{mix}}(\varepsilon) \lesssim T_{\text{mix}}(1/4) \cdot \log(1/\varepsilon)$.

So that's why we say $T_{\text{mix}}(1/4)$ is 'the' mixing time, with no quantification over ε.

§19.4 Mixing times for the random walk on the hypercube

Now let's look at mixing times for Glauber dynamics on $\mathsf{Unif}\{\pm 1\}^n$. I want to use the coupling we discussed earlier to show that the TV distance between any two copies of this Markov chain (the simple random walk on the hypercube) is going down geometrically. Basically, I want to bound the probability that the two copies of the chain haven't yet collided.

The key observation is that because in every single step, we're picking the same coordinate to update and the same choice of update to that coordinate — once two copies of the chain agree in some coordinate, they agree in that coordinate forever. So the set of all disagreeing coordinates is shrinking; and we want to say that set gets to the empty set as fast as possible.

So we want the set of disagreeing coordinates between X_t and Y_t to shrink to 0 rapidly. In particular, let's let μ_0 and ν_0 be arbitrary initial distributions. and let $X_0 \sim \mu_0$ and $Y_0 \sim \nu_0$. And let's look at $\left\|\mu_0 P^t - \nu_0 P^t\right\|_{\mathsf{TV}}$. (Imagine for instance taking ν_0 to be the stationary distribution, so that the second term is just the stationary distribution.) Again, by our coupling, we have

$$\left\|\mu_0 P^t - \nu_0 P^t\right\|_{\mathsf{TV}} \leq \mathbb{P}[X_t \neq Y_t].$$

And we know $X_t \neq Y_t$ if and only if there's at least one coordinate where they still disagree; this can only occur if that coordinate has *never* been picked in the entire sampling process (since once that coordinate is picked, I'll be updating it to the same thing forever). So this is at most

$$\mathbb{P}[\text{exists } i \in [n] \text{ which has not been picked by time } t].$$

And now I'm looking at the probability of there existing some event, so we can use a union bound; this is at most

$$\sum_{i \in [n]} \mathbb{P}[i \text{ is not picked by time } t].$$

And now this is something that's easy to calculate — the only way i isn't picked is if at every single step it's not picked; and this occurs independently with probability $1 - \frac{1}{n}$ at each step. So this is equal to

$$n\left(1 - \frac{1}{n}\right)^t,$$

since at every time we're doing an independent trial.

And the nice thing about this bound is it's completely independent of the initial distributions μ_0 and ν_0 we started with. And we also see that if I want this to be at most ε, then I should take $t \asymp (n \log(n/\varepsilon))$; and this will guarantee that it's at most ε.

So this shows that $T_{\text{mix}}(\varepsilon) \leq O(n \log(n/\varepsilon))$.

And this analysis is essentially sharp. This is a classic instantiation of the *coupon collector* problem — you have n coupons and you're picking a random coupon, and you want to ensure you've picked every coupon at

least once. It's known that it requires at least $n \log n$ steps to guarantee that with good probability you've picked everything.

So this is almost linear time for sampling (though this distribution is not so interesting to sample from). This is the gold standard for Markov chains that are like this (where you pick a single coordinate and update it), because you have to at least pick every coordinate at least once; so this is what we'll be aiming for in general.

§19.5 Debrushin influence

Now we'll talk about how to prove these types of mixing time bounds (which we call *optimal mixing* results) for distributions which no longer have joint independence. We'll look specifically at distributions coming from graphical models like independent sets and coloring and so on.

We'll define a possibly crazy-looking quantity, but we'll see in a moment that it's not actually so scary.

Definition 19.14. Fix a distribution μ on $[q]^n$. For $i \neq j$, define the *Debrushin influence*

$$\mathcal{R}_\mu(i \to j) = \max_{\tau : [n] \setminus \{i,j\} \to [q]} \max_{b,c \in [q]} \|\mu_j^{\tau, i \leftarrow b} - \mu_j^{\tau, i \leftarrow c}\|_{\mathsf{TV}}.$$

So we have a distribution μ on assignments of colors $1, \dots, q$ to coordinates $1, \dots, n$.

Debrushin was a mathematical physicist who used this quantity to study various properties of Gibbs distributions, like correlation decay.

This is some very scary-looking quantity, but the picture to have is that I pin the colors almost all the coordinates (all except i and j) to some τ; and I consider what happens i with either the colors b or c, and I see how that changes the distribution of the assignment to j. This is why it's called the *influence* — we're looking at how the color assigned to i influences the color assigned to j.

For this notation, we use subscripts to denote marginalization and superscripts to denote conditioning. So by $\mu_j^{\tau, i \leftarrow b}$, this means the distribution on colors $1, \dots, q$, where

$$\mu_j^{\tau, i \leftarrow b}(a) = \mathbb{P}_{\sigma \sim \mu}[\sigma(j) = a \mid \sigma(i) = b \text{ and } \sigma(k) = \tau(k) \text{ for all } k \neq i, j].$$

In other words, I'm conditioning on my sample agreeing with everything in the superscript, and I'm looking at what color is assigned to j.

So really, this influence looks at how much the color assigned to i changes the distribution of the color assigned to j.

We'll see some examples of how we can compute this thing, though it may look daunting right now.

Example 19.15

If μ is uniform — more generally, if it's a product distribution, i.e., one where all coordinates are independent — then $\mathcal{R}_\mu(i \to j) = 0$.

You should think of the Debrushin influence as some kind of 'correlation' between coordinates i and j (though it's not a correlation in the usual sense of covariances and so on, it's essentially trying to quantify the same thing).

Here's the theorem (we won't have enough time to prove it, so it has been put in our homework).

Theorem 19.16

If $\max_i \sum_{j \neq i} \mathcal{R}_\mu(i \to j) < 1 - \delta$ for some $0 < \delta < 1$, then Glauber dynamics satisfies

$$T_{\min}(\varepsilon) \leq O\left(\frac{n}{\delta} \log\left(\frac{n}{\varepsilon}\right)\right).$$

For instance, if μ is once again uniform over all possible colorings, then this essentialy recovers the $n \log n$ analysis we did earlier. But this allows for a much larger class of distributions; we just need to ensure that the coordinates of this distribution are not too strongly correlated with each other, and that'll be enough to get essentially the same quantitative dependence for the mixing time.

Student Question. *What is μ?*

Answer. We're fixing some distribution μ, and we're saying that if it satisfies this property (that this quantity is less than 1), then if I run Glauber dynamics for μ, it satisfies this mixing time inequality. (Glauber dynamics means that at each step, I pick a uniform random coordinate and resample it conditioned on everyone else.)

We're not going to prove this; one way to prove it is by a coupling argument, which you'll do in the homework. But we'll see some examples of how to compute this matrix $\mathcal{R}_\mu$, and hopefully convince us that computing this matrix is not so frightening.

§19.5.1 Some examples

One of the first observations is that actually, this matrix has a very nice 'sparsity' structure, especially if your distribution comes from a graphical model.

Lemma 19.17

If μ is a Markov random field on a graph $G = (V, E)$, then for all $i \neq j$ with $ij \notin E$, we have

$$\mathcal{R}_\mu(i \to j) = 0.$$

For example, think of proper colorings, or the Ising model, or independent sets (or the hardcore model). All you really need is the *global Markov property* which we discussed in a previous lecture.

In other words, you can think of the Debrushin influence matrix as really just being a weighted, possibly directed, version of the adjacency matrix of your underlying graph — if your underlying graph is sparse, then the influence matrix will also be sparse, and it'll be easier to satisfy an inequality like the one in the theorem.

Proof. We'll prove this just by picture. Suppose my vertices i and j are not connected by an edge; so maybe I have j somewhere, and it has some neighbors, and then i is somewhere off to the left (not in this neighborhood); maybe it has some paths to j, but this doesn't matter.

In my Debrushin influence, I'm first pinning all vertices except i and j; in particular, I've pinned all vertices in the immediate neighborhood of j. So τ pins the immediate neighborhood of j.

And the immediate neighborhood of j is a separator between i and j — because we've assumed i is not in this neighborhood. So in particular, once I've pinned everything in j's neighborhood, it's not going to see however you change the color assignment to i. You can think of this pinning τ as shielding the effect of i on j.

And that's it — $N(j)$ is a separator between i and j, and then we can use the global Markov property or conditional independence — that the assignment to i and assignment to j are independent once I've conditioned on everything in a separator. □

So this matrix was defined in a very complicated way, but it has a very nice sparsity structure.

Now we'll do a couple of examples, and then wrap up.

Example 19.18

Consider the Curie–Weiss model (the ferromagnetic Ising model on K_n), where

$$\mu(\sigma) \propto \exp\left(\frac{\beta}{2n}\langle \sigma, 1\rangle^2\right).$$

Then for all $i \neq j$, we have

$$\mathcal{R}_\mu(i \to j) \leq \tanh\frac{\beta}{n} \leq \frac{\beta}{n}.$$

In particular, if $\beta < 1$, then Glauber dynamics mixes in $O(n \log n)$ steps.

The calculation of this bound is in the notes; it's not too difficult, but we don't have time.

This result is actually sharp — we'll see next lecture that if $\beta > 1$, then Glauber dynamics requires $\exp(\Omega(n))$ time. That's because if you look at what this distribution looks like, if I plot the mass function for the random variable $\langle \sigma, 1\rangle$ (called the *magnetization*), when $\beta < 1$ this looks like a very nice Gaussian-like unimodal distribution (it's actually sub-Gaussian). And when $\beta > 1$, it looks very bimodal. This bimodality is going to prevent a local Markov chain from mixing — it'll take you exponentially long to move from exploring states in one bump to states in the other bump.

Example 19.19

Let μ be the hardcore model, where $\mu(S) \propto |\lambda|^{|S|}$ for independent sets $S \subseteq V$, where $G = (V, E)$ has maximum degree Δ. Then

$$\mathcal{R}_\mu = \frac{\lambda}{1+\lambda} \cdot A_G.$$

In particular, if $\lambda < \frac{1}{\Delta - 1}$, then Glauber dynamics mixes in $O(n \log n)$ time.

(We use $\mathcal{R}_\mu$ to denote the entire Debrushin influence matrix.)

Proof. By the lemma, we only need to consider the case where i and j are neighbors. The only way that the TV distance can be nonzero is if all the neighbors of j are pinned to be in the independent set (since if any of them is in the set, then j cannot be in the set). So I only need to look at a very simple graph (where there's i and j and some stuff not adjacent to j, and I consider pinning i to be in or out of the set), and for that graph can do the computation and get $\frac{\lambda}{1+\lambda}$. □

Remark 19.20. For comparison, the computational phase transition for sampling occurs at roughly $\frac{e}{\Delta - 1}$. So we're already getting within a constant factor from the threshold where you can expect *any* algorithm to sample from the distribution.

The key takeaway here is that there's actually fairly simple, very versatile, criteria that can guarantee fast mixing, even when your distribution doesn't have jointly independent coordinates.

§20 April 16, 2025 — Spectral gaps and the conductance method

§20.1 Issues with total variation distance

Last time, we started talking about mixing times of Markov chains — in other words, bounding rates of convergence to stationary.

Definition 20.1. For a Markov chain P with stationary distribution μ, the *total variation mixing time* is defined as $\mathsf{T}_{\mathsf{mix}}(\varepsilon) = \sup_{\mu_0} \min\{t \geq 0 \mid \|\mu_0 P^t - \mu\|_{\mathsf{TV}} \leq \varepsilon\}$.

So we take the worst possible starting distribution μ_0, and ask for the first time such that if you apply your Markov chain for that many steps, you're within ε in total variation distance.

Last class, we saw a useful technique for upper-bounding mixing time, based on *coupling*.

Example 20.2

For the (lazy) random walk on $\{\pm 1\}^n$, we have $\mathsf{T}_{\mathsf{mix}}(\varepsilon) \leq O(n \log(n/\varepsilon))$.

Here, we're really looking at convergence to stationary as measured by total variation distance. This is very natural because this kind of guarantee means that if I sample $X_t \sim \mu_0 P^t$ (i.e., I run my chain for t steps), then for any 1-bounded function f, I'll have

$$|\mathbb{E}[f(X_t)] - \mathbb{E}_\mu[f]| \leq \varepsilon.$$

In particular, if you have mixing in total variation, you can use this for various estimation tasks (computing the expectations of various functions, e.g., marginals of coordinates and so on).

So measuring mixing time through total variation distance is very natural.

On the other hand, it turns out total variation distance also isn't super nice to work with in general. Even for very simple and nice Markov chains like the simple random walk on the hypercube, the decay of the total variation distance can behave in a very irregular manner.

For instance, we'll draw a plot of how total variation decays as time progresses (with t on the x-axis and the total variation distance on the y-axis). So P denotes the simple random walk on the hypercube $\{\pm 1\}^n$.

It turns out that what this looks like is it'll stay at nearly 1 for a long time; then we'll hit some threshold at $\frac{1}{2} n \log n$. It's only here where we start making significant progress. Then the mixing time starts dropping very rapidly; there's a window of length $O(n)$ (which is much smaller), and by this point you'll have gotten very close to stationary.

So if you measure progress in terms of total variation distance, you'll find that in the first $\frac{1}{2} n \log n$ steps, the Markov chain has done basically nothing (its total variation distance from stationary is still nearly 1); and then in this small window, it makes very rapid progress.

This is a well-known phenomenon called the *cutoff phenomenon*, which we don't have time to discuss; it's very interesting in its own right. But it suggests that looking at total variation distance in terms of analyzing rates of convergence sometimes isn't so nice.

In an ideal world, what we'd like to say is that *every* step of the Markov chain, no matter what, is making substantial progress towards stationary. Ideally, we'd want to show something like

$$\mathrm{dist}(\mu_0 P^t, \mu) \leq (1 - \delta) \cdot \mathrm{dist}(\mu_0 P^{t-1}, \mu) \quad \text{for all } t \geq 0$$

(for some $\delta > 0$). This is the kind of thing we did when we proved the fundamental theorem of Markov chains — we showed there was a time t^* such that if you looked at the Markov chain P^{t^*}, that chain satisfies an inequality like this. You'd think of this as substantial progress in *every* step.

Student Question. *The place where the cutoff occurs, is it related to t^*?*

Answer. Not quite — already after $O(n)$ steps you'll have that all entries are lower-bounded by something positive, but it'll be really small. The way we defined t^* was such that $P^{t^*}(x \to y) > 0$ for all x and y. But this could be tiny, e.g., $\frac{1}{n^3}$ or something. So it's not quite the same, but you can think of it as intuitively similar.

Basically what the above plot says is that we can't hope for this ideal inequality for TV distance — which we call *contraction* — even for very natural and simple Markov chains. This leads to the following generic theme in the analysis of Markov chains:

Goal 20.3 (Theme)**.** Prove contraction for some other 'distance,' and then at the end, compare this distance with the total variation distance.

We put 'distance' in quotes because it doesn't actually have to be a metric (e.g., satisfy the triangle inequality); it's just some notion of distance.

Remark 20.4. One other reason total variation distance isn't super nice to work with is that it turns out there are Markov chains for which the mixing time is polynomial in n — so it is rapidly mixing — but there's provably no Markovian coupling that can certify this fact! So this is a deficiency behind the technique we used in the previous lecture.

There exist Markov chains with $\mathsf{T}_{\mathsf{mix}}(\varepsilon) \le \mathsf{poly}(n, \log(1/\varepsilon))$, but for which *every* Markovian coupling requires $\exp(\Omega(n))$ time to coalesce.

This isn't necessarily a deficiency of total variation distance itself, but it's a deficiency of the technique we used last lecture.

And these Markov chains aren't some weird contrived things — there are simple and natural ones based on perfect matchings in a graph.

The goal of this lecture is to illustrate this technique, and also show techniques for *lower*-bounding mixing times — e.g., to show a Markov chain requires exponential time to converge to stationary.

§20.2 The χ^2 divergence

Definition 20.5. The χ^2-divergence is defined as

$$\chi^2(\mu \,\|\, \nu) = \mathbb{E}_{x \sim \mu}\left[\left(\frac{\nu(x)}{\mu(x)} - 1\right)^2\right].$$

This is a different way of measuring the 'distance' between two distributions; it's not a metric in the formal sense, because it's not symmetric.

The χ^2-divergence can be related to TV distance:

Lemma 20.6

We always have

$$\|\mu - \nu\|_{\mathsf{TV}}^2 \le \frac{1}{4}\chi^2(\nu \,\|\, \mu).$$

This is essentially by Cauchy–Schwarz. Maybe it's kind of believable because χ^2 divergence looks a bit like TV distance — if we replaced the square with an absolute value, it literally would be TV distance.

We'll instantiate this theme with χ^2 divergence, and this leads to a very nice connection to linear-algebraic properties of the transition matrix.

Goal 20.7. Study the decay of χ^2.

We're going to use this to prove both upper and lower bounds on mixing time.

§20.3 Linear algebraic setup

Before we get there, we'll set up a bit of linear algebraic notation. We'll define an inner product between functions on our state space, which is going to depend on our stationary distribution μ. It's essentially the usual inner product, but reweighting all the states by μ:

Definition 20.8. We define $\langle f, g\rangle_\mu = \mathbb{E}_{x\sim\mu}[f(x)g(x)]$.

Throughout this class, we'll be working just with the class of *reversible* Markov chains, because of the following reason.

Fact 20.9 — If P is reversible with respect to μ, then for all functions f and g, we have

$$\langle f, Pg\rangle_\mu = \langle Pf, g\rangle_\mu.$$

So even though the transition probability matrix P is in general asymmetric (so it's not self-adjoint with respect to the *standard* Euclidean inner product), it's always self-adjoint with respect to this weighted one. This is actually essentially equivalent to reversibility — if we plugged in a function f which is just 1 on one state x and 0 everywhere else, and g which is 1 on y and 0 everywhere else, then this is the same as the definition of reversibility — i.e., if we plug in $f = \mathbf{1}_x$ and $g = \mathbf{1}_y$, then this exactly says that

$$\mu(x)P(x \to y) = \mu(y)P(y \to x).$$

In particular, this implies that P has *real* eigenvalues, and we can order them $\lambda_{|\Omega|} \le \cdots \le \lambda_1$.

What we're going to show is that these eigenvalues — really, the second-largest one — are going to control the rate of decay for the χ^2-divergence.

Here's another fact, which is that through the eigenvalues, we can also give an equivalent characterization for properties like irreducibility, aperiodicty, and ergodicity.

Fact 20.10 — We always have $\lambda_1 = 1$, and $|\lambda_i| \le 1$ for all i. Furthermore:

- P is irreducible if and only if $\lambda_2 < 1$.
- P is aperiodic if and only if $\lambda_{|\Omega|} > -1$.
- P is ergodic if and only if $\max\{|\lambda_2|, |\lambda_{|\Omega|}|\} < 1$.

This is already a very neat connection between something purely combinatorial or probabilistic, and a linear algebraic way of understanding things. We won't do a full proof of this, but we'll sketch a couple of the interesting directions. Perhaps the most relevant are that if $\lambda_2 < 1$ then the chain is irreducible, and if $\lambda_{|\Omega|} > -1$ then the chain is aperiodic. We'll prove the contrapositive of the two above statements.

Claim 20.11 — If P is not irreducible, then $\lambda_2 = 1$.

Proof. The way to see this is to think about what it means for P to be not irreducible. We're assuming P is reversible, so we can think about it as a simple random walk on an undirected graph; irreducibility is equivalent to that graph being connected. So if P is not irreducible, that means it has at least two connected components. Each of these components will have its own stationary distribution — a distribution which is supported only on that component and 0 everywhere else, and is stationary. And you have two of these; that means the eigenspace corresponding to eigenvalue 1 has dimension at least 2 (spanned by those distributions — all linear combinations of those two distributions also have eigenvalue 1), which means your second eigenvalue also has to be 1. □

Student Question. *How do we know the eigenvalues are real?*

Answer. We're assuming reversibility, so my Markov chain is self-adjoint with respect to an inner product, and that's enough for the eigenvalues to be real. For that, you can imagine this allows me to renormalize the matrix P by hitting it on both sides by the square root of a diagonal matrix, so that it becomes symmetric; and this renormalization doesn't change the fact that the eigenvalues are real.

Claim 20.12 — If P is not aperiodic, then $\lambda_{|\Omega|} = -1$.

Proof. Again, we're working with a reversible chain, in which case I'm looking at a random walk on an undirected (possibly weighted) graph; aperiodicity is equivalent to this underlying graph being *non*-bipartite. In other words, if I'm assuming the chain is not aperiodic, then the underlying graph *is* bipartite. And now I want to use this bipartiteness to construct an eigenvector whose eigenvalue is -1.

To do this, say I have my bipartite graph with bipartition $L \cup R$, and the transitions of the chain are supported on edges crossing this bipartition (with no edges within either side). Then I can take the function which is $+1$ on everyone on the left side, and -1 on everyone on the right side. In other words, we define

$$f(x) = \begin{cases} +1 & \text{if } x \in L \\ -1 & \text{if } x \in R. \end{cases}$$

Then we have $Pf = -f$. □

The nice thing here is you can tell whether a reversible Markov chain is irreducible or aperiodic by looking at its eigenvalues. And the nice thing is you can make this quantitative — you can show that if the eigenvalues are *bounded* away from 1, then actually the underlying graph is very well-connected, and so the Markov chain is going to mix rapidly.

Definition 20.13. The *absolute spectral gap* of P is defined as $\gamma^* = 1 - \lambda^*$, where $\lambda^* = \max\{|\lambda_2|, |\lambda_{|\Omega|}|\}$.

In other words, we just showed that $\lambda^* < 1$ if and only if the chain is ergodic.

§20.4 Mixing times and the spectral gap

Here's the theorem, which says you can bound the mixing time in terms of the spectral gap — it gives both an upper bound *and* a lower bound.

Theorem 20.14

For every initial distribution μ_0, we have

$$\mathsf{T}_{\mathsf{mix}}(\varepsilon;\mu_0) \le \frac{1}{\gamma^*}\log\left(\frac{\sqrt{\chi^2(\mu_0 \,\|\, \mu)}}{2\varepsilon}\right).$$

Also, we have

$$\left(\frac{1}{\gamma^*}-1\right)\log\frac{1}{2\varepsilon} \le \mathsf{T}_{\mathsf{mix}}(\varepsilon) \le \frac{1}{\gamma^*}\left(\frac{1}{2}\log\frac{1}{\mu_{\min}}+\log\frac{1}{2\varepsilon}\right),$$

where $\mu_{\min} = \min_{x\in\mathrm{supp}(\mu)}\mu(x)$.

(The upper bound in the second statement comes from maximizing the right-hand side over all μ_0.)

Typically $\mu_{\min}$ is going to be exponentially small, but that's fine because of the log — so if you want to prove a polynomial bound on the mixing time, you just want to show that $1/\gamma^*$ is polynomially bounded in n. And this is also necessary, because of the lower bound on mixing time.

Student Question. *Is the upper bound tight?*

Answer. In general, say my Markov chain is a random walk on a graph, and say that graph is an expander, so that $1/\gamma^*$ is a constant. I still need $\log n$ steps to mix, because that's the diameter of the graph; and μ is uniform over the vertices, so $\mu_{\min}$ is going to be $1/n$. So this is tight (up to constants, at least).

Student Question. *What's the usual magnitude of things — is γ^* usually constant, or...*

Answer. For something like the hypercube, γ^* is going to be $1/n$, and that's sharp. Typically for Markov chains like Glauber dynamics, it's going to be $1/n$ or worse; we'll see an example in this lecture where $1/\gamma^*$ can be exponentially large. You *hope* that it's polynomial in your input size, but that's not always achievable.

§20.5 The Dirichlet form

Before we prove this, we'll give an alternative way of thinking about the spectral gap.

Remark 20.15. We can also define the usual spectral gap as $\gamma = 1-\lambda_2$. It's typically enough to just look at this — for example, if my Markov chain has been lazified (so I add a $1/2$ probability of staying at the current state every step), then that ensures all the eigenvalues are nonnegative, so $\lambda_{|\Omega|}$ is going to play no role. So if P is lazy, we have $\gamma_* = \gamma$. And working with just the spectral gap itself is a little more convenient.

Now we'll give an alternative interpretation of the spectral gap.

Definition 20.16. The Dirichlet form is the quantity

$$\mathcal{E}_P(f,f) = \frac{1}{2}\sum_{x,y\in\Omega}\mu(x)P(x\to y)(f(x)-f(y))^2.$$

What is this expression trying to capture? I have my Markov chain, and it has some edges; say I have an edge between x and y. This quantity is capturing the *local variation* of this function along the edges of the Markok chain — you're only comparing the function values across adjacent states. So this is sort of the 'local variation' of f.

We're going to compare this with the 'global variation' of f, i.e., the variance — for comparison, one way to write $\mathrm{Var}_\mu(f)$ is by an analogous quantity, except that we sum over *all* states, and instead of the transition probabilities we have $\mu(x)\mu(y)$ — so

$$\mathrm{Var}_\mu(f) = \frac{1}{2}\sum_{x,y\in\Omega} \mu(x)\cdot\mu(y)\cdot(f(x)-f(y))^2.$$

This kind of makes it more apparent that we're trying to compare these two quantities.

Here's another fact, which says we can reinterpret the spectral gap using these things.

Fact 20.17 — The spectral gap is given by

$$\gamma = \inf_f \frac{\mathcal{E}_p(f,f)}{\mathrm{Var}_\mu(f)}$$

(where the infimum is taken over all nonconstant f).

So we're looking at the ratio of the local variation and the global variation.

This is essentially true by the variational characterization of eigenvalues. We can interpret the Dirichlet form algebraically as

$$\mathcal{E}_p(f,f) = \langle f, (I-P)f\rangle_\mu.$$

Student Question. *Is this the same thing as a Rayleigh quotient?*

Answer. Yes, you can think of it as a Rayleigh quotient.

Student Question. *What is γ without the star?*

Answer. It's $1-\lambda_2$.

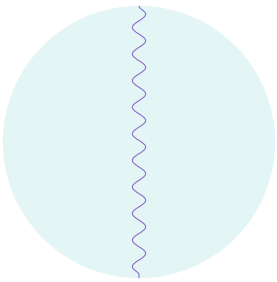
§20.6 Conductance

One reason we introduced this interpretation of eigenvalues is to state a more combinatorial way of thinking about this spectral gap. Unfortunately there's a lot of definitions today, but hopefully this combinatorial definition will make things feel a bit more concrete.

Definition 20.18. For $S\subseteq\Omega$ satisfying $0<\mu(S)<1/2$, we define the conductance of S as

$$\Phi(S) = \mathbb{P}_{x\sim\mu, y\sim P(x\to\bullet)}[y\notin S \mid x\in S] = \frac{\sum_{x\in S, y\notin S}\mu(x)P(x\to y)}{\sum_{x\in S}\mu(x)}.$$

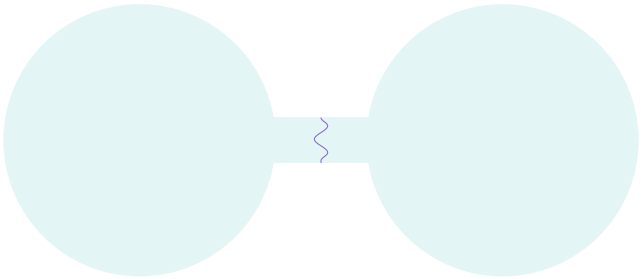
Why are we introducing this? This thing has a very nice interpretation as a picture. Essentially, the way you should think about conductance is it's the probability of 'escaping' a set S.



Imagine this big circle is our state space Ω, and I've cut it into two pieces — S on the left, and $\Omega \setminus S$ on the right. Think of $\mu(S)$ as the 'volume' of S. And along the boundary between S and its complement, you have all the transitions $\sum_{x \in S, y \notin S} \mu(x) \cdot P(x \to y)$; the way to think about this is as a sort of 'surface area.'

And what we want to say is basically that the surface area is very large compared to the volume (if we want something like fast mixing).

The conductance $\Phi(S)$ is also sometimes called the isoperimetric constant for S or the bottleneck ratio for S — the way to think about it is that this quantity is small if and only if S forms a sort of bottleneck in the state space. The above picture is a 'good' situation. A 'bad' situation would be something like two circles with a very thin thing connecting them; then if you took S on the left and $\Omega \setminus S$ on the right, you'd have large volume and tiny surface area.



And the idea is if your Markov chain looked like this, then it would mix very slowly, because it would take you a very long time to go from a state in S to a state not in S. So this picture would be a barrier to rapid mixing, because it'd take a long time to reach $\Omega \setminus S$ from S.

Similarly to how we're taking the worst-case function for γ, we'll also look at the worst-case S here.

Definition 20.19. We define the conductance of the chain as

$$\Phi(P) = \min \Phi(S),$$

where the minimum is over all $S \subseteq \Omega$ with $0 < \mu(S) \leq 1/2$.

For example, $\Phi(P) > 0$ if and only if the Markov chain is connected (i.e., every cut has at least one outgoing edge). What we want to say is that if this thing is lower-bounded by some quantity, then in some sense *every* set S has a large set of edges going out — it's expanding in some sense.

Student Question. *Why do we have the requirement $0 < \mu(S) \leq 1/2$?*

Answer. If I replace S with its complement, the surface area hasn't changed, but maybe the volume is different; and you always want to compare the surface area with the *smaller* of the two halves.

The notion of conductance is sort of an easier way to understand why a Markov chain might or might not mix. If the conductance is small, there exists S for which the state space looks like the second picture, and this cut is a barrier to mixing. Conversely, if all your cuts are very well-expanding, then it turns out your Markov chain *will* mix rapidly.

There's also a theorem that relates conductance to the spectral gap.

Theorem 20.20 (Cheeger's inequality)
Assuming P is reversible, we have

$$\frac{1}{2}\Phi(P)^2 \leq \gamma \leq 2 \cdot \Phi(P).$$

In particular, if your conductance is small, the spectral gap is small and the chain mixes slowly; if it's large, then the spectral gap is large and the chain mixes rapidly. So this means small conductance is really the same thing as small gap, which is really the same thing as large mixing time. And large conductance is equivalent to large gap is equivalent to small mixing time. (For this implication, we're also assuming P is lazy.)

The first inequality is quite a bit more difficult than the other one, so we won't do it in class; if you're interested, there's a very nice book b yLevin–Peres–Wilmer on Markov chain mixing times.

The second is not too difficult, and goes by looking at the variational characterization of the spectral gap — you can take S and construct a function such that when you plug it in, you get literally the conductance.

Proof of upper bound. For $S \subseteq V$ such that $0 < \mu(S) \leq 1/2$, we have

$$\frac{\mathcal{E}_P(\mathbf{1}_S, \mathbf{1}_S)}{\mathrm{Var}_\mu(\mathbf{1}_S)} \leq 2 \cdot \Phi(S)$$

(since the denominator is $\mu(S)\mu(\Omega \setminus S)$, and the numerator is literally equal to the surface area). Now you can take a minimum over all S, and this is always going to be an upper bound on the spectral gap. □

In the remaining time, we'll see a proof of at least the upper bound on the original theorem (on mixing times and spectral gap), and we'll use the theorem to prove a lower bound on the mixing time of the Curie–Weiss model.

§20.7 Slow mixing for the Curie–Weiss model

Recall the Curie–Weiss model is the ferromagnetic Ising model on the complete graph, where

$$\mu(\sigma) \propto \exp\left(\frac{\beta}{2n}\langle \sigma, \mathbf{1}\rangle^2\right)$$

(we call $\langle \sigma, \mathbf{1}\rangle$ the *magnetization*). Last class, we saw that if $\beta < 1$, then Glauber dynamics mixes in $O(n \log n)$ time. Today we'll see the other side of this phase transition.

Theorem 20.21

If $\beta > 1$, then Glauber dynamics has $\mathsf{T}_{\mathsf{mix}} \geq \exp(\Omega(n))$.

So at this phase transition, the mixing time dramatically slows down from near-linear to exponential.

The proof is going to be by constructing a set S which forms a bottleneck in the state space — i.e., constructing a set S for which the conductance is exponentially small.

First, why do we have this threshold on β? Conceptually, the way to think about what's happening here is that if I plot the distribution of the magnetization $\langle \sigma, \mathbf{1}\rangle$, if $\beta < 1$ there's going to be a nice unimodal Gaussian-looking distribution; but if $\beta > 1$, it turns out (and we'll see this in the proof) that the distribution becomes *bimodal*, and these two modes are going to be separated by a distance $\Omega(n)$. So when $\beta < 1$, you have a nice sub-Gaussian thing — the distribution of the magnetization looks like if you added up a bunch of independent ± 1 random variables. But if $\beta > 1$, the interactions become strong enough that you have very strong correlations.

This picture is what'll suggest a good cut to show slow mixing. Let

$$S = \{\sigma \mid \langle \sigma, \mathbf{1}\rangle > 0\}.$$

So we're just cutting the state space right along the y-axis of the above picture. On the right side, we have all configurations with strictly positive magnetization; and on the left, we have all configurations with non-positive magnetization. Basically, what we're going to argue is that all the states in between — that's going to be our surface area — is going to be very small relative to the probability mass of either side.

So we'll show that

$$\Phi(S) \le \exp(-\Omega(n)),$$

which will imply that $\mathsf{T}_{\mathsf{mix}} \gtrsim \exp(\Omega(n))$.

Now let's compute what this thing is. Recall that the definition of conductance is

$$\Phi(S) = \frac{\sum_{\sigma \in S, \sigma' \notin S} \mu(\sigma) \cdot P(\sigma \to \sigma')}{\sum_{\sigma \in S} \mu(\sigma)}.$$

We claim that this is always upper-bounded by the following. Basically, Glauber dynamics at every step only updates a *single* coordinate — so at every step, I can only shift the magnetization by 2. So basically, what we want to claim is that the surface area is just everything in this tiny region around 0. A more formal way to say this is that this is upper-bounded by

$$\frac{\sum_{\sigma : \langle \sigma, \mathbf{1} \rangle = 0} \mu(\sigma)}{\frac{1}{2} - \sum_{\sigma : \langle \sigma, \mathbf{1} \rangle = 0} \mu(\sigma)}$$

(we're assuming for convenience that n is even; this is not essential). So basically, what we're claiming is this conductance is upper-bounded by the total probability mass of all the states literally in the middle, divided by some constant. This is because Glauber updates only one coordinate in each step, so we only get a nonzero contribution if the Hamming distance between σ and σ' is at most 1.

So now we want to show that

$$\sum_{\sigma : \langle \sigma, \mathbf{1} \rangle = 0} \mu(\sigma) \le \exp(-\Omega(n)).$$

To do this, basically what we want to claim is that the relative mass of all configurations with 0 magnetization is way, way, way smaller than all configurations with magnetization at one of the two modes. So basically, I want to compare the two probabilities at the corresponding x-coordinates of the graph.

So it suffices to show there exists $m^* \in [-1, 1]$ such that

$$\sum_{\sigma : \langle \sigma, \mathbf{1} \rangle = 0} \exp\left(\frac{\beta}{n} \langle \sigma, 1 \rangle^2\right) \le \exp(-\Omega(n)) \cdot \sum_{\sigma : \langle \sigma, \mathbf{1} \rangle = m^* n} \exp\left(\frac{\beta}{2n} \langle \sigma, \mathbf{1} \rangle^2\right)$$

(to show our probability is small we just have to show the left-hand side is exponentially small if I sum over the *entire* partition function; and here we're saying something stronger, that it'll be true even if we sum over configurations with a particular fixed magnetization). And we're going to find m^* basically by understanding the 1-dimensional landscape of this picture.

So now let's do a little bit of calculation. In these summations, I've really fixed a magnetization, so everything inside the exponential is independent of the choice of σ; this means the LHS is just $\binom{n}{n/2}$ (the exponential has 0 on the inside), and the RHS is

$$\binom{n}{\frac{1+m^*}{2} \cdot n} \exp\left(\frac{\beta}{2n} \cdot (m^* n)^2\right) = \binom{n}{\frac{1+m^*}{2} \cdot n} \cdot \exp\left(\frac{\beta}{2} \cdot n \cdot (m^*)^2\right).$$

More generally, for all magnetizations m, I can write

$$\sum_{\sigma : \langle \sigma, \mathbf{1} \rangle = mn} \exp\left(\frac{\beta}{2n} \langle \sigma, \mathbf{1} \rangle^2\right) = \binom{n}{\frac{1+m}{2} \cdot n} \cdot \exp\left(\frac{\beta}{2} \cdot n \cdot m^2\right).$$

And if we apply Stirling's formula to the binomial coefficient, we get that this is

$$\mathsf{poly}(n) \cdot \exp\left(n \cdot \left(h(m) + \frac{\beta}{2} \cdot m^2\right)\right),$$

where $h(m) = -\frac{1-m}{2}\ln\frac{1-m}{2} - \frac{1+m}{2}\ln\frac{1+m}{2}$ is some sort of entropy. We can ignore the $\mathsf{poly}(n)$ terms, since the remainder of the thing is exponential.

And then we can plot what this function (on the inside of the exponential) looks like. When $\beta < 1$, it's a nice unimodal function. When $\beta > 1$, it'll have two global maxima, and the center (at 0) will be a local minimum. This already gives us everything.

More explicitly, let $\phi(m)$ be the thing inside the exponential.

Claim 20.22 — IF $\beta > 1$, then there exists $m^* \neq 0$ such that $\phi(m^*) > \phi(0)$.

Then the $\exp(-\Omega(n))$ term in the desired bound is going to be $\exp(n \cdot (\varphi(0) - \varphi(m^*)))$, where $\varphi(0) - \varphi(m^*)$ is some negative constant independent of n.

Here, all you really have to prove is that when $\beta > 1$, around 0 this function φ is strictly convex (so it's going to increase if you move on either side).

Unfortunately we don't have time to discuss the proof of Theorem 20.14. But maybe the main conceptual thing to take away from this is that mixing times are really related to whether or not there exist bottlenecks in the state space; and here we're quantifying whether or not a thing is a bottleneck by the conductance.

And you can use this to show various lower bounds on mixing time, e.g., for Curie–Weiss. You can *also* use this method to prove *fast* mixing for other classes of chains, though we didn't get to discuss that this lecture.

And another takeaway is that you can interpret everything linear-algebraically through the eigenvalues of the transition matrix P.

§21 April 23, 2025

Today we'll continue discussing the relationship between the spectral gap and mixing time for Markov chains. Then we'll discuss how to relate mixing times to properties of the stationary distribution — particularly concentration probabilities.

§21.1 More about the spectral gap

We're working with reversible MCs, where the transition matrix is guaranteed to have real eigenvalues in $[-1, 1]$, with the top eigenvalue being equal to 1. We saw that for mixing, it was important to ensure there's a spectral gap — a gap between the second-largest eigenvalue and 1 (the largest one).

Fact 21.1 (Poincare inequality) **—** The spectral gap satisfies

$$\gamma = 1 - \lambda_2 = \inf_{f \text{ nonconstant}} \frac{\mathcal{E}_P(f, f)}{\mathrm{Var}_\mu(f)}.$$

Theorem 21.2

If P is reversible and lazy and $\gamma = 1 - \lambda_2$, then

$$\left(\frac{1}{\gamma} - 1\right) \log \frac{1}{2\varepsilon} \leq \mathsf{T}_{\mathsf{mix}}(\varepsilon) \leq \frac{1}{\gamma} \log \frac{1}{2\varepsilon\sqrt{\mu_{\min}}}.$$

We also saw a more intuitive combinatorial 'interpretation' (in some sense) of the spectral gap, using Cheeger's inequality. We draw the 'good case' as a big circle, with S being one half of it. Here S has large 'surface area,' so we have large conductance $\Phi(S)$.

The bad case is when our state space looks like a hourglass. Then if you cut in the middle, we'll have a set S with large volume but very small 'surface area,' so it'll have small conductance.

You can see that in the picture on the right, at least, we'd expect the MC to mix slowly — it'll take exponential time to go from one cluster to another. And Cheeger's inequality relates the spectral gap (a purely linear algebraic quantity) with this more combinatorial one (conductance).

Theorem 21.3 (Cheeger's inequality)

We have $\frac{1}{2}\Phi(P)^2 \leq \gamma \leq 2 \cdot \Phi(P)$.

Let's do a couple of examples, which saturate both these bounds.

Example 21.4

Consider the simple random walk on $\{\pm 1\}^n$, where I pick a uniformly random coordinate and flip it.

In this case, it's known that $\gamma = \frac{1}{n}$ (exactly). Actually, if you use the interpretation of the spectral gap as the *Poincare constant*

$$\gamma = \inf_{f \text{ nonconstant}} \frac{\mathcal{E}_p(f, f)}{\mathrm{Var}_\mu(f)},$$

we actually proved this on the midterm when we showed a bound on Lipschitz functions on the cube.

And the upper bound (up to a factor of 2) is attained by the following set: If I have a cube, I can cut the cube along one coordinate direction. So I cut it down the middle by thresholding on one coordinate — you take a box and literally slice it in half along one of the coordinate directions. You can compute the conductance of that cut, and it'll also be equal to $\frac{1}{n}$. So in this case, we have $\gamma \asymp \Phi(P)$.

And in general, when I have Glauber dynamics with Dobrushin's condition, I'll also have spectral gap of the order $1/n$ — that's sort of the best thing you can hope for, for those types of chains.

Example 21.5

Consider the simple random walk on the n-vertex cycle (so at every step, we pick uniformly at random whether to step to the left or right).

This graph is simple enough that you can compute its eigenvalues very explicitly (there are nice algebraic ways to do this), and you can show that $\gamma \asymp \frac{1}{n^2}$. (There's an exact formula involving sin and cos and so on.)

And you can see the conductance is of order $\frac{1}{n}$ — the worst case is to cut down the middle, where I only have two vertices crossing the cut and I have $\frac{n}{2}$ vertices on each side.

So that's where the left-hand inequality in Cheeger is saturated — where you really need the square in the conductance.

§21.2 Intuition for spectral gap vs. mixing

A nice thing about using conductance is that it has a very nice intuitive interpretation for why it's relevant in mixing. For some intuition, let's also look at why eigenvalues might be related.

If I have an ergodic Markov chain P, then what we're expecting is that if I look at powers P^t, this thing as a *matrix* is going to be converging to a single deterministic matrix, namely the one which is given by putting the distribution μ on every single row of this matrix, i.e.,

$$P^t \to \mathbf{1}\mu^\intercal.$$

This is essentially by the fundamental theorem of Markov chains.

The question of mixing is:

> **Question 21.6.** How fast is this convergence (in terms of matrices)?

On the right-hand side, one thing to recognize is that $\mathbf{1}$ and μ are essentially the eigenvectors of P corresponding to the top eigenvalue 1 — specifically, $\mathbf{1}$ is the right eigenvector and μ is the left eigenvector.

So one way to think linear algebraically is that the rate of convergence should correspond to how fast the *other* eigenvalues of P^t decay to 0. And we know the eigenvalues of P^t are just given by taking the eigenvalues of P and raising them to the tth power.

So at least one intuitively expects that the rate of convergence should be governed by how close these eigenvalues are to the top eigenvalue 1 — for mixing, we need λ^t to be small for all nontrivial eigenvalues λ.

§21.3 Proof of Theorem 21.2

So that's a bit of informal intuition as to why the theorem is true; now we'll go through a proof of the theorem, at least the upper bound.

Last time, we briefly mentioned how eigenvalues are related to the χ^2-divergence:

> **Definition 21.7.** The χ^2-divergence between two distributions μ and ν is defined as
>
> $$\chi^2(\nu \parallel \mu) = \mathbb{E}_{x\sim\mu}\left[\left(\frac{\nu(x)}{\mu(x)} - 1\right)^2\right].$$

You can think of this as some weighted kind of distance, but it's not symmetric.

We also saw that the χ^2-divergence gives an upper bound on the TV distance — if I can show the χ^2-divergence to stationary is small, that also means I'm close to stationary in TV.

The key inequality is we're going to show that the χ^2 divergence decreases multiplicatively at *every* step of the chain:

> **Claim 21.8 —** For all ν, we have
>
> $$\chi^2(\nu P \parallel \mu) \le (1-\gamma)^2 \cdot \chi^2(\nu \parallel \mu).$$

In other words, every single step of the chain is making progress towards stationary, at least as measured by χ^2 divergence.

Student Question. *What's the cutoff — since it already seems we're decaying exponentially?*

Answer. The thing is that with TV distance I'm always starting at 1. But the χ^2 divergence can initially start off way bigger.

Remark 21.9. Here we're assuming the MC is lazy; that's not necessary, but otherwise you have to replace the spectral gap with the absolute spectral gap $(1 - \max\{|\lambda_2|, |\lambda_n|\}$.)

If we can show this inequality, then we'll be done — we can just iterate this inequality, figure out what the maximum possible starting value $\chi^2(\nu \parallel \mu)$ is, and set t accordingly. So we'll now just prove this key inequality.

Proof of Claim 21.8. We want to interpret the quantities appearing here in terms of linear algebra. Recall that we defined an inner product

$$\langle f, g\rangle_\mu = \mathbb{E}_{x\sim\mu}[f(x)g(x)]$$

(if you like, you can think of μ as being uniform, in which case this is the (normalized) standard Euclidean inner product). We'll also write

$$\frac{d\nu}{d\mu}(x) = \frac{\nu(x)}{\mu(x)}$$

(this is some function on my state space).

We claim that you can write the χ^2-divergence as

$$\chi^2(\nu \parallel \mu) = \left\langle \frac{d\nu}{d\mu}, \frac{d\nu}{d\mu}\right\rangle_\mu - \left\langle \frac{d\nu}{d\mu}, \mathbf{1}\right\rangle_\mu^2$$

(you can check this just by expanding things out and using the fact that $\mathbb{E}[\frac{d\nu}{d\mu}] = 1$).

Now we want to compute $\chi^2(\nu P \parallel \mu)$ and relate it to the original χ^2-divergence. This is

$$\left\langle P\frac{d\nu}{d\mu}, P\frac{d\nu}{d\mu}\right\rangle_\mu - \left\langle P\frac{d\nu}{d\mu}, \mathbf{1}\right\rangle^2.$$

(Here we're using the fact that $\frac{d(\nu P)}{d\mu} = P\cdot\frac{d\nu}{d\mu}$, which is a short calculation using something like reversibility.)

What this implies is now if I take the difference of these two things, I can write this as

$$\chi^2(\nu \parallel \mu) - \chi^2(\nu P \parallel \mu) = \mathcal{E}_{P^2}\left(\frac{d\nu}{d\mu}, \frac{d\nu}{d\mu}\right).$$

And once we've written it in this way, now we're very happy — I have a Dirichlet form, so I can use the assumption that I have a spectral gap. So I can compare the right-hand side with the χ^2-divergence again — this is going to be lower-bounded by

$$(1 - \chi_2^2)\cdot\chi^2(\nu \parallel \mu)$$

(by the Poincare inequality). And now if we rearrange everything, we get exactly the inequality we want — this shows

$$\chi^2(\nu P \parallel \mu) \le \lambda_2^2\cdot\chi^2(\nu \parallel \mu).$$ □

Student Question. *$\lambda_2 < 1$ is guaranteed by irreducibility?*

Answer. Yes. Here we're also using laziness because for the second-largest eigenvalue of P^2, you have to consider both λ_2 and the smallest eigenvalue of P; but here we're assuming the chain is lazy, so you only have to look at λ_2.

We briefly mentioned last time that for Cheeger, the upper bound is easy (you take f which is the indicator of your set). The lower bound is more difficult; there's a nice proof in this book on Markov chain mixing times.

That's all we'll say about this mixing theorem; the proof is really more about writing out the definitions, and maybe there's one slight calculation to do. But it really says that the eigenvalues *really* capture the decay rate for χ^2-divergence.

§21.4 Concentration via mixing/isoperimetry

Now we'll illustrate some connections between MC mixing and properties of the stationary distribution. One of the main motivations for studying MCs is an algorithmic perspective — if you want to sample from some distribution, you can design a chain with that as its stationary distribution, and run the chain.

But Kuikui will show us that even if you don't care about sampling, you should still care about MCs, because you can use them to deduce useful properties of the stationary distribution.

We've already kind of seen some hints of this connection already. For instance, when we looked at the Curie–Weiss model:

Example 21.10

Consdier the Curie–Weiss model

$$\mu(\sigma) \propto \exp\left(\frac{\beta}{2n} \cdot \langle \sigma, \mathbf{1}\rangle^2\right).$$

In the high-temperature regime $\beta < 1$, we saw two phenomena:

- We have $O(n \log n)$ mixing of Glauber dynamics.
- The magnetization $\langle \sigma, \mathbf{1}\rangle$ is sub-Gaussian.

We sketched why this was the case by looking at the distribution of the magnetization; it'll look like a unimodal function with a single maximum in the middle. So somehow in this high-temperature regime, you're getting both concentration and mixing of Glauber dynamics; these are two ways of saying that the distribution is behaving as if the spins were independent and uniformly sampled.

We also saw (by looking at the definition of the Poincare constant) that if you have a spectral gap, sometimes you're already getting a bound on the variance of a function for free — it's equivalent to saying that

$$\mathrm{Var}_\mu(f) \le \frac{1}{\gamma} \cdot \mathcal{E}_p(f, f).$$

If I can bound the variance, that's saying something about the concentration properties of f (e.g., I can use Chebyshev). And typically, the Dirichlet form on the RHS is very easy to bound. For example, in Glauber dynamics, at every step I'm picking a uniformly random coordinate and resampling it. So if my function is Lipschitz with respect to Hamming distance, that immediately gives me a bound.

Corollary 21.11

If f is Lipschitz in the sense that $|f(x) - f(y)| \le L$ for all $x, y \in \Omega$ such that $P(x \to y) > 0$, then $\mathrm{Var}_\mu(f) \le L/\delta$.

Example 21.12

For Glauber dynamics, this notion of Lipschitzness becomes Lipschitzness with respect to the Hamming distance. We've already seen this type of concentration inequality, with McDiarmid.

Remark 21.13. For the Hamming distance and Glauber dynamics with respect to $\mathsf{Unif}\{\pm 1\}^n$, this recovers the Efron–Stein inequality — something we proved on our midterm.

So with the Poincare inequality, we can often easily get bounds on the variance of a function.

So that's one connection. One last connection (kind of informal) is that we can kind of interpret a lower bound on conductance as being a concentration property, in the following sense. Let

$$A_s = \{x \in \Omega \mid f(x) \le s\}$$

(for some threshold s) — so we're looking at the level sets of our function.

If you apply whatever lower bound on conductance you have for these sets A_s, then you'll get some kind of a concentration statement. Concentration around $\mathbb{E}_\mu[f]$ is really equivalent to saying that $\mu(A_{\mathbb{E}_\mu(f)+t})$ is close to 1 (as a function of t).

And if we have a conductance lower bound on these sets A_s, you can informally think of it like the following (we'll make this more formal in a moment). Informally, we have

$$\frac{\mu(A_{s+\varepsilon} \setminus A_s)}{\mu(A_s)} \approx \mathbb{P}_{x\sim\mu, y\sim P(x\to\bullet)}[y \notin A_s \mid x \in A_s]$$

Think of the MC as making local perturbations — so y shouldn't be too far from x. And we're looking at, if I start in a state where f is small, what's the probability I leave that set and go to a state with large f? If you don't look too closely at this, these are sort of approximately the same. And if I have a lower bound on the right-hand side, I have a lower bound on the left one, which really is a concentration statement — telling me these probabilities get close to 1 as I increase t.

So what we're saying is that assuming this approximation is true, a lower bound on this implies concentration.

§21.5 Gausian concentration inequality

This is sort of a very informal way of thinking about the connection between concentration and mixing. Now we'll say something a bit more formal. We'll go back to a very basic setting, namely the setting of standard Gaussians. But here we're not going to use independence.

We're going to prove the following:

Theorem 21.14 (Gaussian concentration inequality)

Let $\gamma_n = \mathcal{N}(0, I)$ be a n-dimensional standard Gaussian, and let $f : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz with respect to Euclidean distance — i.e.,

$$|f(x) - f(y)| \le L \|x - y\|_2 \quad \text{for all } x, y \in \mathbb{R}^n.$$

Then

$$\mathbb{P}_{X\sim\gamma_n}[f(X) - \mathbb{E}_{\gamma_n} f \ge t] \le \exp\left(-\frac{t^2}{2L^2}\right).$$

This is a different kind of Lipschitzness assumption — for McDiarmid we assumed Lipschitzness with respect to Hamming distance (where you change a small number of coordinates, but you're allowed to do so arbitrarily). Here we're asking for something smoother and in many ways weaker. And we're claiming you have a very strong concentration inequality.

We're going to prove this using isoperimetric type inequalities.

This thing has many different proofs, but we'll use one which illustrates this theme.

To prove this, we'll define a couple of things.

Definition 21.15. For $A, B \subseteq \mathbb{R}^n$, their Minkowski sum is defined as

$$A + B = \{a + b \mid a \in A,\, b \in B\}.$$

Geometrically, it's a rather nice thing. Imagine I take a ball and add it to a triangle. Then I just look at every single point in the triangle and center a ball around it, and I take the union of all those. That'll give me a rounded-off triangle (where the corners are rounded, and you have straight edges on the sides).

The reason we're defining this is we're going to use this to define a notion of surface area.

Definition 21.16. We define $A_t = A + t \cdot \mathbb{B}_2^n$, where $\mathbb{B}_2^n$ is the Euclidean ball of radius 1.

In other words, we have

$$A_t = \{x \in \mathbb{R}^n \mid \|x - a\|_2 \leq \text{ for some } a \in A\}.$$

We're going to use this notion to define surface area:

Definition 21.17. The surface area of A is defined as

$$\text{Area}(A) = \lim_{\varepsilon \to 0} \frac{\text{Vol}(A_\varepsilon) - \text{Vol}(A)}{\varepsilon}.$$

So imagine that instead of adding a big ball, I add a bunch of tiny ones; that thickens up the set along the boundary very slightly.

So imagine I have A (drawn as a square) and I cover it with a bunch of small balls, each of radius ε; and I let their radii go to 0. So the numerator is essentially capturing the stuff on the border.

For instance, the classic isoperimetric inequality in Euclidean space is if I fix some amount of volume, then the sets which have the smallest surface area with that givene volum look like a Euclidean ball. In some sense, Euclidean balls are the most efficient in terms of containing some volume for some amount of surface area.

Here we'll be working in Gaussian space, so we need an analog of that theorem.

Theorem 21.18 (Borell's Gaussian isoperimetric theorem)

Fix $A \subseteq \mathbb{R}^n$, and let $H \subseteq \mathbb{R}^n$ be a half-space. Suppose that $\gamma_n(A) = \gamma_n(H)$.

Then for every t, we have

$$\gamma_n(A_t) \geq \gamma_n(H_t).$$

A half-space means I pick a vector and look at all the points on one side of thee (shifted) hyperplane formed by this vector — equivalently, a set of the form $\{x \in \mathbb{R}^n \mid \langle x, v\rangle \leq s\}$ for some $v \in \mathbb{R}^n$ and $s \in \mathbb{R}$.

So this in some sense characterizes the sets with the minimum surface area for a given (Gaussian) volume. THis is equivalent to the formulation

$$\lim_{\varepsilon \to 0} \frac{\gamma_n(A_\varepsilon) - \gamma_n(A)}{\varepsilon} \geq \lim_{\varepsilon \to 0} \frac{\gamma_n(H_\varepsilon) - \gamma_n(H)}{\varepsilon}$$

(we can think of this as the 'Gaussian surface area').

So kind of the pont here is that we characterized the minimizers, for a given amount of volume, of surface area. And we can use this to lower-bound things like the conductance of some MC.

We won't have time to prove this statement, but we'll use this to prove the Gaussian concentration inequality — to illustrate the theme of how isoperimetric or conductance or mixing inequalities can be used to prove concentration.

Proof of Gaussian concentration inequality. For convenience, rather than working with the mean, we'll work with the *median* of the distribution. So we fix $f : \mathbb{R}^n \to \mathbb{R}$ which is Lipschitz, and we'lel let m denote a median of f — in other words, a number such that

$$\mathbb{P}_{x\sim\gamma_n}[f(x) \geq m] \geq \frac{1}{2} \quad \text{and} \quad \mathbb{P}_{x\sim\gamma_n}[f(x) \leq m] \geq \frac{1}{2}.$$

We showed in our first homework that any median is always going to be within 1 standard deviation of the mean; so proving things for the median is essentially without loss of generality.

And our goal is to show concentration around m.

Why was the median sort of convenient for us? Let's again look at these level sets — let

$$A = \{x \in \mathbb{R}^n \mid f(x) \leq m\}.$$

We assumed that $\gamma_n(A) \geq \frac{1}{2}$ and also that $\gamma_n(\mathbb{R}^n \setminus A) \geq \frac{1}{2}$. And we want to use the Gaussian isoperimetric theorem. So we want to choose some half-space with the same Gaussian volume. But since A and its complement have volume at least $\frac{1}{2}$, this half-space had better be centered.

So let $H = \{x \in \mathbb{R}^n \mid \langle x, v\rangle \leq s\}$ be a half-space with Gaussian volume $\gamma_n(A) = \gamma_n(H)$; the fact that $\gamma_n(A)$ and $\gamma_n(\mathbb{R}^n \setminus A)$ are at least $\frac{1}{2}$ means we must have $s = 0$ (we don't care what v is; it can be whatever you want, e.g., e_1).

So now we're slicing space into two halves and looking at one side of it, and that's going to be H.

Now I want to use the comparison between how much H expands and how much A expands. So by isoperimetry, we get that for any t, we have

$$\gamma_n(A_t) \geq \gamma_n(H_t).$$

And what is $\gamma(A_t)$? We claim that we can relate it to the tail probability we want.

> **Claim 21.19 —** We have $\gamma_n(A_t) \leq \mathbb{P}_{x\sim\gamma_n}[f(x) \leq m + t \cdot L]$.

In particular, if I chain these together, this gives me a lower bound on the probability that $f(x) \leq m + tL$.

Proof. The proof of this claim is just Lipschitzness — because f is L-Lipschitz, we know that if $a \in A$ and x is such that $\|x - a\|_2 \leq t$, then $|f(a) - f(x)| \leq t \cdot L$. This just says that if I have a point x in A_t, it's within t from a point that has f-value at most m; and the difference between the f-values at a and x is going to be at most tL by Lipschitzness.

So all I'm saying is that if I sample a random point x and it lands in A_t, then I know this event $f(x) \leq m+tL$ also holds. □

Now let's combine all these things together — this implies that

$$\mathbb{P}_{x\sim\gamma_n}[f(x) \geq m + tL] \leq 1 - \gamma_n(A_t) \leq 1 - \gamma_n(H_t).$$

Now let's also draw what H_t looks like. Remember that with H, I'm just looking at a hyperplane (say the one whose normal vector is the first standard basis vector e_1). Then in H_t, I just thicken it a bit — basically I just shift the threshold for this hyperplane to the right, to t.

And basically what I want to say is that $\gamma_n(H_t)$ is very easy to understand — it essentially looks like the tail of a one-dimensional Gaussian. And now I can use standard tail bounds for the standard Gaussian. So we get the bounds

$$\mathbb{P}_{g\sim\mathcal{N}(0,1)}[g \geq t] \leq \exp\left(-\frac{t^2}{2}\right). \qquad \square$$

The point was we were able to use isoperimetry to reduce from bounding level sets of a function to bounding *half-spaces*, which is much easier (it reducs to a 1-dimensional problem). So really, the key step of the proof was the isoperimetric inequality.

Student Question. *We had the median of f on the left, and the expectation in the original statement?*

Answer. We know the median and mean differ by at most a single SD, do proving concentration for the median is without loss of generality.

§21.6 Dobrushin's condition to concentration

In the last ten minutes, we'll just state something more general. This isoperimetric theorem is very beautiful — it characterizes exactly what the worst-case sets you should look at are.

But now we'll state a theorem that doesn't have any of this, and is more directly related to mixing or showing the connection between mixing and concentration, although this one we won't have time to prove (the proof is in the notes; you can think of it as bonus material).

This basically shows that if I work in the discrete setting and assume something like Dobrushin's condition (which implies fast mixing for Glauber dynamics), that implies fast mixing for Glauber dynamics.

Theorem 21.20 (Chatterjee 2005)

Let μ be a probability measure over $\{\pm 1\}^n$, and suppose there exists $0 < \delta \leq 1$ such that Dobrushin's condition is satisfied, i.e.,

$$\max_{i\in[n]} \sum_{j\neq i} \max_{\tau:[n]\setminus\{i,j\}\to\{\pm1\}} \left\| \mu_j^{\tau,i\leftarrow -1} - \mu_j^{\tau,i\leftarrow +1} \right\|_{\mathsf{TV}} \leq 1-\delta.$$

Then for every $f : \{\pm 1\}^n \to \mathbb{R}$ which is L-Lipschitz with respect to the Hamming distance, we have

$$\mathbb{P}_{x\sim\mu}[f(x) - \mathbb{E}_\mu f \geq t] \leq \exp\left(-\frac{\delta t^2}{2nL^2}\right).$$

This was a nice condition that implies fast mixing of Glauber dynamics, using a coupling-based proof. And under this kind of cnodition, we get concentration.

Think of δ and L as constants independent of n.

Note that if we take $\delta = 1$, which corresponds to the case where you have $\mathsf{Unif}\{\pm 1\}^n$ where all coordinates are independent of each other, then this recovers McDiarmid's inequality as a special case.

And now this kind of concentration inequality applies to a much broader class of distributions, one which may have very nontrivial correlations between the coordinates.

Student Question. *Does this also work for the multicolored version?*

Answer. Yes, you can also replace $\{\pm 1\}^n$ with $[q]^n$ if you like.

Student Question. *And is it still the Hamming distance?*

Answer. Yes. And in Dobrushin, we maximize over all possible partial assignments τ and all possible pairs of colors for the ith coordinate.

The proof of this is quite nice — it really uses the coupling that certifies fast mixing of Glauber dynamics.

The connection between mixing and concentration goes much deeper — there are many other related inequalities like this used to imply fast mixing that also imply concentration inequalities of this form — but we don't have time to discuss them here (there are references in the notes).

§22 April 28, 2025 — Probability and the geometry of polynomials

The 5th pset is due tonight. The final pset will be out tonight as well; it will be on the shorter side, so we have more time to work on our final projects.

This week we'll switch gears and discuss a beautiful and unexpected connection between probability theory and the geometry of polynomials.

§22.1 Real-rootedness

To start this discussion, let's start with a one-dimensional setting. Suppose that X is a random variable taking values in the discrete set $\{0, 1, 2, \dots, n\}$ (in a moment, we'll go to a more general setting). If I have a discrete random variable taking nonnegative integer values, then I can encode its distribution into a polynomial:

Definition 22.1. We define $p_X(t) = \sum_{k=0}^{n} \mathbb{P}[X = k] \cdot t^k = \mathbb{E}[t^X]$.

So this is really just a different way of writing the standard moment generating function of X; but we're writing it as a polynomial, because we're assuming X takes values in $\{0, 1, 2, \dots, n\}$.

Question 22.2. Can we use the properties of this polynomial to deduce interesting consequences for our original random variable X?

Because we're talking about polynomials, it makes sense to ask about e.g., the location of its zeros, and how that's related to probabilistic properties.

So the theme is to study analytic or algebraic properties of this generating polynomial p_X, and use this to get information about the original random variable X.

Later in the lecture, we'll see a natural application of this method. But to get warmed up, let's start by proving the following lemma, which already illustrates one such connection.

Lemma 22.3

The following two statements are equivalent:

(1) We can write $X = \sum_{i=1}^{n} X_i$ for independent $X_i \sim \mathsf{Ber}(p_i)$.

(2) The polynomial p_X is real-rooted.

The first statement is a purely probabilistic claim — that we can decompose X as a sum of independent Bernoullis. (Think of this as a binomial distribution, though the technical name for this is a *Poisson binomial* distribution, since the p_i's aren't necessarily equal.)

Meanwhile, the second *a priori* has nothing to do with probability, at least on the surface. But we claim that these two statements are completely equivalent!

To demystify this, let's see the proof.

Proof (1) $\implies$ (2). We're assuming X can be decomposed as a sum of random variables. So if we use the definition of p_X, we have

$$p_X(t) = \mathbb{E}[t^X] = \mathbb{E}[t^{X_1+\cdots+X_n}] = \prod_{i=1}^n \mathbb{E}[t^{X_i}]$$

(using independence to push the expectation inside).

And now X_i is a random variable taking values in $\{0,1\}$, so this is really just

$$p_X(t) = \prod_{i=1}^n ((1-p_i) + p_i \cdot t).$$

That's a factorization of this polynomial that certifies it's real-rooted. □

The other direction is kind of the same, in some sense.

Proof (2) $\implies$ (1). Suppose p_X is real-rooted, so we can factorize

$$p_X(t) = a_0 \prod_{i=1}^n (t + r_i).$$

Since we know p_X has nonnegative coefficients (probabilities are always nonnegative), it's positive on all of $\mathbb{R}_{\geq 0}$; this means all its roots are negative, so $r_i \geq 0$.

We also know $p_X(1) = 1$ (we're just summing all the probabilities $\mathbb{P}[X = k]$). This implies a_0 must take a particular form — we'll have

$$a_0 = \prod_{i=1}^n \frac{1}{1+r_i}.$$

In particular, now we can rewrite this polynomial again as

$$p_X(t) = \prod_{i=1}^n \frac{t+r_i}{1+r_i}.$$

And now we claim that from this form, you can exactly just extract out the correct p_i's to decompose your random variable X — we can write each factor as

$$\frac{r_i}{1+r_i} + t \cdot \frac{1}{1+r_i},$$

so now if we let $p_i = \frac{1}{1+r_i}$, then X is equal (in law) to a sum of independent Bernoullis $X_i \sim \mathsf{Ber}(p_i)$. □

From this, you can already get some pretty interesting consequences. If X was just a sum of independent Bernoullis, you'd expect it to have nice properties. And you can show that real-rootedness implies (this is a result due to Newton, called Newton's inequalities) that the sequence of probabilities $\{\mathbb{P}[X = k]\}_{k=0}^n$ is *unimodal*, and even log-concave — so it'll increase to some point, and start decreasing after.

You can also go the other way — you can deduce properties of real-rooted polynomials from probability. If we have a real-rooted polynomial with nonnegative coefficients (summing to 1), then we know we can write it as the generating polynomial a sum of independent Bernoullis. And such a distribution must satisfy nice concentration probabilities, which lets us get bounds on the coefficients — the coefficients corresponding to very small or very large powers of t must be exponentially small. In other words, this means for every real-rooted polynomial p with nonnegative coefficients, the low-order and high-order coefficients are small (relative to the sum of all the coefficients), by using concentration inequalities for sums of independent random variables.

So you can kind of use this lemma as a dictionary between polynomials and probability measures.

Remark 22.4. There are some very nice extensions of this, or variants of the hypothesis where you change the zero-free region. For example, it's known that if p_X doesn't have zeros in a neighborhood of 1, then X satisfies a central limit theorem (assuming its variance isn't too small).

This kind of statement can be useful in applications where you no longer have a lot of independence; and we'll actually see one such application in a moment. It's perhaps surprising that you can still deduce something like a CLT even without independence, only assuming a zero-free region for the generating polynomial.

This is due to Michelen–Sahasrabudhe 2019.

§22.2 An application — the monomer-dimer model

Now we'll turn to a concrete example coming from statistical physics and the study of Gibbs distributions.

We'll study what's essentially the hardcore model, but on a specific class of graphs (and instead of thinking about independent sets, we'll think about matchings).

Example 22.5

Given a graph $G = (V, E)$ and parameter $\lambda \geq 0$, we consider the Gibbs distribution defined by

$$\mu(M) \propto \lambda^{|M|} \quad \text{for all matchings } M \subseteq E.$$

A matching means our edges aren't allowed to share any vertices. We're allowing matchings of any size — for example, the empty set is also a valid matching (and *perfect matchings* are ones with exactly $\frac{n}{2}$ edges).

So we have a distribution; and we can now look at its partition function

$$Z_G(\lambda) = \sum_M \lambda^{|M|}$$

(where the sum is over all matchings $M \subseteq E$). This is the normalizing constant for the Gibbs distribution; and it's also a polynomial (since $|M|$ only takes nonnegative integer values).

We're going to prove the following theorem. It's actually rather surprising, because it has consequences for the structure of this Gibbs distribution — it'll imply no phase transition occurs on any class of graphs whatsoever. This is quite surprising, because for many models we've seen, they do exhibit phase transitions (even on simple graphs like the complete graph).

In fact, we'll even extend this further to allow weights on the edges. (When we prove it, we'll actually need this generalization to weights.)

Theorem 22.6 (Heilmann–Lieb 1972)

For all graphs G and all weight functions $w : E \to \mathbb{R}_{>0}$, the function

$$Z_G(\lambda) = \sum_M \lambda^{|M|} \cdot \prod_{uv \in M} w_{uv}$$

is real-rooted.

The Gibbs model we defined corresponds to the case where all the weights are 1.

In particular, the earlier lemma says that if I draw a random matching $M \sim \mu$ from the Gibbs distribution and look at its number of edges $|M|$, then that random variable can be decomposed as a sum of independent Bernoullis — in particular, it'll satisfy a CLT and have very nice concentration properties and be unimodal, and so on.

Remark 22.7. As a brief word on the name of this model, *dimers* correspond to edges in M, and *monomers* correspond to unmatched vertices. (This is terminology from statistical physics.)

When you see this, you should be very surprised — usually partition functions will have an insane set of zeros, but for this model we can actually show that all its zeros lie on the negative real line.

§22.3 Lack of phase transitions

Before we prove this theorem, let's first say a few words about why this implies that the monomer-dimer model does not exhibit a phase transition.

In this course, we've seen lots of examples where we do have phase transitions — for example, in Curie–Weiss there's some critical temperature below which our distribution is unimodal, but above it there's a phase transition and it becomes bimodal. We claim that this theorem implies no matter what classes of graphs you look at, there *won't* be a phase transition.

Why? When we talk about phase transitions, we usually mean that some property of the system becomes very sensitive to perturbations in the temperature. For example, for water, at 99 degrees Celsius it's still liquid, and at 101 it becomes a gas. So the behavior of this system is sensitive around this critical temperature.

How do you detect this using the language of polynomials? One of the key insights is that many observables of your system can be obtained analytically from this polynomial.

Definition 22.8. We define the free energy of the distribution as $\mathcal{F}_G(\lambda) = \log Z_G(\lambda)$.

You can think of $Z_G(\lambda)$ as a reparametrization of the moment generating function; then the free energy is just the cumulant generating function.

The claim is that many properties of the system can be deduced analytically from this.

Example 22.9

If we sample $M \sim \mu$, then $\mathbb{E}[|M|]$ can be written as

$$\mathbb{E}[|M|] = \frac{d}{d\lambda} \mathcal{F}_G(\lambda) \Big|_{\lambda=1}.$$

(We proved this for the hardcore model on our homework.)

So observables can be obtained analytically from the free energy. (If you want to look at fancier observables, you should look at a multivariate extension of this polynomial; but to keep things simple, we'll work with the univariate setting.)

Once you know this, the question becomes:

Question 22.10. Where does $\mathcal{F}_G$ behave poorly (e.g., where does it blow up, where is it not smooth)?

Of course, Z_G is just a polynomial, so it's very smooth; the source of any bad behavior must come from the log part. In particular, the free energy is only going to be ill-behaved at the *zeros* of this polynomial (i.e., $\mathcal{F}_G(z)$ is only going to be ill-behaved in a neighborhood of the zeros of Z_G). In particular, where the zeros accumulate — that's where you should look for phase transitions.

So what this says is that phase transitions occur where the zeros of $Z_G(\lambda)$ accumulate as $n \to \infty$. Here we're thinking about a sequence of graphs G_n where the number of vertices is going to ∞ — e.g., a large box in the $\mathbb{Z}^2$ lattice.

And basically what Theorem 22.6 tells us is that all the zeros lie on the negative real axis. This means you'll never see them, because all our inputs to the monomer-dimer model are *nonnegative* λ — so they can't produce phase transitions for $\lambda \geq 0$.

Remark 22.11. This is one way you can detect the presence of a phase transition. Actually, to make this even more convincing, you can also show (though we won't do so in this lecture) that for *every* $\lambda \geq 0$, you always have exponential decay of correlations (in the sense of strong spatial mixing). And you can also show that local Markov chains for sampling (e.g., Glauber dynamics) always mix in polynomial time, no matter what $\lambda \geq 0$ you pick. So in multiple senses, there are no phase transitions for this model — not only can you say there are no phase transitions in the sense of zeros, but also in these other senses we've looked at in the course.

Remark 22.12. There are many other very beautiful theorems in this area. One that we'll also mention now, and which gave birth to this connection between zeros of polynomials and statistical physics, is a very beautiful theorem due to Lee–Yang from 1952, called the 'circle theorem' for the ferromagnetic Ising model. (We won't state it.)

§22.4 Proof of the Heilmann–Lieb theorem

In the rest of the lecture, we'll see a complete proof of Theorem 22.6.

§22.4.1 Some simplifications

First, we're going to rewrite the partition function slightly, in a way that'll be more convenient for the proof technique; and we'll introduce some signs into the coefficients.

Definition 22.13. We define the matching polynomial as

$$\mathcal{M}_G(z) = \sum_M z^{n-2|M|}(-1)^{|M|} \prod_{uv \in M} w_{uv}$$

(where the sum is over all matchings).

(This is the version that combinatorialists use.) Here $n - 2|M|$ is really just counting the number of monomers — the number of vertices that don't have an incident edge in the matching. The other thing we've done is just introduced a $(-1)^{|M|}$ term.

By some direct calculations, it turns out that it's sufficient to show zero-freeness for this polynomial — there's a fairly straightforward transformation between Z_G and $\mathcal{M}_G$ (it's in the notes).

> **Fact 22.14 —** The polynomial $\mathcal{M}_G$ is real-rooted if and only if Z_G is.

So it suffices to establish real-rootedness for this new polynomial.

One more assumption we're going to make is the following: It suffices to prove the theorem when every edge is present in the graph. This is because we're assuming we have some weight function on the edges (we're allowing them to be arbitrarily small, we just want them all to be positive). So it suffices to prove Theorem 22.6 where G is the complete graph, but $w : \binom{V}{2} \to \mathbb{R}_{>0}$ are positive but otherwise arbitrary.

If you can do this for the complete graph, then you can also prove it for any graph — we can take any pair of vertices that don't form an edge in our graph, create a weight for it, and then send that weight to 0 smoothly. It's well-known that if we have a sequence of real-rooted polynomials which converge coefficient-wise to some limiting polynomial, then that limiting polynomial must also be real-rooted:

> **Fact 22.15 —** If $f_n \to f$ coefficient-wise and the f_n are all real-rooted, then f is also real-rooted.

So we've done two simplifications (rewriting the polynomial — though it's not clear yet why this is a simplification — and assuming that all edges are present, though possibly with arbitrarily large or small weights).

§22.4.2 Decomposing the matching polynomial

The rough strategy is to go by induction. To set up this strategy, we need two claims. First, we need to be able to decompose this polynomial.

> **Claim 22.16 —** We can decompose $\mathcal{M}_G(z) = z\mathcal{M}_{G-v}(z) - \sum_{u\sim v} w_{uv} \cdot \mathcal{M}_{G-u-v}(z)$.

(We use $G - v$ to denote the graph with v removed, and $G - u - v$ to denote G with both u and v removed.)

Basically, the idea is that once I've decomposed my matching polynomial as a sum of smaller ones, I roughly want to say that if each of these guys is real-roted, then so is $\mathcal{M}_G$.

Proof. The idea is that we can take the space of all possible matchings in G, and partition them into ones that don't have v matched (i.e., ones where v is a monomer) and ones that include the edge uv for some $u \sim v$. The first term counts all matchings in the former set, and the second counts all matchings in the latter set.

So $z \cdot \mathcal{M}_{G-v}(z)$ counts all matchings of G in which v is a monomer, and $-w_{uv}\mathcal{M}_{G-u-v}(z)$ counts all matchings of G which use the edge uv (the minus sign corresponds to the fact that we've introduced a $(-1)^{|M|}$ term). □

§22.4.3 Interlacing

We said earlier that to do induction, what we want to say is that if we know each of these smaller polynomials is real-rooted, then the polynomial for G is also real-rooted. That's not generically true — in general, adding two real-rooted polynomials won't produce a real-rooted polynomial. But it turns out it *is* true if we know something about how these polynomials relate to each other.

For that we'll need a notion about pairs of real-rooted polynomials, called *interlacing*.

Definition 22.17. Let $p(z) = \prod_{i=1}^{n}(z - r_i)$ and $q(z) = \prod_{i=1}^{m}(z - s_i)$ be real-rooted polynomials, where $|n - m| \leq 1$. We say q interlaces p if

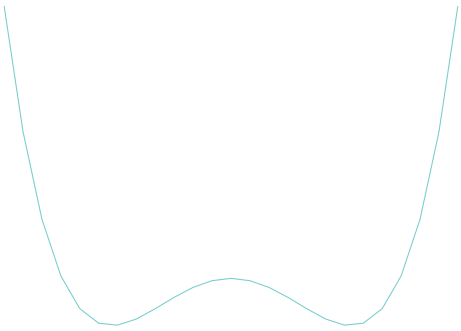
$$s_n \leq r_n \leq s_{n-1} \leq r_{n-1} \leq \cdots \leq s_1 \leq r_1.$$

This basically says that the roots alternate between roots of q and roots of p. Kind of the picture is that maybe we have some roots for our polynomial p (we draw p as a w-shaped quartic with 4 roots); and maybe my other polynomial q has roots going in between the roots of p (we draw q as a slightly shifted quartic).

We're also going to need to allow them to have degrees differing by 1, in which case q might have an additional root on the left. But the key thing is we want the roots to alternate between those of q and those of p.

Why is this relevant? In general, if we have two real-rooted polynomials and we add them, we're not necessarily going to get a real-rooted polynomial back. But we claim that if they interlace, then their sum *will* be real-rooted.

To see this, let's try to draw a better picture. We draw p in green and q in orange (where p has four roots and q has three in between the four of p, and both have positive leading coefficient):



Now what we want to say is that the sum of these two polynomials is also real-rooted. Why is that the case?

If I look at the sum, then out here, beyond the last root of p, the sum is going to be positive (because both of these polynomials are positive).

But at the last root of q, q evaluates to 0 and p is negative, so their sum is negative. This means at some point in between, the sum has crossed over, so it hit 0. And you can use the same argument to find roots in all the other intervals as well — so between the third root of q and the fourth of p, there's a zero.

Similarly looking at the second root of q, at this point $q = 0$ and $p > 0$, so $p + q > 0$. Meanwhile, at the next root of q we have $p + q < 0$ (since p has become positive). So there's a root in the interval between them.

In any case, you can use interlacing to show there must exist zeros in each of these intervals between the zeros of q.

So the takeaway is that in general if I add two real-rooted polynomials I might not get a real-rooted one, but if they interlace then I do.

So the induction is not just going to be that the matching polynomials are real-rooted, but also that they satisfy a nice interlacing probability.

Student Question. *Doesn't this only give us* 3 *roots, while we have degree* 4*?*

Answer. Here the leading coefficient is positive and it's degree 4, so it goes to $+\infty$ at $-\infty$. And it's 0 at the last root of p, so we get one more root of $p+q$ somewhere to its left.

Definition 22.18. We say p and q are strictly interlacing if all th inequalities are strict.

§22.4.4 The induction

Now we'll do the actual induction. Our induction hypothesis will be the following:

Claim 22.19 — For all complete graphs on at most n vertices, and all positive weight functions $w : \binom{V}{2} \to \mathbb{R}_{>0}$, the following two statements are true:

(1) $\mathcal{M}_G(z)$ is real-rooted.

(2) $\mathcal{M}_{G-v}(z)$ strictly interlaces $\mathcal{M}_G(z)$ (for all $v \in V$).

Example 22.20

Imagine that G is a 4-cycle (this is not a complete graph, but you can still get (not necessarily strict) interlacing). Then we'll have

$$\mathcal{M}_G(z) = z^4 - 4z^2 + 2$$

(z^4 corresponds to the empty matching — the highest degree term always does; the $4z^2$ term corresponds to the four matchings with one edge; and then there are two matchings with two edges).

If I delete a vertex, then I also delete its incident edges, so I get a path on three vertices with 2 edges; then

$$\mathcal{M}_{G-v}(z) = z^3 - 2z.$$

We can compute the roots — the latter will have roots 0 and ± 2, while the first will have four roots symmetric about 0, and they will interlace.

Now let's do the inductive step (the base case is kind of immediate).

Proof of inductive step. Suppose we now have $n+1$ vertices. By Claim 22.16, we can decompose

$$\mathcal{M}_G(z) = \mathcal{M}_{G-v}(z) - \sum_{u \neq v} w_{uv} \cdot \mathcal{M}_{G-u-v}(z)$$

(where v is arbitrary). We know by the induction hypothesis that the polynomials $\mathcal{M}_{G-v}(z)$ and $\mathcal{M}_{G-u-v}(z)$ are real-rooted, and they interlace.

Now let's give some names to the roots of $\mathcal{M}_{G-v}$ — let $r_n < \cdots < r_1$ be the roots of $\mathcal{M}_{G-v}$. We know by interlacing that each of these other matching polynomials $\mathcal{M}_{G-u-v}$ are going to have

$$\operatorname{sgn}(\mathcal{M}_{G-u-v}(r_k)) = (-1)^{k-1}$$

for all k (they're all polynomials with leading coefficient 1, so they're positive for large inputs, and they alternate in sign at the r_k's by interlacing).

And we have a bunch of positive weight functions, so when we add up all these numbers, that'll preserve the signs; and $\mathcal{M}_{G-v}$ evaluates to 0 on all these inputs. So that implies

$$\operatorname{sgn}(\mathcal{M}_G(r_k)) = (-1)^k$$

for all $k = 1, \dots, n$ (the sign has changed once because we had a minus sign in front of the $\mathcal{M}_{G-u-v}$).

Then by the same argument that we used to prove that sums of interlacing polynomials had real roots, we can construct roots of $\mu_G(z)$ between the roots $r_1, \dots, r_n$ of μ_{G-v}. This will give real-rootedness, and it'll also give the interlacing hypothesis.

Drawing a picture, if we write out all the roots $r_8 < r_7 < \cdots < r_1$ on the number line, we know that $\operatorname{sgn}(\mathcal{M}_G(\bullet))$ is -1 at r_1, then $+1$ at r_2, then -1 at r_3, and so on. And because we have a leading coefficient of 1, we also know that somewhere off to the right, it's going to become positive.

So we know by the intermediate value theorem that it'll have a 0 in each one of these intervals. And there's also going to be one off to the left of r_8.

This is also exactly the interlacing property we needed between $\mathcal{M}_G$ and $\mathcal{M}_{G-v}$ — we've shown that the roots of $\mathcal{M}_G$ interlace those of $\mathcal{M}_{G-v}$ by the intermediate value theorem. □

Student Question. *Why did we assume the graph was complete?*

Answer. We wanted it to be strictly interlacing, to avoid some edge cases where one root lies on top of another. But that's more of a technical detail.

§23 April 30, 2025 — Random spanning trees

Today we'll continue discussing connections between analytic or algebraic properties of polynomials and properties of discrete distributions. The example for today's lecture will be the uniform distribution over *spanning trees* of a graph.

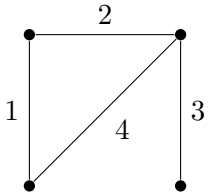
Example 23.1

Our input is a connected graph $G = (V, E)$. We define μ to be the uniform distribution over spanning trees of G.

This distribution, beyond being a very natural one to study, also has algorithmic applications — for instance, some of the state-of-the-art solvers for TSP use this as a basic subroutine.

Example 23.2

Suppose G is a triangle with one extra edge (with the triangle labelled 124, and the edge 3 hanging off the 24 vertex). Then the spanning trees are $\{123, 134, 234\}$, and μ will be uniform over these subsets of edges.



Last time, we took our distribution, encoded it as a polynomial, and used properties of that polynomial to say something about our distribution. Here we'll define a generating polynomial. It'll be a multivariate polynomial — we'll have one variable for every edge in the graph, and we define

$$g(z) = \sum_T \prod_{e \in T} z_e$$

(where the sum is over all spanning trees T). So we have one variable z_e for every edge. This is a multivariate polynomial; it's homogeneous of degree $n-1$ (where n is the number of vertices).

The goal for today's lecture is to show some remarkable properties of this polynomial, and use that to deduce properties of the uniform distribution over spanning trees.

Example 23.3

For the above graph, the corresponding polynomial is

$$g(z_1, z_2, z_3, z_4) = z_1z_2z_3 + z_1z_3z_4 + z_2z_3z_4.$$

§23.1 Kirchoff's matrix tree theorem

The first theorem is that you can write this polynomial as a determinant of a very natural matrix associated to this graph. In particular, this gives us a way to count the *number* of spanning trees in any graph.

Theorem 23.4 (Kirchoff's matrix tree theorem)

For any connected graph $G = (V, E)$, we have

$$g(z) = \frac{1}{-} \det\left(\frac{1}{n}\mathbf{1}\mathbf{1}^\intercal + \sum_{e \in E} z_e b_e b_e^\intercal\right),$$

where for every edge $uv \in E$, we define $b_{uv} \in \mathbb{R}^V$ to be the vector

$$b_{uv}(x) = \begin{cases} +1 & x = u \\ -1 & x = v \\ 0 & \text{otherwise.} \end{cases}$$

So we think of b_e as we're taking the edge e, and we arbitrarily pick an orientation of this edge (it doesn't matter which one I pick); if I orient it $u \to v$, then at the head of this oriented edge I put a $+1$, and I put a -1 on the tail (and 0 everywhere else). Think of it as essentially the indicator vector of an edge, except that we've introduced a sign to one of the endpoints.

To convince you that this is not some totally crazy matrix:

Fact 23.5 — We have $\sum_{e \in E} b_e b_e^\intercal = D_G - A_G$, where D_G is the diagonal matrix whose entries are the deegrees of the vertices, and A_G is the adjacency matrix.

(This matrix $D_G - A_G$ is sometimes called the Laplacian of G.) So this is just to say that the matrix appearing up there is not some totally contrived thing; it's fairly natural.

And the appearance of the all-1's matrix is also natural — b_e has a $+1$ and -1 entry and 0's everywhere else, so it's orthogonal to the all-1's vector. So the all-1's vector really makes sure that this matrix doesn't have a 0 eigenvalue.

Maybe this looks complicated; but at least, given a graph G, we can compute this matrix, and we can compute determinants in polynomial time. So this gives a polynomial time algorithm to *count* the number of spanning trees in a graph.

Student Question. *What are other applications of the Laplacian?*

Answer. The Laplacian encodes a bunch of properties of the graph. For example, it encodes cuts — if I look at $\mathbf{1}_S^\intercal(D_G - A_G)\mathbf{1}_S$, this counts the number of edges cut by S.

Laplacians also have a nice interpretation in terms of the theory of electric circuits and so on, but we won't get into that this lecture.

We'll also see later that this form for the generating polynomial also lets us deduce many interesting and useful properties about the uniform distribution over spanning trees.

But first we want to prove this theorem. To do this, we need one linear algebraic fact, which gives us a way to decompose a determinant of a sum of rank-1 matrices into a sum of slightly simpler determinants.

Fact 23.6 (Cauchy–Binet formula) — For all vectors $x_1, \dots, x_m, y_1, \dots, y_m \in \mathbb{R}^n$ (where $m \geq n$),

$$\det\left(\sum_{i=1}^m x_i y_i^\intercal\right) = \sum_{S \in \binom{[m]}{n}} \det\left(\sum_{i \in S} x_i y_i^\intercal\right).$$

We're not going to prove this identity (there's one on Wikipedia using characteristic polynomials), but we'll use it to prove Kirchoff's theorem.

Proof of Kirchoff. We have a determinant of a sum of a bunch of rank-1 matrices, so let's first apply Cauchy–Binet; then the right-hand side of the theorem statement can be decomposed as

$$\frac{1}{n} \sum_{T \subseteq E : |T| = n-1} \det\left(\frac{1}{n}\mathbf{1}\mathbf{1}^\intercal + \sum_{e \in T} z_e b_e b_e^\intercal\right) + \frac{1}{n} \sum_{F \subseteq E : |F| = n} \det\left(\sum_{e \in F} z_e b_e b_e^\intercal\right).$$

The first term is are all the subsets which include the all-1's vector; and the rest is all the subsets which don't. This is just applying Cauchy–Binet directly — I'm summing over all sets of vectors of size n, and I'm splitting into those which include the all-1's vector, and those which do not.

Now I want to simplify this expression.

Claim 23.7 — All the numbers in the second term are 0.

Claim 23.8 — For the first sum, if T is not a spanning tree, then we get a 0.

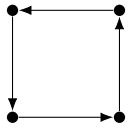
In particular, these two claims will reduce our very long expression just to a summation over *only* spanning trees.

Proof of Claim 23.7. Here we have a subset of edges of size exactly n (the number of vertices). In particular, this subset of edges must contain a cycle — if $|F| = n$, then F contains a cycle C.

And we claim the existence of a cycle already implies that the correspnding determinant must be 0. The reason for this is if I just look at the set of vectors $\{b_e \mid e \in C\}$, all I need to say is that this is linearly dependent — so there's no way the corresponding matrix in there can be full rank (if it's full rank the determinant is nonzero; if it's not full rank the determinant is 0).

So the key observation is that this set of vectors $\{b_e \mid e \in C\}$ is linearly dependent, and so $\sum_{e \in F} z_e b_e b_e^\intercal$ cannot have full rank.

This is actually not too difficult to see. Imagine I have a cycle on four vertices. And I get to orient these edges however I like. Imagine I've oriented them as a directed cycle.



Then it's easy to see that if I just sum all these vectors, I'll get 0 — every vertex has a directed edge going into it (contributing a $+1$) and one going out (contributing a -1), so $\sum_{e\in C} b_e = 0$. □

Proof of Claim 23.8. We're saying T is not a spanning tree, but it has exactly $n-1$ edges; so it must have at least two connected components (when we look at all the vertices) — i.e., (V,T) must have at least two connected components C_1 and C_2. Now we want to use these two connected components to construct a nonzero vector which is orthogonal to all the vectors in that sum; that will imply the matrix is also not of full rank (all those vectors lie in some $(n-1)$-dimensional subspace).

To see this, we can look at the following vector (similarly to what we did when looking at eigenvalues vs. ergodicity of Markov chains) — consider the vector

$$x = \frac{1}{|C_1|}\mathbf{1}_{C_1} - \frac{1}{|C_2|}\mathbf{1}_{C_2}.$$

This vector is clearly orthogonal to the all-1's vector. Moreover, because every single edge in T is fully contained in one of these components, it's also going to be orthogonal to each one of those vectors b_e (for $e \in T$). So $x \perp \mathbf{1}$ and $x \perp b_e$ for all $e \in T$. This also implies $\frac{1}{n}\mathbf{1}\mathbf{1}^\intercal + \sum_{e\in T} z_e b_e b_e^\intercal$ doesn't have full rank. □

These two claims mean we can simplify our expression significantly — we can ignore all the terms in the second sum, and we can also restrict our attention to just spanning trees in the first sum. So these two claims imply

$$\mathrm{RHS} = \frac{1}{n}\sum_{T\subseteq E \text{ spanning tree}} \det\left(\frac{1}{n}\mathbf{1}\mathbf{1}^\intercal + \sum_{e\in E} z_e b_e b_e^\intercal\right).$$

And now all we need to show is that each one of these terms is what we want.

Claim 23.9 — If T is a spanning tree, then we have

$$\frac{1}{n}\det\left(\frac{1}{n}\mathbf{1}\mathbf{1}^\intercal + \sum_{e\in E} z_e b_e b_e^\intercal\right) = \prod_{e\in T} z_e.$$

Proof. Let's massage this thing a little bit — we'll collect everything into a matrix. So we define $M_T \in \mathbb{R}^{n\times n}$ by taking these vectors as the columns of M_T — i.e.,

$$M_T = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{1}/\sqrt{n} & \sqrt{z_{e_1}} b_{e_1} & \cdots & \sqrt{z_{e_{n-1}}} b_{e_{n-1}} \\ | & | & \cdots & | \end{bmatrix}.$$

Then the thing we want is just

$$\frac{1}{n}\det(M_T^\intercal M_T).$$

So now let's look at what this product of matrices looks like. We can write it as a block matrix where we have a 1 in the top-left corner, 0 for the remaining parts of the first row and first column, and beneath it

we'll have $B_T^\intercal B_T$, where we call $B_T \in \mathbb{R}^{n\times(n-1)}$ the matrix consisting of the rightmost $n-1$ columns of M_T (the ones with the b_{e_i}'s). In particular, because this matrix has these two blocks, we'll have

$$\det(M_T^\intercal M_T) = \det(B_T^\intercal B_T).$$

And now we want to calculate what this thing is.

So now let's do Cauchy–Binet again — I have a product of matrices which are not square, so I can use Cauchy–Binet again. Then we have

$$\frac{1}{n}\det(B_T^\intercal B_T) = \frac{1}{n}\sum_{U\subseteq V:|U|=n-1}\det(B_{T,U}B_{T,U}^\intercal),$$

where $B_{T,U}$ means we delete the row corresponding to whatever vertex was not included in U. In other words, B_T is some matrix where the columns are indexed by edges of T, and then we delete one vertex $u \in V$ (so $U = V\setminus\{u\}$), and now we have all the remaining rows and we look at the submatrix in here.

So really we're thinking about this matrix where the rows are indexed by vertices and columns by edges; and there's going to be a nonzero entry between a given row corresponding to a vertex and column corresponding to an edge if and only if that vertex is an endpoint of that edge.

In particular, now we have a determinant which is a product of two square matrices, so it's really equal to just the determinant of one of them squared; so we get

$$\frac{1}{n}\sum_{|U|=n-1}\det(B_{T,U})^2$$

(because the determinant of a product of two square matrices is the product of determinants, and the determinant of a transpose is the same as the determinant of the original). So really the claim we want to show boils down to — or at least, is implied by — showing that

$$\det(B_{T,U}) = \sqrt{\prod_{e\in T} z_e}$$

for all $U \subseteq V$ with $|U| = n-1$. So now we just need to verify this last claim.

So now let's prove this. To make this calculation convenient, the key claim is the following:

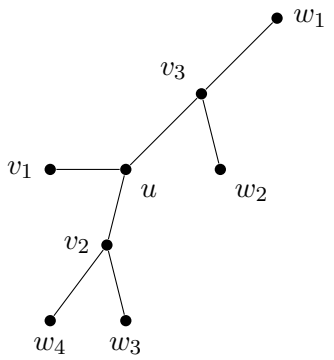
> **Claim 23.10 —** We can write $B_{T,U}$ as an upper triangular matrix, where along the diagonal we have $\sqrt{z_e}$'s, below the diagonal we have 0's, and above the diagonal we have some stuff (which doesn't matter).

This is convenient because if I have an upper triangular matrix, then the determinant is just the product of diagonals (if you use the definition of a determinant in terms of summing over permutations, the moment you pick an entry below the diagonal you get a 0).

When we say 'write,' we mean that we're going to *order* the edges and vertices in a certain way so that it has this form.

Assuming we have this claim, we'll have proved the theorem. So now let's just construct this ordering — how to order the vertices (or rows) and edges (or columns).

We'll do this ordering just by running a BFS starting from the special vertex u that we removed earlier. As a simulation of how this will work, we'll prove it by picture:



We'll write down B_T, and if you remove thee first row we'll get $B_{T,U}$.

So on the first row, I have u. Then on the first round of BFS I see v_1, v_2, v_3. And in the columns, I'll write down the corresponding edges (in the order I traversed them in) — so I write down the edges uv_1, uv_2, and uv_3. For convenience let's set all the z_e's to 1, so I get

$$\begin{bmatrix} -1 & -1 & -1 \\ +1 & 0 & 0 \\ 0 & +1 & 0 \\ 0 & 0 & +1 \\ \mathbf{0} & & \end{bmatrix}.$$

Now I'll continue my BFS. I'll explore w_3 and w_4 (from v_2) and w_1 and w_2 (from v_3). And I explore the corresponding edges in the same order; so I put in v_2w_3, v_2w_4, v_3w_1, v_3w_2.

Now the key insight is when I fill in these entries, every time I hit a new vertex, I'm going to put the corresponding nonzero entry in the corresponding diagonal; and for the vertex I've already visited, that's only going to go above the diagonal (into the upper-right block of the matrix). So this will tell us the resulting matrix is upper triangular. So here we'll have

$$\begin{bmatrix} -1 & -1 & -1 & & & & \\ +1 & 0 & 0 & & & & \\ 0 & +1 & 0 & -1 & -1 & & \\ 0 & 0 & +1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & +1 & & & \\ & & & & +1 & & \\ & & & & & +1 & \\ & & & & & & +1 \end{bmatrix}.$$

□

So this concludes the proof of the matrix tree theorem. □

§23.2 Negative correlation for spanning trees

Now we'll say some results about the properties of uniform spanning trees that you get from this.

Theorem 23.11

Let μ be the uniform distribution over spanning trees in a connected graph $G = (V, E)$. Then for all pairs of distinct edges $e, f \in E$, we have

$$\mathbb{P}_{T\sim\mu}[f \in T \mid e \in T] \le \mathbb{P}[f \in T].$$

So this is a correlation inequality — if I sample a random tree from μ, the probability it contains both these edges is always at most the product of probabilities of each individual edge. Or written another way, the probability that f is in T conditioned on that e is is at most the unconditional one.

In other words, if I sample a random spanning tree and I tell you some edge e is in this tree, that's going to *decrease* the likelihood that some other edge f is in the tre.

And further, we can strengthen this statement to condition on other edges as well:

Theorem 23.12

Let μ be the uniform distribution over spanning trees in a connected graph $G = (V, E)$. Then for all pairs of distinct edges $e, f \in E$ and all $F \subseteq E \setminus \{e, f\}$, we have

$$\mathbb{P}_{T\sim\mu}[f \in T \mid e \in T, F \subseteq T] \leq \mathbb{P}[f \in T \mid F \subseteq T].$$

So this is negative correlation for *all* possible conditional distributions.

Remark 23.13. This is a very nice property because it turns out you can deduce many nice consequences from this. For instance, you can show if you have negative correlation for all conditional distributions, then Lipschitz functions have sub-Gaussian concentration.

It also implies that simple natural local Markov chains for sampling spanning trees mix rapidly. There's a nice paper by Garbe–Vondrak for the concentration statement, and Feder–Mihail for the mixing statement.

But in any case, this establishes some very nice properties in a setting where we do not have independence (and where things like Dobrushin and path coupling are not available to us).

§23.3 Real stability

Now we'll prove (or at least sketch) this, using the fact that we can write our generating polynomial as a determinant.

To do this, we'll revisit the notion of zero-freeness; but now we want to define what zero-freeness means in the context of multivariate polynomials.

Definition 23.14. We say a polynomial $p(z_1, \dots, z_n)$ is real stable if:

- p has real coefficients.
- $p(z_1, \dots, z_n)$ is nonzero if $\mathrm{Im}(z_i) > 0$ for all i.

So in other words, you should think of this as being zero-free with respect to the entire upper half plane in the complex plane.

When you first see this, it might seem like a rather strange definition. But to get a little bit of intuition, let's see an equivalent way to phrase real stability.

First, if p has only *one* variable, then real stability is equivalent to real-rootedness. The reason for this is because we're assuming it has real coefficients, so we know all its roots come in complex conjugate pairs. So if it has a complex root not on $\mathbb{R}$ with negative imaginary part, then it must also have a root with positive imaginary part. So this is one way to generalize real-rooted polynomials to multivariate polynomials.

Fact 23.15 — For a univariate polynomial, it's real stable if and only if it's real-rooted.

Here's one lemma, which is an extension of this and is not hard to prove:

Lemma 23.16

The following are equivalent:

(1) p is real stable.

(2) The univariate polynomial $t \mapsto p(x+tv)$ is real-rooted for all $x \in \mathbb{R}^n$ and all $v \in \mathbb{R}^n_{>0}$.

The second condition says that if we look at any univariate restriction in the direction of a positive vector, that restriction is real-rooted.

What is the connection with negative correlation?

Proposition 23.17

Let μ be a distribution on $2^{[n]}$, and let g_μ be its multivariate generating polynomial

$$g_\mu(z_1, \dots, z_n) = \sum_{S \subseteq [n]} \mu(S) \prod_{i \in S} z_i.$$

If g_μ is real stable, then μ is negatively correlated, i.e., for all $i \neq j$,

$$\mathbb{P}_{S \sim \mu}[i, j \in S] \leq \mathbb{P}_{S \sim \mu}[i \in S] \mathbb{P}_{S \sim \mu}[j \in S].$$

This conclusion is the same statement as $\mathbb{P}[f \in T \mid e \in T] \leq \mathbb{P}[f \in T]$ for spanning trees (the version without the additional conditioning on F).

Before we prove all this stuff, we'll at least say one more thing, which is what we'll combine with this proposition to prove the theorem — that the corresponding polynomial for spanning trees is indeed real stable.

Theorem 23.18

For all positive semidefinite matrices $A_1, \dots, A_m \in \mathbb{R}^{n \times n}$, the polynomial $\det(\sum_{i=1}^m z_i A_i)$ is real stable.

In particular, this means the generating polynomial for spanning trees is real stable (for all graphs G).

So if you combine this theorem with Proposition 23.17, then we get that the uniform distribution over spanning trees must be negatively correlated.

Let's start by proving this proposition, since it's maybe a surprising connection between probability and zeros, though it's of a similar flavor to what we showed last lecture.

Proof of Proposition 23.17. Let's fix $i \neq j$. What we want to do is marginalize out all the other variables and just focus on z_i and z_j — we want to reduce this to a simple bivariate polynomial of degree 2. So what we're going to do is plug in 1's everywhere else — we let

$$p(z_i, z_j) = g_\mu(\mathbf{1}, z_i, \mathbf{1}, z_j, \mathbf{1}).$$

So we're only going to leave z_i and z_j as my two alive variables; for everyone else, I'm ust going to plug in 1's. What this looks like is, I'm really going to get four different kinds of sets. I have all the sets which neither contain z_i nor z_j, all the sets containing one of z_i or z_j, and all the sets that contain both. So we get

$$p(z_i, z_j) = \sum_{S \not\ni i,j} \mu(S) + z_i \sum_{S \ni i, S \not\ni j} \mu(S) + z_j \sum_{S \ni j, S \not\ni i} \mu(S) + z_i z_j \sum_{S \ni i,j} \mu(S)$$

(we're plugging in 1's for all the other variables). And these are very nice probabilities we can interpret — they're just the marginals of i and j. So

$$p(z_i, z_j) = \mathbb{P}_{S\sim\mu}[i, j \notin S] + \mathbb{P}_{S\sim\mu}[i \in S, j \notin S] + \mathbb{P}_{S\sim\mu}[j \in S, i \notin S] + \mathbb{P}_{S\sim\mu}[i, j \in S].$$

So now I've got a simple bivariate polynomial of degree 2, with only 4 terms, and the coefficients correspond to the marginals of i and j.

And because I'm plugging in real number to g_μ, if g_μ is real stable, then this bivariate polynomial is also real stable (if I plug in a complex number with strictly positive imaginary part, it's going to be nonzero).

So now we just have to work with this bivariate polynomial that's real stable.

It'll be more convenient to work with another slight change of variables: we define

$$q(x, y) = p(1 + x, 1 + y)$$

(so we're replacing z_i with $1 + x$ and z_j with $1 + y$). This just simplifies the polynomial slightly, and we get

$$q(x, y) = 1 + \mathbb{P}_{S\sim\mu}[i \in S]x + \mathbb{P}_{S\sim\mu}[j \in S]y + \mathbb{P}_{S\sim\mu}[i, j \in S]xy.$$

And we know this is real stable.

So now we just need to study bivariate degree-2 real stable polynomials. Here's the key lemma:

Lemma 23.19

A polynomial $a + bx + cy + dxy$ is real stable if and only if we have $bc \geq ad$.

And that's exactly what we want — we want the product of the marginals to be at least the probability that both elements are in the set.

And actually the proof of this key lemma is a very quick calculation.

Proof. What does it take for this polynomial to evaluate to 0? We can factor this out as $a + bx + y(c + dx)$, so it's 0 if and only if

$$y = -\frac{a + bx}{c + dx}.$$

So we want to argue that if x has a positive imaginary part, then y must have a negative imaginary part (if and only if that inequality is true): We want to show that the statement

$$\operatorname{Im}\left(-\frac{a + bx}{c + dx}\right) \leq 0 \text{ for all } x \text{ with } \operatorname{Im}(x) > 0$$

holds if and only if $bc \geq ad$.

And this can be done just by directly calculating what this imaginary part is — it turns out that

$$\operatorname{Im}\left(-\frac{a + bx}{c + dx}\right) \propto (ad - bc)$$

(where the proportionality constant is positive). □

□

In the last five minutes, we'll prove the theorem — that these determinantal polynomials are real stable — so that we have a complete proof of everything.

Proof of Theorem 23.18. The key idea is to reduce to the fact that we know the characteristic polynomial of every symmetric matrix has real roots (in other words, every symmetric matrix has real eigenvalues).

We're going to use Lemma 23.16. We'll also assume these matrices are positive *definite*, rather than positive semidefinite; we can do this without loss of generality by standard limiting arguments.

So we assumee $A_1, \dots, A_n \succ 0$. And we'll use the definition of real stability in Lemma 23.16 — we fix $x \in \mathbb{R}^n$ and $v \in \mathbb{R}^n_{>0}$, and we'll show that

$$\det\left(\sum_{I=1}^{m}(x_i + tv_i)A_i\right)$$

is real-rooted (this is a univariate polynomial in t).

To do this, we'll split this thing up and use the fact that the A_i's are positive definite — we can split this as

$$\det\left(\sum_{i=1}^{m} x_i A_i + t\sum_{i=1}^{m} v_i A_i\right),$$

where the $v_i A_i$ are positive definite and $\sum_{i=1}^m x_i A_i$ is symmetric. Let's give these guys names; so we call $\sum_{i=1}^m x_i A_i = B$ and $\sum_{i=1}^m v_i A_i = C$.

Because C is positive definite, we can take square roots and inverse square roots; so this thing is equal to

$$\det(C) \cdot \det(tI + C^{-1/2} B C^{-1/2})$$

(here we're taking all the eigenvalues, and taking 1 over their square root to get $C^{-1/2}$; and this is well-defined because C is positive definite).

So now I have the characteristic polynomial of some symmetric matrix, and I know it always has real roots. So that's it — that also proves every determinantal polynomial (with positive definite, or more generally PSD, matrices) is real stable. □

Next week Kuikui will not be here. We'll have guest lecturers by Guy Bresler, who will be talking about percolation in $\mathbb{Z}^2$.

§24 May 5, 2025

§24.1 Introduction to percolation

It's raining. Imagine you are a tree, and you've lived a long time and your roots are deep.

> **Question 24.1.** Am I going to get any of the amazing water falling from the sky?

What does that depend on? The water lands on the earth and gradually filters through the dirt, and may or may not reach the roots of this tree.

We're going to try to understand this question by studying a math model for it, called *percolation*. We'll specifically study this on the lattice $\mathbb{Z}^2$.

Here's a few reasons you might care about this model. It's sort of a classic probability model. You'll see things that look sort of like percolation all over the place. For example, random graphs are percolation on K_n; we'll see percolation on $\mathbb{Z}^2$. There's also all kinds of associated techniques from discrete probability that will come up.

There's also deep connections to Boolean analysis (analysis of Boolean functions); in fact lots of Boolean analysis used in math and CS was invented in order to study percolation.

Finally, a related reason is *sharp threshold* phenomena. These show up all over the place (in physics, they're also known as phase transitions) — you cool water, and all of a sudden it freezes. We'll see similar things here.

§24.2 The model

We'll start with the graph $\mathbb{Z}^2$, where we have points in a grid and edges between nearest neighbors. So this is our graph. You could start with any other graph, but this is the classic one.

And what we'll do is we keep each edge with probability p, independently. This induces a probability measure over the infinite configuration of edges (if they exist or not); we'll denote that by $\mathbb{P}_p$.

Definition 24.2. Edges that we keep are called open.

This alludes to the flow of water through dirt or a coarse rock — an edge being open (or retained) means that water can flow through it.

From this measure, we get a random configuration of edges, which we denote by $\omega \in \{0,1\}^{E(\mathbb{Z}^2)}$.

This is a (discrete) probability model. It's so simple -0- it's just a ton of Bernoullis — but you can ask a ton of questions, and it's quite rich.

§24.3 Infinite components

One question you might care about:

Definition 24.3. We define $\theta(p) = \mathbb{P}_p[0 \overset{\omega}{\leftrightarrow} \infty]$.

In words, this is the probability that the origin is in an infinite component (in our random configuration ω). We keep each edge with probability p and that gives you a bunch of components. We consider the component in which the origin 0 sits, and ask, does this thing go out to ∞?

There's another number that you might care about. This is asking, is 0 in an infinite component? But we can ask another question:

Question 24.4. Does there *exist* an infinite component?

It turns out the existence of an infinite component can only have probability 0 or 1, not in between (this follows from Kolgomorov's 0–1 law, though we're not expected to have seen that before).

But we're going to study $\theta(p)$, which is very concrete — it's a number between 0 and 1.

In fact, the way it looks — we'll plot it with p on the x-axis, and the probability on the y-axis.

If $p = 0$, the probability is 0 (there are no edges).

It's not obvious that it doesn't just start going up. But it turns out that in fact, it stays 0 for a while. And then at some point, it starts going up.

And if $p = 1$, then I've retained the whole lattice, so $\theta(p) = 1$.

We define the critical probability p_c as the point where $\theta(p)$ becomes nonzero.

So as soon as my probability exceeds this critical threshold, I have a positive probability of having the origin in an infinite component.

Question 24.5. What is p_c?

It turns out that p_c was a big question studied by mathematical physicists and probabilists in the mid-1900s for decades. Of course, physicists had guesses and heuristic arguments, which were non-rigorous; it took until 1980 to identify it.

Theorem 24.6 (Kesten 1980)
We have $p_c = \frac{1}{2}$.

We should have defined p_c (we just did it by picture):

Definition 24.7. We define $p_c = \inf\{p \mid \theta(p) > 0\}$.

So I'm looking at all p for which $\theta(p)$ is positive, and I look at the greatest lower bound (basically the minimum). So the point is once you take a value bigger than p_c, then $\theta(p)$ is strictly positive. And again by the Kolgomorov 0–1 law, even though the origin has probability 0.0001 of being in an infinite component, with probability 1 there exists an infinite component *somewhere*.

This is the theorem we'll do for this lecture and next — we'll pretty much fully rigorously prove it. We only need to black-box one thing. It's a sort of isoperimetric inequality. We've seen a bit of isoperimetry; we're going to talk a bit about isoperimetry on the Boolean hypercube, and there'll be a statement we won't prove because it's a bit tedious. But other than that, we'll prove everything! It's pretty wild that it's doable in 2 lectures; that's because the proofs have been pruned and made more beautiful over the years.

§24.4 The lower bound

We'll show that $\theta(\frac{1}{2}) = 0$, which implies that $p_c \geq \frac{1}{2}$.

The key idea, which is ingenious and also quite obvious in retrospect, is to consider the *dual* lattice. For every edge in the original graph (before we keep or throw away edges), we consider the unique edge that bisects it — if it goes between (a, b) and $(a+1, b)$, we consider the edge $(a+\frac{1}{2}, b+\frac{1}{2})$ to $(a+\frac{1}{2}, b-\frac{1}{2})$.

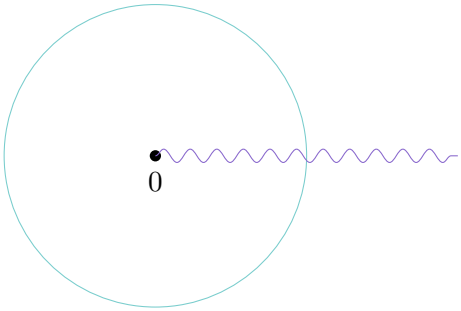
The second thing is, we should define a probability measure on this dual graph. What we'll do is, using ω_e to denote the indicator for a particular edge in the original, we set

$$\omega_e = 1 \text{ if and only if } \omega_e^{\text{dual}} = 0.$$

So you basically just flip every edge.

Why in the world is this useful? It turns out there's a very easy description of when the origin lies in an infinite component, in terms of the dual.

If there's a (closed) loop around 0 in the dual process, you can convince yourself that there's no way for a path from the origin to leave to ∞ — if you imagine a path out from 0 to ∞, there's some edge where this infinite path escapes the loop, and then we can't have included that loop edge in the dual.



So to write this:

> **Claim 24.8 —** The event $\{0 \overset{\omega}{\leftrightarrow} \infty\}$ occurs if and only if there is not a closed circuit in the dual process.

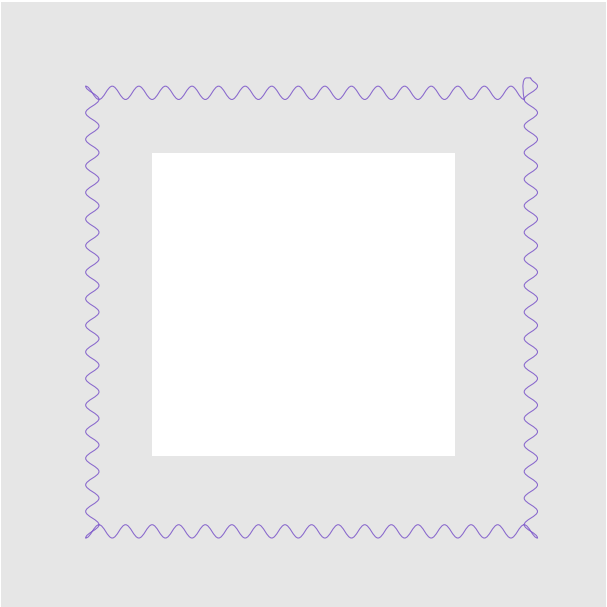
(We only emphasized one direction, but it is an if and only if; the other direction is slightly less obvious, and we don't need it.)

§24.5 Closed circuits in the dual

We're trying to show that percolation doesn't happen at $p = \frac{1}{2}$, which means we want to show there exists a circuit in the dual lattice.

What we'll do is consider square annuli, where the inside square has length 2ℓ and the outside square has length 6ℓ.

And we'll let C_k be the event that there is a closed circuit within this annulus with $\ell = 4^k$.



We want to show these things exist. The calculation is a bit straightforward, but there is one idea. You want to show that something definitely happens. What's one way to do that? Well, say we want to show there is definitely one coin toss that lands heads. You can take any tiny probability, say 10^{-10} for heads, and if I toss it enough times, eventually there will be a heads; as long as those coin tosses are independent. I want to show something happens, so I just have many, many trials to make it happen. That's what we're doing here. That's why the 2ℓ and 6ℓ and 4^k are all here — what happens is you get this little annulus here; and then this is growing quickly enough that the next annulus is much bigger. So these events are each

functions of *disjoint* sets of independent edges, and therefore the event that there's a closed circuit within each annulus is independent of the others.

So these C_k's are *independent* events, because they're functions of disjoint edges.

Claim 24.9 — There exists $c' > 0$ such that $\mathbb{P}_{1/2}[C_k] > c'$ for all $k \geq 1$.

So this holds uniformly in k — it doesn't matter how big the annulus is. There's some kind of scale invariance where as I blow these things up, these things have a uniform lower bound. (In fact, it turns out they have a limit, though it'd take a semester to develop the machinery for that.)

This will imply the thing we want. Why? It implies that

$$\mathbb{P}_{1/2}[\text{exists a closed circuit around } 0] = 1 - \mathbb{P}_{1/2}[\text{no closed circuit}].$$

And that's at least

$$1 - \mathbb{P}_{1/2}[C_k^{\mathsf{c}} \text{ occurs for all } k \geq 1].$$

And these are all independent, so this is at least

$$1 - \prod_{k \geq 1} (1 - c') = 1.$$

So we're done. (This is just the arithmetic for saying that if you toss a coin infinitely many times with positive heads probability, we're guaranteed to get a heads at some point.)

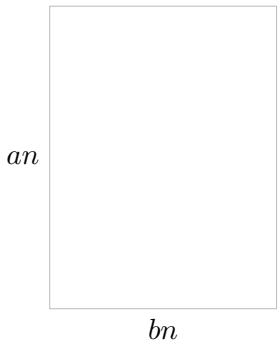
At this point we should ask, why in the world is this claim true? This is something quite interesting. That's what we'll do in the next lecture.

The reason this is true is because of a more general machinery, called Russo–Seymour–Welch theory. It's encapsulated in the following theorem.

Theorem 24.10 (Russo–Seymour–Welch)

Let $a, b > 0$. Then there exists $c > 0$ (which is a function of a and b) such that for all $n \geq 1$, if we let $H_{a,b}(n)$ be the event that there is a left-to-right crossing in an $an \times bn$ rectangle, we have

$$c < \mathbb{P}_{1/2}[H_{a,b}(n)] < 1 - c.$$



What is the content of this theorem? The point is that c and $1-c$ don't depend on n — this holds uniformly in n (for rectangles of a given aspect ratio). This is a priori not obvious — if we have $n = 10^{10}$ vs. $n = 10$, we get the same uniform bound. It's between 0.2 and 0.8 or something; it's not going to 0 or 1.

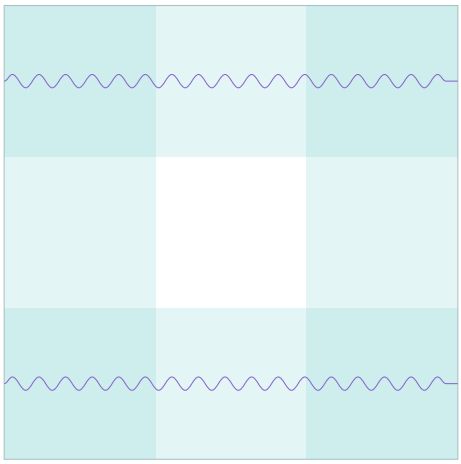
First let's see how to prove the claim using this.

Proof of Claim 24.9. We have side-lengths of 2ℓ and 6ℓ. First, Theorem 24.10 implies that $\mathbb{P}[H_{2,6}(\ell)] \geq c$ for some $c > 0$. Of course, 2 and 6 are no coincidence. We'll identify our favorite 2×6 aspect ratio rectangle, and consider the probability of a horizontal crossing event; and this is constant.

Now, how are we going to get a cycle? We can consider more $2\ell \times 6\ell$ rectangles — at the bottom and left and right of the picture. If we have all four of those crossings (in the long directions of a $2\ell \times 6\ell$ rectangle), well, we get our circuit. So we get

$$\mathbb{P}_{1/2}[C_k] \geq \mathbb{P}_{1/2}[H_{2,6}(4^k)]^4 \geq c^4.$$

But the question you should be asking yourself is, why do we get to write the first inequality? There's something called the FKG inequality (something like that is in our final problem set, so we will know about it very soon).



As a quick reminder:

Definition 24.11. We say a function $f : \{0,1\}^n \to \mathbb{R}$ is monotone (increasing) if for $x \preceq y$ (meaning $x_i \leq y_i$ for all i), we have $f(x) \leq f(y)$.

In our context, this means adding edges only makes the event go from not happening to happening. (The space we're working in is edges; if you add edges, f asks does there exist a lengthwise crossing, and by adding edges you can only make this event happen.)

So we indeed have monotone events.

Theorem 24.12 (FKG inequality)

For two increasing functions f and g, we have $\mathbb{E}[f(x)g(x)] \geq \mathbb{E}[f(x)]\mathbb{E}[g(x)]$.

In words, this says monotone events are positively correlated. By dividing, you can also formulate this in terms of conditioning.

This seems sort of obvious, but it's very useful, in many places; percolation is maybe one of the most powerful. But if you study random graphs and events like 'is there a Hamiltonian circuit' or 'is there a clique of size k,' all these events are increasing; so you can immediately say that if someone tells you there is a Hamiltonian cycle, the probability of every other increasing event only goes up.

So that justifies this, because these are increasing events. □

Student Question. *Why did you use 2ℓ and 6ℓ?*

Answer. It probably shouldn't matter, as long as the events C_k are functions of independent edges.

§24.6 Proof of Theorem 24.10

The final thing is, everything has been done except for RSW. So we want to prove this statement about crossing probabilities being scale-invariant — all they depend on is the aspect ratio.

Suppose we fix some aspect ratio a and b. If $b \gg a$, it's going to be harder to get a crossing.

There's a straightforward monotonicity that the probability only decreases with b (for example, if you have a horizontal crossing of the full thing, you also have a horizontal crossing of the half-length thing).

§24.6.1 Step 1 — reduction to $2n \times 3n$

We want to show this is true for all a and b. Step 1 is:

Claim 24.13 — It suffices to show the statement for the particular aspect ratio $2n \times 3n$.

So we're trying to show there exists c such that

$$c \le \mathbb{P}_{1/2}(H_{2,3}(n)) \le 1 - c$$

for all n.

So let's prove that. But for that, we need a bit of a diversion.

Claim 24.14 — We have $\mathbb{P}_{1/2}(H_{1,1}(n)) = 1/2$.

So first, we're considering horizontal crossings of a *square*. How do you prove this?

Proof. There's a horizontal crossing in our square if and only if there is no vertical crossing in the dual process. So we can write

$$\mathbb{P}_{1/2}[H(\square)] = 1 - \mathbb{P}_{1/2}[V(\square)]$$

(where crucially, the dual process still has probability $1/2$ — we're at the *self-dual* point where the process and dual process are the same, since we keep edges there with probability $1 - p$). And by symmetry, the event you have a vertical crossing is the same as a horizontal crossing; so we get $\mathbb{P}_{1/2}[H(\square)] = \frac{1}{2}$. □

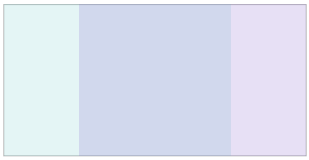
Now we're going to prove Step 1. We suppose that

$$c \le \mathbb{P}_{1/2}[H_{2,3}(n)] \le 1 - c.$$

And we're going to show how to 'bootstrap' that to get to $2n \times 4n$.

What we do is we add little lines at x-coordinate n and $3n$, so now we have a $2n \times 2n$ square in the middle, and $2n \times 3n$ rectangles on the left and right (overlapping on that square).

And now we can consider the events that we have a horizontal crossing of the $2n \times 3n$ rectangle on the left, the $2n \times 3n$ rectangle on the right, *and* a vertical crossing of the square in the middle. So we have three separate events — this $2n \times 3n$ horizontal crossing in red, the yellow crossing which is again a $2n \times 3n$ horizontal crossing, and then the blue vertical crossing in the square.



If all three of these things happen, then I have a horizontal crossing of the $2n \times 4n$ rectangle. So this tells us

$$\mathbb{P}[H_{2,4}(n)] \geq \frac{1}{2} \cdot \mathbb{P}[H_{2,3}(n)]^2$$

(the $\frac{1}{2}$ comes from the vertical crossing of the square).

Each of these is a constant, so this is also going to be bounded by a constant. (We'll just focus on the lower bound.) Here we used FKG as well, since these are positively correlated events.

You can keep doing this to stitch things together and make the aspect ratio grow arbitrarily.

§24.6.2 Step 2 — proof for $2n \times 3n$

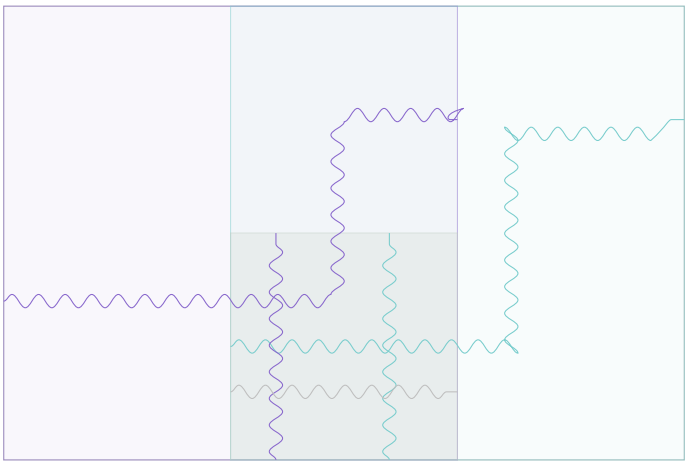
Step 2, of course, is to show the claim for the 2×3 rectangle. As a little hint, you can in fact show that

$$\mathbb{P}_{1/2}[H_{2,3}(n)] \geq 2^{-7}.$$

We're trying to leverage the same sorts of ideas we've been playing with — FKG and the fact monotone events are correlated. And we want to construct a left-to-right crossing.

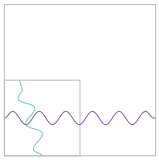
What can we start with? Right now, all we know is that the probability of crossing a square is $\frac{1}{2}$. So if we're going to start with a square, let's make some squares. We draw a $2n \times 2n$ square on the left and the right; and let's draw crossings. But before we do that, we'll also draw $n \times n$ squares in the middle.

Let's draw a horizontal crossing of the middle-bottom square, and two vertical crossings (in red, yellow, and blue, respectively). The magic is, I want to consider now a crossing of the $2n \times 2n$ square on the right. Let's consider a horizontal crossing. But I don't just want to consider *any* horizontal crossing; I want to consider a horizontal crossing that hits my blue vertical crossing. And similarly with the left — I want to construct a horizontal crossing that hits my yellow vertical crossing.



Then the question is, why does this happen with some probability bound? We're considering a left-to-right crossing that happens to hit this little crossing of the little square. And the claim is that we can bound that. That's quite subtle.

What we'll do is consider just the blue 2×2 square on the right; so we have this little 1×1 in the bottom-left. And we're interested in the probability of having a vertical crossing of the small one, and a crossing from the right of the big square that hits it.

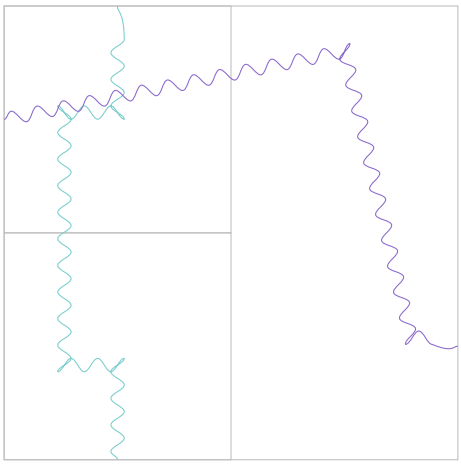


We claim this is at least the probability of having a vertical crossing of the little square (which is 1/2), and a horizontal crossing, times $\frac{1}{2}$ — the subtle claim is that we claim that a horizontal crossing has probability at least 1/2 of hitting this little blue crossing.

Once we have this, we're good to go (we get 1/8 from this picture, another 1/8 from the yellow picture, and then there's the red horizontal crossing of the 1×1).

So let's condition now that for sure, there's some little vertical crossing and big horizontal crossing. Then we claim the probability there's a horizontal thing from the right hitting the vertical crossing is at least $\frac{1}{2}$. How do we show this?

The point is we reflect this *exact* vertical crossing. Any horizontal crossing has to hit either this vertical crossing, or its reflection; and those two are equal by symmetry. So each has probability at least $\frac{1}{2}$.



§24.7 Forecast

This time, everything was very elementary. Next class will have a bit more machinery and connections to other areas in probability and CS and so forth.

We're going to show the in some ways more interesting direction — that if $p > \frac{1}{2}$, then $\theta(p) > 0$ — there's a positive probability that the origin is an infinite component.

The key object is to consider a $n \times 2n$ (technically $(n-2) \times 2n$) rectangle. We consider the more difficult crossing direction, and call that event J_n. The main thing — the first thing is that J_n is a function of all the indicators of edges within the rectangle, so it's a Boolean function $J_n : \{0,1\}^N \to \{0,1\}$ (where $N \approx 2n^2$).

What we're going to do is, we're going to understand this function — if we consider the probability that J_n occurs as we vary p, this probability will be very small for a while, and then it's very suddenly going to steeply grow for a while, adn get very close to 1. This is what's known as a *sharp threshold* — when it goes from ε to $1-\varepsilon$ in a very narrow window. We'll show this sharp threshold.

And then we'll bootstrap this to argue that the percolation threshold is 1/2. What we'll do is show if you're slightly above 1/2, the derivative of this probability is huge, and so the probability of this crossing will be very close to 1 (not just positive); and we'll use that to construct the percolation event we care about.

§25 May 7, 2025

§25.1 Setup

Today we're going to see the other half of the proof of why the percolation threshold is $\frac{1}{2}$.

We're looking at the 2D lattice where we retain every edge with probability p. And we're asking, what's the probability that the origin is contained inside an infinite component? So that's what we're talking about. And what we want to show today is:

Theorem 25.1
If $p > \frac{1}{2}$, then $\theta(p) = \mathbb{P}_p[0 \overset{\omega}{\leftrightarrow} \infty] > 0$.

This is the percolation event — that we have a positive probability of a specific vertex being in an infinite component. (As we said last time, by Kolgomorov's 0–1 law, this means with probability 1 there's an infinite component somewhere.)

§25.2 The finite size criterion

The key proposition is a surprising statement about crossing probabilities.

Notation 25.2. Let J_n be the event that we cross a $2n \times (n-2)$ rectangle in the long direction.

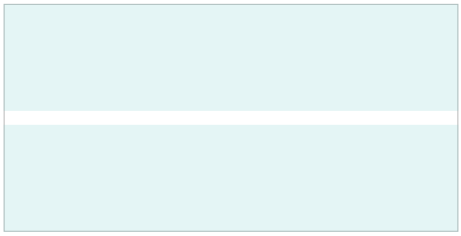
Proposition 25.3 (Finite size criterion)
If there exists n such that $\mathbb{P}_p[J_n] \geq 0.98$, then almost surely there exists an infinite component.

So whatever p is, if it's such that the probability of this crossing is pretty big, then boom! you get an infinite component. This isn't what we actually wanted to show — we wanted to say that if $p > \frac{1}{2}$ we get an infinite component — but this seems like a massive reduction in the effort it'll take us.

So this is what we'll dproveo to start with.

Proof. Let $\varepsilon \leq 0.02$ (this is just to avoid pushing around 0.98 and so on). We're going to show that if $\mathbb{P}_p[J_n] \geq 1 - \varepsilon$, then $\mathbb{P}_p[J_{2n}] \geq 1 - \frac{\varepsilon}{2}$ — so if we consider J_{2n}, it actually has much *higher* probability. This is quite striking — it says that if we consider $\mathbb{P}_p[J_n]$, if at some point it crosses the 0.98 line (for some p), then actually it's very quickly going to go up to 1.

Guy really likes the geometric ideas from RSW theory of stitching stuff together. This first statement is of this form. At a high level, we're going to leverage the fact that if ε is small, then ε^2 is very very very small. And so what we're going to try to do is: So here's J_{2n}, which is $(2n-2) \times 4n$. We're going to identify two slightly ugly-shaped rectangles that are disjoint, within this bigger J_{2n} rectangle.



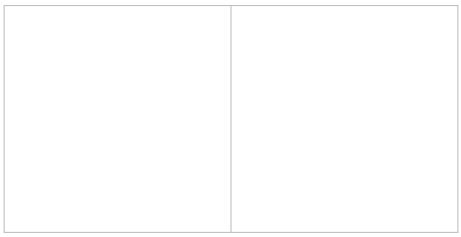
So a crossing event happens in J_{2n} if it happens in either one of these. Now we have much skinnier rectangles, and we want to talk about the crossing probabilities in those.

The first step is:

> **Claim 25.4 —** We have $\mathbb{P}_p[H_{n-2,4n}] \geq 1 - 5\varepsilon$.

Proof. We're talking about the event that we have a crossing of the $(n-2) \times 4n$ rectangle. We split it into $2n$ and $2n$. By definition, with probability $1-\varepsilon$ I have a crossing of the half on the left, and with probability $1-\varepsilon$ I have a crossing on the right.

And then we try to join them up — so we can consider a third horizontal crossing in the middle. And then we consider two vertical crossings. And together, we have a left-to-right crossing of the entire rectangle, using five J_n events (we're again using FKG, so that monotone increasing events are positively correlated; that's what allows us to stitch these events). □



So now we have a crossing of this narrow rectangle. Step 2 is that

$$\mathbb{P}_p[J_{2n}] \geq 1 - (5\varepsilon)^2,$$

since as long as we don't fail in the first skinny rectangle *or* the second, we succeed. And ε was chosen small enough that this is at least $1 - \varepsilon/2$. □

> **Remark 25.5.** The reason for $n-2$ (as opposed to having n) is just so that our two skinny rectangles don't share edges on the boundary.

So this means the probability of each one not happening — they decrease geometrically. What should we do with this? Well, what if we consisder a bunch of them — J_n, J_{2n}, J_{4n}, ...? So we consider the infinite sequence of J_n's; what can we say? Let's consider

$$\mathbb{P}_p[J_{n2^r}^{\mathsf{c}} \text{ occurs infinitely often}].$$

Borel–Cantelli tells us that if $\sum_i \mathbb{P}[E_i] < \infty$, then

$$\mathbb{P}[E_i \text{ occurs infinitely often}] = 0.$$

In other words, with probability 1, only a finite number of the E_i's occur.

But that's exactly the situation we're in — J_n^{c} occurs with probability at most ε. And then the next probability is going to be $\frac{\varepsilon}{2}$, then $\frac{\varepsilon}{4}$, and so on. So we get that only a finite number of the J_{n2^r} fail to occur.

How do we use this to say there's an infinite component? This certainly says there are bigger and bigger components (each J_n itself entails a crossing), but that doesn't imply there's an *infinite* component. It could mean there's finite-sized components of all sizes, but we want to say there's genuinely an infinite component.

Let's start at the n from which all the future J_n's occur. And let's arrange them!

We've concluded there exists an infinite component. But can we conclude from this that $\theta(p)$ — the probability the origin is in an infinite component — has to be positive?

Claim 25.6 — If $\mathbb{P}_p[\text{exists infinite component}] = 1$, then $\theta(p) > 0$.

Proof. Consider the contrapositive, so $\theta(p) = 0$. By translation invariance of the model, the same is true for every other vertex. And then you conclude what you want — if every single vertex has probability 0 of being in an infinite component, then there can't possibly be an infinite component (you can write this formally using a union bound). □

Remark 25.7. Guy has taught this course anumber of times; as the class goes, you get a diversion of the comfort level of certain probabilistic concepts, so some people don't want to see the union bound and some do. He thinks it's valid; every once in a while you do want to see some basic thing done carefully. There's so many layers of abstraction and intuition that you end up skipping lots of little things, but it's useful to periodically write out every little thing.

§25.3 The plan

Now all we need to do is show that there's some n (for $p > \frac{1}{2}$) where this thing J_n has pretty high probability. So how are we going to do that?

Now we get into some slightly heavier machinery. Basically, what we're going to do is as follows.

Suppose that $p_c = \frac{1}{2} + 2\delta$; in particular, this means $\theta(\frac{1}{2} + \delta) = 0$. We're trying to get to some contradiction. So what we're saying is — suppose that $\theta(p)$ is still 0 past $\frac{1}{2}$, for a little bit. We want to argue that this is *not* true — we want it to be the case that this goes up.

Then we can conclude from our proposition that $\mathbb{P}_p[J_n] < 0.98$ for all n; otherwise we'd have the percolation event happening. (This is for all $p \in [\frac{1}{2}, \frac{1}{2} + \delta]$.)

Now let's imagine plotting $\mathbb{P}_p[J_n]$.

What we're going to show is that as a consequence of both $\theta(\frac{1}{2} + \delta)$ being 0 and this crossing probability being not so big,

$$\frac{d}{dp}\mathbb{P}_p[J_n] \to \infty \quad \text{as } n \to \infty.$$

(This is something we will show in a moment.)

The structure of this argument is kind of weird (you can do it more directly, but this is more clean). So what we're saying is, suppose the percolation probability is 0 in the range $\frac{1}{2}$ to $\frac{1}{2} + \delta$. First of all, this is saying that in this range, $\mathbb{P}_p[J_n]$ is not above 0.98. But not only that, you also have a huge derivative — your derivative is going to ∞ as $n \to \infty$. So this thing grows very quickly. And that's impossible — if you integrate the derivative across a constant-sized interval and the derivative is going to ∞, then your integral is going to ∞. But it can't be, because it's a probability!

Remark 25.8. If you formulated this in the forwards direction, you'd say that if it's not so big, then your derivative is huge and you're quickly going to cross 0.98, and then you have percolation.

So we draw two pictures. Last time, we showed that for $p \le \frac{1}{2}$, we have $\theta(p) = 0$ (the probability of the origin being in an infinite component). Now we're saying, suppose it was 0 for a while longer — all the way to $\frac{1}{2} + \delta$. We're going to try to arrive at some contradiction — something terrible is going to happen. And how this goes is, well, the percolation probability is 0. If $\theta(p) = 0$, you can't have an infinite component, which implies that $\mathbb{P}_p[J_n] < 0.98$ (by the contrapositive of Proposition 25.3).

Then, we're going to use both of these together to argue that $\frac{d}{dp}\mathbb{P}_p[J_n] \to \infty$ (as $n \to \infty$), for every $p \in [\frac{1}{2}, \frac{1}{2} + \delta]$. And by calculus, that's contradictory — it's getting steeper and steeper, so it has to go infinitely high, but it's a probability so it's bounded by 1. So that'll be our contradiction — we've derived something nonsensical by assuming our critical probability is strictly bigger than $\frac{1}{2}$.

Remark 25.9. The way to think about this is, let's go slightly above $p = \frac{1}{2}$. If we're already above 0.98, we're good to go. If we're not, then the derivative is huge, so very quickly you do cross 0.98.

Remark 25.10. We're studying percolation and it might seem kind of special, but many of the ideas behind these arguments are much more broadly applicable, across many places. In particular we'll see that talking about derivatives appears in sharp thresholds and many areas of math and CS.

§25.4 Influence

Theorem 25.11

Let A be an increasing event on Ω. Then

$$\frac{d}{dp}\mathbb{P}_p[A] = \sum_{e \in \Omega} \mathrm{Inf}_e^p(A).$$

As far as we're concerned, we really just care about J_n; *increasing* means if I add more edges, the event can't go from occurring to not occurring, and a crossing is certainly an infinite event.

So we're taking a derivative with respect to p. And it has this wonderfully clean expression. Here Ω is some region of interest (just because weird things happen in infinite spaces; we can restrict to a finite domain because J_n is finite). It's given by the sum over all the edges, of this thing called 'the influence of edge e on A.' What is that? It's actually a very natural notion:

Definition 25.12. We define the influence of e on A as

$$\mathrm{Inf}_e^p(A) = \mathbb{P}_p[A(\omega^{+e}) \neq A(\omega^{-e})].$$

So we're looking at the event that A when you *include* edge e does happen, but when you *don't* include e it does not happen. (Perhaps we should have written $\mathbf{1}_A$ instead of A; you can think of events as functions with values 0 and 1.)

So this is saying, the probability that edge e makes a difference — that if e was left out then A doesn't happen, but if e was included then A does happen.

You can also think of it as, you're going to cast your vote in an election, and this is the probability that your vote makes a difference (that if you vote one way one thing happens, and if you vote the other way the other thing happens).

Remark 25.13. The standard terminology for $A(\omega^{+e}) \neq A(\omega^{-e})$ is that $\omega \in \mathrm{Piv}_e(A)$ — that edge e is pivotal for A.

We're going to prove this, but first the reason this is useful is because we wanted to talk about $\frac{d}{dp}\mathbb{P}_p[A]$, the derivative of this really complicated thing; and it's unclear how we could have computed it. But all of a sudden, it's decomposed into this nice thing.

Proof. Imagine we put a separate parameter p_e on every edge. As a reminder of some calculus, if we have a function of many variables

$$f(p_1, \dots, p_N) = \mathbb{P}_{\vec{p}}[A],$$

then if we take the derivative where we set each of these parameters to p, we get

$$\frac{d}{dp} f(p, p, \dots, p) = \sum_e \frac{\partial f}{\partial p_e}(p, \dots, p).$$

This has some name (maybe the chain rule).

We're going to apply this. The point is we just need to compute all these partial derivatives. So we need to compute a partial derivative of the probability of interest; really what we should be looking at is

$$\mathbb{P}_{\vec{p}+\varepsilon\vec{u}_e}[A] - \mathbb{P}_{\vec{p}}[A]$$

(where u_e is the eth standard basis vector).

We have two separate probabilities, one at a parameter slightly bigger than the other; what should we do to compare them? We'll couple them.

First, if I have $X \sim \mathsf{Ber}(p)$ and $Y \sim \mathsf{Ber}(p+\varepsilon)$, how do I couple them so that $X \leq Y$ almost surely? I can't just sample two independent Bernoullis. Really you should sample $U \sim \mathsf{Unif}[0,1]$; think of a number line with p and $p+\varepsilon$, and then we set $X = \mathbf{1}[U \leq p]$ and $Y = \mathbf{1}[U \leq p+\varepsilon]$.

So we're going to do that here. To save a bit of writing, we'll write this as

$$\mathbb{E}_{\vec{p}}[\mathbf{1}_A(\omega_{\sim e}, \omega'_e) - \mathbf{1}_A(\omega_{\sim e}, \omega_e)]$$

(where $\omega_{\sim e}$ indicates ω at all the edges except for e), where I use the above coupling to compute ω'_e and ω_e.

This comes up surprisingly often — to compare two probabilities, you can couple them however you want because of linearity of expectation.

So you do this, and now we can calculate it out. If $\omega_{e'} = \omega_e$, then this is 0. If $\omega_{e'} = 1$ and $\omega_e = 0$, well, then we don't know what happens, but we might get something nonzero. So what we get is

$$\varepsilon \cdot \mathbb{E}_{\vec{p}}[\mathbf{1}_A(\omega^{+e}) - \mathbf{1}_A(\omega^{-e})].$$

And that's the same exact expression we wanted — this is

$$\varepsilon \cdot \mathbb{P}_{\vec{p}}[\omega \in \mathrm{Piv}_e(A)].$$

(In the definintion of influence, we wrote $A(\omega^{+e}) \neq A(\omega^{-e})$; but the only way they can be not equal is if the first is 1 and the second is 0, since A is increasing). □

§25.5 Edge isoperimetry

So we said we wanted to show the derivative goes to ∞, and we just found a nice way to deal with derivatives; so now let's do that.

But for this, we'll need to black-box some stuff. We're going to spend 5 minutes talking about edge isoperimetry on the Boolean hypercube, and then we'll state the theorem (it won't make sense otherwise).

If $A \subseteq \{0,1\}^n$ (you can think of what configurations of edges result in the event of interest, e.g., a crossing), then there's an inequality about $|\partial_e(A)|$ (here e is not a particular edge; $\partial_e(A)$ denotes the entire edge boundary, i.e., edges where one endpoint is inside the set and the other is outside). There's an inequality saying

$$|\partial_e(A)| \geq |A| \cdot \log \frac{2^n}{|A|}.$$

This is very much like the isoperimetric inequality we saw on the sphere, and it happens to be tight for a subcube. So there are certain sets that optimize this, and you can check that having a subcube does.

This is maybe not so surprising. One thing we will remark — we saw a connection between concentration and isoperimetry on the sphere. It turns out you can derive concentration of measure on the cube from this — there's a very general connection between isoperimetry and concentration.

So if you have a Boolean function $f : \{0,1\}^n \to \{0,1\}$, then if you set $A = f^{-1}(1)$ (that's just the set of points on the cube that get mapped to 1) — it turns out that the normalized edge boundary is

$$\frac{|\partial_e(A)|}{2^{n-1}} = \mathrm{Inf}^{1/2}(f) = \sum_{i=1}^n \mathrm{Inf}_i^{1/2}(f)$$

(where Inf denotes the total influence). We care about influences, because they're showing up in the derivative we want; and it turns out the sum of influences is just the edge boundary.

So that's just to convince us that influences have something to do with isoperimetry.

Where are we going with this? Guy will tell us another inequality that's not this one, but is very useful for us (and all over computer science).

But before that, let's interpret this — let $\mu = \frac{|A|}{2^n} = \mathbb{E}f$. Then we get a nice inequality, which is that

$$\mathrm{Inf}^{1/2}(f) \geq \mu \log(1/\mu).$$

You may or may not care about this, but it's interesting, in part because this is a geometric thing (how big are boundaries compared to the stuff inside). But influence is kind of an analytic object — what's the probability that my vote matters? It's always interesting to see that two different things are connected in such a way.

This is the vague stuff; now we'll write down the actual inequality that we care about.

We'll write down two versions of this theorem — one because it's very comparable to Efron–Stein. (This is a coordinate-by-coordinate decomposition of the variance — you look at the variance, and imagine randomizing each coordinate separately, and it's lower-bounded by a certain sum of coordinate-wise variances.) (People don't remember this, so Guy skips it.)

Theorem 25.14 (KKL)
We have $\mathrm{Inf}^p(f) = \sum_{i=1}^n \mathrm{Inf}_i^p(f) \geq c \cdot \mathrm{Var}_p(f) \cdot \log \frac{1}{m_p}$, where $m_p = \max_i \mathrm{Inf}_i^p(f)$.

If f is the voting rule (everyone votes D or R, and there's some complicated rule deciding who wins), and you ask, what's the probability my vote is the deciding vote? So that's what we're looking at, and adding it up over all individuals. And m_p is the maximum individual influence on the election. What this theorem is saying is that the total influence is lower-bounded by the variance of f (that makes sense — if f is just the constant function then no one has any impact whatsoever). So this is natural. But the magical thing is the $\log 1/m_p$ — if nobody has high influence, then we get an extra log factor. And it turns out it matters a lot. (If you removed the log, you'd basically get Efron–Stein; so this is a massive strengthening of that.)

§25.6 Bounding the derivative

Certainly if you do some pattern matching, we have our derivative expression

$$\frac{d}{dp}\mathbb{P}_p[A] = \sum_e \mathrm{Inf}_e^p(A).$$

This is very natural because we're imagining increasing the probability of each edge by a tiny amount; and for the right, that effect should depend on the probability that edge even mattered (if it didn't, increasing its probability shouldn't do anything). So there we had a sum of influences, and here we do too. So this gives a lower bound on the derivative, which is what we wanted.

> **Remark 25.15.** This is a variant of the KKL (Kahn–Kalai–Linial) lemma. If you're interested, you can learn Boolean analysis (Dor Minzer teaches a class on this periodically).

We want to use this, even if we can't prove it (it would take probably 6 lectures to prove it — it's actually quite a deep result and uses something called *hypercontractivity*).

So we're going to apply this to our situation — we get

$$\frac{d}{dp}\mathbb{P}_p[J_n] = \sum_e \mathrm{Inf}_e^p(J_n) \geq c \cdot \mathrm{Var}_p(J_n) \cdot \log \frac{1}{m_p}.$$

(the sum is over the edges in the rectangle of interest for J_n).

What can we say about $\mathrm{Var}(J_n)$? We're trying to get lower bounds, so if the variance is really tiny we'd be out of luck.

Well, last lecture we saw by RSW that J_n is bounded away from 0; and we assumd it's bounded away from 1. So its probability is bounded between c' and $1-c'$, which means its variance is at least $c'(1-c')$.

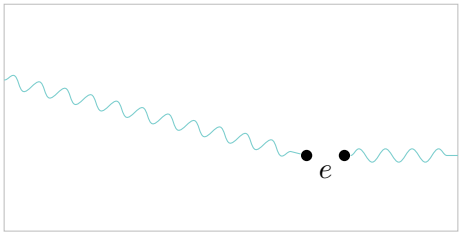
So that's just some constant, and we don't have to worry about it; all we have to do now is show that $m_p \to 0$, so that $\log \frac{1}{m_p} \to \infty$.

Suppose that $\theta(\frac{1}{2}+\delta) = 0$ (this was our assumption in the thing we're trying to show). We want to show this implies $m_p \to 0$ — i.e., that every edge has influence going to 0 (because this is the maximum).

So why is it that every edge has influence going to 0 on the event that we have this crossing? Well, let's unpack what's the influence — this is

$$\mathbb{P}[J_n(\omega^{-e}) = 0 \text{ and } J_n(\omega^{+e}) = 1]$$

(so without edge e there is no crossing, and with edge e there is a crossing). So let's draw a picture.



What do the other edges have to look like so this happens — so if you include e there is a crossing, and if you remove e then there isn't?

That means every possible crossing has to go through edge e. If you had some crossing above, then this edge becomes irrelevant — J_n would happen no matter what. On the other hand, if you had a cut elsewhere, then the existence of e would not suddenly cause a crossing to happen. So this event that e is pivotal means there has to be a crossing from one side to an endpoint of e, and from the other endpoint of e to the other side (there could be multiple). Certainly in this configuration as we drew it, e is pivotal (it decides the outcome); but you can convince yourself the converse is true too (if e decides the outcome, things have to look like this).

Now what are we going to do? Well, e is either on the left or right half of the thing; let's say it's on the right. And then we'll look at the left half. And, well, we have a left-to-right crossing of the left half.

So we've shown that

$$\mathbb{P}[J_n(\omega^{-e}) = 0 \text{ and } J_n(\omega^{+e}) = 1] \le \mathbb{P}_p[H_\square(n)]$$

(the right-hand side is the crossing probability of a $n \times n$ square). What do we do now?

Actually the original argument Guy had was kind of sketchy, but he has a fix. Instead of just the line being in the middle, we'll consider the endpoint of e on the left, and we'll call this the origin. And then we'll say, well, this particular point (the left endpoint of e) has to go all the way to the boundary. In particular, the event it went to the boundary is evidence that 0 connects out to the boundary of $[-n, n]^2$. (We've identified the origin with the left endpoint here; it's travelled at least n to the left, and it certainly hasn't gone more than n upwards or downwards, so we get a path in the longer direction, which means somehow there is a path from the origin tothe boundary of this box.)

And what is the probability that 0 connects to the boundary? This is the very last piece of the argument. Well,

$$\mathbb{P}_p[0 \leftrightarrow \text{boundary of } [-n, n]^2] \to \mathbb{P}[0 \leftrightarrow \infty]$$

as $n \to \infty$ (as n grows, this event shrinks — if you reach the boundary of $[-10^6, 10^6]$, then you're certainly reaching the boundary of a small box). (With monotone events we can move the limit inside.) And this is 0 — that's where we finally use the fact that $\theta(p) = 0$.

So what just happened? All we said was that if $\theta(p) = 0$, then it's actualy unlikely to have this situation that you have edge e included in paths; and so the influence of every edge is going to 0. Then we go to our isoperimetric type inequality and conclude that the *total* influence is growing. And the total influence is the derivative, so we've shown that's going to ∞!

Student Question. *You lower-bounded the variance by c' in the fixed-aspect ratio case; but if this were applicable, wouldn't that contradict the $\varepsilon \to \varepsilon/2$ thing?*

Answer. Yeah we cheated here, RSW is only applicable for $p = \frac{1}{2}$. Now we're assuming p is bigger. So we should use the fact that it's less than 0.98 for the variance bound. We do still use RSW to get the lower bound — we can use monotonicity to say, well, if $p > \frac{1}{2}$ then the probability is at least as big as the RSW one (which is some constant), and it's upper-bounded by 0.98.

§26 May 12, 2025 — Local sampling algorithms

As a friendly reminder, your final project is due tonight; if you need more time or still have late days, you can turn it in as late as Thursday, but try not to turn it in later than that.

We've already covered most of the main ideas in the course, so today Kuikui will talk about something kind of cute, the topic of local sampling algorithms.

§26.1 Local sampling

We're trying to sample a configuration according to a graphical model or a Markov random field (e.g., a random independent set or random graph coloring). But now I'm in a setting where the input graph is absolutely gargantuan — so large you can't even fit it in main memory (or maybe even infinite). But you still want to draw samples for e.g. colorings of some subset of vertices in the graph.

Now, because the input is so large you can't fit the whole thing in main memory, we need to be a bit careful about what we mean by the input and output.

For simplicity, in this lecture we'll just work with random independent sets in the hardcore model.

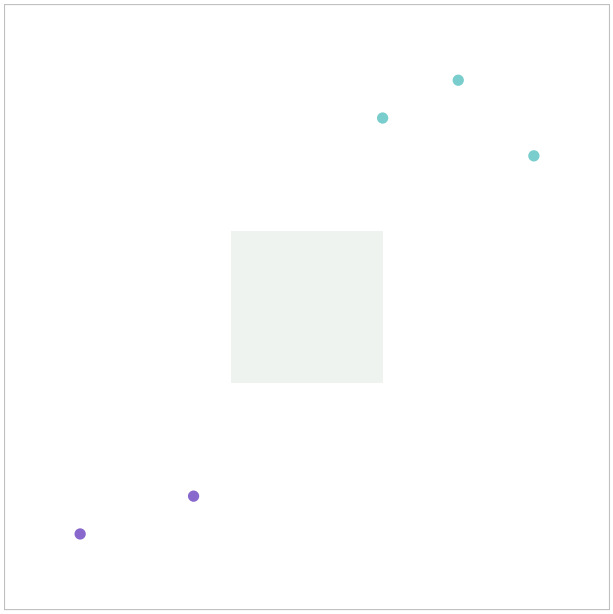
Problem 26.1 (Local sampling)

- **Input:**
  - A massive graph $G = (V, E)$, which is specified as follows: We're given the source code for a deterministic function $\mathsf{Neighbors}$ which, given some vertex v, outputs all its neighbors.
  - A set $S \subseteq V$.
  - A boundary condition $\tau : \Lambda \to \{\mathsf{in}, \mathsf{out}\}$, where $\Lambda \subseteq V \setminus S$.
- **Output:** A sample $\sigma \sim \mu_S^\tau$, where $\mu(I) \propto \lambda^{|I|}$ for all independent sets I.

So you're given the source code for how to generate the neighborhood of a given vertex. Maybe a representative example to keep in mind is $G = \mathbb{Z}^d$ — if I tell you the coordinates of some vertex, you can tell me the coordinates of all its neighbors (just increment or decrement the coordinates by 1).

We're also given a subset of vertices S, and a 'boundary condition' τ that pins an assignment of all the vertices in Λ. And we're supposed to output a random assignment σ on S, drawn from the Gibbs distribution where we condition on τ (and we only look at the vertices in S).

As an example, imagine I have a very big box, e.g., $\mathbb{Z}^2$. And maybe I only care about the vertices in a little box around the origin (so this is S). And maybe I condition on some vertices on the outside to be out (in blue), and others to be in (in blue). And I want to draw a sample for just the vertices in this small box.



In an even more extreme case, we can imagine I have my infinite graph, I don't condition on anything, and we just want a sample for the origin.

The key is we want our algorithm to have runtime only depending on $|S|$ and $|\Lambda|$, not the whole graph — the whole graph is so big you can't fit it in main memory. For example, if $\Lambda = \emptyset$ and S is a single vertex, we want our algorithms to run in *constant* time. This seems almost impossible, but we'll be able to do this.

But because our algorithm is supposed to run in constant time, it's definitely not going to see the whole graph — it's going to see a vanishingly small fraction.

> **Goal 26.2.** Get an algorithm whose runtime only depends on $|S|$ and $|\Lambda|$. (We think of these as constants — the representative case is $|S| = 1$.)

If our graph is infinite (e.g., $\mathbb{Z}^2$), then this might not be well-defined — the notion of 'the Gibbs distribution' might not make sense. It could be for some values of λ that there's no unique Gibbs distribution on $\mathbb{Z}^2$; so it could be I need to even specify which one I'm looking at.

Also, we're shooting for algorithms that only look at a small portion of the graph, so it makes sense to assume μ satisfies good correlation decay.

So we're going to assume that μ is well-defined and satisfies *strong spatial mixing*. (If the graph is infinite, this will imply there is a unique Gibbs distribution, so this all makes sense. If you're not comfortable with infinite graphs, just think of it as very large but finite; but we'll still need to assume something like strong spatial mixing.)

We also assume throughout that our graph has bounded maximum degree.

§26.2 Reduction to a single vertex

Let's first try to simplify our task — let's reduce to the case where $|S| = 1$ (so we just have a single vertex).

What we want to say is that if I have an algorithm that solves this when $|S| = 1$, I can easily turn this into an algorithm which works for general S.

The way we do this is to inductively sample one vertex at a time — suppose $S = \{v_1, \dots, v_\ell\}$. Then I can use the single-vertex algorithm to sample

$$\sigma(v_1) \sim \mu_{v_1}^{\tau}$$

(the assignment for the first vertex). Then conditioned on what I've already sampled, I sample the next one — so I sample

$$\sigma(v_2) \sim \mu_{v_2}^{\tau, v_1 \leftarrow \sigma(v_1)}.$$

And I continue doing this, up to

$$\sigma(v_\ell) \sim \mu_{v_\ell}^{\tau, v_1 \leftarrow \sigma(v_1), \dots, v_{\ell-1} \leftarrow \sigma(v_{\ell-1})}.$$

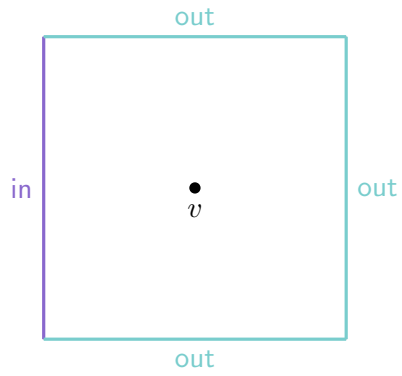
In particular, I just need to solve the sampling problem when I'm looking at just a single vertex (but I might have some arbitrary boundary condition in τ).

§26.3 Approximate sampling

As a warmup, let's first consider the setting where we only want to draw *approximate* samples — to sample from a distribution which is close in total variatio distance.

Here, we claim our assumption of strong spatial mixing already gives everything we need to solve this problem. The picture is: My graph is very large, but maybe let's focus on just the vertices that are at distance at most L from our vertex v (where L is some parameter we'll choose in a moment; think of it as some very large constant, which will maybe depend on how good an accuracy we want and the rate at which we have exponential decay of correlations, and the max-degree).

I want to draw a random sample for v according to its marginal distribution. Now, if I have something like correlation decay, I can look at an alternative picture. Maybe the above picture is the true distribution that we want. And we're going to kind of approximate the marginal distribution at v by a different picture, where we arbitrarily pin the vertices on the boundary of this box to some assignments.



So I'm still looking at the marginal for v, but now I'm conditioning on the assignments for all the vertices exactly L away.

Now, what have we done? Well, L is some large constant, and my maximum degree is also some constant, so the size of this ball is some large constant. And because I've pinned all the vertices at distance L away, the distribution at v will be independent of everything outside this ball.

Student Question. *How do you know your assignment is consistent?*

Answer. For the hardcore model, let's just say I pin everything to be out. For some models like Ising, everything can be extended so this isn't an issue; but here let's just pin everyone to be out, so that this is always extendible.

Once I've fixed the distribution at the boundary, the distribution of v is going to be independent of everyone outside the ball. So now I can brute-force, because the size of this ball is constant.

And correlation decay (more precisely, SSM) lets us say these pictures are very close in total variation distance — if we call this boundary σ, then

$$\|\mu_v^{\sigma,\tau} - \mu_v^{\tau}\|_{\mathsf{TV}} \lesssim (1-\delta)^L.$$

So the basic idea is once you have correlation decay, for approximate sampling I can just truncate the graph to some large radius, and once I've truncated I can do brute force. And once I've brute force enumerated everything, I can just draw a random sample.

Algorithm 26.3

(1) Truncate by pinning $\sigma(u) = \mathsf{out}$ for all u such that $\mathrm{dist}_G(u,v) = L$.

(2) Brute-force enumerate all configurations on the radius-L ball $\mathbb{B}_L(v)$ which are consistent with both σ and τ.

(3) Sample from $\mu_v^{\tau,\sigma}$.

If we have exponential decay, we want to set

$$L \asymp \frac{1}{\delta} \log \frac{1}{\varepsilon}.$$

This yields an algorithm running in very large, but still constant, time. There'll be $2^\bullet$ where $\bullet$ is the size of the ball. If we have maximum degree Δ, the size of the ball will be Δ^L; so we'll have runtime something like

$$2^{\Delta^{(1/\delta)\log(1/\varepsilon)}}.$$

This is absolutely nuts, but it's still constant, because everything is a constant.

This is maybe not very satisfactory. There are ways to turn this into — to make this exponentially faster ($\Delta^{(1/\delta)\log(1/\varepsilon)}$, without the 2). This requires additional machinery. At a high level, instead of doing brute force, you convert your problem into a problem on a tree, and on the tree you can use 'belief propagation.' For this to work, you need something slightly stronger than SSM on the original graph; you need SSM on this tree.

But the point is you get some insane but constant runtime (and there are ways to make this better).

Remark 26.4. This was just an algorithm for sampling an assignment for a single vertex. Now suppose we push it through the previous reduction. Then the errors can compound — the TV distance will no longer be ε, but $\varepsilon\ell$ (where ℓ is the number of vertices — you can argue this using the coupling lemma). In particular, if you use this to sample for a subset of vertices S, your runtime will be

$$2^{\Delta^{(1/\delta)\log|S|/\varepsilon}}.$$

So the point is approximate sampling somehow needs you to be very careful that your TV error doesn't compound as you increase the number of vertices you're trying to sample.

§26.4 Perfect sampling

Now we'll turn to a slightly different setting, where I really do want perfect samples from the distribution, at the cost of making the *runtime* random. In CS this is called a *Las Vegas* algorithm (which only has polynomial *expected* runtime, but is required to output the right answer when it does terminate). We'll talk about a very nice algorithm due to Anand and Jerrum.

Assume strong spatial mixing (SSM).

Goal 26.5. Do perfect sampling in $O(1)$ expected runtime.

Again, for simplicity we'll work with independent sets (there's a more general algorithm for general spin systems or Markov random fields).

And for simplicity, let's assume something even stronger than SSM — let's assume for now (just to illustrate the algorithm) that

$$\lambda < \frac{1}{\Delta - 1}.$$

This is the same condition as Dobrushin's condition for rapid mixing of Glauber dynamics (and also implies SSM).

§26.4.1 A recursive approach

The whole difficulty behind the sampling problem is that we do not know what the actual marginal probabilities are. If we *did* know

$$\mathbb{P}[v \leftarrow \mathsf{out} \mid \tau],$$

then we would be totally good — we could of course solve this in constant time. Of course, we don't know this probability; in general, computing this is hard. But we *do* know it in a couple of special cases — it's easy if τ happens to pin all the vertices in the immediate neighborhood of v. If it pins all the vertices in the neighborhood to be out, then

$$\mathbb{P}[v = \mathsf{out}] = \frac{1}{1+\lambda}$$

(this was on the homework); and if it pins any of them to be in, then this is 1 (you can't have v in the independent set).

So this is hard in general, but it's easy if I happen to know the assignments for the vertices in the immediate neighborhood of v. This same idea is what we used in the truncation argument — if we already knew the vertices in some large ball around v, then we could brute force inside.

Of course, we don't actually know the assignments for the neighborhood of v in general. But we can try to recurse — to call our algorithm for each of those neighbors.

And the trick is to use $\lambda < \frac{1}{\Delta - 1}$ to make sure that the recursion doesn't grow unboundedly.

§26.4.2 A first attempt

Here's a first attempt using recursion:

Algorithm 26.6

(1) Let $u_1, \ldots, u_d$ be the neighbors of v.

(2) For $i = 1, \ldots, d$, we recursively call the algorithm to sample an assignment $\sigma(v_i)$, conditioning on everything we've sampled before (so τ, $\sigma(v_1)$, $\ldots$, $\sigma(v_{i-1})$).

(3) If for all i we have $\sigma(v_i) = \mathsf{out}$, then output $v \leftarrow \mathsf{out}$ with probability $\frac{1}{1+\lambda}$ (and in with probability $\frac{\lambda}{1+\lambda}$).

This algorithm will definitely not work. For one thing, it will never terminate — there are recursive calls which bounce between v and a neighbor and back to v and that neighbor and so on.

So this is no good, because of infinite recursion.

§26.4.3 Stopping the recursion

So we need a way to stop the recursion, somehow. A very cute idea to get around this is to use the following lemma, which we also proved on the homework.

Lemma 26.7

For every $\tau : \Lambda \to \{\mathsf{in}, \mathsf{out}\}$ with $\Lambda \subseteq V \setminus \{v\}$, we have

$$\mathbb{P}[v \leftarrow \mathsf{out} \mid \tau] \geq \frac{1}{1+\lambda}.$$

This was on the homework; it's from the fact that no matter how you pin vertices in the immediate neighborhood, the resulting probability is always at least this much (the worst case is when all of them are pinned to be out).

How can we use this? Imagine the following picture, where we have a bar representing the interval $[0,1]$. There's some unknown probability, which is

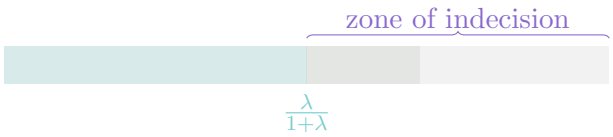
$$\mathbb{P}[v \leftarrow \mathsf{out} \mid \tau].$$

If we knew what this probability actually is, then we could sample a uniformly random number in $[0,1]$; and if it's below this threshold we'd set v to be out, and otherwise we'd set it to be in.

We can always $U \sim \mathsf{Unif}[0,1]$ and test how U compares to some threshold. And what this lemma tells us is that there's some number $\frac{1}{1+\lambda}$ which is always a lower bound on the true probability.

So if U lands below this number $\frac{1}{1+\lambda}$, then I can *immediately* output that v gets out — I no longer have to run the recursion. So we sample $U \sim \mathsf{Unif}[0,1]$. If $U \leq \frac{1}{1+\lambda}$, then we can safely output $v \leftarrow \mathsf{out}$ without having to look at the neighbors.

So there's some positive probability that we straight-up terminate the recursion already. It's only if I get a bit unlucky and my random variable lands in the zone between $\frac{1}{1+\lambda}$ and 1 — what the algorithm calls the *zone of indecision* — that I actually need to sample assignments for my neighboring vertices, and use those to sample v. (There's some residual probability in that zone that v gets out and some that v gets in, but we don't know it.)



So the algorithm we'll do just modifies the above one, with an additional if-else check. And we'll show that algorithm is not only correct, but also terminates in expected constant time.

Algorithm 26.8 (Anand–Jerrum sampler (2022))

(1) Sample $U \sim \mathsf{Unif}[0,1]$.

(2) If $U \leq \frac{1}{1+\lambda}$, stop and output $v \leftarrow \mathsf{out}$.

(3) Otherwise:

- Order the neighbors of v as $u_1, \ldots, u_d$.
- Recursively call the sampler to sample $\sigma(u_i) \sim \mu_{u_i} \mid \tau, \sigma(u_1), \ldots, \sigma(u_{i-1})$.
- Compute $p = \mathbb{P}[v \leftarrow \mathsf{out} \mid \tau, \sigma]$.
- If $\frac{1}{1+\lambda} < U \leq p$, output $v \leftarrow \mathsf{out}$; otherwise output $v \leftarrow \mathsf{in}$.

So this is really a simple modification to an otherwise infinitely recursing algorithm — add a check for the case where we can already decide v should be pinned out, without looking at its neighbors.

Remark 26.9. One thing that may be slightly confusing: Suppose you get to the recursive case, and you recursively call the algorithm for u_i. Maybe you get unlucky there, so you also have to recurse on its neighbors, which include this v; and there you managed to sample an assignment for that vertex. This is *not* the same as the assignment you will eventually output here.

So it could be in the recursive calls to this algorithm, at some point in the recursion you had to sample an assignment for the original vertex v. But that assignment is not the one you'll output eventually — that assignment was drawn conditioned on a bunch of crazy boundary conditions you introduced during the process.

The key insight in the algorithm is to do some kind of lazy computation — delay having to look at the neighbors as much as possible.

§26.4.4 Runtime

Let's see that this algorithm is actually correct. Actually, let's first see that it has constant expected runtime.

Claim 26.10 — If $\lambda < \frac{1}{\Delta}$, then:

- The algorithm terminates with probability 1.
- The expected runtime is $O_{\Delta,\lambda}(1)$.

Proof. Basically, what we have to show is that the recursion isn't unbounded. You can imagine the recursive calls to the algorithm as growing some kind of random tree, where the root is the original call to the algorithm (I give you v and τ), its recursive calls spawn d children, each of those spawns its own recursive calls, and so on.

We'll compare that with a simpler-to-analyze tree, which is a Galton–Watson tree. Let ξ_d be the distribution with

$$\xi_d(0) = \frac{1}{1+\lambda} \quad \text{and} \quad \xi_d(d) = \frac{\lambda}{1+\lambda}.$$

When I'm at a certian recursive call, if I get unlucky and need to recurse on my neighbors, then the number of recursive calls I spawn follows this (I have a $\frac{\lambda}{1+\lambda}$ probability of spawning d recursive calls, corresponding to my d neighbors).

So the number of recursive calls spawned from a given call at some vertex v is distributed according to ξ_d, where d is the number of unpinned neighbors in this current call (which is always at most Δ).

The second observation is that we can compare all these distributions in the stochastic domination sense — we always have $\xi_d \preceq \xi_\Delta$ (meaning there's a coupling such that with probability 1, if I sample from this coupling, the first random variable is always at most the second).

So now we just need to say that if I look at the number of recursive calls at a given depth, that thing is going to be stochastically dominated by the corresponding Galton–Watson branching process with this offspring distribution. And we'll say that branching process is subcritical; so it must die off with probability 1, it has bounded expectation, and so on.

So the above claims imply that if

$$Y_\ell = \#\text{depth-}\ell \text{ recursive calls to the algorithm,}$$

then if I look at the stochastic process $\{Y_\ell\}_\ell$, it's stochastically dominated by the corresponding Galton–Watson branching process $\{Z_\ell\}_\ell$ with offspring distribution ξ_Δ.

And if I have this, we're basically done. The stochastic process on the left is only smaller than the one on the right. And if $\lambda < \frac{1}{\Delta}$, then $\{Z_\ell\}$ is subcritical. □

So we couple the stack-trace of this algorithm with a subcritical branching process, which means the thing is bounded in expectation and terminates with probability 1.

§26.4.5 Fake argument of correctness

Now we need to argue correctness. Actually Kuikui will handwave an argument that's very much not rigorous, and then say in words what the right thing to do is. But he'll give us the intuition for why it's correct.

Here's a definitely not correct proof: I have a recursive algorithm, so I can try to do induction. This is already a nonsense statement, because this algorithm can go to ∞, so there's no base case. But we're going to do it anyways.

Suppose you could actually do the induction. What would the induction say? I want to say that if I recurse, then I did perfectly sample from the right distribution. So what is

$$\mathbb{P}_{\mathsf{Alg}}[v \leftarrow \mathsf{out} \mid \tau]?$$

Well, there's a $\frac{1}{1+\lambda}$ probability I don't recurse; and there's some probability that I do have to recurse. And assume by 'induction' that we did get a real sample for the boundary. Then I'm really taking an expectation over a random sample for the boundary of the following quantity; so

$$\mathbb{P}_{\mathsf{Alg}}[v \leftarrow \mathsf{out} \mid \tau] = \frac{1}{1+\lambda} + \frac{\lambda}{1+\lambda}\mathbb{E}_{\sigma\sim\mu^{\tau}_{N(v)}}\left[\frac{\mathbb{P}_{\mathsf{True}}[v \leftarrow \mathsf{out} \mid \tau, \sigma] - \frac{1}{1+\lambda}}{\frac{\lambda}{1+\lambda}}\right].$$

(The reason we can write $\mathbb{E}_{\sigma\sim\mu_{N(v)}}$ is we're saying 'by induction' σ was drawn from the correct distribution.)

Now we can cancel everything, and we get

$$\mathbb{E}_{\sigma\sim\mu^{\tau}_{N(v)}}[\mathbb{P}_{\mathsf{True}}[v \leftarrow \mathsf{out} \mid \sigma, \tau]],$$

which by the law of total expectation is really the true probability you wanted to sample.

The key point of this calculation was to convince us that at least all these things were done the correct way — $\frac{1}{1+\lambda} < U \leq p$ is the right way to set the probabilities.

§26.4.6 The actual argument of correctness

So what's the actual argument? The way you actually argue this is, we define for the sake of analysis, a separate version of this algorithm where the moment you hit depth h of the recursion, you say OK I'm not going to continue recursing, I'll just brute-force compute everything. Then using actual induction, you can prove that algorithm is correct — the base case where you hit depth h, you've brute-force computed everything correctly. This algorithm is definitely going to have insane runtime, but it's definitely correct, because the base case is correct, and then by this induction the whole algorithm is going to be correct.

Now we want to say this other algorithm, where I enforced boundedness, outputs a distribution which is sufficiently close in TV distance to the original distribution we wanted to analyze (and we want it to decay as the depth h goes to ∞). And that also is not difficult to show.

But the key way to formalize the inductive argument is to first define an alternative algorithm where you forcibly stop at h, and then you couple that thing to the original algorithm you wanted to analyze to begin with.

§26.5 Conclusion

It is the last lecture, so Kuikui will briefly summarize all the things we learned. Hopefully we remember some of them, and hopefully it'll be useful to us at some point; or if not, hopefully we feel at least like we learned something beautiful or interesting.

In the beginning of the course, we promised three things. We said we'd see a lot of models; we saw *percolation* (in particular we focused a lot on Erdős–Rényi), *martingales* (e.g., Gambler's ruin), *Markov chains*, and more generally *graphical models* (things like random independent sets, Gibbs distributions, Ising models, and so on). We progressively got richer and richer models with more dependences, and this enriched our tools for modelling things in the real world.

And there's many problems we studied on them, or questions. We saw phase transitions of all kinds — we started off with phase transitions in connectivity in Erdős–Rényi, but there are also others in the ability to differentiate between random samples of two distributions, the typical structure of a sample from a Gibbs distribution, and so on. We also saw average-case complexity of optimization problems — for instance, finding a maximum clique in a graph (this is canonically one of those NP-hard problems, even to approximate, but you can do reasonably well with a greedy algorithm if your input graph is random). We also saw problems of sampling from a distribution. We also looked at statistical inference — one lecture on the broadcast process, and another on contiguity (which lets you transfer results about a simpler distribution to a more complex one, like for random d-regular graphs).

And finally, we saw many different techniques. We saw the probabilistic method, for constructing otherwise difficult-to-construct objects. We saw the 1st and 2nd moment methods, and also the Lovász local lemma. We saw strong concentration inequalities, which rely on independence or more generally bounded dependence, and so on — for instance, sub-Gaussian random variables. We saw coupling-based methods — for Markov chains, for comparing two distributions (e.g., stochastic domination), and so on.

We also saw algorithmic methods — for instance, in the most recent homework, we used a Markov chain to prove something about the original distribution you cared about.

There's certainly not enough time in a single course to go into all of these in greater depth, and there are many other interesting phenomena and useful tools. But hopefully this gives you some idea of the tools and methods and questions that arise in this area.