Thresholds, sharp thresholds, 2nd moment methods, and planted models

Talk by Will Perkins

Notes by Sanjana Das

April 12, 2024

§1 Introduction

Today we'll talk about a couple of different things. First, Will wrote a survey about thresholds and sharp thresholds; so he'll give his perspective on this big concept and talk about a couple of open problems on a general level. Then he'll talk about random graph thresholds for *particular* properties, such as colorability. He'll give us some intuition for what we expect to happen, and some methods we use to study these properties (in particular, moment methods); and he'll show us some plots used to get some feeling for these problems. And he'll talk about where things work or don't work, and what you might try to do when they don't work.

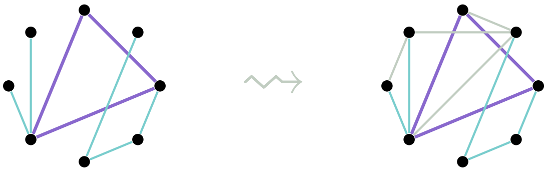
§1.1 Thresholds

We're going to be working with the Erdős–Rényi random graph $\mathcal{G}(n, p)$, and we're interested in *thresholds*, specifically for monotone properties.

Definition 1.1. A graph property $\mathcal{A}$ is monotone if it continues to hold when you add more edges.

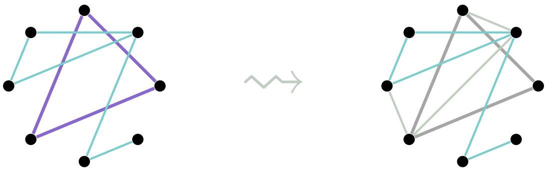
Example 1.2

The properties of being connected or containing a triangle are monotone (if a graph has a triangle and you add more edges, it still has a triangle).



Example 1.3

The properties of having an even number of edges or an isolated triangle are *not* monotone (if you have an isolated triangle and you add edges to make it no longer isolated, you could destroy the property).



Throughout this talk, we'll focus only on monotone properties.

Definition 1.4. We say p^* is a threshold for a property $\mathcal{A}$ if:

- For $p \gg p^*$, we have $\mu_p(\mathcal{A}) \to 1$.
- For $p \ll p^*$, we have $\mu_p(\mathcal{A}) \to 0$.

We use $\mu_p(\mathcal{A})$ to denote the probability that $\mathcal{A}$ holds in $\mathcal{G}(n,p)$. So up to constants, the threshold p^* is the point at which the property $\mathcal{A}$ 'starts to hold' in $\mathcal{G}(n,p)$. (We think of p^* as depending on n.)

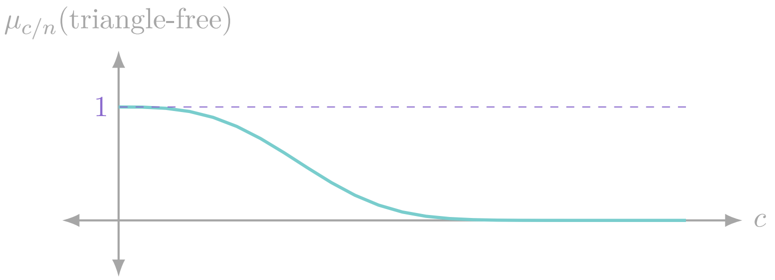
Example 1.5

If $\mathcal{A}$ is the property of containing a triangle, then $p^* = 1/n$ (up to constants).

In fact, Erdős and Rényi (in their first paper on $\mathcal{G}(n,p)$) showed that if $p = c/n$ for a constant c, then letting X be the (appropriately normalized) number of triangles in $\mathcal{G}(n,p)$, as $n \to \infty$ we have $X \to \textsc{Poisson}(c^3/6)$. This actually tells you something stronger than just the threshold — it says that

$$\mu_p(\text{triangle-free}) \to e^{-c^3/6}$$

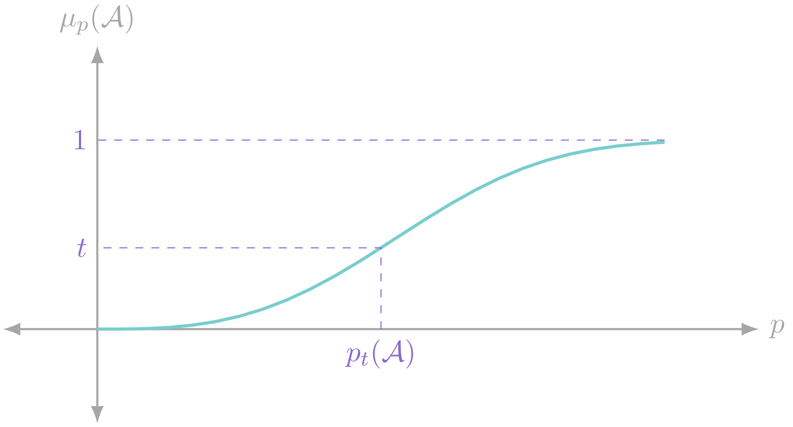
as $n \to \infty$. So we can imagine plotting the probability that $\mathcal{G}(n,p)$ is triangle-free as a function of c (where $p = c/n$), and we'll get some nice function that goes down to 0 as c grows.



§1.2 Scaling windows

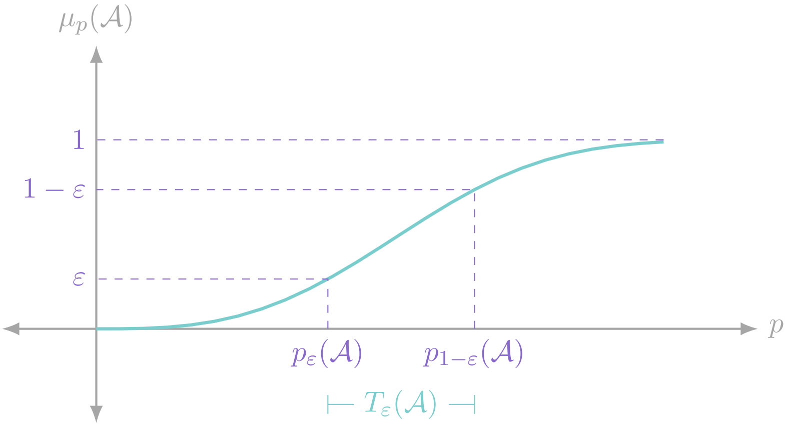
The concept we'll focus more on is that of *scaling windows*.

Definition 1.6. We use $p_t(\mathcal{A})$ to denote the value of p for which $\mu_p(\mathcal{A}) = t$.



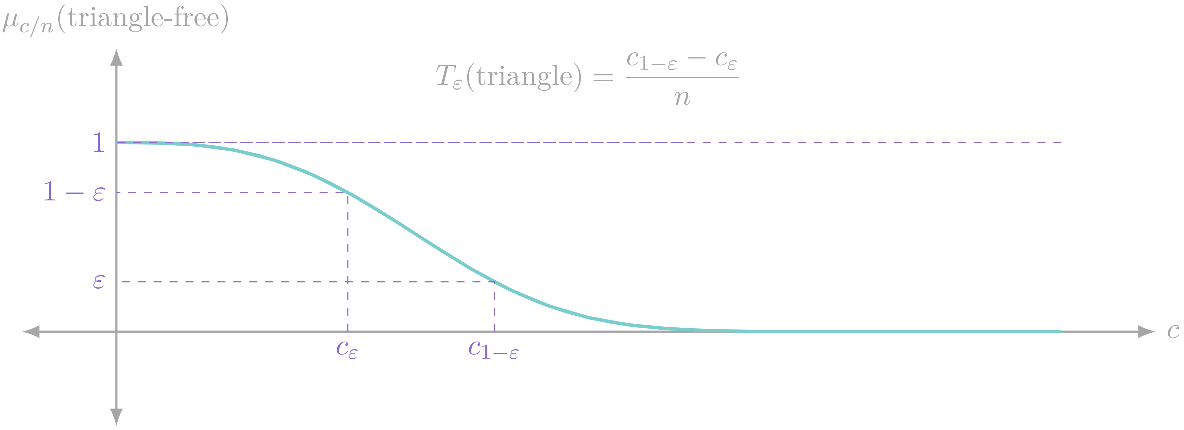
Definition 1.7. The scaling window for a property $\mathcal{A}$ is defined as

$$T_\varepsilon(\mathcal{A}) = p_{1-\varepsilon}(\mathcal{A}) - p_\varepsilon(\mathcal{A}).$$



Example 1.8

If we take $\mathcal{A}$ to be the property of containing a triangle, then both $p_\varepsilon(\mathcal{A})$ and $p_{1-\varepsilon}(\mathcal{A})$ are on the order of $1/n$ — we consider the points with y-values $1-\varepsilon$ and ε (respectively) in the earlier graph for $\mu_{c/n}$(triangle-free), which have x-values differing by a constant (and the x-axis plots c, while $p = c/n$).



We'll usually only care about the *asymptotics* for $T_\varepsilon(\mathcal{A})$.

Remark 1.9. It's not obvious that the scaling windows for different values of ε are on the same order of magnitude, and this may be false. You might be able to get a counterexample by taking some sort of maximum of different properties — e.g., taking $\mathcal{A}$ to be the property that either $\mathcal{A}_1$ or $\mathcal{A}_2$ occurs, where $\mathcal{A}_1$ and $\mathcal{A}_2$ are two very different properties. But for most 'natural' properties, the different scaling windows really should be on the same order.

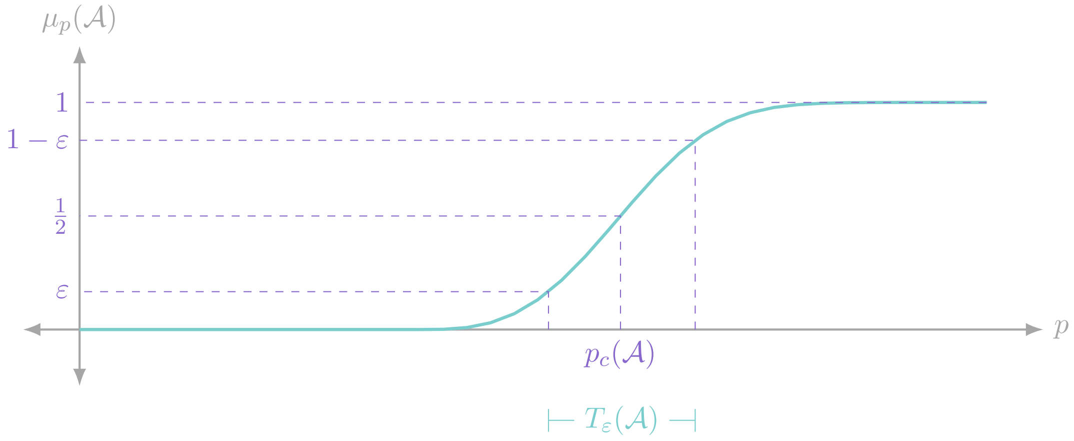
§1.3 Sharp thresholds

Note that for the property of containing a triangle, the threshold and scaling window are *both* $\Theta(1/n)$. On the other hand, we think of a threshold as *sharp* if this is *not* the case — meaning that the scaling window is much smaller than the threshold.

Definition 1.10. We say a monotone property $\mathcal{A}$ has a sharp threshold at p^* if for all $\varepsilon > 0$:

- If $p \geq (1+\varepsilon)p^*$, then $\mu_p(\mathcal{A}) \to 1$.
- If $p \leq (1-\varepsilon)p^*$, then $\mu_p(\mathcal{A}) \to 0$.

This essentially means that the transition of $\mu_p(\mathcal{A})$ from 0 to 1 happens much faster. Another way of phrasing this definition is that we can define the *critical probability* $p_c(\mathcal{A})$ as the value of p for which $\mu_p(\mathcal{A}) = 1/2$; then saying $\mathcal{A}$ has a sharp threshold is equivalent to saying that $T_\varepsilon(\mathcal{A}) = o(p_c(\mathcal{A}))$ for all ε.



In particular, the property of having a triangle does *not* have a sharp threshold. On the other hand, here is a property that *does* have a sharp threshold.

Example 1.11

Let $\mathcal{A}$ be the property of being connected. Then $\mathcal{A}$ has a sharp threshold at $(\log n)/n$.

In order to prove this, let's first review how you prove the threshold result for triangles (i.e., the statement in Example 1.5). If we let X be the number of triangles, then the point is that we can explicitly compute $\mathbb{E}[X] = \binom{n}{3}p^3 \sim n^3p^3/6$, and $p \asymp 1/n$ is exactly where this quantity transitions from going to 0 (as $n \to \infty$) to going to ∞. And meanwhile, we can compute the variance of X as well — we get that $\text{Var}(X) \ll \mathbb{E}[X]^2$, so when $\mathbb{E}[X]$ is large, X itself is large with high probability.

And we can actually do something similar here (for connectivity). It turns out that the right thing to look at is whether there are *isolated vertices* — if the graph is connected then of course it doesn't have isolated vertices, and it turns out that if it doesn't have isolated vertices, then it's connected with high probability (this is not obvious).

So then we can do a similar moment computation for the *number* of isolated vertices, which we call Y — first, its expectation is

$$\mathbb{E}[Y] = n(1-p)^{n-1}$$

(since there's n possible vertices, and each is isolated if and only if the other $n-1$ vertices aren't connected to it, which occurs with probability $(1-p)^{n-1}$). And to find the threshold, we basically solve this equation for $\mathbb{E}[Y] = 1$ (since below this value of p it'll go to 0, and above it'll go to ∞), and we find that this occurs at $p = (\log n)/n$. And more precisely, if $p = (\log n + c)/n$ (for some constant c), then $\mathbb{E}[Y] \sim e^{-c}$; and now as $c \to \infty$ we have $\mathbb{E}[Y] \to 0$, while if $c \to -\infty$ we have $\mathbb{E}[Y] \to \infty$. (We're thinking of first choosing c and only then taking $n \to \infty$.) And we can show that Y is actually (approximately) Poisson, which means

$$\mathbb{P}[Y = 0] \sim e^{-e^{-c}}.$$

And this tells us the scaling window — we get

$$T_\varepsilon(\text{connected}) = \Theta\left(\frac{1}{n}\right)$$

(since we make this probability ε or $1 - \varepsilon$ by moving around the constant c appropriately, and p involves the term c/n). Meanwhile, the critical probability is

$$p_c(\text{connected}) = (1 + o(1))\frac{\log n}{n}.$$

So in particular, connectivity has a sharp threshold — but we actually know more, because we can say the exact order of the scaling window (not just that it's $o(p_c)$).

§1.4 Our main questions

Imagine you have a monotone property $\mathcal{A}$ that you care about; what questions regarding thresholds could you ask? There are four main ones we'll focus on.

Question 1.12. What is the asymptotic order (i.e., Θ-notation) of $p_c(\mathcal{A})$?

Question 1.13. Is the threshold for $\mathcal{A}$ sharp?

Question 1.14. If the threshold for $\mathcal{A}$ is sharp, what are the *first-order* asymptotics of $p_c(\mathcal{A})$?

Question 1.15. If the threshold for $\mathcal{A}$ is sharp, what is (the asymptotic order of) $T_\varepsilon(\mathcal{A})$?

(Question 1.12 asks e.g., whether $p_c(\mathcal{A}) = \Theta((\log n)/n)$, and Question 1.14 asks e.g., whether the correct leading constant is 1 or 1.7 or so on. We think of p_c as a sort of proxy for where the threshold is; in particular, Question 1.14 is only really interesting when we have a sharp threshold — otherwise it doesn't really matter, because e.g., $p_{1/3}(\mathcal{A})$ and $p_{1/2}(\mathcal{A})$ could have different leading constants.)

If we could answer all these questions about $\mathcal{A}$, then we'd understand $\mathcal{A}$ pretty well.

Remark 1.16. There are also other questions you could ask, such as about *large deviations* — here we're talking about what *typically* happens in $\mathcal{G}(n, p)$ (for which values of p is the property likely to hold or not hold?). But you can also consider values of p for which it's unlikely to hold and ask *how* rare it is. (But we won't talk about this here.)

It's a fact that there always *is* a threshold (though not necessarily a sharp one).

Theorem 1.17 (Bollobás–Thomason)

Every monotone property has a threshold.

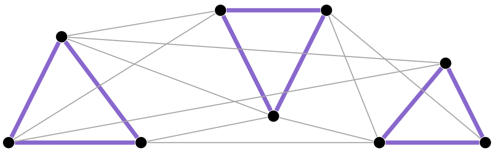
The proof is essentially to take independent copies of your random graph. (You can generate $\mathcal{G}(n, p)$ by taking the union of k independent copies of $\mathcal{G}(n, q)$ for $q \approx p/k$, and if the property holds in any one of these copies, then it also holds in their union; this allows you to jump from small constant probability to probability very close to 1.)

§1.5 Some concrete problems

In the study of these types of questions, there's a few big problems that have motivated people.

Question 1.18 (Shamir's problem)**.** When does a random graph have a triangle factor (or when does a random hypergraph have a perfect matching)?

(A *triangle factor* is a collection of vertex-disjoint triangles covering all vertices.)



For a long time, even Question 1.12 (of understanding the order of p_c) wasn't known.

Question 1.19 (Random k-SAT)**.** Suppose we pick a random Boolean formula with k literals per clause (for a certain number of clauses); when (i.e., for what regimes of the number of clauses) is it satisfiable?

Question 1.20 (Random graph coloring)**.** When is a random graph 3-colorable (or q-colorable)?

This problem was actually from the original paper of Erdős and Rényi, and it's still open — we know p_c is $\Theta(1/n)$, but we don't know its first-order asymptotics.

§1.6 Some tools

As people study these particular problems, they've developed several general tools. For Shamir's problem, the fractional or full Kahn–Kalai conjecture (which was recently proved) gives tools for finding the asymptotic order of p_c. Shamir's problem was first solved by Johansson–Kahn–Vu without it, but gives tools for solving several problems like this as well.

Then there's Question 1.13 of whether the threshold is sharp. For random k-SAT, we actually have an extremely satisfying answer, due to Friedgut — very roughly speaking, a monotone property has a sharp threshold unless it's essentially the property of containing some constant-sized subgraph or a (constant-sized) list of such subgraphs. For example, containing a triangle is a property of having some constant-sized subgraph, so it doesn't have a sharp threshold; but any property that can't be *approximated* by such a property (of containing a list of small subgraphs) *does* have a sharp threshold.

Remark 1.21. The word *approximate* is necessary — for example, consider the problem of being bipartite (which doesn't have a sharp threshold). Being bipartite is equivalent to not containing any odd cycles, and on one side we can approximate it by the property of not having an odd cycle of lengths 1 to 100 (for example); increasing the value of 100 gives better and better approximations (but we do need these to be approximations in order to have a finite list).

So we have general-purpose tools for Questions 1.12 and 1.13. Will hopes that people might be able to come up with general-purpose tools for Question 1.15 (of bounding scaling windows) as well. Friedgut's theorem gives *some* bound in this direction if we dig into the details of the proof — for example, for random graph coloring we know $p_c(\mathcal{A}) = \Theta(1/n)$, while Friedgut's theorem gives that

$$T_\varepsilon(\mathcal{A}) = O\left(\frac{1}{n \log\log n}\right)$$

(or something similar).

One possible way to bound scaling windows is by *hitting time results* — sometimes you can show that around the threshold, the property $\mathcal{A}$ is very similar to some simpler property (for example, we saw this with connectivity in Example 1.11 — around the threshold, connectivity is very similar to not having an isolated vertex, and isolated vertices are much easier to study). But for many 'hard' (e.g., NP-hard) properties, you don't expect something like this to hold. And it would be great if we could come up with tools for handling such situations.

Finally, Question 1.14, on the first-order asymptotics for the threshold, is fiendishly difficult — even in many of the cases where we know there *is* a sharp threshold by Friedgut's theorem, we don't know the first-order asymptotics. We'll mostly focus on this question for the rest of the talk — we'll consider problems like random graph coloring and random k-SAT, and we'll talk about how you might *predict* what the first-order asymptotics *should* be, as well as how you might start proving things.

§2 Second moment methods

In both our simple examples (Examples 1.5 and 1.11), the way we found the thresholds was by reducing the problem to understanding a random variable whose expectation and variance we could control. (We did this directly for Example 1.5 (for triangles); for Example 1.11 (for connectivity) it took a bit more fiddling around.) You might hope to be able to do something like this more for other problems — even if the initial structures or properties we're considering are complicated, we might hope that if we really understand what's causing the behavior of these properties, then we can condition on the appropriate things and reduce everything to some random variable whose expectation and variance we can control (which tells us something about the original distribution). This in some sense would be a dream, but it's actually the idea behind lots of the methods used here — we try to pinpoint what's really going on with our property and hope that in the end, things are controlled by a random variable whose mean and variance we can understand.

§2.1 Our models

We'll focus on the problems of random graph coloring and random k-SAT; we'll start by more precisely defining the models for how these problems are set up.

Question 2.1 (Random graph coloring)**.** Fix $q \geq 3$. Is $\mathcal{G}(n, d/n)$ q-colorable?

Here d is the 'average degree' of our random graph — so we're considering the Erdős–Rényi random graph where each edge is included with probability d/n, and studying the property of q-colorability (in particular, we want to understand for which d this random graph is q-colorable with high vs. low probability).

Question 2.2 (Random k-SAT)**.** Suppose we take a uniform random Boolean k-CNF formula on n variables with $m = \alpha n$ clauses (where a *clause* is an **OR** of k literals, i.e., variables and their negations — e.g., if $k = 3$, then $\overline{x_1} \vee x_3 \vee \overline{x_5}$ is a valid clause) — this means we choose the k literals in a clause uniformly at random, and we do this independently for each clause. Is this formula satisfiable?

For a formula to be *satisfiable* means that there is an assignment of `True` and `False` to each variable such that each clause has at least one true literal. (Again, we want to understand for which values of α this random formula is satisfiable with high vs. low probability.)

We'll also consider a slight modification of k-SAT called k-NAE-SAT (NAE stands for 'not all equal'), where instead of just asking each clause to contain at least one true literal, we ask that it contains at least one true literal as well as at least one false literal (in other words, the k literals in the clause are not all equal).

Question 2.3 (Random k-NAE-SAT)**.** Suppose we take a uniform random Boolean k-CNF formula on n variables with $m = \alpha n$ clauses. Does this formula have an assignment in which every clause has at least 1 and at most $k - 1$ true literals?

§2.2 Random graph colorings

We'll start by considering random graph colorings — so we consider $\mathcal{G}(n, d/n)$, and we want to understand whether it has a q-coloring.

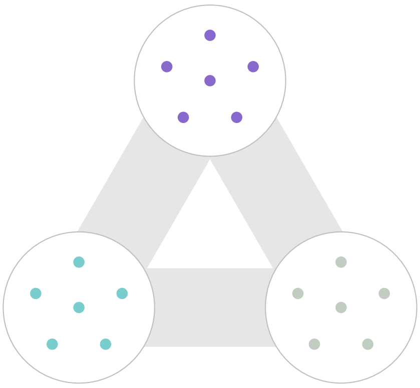
§2.2.1 A first attempt

One natural random variable we might try to look at is

$$X = \#(q\text{-colorings of } \mathcal{G}(n, d/n)).$$

And we can try to look at the moments of X — for the first moment $\mathbb{E}[X]$, if $\mathbb{E}[X] \to 0$ then with high probability our graph isn't q-colorable (by Markov). Meanwhile, if $\mathbb{E}[X] \to \infty$, then we can try looking at the *second* moment — if we can show the second moment is $\mathbb{E}[X^2] = (1 + o(1))\mathbb{E}[X]^2$, then we get that when $\mathbb{E}[X] \to \infty$ the graph *is* q-colorable with high probability.

The calculation of the first moment $\mathbb{E}[X]$ is actually quite interesting. There's q^n possible colorings, and we'll estimate the probability that each of them is a valid (i.e., proper) q-coloring for our graph. We can think of the q-coloring as splitting the vertices into q parts (namely, the color classes); and for most colorings these parts are roughly balanced, so we'll imagine the part sizes are all equal (i.e., of sizes n/q). Then for the coloring to be proper, we need there to be no edges inside each part (we can imagine having fixed the parts first, and only *then* placing down the edges).



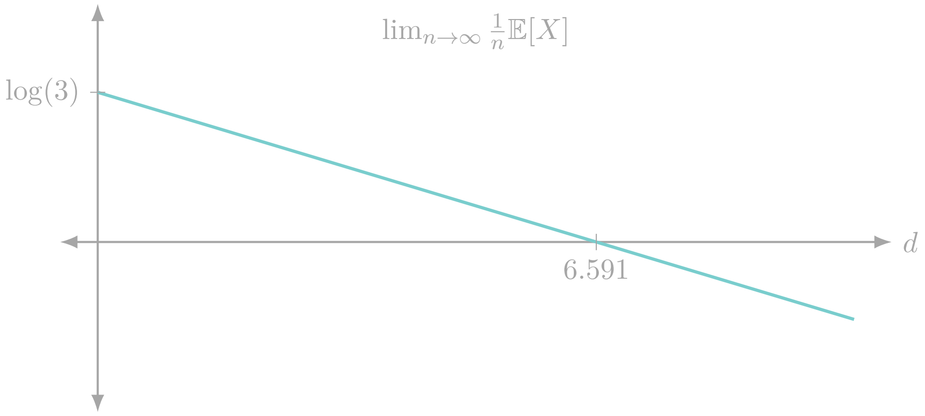
And roughly a $1/q$-fraction of all edges (i.e., roughly $n^2/2q$ edges) are inside a part; each of these edges *doesn't* appear with probability $1 - p$ (where $p = d/n$), so the probability that none of them appear (and therefore that this coloring is proper) is roughly $(1-p)^{n^2/2q}$, giving

$$\mathbb{E}[X] \approx q^n(1-p)^{n^2/2q} = q^n \left(1 - \frac{d}{n}\right)^{n^2/2q} \approx q^n e^{-dn/2q}.$$

And we want to know whether this quantity goes to 0 or ∞ (depending on d). As we change d, this changes on an *exponential* scale (as a function of n), so the right quantity to look at is

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbb{E}[X] = \log q - \frac{d}{2q}.$$

If this quantity is positive, then $\mathbb{E}[X]$ is exponentially big; and if it's negative, then $\mathbb{E}[X]$ is exponentially small. And as we increase d this gets smaller and smaller, so there's some point at which it crosses 0 — for $q = 3$, this value is $d \approx 6.591$.



§2.2.2 Moving to $\mathcal{G}(n, m)$

We could actually improve this analysis (i.e., get a smaller value of d for which the random variable we're considering has tiny expectation) by moving to the model $\mathcal{G}(n, m)$ (where we *fix* a number m of edges and place them down randomly) rather than $\mathcal{G}(n, p)$ (where we place down each edge independently).

If $p = d/n$, then with high probability the number of edges in our graph will be roughly $p\binom{n}{2} \approx dn/2$. So instead of working with $\mathcal{G}(n, p)$ with $p = d/n$, we'll work with $\mathcal{G}(n, m)$ with $m = dn/2$, and we'll do the same exercise there — we'll define

$$X = \#(q\text{-colorings of } \mathcal{G}(n, dn/2)),$$

and we'll attempt to compute $\mathbb{E}[X]$.

We can again fix some coloring and check the probability that it's proper, and we can again assume it's balanced (since most colorings are roughly balanced). Now we've got a fixed number of edges (namely, $dn/2$), and for each edge, we're picking its two endpoints randomly (independently over all edges). Once we've picked the first endpoint, the edge is 'good' if and only if the second endpoint is not in the same part, which means a $(1 - 1/q)$-fraction of the choices are good; and so we get

$$\mathbb{E}[X] \approx q^n \left(1 - \frac{1}{q}\right)^{dn/2}.$$

And now we can normalize in the same way as before to get

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbb{E}[X] = \log q + \frac{d}{2} \log\left(1 - \frac{1}{q}\right) \tag{1}$$

(and we're again interested in the value of d at which this crosses 0). This turns out to actually be much better — for example, when $q = 3$, the value we get is $d \approx 5.419$ (as opposed to 6.591).

This is a bit puzzling — we normally expect $\mathcal{G}(n, p)$ and $\mathcal{G}(n, m)$ to behave very similarly (where m is the typical number of edges in $\mathcal{G}(n, p)$), so why doesn't that happen here? Intuitively, the point is that we're considering the *number* of colorings. And with $\mathcal{G}(n, p)$, there's the possibility of having *very* few edges, and therefore an *extremely* large number of colorings. This event is rare — you typically should have roughly the expected number of edges — but since its contribution is huge, there's a fight between these large deviations and it ends up really distorting the first moment picture. So switching to $\mathcal{G}(n, m)$ — which bans this rare event, and therefore removes the distortion — actually makes a huge difference.

What we're really doing here — by shifting from $\mathcal{G}(n, p)$ to $\mathcal{G}(n, m)$ — is noticing that the random variable X counting the number of colorings isn't 'well-behaved' (it has a tiny probability of being gigantic, which messes up the first moment computation). So to fix this, we *condition* on an event that holds with high probability (that the number of edges is roughly what it should be) — this conditioning is what we're doing by shifting to $\mathcal{G}(n, m)$. And this makes our random variable more well-behaved (though it's still not super well-behaved, as we'll see later).

So now we're going to shift to the model $\mathcal{G}(n, m)$, with a fixed number of edges; and we'll let d_{1st} be the value of d for which (1) crosses 0 (the reason for the name is that it's our guess for the threshold that comes from a first moment computation). If $d > d_{\text{1st}}$ then $\mathbb{E}[X] \to 0$, so by Markov the graph is *not* colorable with high probability. (More precisely, we need d to be 'significantly' greater than d_{1st} — i.e., $d \geq (1+\varepsilon)d_{\text{1st}}$ for some fixed ε — but this is automatic if we think of d as a constant not depending on n.) Meanwhile, if $d < d_{\text{1st}}$ then $\mathbb{E}[X] \to \infty$. So we can *try* to use the second moment method to show that $X \geq 1$ with high probability. And this is what we'll talk about next — what is the second moment of X, how can you look at it, and what does it tell you?

> **Remark 2.4.** We stated earlier that we could assume our coloring was balanced, since most colorings are; but we've just seen that it makes a difference whether we work with $\mathcal{G}(n, d/n)$ or $\mathcal{G}(n, dn/2)$ (even though most instances of $\mathcal{G}(n, d/n)$ have roughly $dn/2$ edges), so we might want to be wary of this assumption as well. But it turns out that it's actually fine. For the case where $\mathbb{E}[X] \to 0$, the assumption actually can be made into an inequality pointing in the right direction — a balanced coloring is *more* likely to be proper. And for the case where $\mathbb{E}[X] \to \infty$, we can redefine X to consider only colorings that are roughly balanced (and we'll still have $\mathbb{E}[X] \to \infty$, since this has very little effect on the total number of colorings considered).

§2.2.3 The second moment

We're interested in the second moment $\mathbb{E}[X^2]$ — we can compute this by summing over *pairs* of colorings $\sigma, \tau \in [q]^n$ and checking the probability that *both* are proper colorings of our random graph, meaning that

$$\mathbb{E}[X^2] = \sum_{\sigma,\tau \in [q]^n} \mathbb{P}[\sigma \text{ and } \tau \text{ are both proper colorings of } \mathcal{G}(n, d/n)].$$

We're again only going to focus on balanced colorings (which is fine for the same reasons as in Remark 2.4). Then we can rewrite this as

$$\mathbb{E}[X^2] = \sum_{\sigma} \mathbb{P}[\sigma \text{ proper}] \sum_{\tau} \mathbb{P}[\tau \text{ proper} \mid \sigma \text{ proper}].$$

If we ignored the inner sum (so we just had $\sum_\sigma \mathbb{P}[\sigma \text{ proper}]$), then this would just be the first moment $\mathbb{E}[X]$. And the inner sum will have roughly the same value for *any* balanced σ (by symmetry — the only thing about σ that matters is its part sizes). So we can fix some canonical balanced coloring σ, and then we get

$$\mathbb{E}[X^2] \approx \mathbb{E}[X] \cdot \sum_{\tau} \mathbb{P}[\tau \text{ proper} \mid \sigma \text{ proper}]. \tag{2}$$

So what we really want to do is focus on the sum $\sum_\tau \mathbb{P}[\tau \text{ proper} \mid \sigma \text{ proper}]$ (specifically, we *want* to show it's $(1 + o(1))\mathbb{E}[X]$, in order to be able to use the second moment method).

If we fix σ and consider some other balanced coloring τ, what properties of τ does $\mathbb{P}[\tau \text{ proper} \mid \sigma \text{ proper}]$ depend on? To compute this probability, we imagine placing $m = dn/2$ edges by choosing their endpoints at random from different colors of σ, and hoping that they're also in different colors of τ. So what really matters is the sizes of the 'overlaps' between σ and τ — we want to record what fraction of vertices get each pair of colors under σ and τ (e.g., what fraction of vertices are red in both σ and τ, or red in σ and blue in τ, or so on). We can write this as an *overlap matrix* — a $q \times q$ matrix recording all these fractions (which will have row and column sums roughly $1/q$).

			τ	
		🟪	🟦	⬜
	🟪	x_{11}	x_{21}	x_{31}
σ	🟦	x_{12}	x_{22}	x_{32}
	⬜	x_{13}	x_{23}	x_{33}

For example, two possible overlap matrices are

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \text{ and } \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{bmatrix}$$

(the first corresponds to σ and τ being completely uncorrelated, and the second to σ and τ being the exact same). And the first overlap matrix is much more likely to occur (i.e., there's many more τ's with the first overlap matrix than the second), but the second has much bigger contribution to $\mathbb{E}[X^2]$ (since $\mathbb{P}[\tau \text{ proper} \mid \sigma \text{ proper}]$ is 1 if $\tau = \sigma$ and much smaller if they're unrelated). So there's some battle between the probability $\mathbb{P}[\tau \text{ proper} \mid \sigma \text{ proper}]$ corresponding to an overlap matrix, which we'll describe as its *energy* (after a suitable transformation), and the number of colorings τ with that overlap matrix, which we'll describe as its *entropy*.

More precisely, $\mathbb{E}[X^2]$ will be exponential in n, so the correct thing to consider is $(\log \mathbb{E}[X^2])/n$ (similarly to before, where we considered $(\log \mathbb{E}[X])/n$). We'll get one term from the $\mathbb{E}[X]$ factor in (2) and another term from the second factor, which we can write a sum over all overlap matrices M. There's only polynomially many overlap matrices (there's at most n choices for each entry, and a constant number of entries), so we can replace this sum with a maximum (this introduces an additive error of at most $(\log n)/n$, which doesn't matter). Then this second factor has a contribution of

$$\frac{1}{n} \log \sum_{\tau} \mathbb{P}[\tau \text{ proper} \mid \sigma \text{ proper}] \approx \max_{M} \{\mathsf{Energy}(M) + \mathsf{Entropy}(M)\} \tag{3}$$

(where M ranges over all overlap matrices), if we define the entropy and energy appropriately — specifically,

$$\mathsf{Energy}(M) = \frac{1}{n} \log \mathbb{P}[\tau \text{ proper} \mid \sigma \text{ proper}]$$

for any (σ, τ) with overlap matrix M (this probability only depends on their overlap matrix), and

$$\mathsf{Entropy}(M) = \frac{1}{n} \log \#\{\tau \mid M \text{ is the overlap matrix of } (\sigma, \tau)\}.$$

§2.3 Some plots

We've now got a formula (3) that estimates the extra factor going into the second moment for graph coloring; and we get to use the second moment method if $\mathbb{E}[X^2] = (1 + o(1))\mathbb{E}[X]^2$, meaning that this extra factor is not much more than the first moment $\mathbb{E}[X]$. And we can imagine trying to plot $\mathsf{Energy}(M) + \mathsf{Entropy}(M)$ as a function of M (for a given value of d), in order to figure out where its maximum is. We're not actually going to do this, because M is a matrix and so there'd be too many variables. Instead, we'll do this for k-SAT and k-NAE-SAT.

First let's consider k-SAT; here we'll take our random variable X to be the number of satisfying assignments (to our random k-CNF). (We'll write assignments as ±1-vectors, and we'll still refer to them by symbols such as σ and τ.) Then we have

$$\mathbb{E}[X^2] = \sum_{\sigma,\tau} \mathbb{P}[\sigma \text{ and } \tau \text{ are both satisfying}].$$

By symmetry, we can assume that $\sigma = (1, 1, \dots, 1)$ (every time we build a clause, we're placing random *literals* in it — meaning we're possibly negating our variables — so it's symmetric whether we set any given variable to `True` or `False`). Then for each τ, the probability that σ and τ are both satisfying only depends on the number of 1's in τ, which we'll call βn, and we can compute that

$$\mathbb{P}[\sigma \text{ and } \tau \text{ both satisfying}] = (1 - 2^{1-k} + 2^{-k}\beta^k)^{\alpha n}$$

(there's αn independent clauses; for each clause (which we can imagine building by independently choosing k literals), the probability each of σ and τ fails to satisfy it is 2^{-k}, while the probability that *both* fail to satisfy it is $2^{-k}\beta^k$).

Meanwhile, the *number* of τ's corresponding to a given β is $\binom{n}{\beta n} \approx \beta^{-\beta n}(1-\beta)^{-(1-\beta)n}$. And finally, there's 2^n choices for σ (all of which are symmetric). So we get that

$$\frac{1}{n}\log \mathbb{E}[X^2] \approx \max_{\beta}\left\{\log 2 + \mathsf{H}(\beta) + \alpha\log(1 - 2^{1-k} + 2^{-k}\beta^k)\right\},$$

where $\mathsf{H}(\beta) = -(\beta\log\beta + (1-\beta)\log(1-\beta))$. (We again can convert the sum over β into a maximum, since there's only n choices for β.) We can define a function $f^2_{k\text{-SAT}}$ capturing this quantity as a function of β (the superscript of 2 is to emphasize that this comes from a second moment; it doesn't denote a square) — so we define

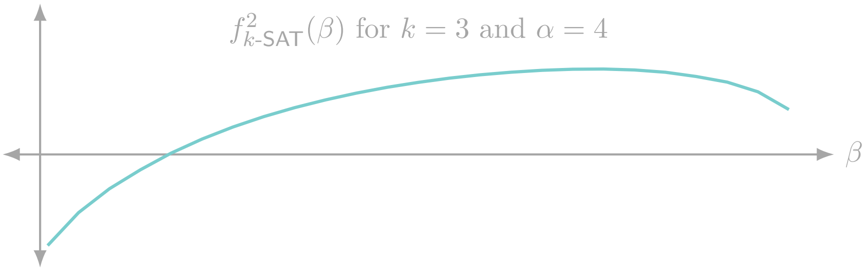
$$f^2_{k\text{-SAT}}(\beta) = \log 2 + \mathsf{H}(\beta) + \alpha\log(1 - 2^{1-k} + 2^{-k}\beta^k).$$

(We can think of $\mathsf{H}(\beta)$ as the entropy of β, and $\alpha\log(1 - 2^{1-k} + 2^{-k}\beta^k)$ as the energy.)

And we can do the same for k-NAE-SAT; there the analogous function will be

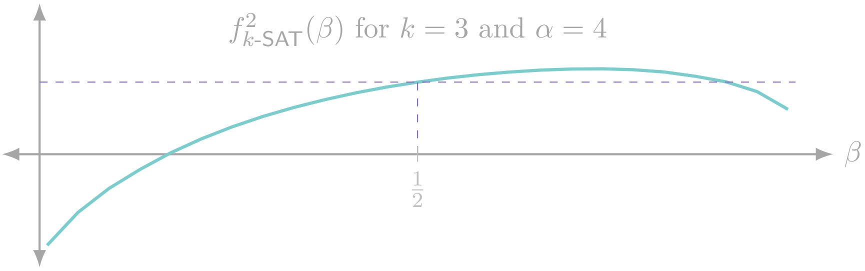
$$f^2_{k\text{-NAE-SAT}}(\beta) = \log 2 + \mathsf{H}(\beta) + \alpha\log(1 - 2^{2-k} + 2^{1-k}(\beta^k + (1-\beta)^k)).$$

We can now imagine plotting these (as functions of β, for fixed values of α). Intuitively, what such a plot captures is, on an exponential scale, what's the contribution of pairs of assignments with overlap β to the second moment?

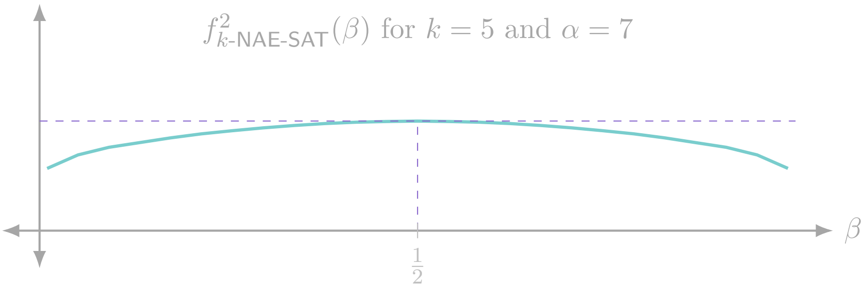


What do we *want* this plot to look like? We're trying to use the second moment method, so we want to have $\mathbb{E}[X^2] = (1 + o(1))\mathbb{E}[X]^2$. Intuitively, we get a contribution of $\mathbb{E}[X]^2$ to the second moment from $\beta = 1/2$ (which corresponds to the two assignments being independent — we can see this from the above calculation, as for $\beta = 1/2$ we get $\mathbb{P}[\sigma \text{ and } \tau \text{ are both satisfying}] = (1 - 2^{-k})^2 = \mathbb{P}[\sigma \text{ satisfying}]\mathbb{P}[\tau \text{ satisfying}]$). So we *want* this function to be maximized at $\beta = 1/2$ — then we can (very roughly speaking) say that $\mathbb{E}[X^2]$ isn't much bigger than $\mathbb{E}[X]^2$ (since the biggest contribution comes from $\beta = 1/2$, and this contribution is just $\mathbb{E}[X]^2$).

But for k-SAT, we're in trouble — the problem is that the entropy function $\mathsf{H}(\beta)$ is symmetric about $1/2$, but the energy function $\alpha \log(1 - 2^{1-k} + 2^{-k}\beta^k)$ is *not* symmetric about $1/2$. This means the derivative of our function at $1/2$ will not be zero, so we certainly can't have a maximum there.

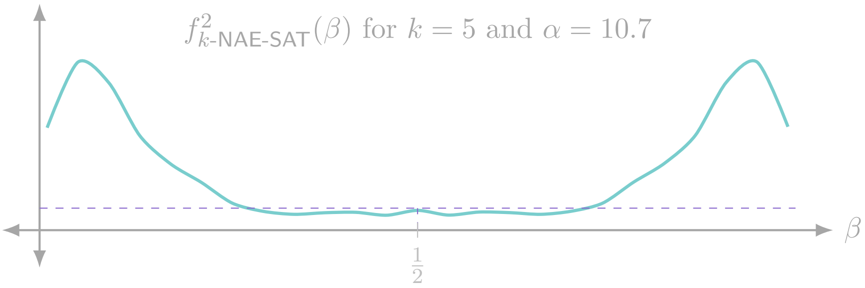


For k-NAE-SAT we're actually doing better, because the function really is symmetric about $1/2$ (which means there's hope that its maximum is there). We'll take $k = 5$ for concreteness; then when $\alpha \approx 7$, things are really nice — the maximum of the function really is at $1/2$.



So here, we really *can* use the second moment method to get what we want (that the formula is satisfiable with high probability). But on the other hand, we can see that at $\beta = 1/2$ this function is positive. This means $\mathbb{E}[X]$ is quite large — so α is actually significantly below the first moment threshold. (As α increases, $\mathbb{E}[X]$ decreases; and the first moment threshold is when $\mathbb{E}[X] = 1$, meaning that we expect to have just one satisfying assignment. Here we expect to have *many* satisfying assignments, so we're pretty far from the first moment threshold.) So we get *some* lower bound for the actual threshold, but it doesn't match the upper bound we get from the first moment calculation.

Meanwhile, if we take $\alpha \approx 10.7$, then the function really is 0 at $\beta = 1/2$, which is what we'd want in order to match the first moment threshold. But even though we have a *local* maximum at $\beta = 1/2$, it's not actually a *global* maximum — we instead have annoying things popping up on the sides.



And so here, the second moment computation *doesn't* work (because we don't have a global maximum at $1/2$). What's going wrong is that we're getting strange peaks close to 0 or 1, corresponding to solutions that really agree or really disagree with σ (for NAE-SAT, really agreeing and really disagreeing are symmetric).

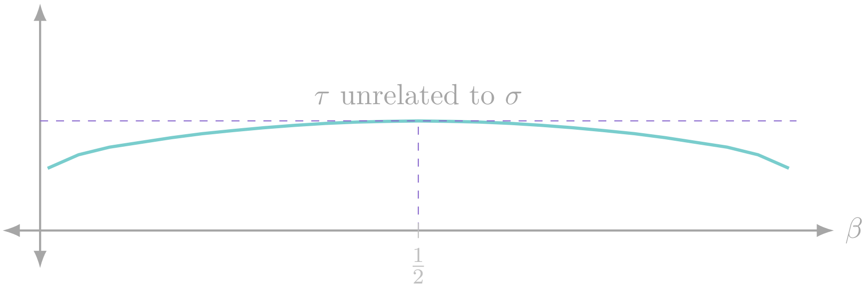
§2.4 Planted models

We'll now talk about a different way of generating random instances. First working with k-SAT for concreteness, imagine that we first choose a 'special' assignment $\sigma \in \{\pm 1\}^n$ uniformly at random, and then we choose our k-CNF formula at random *conditioned* on σ being a satisfying assignment — so we choose each of the αn clauses only from the set of clauses that σ satisfies. This is called the *planted model* (for k-SAT). More generally, in a planted model, instead of first picking out our instance (e.g., our formula or random graph), we first pick a *solution*, and then we choose our instance conditioned on this really being a solution.

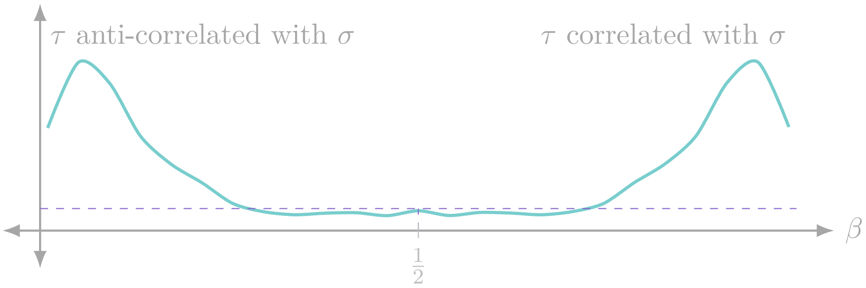
Now we've got an instance coming with one very special solution, but we can still consider the same variables as before — how *many* colorings does our random graph have, or how many satisfying assignments does the formula have? And we can imagine computing the first moment of this random variable; we've got one special solution, so we can again group the remaining solutions by their 'distance' to the special one and talk about the contributions to the first moment from each of these distances, which gives us another curve (similarly to how we got a curve $f^2_{k\text{-SAT}}(\beta)$, for example, for the second moment in the usual model).

And in fact, the first moment curve we get here will be exactly the second moment curve in the usual model, except with the constant $\log 2$ removed (this is essentially true by definition).

So the above plots have an intuitive interpretation in terms of the planted model. When the plot looks like a single hump at 1/2, this means, loosely speaking, almost all the solutions to the formula are 'far away' — they look completely random with respect to our special solution σ.



Meanwhile, when the plot has bumps near the ends, this means most solutions are actually correlated (or anti-correlated) with our special solution σ.



> **Remark 2.5.** These statements are really just statements about the contribution of other solutions to the *first moment*. They're not statements about typicality (e.g., that for most instances generated in this way, most solutions will be correlated with σ) — that would require us to show that the planted model really does behave like its first moment, which needs argument.

§2.5 The random discrepancy model

Finally, we'll look at one more model, called the *random discrepancy model* or *symmetric binary perceptron.* Here we're picking m independent n-dimensional standard Gaussians $Z_1, \ldots, Z_m$, which we think of as 'constraints' (so each of these is a vector of length n, and there's m of them; we'll let $m = \alpha n$). And given κ, we say $\sigma \in \{\pm 1\}^n$ is a *low-discrepancy solution* if $|\sigma \cdot Z_i| \leq \kappa\sqrt{n}$ for all $i \in \{1, \ldots, m\}$.

If we fix σ, then for each i, the dot product $\sigma \cdot Z_i$ is a Gaussian random variable with mean 0 and variance n, which means $\mathbb{P}[|\sigma \cdot Z_i| \leq \kappa\sqrt{n}]$ is some constant. But we want this to hold for *all* i (this is why we think of each Z_i as defining a constraint).

If we want to analyze the number of low-discrepancy solutions, which we'll again call X, we can compute the first and second moments in the same way as with our other models (e.g., graph coloring and k-SAT). For the first moment, we have

$$\mathbb{E}[X] = 2^n(\mathbb{P}_{z \sim \mathcal{N}(0,1)}[|z| \leq \kappa])^{\alpha n}$$

(since we can first fix σ, and then each $\sigma \cdot Z_i$ is an independent Gaussian, and there's $m = \alpha n$ of them).

Meanwhile for the second moment, when we're considering two solutions σ and τ, what matters is their overlap — if they have βn entries in common then $\sigma \cdot Z_i$ and $\tau \cdot Z_i$ are $(1 - 2\beta)$-correlated Gaussians, which means we get some formula where the energy comes from the probability that two $(1 - 2\beta)$-correlated standard Gaussians are between $-\kappa$ and κ.

The curve we get from this in some sense looks a lot like the one from NAE-SAT — the problem setup is actually very similar, in that we're looking at a certain collection of inner products and we want them all to be at most something and at least something. In particular, the curve is symmetric about 1/2, and we again have a local maximum there. But this time it's actually a *global* maximum — and even more, instead of going *up* at 0 and 1 (as with NAE-SAT), here we actually go *down* (below the x-axis).

This means a couple of things. First, $\beta = 1/2$ turns out to be the global maximum for *any* $\alpha < \alpha_{1\text{st}}$ (where $\alpha_{1\text{st}}$ is the first moment threshold — the value of α for which the first moment is 1) — so whenever the first moment curve tells us that there's expected to be exponentially many solutions, the second moment curve is maximized at $\beta = 1/2$. And this means first and second moment methods fully *work* for this model (to get the exact threshold — because for all $\alpha < \alpha_{1\text{st}}$ we can then use the second moment to deduce that there *are* solutions with high probability, while for $\alpha > \alpha_{1\text{st}}$ the first moment is enough to tell us there aren't), which is kind of crazy.

The second thing it means is that if we move to the planted model (where we plant a solution σ), then with high probability there are actually no solutions within εn of σ (for some small $\varepsilon > 0$ corresponding to where the curve goes below the x-axis near 0, because the curve going below the x-axis means we expect exponentially few solutions with such values of β).

§2.6 When does the second moment work?

In fact, these two statements about the random discrepancy model — that there's a maximum at $\beta = 1/2$ for α all the way up to the first moment threshold (meaning that the second moment computation works), and that with high probability in the planted model there's no solutions 'close' to σ — are not unrelated. And the connection actually means we can know, even before looking at the plots, that the second moment method for graph coloring or NAE-SAT won't work all the way up to the first moment threshold.

With graph coloring, for example, the fact that the curve goes 'up' (above the x-axis) when we're close to σ means that if there's one coloring, then there's actually exponentially many — we get a whole *cluster* of colorings which are 'close' to each other. And it intuitively makes sense that this happens, because of isolated vertices — the graph will typically have isolated vertices, and we can color them however we want, producing exponentially many colorings (as long as there's one to start with).

And if we know there's always either 0 or at least $2^{0.1n}$ colorings (for example), then the first moment threshold *can't* be the correct threshold — the first moment threshold is where our *expected* number of colorings goes from exponential to exponentially rare, but at the actual threshold, the number of solutions should *still* be exponential (since when there's a coloring, there's exponentially many).

(The same thing happens with NAE-SAT, where instead of isolated vertices we've got variables that don't appear anywhere.)

How could we try to fix this? One idea is that when we're trying to count the number of colorings, instead of counting each individual coloring, we actually count *clusters* — so if we've got a cluster of colorings of size $2^{0.1n}$, we count it as 1 instead of $2^{0.1n}$. And then we instead try to do first and second moments on the number of *clusters* instead. This is more complicated, but it's actually the right idea.

§2.6.1 Second moments and the planted model

Here's a vague but pretty general statement: when the second moment works, the planted model should be 'close' to the usual model with a randomly chosen solution.

First, what does this mean? We can think of the planted model as a distribution not just on instances, but solution-instance *pairs* (e.g., we're picking both a satisfying assignment and a formula, or both a coloring and a graph). But we could also imagine picking a random graph and a uniformly random coloring, or picking a random k-SAT instance and a random solution. And the above statement says that when the second moment works, these two distributions on solution-instance pairs are 'close.'

There's several ways to make this rigorous (in particular, what do we mean by close?), and the best we can hope for is *asymptotic contiguity* — that any statement that holds with high probability in one model also does in the other.

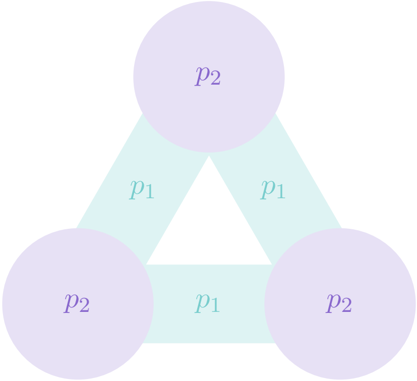
> **Remark 2.6.** Certainly this can't be true above the actual threshold (i.e., in the regime where the usual model has no solution with high probability) — because then there exists a solution with probability 1 in the planted model and with very small probability in the usual model. (It doesn't even really make sense to talk about the usual model with a random solution, because there usually aren't any solutions to choose from.)

And for random graph coloring, the best lower bound we have on the q-colorability threshold is actually the point up to which this statement is true. We essentially analyze the planted model and say that up to a certain point these two distributions (the planted model, vs. a random instance with a random solution) are close; and then since the planted model always has a solution, with high probability the usual model should also have a solution. (Meanwhile, the best *upper* bound on the threshold comes from some variant of the first moment method.)

§3 Conclusion

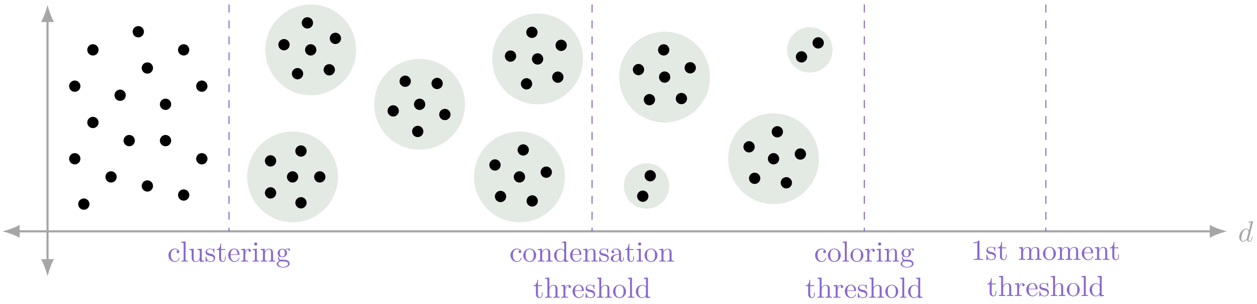
We can try to use moment methods to understand the random model (i.e., to understand whether there is a solution, and how many there are). And we've seen that the second moment computation is actually an optimization problem, and we really want to understand what the optimizer is (e.g., we wanted our second moment curves to be maximized at $\beta = 1/2$). And this is closely related to the planted model (specifically, to the *first* moment in this model).

The planted model is also interesting in its own right. For graph colorings, it's a variant of another model, called the *stochastic block model.* Here we plant a partition of our graph, and we have two different probabilities — we have one probability for edges that cross the partition (meaning that they're added with this probability), and another for edges inside parts. And when we're working with this model, we're usually interested in the task where we're given this graph, and we want to recover some information about the original partition.



The planted model for colorings is an extreme version where edges within parts have probability 0. And here we're given the graph, and we're interested in whether we can recover the planted coloring (better than guessing randomly). We can think of the point up to which we can or can't distinguish this planted model from a truly random graph $\mathcal{G}(n, p)$ as the 'information-theoretic threshold.' And it turns out that this threshold has another interesting interpretation — if we work with the random graph $\mathcal{G}(n, d/n)$ and let X denote its number of colorings, then this is the threshold for where $X^{1/n}$ converges in probability to its expectation — meaning that the first moment is correct on the level of the exponent for the number of colorings (i.e., if the first moment says there's γ^n colorings on average, then there really are typically $(\gamma + o(1))^n$ colorings). We call this the *condensation threshold.*

We can imagine drawing a cartoon of what the colorings of a typical random graph look like, as a function of d. (Here the dots represent different colorings, and closeness represents similarity between two colorings.)



When d is small, we've got tons of colorings, all 'connected' to each other. Then at some point, the colorings shatter into exponentially many clusters, which are far apart and roughly of the same (exponentially large) size. And then at the condensation threshold, now there's a handful of large clusters which cover a constant fraction of everything, and the remaining clusters are all much smaller (so we've now got clusters of very different sizes). And then there's the coloring threshold, at which point there's actually no colorings. (And the first moment threshold is somewhere beyond that.)

> **Remark 3.1.** When q is large, the three thresholds (the condensation threshold, coloring threshold, and first moment threshold) are all within an additive constant of each other; they're all roughly $2q \log q$.

And roughly speaking, once we condition on the right things, we can use the first and second moment methods to get up to the condensation threshold, but they won't work past that. More precisely, it's possible to prove that the planted model and truly random model are 'close' up to a certain point — which means that since the planted model has a solution, so does the random one — and this is exactly the same point up to which we have $X^{1/n} \to (\mathbb{E}[X])^{1/n}$ (in probability).

Meanwhile, the picture for the perceptron (i.e., random discrepancy) model is quite different. As mentioned earlier, the second moment curve — and therefore the first moment curve in the planted model — goes *down* near 0. This means the solution space doesn't have clusters — almost all the solutions are hanging out by themselves. And then the first moment threshold *is* correct (i.e., it's the actual threshold to have a solution).

Finally, we'll briefly return to the question of scaling windows mentioned in the first part of the talk (Question 1.15). For this model (the perceptron), Ashwin and Mehtaab found the correct scaling window (e.g., if we imagine we're adding constraints one at a time, exactly how many constraints do we need to add to go from 99% to 1% probability of having a solution?). This is quite surprising because unlike with graph connectivity (as in Example 1.11, a simpler example where we know the scaling window), this is a model where it's 'hard' to tell whether a solution exists — in particular, it's not the case that whether a solution exists is close to an easier property (in Example 1.11, this easier property was isolated vertices).

And it turns out that the scaling window is actually *constant* (for comparison, Friedgut's theorem only gives an upper bound of $n/(\log \log n)$). In the general setting, we have very few tools to bound scaling windows beyond Friedgut's theorem (we don't even know of general-purpose tools to do a *bit* better); but this particular model is special, and we can get really strong results.