Random reconstruction in two dimensions

Talk by Bhargav Narayanan

Notes by Sanjana Das

November 17, 2023

§1 Random reconstruction

§1.1 Reconstruction problems

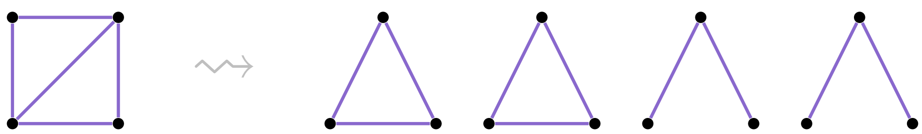
The sort of questions we'll talk about today are of the following form:

Question 1.1. Is it possible to reconstruct a discrete structure S from small snapshots?

Maybe the most famous question of this nature is the Kelly–Ulam graph reconstruction conjecture.

Conjecture 1.2 (Kelly–Ulam) — Any graph with $n \geq 3$ vertices can be uniquely reconstructed from its *deck*, the multiset of the n induced subgraphs obtained by deleting one vertex.

(In the deck, we're only given the isomorphism types of each induced subgraph, and not the vertex labels.)



This seems like it should be massively true, and you probably shouldn't even need so many subgraphs to reconstruct the graph. But we don't know how to prove it.

As another example, we can consider such questions in finite groups.

Question 1.3. Suppose that we have a set $S \subseteq \mathbb{Z}/n\mathbb{Z}$, and we're shown what all small subsets of S look like, up to translation — so we define the *k-deck* of S as

$$D_k(S) = \{\mathrm{Orb}(T) \mid T \subseteq S, \, |T| = k\}$$

(where $\mathrm{Orb}(T)$ is the orbit of T under translations). Can we say what S looks like up to translation?

(In this talk, all sets are actually multisets unless otherwise specified.)

Theorem 1.4 (Pebody)

For $S \subseteq \mathbb{Z}/n\mathbb{Z}$, we can reconstruct S up to translation using just its 6-deck $D_6(S)$.

There are also similar results about reconstructing subsets of $\mathbb{R}^2$ up to rigid motion, and there are interesting open problems about what happens when we replace $\mathbb{R}^2$ with $\mathbb{R}^3$. In particular, it's known that in $\mathbb{R}^2$, if we get to see what all subsets of size 20 (for example) look like up to rigid motion, then we can reconstruct the original set (up to rigid motion). But we don't know this in $\mathbb{R}^3$ (with 20 replaced by any finite number). In general, 2-dimensional problems tend to be easier than their higher-dimensional analogs.

§1.2 Probabilistic reconstruction problems

Today we'll talk about some *probabilistic* reconstruction problems. Here, we're still trying to reconstruct an object using its k-deck (for some appropriate definition of the k-deck). But instead of proving results for *arbitrary* (worst-case) objects, we want to know what we can do for *typical* objects.

For the types of problems mentioned earlier, these types of questions are often a lot easier — for example, you can reconstruct a random graph almost immediately. But these types of questions turn out to be more interesting on the geometry of the lattice; so almost all the questions we'll talk about will have some sort of lattice structure.

The real-world motivation for where these sorts of questions come from is the simple one-dimensional problem of sequencing DNA.

> **Question 1.5** (Shotgun sequencing of DNA)**.** Suppose S is a uniform random string in $\{0,1\}^n$, and we're given its k-deck, the multiset of all (consecutive) length-k substrings. For what values of k do we have enough information to reconstruct S?

(Here we're working over a 2-letter alphabet $\{0,1\}$ rather than $\{\mathsf{A},\mathsf{C},\mathsf{G},\mathsf{T}\}$ for simplicity.)

It's known that the answer is $k \approx \log n$ (we won't prove this, but it'll become clear from some other things we'll talk about); this was known since the 1990s, and it's useful but not too hard.

There are also variants that are more relevant to practice — for example, what if you get substrings with some errors? Lots of these variants are pretty well-understood as well.

§1.3 Two-dimensional reconstruction problems

Today we'll look at variants of such problems in two dimensions. (For most of what we'll talk about, we don't know how to solve the analogous problems in higher dimensions.)

In 2015, Mossel and Ross raised a number of questions of this nature: why stop with strings, and what happens with more general structures? In particular, they raised the following general question.

> **Question 1.6** (Mossel–Ross 2015)**.** Can we construct a labelled graph from its r-balls?

This means that the deck shows us what the graph looks like around every vertex, and we want to piece this together to get back the whole graph.

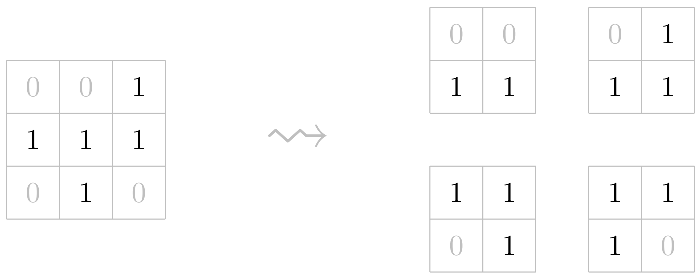
It's hard to say precise results at this level of generality, but in two dimensions, they raised two concrete questions. The first question is the *jigsaw* problem.

> **Definition 1.7.** A $n \times n$ jigsaw is a q-coloring of the edges of a $n \times n$ grid. Its deck is the multiset consisting of what each vertex looks like locally (the colors of its four incident edges, with orientation).

(Here we're looking at balls of a fixed size, but we'll think of q as varying. We also extend the edges a bit at the boundary, so that every piece in the jigsaw has four edges.)

Another natural question is with *pictures*, where we take a $\{0,1\}$-coloring and look at larger pieces.

> **Definition 1.8.** An $n \times n$ picture is a filling of the $n \times n$ grid with 0's and 1's (i.e., an element of $\{0,1\}^{n^2}$). We define its k-deck (denoted by D_k) as the set of $k \times k$ subpictures.



(Again, we get the pieces with orientation.)

We consider the following questions:

> **Question 1.9.** Suppose we sample a random $n \times n$ jigsaw. For what q (as a function of n) can we (typically) reconstruct it from its deck?

> **Question 1.10.** For what k (as a function of n) can we reconstruct a random $n \times n$ picture?

§1.4 History

Mossel and Ross looked at the probability that a random jigsaw is reconstructible, and showed that

$$\mathbb{P}[\text{reconstructible}] \to \begin{cases} 1 & \text{if } q = \omega(n^2) \\ 0 & \text{if } q = o(n^{2/3}). \end{cases}$$

They thought that the threshold should occur at n^α for some α, but weren't sure what α should be. Then BFMNPS improved the first case by showing that

$$\mathbb{P}[\text{reconstructible}] \to 1 \quad \text{if } q = n^{1+o(1)}.$$

(More explicitly, this asymptotic notation means that for every ε and sufficiently large n, if $q > n^{1+\varepsilon}$ then the jigsaw is reconstructible with high probability.)

Meanwhile, for the picture problem, Mossel and Ross showed that

$$\mathbb{P}[\text{reconstructible}] \to \begin{cases} 1 & \text{if } k \geq C\sqrt{\log n} \\ 0 & \text{if } k \leq c\sqrt{\log n}. \end{cases}$$

Later Ding and Liu worked out the sharp threshold — they showed that

$$\mathbb{P}[\text{reconstructible}] \to \begin{cases} 1 & \text{if } k \geq (1+\varepsilon)\sqrt{2\log_2 n} \\ 0 & \text{if } k \leq (1-\varepsilon)\sqrt{2\log_2 n}. \end{cases}$$

(These results also have analogs that work in higher dimensions.)

§1.5 Main results

For the jigsaw problem, the authors determine the order of the threshold.

Theorem 1.11 (Ballister–Bollobás–Narayanan)
For the jigsaw problem, we have

$$\mathbb{P}[\text{reconstructible}] \to \begin{cases} 1 & \text{if } q \geq Cn \\ 0 & \text{if } q \leq cn. \end{cases}$$

Unfortunately, this doesn't get a sharp threshold — they have guesses for where the sharp threshold should be, but don't know how to prove it.

Meanwhile, for the picture problem, it turns out that we can get a 2-point result.

Theorem 1.12 (Narayanan–Yap)
For the picture problem, there exists $k_c \asymp \sqrt{2\log_2 n}$ such that

$$\mathbb{P}[\text{reconstructible}] \to \begin{cases} 1 & \text{if } k > k_c \\ 0 & \text{if } k < k_c. \end{cases}$$

There's a specific formula for k_c; and we know that if we're one above it then we can reconstruct, and if we're one below it then we can't. (The authors have pretty good guesses for what happens at k_c as well, but don't know how to push things through in that case.)

§2 Ideas for lower bounds

We'll talk about the two problems interchangeably, because the methods are similar.

§2.1 Local obstacles

The earlier lower bounds came from looking at local obstacles. For example, for jigsaws, suppose we have two portions of the jigsaw which look as follows.

Then when we're putting together the jigsaw, we won't know which location has the purple edge and which has the green one — the two are interchangeable (i.e., if we took a jigsaw and flipped the purple and green edges around, we'd get the same multiset of pieces).

More generally, this occurs whenever the six edges other than the middle one line up; so the expected number of such pairs is n^4q^{-6}. This is how Mossel and Ross showed that when $q \ll n^{2/3}$, we can't reconstruct — at such values of q we start seeing these configurations, and that's an obstacle to reconstruction.

§2.2 Entropy

But in fact, there's an even bigger obstacle staring us in the face. As a simple question, why can't you reconstruct a picture from its 1-deck? One answer is just that there's not enough information. And it turns out that the correct obstacle to consider in this problem is entropy — looking at the amount of information you have.

For pictures, the point is that if k is too small, there's just not enough information to be able to reconstruct. The number of possible pictures is 2^{n^2}; meanwhile, the number of possible k-decks is at most the number of solutions to

$$x_1 + \cdots + x_{2^{k^2}} = (n-k+1)^2$$

(there are 2^{k^2} possible $k \times k$ subpictures; here x_i represents the number of times the ith subpicture appears in the deck). This immediately tells us that the probability we can reconstruct is bounded above by the ratio of these quantities, which gives that

$$\mathbb{P}[\text{reconstructible}] \le \binom{(n-k+1)^2 + 2^{k^2}}{2^{k^2}} \cdot 2^{-n^2}.$$

This goes to 0 when $k < k_c$ (for our definition of k_c), showing that in tht case, we can't reconstruct.

Remark 2.1. We expect that it *should* be true that as soon as you have enough information, you should be able to reconstruct. Depending on what k_c is, you might or might not have enough information at k_c; at $k_c + 1$ you definitely have enough information, and we show that reconstruction *is* possible.

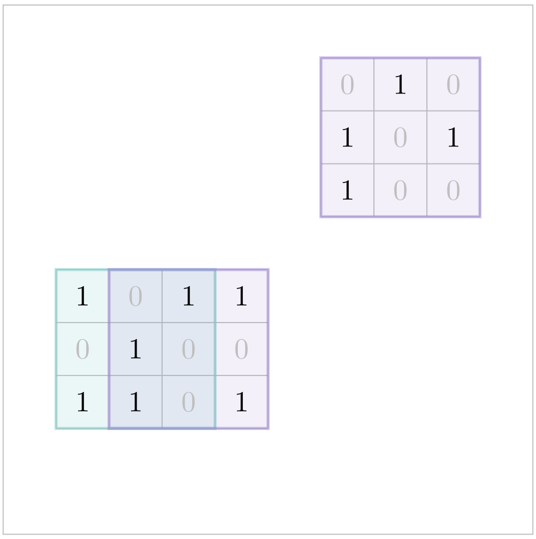
The more fine-grained conjecture is that at the critical point k_c, you can look at whether this ratio goes to 0 and ∞, and that should determine whether reconstruction is possible or not. But we don't know how to prove this (since we need a bit of wiggle room in the upper bound argument).

So the short summary is that you look at how much information you have, and that gives a bound that we expect should be right. (The authors think this is where the answer ought to be in higher dimensions as well, though we don't know how to prove that.) For pictures, we more or less get upper bounds that match the information-theoretic lower bounds; for jigsaws, we get there up to a constant factor.

§3 Upper bounds

Now we'll talk about upper bounds, and why there are some difficulties that are fine in two dimensions, but which we don't know how to deal with in higher dimensions.

Let's do a concrete calculation in the picture situation. Imagine we have a fixed $k \times k$ tile, and we're looking for which tile fits next to it. There's one tile in the deck that *should* slot in next to it (the one that actually belongs there); so we can ask, what's the probability that there's a *different* tile that also fits? (We call such tiles *impostors*.)



We can compute the expected number of impostors — if we're trying to find an impostor in a generic location, we need to engineer $k^2 - k$ equalities of random bits (the first $k - 1$ columns of the impostor need to match our tile), which means

$$\mathbb{E}[\#\text{impostors}] \approx n^2 \cdot 2^{-k^2+k} \approx \frac{1}{k}$$

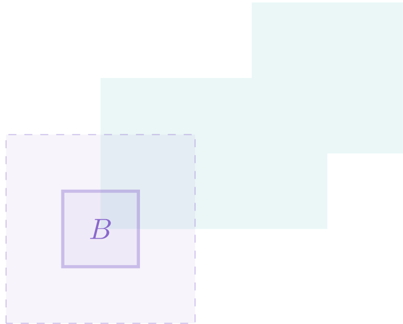
(when $k \sim k_c$). This tells us that *most* tiles aren't going to have impostors that slot in next to them. But if your reconstruction algorithm is to just greedily piece things together (finding a piece that slots in and adding it), you'll make lots of mistakes (since you have to run for n^2 steps).

So it's not enough to look *locally*; and that's not surprising, because the bottleneck in the lower bounds doesn't come from local obstacles, but rather global ones. (This reconstruction procedure would work if you increased k_c to $k_c + \log n$, but we're trying to get an exact answer.)

Here's a better strategy: suppose that we've pieced together some part S of our picture, and we're confident that it's okay.

And suppose we have another piece B and we're asking, should we put it down? If B isn't near the edge, then it should fit into some window of tiles that also fit together. So instead of asking whether we can just slot in B, we ask, can we slot in some $w \times w$ window around B?

(It's possible that the $w \times w$ window you find will have some errors; but we might hope that there's enough constraints that only the right piece will fit in.)



It turns out that this is a good algorithm for the right scale of window size — you *can* piece the puzzle together this way. What's interesting is how you analyze whether the algorithm works or not. We're trying to understand things of the form

$$\mathbb{P}[\text{there exists a fake } w \times w \text{ window}].$$

The naive thing to do is to use a union bound. But this doesn't work — it's too expensive.

Instead, what ends up working is contour arguments from percolation theory. The idea is that instead of union bounding over everything, we do a more efficient union bound over the things we really want to pay for, and that's the *interfaces* between tiles that are correct and tiles that are fake. So we decompose into contours and do a union bound over all possible diagrams with contours; and that's efficient enough to get sharp bounds.

This doesn't work in three dimensions because there we don't have contours, but rather surfaces; and we don't know how to deal with those.