1 Expander walk Chernoff bounds

Chris Beck

Abstract: Expander walk sampling is an important tool for derandomization. For any bounded function, sampling inputs from a random walk on an expander graph yields a sample average which is quite close to the true mean, and moreover the deviations obtained are qualitatively similar to those obtained from statistically independent samples. The ”Chernoff Bound for Expander Walks” was first described by Ajtai, Komlos, and Szemeredi in 1987, and analyzed in a general form by Gilman in 1988. A significantly simpler analysis was given by Healy in 2005, who also gave a more general form in which the function may differ from step to step in a certain sense. I will give an exposition of Healy’s proof and describe minor variations and extensions.

What are uses of expander walks?

1. Sampling: Find the mean of a bounded function in some domain. Instead of taking independent samples, if you take correlated samples from an expander walk, you can approach the mean optimally. We can get error reduction for BPP.

2. Amplifying security (hardness) of 1-way functions. Yao showed in 1982 that if you have a function that is polynomially hard on $1 - \frac{1}{\text{poly}(n)}$ inputs, you can get a harder 1-way function that a polynomial time algorithm can only succeed with exponentially small probability.

   Alternately take steps on an expander and taking a permutation, you get optimal amplification.

For all $r \geq 0$,

$$\begin{aligned}
\mathbb{E}(e^{r \sum X_i}) &\leq \prod_i \left(\alpha_i + \frac{\lambda (e^r - 1)^2}{1 - \lambda e^r} \right) \\
&\leq \prod_i \left(1 + (e^r - 1)\mu_i + \frac{\lambda (e^r - 1)^2}{1 - \lambda e^r} \right) \\
&\leq \prod_i \left(e^{(e^r - 1)\mu_i} + \frac{\lambda (e^r - 1)^2}{1 - \lambda e^r} \right) \\
&= \exp\left((e^r - 1)\mu + \frac{k\lambda (e^r - 1)^2}{1 - \lambda e^r} \right).
\end{aligned}$$

using $\alpha_i \leq 1 + (e^r - 1)\mu_i$ (Taylor expansion, convexity). (We assume $r \leq \frac{1}{2}$, so $r \leq e^r - 1 \leq r + \frac{2r^2}{3} \leq \frac{4r}{3}$; we also assume $e^r \leq \lambda^{-\frac{1}{2}}$, which gives $1 - e^r\lambda \geq 1 - \lambda^{-\frac{1}{2}}$. Important is that r a small value bounded away from 1. We want r to be as large as possible without breaking things.)

Observation: for all $\lambda \in [0, 1]$, $1 - \lambda \leq \ln \frac{1}{\lambda}$.

Thus

$$\begin{aligned}\mathbb{P}\,(\text{bad}) &\le \frac{\mathbb{E}(e^{rX_i})}{e^{r\mu - r\varepsilon k}}\\ &\le \exp\left((e^r - r - 1)\mu + k\left(\frac{\lambda(e^r-1)^2}{1-\lambda e^r} - r\varepsilon\right)\right)\\ &\le \exp\left(\frac{2}{3}r^2\mu + k\left(\frac{2\lambda(4r/3)^2}{1-\lambda} - r\varepsilon\right)\right)\\ &\le \exp\left(k\left(\frac{4/3r^2}{1-\lambda} - \varepsilon r\right)\right)\end{aligned}$$

using $\frac{1}{1-\sqrt{\lambda}} = \frac{1+\sqrt{\lambda}}{1-\lambda} \le \frac{2}{1-\lambda}$. Now take $r = \varepsilon\frac{1-\lambda}{2}$. We check $r \le \frac{1}{2}$ and $e^r \le \lambda^{-\frac{1}{2}}$, both true.

How can we extend this? What more is true? There's the basic version with one set; when sets can move arbitrarily, it's surprising you can get a similar result. Is there a derandomized Azuma's inequality? Consider some martingale. The most natural wayis to let the sets depend on the history of vertices walked to. (In our setting, the sets can moved around but have been fixed; they're not adaptive.)

Zuckerman has a paper: how to recycle random bits. They try to derandomize a BPP algorithm: hash down the bits used and reuse them. It's one of a bunch of results that say it's okay to leak randomness.

The significant barrier is the following.

Lemma 1.1: Let $f : \{0,1\}^n \to \mathbb{R}$ be a 1-Lipschitz function. Consider the Doob martingale of f: choose bits for input in a fixed order, and evaluate f at the end. Track how the conditional expectation changes.

Then

$$\mathbb{P}_{x\in\{0,1\}^n}[f(x) - \mathbb{E}f > \varepsilon\sqrt{n}] \le \exp(-\Omega(\varepsilon^2)).$$

Just Lipschitz gives this bound. Question: does there exist $G : \{0,1\}^{\frac{n}{4}} \to \{0,1\}^n$ such that for all f as above,

$$\mathbb{P}[f(G(s)) - \mathbb{E}f \le n^{1-\delta}]$$

for some $\delta < 1$?

For example, break up the blocks into size n^ε, then do an expander walk. Revealing the blocks one at a time, there shouldn't be too much drift.

But the answer to the question is NO. Given G, construct f to be the set of outputs of the generator. Use a sphere-packing bound. There's no way to derandomize arbitrary Lipschitz functions.

Suppose we allow each f_i to depend on 2 vertices. You can change the analysis to put the weights on the edge entries of the random walk matrix. You can still get bounds on the bookkeeping entries. The mean will still drift; you get tight concentration around the new mean. What happens when the functions have overlapping information?

We proved before that

$$\mathbb{P}[\sum_i X_i - \mu > \varepsilon k] \le e^{-\frac{\varepsilon^2(1-\lambda)}{8}}.$$

Recall that Chernoff-Hoeffding tells us the following: let X be a sum of discrete random variables in $[-1,1]$, $\sigma^2 = \mathrm{Var}(X)$, $\mu = \mathbb{E}X$,

$$\mathbb{P}(X > \mu + t\sigma) \le e^{-t^2/2}$$

We can generalize this to our setting as well. We need all the random variables to have small variance.

Lemma 1.2: Let $\sigma^2 = \max_i \mathrm{Var}(f_i)$. Suppose all rv's are in $[0,1]$. Then over an expander walk,

$$\mathbb{P}[\sum X_i - \mu > t\sigma k] \le e^{-\frac{\sigma^2 t^2 (1-\lambda)}{100}}.$$

NOT STRONGER

Get tighter bounds on the entries for b, c in the bookkeeping matrices.

Observation: For D_i as before, prove $\widetilde{D_i} \le \begin{pmatrix} \alpha_i & O(r\sigma) \\ O(r\sigma) & \lambda e^r \end{pmatrix}$. Intuitively, $\|\Pi_{\mathbb{1}^\perp} D_i \mathbb{1}/n\|_2^2 = \mathrm{Var}[e^{rX_i}] = O(r^2)\mathrm{Var}(X_i)$ by Taylor expansion.

We get

$$\begin{aligned}
\mathbb{P}(X_i \ge \mu + t\sigma k) &\le \exp\left(\frac{O(r^2)\lambda}{1-\lambda}\sigma^2 + a\right) \\
&= \exp\left(\frac{O(r^2)\lambda}{1-\lambda}\sigma^2 k - rt\sigma k\right) \\
&\le \exp\left(-\frac{t^2\sigma^2(1-\lambda)}{100\lambda}\right)
\end{aligned}$$

by taking $r = \frac{t\sigma(1-\lambda)}{100\lambda}$.

2 Average-case lower bounds

Let $f : B^n \to B$ be a boolean function. A formula is built from and, or, not with fan-in 2. (We work with formulas, not circuits.) Let $L(F)$ be the number of leaves of a formula F. Let $L(f) = \min_{F \text{ computes } f} L(F)$. The depth is the depth of the formula tree. We know that $\mathrm{depth}(f) = O(\lg L(f))$.

We can consider worst-case lower bounds and average-case lower bounds. For worst-case, we need to show every formula computing it exactly is large. Any formula that approximates the formula must be of large size. We want to show that this is true even if it agrees with just $\frac{1}{2} + \varepsilon$ proportion of the input; ideally ε is exponentially small.

Applications include hardness vs. derandomization results of Wigderson. It's related to deterministic extractors: a function with small correlation with any formula of a class gives an extractor for that class.

A history of results for worst-case.

- S, 1961, $\Omega(n^{1.5})$ for the parity function
- K, 1971, $\Omega(n^2)$

- A, 1987 gave a general method for lower bounds larger than $\Omega(n^2)$. $\Omega(n^{2.5})$ for "Andrev's function."
- Hastad, 1998, $\Omega(n^{3-o(1)})$. This is the best that can be proved for Andrev's function. (There is now a slight improvement in the $o(1)$.)

For the average-case,

- San, 2010: showed cn for $\frac{1}{2}+2^{-\Omega(n)}$. His method can be pushed down to $\Omega(n^{1.5-\varepsilon})$ with $\frac{1}{2}+2^{-n^{\varepsilon}}$.
- Rei, 2011: $\Omega(n^{2-o(1)})$ for parity, $\frac{1}{2}+2^{-n^{\varepsilon}}$. This follows from works in quantum computing. A formula of size n has a good approximation by a polynomial of degree $\sqrt{n}$.
- $\Omega(n^{2.499})$ for Andrev's function, $\frac{1}{2}+2^{-n^{\varepsilon}}$.
- $\Omega(n^{3-o(1)})$ with $\frac{1}{2}+\frac{1}{\text{poly}(n)}$ for a function in NP. They had nice methods.
- Combining methods from the 2 results above, $\Omega(n^{2.999})$ for $\frac{1}{2}+2^{-n^{\varepsilon}}$. (Actually we get $\frac{n^{3-o(1)}}{r^2}$ for $\frac{1}{2}+2^{-r}$.) It's a function related to Andrev's function.

Use the method of random restrictions. Given a formula of size $L(F)$, WLOG x_n appears in $L(F)/n$ n times. Fix its value randomly. The formula shrinks by this number of leaves, but something more happens. For an $\wedge$ gate with $x_n = 0$ or $\vee$ with $x_n = 1$. You actually remove $\frac{1.5L(F)}{n}$ leaves.

We know that

$$L(F') \le L(F) - \frac{1.5}{n}L(F) = \left(1-\frac{1.5}{n}\right)L(F) \le \left(1-\frac{1}{n}\right)^{1.5}L(F) \le \left(\frac{n-1}{n}\right)^{1.5}L(F).$$

This gives $L(F) \le n^{1.5}$ by induction, since the hypothesis is $(n-1)^{1.5} \le L(F')$. Note $L(F')$ is a parity function as well, for one variable less.

We obtain the constant 1.5 because by expectation when we fix a variable we remove 1.5 leaves.

Is 1.5 optimal? Of course not. People kept improving this constant.

Hastad proved that the shrinkage exponent is 2: you can do more or less the same argument with 2. You have to restrict by a lot of variables—most of the variables. The best theorem we know is the following.

Theorem 2.1 (Avi Shetl)**:** For all f,

$$\mathbb{E}_{\rho\in R_p}(f|_\rho) = O(\rho^2 L(f) + p\sqrt{L(f)}).$$

Here, with probability p we leave a variable as it is, with probability $1-p$ we restrict it to 0 or 1. To get better results, apply with p small. The last term is small in all applications.

In the original results of Hastad there was a polylog factor.

Why does this imply $2-o(1)$? Take $p = \frac{\text{poly}\log n}{n}$. It's tight for the XOR function.

We know there exists $h : \{0,1\}^{\lg n} \to \{0,1\}$ such that $L(h) \geq \Omega\left(\frac{n}{\lg n}\right)$ by a counting argument. Assume we know such a function. Let's try to use such a function to prove a bound of n^3.

Partition the variables into $\lg n$ groups of $\frac{n}{\lg n}$ variables. For $1 \leq i \leq \lg n$, let

$$\text{eq:csdm-3-17-15-1} y_i = \bigoplus_{j=1}^{\frac{n}{\lg n}} x_{i,j} \tag{1}$$

Then $h(y_1, \ldots, y_{\lg n})$ has formula size close to n^3.

Call the function $H(x_1, \ldots, x_n)$. Do a random restriction with $p = \frac{\text{polylog}\, n}{n}$. With high probability, we get a formula of size $\frac{1}{n^{2-o(1)}} L(H)$. With high probability in each group there is at least 1 variable not fixed, you can fix all the rest. The resulting function is as hard as the original function h.

Compare with Tribes.

Let $z_1, \ldots, z_n$ be additional variables. In order to describe h we need to describe a truth table of n bits. $z_1, \ldots, z_n$ is the truth table of h. Consider $A(\overline{z}, \overline{x})$, at least as hard as h.

For some h, the formula size is large. Consider the formula with this particular fixing of z_i. This formula is at least the formula size for that h, so we get $n^{3-o(1)}$.

In the average-case problem, we have to make sure each ingredient works with high probability.

First we have this function h of $\lg n$ bits. This h was hard to compute by a formula. It seems clear that we need h hard to approximate; we need a bound for the average-case formula size of h. There is a little problem: $f : B^{\lg n} \to B$. We want the lower bound to be with exponentially small correlation. Fix $r = n^\delta$. (r will be the number of groups.) We want the lower bound to be with correlation $\varepsilon = 2^{-r^\delta}$. We cannot achieve an exponentially small correlation with a function on $\lg n$ bits.

Thus we need to take h from a larger number of inputs, $h : B^r \to B$. A counting argument shows there are h hard to approximate. By a counting argument for almost all functions, if you want to approximate with 2^{cn}, you need an exponential size formula.

But now we cannot hardwire the whole truth table for h! Consider (1) with $1 \leq i \leq r$ and with $\frac{n}{r}$ instead of $\frac{n}{\lg n}$.

How do we find h? We want h to have formula size $L(h) \gg \frac{n}{\lg n}$. There is an easy way to do this: Take variables $z_1, \ldots, z_{4n}$; define h by these variables. To give the truth table we need 2^r bits. Instead take an error-correcting bits ECC: $B^{4n} \to B^{2^r}$. Any error-correcting code with distance $\geq \frac{1}{2} - \varepsilon$ will do.

Take the error-correcting code to be with distance $\frac{1}{2} - 2^{-\frac{r}{4}} = \gamma$.

Theorem 2.2: For almost all z (with probability $1-\varepsilon$), $\text{ECC}_z : B^r \to B$ is a function that cannot be approximated (better than $\frac{1}{2} + \varepsilon$) by a small formula ($\frac{n}{\text{polylog}\, n}$).

The proof uses the Johnson bound. In general you have 2^{2^r} functions, and you have 2^{4n} functions from ECC_z: every 2 functions are almost orthogonal to one another. The Johnson bound says that for any point you take, if you take a ball with radius $\frac{1}{2} - \sqrt{\frac{\gamma}{2}}$, it still contains a small number of codewords, polynomial in 2^r. Codewords are orthogonal, so they cannot correlate too well. The number of easy functions covers a small number of codewords. For almost all z we are OK. We needed $2^{4n} \gg 2^{\text{polylog}\, n}$, the number of z's is large compared to the formula. Each formula can correlate with some codewords but not that many. The set of functions that correlate well with it is about 2^r.

In Andrev's function, replace $z_1, \ldots, z_n$ by ECC_z.

The other component is the random restriction. When we restrict the circuit, we get shrinkage in expectation, and with probability $1-$(small constant). However we need shrinkage with high probability. When we do the random restriction, we restricted the variables; fix the values at random. Otherwise it could happen that only when shrinkage does not occur the formula approximates well the function.

If you do a random restriction, then we don't get restriction with high probability. Maybe in the formula only x_1, x_2 occur many times. In a random restriction we only restrict x_1 with probability p not exponentially small. If we see a variable that appears many times, we have to restrict it. Otherwise choose an input variable randomly and restrict it (and repeat).

Now shrinkage occurs with high probability. There are 2 ways to prove it. In the original paper, we proved it with exponent 1.5. In the second paper, they had 2.

First proof (exponet 1.5): Apply Azuma' inequality to get a concentration bound; get a shrinkage with high probability. Originally you can't apply it because the variable fixed could appear a lot.

Second proof (exponent 2): Look at the original formula, partition it into subformulas; for each subformula you have shrinkage with some probability; there are many of them so you can hope for a concentration bound. There are dependencies between the formulas. Remove all the variables that appear many times; there is a small amount of dependencies. Separate into groups that are completely independent. Apply Chernoff and get shrinkage with high probability.

There is still a small problem that turns out not to be that small. Our restriction is not a completely random restriction. We don't choose the input variables to restrict randomly. It's an adversarial choice of variables. Maybe the adversary decides to fix all variables in some blocks. A small number of blocks is okay because we cannot even approximate the function. What if the adversary kills all the blocks except 1? Have more variables to encode the blocks.

We found 2 ways to deal with this.

1. Don't change the definition of the function. It cannot happen too many times that the adversary choose the variables. The threshold was $\frac{L(F)}{n} n^{\alpha}$. It can only a small fraction of the times; otherwise the formula shrinks by too much. We remain with $\approx r \operatorname{polylog} n$ variables. Each step, choose some variables to fix, say $\frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \ldots$ In each of the blocks, only a small fraction are fixed by the adversary. Without loss of generality assume the adversary fixes the last ones. Let's not fix them, and just mark them. The ones that are not fixed in the next block, we'll let him fix.

2. Change the function. Take $x_1, \ldots, x_n$, produce $y_1, \ldots, y_r$. Let $y_1, \ldots, y_r = \text{Ext}(x_1, \ldots, x_n)$ where Ext is a bit-fixing extractor.

Why is there a problem? We had a function that took $x_1, \ldots x_n$ and outputted $y_1, \ldots, y_r$. But when we did a random restriction, sometimes many of the y_i's are fixed. We want a function that no matter how we fixed the x_i, $y_1, \ldots, y_r$ will not be fixed, and be close to the uniform distribution. This is exactly what a bit-fixing extractor does.

We can take a known extractor and apply it. We can use additional input bits to make it better.

A bit-fixing source is a source where k of n variables $x_1 \cdots x_n$ are unfixed. It is a deterministic function such that any bit-fixing source is close to being uniformly distributed. A bit-fixing source is exactly what happens when you fix most bits (worst-case). Everything else works. We have to show it's hard to approximate even when we compose it with a bit-fixing extractor. Use a union bound on restrictions. Redo the proof with these 3 ingredients.

Anup Rao: bit-fixing with exponentiall low error, good parameters. We have an additional seed though, so we can use a seeded extractor with n bits, which is much easier.

The function is in NC^1? (Extractor?)

Natural proofs: the reason we cannot prove lower bounds is because these methods, if they gave better lower bounds they also give better factoring algorithms. The limit stopped roughly when can compute pseudorandom functions. Circuits of larger size can compute PRF's, like middle bit of multiplication. Then natural argument kicks in.

3 Random walks that find perfect objects and the Lovász local lemma

At the heart of every local search algorithm is a directed graph on candidate solutions (states) such that every unsatisfactory state has at least one outgoing arc. In stochastic local search the hope is that a random walk will reach a satisfactory state (sink) quickly. We give a general algorithmic local lemma by establishing a sufficient condition for this to be true. Our work is inspired by Moser's entropic method proof of the Lovász Local Lemma (LLL) for satisfiability and completely bypasses the Probabilistic Method formulation of the LLL. Similarly to Moser's argument, the key point is that the inevitability of reaching a sink is established by bounding the entropy of the walk as a function of time.

Speaker: Dimitris Achlioptas

4 Tractability as Compressibility

4.1 Introduction

Given a collection of constraints over a collection of variables consider the following generic constraint satisfaction algorithm: start with a random assignment of values to the variables; while violated constraints exist, select a random such constraint and address its violation by assigning fresh random values to its underlying variables. We will prove that this process terminates relatively quickly if the following is true: the amount of information (bits) needed to encode the newly violated constraints after each step is strictly less than the amount of randomness (bits) that will be consumed to address their violation later on.

Speaker: Dimitris Achlioptas

Given Ω and subsets (flaws) $F = \{f_1, \dots, f_m\} \subseteq \mathcal{P}(\Omega)$ we want to find $\sigma \in \Omega$ avoiding all flaws in F.

- Specify a directed graph D on Ω such that
  - every flawed object has outdegree at least 1.
  - every flawless object has outdegree 0.
- Take a random walk.

For all $\sigma \in \Omega, f \ni \sigma$, define a nonempty set $A(f, \sigma) \subseteq \Omega$ of **actions**. Each $\tau \in A(f, \sigma)$ becomes an arc $\sigma \xrightarrow{f} \tau$ in a digraph D. (Think of the edges as colored by the flaws f.)

Definition 4.1: D is **atomic** if for every $\tau \in \Omega, f \in F$ there is at most one arc $\sigma \xrightarrow{f} \tau$.

I.e., there will not be two states that go to a state τ for the same reason (addressing flaw f).

The classic examples are:

Example 4.2: The k-CNF formula $F = \bigwedge_i c_i$ with n variables.

- $\Omega = \{0, 1\}^n$.
- $f_i = \{\sigma \in \Omega : \sigma \text{ violates } c_i\}$.
- $A(f_i, \sigma)$ are 2^k mutations of σ through $\text{var}(c_i)$.

Example 4.3: q-coloring graph $G(V, E)$ with n vertices.

- $\Omega = [q]^n$.
- $f^c_{u,v} = \{\sigma \in \Omega : \text{col}(u) = \text{col}(v) = c\}$.
- $A(f^c_{u,v}, \sigma) = \{\text{All } q \text{ mutations of } \sigma \text{ through } v\}$.

  But we can adapt the functions to the current state, $A(f^c_{u,v}, \sigma) = \{\text{Only conflict-free mutations}\}$.

The Lovasz Local Lemma gives $q > e\Delta$, but with the second set of actions, we solve the problem with $q \geq \Delta + 1$.

In the variable setting atomicity is implied by

- flaws are partial assignments (flattening—this is good because it refines accounting of conflict)
- actions modify the variables of the flaw addressed (locality—a genuine but natural restriction)

(In satisfiability there is only 1 way to violate the constraint.)

The three main quatities are

Definition 4.4: The **amenability** of f_i is $A_i := \min_{\sigma \in f_i} |A(\sigma, f_i)|$. It is a lower bound width of the repertoire available to address a flaw. (It will give a lower bound on entropy.)

Define the **potential causality** digraph to consist of edges $i \to j$ if there is any $\sigma \xrightarrow{f_i} \tau$ such that $f_j \in U(\tau) \backslash (U(\sigma) \backslash f_i)$. Think of it as a "pessimistic projection/shadow" of the graph: i causes j if there is a single transition where trying to address f_i we introduce flaw j; for $i = j$ we mean that addressing f_i doesn't fix it.

A digraph D is **transient** if for every $i \in [m]$,

$$\sum_{j \leftarrow i} \frac{1}{A_j} < \frac{1}{e}.$$

Theorem 4.5: Let $\sigma_1 \in \Omega$ be arbitrary. At time t, let f_i be a random flaw in $\sigma_l t$. Address f_i by taking a uniformly random action in $A(f_i, \sigma)$.

Let $T_0 = \ln |\Omega| + |U(\sigma_1)|$. If D is transient the probability that the walk does not reach a sink within $t = O(T_0 + s)$ steps is $< 2^{-s}$.

There is no need for a uniform sample to start off, and the running time depends on $|U(\sigma_1)|$, not $|F|$. ($|F|$ could be exponential; this is efficient as long as $|U(\sigma)|$ is polynomial.

There is an extension that involves weights.

4.2 Proof

We'll prove something slightly weaker than the theorem, under the assumption $\sum_{j \leftarrow i} \frac{1}{A_j} \le \frac{1}{16}$. ($\frac{1}{e}$ is tight: there are SAT instances which violate it minimally and are not satisfied.)

Fix an arbitrary permutation of the flaws; say flaw f_i has priority i and address the flaw with highest priority each time. (We will see how to generalize from this.) Now the digraph is uncolored. The algorithm becomes a uniform random walk on this graph. We'll work with the old causality graph; this doesn't cost us very much. (In principle we could define the causality graph for the new graph. However, the condition on "any" is stringent. It's unlikely to find a clean permutation that substantially sparsifies the graph.

We make another simplification. We trim down the number of actions to the next power of 2. Then we can do everything with bits, and easily talk about entropy. We only blow up each $\frac{1}{A_j}$ by at most 2, so the new sum is $\le \frac{1}{8}$.

Formally, let

$$\mathrm{Pot}(f_i) = \lfloor \lg A_{f_i} \rfloor \le b.$$

Then

$$\text{eq:kraft1} \quad \sum_{j \leftarrow i} 2^{-(\mathrm{Pot}(f_j) - 3)} \le 1. \tag{2}$$

This reminds us of the Kraft inequality.

Our overall strategy is the following: we consume $\mathrm{Pot}(f_i)$ bits of randomness to fix flaw f_i. Consider a ball of radius T around it with respect to the number of bits read, i.e., fix the amount of randomness that the walk will consume. "Read a tape" as you walk along to make your decisions. The first time the walk leaves the ball it will land in $[T, T + b)$. The algorithm will only stop if it finds a perfect object.

While this tree starts off dense, after a certain radius T_0, it dramatically starts to sparsify; most paths die off.

Claim 4.6: The number of paths (trajectories) that cross the ring at T is at most

$$2^{(1-\varepsilon)T+B}$$

where $\varepsilon = \frac{1}{b}$ and $B = \lg|\Omega| + |U(\sigma_1)|$.

The important thing is that neither ε, B rely on T; we have a uniform bound that the growth of the number of bad trajectories grows. Note every particular path has probability $\le 2^{-T}$. Take the union bound, the probability of a path crossing the ring, i.e., not arriving at the sink after T bits of randomness, is

$$2^{(1-\varepsilon)T+B}2^{-T} = 2^{-\varepsilon T+B} \overset{?}{\le} 2^{-\vartheta};$$

when T is large enough depending on ε, B, then we win. More precisely, we want $T \ge \frac{B+\vartheta}{\varepsilon}$.

For expository purposes, we establish progress per step (taking an edge) rather than per random bit consumed. Steps that consume a ton of bits are bad for this analysis. This gives up a constant factor of b in the running time.

Consier a trajectory

$$\sigma_1 \xrightarrow{w_1} \sigma_2 \cdots \sigma_{s-1} \xrightarrow{w_{s-1}} \sigma_s$$

where $s \ge \frac{T}{b}$. The first observation is that it suffices to count $\langle W = w_1, \cdots w_{s-1}, \sigma_s\rangle$. This is because of atomicity: we never have collisions, so that we can unambiguously find all intermediate states given the final state and the flaws addressed. "It is the song the algorithm sings."

This σ_s is independent of how long the algorithm is; it involves at most $\lg|\Omega|$ bits. The significant part is bound the number of bits needed to encode $w_1, \ldots, w_{s-1}$.

Naively it would depende on the log of the total number of flaws. But we exploit the causality happening; the internal structure makes string highly compressible.

For now, we add more information rather than compress. We have a more refined video of what's happening.

We define $B_0 = U(\sigma_1)$ (bag of all flaws that are present initially), and define B_i, M_i as follows. B_i is the flaws broken at step i. (B is for "broken.") They are sets of flaws. Take out the highest priority flaw, take an action, go to a new state. M_i is the stuff that's fixed collaterally (a flaw f_j that was fixed when addressing $f_i, i \ne j$.)

Recover the w_i by taking the highest priority removed from B_i. Actually we change the M_i: consider $m \in M_j$. It entered and exited the system but we never felt its presence. Dealing with higher priority things m got fixed, and it exits the system. It will have a matching set B_i where it entered; go back in time as little as possible to find someplace where it entered. Remove m from B_i and M_j. Then we can recover $w_1, \ldots, w_{s-1}$ from knowing B_i, M_i.

One by one we can remove all of M_i and we obtain subsets B_i^* of the original B_i. The content are the flaws which got introduced by the transition and caused the consumption of randomness further down. It suffices to encode $B_0^*, B_1^*, \ldots, B_{s-1}^*$. B_0^* is special: encode it by $\{0,1\}^{|U(\sigma_1)|}$. This gives the factor $|U(\sigma_1)|$ in B.

Now we need to encode $B_1^*, \dots, B_{s-1}^*$. We break this into 2 parts: the size $|B_i^*|$ of the sets, and the content of the sets. To encode this, write $1^{|B_1^*|}01^{|B_2^*|}0 \cdots 1^{|B_{s-1}^*|}$. This takes at most $2S$ bits, as $\sum_i |B_i^*| \le S$.

Another way to look at this is to start with all the B_i's empty; if something got addressed by an action at step j, go back in time as little as possible to find where it was introduced, and stick it in that B_i^*.

Assign a budget c_i bits to each flaw. (c_i will depend on the potentials.) Ask each flaw to create a prefix-free code it causes. Record the encodings of the flaws in B_i using the prefix-free code for B_i, in some other tape E'.

We know the first thing addressed from looking at B_0^*. The first action breaks $|B_1^*|$ things that are later addressed before time s. Since we know $|B_1^*|$, we can read $|B_1^*|$ codewords, according to the dictionary of flaws that can be introduced in fixing what was fixed, from the encoding E'.

Kraft's inequality[1] applied to (2) says that that we can create a prefix-free code for each flaw of length at most $\mathrm{Pot}(f_i) - 3$ bits. The total length of the encoding is

$$2S + \sum_{w \in W} (\mathrm{Pot}(w) - 3) \le 2S + (T + b) - 3ST + b - S \le -\frac{T}{b} = T\left(1 - \frac{1}{b}\right).$$

($\sum_{w \in W} \mathrm{Pot}(w)$ is total random bits used, which is $T + b$.) This is the promised ε! $2 + 1 = 3$ gives an entropic gain of 1 bit per step.

How to improve this? Rather than gain 1 bit per step, just gain ε bits per step. To get to $\frac{1}{e}$, you do joint coding. The -3 becomes -2; then -2 becomes $\frac{1}{e}$ when you do joint coding.

There is a practical algorithm where given an assignment, you select a violated clause, and flip a single variable. A natural thing to do is look at $e^{-\frac{1}{T}(M-B)}$ where T is the "temperature" (how much you care) (Gibbs sambling), and sample proportionally to this distribution. The results are reasonable. Is there a better functin that $M - B$: you get better results if you just ignore M: $e^{\frac{1}{T}B}$, you do better. We obliterated collateral makes: we just encode what you break!

This gives us to a different class of algorithms. Consider acyclic edge coloring. (Any cycle has ≥ 3 colors.) At each step, give a random color, subject to the condition that no 2 adjacent edges are the same color. If at some point a bichromatic cycle forms, erase all edges in the cycle. Describe a flaw as an edge is not colored. At every step remove one flaw, but potentially introduce a lot of flaws (erasing all edges in a cycle). There is no collateral make! You fix one flaw at a time. The only way something can get colored is if you color it. Then you don't need to know the future to get a good encoding. The encoder and decoder are in much better coordination; you can prove stronger results. The decoder can be aware of the probability distribution the encoder is facing, so can be more efficient.

Acyclic edge coloring: You are given a graph; the only thing you know is a bound on the degree. There can be long cycles. How many additional colors need for acyclic edge coloring? For just edge coloring, 2Δ is enough. In our case, 4Δ, we have a choice of 2Δ. But we might be closing a huge cycle. The length of the cycle has nothing to do with the edge condition, but the proof that it terminates works: it is not how many things you break. You can see the amortization directly: every single thing will be addressed in the future; we can

[1] `https://en.wikipedia.org/wiki/Kraft%27s_inequality`

charge randomness against it. It's a future investment in randomness. Encode the sequence of edges: describe Δ objects for each edge in the cycle: $l \lg \Delta$. This is what it will consume.

5

During last fifty years a strong machine learning theory has been developed. This theory includes: 1. The necessary and sufficient conditions for consistency of learning processes. 2. The bounds on the rate of convergence which in general cannot be improved. 3. The new inductive principle (SRM) which always achieves the smallest risk. 4. The effective algorithms, (such as SVM), that realize consistency property of SRM principle. It looked like general learning theory has been complied: it answered almost all standard questions that is asked in the statistical theory of inference. Meantime, the common observation was that human students require much less examples for training than learning machine. Why? The talk is an attempt to answer this question. The answer is that it is because the human students have an Intelligent Teacher and that Teacher-Student interactions are based not only on the brute force methods of function estimation from observations. Speed of learning also based on Teacher-Student interactions which have additional mechanisms that boost learning process. To learn from smaller number of observations learning machine has to use these mechanisms. In the talk I will introduce a model of learning that includes the so called Intelligent Teacher who during a training session supplies a Student with intelligent (privileged) information in contrast to the classical model where a student is given only outcomes y for events x. Based on additional privileged information x^* for event x two mechanisms of Teacher-Student interactions (special and general) are introduced: 1. The Special Mechanism: To control Student's concept of similarity between training examples. and 2. The General Mechanism: To transfer knowledge that can be obtained in space of privileged information to the desired space of decision rules. Both mechanisms can be considered as special forms of capacity control in the universally consistent SRM inductive principle. Privileged information exists for almost any inference problem and can make a big difference in speed of learning processes.

Speaker: Vladimir Vapnik

The main result of VC theory is that there are only 2 factors important for generalization: the percent of training errors ν_{train} and good functions as candidates for generalization. The capacity of the set of functions is measured by VCdim or VCent. Then with probability $1-\eta$,

$$\mathbb{P}(P_{\text{test}} \in \nu_{\text{train}} + [-1,1]O^*\left(\frac{\sqrt{VCdim - \ln \eta}}{\ell}\right)$$

VCdim gives necessary and sufficient conditions when you don't have a probability measure.

Why do human students require for learning much less examples than machines? The existing machine learning approach is based on a model of learning with trivial teacher, while human learning is based on a model of learning with intelligent teacher. The teacher is allowed to provide additional (privileged) information for the training examples.

"Better than a thousand days of diligent study is one day with Great Teacher."

Model 5.1 (LUPI model)**:** Given iid training triplets $(x_i, x_i^*, y_i), x_i \in X, x_i \in X^*, y_i \in$

$\{-1, 1\}$, x_i^* generated by intelligent teacher according to $p(x^*|x)$, find among given $f(x, \alpha), \alpha \in \Lambda$ the one $y = f(x, \alpha_*)$ that minimizes P_{test}.

Generalization of perceptron with large margin: minimize $R = (w, w)$ subject to $y_i[(w, z_i) + b] \geq 1, i \in [\ell]$. The solution (w_ℓ, b_ℓ) with probability $1 - \eta$ has the bound

$$P_{\text{test}} \nu_{\text{train}} + O^* \left(\sqrt{VCdim - \ln \eta} \right) \ell.$$

In the separable case using ℓ examples, estimate n parameters of w. In the non-separable case, estimate $n + \ell$ parameters (ℓ parameters of slack).

Let $\xi_i^0 = \xi^0(x_i)$ be the slack values.

A real teacher does not know the values of slacks. But he can supply students with a correcting space X^* and a set of functions $\xi(x^*, \delta), \delta \in D$ with VC dimension h^*. The teacher introduces both X^* and $\xi(x^*, \delta), \delta \in \Delta$to speed up convergence from $O\left(\frac{1}{\sqrt{\ell}}\right)$ to $\frac{1}{\ell}$.

SVM take 3: Kernels. $f(x, \alpha) = \text{sign}\left(\sum_{i=1}^{\ell} \alpha_i y_i K(x_i, x) + b\right)$. Solve a quadratic optimization problem.

SVM+: estimate slack functions. Define slack function $\xi_i = y_i[(w^*, z_i^*) + b^*]$ and minimize $R(w, b, w^*, b^*) = (w, w) + \gamma(w^*, w^*) + C \sum((w^*, z_i^*) + b^*)_+$ subject to $\sum \alpha_i y_i = 0, \sum y_i \beta_i = 0$, $\alpha_i \geq 0, 0 \leq \beta_i \leq C$.

The teacher tries to indicate similarity to control the VC-dimension.

Ex.

1. Advanced technical knowledge can act as privileged information.

   The advanced technical knowledge for DNA is the 3D structure.

2. Knowledge of future events. cf. Mackey-Glass time-series prediction. Training triplets are $x_t = x([t-3, t])$ and $x_t^* = x([t - \Delta - 1, t + \Delta + 2])$. In training on past events you can do this.

3. Holistic description. (For different features.) (Moral: classify types of a single digit!)

Knowledge transfer: can knowledge of a good rule in X^* help construct a good rule in X? Given (x_i, y_i), find $y = \text{sign}(f_\ell(x))$; given (x_i^*, y_i), find $y = \text{sign}(f_\ell^*(x^*))$.

The fundamental elements are the smallest number k of vectrs $u_i^* \in X^*$ for which $y = f^*(x^*) = \sum y_i \alpha_i^* K^*(x_i^*, x^*) + b \approx \sum^k \beta_s^* K^*(u_s^*, x^*) + b$. The $K^*(u_s^*, x^*)$ are frames.

Quadratic kernels are easy (eigenvalue problem); general kernels are hard.

Find m regression functions $\phi_k(x)$ given (x_i, z_i^k) where $z_i^k = K^*(u_k^*, x_i^*)$. The image in X of $K^*(u_k, x^*)$ is $\phi_k(x) = \int K^*(u_k^*, x^*) p(x^*|x)\, dx^*$, $k = 1, \ldots, m$.

Combine knowledge tranfer and similarity control.

SRM principle: construct a nested structure on the set of admissible functions and minimize bound over 2 terms

$$\nu_{\text{train}} + \left(\frac{VCdim}{\ell}\right)^{-\delta}.$$

Similarity control is in trying to minimize the δ. Minimize VCdim by knowledge transfer.

6 Kolmogorov width of discrete linear spaces: an approach to matrix rigidity

A square matrix V is called rigid if every matrix obtained by altering a small number of entries of V has sufciently high rank. While random matrices are rigid with high probability, no explicit constructions of rigid matrices are known to date. Obtaining such explicit matrices would have major implications in computational complexity theory. One approach to establishing rigidity of a matrix V is to come up with a property that is satised by any collection of vectors arising from a low-dimensional space, but is not satised by the rows of V even after alterations. In this work we propose such a candidate property that has the potential of establishing rigidity of combinatorial design matrices over the binary eld. Stated informally, we conjecture that under a suitable embedding of the Boolean cube into the Euclidian space, vectors arising from a low dimensional linear space modulo two always have somewhat small Kolmogorov width, i.e., admit a non-trivial simultaneous approximation by a low dimensional Euclidean space. This implies rigidity of combinatorial designs, as their rows do not admit such an approximation even after alterations. Our main technical contribution is a collection of results establishing weaker forms and special cases of the conjecture above. (Joint work with Alex Samorodnitsky and Ilya Shkredov).

Speaker: Sergey Yekhanin

6.1 Introduction

Definition 6.1: Let $V \in F^{n\times n}$ be a square matrix. We say V is (r,d)-**rigid** if for all $V' \in F^{n\times n}$ such that for every $i \in [n]$, $d_H(V_i, V_i') \leq d$, $\operatorname{rank}_F(V') \geq r$.

Valiant defined this in 1977. This very clean definition can be used to prove lower bounds.

Lemma 6.2: A $(\Omega(n), n^2)$-rigid $V : x \in F^n \mapsto Vx \in F^n$ does not have a linear circuit of size $O(n)$ and $\ln n$-depth.

We have the following results.

1. With high probability, a random matrix is $(0.99n, \Omega(n))$-rigid.
2. We have NO explicit constructions for 0.99: $(0.99, 0)$.
3. We have explicit constructions for $\left(r, \Omega\left(\frac{n}{r}\ln\frac{n}{r}\right)\right)$, $r \geq \ln^2 n$. (SSS94, Friedman). In the construction every minor has full rank; if you only change so many entries per row, there is a square minor that will not be touched. This is the "untouched minor" argument.

We've run against the "untouched minor barrier": there are matrices for which every minor has full rank but have circuits of linear size.

Valiant conjectured that geometric design matrices are rigid. The number of hyperplanes in $\mathbb{F}_q^{m+1}$ (points of $\mathbb{PF}_q^m$) is $n = \frac{q^{m+1}-1}{q-1} \sim q^m$. Each hyperplane has $w = \frac{q^m-1}{q-1} \sim q^{m-1}$.

Every 2 hyperplanes share $\lambda = \frac{q^{m-1}-1}{q-1} \sim q^{m-2}$ points. Thus hyperplanes form a (n, w, λ)-combinatorial design. Let $G_{m,q}$ be the matrix corresponding to the geometric design over $\mathbb{PF}_q^m$.

Conjecture 6.3 (Valiant 1977)**:** The matrices $G_{2,q}$ are $(\Omega(n), n^\varepsilon)$-rigid over $\mathbb{F}_2$.

This is not quite true; we have to avoid the cases with low rank.

$$\operatorname{rank}_{\mathbb{F}_2} G_{m,q} \begin{cases} \geq n-1, & q \neq 2^e \\ = (m+1)^e \sim n^{\frac{\ln(m+1)}{m}}, & q = 2^e \end{cases}$$

(Calculation: fixing m and letting $q \to \infty$, $n \sim q^m = 2^{em}$, $(m+1)^e = 2^{e\lg(m+1)} = n^{\frac{\ln(m+1)}{m}}$.) We hope to use combinatorial structure to prove rigidity. However, the combinatorial structure doesn't depend on q, while the rank does, so we'll need more information.

Our goal is to prove that $G_{m,q}$ are $(n^{2\varepsilon+\delta}, 1-2\varepsilon)$-rigid. This would already give lower bounds in communication complexity (separate something about polynomial hierarchy and PSPACE). Hamada's conjecture in design theory says the rank of a design marix is as high as a geometric design with the same parameters.

6.2 Methods

Consider the folloing embedding $\mathbb{F}_2^n \to \mathbb{R}^n$. for any $x \neq 0 \in \mathbb{F}_2^n$, consider the normalized version in $\mathbb{R}^n$, $\frac{x}{\|x\|_2} \in \mathbb{R}^n$. For $V \in \mathbb{F}_2^n$, define

$$A_2(V) = \max_{W \subseteq \mathbb{R}^n, \dim W = r} \min_{v \in V} \|\pi_W(v)\|^2$$

Find a subspace of dimension r that maximizes the smallest projection.

Lemma 6.4: $A_2(\mathbb{F}_2^n) \sim \frac{r}{n}$.

Proof.

1. Upper bound $A_2(\mathbb{F}_2^n) \leq \frac{r}{n}$: First note $A_2(\mathbb{F}_2^n) = A_2(\{e_1, \dots, e_n\})$. For $W \subseteq \mathbb{R}^n$ with basis $w_1, \dots, w_r$, we have

$$\sum_{i,j} \langle w_i, e_j \rangle = r$$

so there exists $j \in [n]$ such that $\sum_i \langle e_j, w_i \rangle \leq \frac{r}{n}$.

2. Lower bound $A_2(\mathbb{F}_2^n) \geq \frac{r}{n}$. Take the space Say $\|v\|_H = a$ and V intersects t blocks, $t \leq k$. We have

$$\sum_i \langle v, w_i \rangle = \sum_i \left(\frac{n}{\sqrt{n}} \frac{1}{\sqrt{\frac{a}{r}}} \right)^2 = \sum \left(\frac{r}{n} \right) a^2 = \frac{r}{n} \sum a^2 \geq \cdots$$

□

Lemma 6.5: Let V_m be a geometric design. Then $A_2(V_m) \leq \frac{r}{n}$.

This is a spectral argument.

One can also prove a stability result. Design matrices are as hard to approximate as all of the boolean cube.

Lemma 6.6: Let $V \subseteq \mathbb{F}_2^{n\times n}$ such that every row of V has weight w, $d \le w$, $V' \in \mathbb{F}_2^{n\times n}$ is a d-perturbation of V. Then

$$A_r(V') \le (\sqrt{A_r(V)} + \sqrt{\frac{d}{w}})^2.$$

A small perturbation of an inapproximable matrix is still inapproximable.

Lemma 6.7: $m = \frac{1}{\varepsilon}$, $d = n^{1-2\varepsilon}$, $\varepsilon = \omega(n^{1-\varepsilon\cdots})$ implies $A_r(V') \le \frac{r}{n}$.

Even if allow perturbation, allow high-dimensional approximation, still as hard to approximate as Boolean cube.

Conjecture 6.8: There exists $\alpha, \delta, \varepsilon = \frac{1}{m} > 0$ such that for all linear spaces $L \subseteq \mathbb{F}_2^n$ where $\dim L \le n^{2\varepsilon+\delta}$, for some $r = \omega(n^{1-\varepsilon\cdots})$, $A_r(L) \ge (1+\alpha)\frac{r}{n}$.

We can prove $o(n^\varepsilon \ln n)$.

Theorem 6.9: For all $L \subseteq \mathbb{F}_2^n$, $\dim L = k$, $A_1(L) \ge \frac{1}{k}$.

Compare $A_1(\mathbb{F}_2^n) = \frac{1}{n}$; there is a gap in approximability. We think of $K \approx n^{2\varepsilon+\delta}$. This gives the result to $O(n^\varepsilon)$.

Theorem 6.10: For all $L \subseteq \mathbb{F}_2^n$, $\dim L = K$, $A_{n^\tau}(l) \ge \Omega\left(\frac{\ln k}{k}\right)$.

We went to $\frac{1}{k}$ to $\frac{\ln k}{k}$. This gives the reult to $n^\varepsilon \ln n$.

(Relation to Kolmogorov width: Minimizing the distance is equivalent to maximizing the projection.) $K_r(V) = \sqrt{1 - A_r(V)}$.

Proof. For all $i \in [n]$, let $w_i = \min_{v\in L,\, i\in \mathrm{Supp}(V)} \mathrm{wt}_n(v)$. Let $\mu(L) = \sum_{i\in[n]} w_i^{-1}$. Take $x \in \mathbb{R}^n$ such that $(\ldots, \frac{1}{\sqrt{\mu w_i}}, \ldots) \in \mathbb{R}^n$; $\|x\|_2 = \sum_{i\in[n]} \frac{1}{\mu w_i} = \frac{1}{\mu}\mu = 1$. For all $v \in L$,

$$\langle x, v\rangle = \sum_{i\in\mathrm{Supp}(v)} \frac{1}{\sqrt{\mu w_i}}\frac{1}{\sqrt{w}} \ge w\frac{1}{\sqrt{\mu w}}\frac{1}{\sqrt{w}} \ge \frac{1}{\sqrt{\mu}} \overset{?}{\ge} \frac{1}{\sqrt{n}}.$$

□

Lemma 6.11: $\mu(L) \le k$.

Proof. Consider the hypergraph whose nodes are coordinates and whose edges are supports of elements in L. Let $\varphi = 0$ be the potential function. Color all nodes white. Pick a white node with the smallest w_i. E_i, $|E_i| = w_1$. Remember E_i, color all nodes in E_i black. φ-tracks black weight.

$A_r(L)$.

$L \subseteq \mathbb{F}_2^n$, $\dim L = k$. $A_k(L) \le \frac{C}{k}$, for all $v \in L$, $\mathrm{wt}(v) = \alpha \frac{n}{\alpha} \cdot (\frac{1}{\sqrt{n}}, \ldots)$ take care of vectors which has high rank. □

Definition 6.12: Let $F \subseteq \mathbb{F}_2^s$. $S \subseteq [n]$. Say S is (c, L)-**attractor** for L, if for all $x \in L$, $|\operatorname{Supp}(x) \cap S| \ge C\frac{|S|}{k}$.

Lemma 6.13: Let $L \subseteq \mathbb{F}_2^n$, $\dim L = k$, ,$[N] - \bigcup_{v \in L} \operatorname{Supp}(v)$. Assume for all $v \in L$, $\mathrm{wt}(v) \in [w, 2w]$, $k \ge 2^{5c+2}$, there exists (c, k)-attractor of size $\frac{N}{2^{4c}}$.

Restrict attention to slice. ...see paper

7

Setup: Given a group G and k high-entropy distributions X_i over G, show that $\prod_{i \le k} X_i$ is nearly uniform over G in L^∞. (This would give ε-close to uniform in statistical distance L^1.)

Consider X, Y distributions over $0.1|G|$. Is XY nearly uniform over $|G|$? Take $Y = G - X^{-1}$; then $1 \notin \operatorname{Supp}(XY)$.

Consider X, Y, Z uniform over $0.1|G|$. The answer depends on the group. Obstacles are:

1. $H \subset G$ a dense subgroup (constant fraction of group), $X, Y, Z \subseteq H$.
2. $G = \mathbb{Z}/p$ (abelian). Take $X = Y = Z = [0, 0.1p]$

Gowers, Babai, Nikolov, Pyber:

Theorem 7.1 (Mixing in quasirandom groups)**:** Let X, Y, Z be independent and uniform over $\ge 0.1|G|$ elements of G. Then the L^∞ bound is

$$|X|_2 |Y|_2 |Z|_2 \sqrt{|G|}/\sqrt{d} \le O(d^{-\frac{1}{2}})/|G|,$$

where G is the minimum dimension of a non-trivial representation of G.

If G is abelian, $d = 1$; if G is nonabelian simple, $d \ge \frac{1}{2}\sqrt{\ln |G|}$ (not far from tight for A_n). $\mathrm{SL}_2(\mathbb{F}_q)$ has $d \ge |G|^{\frac{1}{3}}$. Then $G = \mathrm{SL}_2(\mathbb{F}_q)$ gives XYZ is $\frac{1}{\mathrm{poly}(|G|)}$ close to uniform. (What is the best d? $\sqrt{|d|}$ is upper bound.)

What happens when you throw in dependencies? If AYA' is nearly uniform, and (A, A') is uniform over $\ge 0.1|G|^2$ elements, Y is independent and uniform over $0.1|G|$ elements of G? No. Pick any Y over $0.5|G|$. Given A, define A' as $G - Y^{-1}A^{-1}$, then $AYA^{-1} \ne 1$.

Theorem 7.2 (Interleaved mix)**:** For (A, A'), (B, B') uniform over $\ge 0.1|G|^2$ elements of G^2, $(A, A'), (B, B')$ independent,

$$\left\| ABAB' - \frac{1}{|G|}1 \right\|_\infty \le \frac{1}{|G|^{1+\Omega(1)}}.$$

This recovers the XYZ result. One proof works without representation theory but with Weil bound. Conjecture: there are similar bounds for all almost simple groups.

This generalizes the previous result:

Theorem 7.3 (Longer mix)**:** For A, B uniform over $\geq 0.1|G|^t$ elements,

$$\left\| \prod_i A_i B_i - \frac{1}{|G|} 1 \right\|_\infty \leq \frac{1}{|G|^{1+\Omega(t)}}.$$

7.1 Communication complexity

Alice gets A and Bob gives B. They want to tell $\prod_{i \leq t} A_i B_i = g$ from $\prod_{i \leq t} A_i B_i = h$. If G is abelian, the communication complexity is 2. If $G = \mathrm{SL}_2(\mathbb{F}_q)$, communication is $\Omega(t \ln |G|)$; this holds even for public-coin protocols with advantage for $\frac{1}{|G|^{ct}}$. (?) Reduction from IP gives $\Omega(t)$ lower bound.

7.2 Proof of interleaved mixing $ABA'B'$

Let $C(g) = U^{-1} g U$ be the uniform distribution over the conjugacy class of g.

Lemma 7.4 (Main lemma)**:** Let $G = \mathrm{SL}_2(\mathbb{F}_q)$. With probability $1 - \frac{1}{|G|^{\Omega(1)}}$ over $a, b \in G$, $|C(a)C(b) - U|_1 \leq \frac{1}{|G|^{\Omega(1)}}$.

Proof of interleaved mixing. Suppose for simplicity $(A, A'), (B, B')$ are iid uniform over $S \subseteq G^2$, $|S| = \alpha |G|^2$.

The main tool is Cauchy-Schwarz. By Bayes's rule.

$$\begin{aligned}
|ABA'B'(1) - \frac{1}{|G|}| &\leq |\mathbb{E}_{u,v,u',v':uvu'v'=1} S(u,u')S(v,v') - \alpha^2| \frac{1}{\alpha^2 |G|} \\
(*) &\leq \mathbb{E}_{v,v'} \mathbb{E}_{u,u':uvu'v'=1} (S(u,u') - \alpha) S(v,v') \\
&\leq \sqrt{\mathbb{E}_{v,v'} (\mathbb{E}_{u,u':uvu'v'=1} S(u,u') - \alpha^2) \sqrt{\alpha}} \\
(**) &\leq \mathbb{E}_{v,u,u',x,x':uvu'=xvx'} S(u,u') S(x,x') \\
&= \mathbb{E} S(u,u') S(ux, u'C(x)).
\end{aligned}$$

Where conjugacy classes come up: $v^{-1}x^{-1}uvu' = x'$. Pick a pair, take a step in the Cayley graph $(u,u') \mapsto (ux, u'C(x))$ hits like (taking 2 steps at a time) $(u,u') \mapsto (uxy, u'C(x)C(y))$. □

Most papers study $\mathrm{Supp}(C(a)C(b))$ but we need statistical bounds. They study worst-case a, b, which is insufficient.

Proof of lemma. Observation: $C(a)C(b) = C(C(a)C(b))$. Proof: $w^{-1}u^{-1}auww^{-1}v^{-1}bvw$.

Suffices to show $C(a)C(b)$ hits every class with the right probability.

All but $O(1)$ of $q + O(1)$ conjugacy classes have size $q^2 + \Theta(q)$. Picking a uniform element is close to picking a uniform class. There is an almost 1-1 correspondence between classes and $\mathbb{F}_q$ given by trace. We will show $|\mathrm{Tr} C(a)C(b) - U_q| \leq \frac{1}{q^{\Omega(1)}}$.

$$\mathrm{Tr} C(a)C(b) = \mathrm{Tr} a C(b).$$

Let the conjugator be $\left(\begin{smallmatrix} u_1 & u_2 \\ u_3 & u_4 \end{smallmatrix}\right)$. This is a polynomial in u_1, u_2, u_3, u_4 subject to $u_1u_4 - u_2u_3 = 1$. Substituting the value of u_4, we get $g(x, y, z)$. We need to show $|g(x, y, z) - U_q| \leq \frac{1}{q^{\Omega(1)}}$.

Use the Weil bound: for $f \in \mathbb{Z}[x, y, z]$ irreducible over any field extension and of low (bounded) degree, we have $\leq O(q^{-1.5})$. Prove for $q - O(1)$ values $s \in \mathbb{F}_q$, $g(x, y, z) - s$ is irreducible. This relies only on zero/nonzero coefficient pattern. □

7.3 Multiparty communication complexity

Lemma 7.5: Let $G = \mathrm{SL}_2(\mathbb{F}_q)$, $s \gg m$, $D_1, \ldots, D_s$ be independent distributions on G^m, D_i is pairwise independent, then

$$\left\| D_1 \cdots D_s - \frac{1}{|G|^m} \right\|_\infty \leq \frac{\varepsilon}{|G|^m}.$$

In multiparty communication complexity, each party has an input on their forehead. Consider 3 parties for simplicity. The interleaved product is $P(A, B, C) = A_1B_1C_1 \cdots A_tB_tC_t \in G$. We want to tell $P(A, B, C) = g$ from $P(A, B, C) = h$.

The trivial upper bound is $O(t \ln |G|)$. The reduction from generalized IP gives $\Omega(t)/2^k$ lower bound. The conjecture is that for any simple nonabelian group, $\Omega(t \ln |G|)/2^k$, and is hard even for $k > \lg(\text{input length})$. The second is interesting even if $|G| = O(1)$. This would be a breakthrough in communication complexity theory. Ran Raz: each matrix $n \times n$, entries in $\mathbb{F}_2$.

(Note the lower bound is not better than GIP, because $t \lg |G|$ is exactly the bound length. The function has better structure though.)

Theorem 7.6: For any $k = O(1)$, $\Omega(t, \lg |G|)$, and t large enough.

There is a nontrivial bound when k grows moderately.

Corollary: Leakage-resilient construction is secure even in only-computation leaks model. `http://www.ccs.neu.edu/home/viola/papers/leak.pdf`

The proof of lower bound relies on the interleaved group products theorem.

Proof. Define $f(A, B, C) = 1$ if $P(A, B, C) = g$, -1 if $P(A, B, C) = h$, and 0 otherwise.

Lemma 7.7 (BNS, CT, R, VW)**:** The correlation with c-bit protocols is the **box norm** $\leq 2^c|G|\mathbb{E}\left(\prod_{\alpha,\beta,\gamma \in \{0,1\}} f(A^\alpha, B^\beta, C^\gamma)\right)$ for $A^0, \ldots, C^1 \in G^t$ uniform.

Given that the 2^3 factors are all in $\{1, -1\}$, the expected number is $\mathbb{P}(\text{even number of } -1) - \mathbb{P}(\text{odd number of } -1)$.

Try to solve something harder! We show that the 2^3 tuple $(P(A^i, B^j, C^k))_{i,j,k \in \{0,1\}^3}$ is nearly uniform in G^8. Let $D(t)$ be the 2^3 tuple when $|A^0| = \cdots = t$. $D(t)$ is the componentwise product of s copies of $D(t/s)$. For any r, $D(r)$ is pairwise (3-wise) independent.

Lemma 7.8: If $D_1, \ldots, D_s$ are independent, each is pairwise independent, then $D = D_1 \cdots D_s$ is $\frac{\varepsilon}{|G|^m}$ close to uniform in L^∞.

It's enough to show $m = 3$. "Multiplying pairwise independent distributions flattens them." All we use on G is the $k = 2$ result ($ABA'B'$).

Lemma 7.9: For p, q pairwise independent over G^3, $|p|_\infty, \|q\|_\infty \leq \frac{1}{|G|^{2+\theta}}$,

$$\begin{aligned} \|pq\|_2^2 &\leq \frac{1}{|G|^3} + \frac{1}{|G|^{2+\theta+\Omega(1)}} \\ \|pqpq\|_\infty &\leq \frac{1}{|G|^3} + \frac{1}{|G|^{2+\theta+\Omega(1)}}. \end{aligned}$$

Improve the L^∞ bound if you take 4 copies.

Proof.

$$\begin{aligned} \|pq\|_2^2 &= \sum_{x_i y_i = z_i w_i} p(x_1, x_2, x_3) q(y_1, y_2, y_3) p(z_1, z_2, z_3) q(w_1, w_2, w_3) \\ &\leq n^4 \sum_{xy = zw} A(x, w) B(y, z) \end{aligned}$$

where $A(x, w) = \sum_a p(x_1 z, x_2, x) q(y_1, y_2, w)$ and similarly for $B(y, z)$. Apply the functional version of the $k = 2$ result. □

□

The communication complexity of deciding $ABA'B' = g, h$ for A_n is $\omega(1)$ for deterministic protocols. Use nontrivial structure of A_n. (See papers on conjugacy classes on A_n. Need conjugacy class result.)