Contents

1 Parallel computation

1.1 Randomized composable core-sets for distributed optimization, Vehab Mirrokni

Large-scale graph mining Algorithms, tools: Ranking, pairwise similarity, graph clustering, balanced partitioning, embedding.

Problems

1. diversity maximization: composable core-sets

2. clustering problems: mapping core-sets

3. submodular.coverage maximization: randomized composable core-sets.

Composable core-sets are useful. Inspired by processing big data. Extract and process a compact representation of data:

1. sampling: small subset

2. sketching

3.

Divide the input of size N into pieces $T_1, \ldots, T_m$, and send sets S_i back. Run the algorithm on the $S_1, \ldots, S_m$ to find the output set.

Key example in combinatorial geometry. Ex. farthese distance pair of points. An α-core set is S with $f(S) > \alpha f(P)$.

1. The convex hull is a 1-core set.

Composable core-sets: partition input into parts $T_1, \ldots, T_m$. In each part, select $S_i \subseteq T_i$. Let $S = \bigcup_i S_i$.

$f_k(S) := \max_{S' \subseteq S, |S'| \le k} f(S')$. ALG(T) is output of ALG on input set T, of size at most k. ALG is α-approximately composable coreset if for any T_i,

$$f(\bigcup(ALG(T_i))) \ge \alpha f_k(\bigcup T_i)$$

?: c-composable coreset of size k: chunks of size $\sqrt{nk}$.

Mp-reduce model.

1. Diversity maximization: maximize sum of pairwise distances diversity(S).

   Local search for diversity maximization (1 machine): Initialize S with arbitrarily k points which contains two fartest points. While there is a swap that improves diversity by $1 + \frac{\varepsilon}{n}$, then swap. This give 2-approximation.

   A local search algorithm computes a constant-factor composable coreset for diversity. Greedy algorithm computes a 3-composable core-set.

2. Distributed clustering

   balanced version: size constraint on each cluster.

   Approx ratio: small constant · best single machine ratio.

   Constant (2) rounds of MapReduce.

   Experiments: (Capacitated k-center)

   $10^7/10^8$ distances on US/world graph: Dividing into 300/1000 pieces, get 1.52/1.58 factor increase in OPT.

3. Submodular function optimization: $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$, or $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$. (Diminishing return.)

   Max-coverage problem is a special case. Applications: data summarization, data clustering, column selection, diversity maximization in search.

   ML applications: exemplar based clustering, active set selections, graph cuts.

   $f(S) = k$-median cost(empty set) - k-m

   There exists no better than $\frac{\ln k}{\sqrt{k}}$ approximate composable core-set for submodular maximization. How about just max-cover?

   None achieve constant factor in constant communication. Lose logarithmic in one of them.

   Randomization comes to rescue. α-approximate randomized composable core-set iff

   $$\mathbb{E}[f_k(\bigcup ALG(T_i))] \geq \alpha \mathbb{E}[f(\bigcup T_i)].$$

   There exists a class of $O(1)$-approximate randomized composable coresets for monotone and non-monotone submodular maximization.

   We show it for β-nice algorithms:

   (a) $x \in T \backslash ALG(T)$: ALG$(T)$ =ALG$(T \backslash \{x\})$. "Independence of irrelevant alternatives"

   (b) $\Delta(x, ALG(T)) \leq ?$.

   Greedy is 1-nice.

$\frac{1}{3}$ analysis.

1. $OPT' = OPT \cap (\bigcup_i S_i)$, $OPT'' = OPT \backslash OPT'$.

2. Linearize marginal contributions of elements in OPT: π permutation on elements of OPT. OPT^x = elements of OPT before x in π.

3. Writing in terms of Δ...

4.

5. $(1 - \frac{1}{e})\frac{1}{3}$.

Hard go beyond $\frac{1}{2}$ with size k¡ impossible get beyond $1 - \frac{1}{e}$. .58-approximate randomized composable core-set of size $4k$. If each machine outputs size k', get $\sqrt{\frac{k'}{k}}$-approximate randomized composable coreset.

Graph mining frameworks

1. iterative MapReduce

2. Iterative MapReduce and DHT (distributed hash table) service (flume): fast random access to shared memory

3. Pregel

4. ASYMP (asynchronous message-passing). New way do some graph algorithms. Better than current optimized algorithms and using less resources.

2 Reductions and equivalence in P

Hard problems in $O(N^2)$ time:

1. Given a set of points on the plane, are there 3 collinear points?

2. Sequence alignment: Given 2 strings what do they have in common? Several notions of similarity: edit distance, longest common subsequence. (Find a maximum length subsequence (can skip letter).) A DP algorithm does this in N^2 time. The larger the subsequence the closer the strings.

   Important in computational biology (genetics).

   The human genome has 3 billion base pairs. Quadratic runtime is too slow.

3. Graph problems: given a $O(n)$-edge graph, what is its diameter. Using APSP takes at least $O(n^2)$. Why do we need n^2 time for a single parameter?

   Even approximation algorithms are hard.

There are no $N^{1.5}$ time algorithms for many problems in dense graphs: diameter, radius, mdeian, second shartest path, shortest cycle...

Why are we stuck on may problems from different subareas of CS? Are we stuck because of the same reason?

Unconditional lower bounds are hard. Instead have a more relaxed approach. Look for hard problems. For NP it's k-SAT: k-SAT is hard because it's NP-complete. *If you can solve it fast you can solve many other problems fast.*

We want to emulate this for other problems.

Addressing the hardness of easy problems:

1. Identify key hard problems.

2. Reduce these to all other hard easy problems.

3. Form equivalence classes.

Understand the landscape of polynomial time.

CNF SAT is conjectured to be really hard. Beyond P vs. NP we have the following conjectures.

1. ETH: 3-SAT requires $2^{\delta n}$ time for some $\delta > 0$.

2. For every $\varepsilon > 0$ there is a k such that k-SAT cannot be solved in $2^{(1-\varepsilon)n}\operatorname{poly}(n)$ time.

Unlike P vs. NP, this actually talks about runtime.

Three more problems we can blame.

1. 3SUM: Given S of n integers are there $a, b, c \in S$ with $a + b + c = 0$?
2. Orthogonal vectors: Given a set S of n vectors for $d = O(\ln n)$ are there $u, v \in S$ with $u \cdot v = 0$?
3. All-pairs shortest paths (APSP): given a weighted graph find shortest distance between all pairs of nodes.

Solutions

1. $O(n^2)$-time algorithm is easy. There is an approximately $\frac{n^2}{\ln n}$ time algorithm for real numbers, $\frac{n^2}{(\ln n)^2}$ time algorithm for integers.

   Folklore: we can assume the integers in $\{-n^3, \dots, n^3\}$ requires $n^{2-o(1)}$ time.

2. $O(n^2 \ln n)$ times algorithm by checking every inner product. The best known algorithm by Williams et al. is $n^{2-\Theta(1/\ln(d/\ln n))}$. Tis required sophisticated techniques.

   Conjecture: OV on n vectors requires $n^{2-o(1)}$ time.

   Williams, 2004: SETH implies the OV conjecture.

3. APSP: state of the art is $\frac{n^3}{e^{\sqrt{\ln n}}}$. The APSP conjecture says APSP on n nodes and $O(\ln n)$ bit weights requires $n^{3-o(1)}$ time.

Reductios can't be arbitrarily polynomial-time or logspace reductions; they have to preserve runtimes.

We defined fine-grained reductions. We want to reduce problem A to B. A, B have runtimes $a(n), b(n)$. Improving $b(n)$ for B improves $a(n)$ for A.

Definition 2.1: For every $\varepsilon > 0$, $\exists \delta > 0$, $O(\alpha(n)^{1-\delta})$ time algorithm that transforms any A-instance of size n to B-instances of size $n_1, \dots, n_k$ so that

$$\sum_i b(n_i)^{1-\varepsilon} < \alpha(n)^{1-\delta}.$$

An improvement in the exponent for b improves the exponent for a: If B is in $O(b(n)^{1-\varepsilon})$ time, A is in $O(\alpha(n)^{1-\delta})$ time. Focus on exponents. We can build equivalences.

Some results.

1. APSP: We can prove equivalence with radius, median, betweenness, negative triangle, second shortest path, shortest cycle... of dense graphs.
2. 3SUM is equivalent to many problems in computational geometry: Three lines going through a point, polygonal containment... [GO95]

   A new class of problems where 3SUM implies hardness. At first they depend on real-number 3SUM. Based on hashing techniques, people got them to depend on integer version instead. Local alignment, etc.

3. Hardness for OV implies sparse graph diameter, local alignment, longest common substring, Frechet distance, edit distance...

These all imply hardness for dynamic problems. Ex. maintaining whether the number of strongly connected components of a graph is ≥ 3. have to recompute with linear time. If you can do better, then OV can be solved faster than we think.

Theorem 2.2: Fast OV implis SETH is false, W04.

Given a CNF... By the sparsification lemma we can assume the umber of clauses is linear in the number of variables. Split variables into sets of size $\approx \frac{n}{2}$. Enumerate through partial assignments of V_1, V_2¿ For $j = 1, 2$ and each partial assignment ϕ of V_i create $(m+2)$ length vector $v(j, \phi)$. The first 2 bits are just to distinguish V_1, V_2. For the last m bits, put 0 if ϕ satisfies the clause, 1 otherwise.

Claim: $v(1, \phi) \cdot v(2, \psi) = 0$ iff $\phi \odot \psi$ (taken together) is a SAT assignment $N = O(2^{n/2})$ vectors of dimension $O(n) = O(\ln N)$ gives an OV instance. So $O(N^{2-\delta})$ time implies SETH is false.

Later on we'll need gadgets and be careful with runtimes...

Another popular conjecture:

Conjecture 2.3: Boolen matrix multiplication (BMM), compute their boolean produce C where $C[i, j] = \bigvee_i (A[i, k] \wedge B[k, j])$. Any combinatorial algorithm for BMM requires $n^{3-o(1)}$ time.

BMM can be computed in $O(n^m)$ time, $m < 2.38$ using ordinary matrix mulitplications. The best known combinatorial techniques get a runtime of at best $n^3/(\ln n)^4$.

BMM conjecture consequences: reductions from BMM are typically used to show that fast matrix multiplication is probably required.

Implications

- a triangle in a grph cannot be found faster than $n^{3-o(1)}$ time by combo algorithm.
- radius of unweighted graphs requires $n^{3-o(1)}$ time via combinatorial techniques.
- max bipartite matching dynamically with nontrivial update times requires fast matrix multiplication.
- CFG parsing requires fast MM.

Problems can give reduction from boolean mult, don't have algorithm for MM. Maybe problem is harder, or we just don't have a nice reduction. MM maybe not right starting point.

Which conjectures are more believable? Besies SETH $\implies$ OV conjecture, no other reductions relating the conjectures are known. However, the decsion tree complexities of both 3SUM and APSP are low (not known for OV). Maybe OV is more believabe.

OV and APSP both admit better than log improvements over the naive runtime. 3SUM does not; maybe the 3SUM conjecture is more believable?

Natural problems they all reduce to: Matching triangles and triangle collection.

2PM: APSP and graph problems, OV and sparse graph problems.

3PM: Dynamic problems.

4PM: hardness for sequence problems.

Tomorrow: Matching triangles.

Approximability: For sparse graphs, even approximating some problems are hard based on OV. For dense graphs, we don't have good techniques.

Conjecture is false for approximating APSP, but we still don't know how to approximate other problems. Questions about weighted graphs: how to preserve the weights.

2.1 Hardness for graph problems

Important graph parameters

- **eccentricity**: $e(x) := \max_y d(x, y)$.
- **diameter**: $\max_x e(x)$.
- **radius**: $\min_x e(x)$.
- **median**: $\min_x \sum_y d(x, y)$.

The best algorithms we know compute all pairs shortest paths, $n^{3-o(1)}$ for dense graphs and n^2 for sparse graphs. Can we do better.

2.1.1 Dense graphs

APSP is equivalent to radius, median and many other graph problems under subcubic reductions. (It is a $n^{3-\delta}$ time reduction that takes instances of B of sizes n_i and produces instances of A such that $\sum_i n_i^{3-\varepsilon} < n^{3-\delta}$.) Equivalence of A, B means that a $O(n^{3-\varepsilon})$-time algorithm for A can be converted into a $O(n^{3-\delta})$-time algorithms for B and vice versa.

We can reduce the following to APSP

- distance product verification
- metricity
- radius
- mdeian
- Wiener index
- betweenness centrality
- replacement paths
- 2nd shortest path
- distance product (equivalence)

- negative triangle (equivalence)

We can easily reduce negative triangle to everything else on the picture, so we complete all the cycles to show equivalence.

Problem 2.4: Distance product: Given two matrices A, B, compute

$$(A * B)_{ij} = \min_k A_{ij} + B_{kj}.$$

Theorem 2.5: APSP in $T(n)$ time implies distance product in $T(n)$ time.

Distance product in $T(n)$ time implies APSP in $T(n) \lg n$ time.

Proof. Matrices correspond to graphs whose nodes correspond to the rows and columns, with the distance between i, j equal to the entry A_{ij}.

In the other direction, tge weighted adjacency matrix f a graph is $w(u, v)$ for edges (u, v) and ∞ for non-edges (u, v). A^k_{uv} is the weight of shortest path on $\leq k$ edges. Use repeated squaring to compute. A more complicated algorithm shaves off the $\lg n$ factor. □

Theorem 2.6 (VW10)**:** Distance product is subcubic equivalent to negative triangle.

Problem 2.7: Input: Graph G with integer edge weights.

Output: Yes if there exist $i, j, k \in V(G)$ such that

$$w(i, j) + w(j, k) + w(k, i) < 0,$$

and no otherwise.

1. First we reduce distance product to all pairs negative triangle: for all $j, i \in V(G)$, is there a k such that $w(i, k) + w(k, j) < -w(j, i)$.

   Create a 3 layer graph I, B, J using A, B. Now add edges from J to I. All pairs negative triangles solves: for every j, i in $J \times I$, is there k such that $A[i, k] + B[k, j] < -w(j, i)$.

   (Is $\min_k A_{ik} + B_{kj} < -w(j, i)$?) Use binary search. Reduce the interval for every pair simultaneously. Assumption on precision. Dependence only on number of nodes, not edge weights. (Now it has log of largest weight dependence.)

2. Reduce all pairs negative triangle to negative triangle.

   Split I, J, K into pieces of small size s.

   (a) Initialize C with O_n.

   (b) For each triple (I_x, J_y, K_z) in turn,

   - while (I_x, J_y, K_z), bas a negtive triangle, report negative triangle, and set $C_{a_x a_y} = 1$. Set $w(a_x, a_y) = \infty$. This ensures (a_x, a_y) doesn't appear in any more negative triangles.

     THe runtime is good: [(number of triples+triangles found)]·T(negative triangle in triple) = $((n/s)^3 + n^2)\cdot$ negative triangle(s). Set $s = n^{\frac{1}{3}}$ to get $n^{2+\frac{d}{3}}$. This is subcubic if $d < 3$.

Theorem 2.8: Negative triangle to radius of undirected graph.

Proof. WLOG assume

1. G is tripartite (create 3 copies).

2. G is a complete tripartite graph (add weights with large weight).

□

2.2 Dynamic problems

Input: an undirected graph G

Update: add or remove edges

Query: are s, t connected.

Trivial algorithm: $O(m)$ updates

Thorup: $O(\lg m(\lg\lg m)^3)$ amortized time per update.

Patrascu-Demaine: $\Omega(\lg m)$ necessary.

What about dynamic (directed reachability)?

The trivial algorithm takes $O(m)$ updates. Using fast matrix multiplication, we can achieve $O(n^{1.57})$. The best cell probe lower bound is still $\Omega(\lg m)$.

There are many examples with huge gaps between upper and lower bounds.

We give much higher lower bound via the "hardness in P" approach.

The 3-SUM conjecture implies polynomial lower bounds for many dynamic problems. After optimizing, we get $m^{\frac{1}{3}-o(1)}$ lower bounds for s, t-reachability, SSR, strongly connected components, maximum matching, connectivity with node updates, approximate diameter.

Maybe 3SUM is not the most appropriate problem.

The BMM conjecture implies tight lower bounds for combinatorial algorithms. THe lower bounds are $m^{1-o(1)}$ for combinatorial algorithms and match the trivial upper bounds. Any improvement will have to use fast matrix multiplication.

A conjecture about a dynamic problem. Given the second matrix vector by vector, find the matrix-vector multiplication one problem at a time. Online matrix vector multiplication conjecture: n^3 required.

Use the right conjecture: The APSP conjecture gives some quadratic lower bounds.

SETH implies very high lower bounds.

2.2.1 Single source reachability

If dynamic $\#SSR$ can be solved with $O(m^{1-\varepsilon})$ update and query timmes, then OVP can be solved in $O(n^{2-\varepsilon})$ time and SETH is false.

If you maintain all these updates and answer queries, you can figure out whether the OV instance was yes or no.

Amortized sublinear update/query time gives $O(nd)$ queries in subquadratic time...

1. Add edge $u_j \to b_i$ iff $b_i[j] = 1$. This encodes the vectors and stays static.

2. Encode the second list of vectors in a dynamic way. For each one we have a stage. For each a_i, dd $s \to u_j$ iff $a_i[j] = 1$. Ask $\#SSR(s)$. Observation: s cannot reach b if a_i, b are orthogonal. If $< n + (1s$ in $a_i)$ then output yes.

3. Remove edges and move on to next a_i.

$O(nd)$ updates, $m = O(nd)$ edges.

With additional gadgets, get loewr bounds for SCC, undirected connectivty with node updates.

Dynamic diameter

Problem 2.9: Input: undirected grph G.

Update: add or remove edges.

Query: what is the diameter of G¿

Naive: $O(mn)$ per update.

Better: Amortized $O(n^2)$.

Theorem 2.10: 1.3-approximation for diameter of sparse graph under edge update with amortized $O(m^{2-\varepsilon})$ updates refutes SETH.

It does use the usual orthogonal vectors reduction!

Proof. Reduce from "Three orthogonal vectors".

Subcubic implies subquadrtic.

Have a's, u's, v's, b's. Add $u'_j \to b_i$ iff $b_i[j] = 1$. The left and right sides are static, encoding A, B. The middle part is dynamic. Go over the the vectors in C. For each c_i,

1. Add $u_j \to u'_j$ iff $c_i[j] = 1$.

2. Ask diameter query. Observation: The distance from a to b is more than 3 iff a, b, c_i re an orthogonal triple.

3. Remove edges and move to next c_i.

□

We gave very high lower bounds for fundamental problems. After identifying the conjecture, the proofs are simple. There are many interesting open questions.

1. Lower bound for decremental reachability. $O(mn^{\frac{9}{10}})$ total update time.

2. Explain gaps between randomized and deterministic upper bounds. (Deterministic conjectures might be needed.)

Tomorrow: lower bounds with much better guarantees: if at least one of APSP, 3SUM, SETH is true, then SSR requires linear updates.

2.3 Quadratic hardness for sequence problems

1. Problems:
    - Discrete Frechet distance
    - Edit distance and LCS
    - Dynamic time warping
2. Bird's eye view on upper bounds
    - dynamic programming, quadratic time
    -
3. Recent conditional quadratic lower bounds (assuming SETH, etc.).

Frechet distance is "dog-walking distance". It is the smallest length leash that enables dog-walking along two routes (paths in the plane). Formally,

$$D_{Fr}(P,Q) = \min_{f,g \in F} \max_{t \in [0,1]} \left\| P(f(t)) - Q(g(t)) \right\|.$$

where F is the set of monotone $[0,1] \to [0,1]$ and $P, Q : [0,1] \to \mathbb{R}^2$.

In the discrete version, $F = \{f : [0,1] \to [n]\}$ and $P, Q : [n] \to \mathbb{R}^2$.

This problem has an easy dynamic programming solution: at each time step, there are 3 possibilities; at least one of the dog and owner jumps. Let $A[i,j]$ be the distance between $P([1,i])$ and $Q([1,j])$; we have

$$A[i,j] = \max[\left\| P(i) - Q(j) \right\|, \min(A[i-1,j-1], A[i,j-1], A[i,j-1])].$$

Can be improved to $O(n^2 \lg\lg n / \lg n)$. There are many algorithms for special cases and variants.

Definition 2.11: The edit distance (Levenshtein distance) "edit" between x, y is the minimum number of symbol insertions, deletions, or substitutions needed to transform x into y. Variants $edit'$: insertions of deletions. $edit' = 2n - 2LCS$.

Edit distance can be computed in $O(n^2)$ using a dynamic programming algorithm. This can be improved to $O\left(\frac{n^2}{\lg n}\right)$. There are better lgorithms for special cases. This is an important problem in computational biology.

There is an $(\lg n)^{O\left(\frac{1}{\varepsilon}\right)}$-approximation algorithm in $O(n^{1+\varepsilon})$ time.

Definition 2.12: Dynamic time warping between x, y

$$A[i,j] = \|x_i - y_j\| + \min(A[i-1,j-1], A[i,j-1], A[i,j-1]),$$

$\mathrm{DTW}(x,y) = A[n,n]$.

This is how people align for speech recognition. It's sum instead of max in Frechet distance.

What do these problems have in common? They are widely used metrics, dynamic programming algorithms with essentially quadratic runing time, and we have idea if/how we can do any better.

Plausible explanation: the problems are SETH-hard.

We go through the Orthogonal Vectors Conjecture.

Theorem 2.13: Assuming OVC conjecture, there is no $n^{2-\Omega(1)}$ algorithm for Frechet, EDIT, LCS, DTW distances unless OVC fails.

The basic approach is to reduce OVP to a distance computation.

- Turn $A \in B^d$ into $|x| \leq nd^{O(1)}$, similarly for B.
- distance is small if there exist $a \in A, b \in B$, $\langle a, b\rangle = 0$, and large otherwise.

2.3.1 Hardness

1. Frechet distance

   Coordinate gadgets: if $a_i = 1$, then put a point slightly higher. For b_i, do the opposite below.

   Vector gadgets: connect the points for the coordinates, and alternate between left and right positions.

   Add 2 points, one at the beginning of each. Concatenate the the first coordinate.

   If the original vectors are orthogonal, the Frechet distance is ≤ 1, by simultaneous jumps.

   If the vectors are not orthogonal, the Frechet distance is large. We must do a simultaneous jumps each time. When both coordinates are 1, we get $F \geq 1.01$.

   For the final curve, for one vector, concatente all vector gadgets with nodes at the beginning and end. For the other vector, concatenate directly. Shift the graphs to the central position.

   If there exist orthogonal vectors, Frechet$(x, y) \leq 1$. Traverse the first curve until the blue vector (the vector with orthogonal counterpart). Traverse the second curve until we are at the beginning of the blue vector. Traverse the blue vectors in parallel. Traverse the second curve until the end, then the first curve until the end.

   If Frechet$(x, y) < 1.01$, there exist othogonal vectors. Can't jump from beginning to end. At beginning of 2 vector gadgets. The corresponding vectors must be orthogonal.

2. Pattern matching with respect to edit distance: easier but has all the main ideas.

   Given 2 sequences, we want to find a substring from the y, of the same size, such that pattern$(x, y) = \min_{y'} edit(x, y')$.

First part: assume vector gadgets and prove hardness. Given $a_i \in A$, α_i, given $b_j \in B$, β_j, if a_i, b_j are orthogonal then $\text{edit}(\alpha_i, \beta_j) = S$, otherwise $= L > S$. Crucial that L does not depend on vectors.

Let t be vector gadget for vector consisting of 1s. intercalate‘$\$'\alpha$. intercalate ‘\$‘\$ replicate $(n-1)t++\text{“\$”}++\mathit{intercalate}\text{‘\$}'\beta++\text{“\$”}++\mathit{replicate}(n-1)t$

YES: there exists an orthogonal pair. Align them, we get $\leq S + (n-1)L$.

NO: Claim 1: optimal subsequence from the lower sequence contains n vector gadgets.

Claim 2: vector gadgets aligned one by one between upper sequence and subsequence. Matching \$ decreases cost.

Get $nL > S + (n-1)L$.

Second part: show vector gadget construction.

Coordinate gadgets: $a_i b_i = 1$ iff edit distance is large.

If $a_i = 1$ then 0001 else 0111. If $b_i = 0$ then 0011 else 1111.

Introduce $d^{O(1)}$ 2's in between the $CG_1(a_j)$'s, etc. Now $edit(\alpha', \beta') = d + 2\langle a, b\rangle$. Bad: edit distance depends on the inner product. Pattern matching for Hamming distance is easy. (Fast fourier transform.)

Instead, $\alpha = 4...4\alpha''3..3\alpha'4..4$, $\beta = 3..3\beta'3..3$. $\text{edit}(\alpha, \beta) = \min(d + 2\langle a, b\rangle, d+1)$. Two ways to match: $\alpha'' \leftrightarrow \beta'$ or $\alpha' \leftrightarrow \beta'$.

Etc., etc.

Gap too small for interesting approximability hardness.

3. Hardness for weighted LCS. Unweighted: write in unary.

   (Omitted.)

Open: hardness of approx for edit, LCS.

2.4 Conclusion

Take a problem X in P, say in $O(n^2)$ time, and prove that X probably cannot be solved in $O(n^{2-\varepsilon})$ time, emulating NP-hardness. We use 3SUM, APSP, and OVC (implied by SETH). Before P was a mess. Now the picture is a directed graph, we understand it better. “SETH-hard”: improving any of them would improve all of them. We still have unclassified problems.

What's next for hardness in P? Gain a better understanding of the conjectures. Find more reasons to believe the conjectures.

If 3SUM is in $n^{1.9}$ time then... If CNF-SAT is in 1.9^n time then...

Replace conjectures with more plausible ones.

Or find more reasons to disbelieve the conjectures.

Interesting progress on 3SUM tomorrow.

Find connections between the conjectures: maybe all these $O(n^2)$ problems are equivalent subquadratic reductions. (OVP, 3SUM, APSP)

Find barriers for relating them.

Classify more problems: if we're stuck with a bound we should know why.

To classify all of P, new conjectures might be needed. Maximum matching, linear programming.

Can we write a small LP for orthogonal vectors or 3SUM?

New class: k-clique. Best combinatorial algorithm does $O(n^k)$. Can get $O(n^{\omega k/3 = .79k})$. (Nesetril)

k-clique based lower bounds: CFG parsing. Can you derive the string from the grammar?

DP gives n^3. Valiant's parser gives $O(n^\omega)$ by matrix multiplication. Faster algorithms imply faster k-clique.

Valiant's parser is not practical because of overhead from matrix multiplication. A combinatorial parser.

RNA folding. Input: sequence $\{A, C, G, T\}^n$. Maximum number of arcs you can draw between matching pairs without crossings. Best algorithms are $O(n^3)$.

Why believe the new k-clique conjecture?

Williams 04: faster k-clique implies faster MAX-CUT.

Best exact algorithms for MAX-CUT.

Relationship between exponential time problems and problems in P.

Take Max-Cut. Can we do with other NP-hard problems?

What about tight reductions within NP-hard prblems? CNF-SAT, Max-Cut, travelling salesman, set cover, steiner tree. Faster for ST implies SC.

Fixed parameter tractability in P. In parameterized complexity: solve NP hard problems in $f(k)n^c$ time on inputs of size n and some natural parameter k. Study fixed parameter tractable algorithms for problems already in P.

Fixed parameter subquadratic: diameter on sparse graphs. No subquadratic algorithm under SETH. k is the treewidth of G.

Upper bound $2^{O(k \lg k)} n^{1+o(1)}$. SETH implies lower bound $2^{o(k)} n^{2-\varepsilon}$. Dependence on k nearly tight.

Hardness of approximation. "Even more relevant lower bounds."

Best approximation for edit distance in subquadratic time? In practice, approximation is often good enough.

Near-linear time polylog approximation is known. Can we get constant or $1 + \varepsilon$?

PCP theorem in P?

Barriers for shaving more log factors, average case hardness, quantum algorithms, space complexity?

3 6-15-15

3.1 Distance oracles

We show we can attain stretch $2k - 1$ with size $n^{1+\frac{1}{k}}$ and $O(1)$ query time.

Core approach: for all u, store a subset $B(u) = \{v_1, \ldots, v_r\}$ and $d(u, w)$.

If we query u, v, find a good node $w \in B(u) \cap B(v)$, and return $d(u, w) + d(w, v)$.

What does the Thorup-Zwick distance oracle look like? Hierarchy of k subsets of nodes. Look at a small ball around v, at vertices in A_1. Look at a larger ball and take nodes from A_2, a sparser subset.

For every level, store the closest node to v in that level, the pivot $p_i(v)$.

One of these pivots will be in bunches of both u, v and give good stretch. We show how to find the good pivot faster.

We want $O(n^{1+\frac{1}{k}})$ size; we cannot even store k pivots per node. We show that a constant number of sets A_i and pivots is sufficient. We can settle for a constant number of subsets and pivots. Instead of searching for a node belonging to both bunches, store some of the distances between the pivot. Go from s to a pivot, from pivot to pivot, and then to t.

Define $A_{\frac{k}{2}}$ by taking every node independently at random with probability $n^{-\frac{1}{2}}$. Store distances $A_{\frac{k}{2}} \times A_{\frac{k}{2}}$. and $(v, p_{k/2}(v))$.

Let $\Delta = d(s,t)$. If the distance between s and $p_{k/2}(s)$ is at most $\frac{k}{2}\Delta$ and similarly for t, then the distance between the pivots is $\leq (k+1)\Delta$.

If $d(s,t) \geq \frac{d(s,p_{k/2}(s))}{k/2}$, then the distance oracle is correct.

Now let $A_{\frac{k}{4}}$ be a random subset of size $n^{\frac{3}{4}}$. We can only store some of the distances. For every $w \in A_{\frac{k}{4}}$, store $B(w, d(...))$ If $d(s, p_{k/2}(s)) > 3d(s, p_{k/4}(s))$ then we're good. We'll have a shorter path between s, t.

CDo this a constant number of times. We are left with dealing with a small and sparse ball around s. It is easy to show a mechanism for dealing with "short distances."

Cluster nodes into small clusters such that

1. every node appears in a small number of clusters.
2. The important ball of s is completely contained in one cluster.
3. Every cluster is very sparse, so invoke a distance oracle with $O(n^{1+\frac{1}{k}})$ size, $O(1)$ squery time and $k' = \frac{k}{128}$ stretch.

The case we didn't cover: What if $d(s, p_{k/4}(s)) \gg d(s, p_{k/2}(s))$. Check whether in good case: Determine $d(s, p_i(t)) \geq i\Delta$ implies $d(s, p_i(t))/i \geq \Delta$, or $d(s,t) \leq 2d(s, p_i(t))$. Constant number of times: improve the upper bound or get a lower bound until know can return the upper bound.

Three components:

1. Constant number of sets A_i
2. mechanism for checking if $d(s,t) \geq d(s, p_i(s))/i$
3. mechanism for short distances.

Oracle cannot give path.

3.2 Consistency in planted bisection model

Connections to random k-SAT consistency. Physicists gave lots of conjectures.

Stochastic block model $G(n, p, q)$: two communities in a population, color randomly. There is probability p of an edge between nodes within the same community, and $q < p$ between communities. More friends in your group on average.

When can you recover both communities exactly (exactly recoverable) with probability tending to 1?

If $|p_n - q_n|$ is large enough then there are efficient polynomial time exact recovery algorithms. To have exact recovery, we need $p_n = \Omega\left(\frac{\ln n}{n}\right)$, otherwise we have vertices of degree 0.

Abbe, Bandeira, and Hall: recoverable if $a + b - \sqrt{ab} > 2$, $p_n = a \ln n/n, q_n = b \ln n/n$.

Us: Exact recovery iff $\mathbb{P}(Y_n \geq X_n) = o\left(\frac{1}{n}\right)$, $X_n = Bin(n/2, p_n)$ and $Y_n = Bin(n/2, q_n)$. If $\frac{1}{n}(\ln n)^3$, we get another characterization.

Proof: Assume $\mathbb{P}(Y_n \geq X_n) = \Omega\left(\frac{1}{n}\right)$. Every vertex has $\Omega\left(\frac{1}{n}\right)$ chance of being bad (most neighbors are other color). $\Omega(1)$ chance some vertex will be bad. We can't expect to get bad vertices right. Think of this as a "local-to-global" result in random graphs:

Theorem 3.1 (Erdős, Renyi)**:** $\mathcal{G}(n, p_n)$ is connected a.a.s. iff it has no isolated nodes a.a.s.

Our result can be read as: $\mathcal{G}(n, p_n, q_n)$ aas has no bad vertices iff minimal bisection is planted bisection.

An exact recovery algorithm: one can recover labels with $o(n)$ mistakes iff $\frac{n(p_n-q_n)^2}{p_n+q_n}$ (fraction you get wrong tends to 0). We take this as a black box.

Algorithm

1. Remove a vertex, apply accurate algorithm to the rest.
2. Put vertex back in
3. Color based on neighbors.

Actually remove some small constant fraction at a time for efficiency.

Properties of this algorithm: when is it getting things wrong?

Definition 3.2: A node is **mediocre** if it has a little more than half of its neighbrs (as a function of p, q) have the same color.

Only mediocre nodes are labelled wrongly. There are few $n^{\frac{1}{4}}$ mediocre nodes (calculate using binomial). If the graph is sparse, they form an independent set. If the graph is dense, every mediocre node has $2n^{\frac{1}{4}}$ more neighbors of its same color.

Corollary 3.3: Every node's true color except possible most mediocre nodes is the majority of the adjacent node's colors (?).

Part 2: when a node disagrees with the majority of its neighbors, recolor it.

This is almost linear time $n \operatorname{poly}\log(n)$.

3.3 Online matrix multiplication

Our new OMv conjecture is a good choice for proving condiional lower bounds on dynamic problems

3.3.1 Dynamic problems

Example: dynamic single source reachability (ss-Reach).

1. Preprocess a directed graph.
2. Give a sequence of updates (insert/delete edges) and queries (can you reach node n from start.

Two types of update time amortized lower bounds imply worst-case loewr bounds.

The naitve algorithm is to grow the BFS tree from s after each update. The update time is m, worst-case. Can we improve the update time from m to $m^{1-\varepsilon}$?

The only unconditional lower bounds give polylog time. So people use conditional lower bounds: assuming conjecture X, get polynomial lower bounds.

There are many popular conjectures:

1. Boolean matrix multiplication
2. 3SUM
3. Multiphase
4. Triangle
5. APSP
6. SETH

BMM, multiphase, 3SUM, and Triangle give hardness for SS-Reach. They give: you can't have better update/query time simultaneously. Each has drawbacks, because they only address combinatorial algorithms, are worst-case, not tight, etc.

Before our work, there are many conjectures, each with drawbacks.

3.3.2 OMv conjecture

Conjecture 3.4 (OMv conjecture)**:** Given boolen M, output $Mv_1, \ldots, Mv_n$ online. There is no $O(n^{3-\varepsilon})$ algorithm.

BMM conjecture implies (in some sense) OMv conjecture. Fast matrix multiplication does not work in an online setting.

There are fast $\widetilde{O}(n^2)$ for special matrices. General algorithm best is $O\left(\frac{n^3}{(\ln n)^2}\right)$. Unconditional lower bounds in related settings: algebraic circuit lower bounds, cell probe model.

Implications. Using OMv we get sronger, amortized lower bounds (many tight), and simpler reductions.

OMv and SETH are enough to prove all known strongest amortized lower bounds for dynamic problems.

Back to our first question: Can we improve update time m to $m^{1-\varepsilon}$.

3.3.3 Proving hardness using OMv

We first reduce to an intermediate problem: the edge query problem. Ask if there is an edge between 2 subsets L, R on either side of a bipartite graph.

Theorem 3.5: If there is an algorithm for edge query with preprocessing poly(n) time and $n^{3-\varepsilon}$ time, then refute OMv.

We reduce SS-Reach ($n^{1-\varepsilon}$ update, query $n^{2-\varepsilon}$) to edge query.

Make directed graph for ss-reach mathching the edge query graph. Update the edges out of s, into t, with the set, using n queries. A fast algorithm for sS-reach gets a fast algorithm for edge queriy. $O(n^2)n^{1-\varepsilon} + nn^{2-\varepsilon} = O(n^{3-\varepsilon})$.

3.3.4 Summary

All known conditional lower bounds for amortized update time fo dynamic problems; lower bounds from OMv is as high as from others. We get 15+ tight bounds.

3.3.5 Open problems

Minimized spanning tree: still $\Omega(n^.1)$ boudn for worst-case, ss-reach for decremental setting, maximum matching.

3.4 Clustered integer 3SUM via addtive combinatorics

Three big questions

1. Is there a subcubic algorithm for APSP?
2. $(\min, +)$-matrix multiplication: subcubic?
3. Special case: $(\min, +)$-convolution (matrix multiplication implies subquadratic)
4. 3SUM in subquadratic time?

The conjecture is no to all these questions.

We look for positive results: in what cases when . What kinds of techniques can give subquadratic/cubic algorithms?

Easy special cases:

1. small integer APSP using fast matrix multiplication.
2. $O(cn \lg n)$ time by FFT if elements in $[c]$
3. Bounded integer $3SUM^+$ by FFT if in $[cn]$.

Small-difference integer (min, +)-convolution: given $\{a_i\}, \{b_i\}$ with $|a_{i+1} - a_i|, |b_{i+1} - b_i| \leq c_i$.

Equivalent problem:

1. binary jumbled indexing (string algorithm). Given a binary string of length n, compute for all i, s_i be the min or max number of 1's over all length i substrings. This was posed by many people.

2. bounded int connected monotone $3SUM^+$ in 2D. Given $A, B, S \subseteq [cn]^2$ that form connected xy-monotone sequences, solve $3SUM^+$.

We give the first truly subquadratic algorithms for this group of problems, in randomized time $\widetilde{O}(n^{(9+\sqrt{177})=12}$.

A group of problem which expands set of problems where we know how to break the barrier.

Bounded integer monotone $3SUM^+$ in dD in $n^{2-\frac{2}{d+O(1)}}$ randomized time. Clustered integer $3SUM^+$.

Clustered integer $3SUM^+$ is the most general class where we have subquadratic algorithms.

Data structure version of thse problems. After preprocessing A, B, S in $\widetilde{O}(n^2)$ time, cna solve 3SUM for any subsets A', B', S' of A, B, S in $\widetilde{O}(n^{13/7})$ time, for arbitary input.

Besides FFT and fast matrix multiplications, we use additive combinatorics (previously almost nothing in algorithm design).

We give the 2D monotone in $3SUM^+$ problem.

1. Divide and conquer 1: divide $[n]^2$ into g^2 $\frac{n}{g} \times \mathfrak{g}ng$ grid cells. Recursively solve in $A^* + B^* = S^*$.

   How many recursive calls. The trivial upper bound is g^2; we get a quadratic algorithm.

   $$T(n) = O(g^2)T\left(\frac{n}{g}\right) + O(n)$$

   When can number of subproblems $\Omega(g^2)$. When A^*, B^*, S^* linear.

   But then we can subtract a linear function, make all values small, solve by FFT.

   If we are in the bad case, must the points be nearly collinear.

Theorem 3.6 (BSG)**:** Let A, B, S be sets of size N in an abelian group. If $|\{(a, b) \in A \times B : a + b \in S\}| = \Omega(\alpha N^2)$, then there exist $A' \subseteq A, B' \subseteq B$ such that $|A' + B'| = O(\left(\frac{1}{\alpha}\right)^5 N)$, $|A'|, |B'| = \Omega(\alpha N)$.

Frieman: $|A' + A'|$ small, then A' is close to collinear in some vague sense.

Simpler proof by Gowers, refined by others. Now it's 2 pages long.

We can't directly apply this theorem: there exists a subset, and they will not cover.

We need a stronger version that allows us to iteratively cover mre.

Corollary: there exist $A_i \subseteq A, B_i \subseteq B$, so that $RA +_S B\beta \bigcup_i (A_i \times B_i)$ has size $O(\alpha N^2)$, $|A_i + B_i| = O(\left(\frac{1}{\alpha}\right)^5 N)$, $k = O\left(\frac{1}{\alpha}\right)$.

1. Apply BSG to A^*, B^*, S^* of grid cells.

   $\widetilde{O}(g^2)$ randomized time

2. For each $(a^*, b^*) \in R$, recurse for points inside cells $a^*, b^*, a^* + b^*$. $O(\alpha g^2)T\left(\frac{n}{g}\right)$ time.

3. For $i = 1, \ldots, k$, compute...

Recurrence

$$T(n) = O(\alpha g^2)T(n/g) + \widetilde{O}(n + g^2) + O\left(\frac{1}{\alpha}\right) \ldots$$

polynomial dependence in α.

Other results are similar; in some cases we can't use recursion. Open: further improve exponents by improving α-dependencies in BSG. Could additive combinatorics help for kSUM, general 3SUM? Etc.

3.5 Sum of Squares lower bounds from pairwise independence

Constraint satisfaction problems: n boolean variables, m k-ary constraints.

Random assignment satisfies $\frac{|P^{-1}(1)|}{2^k}$ constraints. Can we do better? Yes for MAC-CUT, 2LIN, 3MAJ. In other cases, random assignment is optimal: $k - SAT^*$, etc. Approximation-resistant.

Understand as we gradually increase the number of satisfying assignments.

Can beat random on most predicates with sparsity $O(k^2/\ln k)$.

NP-hard to beat random for CSP(P) whenever P supports pairwise indep subgroup: Exists prob dist unif on subgrp inside $\mathbb{P}^{-1}(1) \geq \frac{1}{4}$

What about what space in middle?

CSP(P) is pairwise independent if prob dist μ on B^k.

Captures most of white space. Most predicates with sparsity $\Omega(k^2)$ are p.i. Assuming UGC, p.i. CSPs are approimation resistat.

$\Omega(n)$ round Serali Adams and basic sdp cannot beat random.

SoS missing.

Hierarchical strengthening of SDP.

SoS SDP highly successful convex relaxation for comb optimization

1. sparsest cut

2. sparse coding

3. attach on UG.

All knwn hard instances for weaker hier in poly time of UG, Bal sep, max cut.

SoS against CSP?

k-XOR's requires exponential SOS. Predicates supporting pairwise dindependent subgroup require sexponential SOS.

SOS needs exponential time to beat random for pairwise independent CSP's.

Previous proofs: reduction to resolution width lower bounds. algebraic structure crucial.

Our proof: local Gram Schmidt, does not depend on algebraic structure

At a high level, SoS SDP:

CSPs: optimize over B^n.

Solution: probability dist on B^n.

Search over e^n parameters.

Instead searhc over pseudo-distributions, satisfy only a subset of constraints of being a distributio!. Instead search over $n^{O(d)}$ parameters, d =degree/rounds.

Show instance I with M constraints. ... Deg $d = \varepsilon n$ SOS thinks I is satisfiable.

Instance I so that

1. val is $m||/2^k$.
2. $\widetilde{E}$ number I's constraints sat = m
3. pseudo-expe ≥ 0.

New: We need to show (3).

New basis for small degree $\leq d$ egree polys: Satisfy positivity and pseudo-orthogonality wrt $\widetilde{\mathbb{E}}$ and $\widetilde{[}\chi_i^2] \geq 0$.

Expand a poly in this basis.

1. Start from Fourier characters in some order
2. Run GS using $\widetilde{E}$ inner product.

$\widetilde{E}$ is valid inner product restricted to any d coordinates, corresponds to acutal expectation associated with a local prob dist.

But GS is highly sequential, long range dependent.

Rely on: Expansion in the hyper graph I.

Define distance measure on subsets of size $\leq d$.If S, T are far enoug, thn the local distributions on S, T given by $\widetilde{E}$ are independent.Orthogonalize only in a local neighborhood.

1. fourier characters in carefully chosen order.
2. Run step of GS using $\widetilde{\mathbb{E}}$ inner product.

Open: NP hardness.

3.6 Inapproximability of combo problems as LPs

Encode a problem into a polytope P, Q projecting to P.

Problems:

Say something about problem, not polytope. How you encode as polytope gives different ower bound.

Approx by scaling.

An optimizationproblem has set of feasibles, set of instances, oset of objective functions, $\mathcal{S}, \mathcal{F}$,val. Conider only maximization problems.

Ex. Vertex cover.

Formulation complexity. (C, S)-approximable LP with realiation vectors, affine functions, guarantees $\max(Ax \le b) \le C(f)$.

$$fc_+(\mathcal{P}, \mathcal{C}, \mathcal{S}) = \min\left\{r : \exists LP \text{ with } r\text{constraints capt } \mathcal{P}\right\}.$$

Factorization theorem.

1. (C, S)-approximate slack matrix M of $\mathcal{P}$ is $M(f, s) = C(f) - \mathrm{val}_F(S)$. Factorization is $M = TU + \mu\mathbb{1}$, T with r cols, $\mathrm{rank}_{LP} M$ =smallest r.

2. $fc = \mathrm{rank}_{LP} M$.

Optimal LP: Optimal LP's have same form. All info in problem encoded in...

Objective functions point towards negative octant. Reconstruct LP from factorization of smallest size.

Reduction mechanism: convert inapproximability results of problem to second problem. Given P_1, P_2,a reduction consissts of

1. map of feasible solutions $\gamma : S_1 \to \mathrm{conv}(S_2)$

2. map of instances $\beta : \mathcal{F}_1^{S_1} \to cone(\mathcal{F}_2^{S_2}) + \mathbb{R}$.

val guarantees.

When there is a reduction, cna bound cc of $\mathcal{P}_2$ by $\mathcal{P}_1$. β only act on set of solutions, γ only on instances, independent.

Hard gap between C, S. Measure hardness of gap: not possible LP with small number...

Factor soundness/completeness.

reduction from MaxCUT to vertex cover.

1. $\mathcal{S}$: all cuts (B^V)

2. $\mathcal{F}$ subgraphs of K_n

3. val objective functions $\mathrm{val}_H(s) = |\left\{\{v_i, v_j\} \in E(H) : s(v_i) \oplus s(v_j) = 1\right\}|$.

... fc is superpoly for approx guarantee of $\frac{1}{2} + \varepsilon$ for Max-CUT. SDP: ? conjectural.

All results independent of P vs. NP.

Open

1. Hardness of SDP base problem

2. Reduction for TSP: reductions aren't independent—solutions mapped according to instance.

3. Are there complete problems under this reduction?

3.7 Preserving statistical validity in adaptive data analysis

Analyze data and see whether the conclusions generalize to the whole dataset.

How do you analyze questions such as, does student nutrition affect academic performance?

Pick candidate foods; fit a linear function of 3 selected foods. Obtain linear correlation. Obtain a number which is the significance score: the probability of seeing such an outcome if there is no relationship. Here it is $< 10^{-5}$. You are now an expert on nutrition, can publish books!

Actually the datset you got was generated by 50 uncorrelated gaussians.

What went wrong?

Taking a new sample, you'll against

You're not supposed to use the same dataset to select variables and to test regression.

Freedman's Paradox: such paradoxes can distort the significance levels of conventional statistical tests.

Disconnect between ways they're analyzed and used in practice.

Put in fresh data, put through a procedure (hypothesis tests, regression, learning), give result and statistical guarantees (p-values, confidence intervals, prediction intervals).

Data analysis is adaptive:

1. exploratory data analysis
2. variable selection
3. hyper-parameter tuning
4. shared data - findings inform others

One analyst influences the analyses another analyst is doing.

Start with some fresh data set; analyze it, then perform another analysis based on the first output. The procedure is not independent of the data.

Is this a real problem, or just a theoretical issue?

There is a growing recognition that many published

"Why most published research findings are false," Ioannidis, 2005. "Irreproducible preclinical research exceeds 50%, resulting in approximately US$28B/year loss." Freedman, Cockburn, Simcoe 15.

How Science Goes Wrong, The Economist.

One significant cause is adaptive data analysis.

Rare and impractical to make decision s beforehand. Search for a combination athat yields statistical significance, and to report only what worked.

Simple, concrete problem which captures what arises very well.

Evaluating adaptive queries: Want $\phi_i : X \to [0, 1]$, want $|v_i - \mathbb{E}_{x\sim P}\phi_i(x)| \le \tau$ with probability $1 - \beta$. In context of ML, Kearns: Statistical query oracle.

How many samples will the algorithm need.

A lot of data analysis can be implemented in this model.

Answering non-adaptive SQ's: Given m non-adaptive query functions ϕ_i and n iid samples from P estimate $\mathbb{E}_{x\sim P}[\phi_i(x)]$. Use empirical mean; get Chernoff bound.

Answering adaptive SQ's: estimate expectations of m adaptively chosen functions. What if we use $\mathbb{E}_S[\phi_i]$? For some constant $\beta > 0$, $\tau > 0$, $n \geq m$.

We can do the safe choice, $n = O\left(\frac{m\ln(m/be)}{\tau^2}\right)$.

There exists an algoorithm that can answer t adaptively chosen SQ's with accuracy τ for

$$n_1 = O\left(\frac{\ln m}{\tau^2}\frac{\sqrt{\ln m \ln|X|}}{\tau^{1.5}}\right)$$

in times $O(n_1 \ln|X|)$.

Tool: differential privcy. Look at the outcome of algorithm that's adjacent.

A randomized algorithm A is (ε, δ)-differentially private if for any S, S' such that $\Delta(S, S') = 1$,

$$\mathbb{P}_A(A(S) \in Z) \leq e^{\varepsilon}\mathbb{P}_A(A(S') \in Z) + \delta.$$

Why DP? DP composes adaptively. Combine $(\varepsilon_i, \delta_i)$ DP algorithms gives $(\sum \varepsilon_i, \sum \delta_i)$. In fact composition of m ε-DP algorithms: for every $\delta > 0$ is $(\varepsilon\sqrt{2m\ln\left(\frac{1}{\delta}\right)}, \delta)$-DP. (Square root scaling)

If τ-DP algorithm A outputs $\phi : X \to [0,1]$ then

$$\mathbb{P}_{A,S\sim P^n}[|\mathbb{E}_S[A(S)] - \mathbb{E}_P[A(S)]| \geq \tau] \leq 6e^{-\tau^2 n}.$$

Back to queries. If DP A outputs function then $\mathbb{E}_S[A(S)] \approx \mathbb{E}_P[A(S)]$ with high problity.

Make analyst DP by answering queries with DP Answer queries for $\mathbb{E}_S[A(S)]$ with DP by counting queries.

Further developments.

1. $n = \Omega\left(\frac{\sqrt{m}}{\tau}\right)$ for polytime algorithms assuming OWF exists.

2. $n = O(\sqrt{m}\ln m/\tau^2)$.

3. Stronger tight generalizations for DP algoirthms

4. General adaptive setting and other techniques: description lenght and max-information.

5. Application: algorithms for reusing holdout set in ML

6. Application to maintaining accurate leaderboard in ML competitions.

TCS+ talk Aaron roth on YouTube.

3.8 Dimensionality reduction for k-means clustering

Dimensionality reduction: replace large high-D dataset with sketch.

Soluion on $\overline{A}$ should approximate original solution.

Use as preprocessing step for k-means and PCA, get simultaneous runtime improvement.

Review k-means clustering: Choose k clusters to minimize intra-cluster variance.

$$\sum_{i=1}^{n} \|a_i - \mu(C[a_i])\|_2^2 .$$

k-means ++ initialization with Lloyd's heuristic, k-means algorithm, $O(\ln k)$ approximation guaranteeed, typically performs much better. Use dimensionality reduction.

k-means clustering is low-rank approximation.

Replace every row of matrix with corresponding cluster mean. Only k unique rows, so rank k. error $\|A - C(A)\|_F$

$C(A)$ is actually a projection of A's columns onto a rank k subspace. $X_C X_C^T A = C(A)$.

Rewrite objective function:

$$\min_{\text{rank}(X)=k, X\in S} \left\|A - XX^T A\right\|_F^2 .$$

X is rank k orthonormal matrix and for k-means S is set of clustering indices.

We can solve anyproblem like $\min_{\text{rank}(X)=k, X\in S} \left\|A - XX^T A\right\|_F^2$. if for all rank k X, $\approx \left\|\widetilde{A} - \cdots\right\|$.

Projection-cost preserving sketch.

Specifically we want for all X, $\left\|\widetilde{A} - XX^T\widetilde{A}\right\|_F^2 + c = (1 \pm \varepsilon)\left\|A - XX^T A\right\|_F^2$. If we find γ-approximate solution, get get out $\gamma(1+\varepsilon)$.

This is similar to the coresets of Feldman, Schmidt, Sohler 2013.

k-means clustering is just constrained k-rank approximation. We cna construct projection-cose prserving sketch $\widetilde{A}$ that approximates the distance from A to any rank k subspce. Sronger guarantee than has been sought in prior work on approximate PCS via sketching.

Can reduce to $\approx k$ and give $1+\varepsilon$ projection-cost preservation. Useful for others? Go through different methods. Do analysis for SVD.

Reduce A by projecting onto top k/ε right SV's. Improves on prior work by ε. Robust to fairly robust SVD.

No constant factors on k/ε, typically fewer dimensions are required. Faster dim-reduction techniques.

Johnson-Lindenstrauss random projection matrix. Can get constant factor with $\ln k$ dimensions! First sketch with dimension sublinear in k.

Sketch is data oblivious: Useful in distributed applications.

1. Lowest communication distributed k-means.

2. Streaming PCA in single pass.

Standard sketches for low-rank approximations: span$(\widetilde{A})$ contains a good lw rank approximation for A, but we must return to .. to find it.

Is ε dependence necessary?

First single shot sampling techniques...

Analysis for SVD $A = U\Sigma V^T$. Partial SVD: $\|A - A_m\|_F^2 = \min_m \left\|A - XX^T A\right\|_F^2$. Mult by orthon doesn't change F norm. Clm: $\left\|A_{k/\varepsilon} - XX^T A\right\|_F^2 + c = (1 \pm \varepsilon)\left\|A - XX^T A\right\|_F^2$.

SPlit $A = A_{k/\varepsilon} + A_{\beta k/\varepsilon}$. Projection cost explained by sum of costs. Ignoring tail term is fine. $\backslash k/\varepsilon$ is at most ε times.

$$\left\| XX^T A_{\backslash k/\varepsilon} \right\|_F^2 \le \varepsilon \left\| A - A_k \right\|_F^2 .$$

Analysis is very worst-case.

All proofs have similar flavor: divide into 2 components, except we have to wrory about cross terms. We have some hope of approximating the tail.

We rely on standard sketching tools: approximate matrix multiplication, subspace embeddings.

Work super well in practice. Significantly improve running time.

3.9 Improved noisy population recovery, and reverse Bonami-Beckner inequality

Results from the survey don't reflect exact preferences because of noise in the answers. Can we recover the answers?

Model 3.7: Let π be unknown distribution in B^n with support k. Sample string $\sim \pi$, flip coordinate with prob **y¿** Recover π with L^∞ distance ε.

Motivations

1. mixtures of product binomial distributions. let π_i be product distributions over B^n with weights c_i. Can we learn π_i and c_i from the mixture.

Sample $x_j \sim \pi$. Generalize model of noisy pop recovery: flip $(x_j)_i$ with probability $\pi_{i,j}$. For all $i, j, \mu_{i,j} \le \mu < \frac{1}{2}$.

Population recovery: for all $\mu_{i,j} = \mu$.

Previous work: restriction access, lossy disribution, noisy distribution, list recovery, genreal mixture.

This work: noisy distribution can be recovered in time poly$(k^{\ln\ln k}, n, \frac{1}{\varepsilon})$. For any $\mu > 0$ there exists algorithm fo rnoisy recovery...

Let

$$T_\mu f = \mathbb{E}_{e = N_\mu(x)} f(x+e).$$

Necessary conditions: $\left\| T_\mu \pi_1 - T_\mu \pi_2 \right\|_1 \le \Delta(k, \pi_1, \pi_2)$, then any aglorithm needs $\Omega\left(\frac{1}{\Delta}\right)$ samples.

Sufficient condition! If $\left\| T_\mu f \right\|_1 \ge \Delta(k, \pi_1, \pi_2) \left\| f \right\|$, noisy distribution can be recovered in poly$(\frac{1}{\Delta}, n)$. Convex opt problem, sovle efficiently by MLE.

Information theoretic version. $|\operatorname{Supp}(f)| = k$. Then

$$\left\| T_\mu f \right\|_1 \ge k^{-O(\ln\ln k + \ln\frac{1}{\mu})} \left\| f \right\|_1$$

$\left\| T_\mu f \right\|_1 \ge \mu^{|S|} |\widehat{f}(S)|$.

Generalize! $g(x) = f(x) \mathbb{P}_{e \sim D_\mu}[x + e \in E]$, $\left\| T_\mu f \right\|_1 \ge \mu^{|S|} |\widehat{g}(S)|$.

Choose $E = \{y \in B^n : d(x_1, y) < d(x_i, y),\ \text{for all } x_i, d(x_1, x_i) \geq \ln k/\mu^2\}$. Informal claim: by truncating, g looks like dist on $B(x_1, \ln k/\mu^2)$.

$\mathrm{Supp}(f) \subseteq B(n, r)$ of size k. There exists $S \subseteq [n]$, $|S| \leq \ln k$ such that

$$|\widehat{f}(S)| \geq k^{-\ln(4r)} \left\|f\right\|_1 .$$

Proof idea: poly p of degree $\leq \ln k$, $p(x) = f(x)$ for all $x \in \mathrm{Supp}(f)$. $\sum_S |\widehat{p}(S)| \leq kr^{\ln k} \left\|f\right\|_1$.

recover $\mathrm{poly}(k^{\ln \ln k}, n, \frac{1}{\varepsilon})$ time.

Open

1. Recover in time $\mathrm{poly}(k, n, \frac{1}{\varepsilon})$.
2. Can we recover the noisy distribution without knowing μ?
3. Can we learn mixtures of product distributions in $\mathrm{poly}(2^k, n, \frac{1}{\varepsilon})$?

3.10 PRGs for spherical caps

Prob method gives $2\ln(n/\varepsilon)$. We get $O(\ln n + \ln\left(\frac{1}{\varepsilon}\right) \ln\ln\left(\frac{1}{\varepsilon}\right))$.

Similar for spherical Gaussian.

1. Iterated dimension reduction.

   Strong quantitative bounds for truncated moment problem for mixtures of smooth random variables. How many moments of a random variable must we match to approximate it?

2. pseudorandom projection matrices

Goal: generator G such that $w \in \mathbb{R}^n$, $x \sim \mathbb{S}^{n-1}$, $b \sim B^r$, $\langle w, x\rangle \sim_{\varepsilon}^{CDF} \langle w, G(y)\rangle$.

Random projection of w onto dimension 1.

2 step dim reduction. Project $Q_1 : \mathbb{R}^n \to \mathbb{R}^{\sqrt{n}}$. $x_1 \sim \mathbb{S}^{\sqrt{n}-1}$. Derandomize the first step. Use a pseudorandom projection matrix P_1. P_1 pseudorandom projection can be sampled in $\sim \lg n$ bits. Reduction to $\sqrt{n}$ dimensional problem. $n \to n^{\frac{1}{2}} \to n^{\frac{1}{2^2}} \cdots$. $x_t \sim \mathbb{S}^{\sim \lg n}$.

If we have P_i¡ we can sampl using $\sim \lg n$ random bits.

2. Two-step generator

Need $\langle Qw, x_1\rangle \sim_{\varepsilon}^{CDF}$. Make have approximately equal lower order moments. Mtach enough lower moments. How many moments to match?

3. Moment matching.

Truncated moment problem: Suppose for all $d \leq r$, $\mathbb{E}(X^r) = E(Y^r)$, how large is $\mathrm{DCDF}(X, Y)$?

$r = \left(\frac{1}{\varepsilon}\right)^C$ for ε CDF distance bound.

Main observation: very general, holds for all rv with matching moments. Understand structure of rv's better!

Only $\left\|Qw\right\|_2$ matter by rotation invariance. $M = \left\|Qw\right\|_2, N = \left\|Pq\right\|_2$¿ "Mixtures of different scalings of z_1.

For random variables of the form MZ, NZ,

1. $\mathbb{E}M^d \le \mathbb{E}N^d$
2. Moments of M don't grow too fast
3. Z sym, C^∞ CDF.

Matching $\sim \frac{\ln\left(\frac{1}{\varepsilon}\right)}{\ln n}$ moments enough. Cf. structure of rv and moment matching : Anandkumar, Kakade, Hsu.

Approximate orthogonal designs. t-wise independence fools degree t polynomials on the hypercube.

Orthogonal t-design fools degree t polynomial on ...

Orthogonal t design fools degree t polys on rotation matrices. $|\mathbb{E}_D p - \mathbb{E}_{\mathcal{H}} p| \le \varepsilon$. Real analogues of unitary designs studies in quantum computing.

Using Brandao-Harrow, Horodecki 12, based on Bourgain-Gamburd12, expander walks on Lie groups.

Draw R according to D, P =first $\sqrt{n}$ rows of R Uniform random projection Q choosing first $\sqrt{n}$ rows of rotation matrix R. Even moments of $\|Qw\|_2$ are polys in entries of S.

Degree 2 on sphere? Require many more ideas, because here we're taking threshold. Rotation invariant.

3.11 Rectangles are nonnegative juntas

Alice and Bob have $x, y \in B^n$ and want to compute $F(x, y)$ minimizing communication.

We want to understand communication complexity of $f \circ g^n$, g a 2-party function (gadget), $B^b \times B^b \to B$, $x, y \in (B^b)^n$. Inputs x, y encode $z = g^n(x, y)$.

Intuition, inputs to outer function hidden by gadgets. Spend communication to query. CC of converse function explained by query complexity of outer. Conjecture: simulate cost d rand protocol for $f \circ g^n$ using height d rand dt for f. Gadget must be chosen carefully.

$$BPP^{cc}(f \circ g^n) = BPP^{dt}(f).$$

We prove it for degree d conical junta for f.

Definition 3.8: Conical d-junta: nonnegative combination of d-conjunctions $(\sum a_i z_{i_1} \cdots z_{i_d})$. From DT, take the sum over accepting paths.

Junta theorem. f partial fnction, g inner-product on $\Theta(\lg n)$ bits, Π is cost d rand prot for $f \circ g^n$. Then exists conical d junta approximates probability of acceptance. For each $z \in B^n$, associate 2-party encoding.

$$h \approx \mathbb{P}_{(x,y)\sim(g^n)^{-1}(z)} \Pi(x, y) \text{ accepts}$$

Gadgets hide info from protocols. Polynomial approximation. Sherali-Adams vs. LP's.

Picture. communication matrix of $f \circ g^n$. Look at cc of gadgets, g^n. Partitioned into bands. Understand $\mathbb{P}$ accept. induces partition of communication matrix. Suffices to understand for single rectangle. What's the fraction inside the rectangle. Main theorem: exists conical d-junta h, $\mathbb{P}[(x, y) \in R] \approx h(z)$.

Corollaries: simulation theorems.
Communication-to-query simulation for NP.

$$\mathsf{NP}^{cc}(f \circ g^n) = NP^{dt}(f)\Theta(b = \Theta(\lg n)).$$

NP, QPP (smooth rectangle approx rank$_+$), SBP (corruption), PostBPP communication analogues. P, PP. BPP, MA?

Resolve open problems: query lower bound gives communication lower bound. SBP^{cc} not closed under $\cap$. small bounded-error comp: yes accept $\geq \alpha$, no accepted with $\leq \frac{\alpha}{2}$. (2) corruption does not characterize MA^{cc}, $\not\subseteq SBP^{cc}$. No efficient error amplification for ε-rank$_+$. Clique vs. independent set problemL coNP$^{cc} \gg$...

Open

1. more applications of junta theorem
2. simulation theorems for BPP
3. improve gadget size down to $b = O(1)$. Would give new proof of $\Omega(n)$ bound for set-disjointness.

Muliparty?

3.12 Poly low-error PCPs with polylog n queries via modular composition

CIRCUIT-SAT$\in \mathsf{NP} = PCP$.

Parameters

- ε error probability
- Σ alphabet size
- k number of provers.

We require $r = |R| = O(\ln |S|)$. ALMSS give $\varepsilon < 1, k, \Sigma$ constant.
What other parameters possible?

- $\varepsilon \geq \frac{1}{\text{poly}(|S|)}$.
- $\Sigma \geq \left(\frac{1}{\varepsilon}\right)^{O\left(\frac{1}{k}\right)}$.
- $k \geq 2$.

BGLR93 conjecture: These are the only constraints.

Results: fix $\varepsilon = \frac{1}{|S|}$. $\Sigma \geq |S|^{\Omega\left(\frac{1}{k}\right)}$, $k \geq 2$.

1. Previously, $\Sigma > e^{|S|}$ or $k > \ln(|S|)^{\Omega(1)}$.

2. Our result: $k = \ln\ln(|S|)^{O(1)}$. $\Sigma = |S|^{1/\operatorname{poly}\log\log(|S|)}$. (Exponential improvement.)

Derandomized repetition: Σ constant, $k = \ln|S|$. AS98, RS97, DFKRS99, 11: for any $\delta \in (0,1)$, $\Sigma = |S|^{1/\ln^\delta |S|}$, $k = O\left(\frac{1}{\delta}\right) \ln^\delta |S|$.

PCP's are made of

- components: using locally-decodable codes, sum-check, etc.

  Hardwired mess?

- composition: recursive application of components

  Principal idea from AS92.

  PCP of proximity: modular composition for large error parameters.

  AS93, RS97: local list-decoding of codes.

  Local-reader, LDRC (for small error)

  DH13: list-decoding PCP's, modular compositions for small error. Gets stuck for many compositions.

  Distributional decodale PCP's, handles non-constant number of compositions.

  How does modular composition work? Projection PCPs. Special A, other B_i.

  V verifies

  - A's answer, $\Phi(\gamma)$ for some Φ
  - consistency: $\forall i, p_i(\gamma) = \beta_i$.

  γ has large alphabet... Try to decrease using composition.

  V' access other subroutines A', B_i'. ???

  Attempt 2: Find $f_i(x)$, for some x such that $\Phi(x)$.

  Attempt 3... MORE SNOWMEN with scrolls and laser powers

We seem to have found more correct way to compose PCPs.

3.13 List decoding radius of Reed-Muller codes over small fields

Codes:

- rate (how many codewords?)
- minimum (pairwise) distance $d_{\min}$.

List decodability.

Reed-Muller codes. points are $\mathbb{F}^n \to \mathbb{F}$, codewords are $P, \deg(P) \le d$. Schwartz-Zippel: $\delta_{\min} = 1 - \frac{d}{p}$.

List-decoding radius is $\max r_0$ such that in a ball of radius $r_0\varepsilon$, the number of codewords is independent of n. Look at a ball of radius $1 - \frac{d}{p}$. How many codewords? p^n. Look at polynomials $P(x) = (L(x) - 1)\cdots(L(x) - d)$. p^n such functions. Does not rule out list decoding radius being $1 - \frac{d}{p}$.

Past work: List decoding, small fields.

1. Goldreich-Levin: $p = 2, d = 1$. Linear polys over $\mathbb{F}_2$.

2. Goldreich, Rubinfeld, Sudan, 1995.

3. Gopalan, Klivans, Zuckerman, 2008. For any fixed degree, indpendent of n. All d, $p = 2$. Specific to $p = 2$. $c(d, p, \varepsilon)$ codewords. Conjectured true for all p.

   Gave black box algorithm: combo bound gives algorithmic construction.

4. Gopalan showed this for $d = 2$ and any p.

Beyond $\delta(d, p)$. Fix $e < d$. Ball of radius $\delta(e, p) - \varepsilon = 1 - e/p - \varepsilon$. Number of codewords $> e^{n^{d-e}}$.

$p = 2$: tight.

We extend it to all p.

Corollary: weight distribution. Number of codewords of weight $\leq 1-e/p-\varepsilon$ is $e^{\Theta_{p,d,\varepsilon}(n^{d-e})}$.

Given $\mathbb{P}(P(x) = g(x)) > \frac{d}{p} + \varepsilon$.

Simple cases: $g(x)$ constant implies $P(x)$ is constant. $g(x) = g(x_1, \dots, x_c)$ implies $P(x) = P(x_{[1,c]})$. Only p^{p^c} independent of n. (Key observation.)

Given $g : F^n \to F$. Approximate.

1. Weak regularity lemma: get low complexity proxy g' for g mde of few low degree polynomials.

   if $\mathbb{P}(P = g) > \frac{d}{p} + \varepsilon$, ithen $\mathbb{P}[P = T_P(P_{[1,c]})] > \frac{d}{p} = \frac{\varepsilon}{2}$. if actually variables, then done. Here, low-degree polynomials.

2. Any f close to g, g' is a composition of few low degree polynomials.

   So we need higher-order Fourier analysis to show independent enough.

   $P(x) = T(P_1, \dots, P_c)$. Number of P's $< c(p, d, \varepsilon)$. P_i's regular by higher-order FA.

Follow-up: extend to nonprime fields.

With Lovett, extend to large fields.

List-decoding radius is min distance.

3.14 Cpacity of causal binary channels

Alice transmits a message to Bob. Eve can corrupt the message. Several models:

1. random noise (Shannon)

2. worst case (Hamming): best bounds: Gilbert-Varzamov, etc.

3. (weaker channel model) list decoding, weakened reconstruction goal

4. causal adversary: between random noise and worst case adversaries. Even decides how to corrupt based only on previously transmitted messages.

We give tighter bounds for causal adversary.

Upper bound: need adversarial strategy for Eve. Babble-and-push attack.

- Babbling phase: behave like random noise. Randomly tamper np bits.
- Push phase: construct set of codewords based on coruped bits transmitted so far.

  Sleect one codeword from set and push transmitted codeword towards selected one. With Plotkin's bound, the tampered word lies midway between transmitted and selected word. Both of them look the same to Bob.

Lower bound:

1. Encoder: deterministic codes.

   After $\approx nR$ bits Eve can learn the codeword.

2. Encoder: cdoes with 1-time randomness. Messages mapped to different codewords with different probabilities. After $n(R+\varepsilon)$ bits can learn the codeword.

   Distribute randomness into codewords independently. Keep tossing the coin and adding randomness every other time.

3. Encoder: concatenated codes with privte randomness.

   Behaviors of Eve: enclosed by curves: use at beginning vs. end. Bob's guess of Eve's behavior.

   Intersect. Possible successful decoding points.

4. Decoding: list-decoding and unique decoding. Obtain list of messages, wirte up encoings. Consistency checking. Two words consistent if differ in limited number of places.

   Look at Hamming balls $r = \frac{n-1}{4}$. If in intersection or non, then fails consistency checking. Increase t to next position.

   If in exactly 1 ball, with high probability can decode.

Where is causality used? Eve does not know what suffix of the transmitted codewordl. There is sufficient randomness i the suffixes of codewords by design. Eve cannot push all the balls close to each other.

3.15 Linear-sized Spectral sparsification in almost quadratic time and regret minimization beyond matrix multiplicative updates

Given G we want to turn it into sparser $\widetilde{G}$ with $L_G = \sum_{e\in E} w_e L_e$, $L_{\widetilde{G}} = \sum_{e\in E} s_e L_e$, $|\widetilde{E}| \leq \widetilde{O}(n)$. Want

$$(1-\varepsilon)L_G \preceq L_G \preceq (1+\varepsilon)L_G.$$

More generally, turn $\sum_{e\in E} v_e v_e^T = 1$ into $\sum s_i v_i v_i^T$.

History

1. Subdivide graph into expanders and repeatedly sample.
2. Sample by effective resistance, matrix concentration bounds
3. $O\left(\frac{n}{\varepsilon^2}\right)$, $\widetilde{O}(n^4)$ running time. Novel ad-hoc potential function.
4. $O\left(\frac{n}{\varepsilon^2}\right)$, $\widetilde{O}(n^{2.001})$ running time. Novel ad-hoc potential function. Optimization techniques: Regularization/smoothing.

We show a family of algorithms parametrized by $q \geq 2$: $O(\sqrt{q}\frac{n}{\varepsilon^2})$, $\widetilde{O}(n^{2+\frac{1}{q}})$. Construct sparsifier iteration by iteration.

$$A^{(t)} = A^{(t-1)} + s_{e_t} v_{e_t} v_{e_t}^T.$$

Main contributions: Unifying framework based on optimization techniques subsumes previous potential techniques.

1. Dual: online optimization: follow the regularized learder, mirror descent.
2. Primal: non-smooth optimization: smoothing, coordinate descent.

Computational speed-up.

Sparsification vs. online optimization. Given an algorithm $X^{(t)} \in \Delta_n$, adversary $M^{(t)} \in S^n(\mathbb{R})$ loss matrix, loss suffered is $M^{(t)} \bullet X^{(t)}$ matrix inner product. Repeat T times. Minimize $R_T = \sum_{t=1}^T M^{(t)} \bullet X^{(t)} - \min_{U \in \Delta_n} \sum^T M^{(t)} \bullet U$. First is loss of algorithm, second is loss of best desnity matrix, $\lambda_{\min}(\sum^T M^{(t)})$

MMWU algorithm: trivial generalization to SDP world

$$X^{(t)} \propto e^{-\alpha \sum_{i=1}^{t-1} M^{(i)}}.$$

General version of MMWU regret bound:

$$R_T \leq \frac{\ln n}{\alpha} + \alpha \sum_{t=1}^{T} (M^{(t)})^2 \bullet X^{(t)}$$

fixed diamter term, per-iteration widgdth terms.

Larger α, Explore space faster; more apt to be fooled by adversry.

$$\lambda_{\min}\left(\sum^T M^{(t)}\right) \geq \sum M \bullet X - ...$$

Idea: use regret bound to lower ound $\lambda_{\min}$. Similar for $\lambda_{\max}$.

Choose edges and weights to make $\sum^T s_e v_e v_e^T \bullet X$ large.

Get Spielman-Srivastava result. Pick e with $p_e = \frac{\|v_e\|^2}{n}$. Ensures $s_e v v^T$ is unbiased estimate.

$$\mathbb{E}\lambda_{\min} \geq 1 - \frac{\alpha n}{T} - \frac{\ln n}{\alpha}$$

$\alpha = \frac{2 \ln n}{\varepsilon}$, $T = \frac{4n \ln n}{\varepsilon^2}$.

Can we do better? MMWU is instantiation of larger class of online algorithms: follow the regularized leader. Each FTRL gives different tradeoff.

$MD_{\frac{1}{2}}$ algorithm. Instead of exponential, $X^{(t)} = (cI - A^{(t)})^{-2}$, for $c > 0$ such that $X^{(t)} \in \Delta_n$. Modify edge choise. $\geq 1 - \frac{\alpha\sqrt{n}}{T} - \frac{\sqrt{n}}{\alpha}$. Slower at moving around. $T = O\left(\frac{n}{\varepsilon^2}\right)$.

$p_e = \frac{\|v_e\| \|v_e\|_{X^{(t)}}}{\sqrt{n}}$.

Primal view: smoothing and regularization.

- Design potential function catpure eigenvalue of A: $\Phi(A^{(t)}) \approx \lambda_{\min}$.
- Perform coordinate descent in edge basis.
- Converges fast if Φ has Lipschitz gradient (smooth).
- Note $\lambda_{\min} = \min^{(t)}_{X \in \Delta_n} \bullet X$ not smooth So regularize: $\frac{1}{\alpha}F(X)$. Regularized introduces error $\frac{1}{\alpha}\Delta$ error. But stable under perturbation. Apply coordinate descent with regularized potential. $-\alpha \left\| s_e v_e v_e^T \right\|_*^2$. (width term) Regret bound.

MMWU is based on entropy regularizer $F(X) = X \bullet \ln X - I \bullet X$. Asymptotically optimal for small size steps. We base on $\sqrt{}$regularizer $\mathrm{Tr}(X^{\frac{1}{2}})$.

QUestions: other applications? (Steltjes ransform in random matrix theory, multi-armed bandit) other usedful regularizer? (ex. logdet regularizer)

4 6-16-15

4.1 Forrelation

What's the biggest dvantage a quantum computer ever guives you for anythign

$\widetilde{O}(n^2)$ and $2^{(\widetilde{O}(n^{\frac{1}{3}})}$. These are only conjectural.

We like to work in the black box model, where we can prove exponential and larger quantum speedupt=s

$f : [N] \to [M]$. Querying means you can feed it a superposition and the black box gives a superposition of answers.

Let P be a promise problem about f, for example, is f 1-to-1 or 2-to-1? Is f periodic or far from periodic?

Two quantities we care about:

1. $Q(P)$, bounded-erro quantum query complexity of P
2. $R(P)$, bounded-error randomized query complexity of P.

Shor's result is

1. $Q(PeriodFinding) = O(1)$.
2. $R(PeriodFinding) = \widetilde{\Omega(N^{\frac{1}{4}})}$.

Buhrman et al.'s speedup question: Is this the best possible? Could there be a property of N-bit strings that took only $O(1)$ queries to test quamtumly, but $\Omega(N)$ classically?

For all of them the quantum randomized query complexity seems to hit ceiling at $\sqrt{N}$. (Glued trees problem; a quantum algorithm finds the exit vertex in $Q = O(\text{poly}\log N)$, $R = \Omega(\sqrt{N})$.) Permutation symmetry: only polynomial quantum speedups are possible. Until this morning, the best speedup was $N^{\frac{1}{2}}$; now there's a $N^{\frac{1}{4}}$. Now there's a fourth root separation.

1. We give the largest known quantum speedup, a problem with

$$Q = 1, \qquad R = \Omega\left(\frac{\sqrt{N}}{\ln N}\right)$$

   for classical peopel: A lower bound on number of randomized queries needed to detect small pairwise covariances in real Gaussian variables $x_1, \ldots, x_N$.

   Optimality of speedup: for eveyr partial Boolean function P, if $Q(P) \le T$ then $R(P) = O(N^{1-\frac{1}{2T}})$. This answers Buhrman's question in the negative.

   Classical: Rand alg to approximate bounded low-degree polynoimals.

Problem 4.1: Given $f, g : B^n \to \{-1, 1\}$, $N = 2^n$, let (how correlated with Fourier transform)

$$\Phi_{f,g} := \frac{1}{2^{3n/2}} \sum_{x,y \in \{0,1\}^n} f(x)(-1)^{x \cdot y} g(y).$$

Determine whether $\Phi_{f,g} \ge 0.6$ or $|\Phi_{f,g}| \le 0.01$.

Introduced this problem as a candidate for a black-box problem in BQP but not in PH (needs a classical circuit lower bound). Showed that $R(Forrelation) = \Omega(N^{\frac{1}{4}})$, $Q = 1$.

The trivial quntum algorithm: start with in zero states, Hadamard, apply f, H, g, H. Can even reduce from 2 queries to 1. Use a control qubit; Hadamard the control qubit.

Problem 4.2 (Gaussian distinguishing)**:** Given $N(0,1)$ Gaussian $x_{[1,M]}$, either all independenct or in fixed low-dimensional supspace $S \le R^M$, Cov $\le \varepsilon$ for all i, j.

GD$\le$ F

$F(x) \sim N(0,1))$, $G = \widehat{F}$. $f() = \text{sign}(F(x))$. Then $\Phi_{f,g} \approx \frac{2}{\pi}$. If iid...

Any classical algorithm for Gaussian distinguishing must query $\Omega\left(\frac{1/\varepsilon}{\ln M}\right)$. In the Forrelation case, $M = 2N$ and $\varepsilon = \frac{1}{\sqrt{N}}$, so get $\Omega\left(\frac{\sqrt{N}}{\ln N}\right)$.

query.

If orthogonal, queries returns independent $N(0,1)$ Gaussian.

Ue Gram-Schmidt and Azuma to argue that first t query responses are close to independent Gaussians.

4.1.1 Classical simulation of k-query Quantum algorithm

A be a quantum algoorithm makes T queries, ten $p(X) = \mathbb{P}(A \text{ accepts } X)$ is a real poly in x_i's of deg $\leq 2T$.

Actually there is a deg $= 2T$ block-multilinear poly, which equals $P(X)$ whenever all $= X$, and bounded in $[-1, 1]$ for $X_i \in B^n$.

Theorem 4.3: TQuery only $O\left(\left(\frac{N}{\varepsilon^2}\right)^{1-\frac{1}{k}}\right)$

Identify influential variables, split...

k-fold forrelation. Conjecture that k-fold forrelation requires $\Omega(N^{1-\frac{1}{k}})$ randomized queries, optimal gap for all k¿

k-fold forrelation is BQP complete for $k = \text{poly}(n)$.

OPen

1. Prove classical lower bound for k-fold forrelation.
2. ny partial boolean P such that $Q(P) = \text{poly}\log N$, $R(P) \gg \sqrt{N}$.
3. Extend to arbirary polynomials, (2: DKPO)
4. Best quantum/classical query complexity separation for sampling problems.

4.2 Quantum information complexity

How much quantum information need to exchange?

- Definition of quantum information complexity.
- Interpretation of amortized communication.
- $QIC \leq QCC$, additivity of QIC
- Direct sum for quantum communication
- Applications to communication lower bound.

2 communication problems:

1. compress messages with low info content.
2. transmit messagee noislelessly over noisy channels.

We focus on (1).

Protocol transcript.

Coding for interactive protocols

1. Can we compress protocols that do not convey much information?

2. What is the amount of information conveyed by a protocol? Amount of info at end of protocol, optimal asymptotic rate.

Define information complexity, cost. Additivity, $IC \leq CC$, operational interpretation, direct sum on composite functions, convexity, continuity, etc.

$$IC(f, \mu, \varepsilon) = \inf_{\Pi} IC(\Pi, \mu).$$

Applications of CIC: direct sum, direct product, exact communication complexity of DISJ_n.

Quantum:

1. quantum entropy $H(A)_P = -\mathrm{Tr}(\rho^A \lg \rho^A) = H(\lambda_i)$ for $\rho_A = \sum_i \lambda_i |i\rangle\langle i|$.
2. No pre-shared entanglement, classical messages; arbitrary pre-shared entanglement, classical messages, etc. Many problems.

   In Yao model: no-cloning: cannot copy m_i, so no transcript. Can only evaluate information quantities on registers defined at asame moment in time. Not well defined.

   Cleve-Buhrman model: m_i's can be uncorrelated to inputs. (Teleport at each time step.)
3. Solutions: keep as much info as possible, measure final correlations.

   Reversible, no garbage, only additional info is function output.

   Bt QIC trivial.
4. Measure correlation at each step.

   Problem: for M messages and total communication C, can be $\Omega(M\dot{C})$. Want $QIC \leq QCC$ independent of M, direct lower bound on communication.

Approach: reinterpret classical info cost.

1. Shannon task: simulate noiseless channel over noisy channel.
2. Simulate noisy channel over noiseless channel.

4.3 Sparse quantum codes from quantum circuits

Quantum error correction allows us to deal with noise in quantum computation. Most of them are satablizer codes.

1. Define stabilizers: Sets of commuiting operators
2. Logical operators: operators anticommute.

Subsystem codes:

1. Use excess logicla qubits as gauge and correct errors up to transformations on gauge space

2. Can be sparser with simpler syndrom measurements, higher threshholds

Isolate disturbance among gauge qubits.

Sparse code: $[n, k, d]$ if encodes k logical qubits into n physical qubits and detect any Pauli error of weight $< d$

local measurements and look at correlations.

Topological codes, LDPC codes. Major challenge, find sparse qc perf well with $k, d = O(n)$

qLDPC? Arithmetic hyperbolic curves: linear number of logical cubits, distance $n^{.3}$. (Guth, Lubotzky)

Given any quantum stabilizer code, given circuit of size s: give you an output code of more general subsystem variety, with parameters $[n, k, d]$, k, d preserved, and now sparse. All gauge generators constant length. Circuit of size s. Prepares particular stabilizer code. Circuit is generator for code. Can always choose to have small size, $n_0 + \sum_i w_i$ (weights of original generators).

Systematic way convert any stabilizer code into sparse subsestem code with same parameters.

Begin with stabilizer code of your choice. Write a quantum circuit for measuring the stabilizers of this code. Ancilla preparation, post-selected measurement (formally). Turn circuit elements into I/O qubits. (Turn time into space. Why circuit size enters into argument.)

Add gauge generators via Pauli circuit identities. Table for Clifford group (not universal): can write down corresponding gauge generators.

Circuits are linear operators preservg the code space $V = |00\rangle\langle 00|$. Gauge equivalence of errors. Equivalent on logical space, only change on gauge cubits decoupled? Push to boundary. $V_E = \pm V_{GE}$. Equivalent from persp of operator act by conjugation. **squeegee**. **Spackling**: can leave residue. Snapshots of history of opertor start at boundary and propagate. Worldsheets.

"Clean windshield." Bugs spread out nonlocally over edge of windshield. Doesn't let us say anything about distance, can propagate to large error. Spread error, make detectable up to distance of code. use fault-tolerance from circuit, arguments from expanders (constant-degree expanders exist with enough fault tolerance).

Spackling shows k preserved.

corollary: almost good sparse subsystem codes. Concatenting code $\sqrt{\ln n}$ times, $d = O(n^{1-\varepsilon})$.

Local subsystem codes with $d = O(L^{D-1-\varepsilon})$, $\varepsilon = O\left(\frac{1}{\sqrt{\ln n}}\right)$.No errors that look like 1D extended objects, have some surface area.

How do classical hard disk drives work? Microscopically chunks of material magnetized pointing up/down. Below certain temperature, hard for all spins to flip. They want to align with their neighbors. If one flips, encouraged to flip.

Isoperimetric argument. Hve to flip s.t. where energy penalty is proportion to surface area. Unlikely. Result: something modeled by Ising model, can encode info, sit there, not touch it for exponentially long time and still get right with constant probability.

Self-correcting memory.

Can we make self-correcting quantum memory?

Our codes are candidate self-correcting quantum memories.

4.4 Parallel repetition for general games

2-prover 1-round gmes.

$(x, y) \sim \mu$. THey win if $V(x, y, a, b)$. $\text{val}(G) = \max_{a,b} \mathbb{P}(\text{win})$. c is the answer length. $c = |a| + |b|$.

Can allow them to share randomness.

Ex. CHSH game. $x, y \in_R B \times B$. $V = 1$ iff $a \oplus b = x \wedge y$, value .75.

Projection game: for all x, y, z there exists unique b¡ $V(x, y, a, b) = 1$. Unique: for all x, y, a $\exists b$ and vice versa.

Required to output n answers. $a_i(x_1, \ldots, x_n)$, etc. Win if $V(x_i, y_i, a_i, b_i) = 1$ for all i.

Clear that $\text{val}(G^n) \geq \text{val}(G)^n$. Is the reverse true: is the best strategy to play independently on all coordinates? No, counterexamples of Raz08.

1. Verb95: $\text{val}(G^n) \to 0$ if $\text{val}(G) < 1$.
2. $\text{val}(G^n) \leq (1 - \varepsilon^{32})^{\Omega(n/e)}$. Improved to 3.
3. For projection: $(1-\varepsilon^2)^{\Omega(n)}$. Raz08: unique game with squared dependence (connection to unique games conj?)

PCP amplicfication: NP-hard to decide if $\text{val}(G)$ or $\text{val}(G) \leq 0.99$. Parallel repetition: δ.

Connections: UGC, foam, ...

That was high-value regime.

Small value regime? DS13:

$$\text{val}(G^n) \leq \delta^{\Omega(n)}.$$

Optimal NP-hardness for approximating set cover.

Theorem 4.4: $\frac{1}{2^c} \leq \delta \leq \frac{1}{2}$, $\text{val}(G) = \delta \implies \text{val}(G^n) \leq \delta^{\Omega\left(\frac{n \lg\left(\frac{1}{\delta}\right)}{\varepsilon}\right)}$.

$\delta \leq \frac{1}{2^c}$ transition (random guessing) $\delta^{\Omega(n)}$.

Reprove PRep theorems.

FIx a strategy for G^n. W event of winning in all coordinates. If$\mathbb{P}(W) \geq$, design a strateg yfor G with $\mathbb{P} > \delta$¿

Embed (x, y) into random coordinate, sampel rest accoring to appropriate distribution and run for G^n.

Tries

1. Set $X_i = x, Y_i = y$, uniform on others. Best boudn $\geq \mathbb{P}(W)$.
2. Need into zoom into W. Sample $A_i, B_i | X_i = x, Y_i = y, W$ approximately. How to sample? Alice, bob only know x, y, resp.

3. $X_i = x, Y_i = y$, jointly sample $R|X_i = x, Y_i = y, W$. (A bunch of properties) Alice and Bob should know W. (?) Also breaks the dependencies. Once sample correct R, can forget W.

Want mutual informations to be small.

Exhibit R.

What if they know approximate distribution? How jointly sample? Use correlated sampling theorems. If approximately know some distribution. Some procedure sample approximately.

What's R? Inspired by previous PR proofs and direct products in communication complexity BRWY13.

?? Pic. Create various symmetries.

Open problems

1. PR for general entangled games (pre-share entanglement), even for high value regime.

   Holds for projection games, and product distributions.

2. PR for 3 prover games, even high-value regime. Define info complexity for multiparty.

4.5 An interactive information odometer

$$IC_\mu(\Pi) := I(\Pi; X|Y) +$$

what bob learns about X and what ALice learns

Information is amortized communication.

$$\lim_{n\to\infty} CC^\varepsilon_\mu(f^n)/n = IC^\varepsilon_\mu(f).$$

Sharp threshold: $IC^{\frac{2}{3}}_\mu(f) = I$. WIth $\ll nI$ cc, success in computing f^n is $2^{-\Omega(n)}$. Like PR theorem for information complexity. Robust characterization.

More applications to CC.

1. Information-theoretic secure communication between untrusted parties.

   NO crypto assumptions!

2. Interactive compression. Execute long protocol reveal few info, try to compress. Can be reduced to potentiall easier problem: IC= lg(CC).

All related to information odometer problem. Same bottleneck.

Maintain an online estimate of the IC of π up to constant factor, with little information overhead.

Why does this task require interaction? Internal information cost is a function of both inputs. Ex. π: alice sends x with probability 0.01, and $z \in_R B^n$ otherwise. Bob never knows whether he's learning something: marginal distribution is same. Variance is huge. Alice knows but not hard to adapt to get neither company can unilaterally estimate the quantity.

Can break π into individual bits send.

If Alice is the speaker in round i, $M_i \sim B(p_i)$.

$p_i = \mathbb{P}(M_i = 1|xm_{<i})$, q_i for bob.

Bob's prior $B(q_i)$ Bob learns from M_i: $D(p_i||q_i) = \Theta(p_i - q_i)^2$ when bounded away from 1.

keep a constant factor estimte for all paths m for all t. Clicks with prob $(p_i - q_i)^2$ prob each round.

Primitive problem: goal: output 1 w.p. $(p_i - q_i)^2$ revealing $\precsim (p_i - q_i)^2$ btis of info.

Naive attempts.

1. Send p_i explicitly.

2. Correlated sampling.

Main result: There is 2-round protocol that is

1. correct: $\mathbb{P}(\tau \text{ outputs } 1) = 2(p-q)^2$.

2. low info: $IC(\tau) \leq O(H(p-q)^2)$.

Running with prob $\approx \frac{1}{I}$ each round. Number of clicks $\approx 2I_{xy}(m_{<t})$, information overhead $\leq O(\lg C)$.

Alice samples $Z_p \in [0,1]$ from density function $\mu_p(z)$. Alice sends Z_p to Bob, Alice sends $(Z_p > p)$, Bob sends $(Z_p > q)$. Click (output 1) if inconsistent.

Correctness: area is $2(p-q)^2$.

Main lemma: $D(Z_p||Z_q) \leq 8(p-q)^2$. Data processing, bounded by entropy of clicking.

ex. if Alice sends a random string, prob click is 0. If $E = 1$, for all i get > 0.

Abort if number of clicks ≥ 10.

With small variation, can implement odometer even in adversarial setup (truthful mechanism for solicitng information).

4.6 Inapproximability of Nash equilibrium

A short algorithmic history of Nash in 2-player games:

1. Finding a Nash equilibrium in 2-player games is PPAD-complete. Finding $n^{-\varepsilon}$-Nash equilibrium is still PPAD-complete.

What about constant? There is a quasi-polynmial algorithm for finding ε-Nash with constant number of players. Can we do better than $n^{\ln n}$?

Our focus is multiplayer games. Fact: normal form representation is exponential in number of players. Instead, we consider oracle access to the payoff tensor.

Exponential lower bounds for this model!

Talk about restricted classes of games. Succinct representation.

1. input is poly-size circuit implementation of value-query oracle

2. each pair of players simulataneously palys a seaparate two-player game. Same strategy in each 2-player subgame, utility is sum

3. Utility depends only on actions of small number of neighbors.

Theorem 4.5: Degree 3, bipartite, polymatrix game where each player has 2 actions: ε-approximate Nash is PPAD-complete.

How do we know right restriction to make? Get nice implications!

This is the "3SAT of PPAD." If you don't see anything to reduce from, use 3SAT. Similarly here.

Def. Arithmetic circuit. Generalized circuit: allowed to close the cycle. This defines CSP. ε-GCircuit: find assignment $x : V \to [0,1]$ such that for all gates (G, v_1, v_2, v), $x[v] = f_G(\pm\varepsilon) \pm \varepsilon$.

Not sure who Alice is playing with.

ε-approx Bayesian Nash equilibrium

Two player game, constant number of actions, ε-approx BN equilibrium PPAD-complete. (only number of types large)

All for additive approximations. Can talk about relative approximations.

ε-well supported relative nash equilibrium $\geq (1-\varepsilon)$ of optimal utility. Das13: utilities in $[-1,1]$ is PPAD-complete (negative payoffs not standard).

We get $[0,1]$.

Non-monotone markets. Raising the price of good G while fixing all other prices strictly increases the demand of G.

Weak gross substitutibility. Raising price of G does not decrease demand of other items. For WGS, approx market equilibrium can be found in polytime.

PPAD-hardness for non-monotone markets: find $\frac{1}{n}$-approximate market equilibrium is PPAD-hard.

Competitive equilibrium from equal incomes: equilibrium computed in practice, assign students to classes at Wharton. Stronger PPAD hardness for this problem.

HPV construction. Average gadgets (DGP)

Cor: ε-GCircuit, TWεRN, B, NMM, A-CEEI (course-match)

Beyond ε-Nash

What is the PCP theorem for PPAD?

Natural equivalent: only satisfy $(1-\delta)$ fraction of gates. Can be verified by reading a constant number of btis. Cojecture: (ε, δ)-GCircuit is PPAD-complete. (equivalent of PCP theorem)

4.7 Succinct obfuscation

Indistinguishability obfuscation: two programs with same functionality can't be told apart.

We have circuit obfuscation. Circuits are as large as their running time. Can we obfuscate small programs with long running times, i.e., Turing machines?

Turing machines with the same functionality and running times are indistinguishable. Should be efficient.

There is io for Turning machine which is efficiency preserving, semi-efficient. $O(M)$ in time $\widetilde{O}(TIME(M,x))$. Time depending on original time (and space). Without space dependence, doesn't generalize to RAM.

Simpler primitive: randomized encoding. Like single-input obfuscation. Encode nothing more than result.

3 different ways to construct.

1. Goal: randomized encoding for TM. Time(TM.encode)≪ Time(M(x)). andomized encoding for circuits: encode all poly-sized circuits in $NC1$. Each gate computed from 1 gate in C and few bits of randomness.

   TM encoding: first approach.

   Convert to circuit representation: width is space, depth is time. Encoding time is size of circuit, at least time complexity of machine. A lot of redundancy, a circuit applied over and over again. (1) C_M is succinctly represented by nex configuration circuit. (2) Each layer is computed from the same and few random bits.

   TM encoding: hardcoded next configuration circuit, K. Obfuscated circuit.

   Feature of proof: encoding is indistinguishable from dummy encoding. Hybrid encoding: modify behavior of one of layers. At time steps $< j$, output dummy layer.

2. CHJV: naive attempt

   to encode (M, x): circuit takes local state and symbol and output next state. Obfuscate this: Problems: doesn't hide CPU, tape, or verify inputs. Encrypt inputs and outputs. Rely on transformation to fixed movement pattern.

   Sign outputs and verify signatures not enough. Adversary can give input what was on tape before but is overwritten. Make it output a time stamp along with symbols; it also checks the timestamp is correct.

   IO compatibility: IO friendly signature schemes.

   Proof, high-level: Obfuscated circuit. Show indistinguishable from obfuscated that contains nothing but output memory which contains the answer. Indistinguishable hyperdistributions.

   Intermediate hybrid: jth configuration needed...

3. KLW: run for j steps, simulate remaining steps, output y. Don't need any memory configuration.

   Can still give CPU illegal inputs. Make the adversary: use a has function which hashes the entire tape. Need a proof that has correct hash. Using new symbol, update hash. Observation: program should not take huge inputs, must be succinct. Need special properties from hash function. Succinct hash value, proofs, update.

   Hash tape to succinct value in tree fashion.

   Security: Idea; provie adversary to correct calculation. What about

   (a) PPT adversary cannot find fake proofs. Not iO-Friendly!

   (b) Hash ouput binding to correct tape. No 'fake' proofs. But h, π.

   (c) Solution: positional merkle trees
   Hash value h is binding at p^*. Information-theoretic. Splittable signature schemes.

4.8 Dimensionality reduction

Depending on application, we may want f to be linear, oblivious to X, or preserve structural information.

In this talk, we want an ε-isometry:

$$\|\Phi(x-y)\|_2^2 \in [1-\varepsilon, 1+\varepsilon] \|x-y\|_2^2 .$$

In other words,

$$\varepsilon_T := \sup_{x\in T} | \|\Phi x\|_2^2 - 1| < \varepsilon.$$

Gordon's theorem Gaussian width

$$g(T) = \mathbb{E} \sup_{x\in T} \langle x, g\rangle .$$

Theorem 4.6: Let $T \subseteq S^{n-1}$. $\Phi \in M_{m\times n}$ Gaussian

$$m \gtrsim \varepsilon^{-2}(g(T)^2 + \ln(n^{-1})).$$

then $\frac{1}{\sqrt{m}}\Phi$ is $|ep$-isometry on T with prob $1-\eta$.

Instance-optimal version of JL, if T is K-sparse vectors, $g(T)^2 \lesssim k \ln\left(\frac{n}{k}\right)$. Also for infinite! Extends to subgaussian.

Gaussian matrix is dense. The time can be large, a bottleneck. The time to embed exceeds the time to solve the original problem.

We want a version of Gordon's theorem for a matrix with fast matrix-vector multiplication.

1. Sparse Johnson-Lindenstrauss Transform: Start with $\Sigma \in \mathbb{R}^{m\times n}$ iid random signs.
2. For each column independently, select exactly s entries uniformly at random without replacement, put rest to 0. Random selectors $\delta_{ij} \in B$.
3. $\Phi_{ij} = \frac{1}{\sqrt{s}}\sigma_{ij}\delta_{ij}$. Φ_x time $O(s\|x\|_0)$

How large to m and s need to be to get

$$\mathbb{E} \sup_{x\in T} | \|\Phi x\|_2^2 - 1| < \varepsilon?$$

Answer depends on appropriate complexity parameter of T. Previously known: $m \gtrsim \varepsilon^{-2}\ln|T|$, $s \gtrsim \varepsilon^{-1}\ln|T|$ ...

A prior not clear how to answer.

We define a new complexity parameter.$\max_{q\leq\frac{m}{s}\ln s}\left[\frac{1}{\sqrt{qs}\left(\mathbb{E}_\eta\left(\mathbb{E}_g \sup_{x\in T}\left|\sum_{j=1}^n \eta_j g_i x_j\right|\right)^q\right)^{\frac{1}{q}}}\right]$.
Given $m, s \gtrsim ..., \kappa < ...,$ get $\varepsilon_T \leq ....$
Unifies known sets for known results qualitatively.
New results for infinite unions of subspaces and manifolds.
Applications.

1. Sparse subspace embedding.

2. Sketching constrained least squares (LASSO). Compressed sensing.

3. Manifolds: with large prob preserve geodesic distances.

4.9 Fully homomorphic signatures from standard lattices

Publicly verifiable computable (PVC): game between Alice, cloud server, and Bob. Alice uploads data x. Arbitrary query/program P to get answer $y = P(x)$. Bob has a "pay-per-bit" channel: he wants to verify $y = P(x)$ without having seen the database.

Homomorphic signatures Alice run signing agorithm wrt a secret key. Upload with corresponding signature to cloud server. Run to get result y. Can also run homomorphically over. $\sigma_{P,y} = eval_{pk}(P, x, \sigma)$. Bob downloads $y, P, \sigma_{P,y}$ (homomorphically derived signature)

1. succinctness: only depends on output of y, not on runtime of P.

   $preprocess_{pk}(P) = pk_P$. $verify(pk_P, y, \sigma_{P,y}) = 1$. (soundness)

2. efficiency: preprocess independent of runtime of P and size of x.

   A can sign many databases and evaluate many programs.

3. security: if $y = P(x)$, the cloud cannot convince that result is $y' \neq y$.

   Challenger gives apk to the adversary. Gives a data base x. Challenger gives σ. Eventually the adversary claims to have come up with P, σ^*, y^*. Adversary wins if valid signature of wrong result:

   $$verify(pk_{P'}, y', \sigma') = 1, y' \neq P(x).$$

   Relaxation: x declared before pk (selective security). (Can also extend to many datasets)

Constructions of homomorphic sigatures.

1. linear functions (bilinaear, RSA, SIS)

2. bounded degree polys (ideal SIS , random oracle)

3. (multilinear maps)

There is secure hom signature scheme for many databases and arbitrary programs represented by circuits where

1. succinctness: poly(λ, d), λ security param, d depth for P.
2. efficiency: runtime of veri is same
3. security assuming hardness of small integer solutions.
4. context hiding: $\sigma_{P,y}$ reveals nothing about x.
5. locally computale
6. composable

Compare to other solutions.

1. CS proofs/SNARKs: cloud gives short proof verify$(y, n) = 1$. All of them use non-standard assumptions which are essential.
2. memory delegation: Bob needs multi-round (interactive) verification.
3. etc.

How to construct? Homomorphic trapdoor functions, to homomorphic signatures. How to instantiate from lattices.

HTF is a family of functions parametrized by $f_{pk,x} : U \to V$.

1. given secret key sk, inverting $f_{pk,i}()$ is easy
2. claw-freeness: hard to find $f_{pk,0}(u) = f_{pk,1}(u^*)$.
3. Given $f_{pk,x_i}(u_i) = v_i$, exist deterministic algorithms $eval_{in}(C, (u_i, x_i)_i) \to u'$ and $eval_{out}(C, (v_i)_i) \to v'$.

   Then $f_{pk,C(x)}(u') = v'$.

   Decouple comp over inputs and outputs separately, such that when apply to input get output.

   How construct homomorphic signatures from HTDF? Fix public key, sample random from output space.

   Sample preimage corresponding to bits. $sign_{sk}(x_i)_i \to \sigma$: sample $u_i, f_{pk,x_i}(u_i) = v_i$. Run homomorphic... work over inputs. Homo derived signature. Run separate and evaluate...

   $verify_{pk}(pk_C, y, \sigma_{C,y}) \to 1$ iff $f_{pk,y}(\sigma_{C,y}) = pk_C¿$ follows by correctness of HTDF if $y = C(x)$.

   If claw-freeness then secure. A gives x to C. Sample signatures ourselves: $v_i = f_{pk,x_i}(u_i)$. Public:$(pk, (v_i)), \sigma = (u_i)_i$. Adversary claims pk, v, σ.. A wins if $ver = 1$. A wins if verify on y' same for some $y' \neq y$. Is clawfreeness!

SIS problem: given $\mathbb{Z} \in (\mathbb{Z}/q)^{mn}$, find nontrivial short $z \in Z^m$, $Az = 0$.

Lattice algorithms: rapgen. short pre-image.

Fix $pk = A$, $sk = T.G \in (\mathbb{Z}/q)^{nm}$ global. $f_{pk_x}(U) = AU - xG$.

Want: given sk, invert $f_{pk,i}$ is easy.

4.10 Monotonicity testing

Property testing.

1. Monotone: accept
2. Far from monotone: reject with high probability

1. Simplest: edge tester with $O(n)$ query complexity.
2. Fischer: $\Omega(\ln n)$ queries.
3. Now: $\Omega(n^{\frac{1}{2}-c})$ lower bound. KMS15: $O(n^{\frac{1}{2}})$ query tester.

For eveyr $c > 0$ there is $\varepsilon(c) > 0$ such that any non-adaptive algorithm, to distinguish $\varepsilon(c)$ far away need $\Omega(n^{\frac{1}{2}-\varepsilon})$

Key ingredient: multidimensional central limit theorems.

We use Yao's minimax principle: tricky distribution over inputs to deterministic algorithms. Need D_{yes} supported on monotone functions, D_{no} supported on far from monotone functions Indistinguisability: for T making $o(n^{\frac{1}{2}})$ queries, difference btween aceping is $< \varepsilon$.

Both supported on linear threshold functions over $\{-1, 1\}^n$.

1. YES: σ_i uniform from $\{1, 3\}$. Monotone
2. NO: $\nu_i = -1$ with prob .1, $\frac{7}{3}$ with 0.9. Far from monotone. (because enough weight on negative)

For any tester making 1 query, $\mathbb{P} - \mathbb{P} = o(1)$.

Proof. difference is at most d_{TV}. Look like Gaussians. Sum of many independent reasonable random variables converges to Gaussian of same mean and variance. Use Barry-Esséen CLT. Apply the CLT twice. Have matching first 2 norms, converge to same Gaussian. □

For q queries instead of 1, use the multidimensional CLT.

Adapt multidim CLT for earth mover distance to get $\Omega(n^{\frac{1}{5}})$. Extebd nyktudun CKT. Their proof techique is Lindeberg's replacement method.

3 new ideas.

1. random variables that match arbitrarily many moments

   By matching h, only error term of order $h + 1$.

2. careful construction of mollifiers in CLT analysis

   Control width of error region where mollifier is in $(0, 1)$. (Using 2 ideas gives $\Omega(n^{\frac{1}{4}})$.)

3. pruning query set to make it nice.

   Delicate CLT analysis gives bound for scattered query sets (no two queries close together). SIlly example: ask same query over and over. Close by queries are likely to take same value, tester does not benefit much.

   Key reduction: every set of $O(n^{\frac{1}{2}-\varepsilon})$ queries can be pruned to become Q' scattered.

Lindeberg's replacement method:

$$|\mathbb{E}[\Phi(X_1 + \cdots + X_n)] - \mathbb{E}[\Phi(Y_1 + \cdots + Y_n)]|.$$

Φ is a smooth approximation to the indicator of union of orthants (hybrid method).

1. Swap out each X_i with Y_i, not introducing too much error.

2. Bound difference via Φ's Taylor expansion.

Adaptive testers still gap. Yao minimax only for nonadaptive. Does have polylog linear tester for linear threshold.

4.11 Quantum spectrum testing

How to mathematically represent the state of an atom. If it's a pure state it's $\rho = |v\rangle \in \mathbb{C}^d$¿

A mixed state is a probability distribution: (orthonormal) $|i\rangle$ with probability p_i. $|i\rangle, p_i$ unknown

Density matrix

$$\rho = p_1|1\rangle\langle 1| + \cdots + p_d|d\rangle\langle d|.$$

The p_i are the spectrum.

What would we want to do?

1. tomography: learn ρ approximately, up to some ε, whp.

2. spectrum estimation: learn $\{p_1, \ldots, p_d\}$.

3. spectrum testing: is $\{p_1, \ldots, p_d\} = \{\frac{1}{d}, \ldots, \frac{1}{d}\}$? rank r/far from rank r?

Setup: some experiment, run multiple times. Want to verify whether matches the theory. Have n copies of state. Running experiment (generating each copy) is expensive operation. Put into quantum computer. Q can measure the states.

Goal: minimize n in terms of d, ε.

Copy complexity is quantum version of sample complexity. Strong connections to the area of learning/testig probability distributions.

Design copy efficient algorithms.

Quantum tomgraphy: learn ρ.

1. Folklore: $n = \Omega(d^2)$ necessary (to learn all parameters)
2. $n \le \widetilde{O}(d^4)$ sufficient (quadratic gap)!
3. common claim: $n \le O(d^2)$ sufficient.

Spectrum estimation. ALgorithm works with $n \le \widetilde{O}(d^2)$. Simpler analysis gets $n \le O(d^2)$. Algorithm fails if $n = o(d^2)$.

If you can learn something, than you can also test in same number of samples.

Property testing: above gives quadratic. Distinguishing Unif(d) vs. Unif($d/2$). is sufficient and necessary: d copies. $n = \Theta(d)$. Quantum version of birthday paradox (d not $\sqrt{d}$)

Us: distinguishing U_d, $U_{d-\Delta}$.

Testing uiform vs. ε-far $\Theta(n/\varepsilon^2)$.

Maximally mixed state. Can test if ρ is maximally mixed, or ε-far.

Also test for rank $\le r$.

Any spectrum test/learning problem reduces to classical problem about random strings.

Let's study classicla analogue: test whether a distribution is uniform. Symmetries

1. Symmetry 1: permute n positions doesn't matter: a histogram is enough.

   Form the histogram

2. Symmetry 2: uniformity is symmetric property, so doesn't care about order. Sort the histogram. (Young diagram λ of partition) Let the frequencies be $\lambda_1 \ge \cdots$.

A uniformity testing algorithm:

1. given young diagram λ,
2. say yes/no based on λ.

Back to quantum: similar symmetries.

1. permuting n copies of ρ doesn't change anything.
2. Only care about spectrum of ρ. (Answers should be same for any $|i\rangle$.

Reduces down to: uniform spectrum testing algorithm: given young diagram, say y/n based on λ where λ:draw sample($a_1, \ldots, a_n \sim P^n$ (eigenvalue distribution), $\lambda = RSK(a)$ (before, sorted-histogram).

RSK algorithm: algorithm mapping strings to Young diagrams. Well-studied, 40 years. Nice combinatorial properties.

Boxes in first row is LIS of a, longest increasing subsequence of a. Enumerative combo, queueing theory, longest increasing subsequences... Mathmos love studying the RSK algorithms applied to random strings.

Main technique: method of moments for random Young diagrams. Can understand quantities like $\sum_i \lambda_i^k$ where $\lambda = RSK(a)$, a randomly distributed.

Technology known as Kerov's algebra of observables studies these moments.

Entropy: open problem.

4.12 Exponential Separation of Information and Communication for Boolean Functions

Classical results about message compression. Alice has x chosen according to publicly known distribution. Wants to send message to Bob so Bob can retrieve with high probability. Need to send $\approx O(H(x))$ bits.

Message-compression theorem: any message can be compressed to its information content.

Interactive-compression problem: what if Alice and Bob engage in an interactive communication protocol? Can the protocol's transcript be compressed to its "information content"?

Standard model of communication complexity. How many bits to exchange to compute $f(x, y)$?

Randomized CC: both players can use private and public random strings, have to compute $f(x, y)$ with probability $> \frac{2}{3}$ (over $(x, y) \sim \mu$).

$$CC(f, \mu) = \min_{\pi \text{ computes } f \text{ over } \mu} (CC(\pi, \mu)).$$

Can every protocol be compressed to its information content? Measure information content by information complexity.

Conditional mutual information is the information that a player who knows X learns about Y by seeing Π, on average.

$$I(\Pi; Y|X) = H(Y|X) - H(Y|X, \Pi).$$

Internal information complexity: amount of information that players learn about each other's input from the interaction.

$$IC(\pi, \mu) = I(\Pi; Y|X) + I(\Pi; X|Y)$$

what Alice learns about Y from Π, plus what Bob learns about X from Π.

$$IC(f, \mu) = \inf_{\pi \text{ computes } f \text{ over } \mu} IC(\pi, \mu).$$

What is the relationship between information and communication complexity? For all π, μ,

$$CC(\pi, \mu) \geq IC(\pi, \mu).$$

Hence for all f, μ,

$$CC(f, \mu) \geq IC(f, \mu).$$

The other direction is not true in general: $CC(\pi, \mu)$ can be much larger than $IC(\pi, \mu)$ because it can be wasteful: send unnecesseary information. But if it's wasteful, maybe it can be compressed.

Compression problem: given a protocol π, can π be simulated by π', such that $CC(\pi', \mu) \approx IC(\pi, \mu)$?

Equivalently, is it true that for all f, μ, $CC(f, \mu) \approx IC(f, \mu)$?

Known compression protocols do not give exactly what we want. For all f, μ,

$$IC(f, \mu) \le CC(f, \mu) \le 2^{O(IC(f,\mu))}.$$

Almost all known techniques for lower bounding CC give the same bound for IC. New functions and techniques may be needed.

We give an explicit f, μ such that $IC(f, \mu) = O(k)$ and $CC(f, \mu) \ge 2^k$ (so compression is not always possible). By Braverman 2012, this is the largest possible gap. GKR2014 have a similar result for a search problem with outpt size double exponential in k.

An application to the strong direct-sum problem. Try to solve simultaneously m instances of the same problem. Alice receives $(x_i)_{i=1}^m$, Bob receives $(y_i)_{i=1}^m$. Goal is to compute $f(x_i, y_i)$ for each i¿ $CC^m(f, \mu)$ is CC of best protocol that answers correctly with prob $> \frac{2}{3}$ on each ?

$$CC^m(f,\ mu) \ge \Omega(mCC(f, \mu))?$$

Let $AC(f, \mu) = \lim_{m\to\infty} \frac{CC^m(f,\mu)}{m}$. Braverman and Rao showed that $AC(f, \mu) = IC(f, \mu)$. Our results shows a gap between AC and CC. Hence a strong direct-sum theorem for CC does not hold.

4.13 Lower Bounds on the Size of Semidefinite Programming Relaxations

Unconditional computational lower bounds for classical combiantorial optimization problems in a restricted but powerful model of computation: all possible linear and semidefinite relaxations.

Goes back to Yannakakis in the 1980's, looking at flawed P vs. NP proofs. Refute all flawed proofs at once.

There are connections to optimizaiton and convex geometry. Settles open questions about semidefinite lifts of polytopes and ranks. Identify optimal approximation algorithms in the model.

For every constraint satisfaction problem, the SoS (Lasserre) hierarchy gives optimal algoritm in this model.

Derive lower bounds for general model from known countersxamples for the SoS model.

Math programming relaxations: powerful generl approach for approximating NP-hard optimization problems.

Three flavors:

1. Linear (LP) ((A)TSP)

2. spectral (Cheeger bounds)

3. semidefinite (SDP) (MAX-CUT, SPARSEST CUT)

Semidefinite relaxations subsume both.

Intriguing connection to hardness reductions, e.g., the unique games conjectures: plausibly optimal polynomial time algorithm.

Need to formalize algorithms. Mathematical programming relaxations: powerful general approach for approximating NP-hard optimization problems.

Formalizes intuitive notion of LP relaxations for problem. Engough structure for unconditional lower bounds. Extends to SDP relaxations (but LP lower bound techniques break down). Not surprising: for some problems, SDP relaxations are strictly stronger.

Test computational complexity conjectures: approximation/PCP: unique games, sliding scale. Averge-case.

We consider the example of MAX-CUT. We give a LP formulations of MAX-CUT. Maximize $f_G(x) = \sum_{ij \in E(G)} (x_i - x_j)^2/4$ over $x \in \{-1, 1\}^n$.

Equivalently, maximize $\sum_{ij \in E(G)} (1 - X_{ij})/2$ over the cut polytope, CUT_n =convex hull of $\left\{xx^T : x \in \{-1, 1\}^n\right\}$. The numebr of facets is exponential, so there is no small direct LP formulation.

The general size n^d LP formulation of MAX CUT: The polytope $P \subseteq \mathbb{R}^{n^d}$ defined by $\leq n^d$ linear inequalities that projects to CUT_N.

Often there is exponential savings: ℓ_1-norm unit ball,...

The cut polytope doesn't have a LP formulation of polynomial size. We want to show this unconditionally.

Known: exponential lower bounds for LP formulations of MAX-CUT. Approximation ration $> \frac{1}{2}$ require superpolynomial size. But the best known MAX CUT algorithms are based on SDP, and have approximation ratios better that $\frac{1}{2}$.

Can we show size lower bounds for SDP formulations of MAX CUT?

A general size n^d-SDP formulation of MAX-CUT: spectrahedron $P \subseteq \mathbb{R}^{n^d} \times \mathbb{R}^{n^d}$ defined by intersecting some affine linear subspace with psd cone.

We show an exponetnial lower bound $2^{\Omega(n^{\cdot 1})}$. Sum of squares is optimal SDP algorithm for MAX CUT.

Upper-bound certificates: algorithm with appoximate guarantee must certify upper bounds on the objective function f_G. We can characterize SDP algorithms by their certificates.

Certificates of degree d sum of squares SDP algorithm: certify $f \geq 0$ for $f : \{\omega 1\} \to \mathbb{R}^n$ iff $f = \sum_i g_i^2$ with $\deg g_i \leq d$.

SoS captures Goemans-Williamson MAX CUT .878 approx:

$$OPT_G - .878 f_G \sum_i g_i^2$$

with $\deg g_i \leq 1$.

Relate general certificates to SoS certificates. For every n^d-size SDP algorithm, can certify $f \geq 0$ iff there exists $P \succeq 0$, for all $x \in \{\pm 1\}^n$, $f(x) = \mathrm{tr} PQ(x)$. For the degree d sum of squares SDP algorithm, $Q(x) = x^{\otimes d}(x^{\otimes d})^T$. A general SDP Q captured by degree d sum of squres if $\deg \sqrt{Q} \leq d$.

Can simulate general n^d size SDP algorithm by degree $O(d)$ (low degree) sum-of-squares.

For every low-degree matrix valued function F¡ can't distiguish between Q and Q' To try to approximate some object, bound the complexity of the approximator. Key technical challenge: $\deg \sqrt{Q'} \gg \deg Q'$ at the heart of sum of squares counterexamples.

Take an indirect approach. find the simplest SDP algorithm Q' that satisfies $\langle F, Q \rangle = \langle F, Q' \rangle$. Measure of simplicity is quantum entropy. There is a closed-form solution $Q'(x) \approx$

$e^{tF(x)}$ where t is entropy defect of Q, at most $\ln n^d$. Then expand $\sqrt{Q'(x)}$ and truncate the power series. Shows degree of apprxomator controlled by F.

Can simulate general small SDP by low-degree SDP.

Open questions:

1. approximation beyond CSP and relatives (rule out .999-approx for TSP by poly-size LP/SDP)

2. strong quantitative lower bounds for approx: rule out .999-approx for MAX CUT by $2^{n^{O(1)}}$-size LP/SDP

3. rule out $2^{n^{.999}}$-size SDP for exact MAX CUT.

4.14 Sub-polynomial cost 2-server PIR

A database has $a = (a_1, \ldots, a_n) \in \{0, 1\}^n$. The user wants to know a_i. the user wants to learn a_i, but the the server shouldn't learn anything about i.

Information theoretic privacy: User behavior should be independent of i. (stronger)

Contrast: Computation PIR, where the server is computationally bounded.

Goal: minimize commuication cost. The obvious way is to request the entire database.

Chor, Goldreich, Kushilevitz, Sudan introduced PIR with multiple servers which cannot communicate.

The user first generates a random string. He queries $Q_j(i, r)$; they send back $A_i(a, Q_i)$. The user calculates $a_i = R(i, r, A_1, A_2)$.

Privacy: the distribution of $Q_j(i, r)$ is independent of i. Communication cost is the total number of bits exchanged. This can be generalied to multiple servers and rounds.

What is known about PIR schemes? For 2 servers the best bound was $O(n^{\frac{1}{3}})$; for k servers, $n^{o(\sqrt{\frac{\ln\ln n}{\ln n}})} = n^{o(1)}$. Can we beat $n^{\frac{1}{3}}$ with 2 servers? To beat this we need new techniques. We achieve the same cost as in the 3-server PIR protocol, $n^{o(\sqrt{\frac{\ln\ln n}{\ln n}})}$.

$\ln n$ is a trivial lower bound. Best known lower bound is $5 \ln n$, relying on quantum ideas. RY06 gave a $\Omega(n^{\frac{1}{3}})$ lower bound for bilinear group-based 2-server PIR capturing function of server answers. Our scheme is blinear and uses a group-based secret sharing schemes not captuerd by RY-model. R-model has extra functionality, can also retrieve certain linear combinations of database bits. Their model is more like LCC; ours is more like LDCs.

There are 2 ingredients.

1. 2-server $O(n^{\frac{1}{3}})$ cost PIR. The encoding is Reed-Muller codes with derivatives.

   Retrieval: polynomial interpolation with derivatives.

2. 3-server $n^{o(1)}$ cost PIR. Encoding is matching vector codes. Retrival is again polynomial interpolation.

We combine these two. The encoding is matching vector codes with derivates, and the retriveal is polynomial interpolation with derivatives.

We need a combinatorial construction.

Definition 4.7: A matching vector family is (u_i), (v_i), $u_i, v_i \in (\mathbb{Z}/m)^k$, with $\langle u_i, v_j \rangle \pmod m = 0$ iff $i = j$.

The servers uses u_i to encode and v_i for retrieval.

Theorem 4.8: Gromulsz gave a MVF over $(\mathbb{Z}/6)^k$ of size n with $k = n^{O\left(\sqrt{\frac{\ln \ln n}{\ln n}}\right)}$.

6 being composite is essential.

Let $(\mathcal{U}, \mathcal{V})$ be a MVF over $(\mathbb{Z}/6)^k$ of size n with $k = n^{o(1)}$. We work ove rthe ring $\mathcal{R} = \mathbb{Z}/6[\gamma]/\langle \gamma^6 - 1 \rangle$, ring with characteristic 6 and element of order 6. Given database a, define

$$F(x) = \sum_{i=1}^{n} a_i x^{u_i}$$

Not really a polynomial since $u_i \in (\mathbb{Z}/6)^k$, but we can evaluate it over powers of γ, since it has order 6: $\gamma^z = (\gamma^{z_1}, \ldots)$. Define $F_\gamma(z) = F(\gamma^z)$, $F_\gamma : (\mathbb{Z}/6)^k \to R$.

We need to define the gradient and hessian of F as $F^i(\mathbf{x}) = \sum_i a_i u_i x^{u_i}$, $F^2(x) = \sum_{i=1}^n a_i (u_i \otimes u_i) x^{u_i}$.

Note that the coefficients of F^1, F^2 are in $\mathbb{Z}/6$, this is why we need R to have characteristic 6. Define F_γ^i similarly.

Servers encode the database as evaluations of $F_\gamma, F_\gamma^1, F_\gamma^2$ at all points of $(\mathbb{Z}/6)^k$.

Retrieval: to find a_τ, pass a line through a random point $z \in (\mathbb{Z}/6)^k$ in the direction v_τ. Consider the restriction of F_γ to this line:

$$F_\gamma(z + tv_\tau) = \sum_{i=1}^{n} a_i \gamma^{\langle u_i, z + tv_i \rangle} = \sum_{l \in \mathbb{Z}/6} \left(\sum_{i : \langle u_i, v_i \rangle = l} a_i \gamma^{\langle u_i, z \rangle} \right) \gamma^{lt}.$$

By the fact it's a MVF, only 1 term has exponent 0. Isolate the index we want on the line. Get more equations from the first and second derivatives.

The protocol picks a uniformly random $z \in (\mathbb{Z}/6)^k$, U sends to S_i, $z + t_i v_\tau$. S sends back $F_\gamma^i, 0 \leq i \leq 2$

Privacy is guaranteed by $z + t_i v_\tau$ uniformly distributed over $(\mathbb{Z}/6)^k$. Communication cost is $O(k^2) = n^{o(1)}$.

Open:

1. Understanding MVF: the current bounds do not rule out $O(\ln n \ln \ln n)$ communication using our approach.
2. Improve lower boudns for 2-server PIR.
3. Does interaction help? All known protocols are single-round.

5 6-17-15

5.1 Limitations of Copper-Winograd method for fast matrix multiplication

Put an identity into a black box to get a speedup (ex. 2×2 matrix with 7 multiplications; $3q + 3$ with $q + 2$, optimize at 6.)

A new technique, Laser method with merging. Laser method with merging applied to T better than Laser method applied to T^M.

Prove limitations on laser method with merging. LMM applied to $T_{CW}(5)^{16}$ cannot prove $\omega < 2.3735$, for $T_{CW}(2.3078)$ cannot prove $\omega < 2.3078$.

Starting point: $R(T_{CW}) \leq q + 2$.

1. Take high tensor power
2. Zero variables and merge tensors obtaining disjoint tensors (use Spencer sets)
3. apply asymptotic sum inequality

What is merging tensors? Sum of non-disjoint matrix multiplication tensors can be equivalent to larger matrix multiplication tensor: $\langle 1, 1, q \rangle + \langle 1, 1, 1 \rangle =$ $an1, 1, q + 1$. Characterize all possible mergings in power of CW tensor.

Question: how large can a subset of $(B^n)^3$ be if

1. every triple satisfies $x_i + y_i + z_i = 2$
2. every two triples satisfy $\langle x_1, y_1, z_1 \rangle, \langle x_2, y_2, z_2 \rangle$ have all corresponding vectors unequal. Answer: $2^{H\left(\frac{1}{3}\right)}$. ($\left(\frac{3}{2^{\frac{2}{3}}}\right)^n$) Exercise!

Our proofs is similar but more complicated.

What next? Get slightly improved upper

- Come up with a new identity. Since 1987 we've been using the same one...
- Group-theoretic method and its extensions (Cohn and Umans).
- Weighted matrix multiplication (Cohn and Umans).

5.2 Locally testable codes

Weakly/strongly locally testable code:

1. poly $\left(\frac{1}{\varepsilon}\right)/O(1)$ queries,
2. $w \in C \implies T^w(z) = 1$.
3.

Main parameter is code length (amount of redundancy).

Goldreich and Sudan: there exists a strong LTC with linear distance and nearly-linear length. Viderman showed there is one with quasi-linear length.

In a LDC, we want to output x_i.

The best known LDCs re only superpolynomial length. Katz and Trevisan showed any q-query LDC must be of length $\Omega(k^{1+\frac{1}{q-1}})$. Introduce relaxed-LDCs which allow output x_i or $\perp$, "I don't know."

1. if δ-cose to C, $\mathbb{P}(D^w(i) \in \{x_i, \perp\}) \geq \frac{2}{3}$.
2. For all w that is δ-close to C and for most $i \in [k]$, $D^w(i) \neq \perp$.

There exist relaxed LDC's with nearly-linear length.

Our gol is to construct short codes that are both strong-LTCs and relaxed-LDCs.

Motivation: the promise required for successfully decoding can be verified by testing procedure.

Theorem 5.1: Exists a binary linear code that is relaxed LDC and strong LTC with ...

Idea: each codeword consists of 3 equal-length parts

1. large distance
2. easy to decode
3. consistency mechanism (PCP of proximity)

Definition 5.2: Verifier is not allowed to read input. PCPPs are PCPs with query access to both proof and input.

- $x \in \implies \exists \pi, V^{x,\pi} = 1$
- x ε-far: $\mathbb{P}(= 0) > \frac{2}{3}$.

Exist PCPPs for NP with nearly linear length and constant query complexity.

π_i is PCPP that asserts the first part of codeword of C is consistent with the ith bit of the second part.

This yields quadratic legnth. With further ideas, nearly linear.

Weak testability: Because there exist PCPPs with nearly-linear length, any linear code can be transformed to LTC at the cost of nearly linear blow-up in the length.

Drawback: results in LTCs are inherently weak

Solution: strong canonical PCPPs.

1. $x \in S \implies !\pi, V^{x,\pi} = 1$
2. Strong soundness: proximity oblivious and must reject any (x', π') with probability poly in distance from $\{(x, \pi(x))\}_{x \in S}$.

Construct scPCPPs for specific statements we need with poly length. C good linear code: eists scPCPP...

How to obtain strong testability?

1. Naive way yields poly length codes.

2. Use alternative approach relying on tensor codes. Use scPCPPs ony for short statements: even wth poly blow-up, length of each proof still sub-linear.

   Local consistency to global consistency.

5.3 Entropy sumset inequality and polynomial convergence to Shannon capacity for all alphabets

Coding for discrete memoryless channels: take a message $m \in M$, encode into a codeword $x \in X^N$, put through noisy channel to received word $y \in X^N$, and decde and output $m^* \in M$. Shannon: there is a threshold $I(W)$, the channel capacity. ε gap. (Ex. X uniform, $I(W) = I - H(W)$. This is symmetric channel capacity. For symmetry channels, exactly channel capacity. Examples. BSC_p, BEC_α.)

Proof: probabilistic method. Random code perfoms well. In order to get $I(w) - \varepsilon$, take block length $N = \frac{1}{\varepsilon^2}$, prob error $e^{-\varepsilon^2 N}$.

Achieve capacity: precise complexty-theoretic formalism:

1. $\mathbb{P}(Dec(W(Enc(m))) = m)$ tiny

2. Block length $N \leq$ poly $\left(\frac{1}{\varepsilon}\right)$

3. Runtime of ENc and Dec poly $\left(\frac{1}{\varepsilon}\right)$.

Seek complexity poy bounded insingle parameter, gap ε to capacity.

Codes close to capacity

1. Forney concatenated codes, decoding complexity $e^{\left(\frac{1}{\varepsilon^2}\right)}$.

2. LDPC codes: approach capacity arbitrarily closely only for erasures

3. Polar codes: achieve capacity for block length $N \to \infty$. $O(N \ln N)$ complexity successive cancallation decorder.

How large take N to be ε within capacity?

First and so far only proven construction to appraoch capacity with poly $\left(\frac{1}{\varepsilon}\right)$.

Extend to q-ary polar codes for $q > 2$. rate $I(W) - \varepsilon$ for $N \geq \left(\frac{1}{\varepsilon}\right)^{c(q)}$.

Key new ingredient: entropy increaes inequality for condition random variables over prime alphabets.

5.3.1 Polar codes: quick primer

Focus on compression (dual viewpoint, cleaner). $(X_i)_{i=0}^{N-1}$ copies of rv supported on $[0, q-1]$.

Goal: compress via linear map to $H(X)N$ symbols over $[0, q-1]$. Now

$$H(U) = H(U_0) + H(U_1|U_0) + \cdots .$$

Find map that's polarizing: conditional entropies at output polarize to 0 or 1.

$$\frac{1}{N}|\{i : H(U_i|U_{<i}) \in (\varepsilon, 1-\varepsilon)\} \to 0$$

as $N \to \infty$. Only $H(X)$ have entropy, and full to close.

Index more or less determined. Suppress output. Only output on non-good indices. If $i \notin$Good, know from encoder. If $i \in$Good, set to more likely bit. Efficiently computed by recursive construction.

Prob doesn't recover is $\le \sum_{i\in Good} H_i \le \delta N$.

Need $N \times N$ matrix ...

Ex. 2 × 2 polarization: $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$. Suppose $X \sim Bernoulli(p)$.

$$\begin{aligned} H(U_0) &= h(2p(1-p)) > h(p) \\ H(U_1|U_0) &= 2h(p) - H(U_0) < h(p). \end{aligned}$$

Recurse: $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes n}$. Map of gates: cf. Fourier circuit, but with functions $V_i + T_i, T_i$. everything boils down to understanding what G_2 does.

W: pair of correlated random variables (A, B), entropy $H(W) = H(A|B)$.

Take wo iid codies (A_i, B_i), output $(A_0+A_1; B_0, B_1)$, $W' = (A_1; A_0+A_1, B_0, B_1)$. $H(W)+H(W') = H(W'')$ chnnel splitting.

Get a Pascal triangle of $W^{\pm\pm\cdots}$. Second moments converge to a limit, monotone convergence theorem.

Entropy increase/no-fixed poitns lemma: If (X_i, Y_i) are iid and $H(X_i|Y_i) \in (\delta, 1-\delta)$, then $H(X_1 + X_2|Y_1, Y_2) \ge H(X_1|Y_1) + \gamma(\delta)$. Second moment increases, converge to $H_\infty = Bernoulli(H(X))$. Entropies polarize in the limit. Poly fst convergence requires $\mathbb{E}[H_n(1 - H_n)] \ll \frac{1}{N} - 2^{-n}$.

Qualititative version. If not δ-near boundary, increase by amount linear in δ.

Binary $q = 2$: Mrs. Gerber's lemma: suffices to consider unconditioned case, check for Bernoulli. General prime q: entropi increase inferred from Sasoglu, but too slow.

Idea: conditions annoying, get rid of them. All possible wys condition Y. Reduce to statement about unconditioned variables. FOr unconditioned, want to show the first statement.

$H(A + B) \ge \max(H(A), H(B))$, $H(A + B)$ ore than some skewed average.

Unconditional entropic inequality. Difficult case: both entropies close to 0 or 1. Almost all weight on 1 symbol. Deviate from uniform distribution, etc. Established poly gap to capacity for polar codes over prime alphabet.

Not hold for non-prime due to existance of nontrivial additive subgroups.

Codes for general alphabets.

Codes with poly gap to capacity for all alphabets. Exponent c in $N(\varepsilon) = O\left(\frac{1}{\varepsilon^2}\right)$ depends on q.

Basic questions about entropy.

$H(X+X') = H(X) + \alpha(q)H(X)(1-H(X))$. Show $\alpha(q) \geq \text{poly}\left(\frac{1}{q}\right)$, $\alpha(q) \leq O\left(\frac{1}{\ln q}\right)$.

Finite group analogye of entropy sumset inequalities. torsion-free case, Tao 10.

Entropic analogs of additive combo theorems. Integers: Abbe, Teleltar.

5.4 List-decoding size of Fourier-sparse boolean functions

Many nice families have sparse fourier expansions: juntas, low-depth decision tree.

In the Boolean case, if f is k-sparse, f has $\mathbb{F}_2$ degree at most $\lg k$.

Learning sparse boolean functions: exactly learn f with running $\text{poly}(n,k)$,

Given uniform and independent random samples from k-Fourier sparse boolean function, learn f. How many samples needed? We don't consider efficiency. Not known whether can do in time $f(k)\,\text{poly}(n)$.

A simple bound is $q = O(nk^2)$ because poly degree $\leq \lg k$. Every 2 fuctions diagress on at least $\frac{1}{k}$ of the inputs by Schwartz-Zippel.

Prob that $g \neq f$ survives $\leq \left(1-\frac{1}{k}\right)^q$. $2^{O(nk)}$ k-Fourier sparse bolean functions. Union bound: Need $2^{O(nk)}\left(1-\frac{1}{k}\right)^q = O(1)$

Known: $O(nk\ln^3 k)$, $O(n^2 k \ln k)$. Restricted isometry property.

Deal with intuition that not too many sparse Fourier coeffs close.

Theorem 5.3: Number of k-Fourier sparse boolean functions of distance at most d from f is $\leq 2^{O(nk)\ln k(d/2^n)}$. For $d = k \approx 2^{\frac{n}{2}}$, bound $2^{O(n^2)}$, tight (indictors of $\frac{n}{2}$ dimensional subspaces).

Learning from samples.

Theorem 5.4: Sample complexity is $O(nk\ln k)$.

Lower bound of $\Omega(k(n-\lg k))$. Only $\ln k$ gap.

Idea: use sampling to show g has succinct representation.

All we need: bounded spectral norm. $\|\widehat{g}\|_1 \leq \sqrt{\frac{kd}{2^n}}$. (Parseval and C-S.)

$$\frac{g}{\|\widehat{g}\|_1} = \frac{1}{\|\widehat{g}\|_1}\sum_{S\subseteq[n]\widehat{(}S)\chi_S}.$$

Convex combo of characters with signs:

$$\mathbb{E}_S[\text{sign}(\widehat{g}(S))\chi_S],$$

weights defined by Fourier coefficients. By Chernoff, $O\left(\frac{nkd}{2^n}\right)$ samples suffice to approx $g(x)$ n every x to additive error $< \frac{1}{2}$.

$\frac{1}{2^n}$ because need approx to be good on all inputs—union bound. g can be represented by a binary string of length $O\left(\frac{n^2kd}{2^n}\right)$.

To get the theorem, modify the algorithm: don't need correct evaluation on all inputs. approx up to $\frac{1}{2}$ for all but $O\left(\frac{1}{k}\right)$ fraction of inputs.

Application to property testing:

Problem 5.5: Given query access to k-Fourier sparse $f : B^n \to \mathbb{R}$, decide whether f is Boolean. For $x \in B^n$, check $f(x) \in B$. How many repetitions do we need? Gur and Tamuz show $O(k^2)$ repetitions suffice: it must be nonzero on many inputs. Uncertainty principle:

$$|\operatorname{Supp}(g)||\operatorname{Supp}(\widehat{g})| \geq 2^n.$$

Apply to $g = f(f-1)$.

Observation: if f non-boolean then a random restriction of f to $O(\ln k)$ dimension subspace of $\mathbb{F}_2^n$ w.h.p also nonboolean.

Improved tester:

1. pick random $n' = O(\ln k)$-dimensional subspace $W \subseteq \mathbb{F}_2^n$. Fourier sparsity can only decrease. Defined on few variables.
2. Learn restriction $f|_W$ assuming it is boolean. If get non-boolean reject. If not consistent with any k-sparse boolean function, then reject. Otherwise get a candidate. Just have to compare with true function. Using roughly k additional samples (others disagree on $\Omega\left(\frac{1}{k}\right)$).

Query complexity $O(n'k\ln k) = O(k(\ln k)^2)$.

We also show a lower bound of $\Omega(k \ln k)$ for 1-sided error.

Open: close gap $nk\ln k$, $k(n - \lg k)$.

Follow-up: improved bound on restricted isometry property of subsampled Fourier matrices.

How many rows from Fourier matri to get whp a matrix have like isometry on k-sparse vectors. Upper bound on this question implies sample complexity in strong manner. More deterministic: allow recovery of all ... simultaneous.

5.5 Non-classical polynomials as a barrier to lower bounds

5.5.1 Nonclassical polynomials: Basics

Classical polynomials, definitions:

1. $f = \sum_{|\alpha| \leq d} c_\alpha x^\alpha$
2. f vanish after taking $d+1$ derivatives along any direction, $D_h f(x) = f(x+h) - f(x)$.

We make several changes. $f : \mathbb{F}_p^n \to T$,

$$f(x) = \frac{1}{p} \sum_{|\alpha| \leq d} c_\alpha x^\alpha.$$

Ex. $p = 2$, $d = 2$.

1. $f(x) = \frac{x_1 x_2}{2} \pmod 1$
2. $f(x) = \frac{x_1}{2^2} \pmod 1$.

The canonical expression, by Tao and Ziegler in their Gowers inverse norm proofs.

$$f(x) = \sum_{k,|\alpha| \le d-k} \frac{c_{\alpha,k}|x|^\alpha}{2^{k+1}} \pmod 1.$$

Write $f = \frac{P_d}{2} + \cdots + \frac{P_1}{2^d}$.

5.5.2 Applications

1. Correlation bounds

   Hard functions against low degree polynomials. For $\deg(f) = d$,

   $$\mathbb{E}[(-1)^{f(x)}\omega^{x_1+\cdots+x_n}] \le e^{-\frac{n}{4^d}}.$$

   Viola and Wigderson's proof techniques extend to nonclassical polynomials.

   $$\mathbb{E}[e(f(x))\omega^{\sum x_i}] \le e^{-\frac{n}{4^d}}.$$

   There is a log-n degree nonclassical poly which correlates well:

   $$\mathbb{E}[e(f(x))\omega^{\sum x_i}] \ge 0.99.$$

   Technique cannot give anything for superlog degrees.

   Construction:$r = 2^k \pmod 3$ for large enough k. Let $A = \frac{r2^k}{3}$: define $f(x) = \frac{A(x_1+\cdots+x_n)}{2^k} \pmod 1$. If $\sum x_i = 3m + q$, then

   $$f(x) = \frac{rm}{2^k} + \frac{rq}{3 \cdot 2^k} - \frac{q}{3} \pmod 1 = \theta_x - \frac{q}{3} \pmod 1.$$

   For $k = 10 \ln n$, the red terms are small.

   Need proof technique that separates classical and nonclassical polys.

2. Correlation bounds (MAJ).

   $$\mathbb{E}[(-1)^{f(x)}(-1)^{\mathsf{Maj}(x)}] \le \frac{d}{\sqrt{n}}.$$

   There is nonclassical f of degree $\lg n$ where this is $\Omega(1)$.

3. Weak OR representation

   $m = p_1 \cdots p_r$. Come up with $P_i : (\mathbb{Z}/p_i)^n \to \mathbb{Z}_{p_i}$, with $P_i(0^n) = 0$ for all i. For $0 \ne x \in B^n$, $P_i(x) \ne 0$ for some i. The degree is $d = \max_i \deg(P_i)$.

   The question for prime m is completely understood. What about composite m?

   Barrington, Beigel, and Rudich showed $O(n^{\frac{1}{r}})$. For primes you need $\Omega(n)$.

   The lower bound is $d = \Omega(\ln^{\frac{1}{r-1}} n)$ by Barrington and Tardos.

   Consider $P(x) = \frac{\sum x_i}{p^{k+1}} \pmod 1$, $p^k \gg n$. $d = (p-1)k+1 = O(\lg n)$. "It looks like you cheated."

   The Barrington and Tardos proof extends. proof technique can't go beyond $\ln n$.

4. PRGs for low degree polynomials.

   Come up with D on B^n,

$$|\mathbb{E}_{x\sim D}[e(P(x))] - \mathbb{E}_{x\sim U}[e(P(x))]| \le \varepsilon.$$

   Viola: seed length $d \ln n + d2^d \ln\left(\frac{1}{\varepsilon}\right)$. Works for nonclassical f. Sum of d copies of small bias genertor with error ε^{2^d} fools degree d polys. Is it tight for nonclassical polys? Does there exist small bias generator with error $\gg \varepsilon^{2^d}$?

VW, BT does not separate. RS do. Viola for PRG?

A litmus test for proof techniques. Check if proof extends to nonclassical, if there is matching bound attained by nonclassical.

Is $|X \cap Y \cap Z| = 0$? $X, Y, Z \subseteq [n]$. Each player sees 2 sets, and want to know whether $|X \cap Y \cap Z| = 0$. How many bits do they need to exchange? It seems like an obstacle to proving stronger lower bounds for multiparty communication.

k players and sets; each knows $k-1$. Deterministic communication lower bound: $\frac{n}{4^k}$.

Lower bound works for quantum protocols, and there is a quantum protocol reaching it. We simplify Sherstov's randomized lower bound.

Note $\frac{k^2 n}{2^k}$ bits suffice. Consider the $2^n \times 2^n$ matrix corresponding to f. Sending a message partitions the matrix into 2 parts corresponding to 0/1. The protocol partitions the matrix into monochromatic rectangles.

A rectangle is a boolean function expressible as a product of 2 boolean functions $R(x,y) = A(x)B(y)$.

Theorem 5.6: If f is computed with c bits then f can be partitioned into 2^c monochromatic rectangles. (f is the sum of 2^c rectangle functions.)

Randomized protocol: nearly 2^c monochromatic rectangles.

Is $|X \cap Y|$ even? requires n bits of information.

Lindsey's lemma: no large monochromatic rectangle. If R is monochromatic and has density δ,

$$|\mathbb{E}_{X,Y}[a(X)b(Y)(-1)^{X\cdot Y}]| = \delta$$

Use C-S repeatedly.

$$\mathbb{E}_{X,Y}[a(X)b(Y)(-1)^{X\cdot Y}]^2 \le \mathbb{E}_X[a(X)^2 \mathbb{E}_Y[b(Y)(-1)^{X\cdot Y}]^2] \le \mathbb{E}_{X,Y,Y'}[b(Y)b'(Y)...]$$

... at most $\mathbb{E}_{A,B}(-1)^{A\cdot B} \le 2^{-\Omega(n)}$. In fact, no large.

is $|X \cap Y| = 0$? requires n bits, even randomized. Ther are large monochromatic rectangles.

Use information complexity. No large 1-monochromatic rectangles.

Information theory stuff does not work in multiparty setting. Cylinder communication: set I,

$$I(x, y, z) = A(x, y)B(y, z)C(z, x).$$

f computed with c bits: This means f is sum of 2^c cyliner functions.

Lemm: no large nearly monochromatic cylinder intersection. (BNS91) Proof similar: square thrice; move the square inside. $\mathbb{E} \leq 2^{-\Omega(n)}$.

This is the proof for generalized inner product. It doesn't work for disjointness. Infomation complexity doesn't seem to work (we don't have the right definitions).

We prove we cannot cover the 1's with nearly monochromatic cylinder intersections. Use distribution, but not hard distribution for protocol. But is very useful in analysis.

1. $4^k \times k$ matrix: eveyr column appears 2^k times except all 1's.
2. $b \in B$ random. Each $sum_i v_i = b \pmod 2$ occurs one additional time.
3. Permute columns.
4. Take multiple (independent) copies of that distribution. Concatenate the matrices.

probability of getting disjoint sets if $\frac{1}{2}$. Most randomness comes from permuting columns.

Prove analogue of Lindsey's lemma. Correlation with parity of whether each part is disjoint is exponentially small. Lower bound for communication complexity. $|\mathbb{E}[I(X_1, \ldots, X_k)(-1)^{\sum_i D_i}]| \leq 2^{-2m}$. The deterministic CC is $n/4^{-l}$.

Randomized lower bound more complicated. Use approximation theory.

$$\left| f((\sum B_i)m/r)(-1)^{\sum B_i} \right| \leq 2^{-12r}$$

OPen: prove anything better than $\frac{n}{2^k}$. Randomized communication lower bound.

5.6 Learning a mixture of gaussians in high dimensions

Input: multi-dimensional data points

Assumption: mixture of Gaussian distributions

Goal: learn weights, means, covariance matrices.

Parameters: weights w_i, means $\mu^{(i)}$, covariance $\Sigma^{(i)}$. Wnat to learn parameters in time poly$(n, k, ...)$. Moitra and Valiant showed that exponential dependence in k is unavoidable in general.

Prior works:

1. general case poly$(n, e^{O(k)^k})$.
2. non-overlapping clusters; spherical $n > k$, independent mean vectors poly(n, k).

   proportional to identity. linearly independent. use tensor decomposition.

3. destiny estimation: 1-dimensional poly(k), higher dimensional, e^n.

Can we learn the parameters with polynomial algorithms for most mixture of gaussians! Yes, worst cases are not everywhere. No additional assumptions like mean separation.

Given an arbitrary instance, nature perturbs the parameters with a small amount ρ of noise.

Given samples from smoothed MoG, learn the smoothed parameters with negligible failure probability $O(e^{-n^c})$ over nature's perturbation. Escape from degenerate cases, become well-conditioned.

learn up to accuracy ε, for high enough dimension $n = \Omega(k^2)$. Fully poly time and sample complexity poly$(n, k, \frac{1}{\varepsilon)}$.

Match the first 6 moments. Decompose the moments tensor. Apply spectral methods. Why does high dimension and smooth help us learn.

1. enough number of moment matching constraints for identifiability. Number of free parameters $\Omega(kn^2)$ less than moment-matching inequalities. 6th moments $\Omega(n^6)$.

2. Enugh randomness in nature's perturbation model for wel-condition Gaussian matrix with prob at least $1 - O(\varepsilon^n)\sigma_m(X) \geq \varepsilon\sqrt{n}$.

Learn 0-mean MoG. Hsu and Kakade: construct lowe rank tensor from the moments entsor:

$$M_2 = \mathbb{E}(xx^T) = \sum w_i \mu^{(i)}(\mu^{(i)})^T + \sigma I_n$$
$$M_3 = ... = \sum_{i=1}^{k} w_i (\mu^{(i)})^{\otimes 3} + \cdots$$

Apply low rank tensor decomposition. $\mu^{(i)}$'s are independent.

$$X_4 = \sum_{i=1}^{k} w_i vec(\Sigma^{(i)}) \otimes vec(\Sigma^{(i)})$$
$$X_6 = \cdots$$

M_4 involves more symmetries: number less thannumber of distinct elements in X_4. Each is linear comb of entries in low rank X_4. Similar for X_6.

$M_4 = \mathcal{F}_4(X_4)$.

Looks like matrix sensing, but standard method does not apply. Exploit low rank property of X_i to unfold M_i? First learn row-span. Do change of variable to reduce number of free variables. Solve system of linear equtions to get new value, as well as previous low-rank matrix.

1. Find the span of $vec(\Sigma^i)$'s.

2. Use span to change variable and unfold M_i to get X_i.

3. Low rank tensor decomposition to recover $vec(\Sigma^i)$'s.

Each is poly time (and samples) and poly stable.

For general MoG, in subspace span$\{\mu^{(i)}\}^{\perp}$, find $\Sigma^{(i)}$ using previous.

We provide a fully poly algorithm under smoothed analysis. Can we relax $n \geq \Omega(k^2)$ by using hihger order moments? Other hard problems in learning?

Random matrix theory gave new perspective on old problems.

5.7 Learning Ising models

what you do once you have the model. Historically people knew what models to use, ex. HMM of speech recognition

Modern applications: unknown structure.

Given data, how to estimate model

Sample complexity; compuational complexity: how does run-time scale with p and n?

Once have graph, learning parameters is easy.

Theorem: can learn graph with $\mathbb{P} \to 1$ with $O(\ln p)$ samples in $\widetilde{O}(p^2)$ time.

Challenge 1: independence of neighbors.

Ex. $\theta_{12} = -\ln\cosh\theta$. $X_1 \not\perp X_2$. Triangle of dependence.

Challenge 2: long-range dependence. Effects can add up. Reason why there is a phase transition.

Neighbors can be independent while faraway nodes can be strongly dependent.

An expenseive algorithm gives a baseline: exhaustive search. Try to find a neighbor. Test all size d neighborhoods. BMS08. Complexity is large. Can we do better.

Go back to the first model on structurelearning: the tree model.

Challenge 1: weak dependence between neighbors. In trees,neighbors are always correlated.

Challenge 2: long-range dependence In trees, correlations decay with influence.

Add in edges according to mutual information. Don't add a cycle. Chow-Liu: cnlearn tree with probability → 1 using $n = O(\ln p)$ in time $\widetilde{O}(p^2)$ (mutual info between every pair of nodes)

fixed parameter tractability: p^c independent of d.

Correlation decay:

1. independence of neighbors: assume neighbors are correlated: $\mathbb{E}X_iX_j \geq \kappa$ for $\{i,j\} in E$.
2. challenge 2: assume model satisfies correlatin decay prop $\mathbb{E}X_iX_j \leq e^{-\gamma d(i,y)}$.

Weak couplings, high temperature Stronger couplings at low temperature.

A picture looks like stronger coupling: no correlation decay. Images.

Maybe existing algorithms work? The ones that don't assume are based on convex optimization.

Mento-Montanari 2009: All known low-complexity algorithms explicitly or implicitly require correlation decay.

Strongly repelling: $\widetilde{O}(p^2)$ hard-core models. Strong long-range dependence. Possible to learn.

A simple algorithm learns any Ising model!

Key structural property of Ising models: influential neighbor lemma: every node i has at least one neighbor with large influence (also true conditioning on an arbitrary set of nodes).

Deals with challenge 1.

Challenge 2: prune neighbors.

runtime dominated by search over nodes u, $\widetilde{O}(p)$ per node, $\widetilde{O}(p^2)$ total.

Another interpretation of overcoming challenge 2: addint them anyway. Once you've conditioned on well chosen set of variables, you do have correlation decay. Conditional correlation decay.

Conclusions:Simple algorithm learns Ising models without correlation decay, without exhaustive search. New information theoretic structural property for Ising.

5.8 Arithmetic circuits

Theorem 5.7: There is an explicit family of polynomials such that a representation

$$f_d = \sum_{i=1}^{s} \prod_{j=1}^{m} Q_{ij}$$

with Q_{ij} degree 1 and involving $\leq \sqrt{d}$ vars has $s \geq 2^{\Omega(\sqrt{d} \ln d)}$.

For a generic f_d, $s \geq 2^{1.5 d \lg d}$.

The same statement also holds for $f_d = IMM_d$. This is bad news: our techniques can't give better lower bounds because there is a matching upper bound for IMM. Answers another question on efficient parallel computation.

Arithmetic circuits are informally the most natural way to compute polynomials.

1. Can explicit polys be efficiently computed? ($VP \overset{?}{=} VNP$)
2. Can computation be efficiently parallelized? How efficiently can we simulate circuits of size s by circuits of size Δ.

Theorem 5.8: Any circuit of size s and degree d cn be simulated by a Δ-depth circuit of size $s^{O(d^{\frac{1}{\Delta-1}})}$.

We can trade off size for structure (regular homogeneous circuit: $s^{O(d^{\frac{2}{\Delta}})}$.

Is this optimal? For $\Delta = 3$ yes, but with caveats.

Strong enough lower bounds for low-depth circuits imply VP$\neq$VNP.

(What is the sweet spot? what depth? homogeoneus?)

Low-depth circuits are easy to analyze. Lots of work on lower bounds for low depth arithmetic circuits in recent years: hope to disvoer general patterns and technical ingredients.

Proof strategy: for $T_i = \prod_j Q_{ij}$,

1. find a geometric property GP of the T_i's.
2. Express the property GP in terms of rank of big matrix M: if T has property than rank($M(T)$) is small.

3. show that $\mathrm{rank}(M(f_d))$ is large.

   If a matrix $M(f)$ has a large upper triangular submatrix, then it has large rank.

   If the columns of $M(f)$ are almost orthogonal then $M(f)$ has large rank (Alon).

T is a product of low degree polynomials iff $V(T)$ is a union of low-degree hypersurfaces iff $V(T)$ has lots of higher-order singularities, iff $V(\partial^{=k}T)$ has lots of points.

How can we convert geometric propery to the rank of a matrix? $V = V(f_1, \ldots, f_m)$ is a variety. let $\mathcal{G}_\ell$ be the set of degree ℓ polynomials.

Hilbert's theorem: If V is large then $G_\ell(V)$ has small dimension. If V has dimension r then

$$\mathcal{I}_l(V) = ...$$

has asymptotic dimension $\binom{n+l}{n} - \Theta(l^r)$.

Hilbert's theorem can be generalized for algebraic restrictions of polynomials.

If Q is a sparse polynomial then for a suitable random algebraic restriction $Q \pmod{\mathcal{I}}$ is a low-degree polynoimal. If T is a sum of product of low arith polynomials, this also hold. Yields lower bounds for homogeneous depth 5 with low bottom fanin.

Shpilka-Widgerson: a depth three circuit of size s cn be converted to a homogeneous depth 5 circuit C of size $s2^{\sqrt{d}}$. Preserves bottom fanin.

There is a meta-strategy common to any recent and older lower bounds. We don't understand the power or limitations of this meta-strategy.

Open: lower bounds for homogeneous depth 3 circuits for polynomials of degree $\gg n$.

5.9 Depth 5 lower bounds for iterated matrix multiplication

$IMM_{m.d}$ is $(1,1)$-entry of the product of d matrices of size $n \times n$. The number of input variables is $N = d \cdot n^2$. A monomial in IMM corresponds to a path in a layered graph. (Two consecutive layers represent a matrix.)

Can IMM be made highly parallel. i.e., efficiently computed by small depth circuit?

VP vs. VNP. VNP coefficient of any given monomial efficiently computable. Are there hard polys in VNP require super-poly size circuit?

Fan-in bounds: $\Sigma^a\Pi^b\Sigma^c\Pi^d$.

Any $size(f) = \mathrm{poly}(N, d)$ implies f computed by depth 4 $\Sigma\Pi\Sigma\Pi^{\sqrt{d}}$ circuit. If $f \in VP$, f has $N^{O(\sqrt{d})}$ size depth 4 circuit. Loer bound $N^{\omega(\sqrt{d})}$ implies VP$\neq$VNP.

IMM lower boudn $N^{\Omega(\sqrt{d})}$. Depth reduction tight at 4.

Depth 5: better depth reduction at depth 5? $f \in VP$ implies $\mathrm{size}(f) = \mathrm{poly}(N, d)$.

Our result: $N^{\Omega(\sqrt{d})}$ for IMM, circuit $\cdots \Sigma^{N^\mu}$, $0 \le \mu < \frac{1}{2}$. Better depth reduction not possible at depth 5 if bottom fan-in is restricted.

Generic strategy: define a complexity measure Φ maps each polynomial to a number. Dimension of projected shifted partial derivatives (DPSP). M is set of multilinear monomials of degree k, S for degree l.

$$DPSP_{k,l}(f) = \dim_{\mathbb{F}} \mathrm{span}(proj(\{\beta\partial_\alpha f : \alpha \in M, \beta \in S\})).$$

Process: randomly select a set of variables V according to some nice distribution. Set variables outside to be 0.

A small circuit is simplified (small Φ). The polynomial still remains complex: large Φ.

Circuit decomposition lemma: depth 5 circuit: C bottom fanin $N^{\mu 1(0,\frac{1}{2}]}$

$$C|_V = C'$$

C' is $\Sigma\Pi\Sigma\Pi^{O(\sqrt{d})}$ and $DPSP_{k,l}(g) = 0$ for any k, l.

Small upper bound on DPSP.

Expand bottom two layers of gates.

Q_{ijr} has 3 type of monomials

1. small, total degree $< 2\sqrt{d}$: put into C'
2. large, exists var with degree ≥ 3: DPSP is 0. (put to g)
3. Large, each variable has degree ≤ 2, support $\geq \sqrt{d}$.

Union bound fails for $\mu \geq \frac{1}{4}$. Instead, apply mild restriction, makes bottom fan-ins $\leq N^{\frac{1}{4}}$. Apply desired full restriction: analyze as above.

non-Chernoff concentration bound.

dimension of matrix is $d^q \times d^q$ where q is large constant depedendent on fain-in p.

Large upper bound on DPSP(IMM), small upper bound on DPSP($C|_V$)

IMM not efficiently compuetd by bottom fan in restricted,

Prove lower bound for $\Sigma \cdots \Sigma$...

5.10 From independence to expansion and back again

Find a k-independent function that can be efficiently computed (time/space).

A data structure for representing a k-independent function $f : [u] \to [r]$ with evaluation time $t < k$ must use at least $ku^{\frac{1}{t}}$ words of space.

Main result.

Theorem 5.9: There is a randomized data structure for representing a k-independent function $f : [u] \to [r]$ with space $O(ku^{\frac{1}{t}}t)$ and evaluation time $O(t \ln t)$.

Previous results: polynomials, graph powering, recursive tabulation.

Constructions of k-independent families of functions based on bipartite expander graphs. Neighbor function $\Gamma : U \to V^d$.

Definition 5.10: k-unique if for every $S \subseteq U$ with $|S| \leq k$ there exists $v \in V$ with exactly one neighbor in S.

Lemma 5.11 (Siegel)**:** $f(x) = \sum_i h(\Gamma(x)_i) \pmod r$ defines k-independent family of functions.

A k-unique $\Gamma : U \to V^d$ gives a k-independent family.

Most graphs give an optimal space-time tradeoff for k-independent hashing.

What's the problem?

1. Explicit k-unique graphs with optimal parameters are not known.
2. Storing a random Γ defeats the purpose of k-independent hashing (use less space!).
3. Verifying a given Γ is k-unique is infeasible.

Instead of sampling a random function, sample a function from a k-independent family. A k-independent function $\Gamma : U \to V^d$ with $|U| = u_i$, $|V = O(ku^{\frac{1}{t}}t)$ and $d = O(t)$ is K-unique with high probability.

Start with a k-unique function/graph. Increase the domain using a graph product. Define a k-independent family. Sample $\Gamma \in \mathcal{F}$ and use to represent a neighbor function. Back to optimal parameters!

Randomized recursive construction of k-unique function.

$$\Gamma(ax) = \bigoplus_i h(a, \Gamma(x)_i).$$

If Γ is k-unique over Σ^j and Γ is k-independent over Σ^{j+1}, then Γ is k-unique over Σ^{j+1} whp. Space $O(ku^{\frac{1}{t}}t^2)$, time $O(t^2)$.

Divide and conquer. Recurse on each character.

Definition 5.12: k-maj-unique if for every $|S| \leq k$, exists $x \in S$ such that majority of vertices in $\Gamma(\{x\})$ have unique neighbor in S.

If Γ is k-maj-unique, $\Gamma(x_1x_2)_i = \Gamma(x_1)_i\Gamma(x_2)_i$ is k-unique over Σ^2.

We need to find (x_1', x_2') with unique neighbor. Find $x_1'\ in\, \{x_1 : (x_1, x_2) \in S\}$ with $> \frac{d}{2}$ unique neighbors. Find $x_2'\ in\, \{x_2 : (x_1', x_2) \in S\}$ with $> \frac{d}{2}$ unique neighbors.

View $x \in [u]$ as string of 2 chracters and recurse on each.

Technique: graph products and alternating between expansion and indpendence.

Open: Optimal expanders without k-independence?

5.11 Super-resolution, extremal functions and the condition number of Vandermonde matrices

Mathematical and algorithmic problem, connect to tools in analytic number theory.

Background and motivation from physics.

There are fundamental limits to physics from optics: Rayleigh, Abbe limit.

Can we recover find-grained structure from coarse-grained measurements? 2014 Nobel prize in chemsitry to super-resolution cameras, led to insights on biology.

Mathematical framework, Donoho, 1991. Consider a superposition of k spikes, each f_j in $[0, 1)$. Signal

$$x(t) = \sum_{j=1}^{k} u_j \delta_{f_j}(t).$$

Recover using coarse measurements. Integrating exponential. Get an evaluation of an exponential sum. Only low freq: $|\omega| \leq m$.

$$v_\omega = \sum_{j=1}^{k} e^{i2\pi f_j \omega} + \eta_\omega$$

Noise vs. noise-free setting.

When can we recover the coefficients and locations from low frequency measurements?

Prony's method, Pisarenk, Matrix pencil. When no noice, there is a polynoimal time algorithm to recover u_j, f_j with $m = 2k+1$, at $\omega = -k, \dots, k$.

What is possible in the noise-free vs. noisy setting is different!

Get an estimator $\widehat{\omega_j} \to u_j, \widetilde{f_j} \cdots$ which converges at a polynomial rate in $|\eta_\omega|$.

There is a poly time algorithm for noisy super-resolution if $m > \frac{1}{\Delta+1}$ separation condition. Let d_w be wraparound distance in $\mathbb{R}/\mathbb{Z}$. $\Delta = \min_{i \neq j} d_w(f_i, f_j)$.

$$\min_\sigma \max_j |\widehat{f_{\sigma(j)} - f_j}| + |u...| \le \varepsilon$$

poly! drive to inverse poly level and get recovery.

Sharp threshold. As soon as $\frac{1-\varepsilon}{\Delta}$. There is pair of separated exponentially close. Fejer kernel.

Asymptotic bounds for $m = \frac{1}{\Delta}$ on grid.

Convex program for $m \ge \frac{2}{\Delta}$ with noise.

Noise-free case: What would make it noise-stable? Let $\alpha_j = e^{2\pi i f_k}$, V_m^k be the Vandermonde problems. It plays a key role in exact inverse problems; poly interpolation, sparse recover, inverse moment problems...

We create 2 matrices and set up generalized eigenvalue problem. $A = V D_u V^H$. Entries of A correspond to measurements with $\omega \in [-m+1, m]$. Similarly, companion matrix $B = V D_u D_\alpha V^H$. Similarly for B.

Generalized eigenvalue problem gives oslution. The only things that work are $\lambda = \frac{1}{\alpha_j}$.

Noise stability?

Vandermonde matrix has full column rank iff α_j's are distinve. This is enough for noise-free recovery. When is matrix pencil method robust to noise? When is generalized eigenvalue problem. Steward, Sun: various stability bounds for generlized eigenvalues/vectors based on condition number.

Show phase-transition for condiiton number of Vandermonde matrix.

Use extremal functions to bound the condition number of Vandermonde matrix. Used to prove sharp inequalities on exponential sums in analytic number theory.

$$\left\| V_m^k u \right\|^2 = (m - 1 \pm \frac{1}{\Delta}) \left\| u \right\|^2 .$$

Tight by Fejer kernel. Smallest SV is exponentially small.

Beurling-Selberg majorant. A function always above sign function and smooth, closest.

1. $\mathrm{sign}(\omega) \le B(\omega)$.
2. $\widehat{B}(x)$ supported in $[-1, 1]$.
3. $\int_{-\infty}^{\infty} B(\omega) - \mathrm{sign}(\omega)\, d\omega = 1$

inverse theorem in F analysis

$$\left(\frac{\mathrm{sign}(\pi\omega)}{\pi} \right)^2 (\cdot)$$

Count primes in some interval: majorize and minorize interval and lower and upper bound how fall inside. Once have majorizer ad minorizer, easy.

Take a step back. Many inverse problems are well-studied in the exact case. When is the solution robust to noise?

1. polynomial interpolation: Lagrange interpolation. $[p_i]V = [p(\alpha_i)]$: find

   Highly unstable, often need exp small error in degree. Statement about reals!

   Ex. roots of unity, FT,well-conditioned.

2. Sums of exponentials: Moitra and Saks. become well-posed in complex plane. If known $f(x)\pm$noise in circle, promise $f \le 1$ in larger: Hadamard 3 circle!

   Noisy inverse problems are better posed over complex plane.

   Connections between test functions in harmonic analysis and preconditionrs.

   Give a way to obliviously rescale rows of unknown Vandermonde.

6 6-18-15

6.1 Streaming interactive proofs

Given g, replaced g with mulilinear $\widetilde{g}$ over large field; it has good error-correcting properties.

View $\mathbf{x} : B^{\lg n} \to B$. Let $|\mathbb{F}| \ge 4 \lg n$, $\tilde{\mathbf{x}}$ be the multilinear extension. V evaluates $\tilde{\mathbf{x}}(r)$ in streaming pass over $\mathbf{x}$ with space $O(\lg n \lg\lg |\mathbb{F}|)$. Is a sketch of input. Write $\tilde{\mathbf{x}}(r) = \sum_j x_j \widetilde{\delta_j}(r)$.

Sends line to prover. λ specified with $\lg n$ elements. Prover sends back the polynomial restriction to line. Making a claim about every point on the line. If the prover sends any other polynomial, is disagrees almost eveyrwhere. Only know r is somewhere along the line and not equal to i.

RangeCount protocol: fix $[n], R \subseteq 2^{[n]}$.

Input: $x_i \in [n]$, range $R^* \subseteq R$. Goal to output $|\{i : x_i \in R^*\}|$. Reduce it to the index problem. Verifier creates a derived stream. Define over different data universe. Every time see new stream update, insert every single range that contains x_i.

Wat frequency of R^* in derived stream: answer with generalized INEX protocol. But V requires $|R|$ time per stream update.

Use a higher-degree extension.

Online interactive proofs (communication model). $O(1)$-round SIPs can solve INDEX. 2 new hierarchies of communication models

1. $\text{OIP}_+[k]$ can simulate all k-message SIPs.

2. $\text{OIP}[k]$ weaker, but can still simulate all known SIPs, and capture the fundamental way SIPs differ from IPs.

AM^{cc} [BFS86]. Alice and Bob want to ocmpute $f(x,y)$. Merlin helps but is untrusted. Random coins are sent to both parties. Merlin gives answer, then communicate in deterministic protocol.

Restricted private coin variant. Secret coinc not seen by Merlin. Alice sends a single message to Bob. Then Bb and Merlin engage in k-message interaction.

In $OIP[k]$: for Bob and Merlin's message to be independent. Bob and Merlin engage in k-message interaction in step 2, not 3.

$OIP[2]$ protocol of cost $O(\lg n \lg\lg n)$ for INDEX.Toss secret coins to determine eval point. Send λ. Alice sends $\widetilde{x}(r)$.

$OIP^{[k]}$ class of functions solved by polylog cost $OIP[k]$ protocols, AM^{cc} functions solved by polylog cost AM^{cc} protocols. Containment with exonential separation:

$$OIP^{[1]} \subset \cdots \subset OIP^{[\geq 4]} = AM^{cc}.$$

2-message OIP are equivalent to $R^{[2,B]}$: without Merlin: Bob sends single to Alice, Alice sends single response. $OIP^{[2]}$Lpretend Bob can send 1 message to Alice even though lets neither Bob nor Merlin talk to Alice. 4: both Bob and Merlin can talk to Alice: simulate all of AM^{cc}.

6.2 Identifying honest EXP^{NP} oracle among many

We have two oracles, one is honest, one is dishonest.

1. Notion of selector: there exists a selector iff we are able to remove short advice.
2. Existence of selector for EXP^{NP}-complete languages.

Instance checker for f checks if given oracle correctly computes $f(x)$ in poly time.

1. There are instance checkers for $P^{\#P}$, PSPACE, EXP-complete.
2. Must be in $NEXP \cap coNEXP$. Note $NEXP \subseteq EXP^{NP}$.

Probabilistic selector for SAT. Make queries: is ψ satisfiable. The honest oracle gives correct answers; the dishonest oracle gives arbitrary answers.

If the oracles agree, then the answer is right. The essence of the task is to determine which is honest when they disagree.

Definition 6.1: A **selector** S for L is prob oracle machine such that if $A_0 = L$ or $A_1 = L$, then

$$\mathbb{P}[S^{A_0,A_1}(x) = L(x)] \geq 0.99.$$

For a deterministic selector, deterministic succeeding always.

Selector for P^{NP}-complete languages: lexicogrphically maximum satisfying assignment. Output kth bit of lexicographically max satisfying assigment of φ.

Given (φ, k) and 2 oracles, make queries (φ, i) to the oracles. If $v_0 = v_1$, output the kth bit. Else WLOG $v_0 < v_1$. If $\varphi(v_1) = 1$, 1 is honest, else 0 is honest.

EXP^{NP}-complete language: succinctly described Boolean formula $\Phi : B^{2^n} \to B$ and k. Output kth bit of lexicographically maximum satisfying assignment. The proof strategy is the same.

1. Is $V_0 < V_1$? binary search and PIT.
2. Is V_1 a satisfying assignment of Φ? Similar to $MIP = NEXP$.

Instance checker vs. selector. If we have a instance checker, we can construct a selector. The task of selectors is strictly easier than instance checking.

$$\begin{aligned} \mathsf{SAT} \in P/\log &\implies \mathsf{SAT} \in P \\ EXP \in BPP/\log &\implies EXP \subseteq BPP. \end{aligned}$$

This follows from the instance checkability of EXP-complete languages.

We can remove short advice when we have a selector.

If there is a selector M, we can remove 1-bit advice. The two oracles gives answer $M(q, 0)$ and $M(q, 1)$.

Theorem 6.2: For any paddable L, TFAE:

1. $\exists$ deterministic selector
2. for any oracle R, $L \in P^R/\log$ implies $L \in P^R$

Even given many oracles, we can still identify an honest oracle. Idea: tournament. Depending on their answer, divide them into 2 teams. Pick 2 and run, eliminate those we doubt.

If no selector exists, then for some relativized world, even 1-bit advice cannot be removed. Construct the oracle by diagonalization.

Other results:

1. There exists a deterministic selector for PSPACE-complete languages.
2. Any languages with deterministic selector in PSPACE.

Upper bound for probabilistic selector: S_2^{exp}.

Close the gap between EXP^{NP} and S_2^{exp}.

Poly advice?

6.3 Nonadaptivity helps testing juntas

A k-junta is a function only depending on k bits.

For $k = 1$, f is a dictator. How can we tell if f is a k-junta or ε-far?

We use the query model.

1. Nonadaptive: fix queries in advance.
2. Adaptive: choose queries based on answers.

How much more powerful are adaptive queries?

Distingish whether f is

- k-junta, or
- ε-close to a k-junta.

Minimize query count q in terms of k and ε.

Junta testing motivation: Boolean function version of finding a low rank model for high-dimensional data. For $k = 1$, this is dictatorship testing, a basic topic in hardness of approximation.

Prior work:

1. Nonadaptive: $O(k^{\frac{3}{2}}(\ln k)^3/\varepsilon)$.

   Lower bound $\Omega(k \ln k), \Omega\left(\frac{k}{\varepsilon \ln\left(\frac{k}{\varepsilon}\right)}\right)$.

2. Adaptive: $O(k \ln k + \frac{k}{\varepsilon})$.

   Need $\Omega(k)$ queries.

The upper adaptive bound almost matches the lower nonadaptive bound.

Does adaptivity even help? It should: Blai's algorithm uses binary search, which is adaptive. Adaptivity also helps for testing signed majority functions and read-once width-two OBDD's. Adaptivity algorithms use binary search.

We show that adaptivity does help by showing a non-adaptive lower bound. For any $0 < c < 1$, a nonadaptive algorithm requires

$$q = \Omega_c\left(\frac{k \ln k}{\varepsilon^c \ln(\ln k/\varepsilon^c)}\right)$$

Taking $\varepsilon = \frac{1}{\ln k}$, adaptive UB is $O(k \ln k)$, nonadaptive LB is $\frac{k(\ln k)^{1+c}}{\ln \ln k}$.

Basic ideas come from CG04's $\Omega(k)$ adaptive lower bound. We give a new analysis of Bla08's LB.

CG04 considers 2 distributions on $n = k + 1$ variable functions.

1. D_{yes}: pick $i \sim U_{k+1}$, set $f_{yes} : B^{k+1} \to B$ uar subject to not depending on i. (k-junta)
2. D_{no}: $f_{no} : B^{k+1} \to B$. Usually far from k-junta.

Need $\Omega(k)$ queries to distinguish these distributions.

How to distinguish? See if f has irrelevant coordinates.

1. pick x uar.
2. query f on $x, x \oplus e_i$ (i-twin). If not equal, output relevant. Repeat $10 \lg k$ times. Output irrelevant.

If i is relevant, $f(x), f(x \oplus e_i)$ are independent: conclue relevant after $O(1)$ i-twins. If i irrelevant, will query $O(\lg k)$ i-twins. Query cost: $(k+1)O(1) + O(\lg k) = O(k)$.

$\Omega(k)$ lower bound. Suppose query f on $x_1, \ldots, x_q$. Is i relevant? To tell, $x_1, \ldots, x_q$ must have an i-twin. Note q points can have i-twins for $\le q-1$ coordinates.

Can't plan this in advance: $x_1, \ldots, x_q$ need $O(\lg k)$ i-twins in all $k+1$-directions. $q = \Omega(k \lg k)$ adaptive LB? (Not quite true: there is an algorithm that does better, nonadaptively.)

Frankl83: There are $q = O\left(\frac{k \ln k}{\ln \ln k}\right)$ points $x_1, \ldots, x_q$ with $\ln k$ i-twins for each i.

We show this is optimal and extend to general ε.

New distributions:

1. Pick $i \sim \{1, \ldots, k+1\}$ uar., etc.

2. D_{no}: $f_{no} : B^{k+1} \to B$ is random ε-biased function.

Bla08: edge-isoperimetric inequality: points $x_1, \ldots, x_q$ can only have $O(q \lg q)$ i-twins. This only care about total number of i-twins, but there could be few directions have lots of i-twins. We want an edge-isoperimetric inequality about most directions.

Our main tool is a new, more fine-grained, edge-iso inequality.

Theorem 6.3: If $x_1, \ldots, x_q$ have m i-twins in d-directions, then $q \ge \frac{md}{\lg m}$.

Set $m = \lg k, d = k$.

Proof technical.

- Answer you get for single query $f(x_j)$ is not too important.
- Analyze a specific martingale with respct to $f(x_i)$.
- Use McDiarmid's inequality with bad events.

Open problem: Prove a separation between adapative and nonadaptive when ε is constant. (Our separation is when $\varepsilon = \frac{1}{\ln k}$.)

6.4 Hard-to-cover CSPs

Define CSP's. Example $3NAE_q$: if 3 values are distinct (in $\mathbb{F}_q$).

Given a CSP, find the minimum number of assignments such that for every C_i, there is at least one assignment satisfying C_i.

It is NP-hard to find the covering number. Approximating better than 2 is NP-hard.

This is a natural optimization problem, and related to the chromatic number of hypergraphs.

Consider the $3NAP_q$ instance. Define the constrained hypergraph where the vertices are x_i and the 3-sets are those in the same constraint. If the chromatic number is χ, the covering number is $\lfloor \log_q \chi \rfloor$. Given a coloring, express each x_i in q-bit strings. For any edge, one assignment will have them distinct.

Classification: P is odd if $\bigvee_i P(x+i)$ true. 3-SAT, 3-XOR are examples. 3-NAE$_3$ and 4-XOR are not odd.

For odd predicates, the covering number is at most q. Distinguish between 1 and 2 is checking satisfiability. **Shaefer's dichotomy** gives a characterization to distinguish between the 2 cases for boolean CSPs.

The covering number is unbounded in general, $O(\lg m)$. How well can we approximate the covering number?

Our work is based on the Unique Games conjecture. (Definition: Given a bipartite graph $G = (U, V, E, \pi)$,...)

Conecture: NP-hard decide between

1. there exists c-covering. ex. $c = 10$

2. No assignment satisfies $> \varepsilon$ of constraints.

(Guruswami GHS02, relation to hypergraph coloring.) Diur-Kol13 for general CSPs.

Covering number of 4-LIN is NP-hard to approximate within any constant factor. Covering number of non-odd predicates supporting balanced pairwise indep is hard to approx given UGC.

Result: assume covering UGC(c) for some c¡ then for any constant q, k, for any non-odd $P \subseteq [q]^k$, covering number of P-CSP instance cannot be approximated within any constant factor in polynomial time.

NP-hard to approximate: Marginals of both $\mathcal{P}_0, \mathcal{P}_1$ uniform on each of k coordinates. Every $a \in \text{Supp}(P_0)$ even partiy, 1/odd.

Start with UG instance. Ask provers to write down labels over some alphabet q. Verifier queries this proof at some locations, checks queried values, accept or reject based. Based on dictatorship test.

Suffices to show for P non-odd, $NAE \supseteq P \supseteq \left\{a + \bar{b} : a \in NAE, b \in [q]\right\}$, others follow by reduction.

2-covering property makes it easier to apply the invariance principle. IS analysis makes the proof easier to analyze.

Open:

- Characterization based on $P \neq \mathsf{NP}$?

- 1-covering (satisfiability) vs. c-coveirng

- UGC/SSE implies covering UGC?

6.5 Subexponential size hitting sets for bounded depth multilinear formulas

Intro to polynomial identity testing.

1. depth 3 circuits, bounded top fan-in: polynomial time, BB.

2. multilinear depth 4, bounded top fan-in: polynomial time, BB (Saraf-Volkovich)

3. multilinear depth 3, bounded distance: quasi-polynomial, BB (Agrawal-Gurjar-...-Saxena)

We give hitting sets for bounded depth multilinear formulas with

$$|\mathcal{H}| = 2^{\widetilde{O}(n^{\frac{2}{3}+2\delta/3})}$$
$$|\mathcal{H}| = 2^{\widetilde{O}(n^{\frac{2}{3}+4\delta/3})}$$
$$|\mathcal{H}| = 2^{\widetilde{O}(n^{1-\frac{1}{e^d}})}$$

, depth 3/4 formulas of size 2^{n^δ}, regular depth d fomulas of size $2^{n^{1/e^d}}$.

We give a reduction to read once algebraic branching progrms.

Depth 3 multilinear formulas $\Sigma\Pi\Sigma$. If every linear function has only 1 variable in its support, the polynomial is computed by width M ROABP.

What if we had a partition $S_1, \ldots, S_k$ such that all vars in S_i appear in different linear functions?

Think of $\mathbb{F}(S_2 \cup \cdots \cup S_k)[S_1]$.

Partition vars into $n^{1-\varepsilon}$ disjoint sets $S_1, \ldots, S_{n^{1-\varepsilon}}$ (hope intersection of eveyr linear function with every S_i contains ≤ 1 variable). Plug in hitting sets for width M ROABPs on the variables of each S_i.

What about linear functions which contain lots of variables? How to find the partition?

1. Get rid of linear functions with large support. $\frac{\partial}{\partial x_1}$ gets rid of bottom... Exists variable x_i which works for $n^{-(1-\varepsilon)}$ bad ones. Take derivatives wrt $\leq n^{1-\varepsilon} \ln M$ variables. No linear function remains with more than n^ε variables.

   How to get black box access to $f_{x_1,\ldots,x_{i_t}}$.

   Applied repeatedly, each query to derivative simulated by 2^t queries to f.

2. Settle for less. If we partition the variables randomly to $n^{1-\varepsilon}$ sets $S_1, \ldots, S_{n^{1-\varepsilon}}$ then with high probability, for every S_i,

   (a) no linear function intersects S_i in $> n^\delta$ vars

   (b) On every × gate, number of linear funcs which intersect S_i in ≥ 2 vars $\leq n^\delta$. (Brute force expand linear functions to get ROABP of width Mn^{n^δ} in vars of S_i.

Pick vars compute derivatives; partition remaining vars using hash functions, plug in copy of ROABP hitting set for each S_i. Lower bound: Find non-zero polynomial vanishes over $\mathcal{H}$.

Open: smller hitting sets for depth 3 and 4. Improve depth d case. (Improve depth 4 parameters, or reduction to depth 4. Non-regular depth d?

6.6 PIT for ROABP

Read-once oblivious: any variable occurs in at most 1 layer, and in specified order.

Polytime whitebox test, quasi-poly blackbox test.

First poly-time whitebox test for sum of 2 ROABPs. (key: variable orders can be different)

Evaluation dimension or partial coefficient dimension (Nisan 1991).

Computed by ROABP of width w and var order $x_1, \ldots,$ iff for all $i \in [n]$, $\dim\{A_e : e \in B^i\} \leq w$.

Proof. 1. Write $A = \sum_{j=1}^{w} P_j Q_j$ where $P_j \in \mathbb{F}[x_1, \ldots, x_i]$ and $Q_j \in \mathbb{F}[x_{i+1}, \ldots, x_n]$. (Partial coefficients.)

2. Small coefficient dimension implies small width. Do layer by layer.

□

Given A, B computed by ROABPs in different variable orders.

1. Find linear dependencies among partial coefficients of A.
2. Verify same dependencies hold among corresponding partial coefficients of B.

Reduces to zero testing for ROAMP of a larger width ($\leq w(w+1)$). Generalize to c ROABPs with complexity poly$(w^{2^c} n^c)$.

Make blackbox with least basis isolation and low-support rank concentration.

QUestions:

1. Bring down time complexity to $w^{O(c)}$?
2. Depth 3 multilinear?
3. Inspired from nonequivalence of 2 ROBP's Are there more connections?

6.7 Matrix rigidity

Rigidity: V is (r, d)-rigid if for any matrix V' where $d(v_i, v_i') \leq d$ we have rank rank$(V') \geq r$.

If V is $(\Omega(n), n^{\varepsilon})$-rigid, the linear transformation that maps a vector x to Vx does not have an $O(n)$-size $O(\ln n)$-depth linear circuit.

Valiant needs $(\Omega(n), n^{\varepsilon})$-rigid. With high probability, a random matrix is $(.99n, \Omega(n))$-rigid. The best explicit matrices are $(.99n, 0)$-rigid. The best are $(r, \Omega(\frac{n}{r} \ln \frac{n}{r}))$, $r \geq (\ln n)^2$.

Take a matrix of full rank; Zaranckiewicz's lemma shows after some changes there will be an untouched matrix. "Untouched minor barrier."

Design matrices are candidates for rigidity over $\mathbb{F}_2$. Design matrix; every row has $n^{1-\varepsilon}$ ones, supports of every two distinct rows intersect by $n^{1-2\varepsilon}$.

Goal: Establish $(n^{2\varepsilon+\delta}, n^{1-2\varepsilon})$-rigidity of matrices $V_{m=\frac{1}{\varepsilon}}$.

Any rigidity proof needs to exhibit a property π that is

1. satisfied by all low rank matrices

2. not satisfied by V_m even after perturbations.

Our property is approximability: after a certain particular embedding into $\mathbb{R}^n$ its rows amit a nontrivial approximation by a low-dimension Euclidean space.

Embedding $\mathbb{F}_2^n \to \mathbb{R}^n$, $x \mapsto \frac{x}{\|x\|_2}$.

$$A_r(V) = \max_{W \subseteq \mathbb{R}^n, \dim W = r} \min_{x \in V} \|\mathbb{P}_W(x)\|^2 .$$

$A_r(V)$ is equivalent to Kolmogorov width of V.

Strategy:

1. Show $A_r(V_m)$ is small.
2. Show that $A_r(V_m)$ is robust under perturbation of rows of V_m.
3. Show that for low-dimensional $\mathbb{F}_2$-linear spaces $L \subseteq \mathbb{F}_2^n$, $A_r(L)$ is large.

Imply that V_m have high rank even after perturbations.

The last one is hardest!

(6.4, 5, 7 in CDSM)

1. $r = o(n) \implies A_r(\mathbb{F}_2^n) \sim \frac{r}{n}$.
2. $r = \omega(n^{1-\varepsilon}) \implies A_r(V_m) \approx \frac{v}{n}$.
3. $r = \omega(n^{1-\varepsilon})$: every row of V'_m differ from corresp of V_m in $O(n^{1-\varepsilon})$: $\approx \frac{r}{n}$.

Conj. 6.8 implies $(n^{2\varepsilon+\delta}, n^{1-2\varepsilon})$-rigidit of matrices V_m.

need better approx for $\mathbb{F}_2$-lin spaces.. weak for hi-dim approx. use more property of $\mathbb{F}_2$-lin than trignular raank. approx for n-subsets of $\mathbb{F}_2$ linear spaces rather than complete spaces?

6.8 Algorithms for strategic agents

What is the revenue-optimal auction for selling multiple items?

Myerson designed a revenue-optimal auction for a single item. Bids are transformed into virtual bids. The item is awarded to the highest virtual bidder.

People tried to extend it to more than 1 item. There are lots of characterizations for special settings. CS has contributed approximation algorithms. The settings are limited; there is no unified approach.

There are formal barriers to finding optimal auctions. Challenges arise for 1 buyer and 2 items.

Treat as algorithmic problem.

Adding the second item, each bidder makes a bid on each item. Transform to virtual bid. Each item is awarded to the highest virtual bids.

Myerson's transformations have a closed form. Ours is randomized and computed by a linear program.

Put into context of algorithmic mechanism design. It's not traditional algorithm design. Input, algorithm, output.

We have to first learn the input from agents' report, and the output feeds the agents' payoff. They can lie. We call it a mechanism instead.

They motivated the following design. How much more difficutlt are optimization problems on strategic input comprared to honest input? We want a mechanism that works on strategic input. We want to put a black box around an algorithm that works on honese input, and get what we want by probing it.

Maybe we can reduce hard problems in algorithic mechanism design to hard problems in algorithm design. This allows a much larger community to work on these problems.

A reduction exists! We can also learn what makes mechanisms so challenging.

We can view seminal results as black-box reductions.

VCG is a black-box reduction: if you have the optimal algorithm for welfare, you can get the optimal mechanism for welfare. Agents report types. Input into an algorithm that works on honest input. Just choose the agent that's outputted.. There exists a payment scheme that makes this truthful. More recent works tries to make it approximation-preserving.

Probe multiple inputs instead.

Put virtual bids into the black box maximizing welfare.

We do care about objectives beyond welfare and revenue. Job scheduling on unrelated machines is paradigmatic. Machine i takes time t_{ij} to process j. Minimize makespan. Machines are strategic agents, you don't know how long it takes them to process the jobs. We know how to truthfully optimize welfare (revenue) if we know how to algorithmically optimize welfare (virtual welfare). There is no reduction from truthfully minimizing makespan to minimizing makespan even in 1 dimensions. A reduction exists if we perturb algorithmic objective.

Theorem 6.4: There is a polynomial time reduction from mechanism design for objective O to algorithm design for same objective O plus virtual welfare.

The reduction is approximation-preserving; there is no restriction on types (dimensions).

"Transitioning from honeset to strategic input is no more computationally difficult than adding virtual welfare to objective."

For now, objective is revenue, and agents are additive. (The value for a set of items is the sum of the values.)

A bidder's type is a vector of values. They are taken from some distribution. (Beliefs are modeled as this distribution.) Bayesian incentive compatibility: if every other agent is telling the truth, I want to tell the truth as well.

A mechanism is a function that takes a input a profile of types and outputs a distribution over allocations and a price for all agents. We can think of a mechnsm as a large-dimensional vector (for every input, what's the output). Thinking of it this way we can try to optimize over all algorithms.

We can write out linear constraints to make M truthful and makes it a feasible mechanism.

The problem is that the input is enormous (too many variables). We need a compact description.

Instead we use the reduced form. We just keep the important information.

1. For every agent and item, what's the probability they get the item?

2. What's the expected price they pay?

Write a LP with much fewer variables. We can still write linear constraints. The new challenge is to write constraints to guarantee it's feasible, correspond to an actual algorithm. This is challenging. Does a reduced form correspond to auction?

There is a different succinct description that we can use, similar to reduced forms.

Given a reduced form does it correspond to actual mechanism? Borders theorem gives conditions for feasibility. Borders requires checking exponentially many inequalities. We give equivalent conditions that can be efficiently checked.

We can solved LP using computationally efficient Border's Theorem. For general settings, use equivalence of separation and optimization.

Every reduced form that is a corner of the space of feasible reduced forms is a virtual welfare maximizer for some virtual transformation. The direction has a component/weight for each i, j, l_i. Optimizising in this direction is like optimizing virtual welfare. Combine with Caratheodory's theorem: every point in P is a convex combinations of corners of P. The optimal mechanism is a distribution over virual welfare maximizers.

Sample corners appearing in convex combination.

1. Write succince LP for optimal truthful mechanism using reduced form

2. Solve LP.

3.

Contribution to pure convex optimization: extend to accommodate approximation errors. We apply it to new auction settings where auctions have budgets.

No matter what the bidders' types, the revenue poptimal auction is a distribution over cirutal welfare maximizers. We discovered the structure through algorithmic study.

How much more difficult are optimization problems on strategic input comapred to honest input? For all objectives in Bayesian setting, reduction exsits from mechanism to algorithm design with perturbed objective.

Look towards future: New framework and tools for multi-dimensional settings. What can we do with them?

6.9 Correlation bounds against monotone NC^1

Theorem 6.5: The k-CYCLE problem on $G(n, p)$ is .51-hard for monotone NC^1.

Think of $k = \ln \ln n$ and $p \approx \frac{1}{n}$.

This is the first correlation bound against NC^1 under any product distribution.

Corollary 6.6: Optimal $\frac{1}{2} + n^{-\frac{1}{2}+o(1)}$ hardness under uniform disribution for explicit N-variable monotone function $\mathsf{Tribes} \otimes ? \otimes \text{AND}$

Lower bound against negation-limited NC^1 circuits with $(\frac{1}{2} - o(1)) \lg n$ negation gates.

Monotone complexity: NC^1 is also monotone formulas (?).

Correlation bound: "$\frac{1}{2} + \varepsilon$ hard for $\mathcal{C}$."

Importance of product distributions (ex. uniform, p-biased, $G(n, p)$).

- influenctial conjectures that k-SAT, CLIQUE, etc. are hard on average.
- boolean analysis of monotone functions (Bollobas-THomason, Friiedgut)
- Correlation inequalities (FKG), monotone coupling

In the monotone setting, we have known separations of monotone classes:

$$mAC^0 \subset mNC^1 \subset mL \subset mNL \subset mAC^1 \subset mNC^2 \cdots mP \subset mNP.$$

Nothing had been done on average-case.

We focus on k-CLIQUE. Consider n-vertex graphs ordered by inclusion.

Razborov, 1986: k-CLIQUE has monotone circuit complexity $n^{\Omega(k)}$. The proof is average-case with respect to a distribution:

- NO: complete $k - 1$-partite graphs (maxterms)
- YES: isolated k-cliques (minterms)

Easy to separate by small non-monotone circuit: use anti-monotone threshold function. This is the "Hamming weight gap": no instances are heavier.

Almost all previous averge-case monotone lower bounds rely on the Hamming weight gap. This is a "barrier" explanation: why techniques in the monotone setting don't work for TC^0.

Average case on $G(n, p)$: there is no Hamming weight gap: correlation $n^{o(1)}$.

Conjecture: k-CLIQUE is hard on $G(n, p)$ for Boolean circuits. THis was proved for AC^0 circuits (Beame 90).

There is a similarity between $G(n, p)$ and the slice distribution $G(n, m)$ (exactly m eges). Berkowitz showed that on eveyr slice distribution, monotone complexity and non-monotone complexity are equal (Berkowitz 82). if k-CLIQUE is hard for monotone circuits on the slice distribution, then $P \neq \mathsf{NP}$. Think of $G(n, p)$ as a blurry slice distribution.

2010: k-CLIQUE is hard for monotone circuits on NO= $G(n, p^+)|k$-clique and YES= $G(n, p^-)|k$-clique. This is still subject to the Hamming weight gap.

Monotone coupling w.r.t. any produce distribution. Exists monotone coupling on YES-NO instances. This comes out of Holley's proof of the FKG inequality. We extend correlation bounds against monotone circuits to against negated circuits (?).

Consider the k-CYCLE problem on k-layered graphs: is there a path going across starting and ending at the same vertex?

Upper bound: find all paths between left and right: compute for two halves and merge the results. Do so with semi-unbounded fan-in monotone circuits of size poly(n) and depth

$O(\ln k)$. So k-CYCLE is in $mSAC^1$. Monotone formulas of size $n^{O(\ln k)}$. In the average case, Γ is such that has cycle with probability .5.

Tight $n^{\Omega(\ln k)}$ lower bound on the size of monotone formulas solving k-CYCLE on Γ with probability .51. (Same lower bound for non-monotone AC^0 formulas proved in 2014.)

Proof sketch: persistent minterms

Consider kn layered vertices (in k layers). Consider $A \subseteq C_k$. Look at inputs which are an "A'-section." Let $\mathcal{M}_A(f)$ be A-section minterms of f.

Consider $\mathcal{M}_A(f)$ over all subformulas f. Consider over all subgraphs $A \subseteq C_k$. Every minterm of $f \vee g$ is a minterm of f or a minterm of g, $\mathcal{M}_A(f \vee g) \subseteq \cup$. Impose constraints on how minterms behave.

$$f^{\cup\Gamma}(X) = f(X \cup \Gamma).$$

If ...

To achieve bottleneck, need density constraints on $\mathcal{M}_A(f^{\cup\Gamma})$.

Pathset ocmplexity lower bound. If sets of minterms are small for all A, f, then formula size is $n^{\Omega(\ln k)}$.

Lemma (useless): $\mathbb{P}_\Gamma[\mathcal{M}_A(f^{\cup\Gamma}$ small$) \geq 1 - O(n^{-\frac{1}{2}})$. We need to take a union bound; $n^{-\frac{1}{2}}$ is too weak.

BAD: if f computes $\text{THRESHOLD}_{\geq n \text{ edges}}$: bad even is $n-1$ edges. Then $\mathcal{M}_{(\text{single edge})}(f^{\cup\Gamma})$ is large, $\Omega(n^2)$. Need to filter out functions creating many minterms with non-negligible probabiity

Replace $\mathcal{M}_A$ with persistent minterms, $\mathcal{P}_A(f^{\cup\Gamma}) \geq 1 - e^{-n^{\Omega(1)}}$.

Consider a sequence of random graphs $\Gamma_0 \subseteq \cdots \Gamma_n$ and monotone functions $f^{\cup\Gamma_i}$. Minterms shrink going up.

Persistent if it's a minterm of a certain number of restricted functions: remains minterm for long duration. $A' \in \mathcal{M}(f^{\cup\Gamma_i})$ for $\binom{d+e-1}{e-1}$ many $i \in [0, m]$. Behaves like usual minterms. (d is depth(f) and e is number of edges. $f \vee g$ is pm of f or pm of g, similarly for $\wedge$.

$\Gamma := \Gamma_0 \cup n^{\frac{1}{4}}$ random length-k paths. $d_{TV}(\Gamma_0, \Gamma_m) = o(1)$. Noise lives within variance. Need for correlation bounds

High-level summary: given a monotone formula which approximates k-CYCLE on Γ, track persistent minterms at all subformulas. There are many persistent k-cycle minterms at output, and few...

Giving bound fo k-CYCLE.

Open: is k-CLIQUE hard for mNC^1 on $G(n, p_{\text{threshold}})$. How to define $\Gamma_0 \subseteq \cdots \subseteq \Gamma_m$ with required properties? This proof isolates nice properties of k-CYCLE.

Trivial: $k - CYCLE \otimes AND_{\lg(\frac{1}{p})}$ is 0.51-hard for mNC^1 under uniform.

Use hardness amplification of O'Donnell to get $\mathsf{Tribes} \otimes k - \mathsf{CYCLE} \otimes \mathsf{AND}$ on n vars, $\frac{1}{2} + n^{-\frac{1}{2}+o(1)}$-hard for NC^1. Best hardness you can possibly achieve (every monotone function has agreemeent $\frac{1}{2} + \Omega(n^{-\frac{1}{2}+o(1)})$ with some function in NC^1.

Can convert monotone bounds to bounds against negation-limited circuits. Upper bound: $\lg n$ negations suffice to compute any function in NC^1. We show hardness against NC^1 with $(\frac{1}{2} - o(1)) \lg n$ negations: halfway to lower bounds.

if a monotone funciotn f is $\frac{1}{2}+\varepsilon$-hard for monotone of given size and depth, f is $\frac{1}{2}+O(2^t)\varepsilon$ hard for circuits with t negations of same size and depth. Use monotone coupling and decomposition of negation-limited circuits.

$\frac{1}{2}+n^{-\frac{1}{2}-\Omega(1)}$ bounds against monotone implies bounds against circuits.

If monotone function f is $\frac{1}{2}+\varepsilon$ hard for monotone fomulas, then f is $\frac{1}{2}+O(t)\varepsilon$ hard for formulas.

7 6-19-15

7.1 Quantum Arthur-Merlin

Definition of IP, AM.

Quantum interactive proof system, introduced by Watrous. The verifier and prover use quantum computers, and communication is quantum.

- $PSPACE \subseteq QIP[3]$.
- $QIP = QIP[3]$. Contrast with classical IPs which conjecturally can't be parallelized into consant-turn IPs.
- $QIP = PSPACE$.
- $QIP[1] = QMA$.

Little is known about $QIP[2]$.

QAM: Arthur sends a classical random string y; Merline sends a quantum state. $QMAM = PSPACE$.

Investigate 2-turn QIP systems more finely. What is a fully quantum Arthur-Merlin proofs system?

Definition 7.1: qq-QAM (in between QAM and $QIP[2]$): Arthur creates polynomially many copies of EPR pair, where the first half of each copy is in S1, the scond half is in S2. He sends S2.

Merlin returns a quantum state ρ. Arthur decides to accept/reject from $x, \rho, S1$ by quantum algorithm.

qq-QAM has a natural complete problem CITM.

Definition 7.2: Instance: quantum circuit C which has some specified input qubits and ℓ specified output qubits.

Yes: there exists a state ρ such that $D(C(\rho), \left(\frac{1}{2}\right)^{\otimes l}) \leq a$.

No: for any state, ...*geb*.

1. do there exist two inputs which have approximately equal images (under 2 circuits)? QIP-complete.

2. do two given pure states have ... QSZK-complete.

Theorem 7.3 (Quantum analogue of Babai's collapse theorem)**:** qq-QAM does not change by adding $O(1)$ turns of classical interactins prior to the communications of the qq-QAM proof system.

$t_m \cdots t_1 - QAM(m)$ into 4 classes.

1. PSPACE= $qcq - QAM = QMAM$
2. cq-QAM
3. qq-QAM
4. qc-QAM

   $AM \subseteq cc - QAM \subseteq cq - QAM = QAM \subseteq qq - QAM \subseteq QIP[2] \subseteq PSPACE.$

1. New upperbound for QAM: $QAM = cq{-}QAM \subseteq qq{-}QAM_1$. (cf. $QAM \subseteq BP{\cdot}PP$).

   QAM$\subseteq QIP[2]_1$.
2. $cc - QAM = cc - QAM_1$.

$c \cdots cqqAM(m) = qq - QAM$. Proof strategy:

1. $m \geq 4$: Use Babai's classical proof.
2. $cqq - QAM \subseteq qq - QAM$: use CITM.

   Let $A = (A_{yes}, A_{no})$ in $cqq - QAM$. On input (x, q), simulate the last 2 turns under the condition that the 1st message in Π was w.

   Show $qq - QAM$ completeness of MaxOutEnt: is the entropy of a given channel large for any input?
3. $ccqq - QAM \subseteq qq - QAM$.

- CITM: is the output of a given circuit close to totally mixed state for any input?
- MaxOutQEA: does a quantum channel have the max output entropy larger than a threshod?

Open:

1. Natural problem in qq-QAM not known to be in cq-QAM. Or equl?
2. Non-trivial loewr and upper bound for qq-QAM? $cq - QAM, QIP[2]$
3. $qq - QAM = qq - QMA_1$.
4. Quantum analogue of Goldwasser-Sipser

7.2 Quantum parallel repetition

Make games much harder against players with quantum power.

2-player 1-round games. Referee sends x, y to 2 players, they send back a, b (without communication); the referee decides whether to accept.

$\text{val}(G)$ is the maximum probability Alice and Bol can win with deterministic strategies, over random questions x, y.

Here we are interested in the quantum value. $\text{val}^*(G)$ is the maximum probability lice and Bob can win with entangled strategies, i.e., making measurements on preshared entangled state. (It doesn't allow them to communicate.)

Clearly $\text{val}(G) \le \text{val}^*(G)$. In the CHSH game, they receive uniform random bits; they win if $a + b = x \wedge y$.

$$\text{val}(CHSH) = \frac{3}{4}$$
$$\text{val}^*(CHSH) = .85.$$

How powerful is quantum entanglement?

Parallel repetition is $G^n = (\mu^n, V^n)$. Referee chooses n independent pairs (x_i, y_i) from μ. Alice and Bob win if they win all n copies. How much harder is this?

How does $\text{val}(G^n)$ behave as a function of $\text{val}(G), n$? Fortnow and Feige gave a counterexample: there exist games G such that $\text{val}(G^2) = \text{val}(G) = \frac{1}{2}$.

Parallel repetition theorem:

$$\text{val}(G^n) \le (1 - \varepsilon^c)^{\Omega\left(\frac{n}{s}\right)}.$$

Dependence on size of answer: $s = |a| + |b|$. The game value decays exponentially in n.

This parallel repetition theorem holds when Alice and Bob are classical.

How does $\text{val}^*(G^n)$ depend on $\text{val}^*(G), n$? Is there exponential decay, or decay at all? For general G, this is not known.

- Feige-Killian repetition of a game G goes to 0 with number of repetitions, but not exponential decay.
- XOR games satisfy perfect parallel repetition.
- For free games, $\text{val}^*(G) \le (1 - \varepsilon^2)^{\Omega\left(\frac{n}{2}\right)}$.
- For projection games, there exists $c \ge 1$, $\text{val}^*(G^n) \le (1 - \varepsilon^c)^{\Omega(n)}$.

We show that for a free game (input distribution is product distribution),

$$\text{val}^*(G^n) \le (1 - \varepsilon^{\frac{3}{2}})^{\Omega\left(\frac{n}{s}\right)} \le (1 - \varepsilon^2)^{\Omega\left(\frac{n}{s}\right)}.$$

We improved the exponent on ε. We don't know of a classical analogue of this theorem: parallel repetition of free games—we don't know we can make it $\frac{3}{2}$. Rate of decay matters for hardness of approximation. We use new tools from quantum communication complexity, so this may not be true for classical value! Barak et al: $\mathrm{val}(G^n) \leq (1-\varepsilon^2)^{\Omega\left(\frac{n}{s}\right)}$. Raz showed there are projection games where ε^2 is optimal. It's plausible that the bound of Barak is tight! Then quantum and classical games behave differently under parallel repetition.

We use a faster quantum communication protocol to obtain a better bound than CS14. The communication protocol is a distributed version of the Grover search algorithm, improving $1-\varepsilon^2$ to $1-\varepsilon^{\frac{3}{2}}$.

High level proof idea: convert a "too good" strategy for the repeated game to a strategy for the original game that does better than $1-\varepsilon$, contradiction.

Intuition: use Bayes's rule:

$$\mathbb{P}\left(\text{win all rounds}\right) = \prod \mathbb{P}\left(\text{win round } i\text{—win rounds} < i\right).$$

If $\gg 1-\varepsilon$, then there are many rounds such that the factor is $> 1-\varepsilon$.

Embed inputs $(x,y) \sim \mu$ into some i of the global strategy conditioned on some event W, $\mathbb{P}(\text{win } i|W) > 1-\varepsilon$. Simulate S conditioned on W, the players won all games in a randomly chosen set C of a certain size. If the subset is large enough ($|C| \geq \Omega\left(\frac{n\gamma}{\varepsilon}\right)$,$\mathrm{val}^*(G^n) = 2^{-\gamma n}$).

When is simulation possible? The quantum mutual informations are small:

$$I(X_i : E_B Y|W=1), I(Y_i : E_A X|W=1) \leq \delta$$

Then A and B can simulate S, condtioned on W with $X_i = x, Y_i = y$, with $O(\delta)$ error. So Alice and Bob can win G whp, contradiction.

Play G^n using strategy S. A sends answers in C rounds to Bob, who computes if W occured. (?) How much did Bob learn about Alice's inputs? $\leq O(Q)$. Using chain rule to get average $\leq O\left(\frac{Q}{n}\right)$. Here Q is $|C|$ times Alice's answer length, $O(\frac{\gamma ns}{\varepsilon})$.

Alice and Bob are searching for a round j in C. Any classical protocol must use at least $\Omega(|C|)$ bits of communication. Use Grover search. Grover search with low error: to search database of C with error probability δ, suffices $\sqrt{|C|\ln\left(\frac{1}{\delta}\right)}$ queries.

Lower communication implies smaller mutual information.

Open:

1. Other uses of efficient communication protocols to prove better direct sum/direct product teorems.

2. Is the classical bound (Barak) for parallel repetition of free games tight?

3. Is $\frac{3}{2}$ tight?

7.3 Quantum query complexity

7.3.1 Bomb query complexity

Modified quamtum query inspired by Elitzur-Vaidman bomb tester.

Given a bomb which is either a dud or a live bomb. Live: explodes on contact with photon. Dud: no interaction with photon. Can we tell them apart without blowing it up?

Put bomb in an interferometer: split the photon. If you see a photon on D_2 you know you have a live bomb. (50% explode, 25% know it's a bomb.) We can improve to ε with a quantum zero effect.

We can rewrite the Elitzur-Vaidman bom in the circuit model: Let $R(\theta)$ be rotation matrix. Add a $R(\theta)$ on top, and repeat $\frac{\pi}{2\theta}$ times.

If live: first register is projected back to $|0\rangle$ on each measurement. Probability of explosion: $\Theta(\theta^2)\Theta(1/\theta) = \Theta(\theta)$.

Bomb query: quanum query where each bit of X is an EV-bomb.

Differences from quantum query:

- extra control register c,
- record register must contain 0 as input.
- must measure query after each input.

$B_\varepsilon(f)$ number of bomb queries needed to have probability of explosion $\leq \varepsilon$.

$$B_\varepsilon = \Theta(Q(f)^2/\varepsilon).$$

Upper bound: quantum zero effect. Simulate each quantum query using $\Theta\left(\frac{1}{\theta}\right)$ bomb queries.

Lower bound: adversary method. General weight adversary bound tightly characterized quantum query complexity.

7.3.2 Algorithms

$O(N)$ bomb query algorithm for OR. Stop at first live bomb. $O(N/\varepsilon)$. We get $Q(OP) = O(\sqrt{N})$. This gives a nonconstructive proof of the existence of Grover's algorithm. Can we generalize this further?

If there is a classical randomized algorithm that computes f using $\leq T$ queries, and an algorithm that predicts the results of each query $\mathcal{A}$ makes, making $\leq$ expected G mistakes, then

$$B_\varepsilon(f) = O(TG/\varepsilon), \qquad Q(f) = O(\sqrt{TG}).$$

For OR, $T = N$, $G = 1$.

Cf. Kothari's algorithm for orace identification.

At each step,

1. use G to predict remaining queries.
2. find the location of the first mistake using $O(\sqrt{d})$ quantum queries
3. This determines the actual query results up to the mistake we just found.

Query complexity $O(\sqrt{TG})$. Log factors can be removed using span programs.

Application: Unweighted single-source shortest paths. Guessing that (v, w) is not an edge, we make $\leq n-1$ mistakes, since each vertex is discovered once.

$$Q(uSSSP) = O(\sqrt{TG}) = O(n^{\frac{3}{2}}).$$

matches lower bound.

Problem: unweighted k-source shortest path. Classical: run BFS k times. Quantum: $G = k(n-1)$ but $T = O(n^2)$ instead of $O(kn^2)$.

2109 on Scott A's blog.

$O(n^{1.75})$ quantum query algorithm for maximum bipartite matching.

Open:

- relate G to classical measures of query complexity (certificate, sensitivity).
- time complexity of algorithms
- algorithm for adjacency list model
- other problems, e.g., matching for general graphs
- $R(f)$ and $B(f)$

Conjecture: $R(f) = O(Q(f)^2)$ (attained by OR).

Projective query complexity, $P(f)$. Allow black-box access to a circuit that compotes O_x but then forces a measurement. Decohere result? $Q(f) \le P(f) \le R(f)$, $P(f) \le B(t)$. $P(OR) = \Omega(N)$. Conjecture: $P(f) = \Theta(R(f))$ for all total functions.

7.4 The social cost of cheap pseudonyms

Eric Friedman, Paul Resnick, The social costs of cheap pseudonyms, 2001. (Started 1998.)

When you can easily have pseudonyms, it's hard to control behavior. Much of the beauty comes from the conciseness of the model, how little they assume. When you have a small probability of malicious action, or innocent mistakes, cooperation is no longer equilibrium. Solution: Treat newcomers with skepticism.

We had to explain people what ebay was, why reputation on the Internet was important, etc.

Literature postview: the idea of reputation is now mainstream; since 1999, 10% of EC papers study reputations.

The cost of changing identity is key. This was a new aspect of the Internet. The technical approach is **dynamical games** (underutilized in the EC community).

An economist, a computer scientist, and a philosopher walk into a bar...

- Philosopher: what is the meaning of self on the Internet?
- Computer scientist: how can we secure identities using cryptography?
- Economist: How do we make money out of this?

How can we create trust; how much loss do various models of trust imply? Why is trust important?

- selling and trading websites like Ebay.

  Reputation informs buyers, incentivizes sellers to honesty and quality service ("shadow of the future"). Note in these systems you don't usually interact with the same person again; how to make this work?

  Self-selection: more honest people will be drawn to be sellers.

- online forums and discussion groups.

- cooperative groups like wikipedia.

- networks of autonomous devices, like BitTorrent.

How/when should reputation systems work and how can we make them better?
Consider a reputation system with different kinds of interactions.

1. anonymous (ID changes every action)

2. pseudonyms (ID changes at will): you can start a new name if you want. If you rip somebody off, you can come in as a new person.

3. identified (real name, ID never changes)

We introduced 1L pseudonyms: you can change once in a lifetime.

In every period, people get matched up and play a prisoner's dilemma. The problem is the folk theorem of economics: in repeated games anything can happen. There are many strange equilibria.

If everyone starts at the same time, and one person comes back with a new name, that person can be identified. Instead we assume a bunch of people arrive/leave at specified times.

What's public is the complete PD matrix and history of plays.

We considered only sequential equilibria, the expected per-period payoff. Intuitively, $V = 1 - \frac{\text{(number of defections)}}{2}$.

What can you do? Local punishment strategy: You have to do something bad to a newcomer. You defect against a new arrival, and cooperate against good people.

But you have to cooperate against new people, otherwise people will defect and come back as new people, and the system will break down.

Mathematically, use the public grim trigger strategy: if anyone non-compliant, shut down the whole strategy and everyone goes home. This isn't realistic.

We added the ideas of "trembling": even if you choose to cooperate, with ε probability you actually defect (something gets lost in the mail for ebay...).

How much loss comes from free pseudonyms? We introduced Pay Your Dues: if you're around for a while and have a good reputation, you can defect against newcomers. (This is realistic.)

You can show this is an equilibrium. There is inefficiency when you have to pay dues.

What are solutions? Pay for IDs. This discourages participation. Often you have to do some annoying tasks to do, ex. fill out a long form.

How would we implement this cryptographically? How could you implement pseudonyms so that people can get a pseudonym and never use it again? With this, you can do well. This never happened though. How could CA's cooperate, etc.?

Our conclusions:

- Cheap pseudonyms imply distrust for newcomers.

  We had to prove things in dynamic systems, modify the model with noise. Without noise we get bizarre effects.

- Expensive pseudonyms help.
- Expensive to change psuedonyms are even better.

Some underserved research areas by EC:

- application of dynamic games: most work is static or 2-period.
- noise ("trembles")
- self-selection (market for lemons)

Now reputation is still held centrally.

7.5 Extractors in AC^0

We systematically ask whether the usual extractors are in AC^0.

We are still far from understanding the power of AC^0.

7.5.1 Seeded extractors

What is known about seeded AC^0 extractors? An extractor is Ext : $B^n \times B^r \to B^m$, on a min-entropy k source.

- Negative: I showed that $m \leq 1.01r$ unless $\frac{k}{n} \geq \frac{1}{\operatorname{poly}\log n}$.
- Positive: $m = r+1, r = n$ by $(x, y, \langle x, y \rangle)$.

Our results:

1. We can extract 1 bit ($m = r+1$) iff
$$r \geq \frac{n}{k \ln^{O(1)}(n)} + 10 \ln n.$$
2. If $\frac{k}{n} \leq \frac{1}{\lg^{\omega(1)} n}$ then we can extract iff
$$\frac{m}{r} \leq 1 + \frac{\lg^{O(1)}(n) k}{n},$$
   1+the entropy rate. Strong extraction (where we have to output the seed) is impossible.

3. If $\frac{k}{n} \geq \frac{1}{\lg^{O(1)} n}$, then we can extract strogly with $m = 1.01r$, $r = O(\lg n)$. It is open whether we can attain $m = \Omega(k)$: can we extract more bits?

To extract 1 bit from entropy k¡ sample $\frac{n}{k}$ bits. If $\leq \lg^{O(1)} n$ apply best-known extractor. If $\geq \log^{\omega(1)} n$, apply inner produce extractor with seed $\frac{n}{k}$. To extract t bits repeat with t independent seeds.

7.5.2 Deterministic extractors

Consider extractors for bit-fixing (restriction) sources. The entropy is the number of unfixed variables.

The switching lemma says that any depth-d cirucit becomes constant on random restriction leaving $\frac{n}{\lg^{d-1} n}$ variables.

Some depth-d circuits are far from constant (good extraction) on any restriction leaving $\frac{n}{\ln^{\Omega(d)} d}$ variables.

Picking restriction after circuit is no better than random.

Ajtai and Linial showed there exist depth-4 circuit $B^n \to B$ that extracts if $k = n - \frac{n}{\operatorname{poly}\log(n)}$ bits uniform, other $n-k$ function of those k. The idea is to combine AL with a sparse linear map $B^{n \operatorname{poly}\log(n)} \to B^n$: any $n \times n$ submatrix has rank $\geq n - \frac{n}{\operatorname{poly}\log(n)}$. We couldn't prove the existence of such a linear map; instead we get a map over a large field and combine with codes, non-linear condenser.

Extractors for independent sources: Best $\frac{k}{n}$ for P-time:

2 sources: Bourgain showed .499. For AC^0, $1 - \frac{1}{\operatorname{poly}\log(n)}$.

3: $n^{-.49}$ (actually now AC^0 : 0.99), Li.

4: Same. AC^0 : 0.99.

$O(1)$: $\frac{\operatorname{poly}\log n}{n}$. AC^0 : 0.01.

AC^0 and 2-party protocols. The better you can extract the better for 2-party protocols.

Open: better AC^0 lower bound? Only known : $\frac{k}{n} \geq \frac{1}{\operatorname{poly}\log(n)}$.

For min-entropy sources we get a complete picture for $m = r+1$, $\frac{k}{n} \leq \frac{1}{\lg^{\omega(1)} n}$.