WHAT'S ON TODAY'S AGENDA

"Every NP statement has an exponentially long proof that can be locally tested by looking at just a constant number of bits"

This is off by an exponential in the proof size – we want a polynomial-size proof that can be checked with a logarithmic amount of randomness – but is still nontrivial

THM 11.19 $\boxed{NP \subseteq PCP_{1,1/2}(poly(n), 1)}$

To prove this theorem, need to find redundant encoding of proofs/assignments that can be checked with very few queries

Suppose we have string $u \in \{0,1\}^n = GF(2)^n$
We can view u as a (representation of a) linear function

$u : \{0,1\}^n \to \{0,1\}$

Encode u by writing down its function table

$$\{u(x) \mid x \in \{0,1\}^n\}$$

String of length 2^n

WALSH-HADAMARD CODE (or just Hadamard code)

Identify $u = (u_1, \ldots, u_n) \in \{0,1\}^n$ with function

$$x \longmapsto u \cdot x = \sum_{i=1}^{n} u_i \cdot x_i \pmod 2$$

$WH : \{0,1\}^n \longrightarrow \{0,1\}^{2^n}$ defined by

$$\boxed{WH(u)} = \{\, u \cdot x \mid x \in \{0,1\}^n \,\}$$

How many different strings in $\{0,1\}^k$? 2^k.

Hence there are 2^{2^n} strings $f \in \{0,1\}^{2^n}$.

Every such f can be viewed as function table $f : \{0,1\}^n \to \{0,1\}$

Out of these 2^{2^n} functions / strings, only 2^n (a logarithmic number) are <u>linear functions</u>

So strings corresponding to $WH(u)$ for some $u \in \{0,1\}^n$ are very rare.

Refer to $WH(u)$ as CODEWORDS of the Walsh-Hadamard code.

An absolutely crucial fact is that any two codewords $WH(u)$, $WH(v)$, $u \neq v$, are <u>very far</u> from each other.

For $f, g \in \{0,1\}^k$ let

$$\delta(f,g) = \Pr_{i \in [k]}[f_i \neq g_i] = \frac{|\{i \in [k] \mid f_i \neq g_i\}|}{2^k}$$

LEMMA For any $u, v \in \{0,1\}^n$, $u \neq v$

$$\delta(WH(u), WH(v)) = 1/2$$

That is, any two codewords differ in exactly half of the coordinates.

Proof? Posed as problem (mildly disguised) on problem set 4).

Can also be phrased as follows:

RANDOM SUBSUM PRINCIPLE

If for $u, v \in \{0,1\}^n$ it holds that $u \neq v$, then for exactly half of all strings $x \in \{0,1\}^n$ it holds that

$$u \cdot x \neq v \cdot x$$

Aside about notation

- Arora-Barak write $u \odot v$
- I will try to consistently write $u \cdot v$
- Also fairly common to write $\langle u, v \rangle$

  Slightly misleading since this is not an inner product (why?) but has many properties of an inner product.

Suppose we are given function table of $f : \{0,1\}^n \to \{0,1\}^n$ (i.e., string $f \in \{0,1\}^{2^n}$)

Want to inspect f in constant # positions and decide whether f is a Walsh-Hadamard codeword, i.e., whether f is linear.

But we already know this can be done using the BLR linearity test:

Pick $x, y \in \{0,1\}^n$ uniformly and independently at random. Accept if

$$f(x+y) = f(x) + f(y)$$

and reject otherwise.

THM 11.21

If $f : \{0,1\}^n \to \{0,1\}$ is such that

$$\Pr_{x,y \in_R \{0,1\}^n} \left[f(x+y) = f(x) + f(y) \right] \geq 1 - \delta$$

for some $\delta < 1/2$, then there exists some linear function $\ell : \{0,1\}^n \to \{0,1\}$ such that $\delta(f, \ell) \leq \delta$

That is, f is δ-close to some linear function.

Aside about terminology

PLEASE NOTE that our "δ-close" is what Arora-Barak refer to as "$(1-\delta)$-close" ☹

Suppose that we want not only to test f for linearity but evaluate the linear function that f encodes. What if function table of f is slightly distorted so that some δ-fraction of values have been corrupted?

Can we still evaluate f with constant number of queries to find value $f(x)$, even if position x is corrupted?

Formally, suppose for $\delta < 1/4$ that f is δ-close to some linear function $\tilde{f}$. Since linear functions have distance exactly $1/2$ between each other, $\tilde{f}$ is unique. Can we evaluate $\tilde{f}$ with constant # queries to f?

Use linearity!

1. Choose $x' \in_R \{0,1\}^n$
2. Set $x'' = x + x'$ [Note that plus = minus over GF(2)]
3 Read $y' = f(x')$
$y'' = f(x'')$
4. Output answer $\tilde{f}(x) = y' + y''$

x' is uniformly distributed, so
$Pr[f(x') \neq \tilde{f}(x')] \leq \delta$ by assumption

x'' also uniformly distributed, so

$$Pr[f(x'') \neq \tilde{f}(x'')] \leq \delta$$

x' and x'' are NOT independent, but for any events A and B the UNION BOUND says that

$$\boxed{Pr[A \cup B] \leq Pr[A] + Pr[B]}$$

So except with probability $1 - 2\delta$ we have $y' = \tilde{f}(x')$ and $y'' = \tilde{f}(x'')$
$= \tilde{f}(x + x')$

in which case

$$y' + y'' = \tilde{f}(x') + \tilde{f}(x + x')$$
$$= \tilde{f}(x)$$

by linearity.

This is called LOCAL DECODING of the Walsh-Hadamard code, or SELF-CORRECTION

Now we want to prove Thm 11.19

$$NP \subseteq PCP_{1,1/2}(\mathrm{poly}(n), 1)$$

Sufficient to prove for one NP-complete language. We will use:

$\boxed{\text{QUADEQ}}$

Given quadratic equations $\{E_1, \ldots, E_m\}$ over n variables, where E_ℓ is on the form

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_{\ell,i,j}\, u_i u_j = b_\ell \qquad (*)$$

$(a_{\ell,i,j}, b_\ell \in \{0,1\})$, is there a $\{0,1\}$-assignment to the u_i:s satisfying all equations?

PROPOSITION

QUADEQ is NP-complete (even if every equation E_ℓ has just a constant number of nonzero coefficients)

Proof Exercise.

(Since $u_i = (u_i)^2$ in $GF(2)$, we can assume that the format is as in (*) above without any loss of generality)

Hence, any QUADEQ instance can be described by

$m \times n^2$-dim matrix A (over $GF(2)$)
m-dim vector b (over $GF(2)$)

Given assignment to $u = (u_1, \dots, u_n)$, assignment to degree-2 monomials given by

$$(u_1 u_1, u_1 u_2, \dots, u_1 u_n, u_2 u_1, \dots, u_n u_n) =$$
$$= (u_1 \cdot u, u_2 \cdot u, \dots, u_n \cdot u) =$$
$$= u \otimes u \quad \text{TENSOR PRODUCT}$$

For such $U = u \otimes u$, instance is satisfied if $A \cdot U = b$

Hence, QUADEQ can be rephrased as the following problem:

Given A, b as above, find n^2-dimensional vector U such that
(1) $A \cdot U = b$
(2) EXISTS u such that $U = u \otimes u$

PCP, ATTEMPT 1

Proof: Assignment $u \in \{0,1\}^n$

Test: Pick equation $\mathcal{E}_\ell$ at random, read required u_i, u_j (for $a_{\ell,i,j} \neq 0$) and check if $\mathcal{E}_\ell$ satisfied

Completeness: OK, $= 1$
Soundness error: $\to 1$
Randomness: Logarithmic
Queries: $O(1)$ [assuming k-QuadEq]

PCP, ATTEMPT 2

Proof: $u \in \{0,1\}^n$

Test: For every equation $\mathcal{E}_\ell$ flip a coin. Sum up all equations for which coin $= 1$

Check
$$\sum_{\ell \in S} \sum_{i,j} a_{\ell,i,j} u_i u_j = \sum_{\ell \in S} b_\ell \qquad (+)$$

Completeness: OK, $= 1$
Soundness: OK, $= 1/2$

Given assignment u, at least one equation $\mathcal{E}_{\ell^*}$ violated

Sum (†) for all other equations in chosen set S. Then flip coin for ℓ^*. Two cases:

1. Sum (†) without ℓ^* is satisfied. Then with prob $= 1/2$ E_{ℓ^*} is included, which violates (†)

2. Sum (†) without ℓ^* is already violated. Then with prob $= 1/2$ we don't pick E_{ℓ^*} and (†) is also violated

Randomness: Polynomial

Query cplx: Expected linear!

PCP, ATTEMPT 3

Proof Ask for $U = u \otimes u$ encoded as $WH(U)$

Test Pick subset S of equations and check (†) as before

Note that we know $\sum_{\ell \in S} b_\ell$ without querying anything. And

$$\sum_{\ell \in S} \sum_{i,j} a_{\ell,i,j} U_{ij}$$

is a linear function! Hence we

read it from $WH(U)$ in an error-correcting way, after first having checked that $\pi = WH(U)$ is indeed an encoding of a linear function. # queries constant.

<u>Completeness</u> OK, $= 1$

<u>Soundness</u> seems OK... Except, how do we know $U = u \otimes u$? We totally don't... If A is invertible, soundness error is 1 ☹

<u>Randomness</u>: polynomial

<u>Query cplx</u>: constant.

PCP ATTEMPT 4 (FINAL)

<u>Proof</u> $\pi \in \{0,1\}^{2^n + 2^{n^2}}$ encoding $WH(u)$ and $WH(U)$ (assuming that $U = u \otimes u$)

Let us denote $f = WH(u)$
$g = WH(U)$

<u>Test</u> ① Test f and g repeatedly (but constant # times) until we are sure except with some probability ε that f is 0.001-close to linear $\tilde{f}$
g is 0.001-close to linear $\tilde{g}$

For the remaining steps, assume f & g are close to these functions $\tilde{f}, \tilde{g}$ (otherwise we will fail, but this is only with probability ε)

(2) $\tilde{f} = WH(u)$ for some u
$\tilde{g} = WH(U)$ for some U

can read from f and g in error-correcting way

Check 10 times:
Pick $r, r' \in_R \{0,1\}^n$
reject unless $\tilde{f}(r)\tilde{f}(r') = \tilde{g}(r \otimes r')$

(3) Pick random subset S of equations as in $(+)$ and check that $(+)$ is satisfied by reading the corresponding position in $\tilde{g} = WH(U)$ from g in an error-correcting way

Accept if the read bit is $= \sum_{l \in S} b_l$.

See pages 251-253 in Arora-Barak for the detailed analysis. Below follow the highlights.

Randomness poly(n) — OK

Query cplx constant — OK

Let us do completeness and soundness together. For the soundness, there is a satisfying assignment u and the proof π is the concatenation of

$$f = WH(u) \quad \text{and} \quad g = WH(u \otimes u)$$

Step 1 In the completeness case, the linearity tests accept with prob 1

In the soundness case, we reject with probability ≥ 0.001 if functions are not that close to linear. Pick some constant K so that

$$1 - (1 - 0.001)^K > 1 - \varepsilon/2$$

Then repeating both tests K times we will reject f, g that are not 0.001-close with probability $1 - \varepsilon$

For the rest of the analysis, assume

$$\delta(f, \tilde{f}) \leq 0.001$$

$$\delta(g, \tilde{g}) \leq 0.001$$

for unique linear functions $\tilde{f}, \tilde{g}$

Step 2 In a correct proof $\tilde{f} = f$, $g = \tilde{g}$ and we have

$$\tilde{f}(r)\tilde{f}(r') = \Big(\sum_{i\in[n]} u_i r_i\Big)\Big(\sum_{j\in[n]} u_j r_j'\Big)$$

$$= \sum_{i,j} u_i u_j r_i r_j'$$

$$= (u \otimes u) \cdot (r \otimes r')$$

$$= \tilde{g}(r \otimes r')$$

which always accepts.

Suppose $\tilde{g} = WH(w)$ for $w \neq u \otimes u^{\neq}$

Write w as n^2-matrix $W = \{w_{ij}\}$

-"- $u \otimes u$ -"- $U = \{u_i \cdot u_j\}$

Then

$$\tilde{g}(r \otimes r') = w \cdot (r \otimes r')$$

$$= \sum_{i,j\in[n]} w_{ij} r_i r_j'$$

$$= r W r'$$

$$\tilde{f}(r)\tilde{f}(r') = (u \cdot r)(u \cdot r')$$

$$= \Big(\sum_i u_i r_i\Big)\Big(\sum_j u_i r_j'\Big)$$

$$= \sum_{i,j} u_{ij} r_i r'_j$$

$$= r\, U\, r'$$

We reject if $rWr' \neq rUr'$

By random subsum principle, at least half of all r satisfy

$$rW \neq rU \qquad (\ddagger)$$

(by same argument as for check of $(\dagger)$ above)

Fix r s.t. $(\ddagger)$ holds

For such r, at least half of r' satisfy

$$(rW)r' \neq (rU)r'$$

So with probability $\geq 1/4$ test rejects

Repeating 10 times, we get rejection probability $\geq 1-(3/4)^{10} > 0.9$

(assuming all error-correcting reads are OK, but if this fails we can add this small probability to previous ε).

Step 3 If we haven't already rejected we can assume that there is some $u \in \{0,1\}^n$ such that $\pi \approx WH(u), WH(u \otimes u)$

where we denote $\tilde{f} = WH(u)$
$\tilde{g} = WH(U)$ for $u \otimes u = U$

If the QUADEQ instance is unsatisfiable, u falsifies at least one constraint.

Picking random S and checking $(\dagger)$, we have 50% probability of detecting this

Repeat whole test 2-3 times as needed to get down below 1/2 soundness error

Formally, we would have to be a bit more careful with our probabilities and do calculations along the lines below.

SCENARIO: No-instance E of QUADEQ
Proof $\pi = f, g$

Case 1

f and g are not close to linear

$\Pr$[failure to detect nonlinearity in step 1] $\leq \varepsilon_1$

Case 2 f, g close to linear but

$f(x) f(y) = g(x \otimes y)$ does not hold

$\Pr[\text{failure to detect this}] \leq$

$$\sum_{\substack{\text{error-correcting}\\ \text{reads}}} \Pr[\text{read fails}] +$$

$$\sum_{i} \Pr[\text{test fails given correct reads}]$$

$$\leq \delta_1^2 + \delta_2^2 \leq \varepsilon_2$$

Case 3 f, g close to encodings of

$WH(u)$ and $WH(u \otimes u)$

$\Pr[\text{failure to detect a falsifying assignment}] \leq$

$$\leq \sum_{\text{reads}} \Pr[\text{read fails}] +$$

$$\Pr[\text{Test}(t) \text{ fails}] \leq \delta_2^3 + \delta_2^3 \leq \varepsilon_3$$

Total failure probability $\leq$

$$\leq \max\{\varepsilon_1, \varepsilon_2, \varepsilon_3\} \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3$$

(And usually epsilons can be made small enough anyway)