Towards the Sliding Scale Conjecture
(Old & New PCP constructions)

Prahladh Harsha
TIFR

[Based on joint works with Irit Dinur & Guy Kindler]
Outline

- Intro, Background & Context
  - Goals and questions in this area

- Old PCP constructions
  - Low degree test
  - Composition

- Something new
Proof Verification: NP to PCP

$\forall \; \text{Theorem}$

$V$ (deterministic verifier)

PCP Theorem [AS, ALMSS]

$V$ (probabilistic verifier)

NP Proof

Completeness: $x \in L \Rightarrow \Pr[V^\pi(x) = 1] = 1$

Soundness: $x \notin L \Rightarrow \Pr[V^\pi(x) = 1] \leq \epsilon$
Parameters of Interest

- Randomness
  - $O(\log n), n$ - length of theorem (size of instance)

- Number of queries
  - As low as possible, $O(1)$ (even 2) if possible

- Soundness error
  - $\epsilon < 1$, as low as possible, ideally $\epsilon \to 0$

- Alphabet Size
  - Not too large $O\left(\exp\left(\frac{1}{\epsilon}\right)\right)$, ideally $\text{poly}\left(\frac{1}{\epsilon}\right)$
Motivating Question

- What is the best polynomial sized PCP (i.e., logarithmic randomness) with \(\text{poly}\left(\frac{1}{n}\right)\) error?
Via Sequential Repetition

The PCP theorem [AS, ALMSS]:
Number of bits read from proof: $t = O(1)$
Soundness error $\epsilon = O(1)$

The PCP theorem + k-repetition (sequential):
Number of bits read from proof: $kt = O(k)$
Soundness error $\epsilon^k = 2^{-O(k)}$

In particular, if $k = \log n$, we get $\epsilon = 1/poly(n)$
This can be achieved in a “randomness efficient” way, keeping
The construction polynomial size.
But, the number of queries increases...
Via Parallel Repetition

2-query PCP theorem [AS, ALMSS]:
Length of alphabet in proof: $t = O(1)$
Soundness error $\epsilon = O(1)$

2-query PCP theorem + $k$-repetition (parallel):
Length of alphabet size: $kt = O(k)$
Soundness error $\epsilon^{O(k)} = 2^{-O(k)}$

In particular, if $k = \log n$, we get $\epsilon = 1/\text{poly}(n)$
Number of queries remains 2

But the randomness increases to $O(k \log n)$ ...
Seeking the smallest $\varepsilon$

- Claim: To get soundness error $\varepsilon = 2^{-t}$ the verifier must read at least $\geq t$ proof bits

- Proof: When reading $t$ bits, there are $2^t$ possibilities, once of which is satisfying. (so a random proof will fool $\geq \varepsilon = 2^{-t}$ fraction of checks in expectation)

- Results in previous slides exhibit best tradeoff wrt. Soundness error vs. number of bits read

- However, these results perform poorly either wrt. number of queries (sequential) or randomness (parallel)
Sliding Scale Conjecture [BGLR 93]

- For all $\epsilon > poly \left(\frac{1}{n}\right)$, there exists PCPs for NP with
  
  - $O(\log n)$ randomness
  - $O(1)$ (even 2) queries
  - $\epsilon$ - soundness error
  - $poly(1/\epsilon)$ sized alphabet

- In particular, $poly(1/n)$ – soundness error with $poly(n)$ sized alphabet.
Why do we care?

- Implies polynomial factor inapproximability of
  - DIRECTED-SPARSEST-CUT [CK]
  - DIRECT-MULTICUT [CK]

- 2-query SSC implies
  - NP hardness of several optimal inapproximability results which are known currently under assumptions
    - NP $\not\subset DTIME (n^{\log \log n})$
    - NP $\not\subset DTIME (n^{\log n})$
## Know Results

<table>
<thead>
<tr>
<th></th>
<th># queries</th>
<th>Soundness error</th>
<th>Alphabet Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sliding Scale Conjecture</td>
<td>$O(1)$ (even 2)</td>
<td>$\epsilon$</td>
<td>$\text{poly} \left( \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>PCP Theorem 92</td>
<td>2</td>
<td>0.999..</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Arora-Sudan 97, Raz-Safra 97</td>
<td>2</td>
<td>$\exp(- (\log n)^{0.1})$</td>
<td>$\text{poly} \left( \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>DFKRS 99</td>
<td>$O\left(\frac{1}{\delta}\right)$</td>
<td>$\exp(- (\log n)^{1-\delta})$</td>
<td>$\exp((\log n)^{1-\delta})$</td>
</tr>
<tr>
<td>DFKRS + seq. rep.</td>
<td>$O\left(\frac{(\log n)^{\delta}}{\delta}\right)$</td>
<td>$\text{poly} \left( \frac{1}{n} \right)$</td>
<td>$\exp((\log n)^{1-\delta})$</td>
</tr>
<tr>
<td>MR 08, DH 09</td>
<td>2</td>
<td>$\epsilon$</td>
<td>$\exp\left(\frac{1}{\epsilon}\right)$</td>
</tr>
<tr>
<td>DHK 15</td>
<td>$\text{poly log log } n$</td>
<td>$\text{poly} \left( \frac{1}{1} \right)$</td>
<td>$n^{1/(\text{poly log log } n)}$</td>
</tr>
</tbody>
</table>
PCP CONSTRUCTIONS
PCP Construction

- All known PCP constructions are based on
  - Low-degree test
    - PCP Theorem [AS, ALMSS], Arora-Sudan 97, Raz-Safra-97, DFKRS 99, BGHSV 04, Moshkovitz-Raz 08, DH 09, DHK 15
  - Direct Product Methods
    - Parallel Repetition [Raz 97], Gap amplification [Dinur 05], Dinur-Meir 11
    - Inapplicable to very polynomial sized very low-error
Low Degree Test

PROBLEM:

Given truth table \( f : \mathbb{F}^m \to \mathbb{F} \),

Check if \( f \) is the evaluation of a degree \( d \) polynomial

Can be checked locally using ADDITIONAL PROOF:

Lines table \( A: \{\text{lines}\} \to \{\text{univariate degree } d \text{ polynomial}\} \)
Low Degree Test (LDT)

Low-Degree-Test (LDT):

1. Pick a random point $x \in \mathbb{F}^m$
2. Pick a random line $l \ni x$
3. Accept if $f(x) = A(l)(x)$.

Completeness: If $f$ is a degree $d$ polynomial, then there exists a lines table $A$ such that

$$\Pr[\text{LDT accepts}] = 1$$
Low Degree Test (LDT)

Low-Degree-Test (LDT):

1. Pick a random point $x \in \mathbb{F}^m$
2. Pick a random line $l \ni x$
3. Accept if $f(x) = A(l)(x)$.

Soundness [Rubinfeld-Sudan’92, ALMSS’92]:

$\Pr[\text{LDT accepts}] > 1 - \delta \Rightarrow f$ is $O(\delta)$-close to a degree d polynomial
Low Degree Test (LDT)

Low-Degree-Test (LDT):

1. Pick a random point \( x \in \mathbb{F}^m \)
2. Pick a random line \( l \ni x \)
3. Accept if \( f(x) = A(l)(x) \).

List-Decoding-Soundness [Arora-Sudan’97, Raz-Safra’97]:

For every \( f: \mathbb{F}^m \to \mathbb{F} \), there exist \( L = O \left( \frac{1}{\delta} \right) \text{poly} \ P_1, P_2, \ldots, P_L \)

\[
\Pr[ \text{LDT accepts and } A(l) \notin \{P_1, P_2, \ldots, P_L\}] < \delta
\]
LDT $\rightarrow$ PCP

- Encode NP witness as a low-degree polynomial
- Setting: $|\mathbb{F}| = n^{1/5}, m = O(1)$ such that $|\mathbb{F}^m| = poly(n)$
- Proof: Evaluation $f$ and lines table
- Consistency can be checked using sum-check protocol (ignore for this talk)

- Parameter Gain:
  Read only $poly(|\mathbb{F}|)$ bits instead of $|\mathbb{F}^m|$ bits
LDT PCP - summary

- Gives us “local to global” connection

- Parameter gain: instead of reading $n \approx |F|^m$ bits, the verifier only reads $|F|\log|F| < \sqrt{n}$ bits.

- The only (?) known way to construct PCPs with small error

- Cannot go “all the way”, i.e. the local views are not local enough $\Rightarrow$ need composition
Why Composition

- LDT based PCPs have large alphabet (i.e., read too many points)

- Alphabet Reduction (aka composition) is done to reduce alphabet size
Reducing \# queries

Verifier’s Actions

1. Read inputs $\Phi$, $r$

1. Compute local window $I$ and local predicate $f$

Idea: Compose!! [ala composition of AS’92]

Use “Inner” PCP Verifier to check if local window satisfies local predicate

Consistency Issue: Inner verifier not only needs to check local predicate is satisfiable (easy), but also that is satisfiable by local window

Resolve Consistency using PCPs that can decode!!
How to resolve consistency issue

- \([\text{AS,ALMSS}]\) – hardcoded into construction
  - Organic to basic building blocks
  - Specialized to specific PCPs (RM, Hadamard based PCPs)

- \([\text{Sze,DR,BGHSV}]\) – “definitional” solution
  (Assignment testers, PCPs of Proximity)
  - Modular
  - Allows more than constant number of composition steps
  - does not work for small soundness error

- Decodable PCPs (dPCPs): “definitional” solution
Decodable PCPs

NP Proof $y$

PCP $\pi$

$\Phi, r$ → ACC/REJ
Decodable PCPs

Decodable PCP (dPCP) – encoding of NP proof

- locally checkable
- locally decodable
Decodable PCPs

Soundness:
For every dPCP $\pi$, there is at most a NP proof $y$
\[ \Pr[\text{Verifier’s output inconsistent with } y] < \delta \]
Decodable PCPs

Soundness: For every dPCP $\pi$, there is a short list of NP proofs $y_1, \ldots, y_L$, such that $\Pr_{(I)}[f(\pi I) \not\in \{y_j\} \cup \{\text{reject}\}] < \delta$.

Inspired by list-decoding soundness of LDT.
Composition w/ decodable PCPs

- Implicit in earlier constructions
- dPCPs make it possible to express existing composition techniques in a generic setting
- Composition Theorem (informal): Outer PCP with soundness error $\Delta$ composed with inner decodable PCP with soundness error $\delta$ and list size $L$ yields composed PCP with soundness error $\delta + L\Delta$
- This framework yields all previous PCP constructions (AS, ALMSS, DFKRS, MR, DH)
Even better composition?

- $t$ rounds of composition results in soundness error of at least $L^t \delta$

- Exponential dependence on $t$ - prohibitively expensive for super-constant rounds of composition

- Question: Can one do better than list-decoding soundness for decodable PCPs and avoid list size $L$?
List-decoding soundness

Low-Degree-Test (LDT):
1. Pick a random point $x \in \mathbb{F}_m$
2. Pick a random line $l \ni x$
3. Accept if $f(x) = A(l)(x)$.

List-Decoding-Soundness [Arora-Sudan’97, Raz-Safra’97]:

For every $f: \mathbb{F}_m \to \mathbb{F}$, there exist $L = O\left(\frac{1}{\delta}\right) \text{poly } P_1, P_2, ..., P_L$

$$\Pr[\text{LDT accepts and } A(l) \notin \{P_1, P_2, ..., P_L\}] < \delta$$
Overcoming list-size bottleneck

- The LDT’s acceptance is explained by $L$ polynomials.

- However, each local view can be consistent with only one element of the list $L$ (distance property of local view).

- Can we use this to remove list-size dependence?
Distributional Soundness:

Allows for additive error in composition $\Delta + \delta + \eta$
Using improved composition

- We will apply PCP composition repeatedly, (as in DFKRS)
- Alphabet size: \( n \rightarrow \sqrt{n} \rightarrow \sqrt[4]{n} \rightarrow \cdots \rightarrow O(1) \)
  (after \( \log \log n \) steps, so we will make \( \log \log n \) queries)
- **Improved composition theorem**: error builds up additively, so \( \epsilon \rightarrow \log \log n \cdot \epsilon \)
- **Theorem**: NP has a PCP verifier with
  - \( \text{poly log log } n \) queries
  - Alphabet size \( n^{\log \log n} = 2^{\log \log n} \)
  - Soundness error \( 1/ \text{poly}(n) \)
THANK YOU