Matrix Rigidity

Rectangular PCPs

Sasha Golovnev

October 21, 2020

Overview

- Recall that we want $M \in \mathbb{F}_2^{n\times n}$, $M \in \mathsf{E}^{\mathsf{NP}}$
$$\mathcal{R}_M^{\mathbb{F}_2}(\varepsilon n) \geq \Omega(n^{1+\delta})\,.$$
- We'll prove that there is $M \in \mathbb{F}_2^{n\times n}$, $M \in \mathsf{P}^{\mathsf{NP}}$!
$$\mathcal{R}_M^{\mathbb{F}_2}(2^{\log n/\log\log n}) \geq \Omega(n^2)\,.$$
- We'll use
  - Orthogonal Vectors
  - Non-deterministic Hierarchy Theorem
  - Rectangular PCPs

Rectangular PCPs

Constraint Satisfaction Problems (CSP)

Definition

A k-CSP is specified by a Boolean function $f\colon \{0,1\}^{\boxed{k}} \to \{0,1\}$. An instance of k-CSP is a formula with n Boolean variables and $m = \mathrm{poly}(n)$ constraints, where each constraint is f applied to a k-tuple of variables or their negations.

3-CSP 2-CSP

3-SAT

Ex. $k=3$ $f = \mathrm{OR}$

$(x_1 \vee \overline{x_2} \vee x_n)$

$(\overline{x_7} \vee x_{10} \wedge x_{11})$

Ex. $k=2$ $f = \mathrm{XOR}$

$(x_1 \oplus x_7)$

$(x_n \oplus \overline{x_5}) = (x_n \oplus x_5 \oplus 1)$

Constraint Satisfaction Problems

Definition

A k-CSP is specified by a Boolean function $f\colon \{0,1\}^k \to \{0,1\}$. An instance of k-CSP is a formula with n Boolean variables and m constraints, where each constraint is f applied to a k-tuple of variables or their negations.

Definition

In Max-kCSP, the goal is to find the maximum number of simultaneously satisfiable clauses.

SAT = can we satisfy all clauses of φ?

MAX-SAT – opt. version of SAT

Given φ, k, can we satisfy $\geq k$ clauses?

$$(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2})$$

is not satisfiable, SAT $\rightarrow$ no

MAX-SAT $\rightarrow$ 3

MAX-CSP

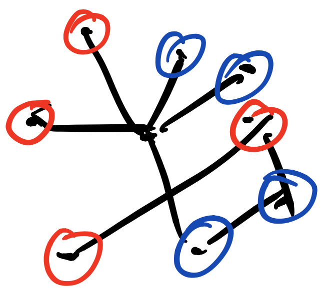
Ex. $k=2$ $\mathcal{F} = XOR$

$(x_1 \oplus x_7)$ $(x_n \oplus x_3)$ ...

MAX-2LIN / MAX-2XOR / MAX-CUT

MAX-CUT problem:

G



cut = # of red-blue edges

MAX-CUT - find a cut (red-blue coloring of vertices) that max # of cut edges (red-blue edges)

Previously: we showed very simple $\frac{1}{2}$-apx for MAX-CUT using randomness

Then we said, pairwise ind. (Toeplitz) derandomizes alg.

every vertex - $x_i \in \{0,1\}$

every edge $(x_1, x_2) \rightarrow (x_1 \oplus x_2)$

$x_i = 0$ if x_i is red

$x_i = 1$ if x_i is blue

sat iff edge is red-blue

G - instance of MAX-CUT

$\Downarrow$

write equivalent

MAX-2XOR fla. φ

Max # of sat clauses in φ

$=$

Max-CUT in G

PCP

- Motivation: Hardness of Approximation

PCP

- Motivation: Hardness of Approximation

- Two views of PCPs:

PCP

- Motivation: Hardness of Approximation

- Two views of PCPs:
    - Probabilistically Checkable Proofs

PCP

- Motivation: Hardness of Approximation

- Two views of PCPs:
  - Probabilistically Checkable Proofs
  - Hardness of Approximation

Probabilistically Checkable Proofs

Recall: $L \in \mathbf{NP}$ if there exists deterministic poly-time V:

$$\begin{aligned} \underline{x \in L} &\implies \quad \underline{\exists \pi} \quad s.t. V^{\pi}(x) = 1 \\ \underline{x \notin L} &\implies \quad \underline{\forall \pi} \quad V^{\pi}\underline{(x)} = 0 \end{aligned}$$

$L = SAT \in NP$

$\varphi \in SAT \Rightarrow \underline{\exists x \in \{0,1\}^n} \quad V^{x}(\varphi) = 1$

$\varphi \notin SAT \Rightarrow \forall x \in \{0,1\}^n \quad V^{x}(\varphi) = 0$

Probabilistically Checkable Proofs

Recall: $L \in \mathbf{NP}$ if there exists deterministic poly-time V:

$$x \in L \implies \quad \exists \pi \quad s.t. V^{\pi}(x) = 1$$
$$x \notin L \implies \quad \forall \pi \quad V^{\pi}(x) = 0$$

In the class **PCP**, V is $\boxed{\text{randomized}}$ and has random access to π

PCP

randomness

queries

Let $r, q : \mathbb{N} \to \mathbb{N}$, then $\mathbf{PCP}(r, q)$ is the class of languages L s.t. there exists a poly-time probabilistic V:

- Efficient: For $x \in \{0,1\}^n$ and a proof π, V has $O(r(n))$ random bits, makes $O(q(n))$ queries to π
- Completeness: If $x \in L$, then V accepts with probability $\geq c = 1$. – perfect completeness
- Soundness: if $x \notin L$, then V accepts with probability $< s < c$.

$= \frac{1}{2}$

PCP for 3-SAT

φ with n vars
m clauses.

$\varphi \in SAT \iff \varphi$ is satisfiable.

Proof(Π) to be $x \in \{0,1\}^n$

$\varphi \in SAT \implies \exists \underline{x \in \{0,1\}^n}$
s.t. efficient V looks
at 1 random clause of φ.
$\implies$ accept φ.

$\varphi \notin SAT \implies \forall x \underline{\in \{0,1\}^n}$
V will detect
unsatisfied clause w.p. $\geq \frac{1}{m}$.

$\log m$ random bits, 3 queries

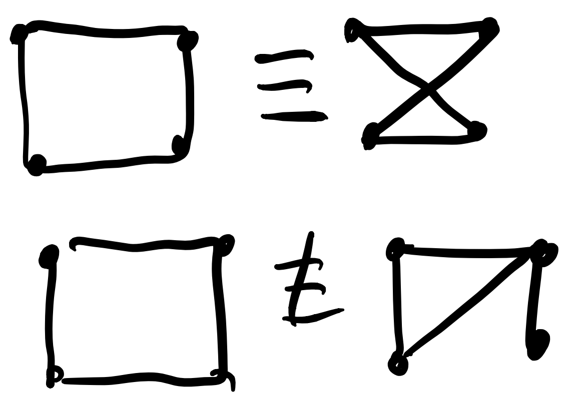
This was example of

$PCP(\log m, 3)$ $\underline{c = 1}$

$s = \underline{1 - \frac{1}{m}}$ for 3-SAT

$GNI(G_1, G_2)$

$G_1 \not\equiv G_2$



$GNI \in PCP(poly(n), 1)$

G_1, G_2 with n

Proof π: $\forall$ every graph on n vertices G.

0 if $G \equiv G_1$

1 if $G \equiv G_2$

0 if $G \not\equiv G_1$, $G \not\equiv G_2$

V has graphs G_1, G_2

picks random $b \in \{1, 2\}$: G_b.

Takes random perm $G_b \to H$

If $G_1 \not\equiv G_2$.

Say, $b=1$. $H \equiv \underline{\underline{G_1}}$

$\Pi(H) = 1$.

V accepts $G_1 \not\equiv G_2$

If $\underline{G_1 \equiv G_2}$

Say, $b=1$, $H \equiv G_1 \equiv G_2$

$\Pi(H) = 0$ or $\Pi(H) = 1$

V accepts w p $\frac{1}{2}$.

PCP THEOREM

Theorem (PCP Theorem [AS92, ALMSS92])

$$\underline{\underline{\mathrm{NP}}} = \mathrm{PCP}(\log n, 1)\,.$$

$O(\log n)$ randomness

$O(1)$ = const queries

$x \in L \Rightarrow$ accept always $\quad c = 1.$

s = any constant

$s = 0.9 \xRightarrow{\text{amplify}} s = 0.01$

$L = \{$ math statements of length n that have proofs of length poly(n) $\}$

$L \in NP$, for every statement there is a proof of length poly(n), Verifier can always check math proofs.

$L \in PCP(\log n, 1)$

Every math proof (of poly-length) can rewritten in some PCP form s.t. one can check correctness of the proof by looking at only 3 bits of the proof.

PCP Theorem

Theorem (PCP Theorem [AS92, ALMSS92])

$$\mathrm{NP} = \mathrm{PCP}(\underline{\log n}, \underline{1}) \,.$$

Theorem (Scaled-up PCP [BFLS91, AS92, ALMSS92])

$$\mathrm{NEXP} = \mathrm{PCP}(\underline{\mathrm{poly}(n)}, 1) \,.$$

<u>Ex.</u> $GNI \in PCP(\mathrm{poly}(n), 1)$

PCP: Hardness of Approximation

PCPT: NP = PCP(log n, 1)

Theorem (PCP Theorem)

There exists a constant $p < 1$ s.t. it's **NP**-*hard to p-approximate Max-3SAT.*

3SAT is NP-hard to solve exactly $\Rightarrow$ 3-SAT is NP-hard

cannot distinguish between 3-SAT φ : $\begin{cases} val(\varphi) = 1 \\ val(\varphi) \leq 0.75 + \varepsilon \end{cases}$

easier ~~than~~ than 3-SAT

Equivalence of Two Views

Example: $\mathbf{NP} = \mathbf{PCP}(\log n, 1)$ implies that k-CSP is $\mathbf{NP}$-hard to p-approximate for some constant p and k.

3-SAT $\in$ NP $\in$ PCP(log n, 1)

φ is a 3-CNF fla. $\exists\, \Pi \in \{0,1\}^n$ assignment $x_1 \ldots x_n$

$\forall\, R \in \{0,1\}^{10 \log n}$

V looks at 3 positions of Π

V decides whether accept/reject φ.

$\forall$ randomness $R \in \{0,1\}^{10 \log n}$

V looks at 3 positions

$x_1 \quad , x_7 \quad , x_{11}$

$(x_1 \wedge x_7 \vee \overline{x_{11}})$ 3-CSP - use whatever fla of 3 xs

$2^{10 \log n}$ different strings R.

" n^{10} clauses

$(x_1 \wedge x_7 \vee \overline{x_{11}})$

$(x_n \wedge x_{n/2} \vee \overline{x_2})$ .....

n^{10} clauses, 3-CSP

Verifier samples a random clause, and verifies whether it's satisfied by proof $x \in \{0,1\}^n$

PCP $c = 1$

$s \le \frac{1}{2}$

φ is SAT $\Rightarrow$ V accepts w.p. 1

φ is not SAT $\Rightarrow$ w.p. $\le \frac{1}{2}$

φ is SAT $\Rightarrow$ V accepts w.p. 1

$\Rightarrow$ every clause of 3-CSP can be sat

φ is not SAT $\Rightarrow$ V accept $\leq \frac{1}{2}$

$\Rightarrow$ $\leq \frac{1}{2}$ clauses can be satisfied.

Hardness of apx view of PCP

PCP starts with like 3SAT

hard $\begin{cases} m \text{ clauses} \\ \leq m-1 \text{ clauses} \end{cases} \Rightarrow \begin{cases} m \text{ clauses} \\ \leq 0.9m \text{ clauses} \end{cases}$

Equivalence of Two Views

Proof View	Approximation View

Equivalence of Two Views

Proof View	Approximation View
PCP Verifier	CSP formula

Equivalence of Two Views

Proof View	Approximation View
PCP Verifier	CSP formula
PCP proof	Assignment to Boolean vars

Equivalence of Two Views

Proof View	Approximation View
PCP Verifier	CSP formula
PCP proof	Assignment to Boolean vars
Length of proof	# of vars

Equivalence of Two Views

Proof View	Approximation View
PCP Verifier	CSP formula
PCP proof	Assignment to Boolean vars
Length of proof	# of vars
# of queries	width of constraints

Equivalence of Two Views

Proof View	Approximation View
PCP Verifier	CSP formula
PCP proof	Assignment to Boolean vars
Length of proof	# of vars
# of queries	width of constraints
# of random bits	$\log$ of # of constraints

Smooth PCPs

$x_1, \ldots, x_n \in \{0,1\}^n$

no favorite variable

Definition

A PCP is smooth if V queries every variable with equal probability (queries every bit of the proof with equal probability).

Smooth PCPs

Definition

A PCP is smooth if V queries every variable with equal probability (queries every bit of the proof with equal probability).

If your proof π is not correct, but differs from it in a few positions, chances are your proof will be accepted by a smooth verifier!

correct $x_1 \dots x_n$ $y_1 \dots y_n$ which differ from $x_1 \dots x_n$ in 1% of positions

Regular PCP may detect; Smooth will not see difference w.h.p.

Rectangular PCPs

Example: Max-2LIN.

Existence of Rectangular PCPs

Theorem (BHPT20)

For any $L \in \mathsf{NTIME}[2^n]$, there exists a PCP verifier V that

Existence of Rectangular PCPs

Theorem (BHPT20)

For any $L \in \mathsf{NTIME}[2^n]$, there exists a PCP verifier V that

- *# of random bits is $n + o(n)$*

Existence of Rectangular PCPs

Theorem (BHPT20)

For any $L \in \mathsf{NTIME}[2^n]$, there exists a PCP verifier V that

- *# of random bits is $n + o(n)$*
- *query complexity is $O(1)$*

Existence of Rectangular PCPs

Theorem (BHPT20)

For any $L \in \mathsf{NTIME}[2^n]$, there exists a PCP verifier V that

- *# of random bits is $n + o(n)$*
- *query complexity is $O(1)$*
- *proof of length $2^n \operatorname{poly}(n)$*

Existence of Rectangular PCPs

Theorem (BHPT20)

For any $L \in \mathsf{NTIME}[2^{\mathsf{n}}]$, there exists a PCP verifier V that

- *# of random bits is $n + o(n)$*
- *query complexity is $O(1)$*
- *proof of length $2^n \operatorname{poly}(n)$*
- *V runs in time $\underline{2^{(1-\varepsilon)n}}$*

Existence of Rectangular PCPs

Theorem (BHPT20)

For any $L \in \mathsf{NTIME}[2^n]$, there exists a PCP verifier V that

- *# of random bits is $n + o(n)$*
- *query complexity is $O(1)$*
- *proof of length $2^n \operatorname{poly}(n)$*
- *V runs in time $2^{(1-\varepsilon)n}$*
- *V is smooth*

Existence of Rectangular PCPs

Theorem (BHPT20)

For any $L \in \mathsf{NTIME}[2^{\mathsf{n}}]$, *there exists a PCP verifier V that*

- *# of random bits is* $n + o(n)$
- *query complexity is* $O(1)$
- *proof of length* $2^n \operatorname{poly}(n)$
- *V runs in time* $2^{(1-\varepsilon)n}$
- *V is smooth*
- *V is almost rectangular*

Existence of Rectangular PCPs

Theorem (BHPT20)

For any $L \in \mathbf{NTIME}[2^n]$, there exists a PCP verifier V that

- *# of random bits is $n + o(n)$* — opt
- *query complexity is $O(1)$*
- *proof of length $2^n \operatorname{poly}(n)$*
- *V runs in time $2^{(1-\varepsilon)n}$*
- *V is smooth*
- *V is almost rectangular* — next time
- *the CSP problem is almost MAX-2LIN* $\equiv$ MAX-CUT