NP $\subseteq$ PCP(poly(n),1) Proof Completed
Hardness of Approximating CLIQUE

Recall QUADEQ:

Instance:

- An $m \times n^2$ matrix A with entries in $\{-1,0,1\}$
- An m-dimensional bit vector b

Problem: Is there an n-dimensional bit vector u such that $A\,(u \otimes u)^T = b^T$?

Recall: Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g, V checks:

- *Linearity*: f is $(1-\delta)$-close to $f' = WH(u)$, i.e., $Pr_x [f(x) = f'(x)] \geq (1-\delta)$, and g is $(1-\delta)$-close to $g' = WH(w)$
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g' = WH(u \otimes u)$ then $A (u \otimes u)^T = b^T$.

Recall: Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g, V checks:

- *Linearity*: f is $(1-\delta)$-close to $f' = WH(u)$,
  i.e., $Pr_x [f(x) = f'(x)] \geq (1-\delta)$, and
  g is $(1-\delta)$-close to $g' = WH(w)$
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g' = WH(u \otimes u)$ then $A (u \otimes u)^T = b^T$.

Problem: Need to update the Consistency and Satisfiability checks, to account for the fact that f and g are close to, but may not equal, f′ and g′.

Recall: Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g, V checks:

- *Linearity*: f is $(1-\delta)$-close to $f' = WH(u)$, i.e., $\Pr_x [f(x) = f'(x)] \geq (1-\delta)$, and g is $(1-\delta)$-close to $g' = WH(w)$
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g' = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

Problem: Need to update the Consistency and Satisfiability checks, to account for the fact that f and g are close to, but may not equal, f′ and g′.

Solution: *Local decoding*

Local Decoding

Given arbitrary $x \in \{0,1\}^n$ and a function f that is $(1-\delta)$-close to a unique linear function f', compute $f'(x)$

Local Decoding

Given arbitrary $x \in \{0,1\}^n$ and a function f that is $(1-\delta)$-close to a unique linear function f', compute $f'(x)$

Local decoding: Given f and x

Choose random $x' \in \{0,1\}^n$

Let x'' be such that $x = x' + x''$

Let $y' = f(x')$ and $y'' = f(x'')$

Output $y' + y''$

Local Decoding

Given arbitrary $x \in \{0,1\}^n$ and a function f that is $(1-\delta)$-close to a unique linear function f', compute $f'(x)$

Local decoding: Given f and x

Choose random $x' \in \{0,1\}^n$

Let x'' be such that $x = x' + x''$

Let $y' = f(x')$ and $y'' = f(x'')$

Output $y' + y''$

Theorem: With probability at least $1 - 2\delta$, $f'(x) = y' + y''$

Local Decoding

Given arbitrary $x \in \{0,1\}^n$ and a function f that is $(1-\delta)$-close to a unique linear function f', compute $f'(x)$

Local decoding: Given f and x

Choose random $x' \in \{0,1\}^n$
Let x'' be such that $x = x' + x''$
Let $y' = f(x')$ and $y'' = f(x'')$
Output $y' + y''$

Theorem: With probability at least $1 - 2\delta$, $f'(x) = y' + y''$
Proof sketch: follows from two facts:

- With probability at least $1 - 2\delta$, $y = f'(x\ )$ and $y' = f'(x')$
- By linearity of f', $f'(x) = f'(x' + x'') = f'(x') + f'(x'')$

Summary: PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g

V checks:

- *Linearity*: f and g are $(1-\delta)$-close to $f' = WH(u)$ and $g' = WH(w)$
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g' = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

If the linearity check conditions hold, then with high probability all of the calculations of f' and g' in the Consistency and Satisfiability tests are correct

Max Clique is $(2 – \varepsilon)$-hard to approximate

Max Clique is (2 – ε)-hard to approximate

Max Clique: Given an undirected graph G=(V,E), find the largest subset of V such that every pair of nodes in the subset is connected by an edge of E

Theorem: For any $\varepsilon > 0$, if Max Clique has a poly-time $(2-\varepsilon)$-approximation algorithm, then NP=P.

Max Clique is $(2 – \varepsilon)$-hard to approximate

Proof: Let $L \in NP$, let V be a PCP for L. Fix instance x of L.

Using V, we'll describe a mapping $x \rightarrow G_x$ from instances of L to instances of Clique, and apply the Gap Lemma.

Max Clique is $(2 - \varepsilon)$-hard to approximate

Proof: Let $L \in NP$, let V be a PCP for L. Fix instance x of L.

Using V, we'll describe a mapping $x \rightarrow G_x$ from instances of L to instances of Clique, and apply the Gap Lemma.

Notation: Let the q positions of the proof that V queries on coin flip sequence τ be $b_{\tau,1}, b_{\tau,2}, \ldots, b_{\tau,q}$.

Max Clique is (2 – ε)-hard to approximate

Mapping : x → G_x

Max Clique is $(2 - \varepsilon)$-hard to approximate

Mapping : $x \rightarrow G_x$

Example: Suppose that V

- Uses two random bits on instance x
- Makes $q = 3$ queries (on instances of any length)
- On random string $\tau = 01$, queries bits

$b_{\tau,1} = 2$, $b_{\tau,2} = 7$, and $b_{\tau,3} = 21$

Max Clique is (2 – ε)-hard to approximate

Mapping : x → G_x

proof at position 2	proof at position 7	proof at position 21	verifier's decision on random string $\tau = 01$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Max Clique is $(2 - \varepsilon)$-hard to approximate

Mapping : $x \rightarrow G_x$

proof at position 2	proof at position 7	proof at position 21	verifier's decision on random string $\tau = 01$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Then G_x contains nodes (τ = 01, 001), (τ= 01,100), and (τ= 01,111)

Max Clique is (2 – ε)-hard to approximate

Mapping : $x \rightarrow G_x$

- Node $(\tau, v_1 v_2 \ldots v_q)$ is in graph G_x if V accepts with random string τ and query results $v_1, v_2, \ldots, v_q$.

Max Clique is $(2 - \varepsilon)$-hard to approximate

Mapping : $x \rightarrow G_x$

- Node $(\tau, v_1 v_2 \ldots v_q)$ is in graph G_x if V accepts with random string τ and query results $v_1, v_2, \ldots, v_q$. G_x has $\leq 2^{r(|x|)} 2^q$ nodes, where $r(|x|) = O(\log |x|)$

is the number of random bits of V on input x

Max Clique is $(2 - \varepsilon)$-hard to approximate

Mapping : $x \rightarrow G_x$

- Node $(\tau, v_1 v_2 \ldots v_q)$ is in graph G_x if V accepts with random string τ and query results $v_1, v_2, \ldots, v_q$. G_x has $\leq 2^{r(|x|)} 2^q$ nodes, where $r(|x|) = O(\log |x|)$

is the number of random bits of V on input x

- Edges are between *compatible* pairs of nodes $(\tau, v_1 v_2 \ldots v_q)$ and $(\tau', v_1' v_2' \ldots v_q')$ i.e., for any i and j, if $b_{\tau,i} = b_{\tau',j}$ then $v_i = v_j'$

Max Clique is $(2 - \varepsilon)$-hard to approximate

Mapping : $x \rightarrow G_x$

- Node $(\tau, v_1 v_2 \ldots v_q)$ is in graph G_x if V accepts with random string τ and query results $v_1, v_2, \ldots, v_q$. G_x has $\leq 2^{r(|x|)} 2^q$ nodes, where $r(|x|) = O(\log |x|)$ is the number of random bits of V on input x

- Edges are between *compatible* pairs of nodes $(\tau, v_1 v_2 \ldots v_q)$ and $(\tau', v_1' v_2' \ldots v_q')$ i.e., for any i and j, if $b_{\tau,i} = b_{\tau',j}$ then $v_i = v_j'$

- G_x can be computed in poly time

Max Clique is (2 – ε)-hard to approximate

Claim: $x \in L \Rightarrow \mathrm{Opt}(G_x) = 2^{r(|x|)}$ and

$x \notin L \Rightarrow \mathrm{Opt}(G_x) \leq (1/2)\, 2^{r(|x|)}$

Max Clique is $(2 - \varepsilon)$-hard to approximate

Claim: $x \in L \Rightarrow Opt(G_x) = 2^{r(|x|)}$ and
$x \notin L \Rightarrow Opt(G_x) \leq (1/2)\, 2^{r(|x|)}$

Proof sketch when $x \in L$:

Max Clique is (2 – ε)-hard to approximate

Claim: $x \in L \Rightarrow Opt(G_x) = 2^{r(|x|)}$ and
$x \notin L \Rightarrow Opt(G_x) \leq (1/2)\, 2^{r(|x|)}$

Proof sketch when $x \in L$: Then on some proof π, V accepts with probability 1.

The $2^{r(|x|)}$ nodes "compatible" with this proof (one node per random string τ) form a clique.

Max Clique is $(2 - \varepsilon)$-hard to approximate

Claim: $x \in L \Rightarrow \mathrm{Opt}(G_x) = 2^{r(|x|)}$ and
$x \notin L \Rightarrow \mathrm{Opt}(G_x) \leq (1/2)\, 2^{r(|x|)}$

Proof sketch when $x \notin L$:

Max Clique is $(2 - \varepsilon)$-hard to approximate

Claim: $x \in L \Rightarrow \mathrm{Opt}(G_x) = 2^{r(|x|)}$ and
$x \notin L \Rightarrow \mathrm{Opt}(G_x) \leq (1/2)\, 2^{r(|x|)}$

Proof sketch when $x \notin L$: Then on all proofs π, V accepts with probability at most 1/2.

The existence of a clique of size greater than $(1/2)\, 2^{r(|x|)}$ would imply a proof on which V accepts with probability $> ½$.

Max Clique is (2 – ε)-hard to approximate

Claim: $x \in L \Rightarrow \mathrm{Opt}(G_x) = 2^{r(|x|)}$ and
$x \notin L \Rightarrow \mathrm{Opt}(G_x) \leq (1/2)\, 2^{r(|x|)}$

Max Clique is $(2 - \varepsilon)$-hard to approximate

Claim: $x \in L \Rightarrow \mathrm{Opt}(G_x) = 2^{r(|x|)}$ and

$x \notin L \Rightarrow \mathrm{Opt}(G_x) \leq (1/2)\, 2^{r(|x|)}$

$< (1-c)\, 2^{r(|x|)}$ for any $c < 1/2$

Recall Gap Lemma

Gap Lemma: Let L be NP-complete. Suppose that there is a poly-time mapping from any instance x of L to instance x′ of maximization problem ∏ such that

$x \in L \Rightarrow Opt(x') = g(x)$ and

$x \notin L \Rightarrow Opt(x') < (1\text{-}c)\, g(x)$

where $g(x) \in \mathbb{N}$, g is poly-time computable, and $0<c<1$.

If ∏ has a poly-time approximation algorithm with approximation ratio $1 + c/(1\text{-}c)$, then NP = P.

Max Clique is (2 – ε)-hard to approximate

Claim: $x \in L \Rightarrow \mathrm{Opt}(G_x) = 2^{r(|x|)}$ and

$x \notin L \Rightarrow \mathrm{Opt}(G_x) \leq (1/2)\, 2^{r(|x|)}$

$< (1-c)\, 2^{r(|x|)}$ for any $c < 1/2$

Max Clique is (2 – ε)-hard to approximate

Claim: $x \in L \Rightarrow \mathrm{Opt}(G_x) = 2^{r(|x|)}$ and
$x \notin L \Rightarrow \mathrm{Opt}(G_x) \leq (1/2)\, 2^{r(|x|)}$
$< (1-c)\, 2^{r(|x|)}$ for any $c < 1/2$

We can now apply the Gap Lemma to conclude that if Clique has a poly-time approximation algorithm with approximation ratio $1 + c/(1-c)$ then NP = P.

Finally, for any $\varepsilon > 0$, there is $c < 1/2$ such that

$$2 - \varepsilon \leq 1 + c/(1-c).$$