Unique Games Conjecture

Jongseo Lee (leejseo)

KAIST

November 8, 2022

Note

The materials are mostly from:

- The book "Computational Intractability: A Guide to Algorithmic Lower Bounds" Online Book Link
- Stanford University CS 354: Unfulfilled Algorithmic Fantasy Course Material Link

Approximating NP-hard Problems

MAX 3SAT: upper and lower bounds match

- There is a polynomial-time $\frac{7}{8}$-approximation algorithm.
- Assuming $P \neq NP$, there is no polynomial-time $(\frac{7}{8} + \delta)$-approximation algorithm.

Vertex Cover: upper and lower bounds does not match

- There is a polynomial-time 2-approximation algorithm.
- Assuming $P \neq NP$, there is no polynomial-time $(10\sqrt{5} - 21 - \delta)$-approximation algorithm. (Note that $10\sqrt{5} - 21 \approx 1.3606$.)

Getting Lower and Upper Bounds Match?

So... what to do? It is very unlikely to...

- Get a better approximation algorithm for Vertex Cover.
- Use even harder math or more finely tunde prover-systems to get better lower bounds.

New Assumption

Assuming something called Unique Games Conjecture, we can obtain

1. *matching* upper and lower bounds for approximating Vertex Cover and several other problems
2. *better* lower bounds for some problems
3. *better* lower bounds for some problems from *variants* of Unique Games Conjecture

Label Cover

Label Cover

- Instance:
  1. A bipartite graph $G = (V \cup W, E)$ and two positive integers $a \geq b$.
  2. For each edge $e \in E$, surjection $\pi_e : [a] \to [b]$.
- Question: Is there a label ℓ with $\ell(v) \in [a]$ for $v \in V$ and $\ell(w) \in [b]$ for $w \in W$ such that, for every $e = (v, w) \in E$, $\pi_e(\ell(v)) = \ell(w)$?

For the approximate version, we want to know whether we can satisfy the fraction δ of the constraints.

Theorem

Label Cover is NP-complete.

Distinguishing Good and Hard Instances

For an instance of the label cover problem U, define $OPT(U) = \delta$ where δ is the largest fraction of edges that can be satisfied.

Theorem

Let $0 < \delta < 1$. There exists a constant C such that:

- Let U be the instance of the label cover problem where $a = \Theta((1/\delta)^C)$.
- It is NP-hard to distinguish between the following cases:
  - $OPT(U) = 1$ (there is a label covering)
  - $OPT(U) \leq \delta$ (no label covering can be that good)

Unique Label Cover

Unique Label Cover

- Instance:
  1. A bipartite graph $G = (V \cup W, E)$ and a positive integer a.
  2. For each edge $e \in E$, bijection $\pi_e : [a] \to [a]$.
- Question: Is there a label ℓ with $\ell(v), \ell(w) \in [a]$ for $v \in V$ and $w \in W$ such that, for every $e = (v, w) \in E$, $\pi_e(\ell(v)) = \ell(w)$?

Surprisingly, we need to allow 2-sided error to get a hard problem from Unique Label Cover.

Theorem

Let $0 < \delta < 1$. The following two cases can be distinguished in polynomial time:

- $OPT(U) = 1$
- $OPT(U) \leq \delta$

The Unique Games Conjecture (UGC)

The Unique Games Conjecture (UGC)

Let $0 < \varepsilon, \delta < 1$. There exists a constant C such that:

- Let U be the instance of the unique label cover problem where $a = C$.
- It is NP-hard to distinguish between the following cases:
  - $OPT(U) \geq 1 - \varepsilon$ (there is a fairly good approximate unique label covering)
  - $OPT(U) \leq \delta$ (no unique label covering can be that good)

Approximating Max Cut

Semi Definite Programming (SDP)

Instance:

- Numbers $c_{i,j}$ for $1 \leq i, j \leq n$
- Numbers $a_{i,j,k}$ for $1 \leq i, j \leq n, 1 \leq k \leq m$
- Numbers b_k for $1 \leq k \leq m$

Qustion:

- Minimize

$$\sum_{1 \leq i,j \leq n} c_{i,j}(x^i \cdot x^j)$$

- Subject to

$$\sum_{1 \leq i,j \leq n} a_{i,j,k}(x^i \cdot x^j) \leq b_k$$

for every $1 \leq k \leq m$ where $x^1, \ldots, x^n \in \mathbb{R}^n$.

Approximating Max Cut

Another Formulation of Max Cut

- Constraints: $x_1, x_2, \cdots, x_n \in \{-1, 1\}$
- Objective function: maximize $\frac{1}{2} \sum_{(i,j) \in E} w_{i,j}(1 - x_i x_j)$

SDP Relaxation of Max Cut

- Constraints: $x_i \in \mathbb{R}^n$ such that $|x_i| = 1$ for $i = 1, 2, \cdots, n$
- Objective Function: maximize $\frac{1}{2} \sum_{(i,j) \in E} w_{i,j}(1 - x_i x_j)$

Approximating Max Cut

The Algorithm Based on SDP Relaxation

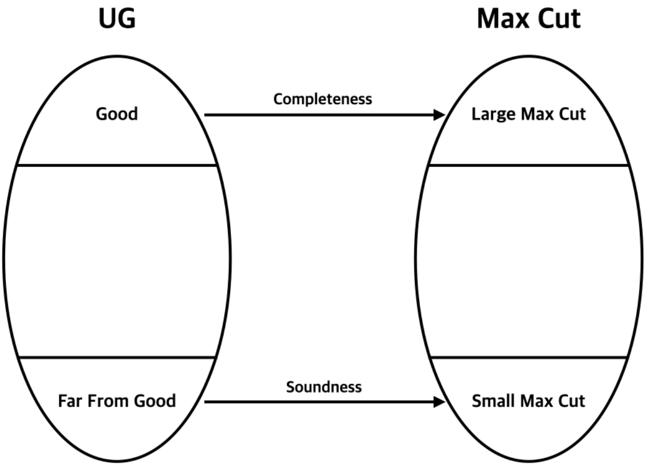
1. Solve SDP Relaxation version of Max Cut to obtain vectors $x_1, x_2, \cdots, x_n$.
2. Pick a random vector r on the n-dimensional unit sphere.
3. Let $V = V_1 \cup V_2$ where $V_1 = \{i : x_i \cdot r \geq 0\}$ and $V_2 = \{i : x_i \cdot r < 0\}$.

Let $SDPOPT$ be the optimal value of SDP relaxation and OPT be the optimal value of the original problem.

Theorem (Integrality Gap)

$\mathbb{E}[\text{cut found}] \geq 0.87586 SDPOPT \geq 0.87586 OPT$.

Inapproximability of Max Cut



Inapproximability of Max Cut

The Long Code

For a message $m \in [q]$, encode m to be a function $f_m : \{-1, 1\}^q \to \{-1, 1\}$ such that $f_m(x) = x_m$.

Although the long code takes 2^q bits long to represent $\log q$ bits long information, there are many useful properties. Appendix

Inapproximability of Max Cut: Constructing the Reduction

Given a unique label covering instance $G = (V \cup W, E)$ and $\pi_e : [a] \to [a]$, we construct a set of 2^a vertices:

$$Cloud(u) := \{u_x : x \in \{\pm 1\}^a\}.$$

Then, for each $(u, v) \in E$, add an edge between u_x and v_y if and only if $x = -\pi_{u,v}(y)$ by appling permutation π to the coordinates.

Inapproximability of Max Cut: Constructing the Reduction

A cut $(S, \bar{S})$ from the graph of cloud vertices is corresponded to set of functions f^u for each vertex u of the original graph:

$$f^u(x) = \begin{cases} 1 & u_x \in S \\ -1 & u_x \notin S \end{cases}.$$

Let's define a cut S in our reduction as $f^u(x) = f_{\ell(u)}(x)$

Inapproximability of Max Cut: Constructing the Reduction

This reduction does not satisfy Soundness, but, still satisfies completeness.
Suppose ℓ satisfies an edge $(u, v) \in E$.
Then, for each edge (u_x, v_y) in our reduction, we have

$$f_{\ell(u)}(x) = x_{\ell(u)} = -y_{\pi_{u,v}}(\ell(u)) = -y_{\ell(v)} = -f_{\ell(v)}(y),$$

so if $\geq (1 - \varepsilon)$ fraction of the unique label coloring edges are satisfied, the cut we found crosses $\geq (1 - \varepsilon)$ fraction of edges.

Inapproximability of Max Cut: Fixing the Reduction

Instead of just adding edges, let's give a weight to each edge.

$$w(u_x, v_y) := \Pr_z[z = -\pi_{u,v}(y)]$$

The PCP verifier for Max Cut

- Pick a vertex $v \in V$ and its two neighbors $w, w' \in W$ at random.
- Pick $x \in \{-1, 1\}^a$ at random.
- Pick $z \in \{-1, 1\}^a$ at random from x by flipping each coordinate with probability p.
- Accept iff

$$f_{\ell(w)}(x \circ \pi_{v,w}) \neq f_{\ell(w')}((-z) \circ \pi_{v,w'}).$$

The completeness and the soundness of the reduction comes from the properties of the long code. Proof Sketch

UGC Implies Optimal Lower Bounds

Vertex Cover on k-Hypergraphs

- Instance: k-Hypergraph $G = (V, E)$. Note that $E \subseteq \binom{V}{k}$.
- Question: Find the size of the smallest set $V' \subseteq V$ such that every edge $E = \{v_1, \cdots, v_k\}$ has some element V' in it.

Max 2SAT

- Instance: A 2CNF formula $\phi = C_1 \wedge \cdots \wedge C_k$.
- Question: How many clauses that an assignment can satisfy?
- Note: There's a randomized polynomial-time algorithm returns $\geq \beta OPT$ where $\beta \approx 0.94$. Let $\alpha_{LLZ} = \frac{1}{\beta}$.

Maximum Acyclic Subgraph (MAS)

- Instance: A directed graph $G = (V, E)$.
- Question: What is the size of the largest acyclic subgraph?

UGC Implies Optimal Lower Bounds

Problem	Best Approx	UGC	$P \neq NP$
Vertex Cover	2	$2 - \varepsilon$	1.36
Vertex Cover on k-uniform hypergraphs ($k \geq 3$)	k	$k - \varepsilon$	$k - 1 - \varepsilon$
Max Cut	α_{MC}	$\alpha_{MC} - \varepsilon$	$17/16 - \varepsilon$
Max 2SAT	α_{LLZ}	$\alpha_{LLZ} - \varepsilon$	APX-hard
MAS	2	$2 - \varepsilon$	$66/65 - \varepsilon$

Note that $\alpha_{MC} \approx 1.14$, $\alpha_{LLZ} \approx 1.06$, and $17/16 \approx 1.06$.

UGC Implies Good but Not-Optimal Lower Bounds

Feedback Arc Set

- Instance: A directed graph $G = (V, E)$.
- Question: What is the size of the smallest set of edges that contains at least one edge from every directed cycle?

Sparsest Cut

- Instance: A graph $G = (V, E)$.
- Question: What is the minimum possible sparsity of a set of vertices of G? Note that sparsity of $V' \subseteq V$ is defined as:

$$\frac{|E(V', V - V')|}{\min\{|V'|, |V - V'|\}}.$$

UGC Implies Good but Not-Optimal Lower Bounds

Min-2SAT-Deletion

- Instance: A 2CNF formula $\phi = C_1 \wedge \cdots \wedge C_k$ where no clause is of the form $\bar{x} \vee \bar{y}$.
- Question: What is the maximum number of clauses that can be satisfied?

Min Uncut

- Instance: A graph $G = (V, E)$.
- Question: Find $V' \subseteq V$ that minimizes $|E(V, V')| + E|(V - V', V - V')|$.

UGC Implies Good but Not-Optimal Lower Bounds

Problem	Best Approx	UGC	$P \neq NP$
Fback Arc Set	$O(\log n \log\log n)$	$\omega(1)$	1.36
Sparsest Cut	$O(\sqrt{\log n})$	$\omega(1)$	None
Min-2SAT-Del	$O(\sqrt{\log n})$	$\omega(1)$	APX-hard
Min Uncut	$O(\sqrt{\log n})$	$\omega(1)$	APX-complete

Note that the value 1.36 is obtained by the reduction from Vertex Cover.

UGC+ Implies Some Lower Bounds

Coloring k-COL Graph

- Instance: A graph G that is (promised as) k-colorable
- Question: Find k' coloring of G.
- Note: We will abuse the notation and call a lower bound on k' as an approximation factor.

Scheduling with Precedence Constraints

- Instance: An acyclic graph G of jobs. For each (i, j), job i should be completed before job j. Each job has a processing time p_i and an importance weight w_i.
- Question: Output a linear ordering of the jobs. Note that we can obtain a completion time t_i for each i from the ordering. Let the i-th job has a completion time c_i. Minimize $\sum c_i w_i$.

UGC+ Implies Some Lower Bounds

Problem	Best Approx	UGC+	$P \neq NP$
Coloring 3-COL Graph	$N^{0.211}$	$\omega(1)$	4
Coloring 2d-COL Graph	$N^{1-\frac{3}{2d+1}}$	$\omega(1)$	$d^{\Omega(\log d)}$
Sch. with Prec. Const.	2	2ε	None

Properties of the Long Code

Distance Between Two Code Words

For two different messages m and m',

$$\Pr_x[f_m(x) \neq f_{m'}(x)] = \Pr_x[x_m \neq x_{m'}] = \frac{1}{2}.$$

k-juntas: Generalized Version of the Long Code

If the value of a function f is completely determined by k variables, it is called k-junta.

Properties of the Long Code

Back to Slide

Given x, sample $z = x + (\textit{noise})$ by filliping each bit of x with the probability of p.

Testing the Long Code

If f is a long code, $\Pr[f(x) = -f(-z)] = 1 - p$.

Testing the Majority Function

Let $Maj(x) := \text{sign}(\sum x_i)$. $\Pr[Maj(x) = -Maj(-z)] = \theta/\pi$ where $\cos(\theta) = z \cdot x = 1 - 2p$.

Theorem

For all ε, p, there are constants k, δ such that for all $f : \{\pm 1\}^q \to \{\pm 1\}$, one of the following holds:

- $\Pr[f(x) = -f(-z)] \leq \Pr[Maj(x) = -Maj(-z)] + \varepsilon$
- f is ε-close to a k-junta

Proof Sketch

Back to Slide

Completeness

If a labeling satisfies $\geq (1-\varepsilon)$ fraction of edges, we get both the edges used in PCP are satisfied with probability $\geq (1-2\varepsilon)$ and they get accepted with probability $(1-p)$. Therefore, the verifier accepts with probability $\geq (1-2\varepsilon)(1-p)$.

Soundness (involves hard maths)

If some long codes get accepted with $\geq \arccos(1-2p)/\pi + \varepsilon$ probability, one can obtain a labeling satisfying $\geq \delta'(\varepsilon, p)$ fraction of edges. Since δ' does not depends on the label set size a, taking a large enough ends the proof.