On the Unique Games Conjecture

Fatima-Zahra Moataz

COATI, INRIA, I3S(CNRS/UNS), Sophia Antipolis, France

March 11,2015

The Unique Games Conjecture (UGC)

- Proposed by Subhash Khot in 2002 [Kho02]
- It states that a problem called Unique Games (UG) is hard to approximate
- Gap-preserving reductions from UG $\rightarrow$ inapproximability results for several other problems
- The conjecture motivated work in the analysis of boolean functions, geometry ...

Outline

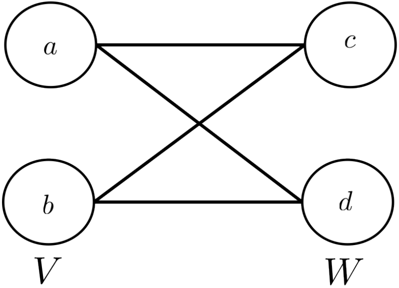
1. Game, what game?
   - Label cover
   - Why Label cover?
2. The conjecture
3. Implications of UGC
   - Analysis of boolean functions
   - Metric embeddings
   - Inapproximability
     - MaxCut
     - UGC and SDP
4. UGC: True or False?

Plan

Label Cover Problem (LC)

Input:

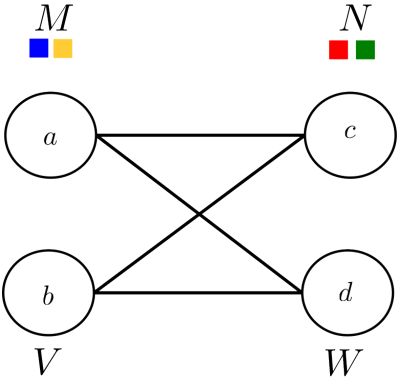
- A bipartite graph $((V, W), E)$



Label Cover Problem (LC)

Input:

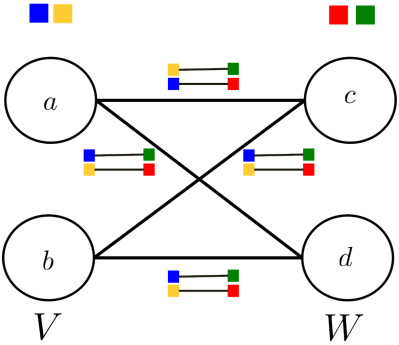
- A bipartite graph $((V, W), E)$
- Two sets of labels M and N



Label Cover Problem (LC)

Input:

- A bipartite graph $((V, W), E)$
- Two sets of labels M and N
- $\forall\ (v, w) \in E$, $\pi_{v,w} : M \to N$

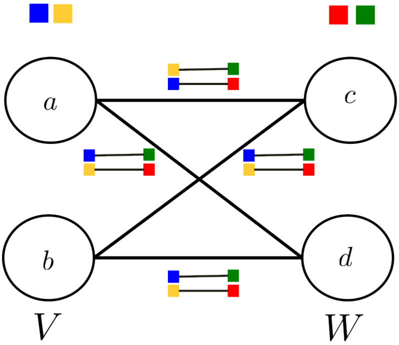


Label Cover Problem (LC)

Input:

- A bipartite graph $((V, W), E)$
- Two sets of labels M and N
- $\forall\ (v, w) \in E$, $\pi_{v,w} : M \to N$

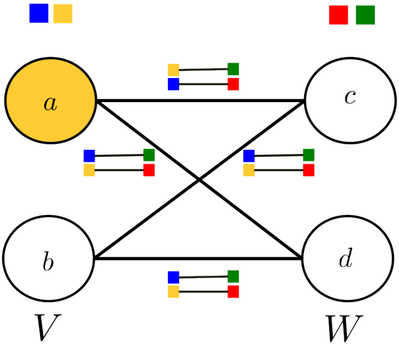
A labeling of the vertices of G:
$l : V \to N$ and $l : W \to M$.
An edge (v, w) is satisified if
$\pi_{v,w}(l(v)) = l(w)$



Label Cover Problem (LC)

Input:

- A bipartite graph $((V, W), E)$
- Two sets of labels M and N
- $\forall\ (v, w) \in E,\ \pi_{v,w} : M \to N$



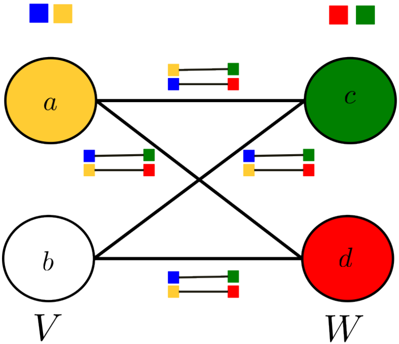
Output: An (optimal) labeling which maximizes the number of satisfied edges

For an instance $\mathcal{U}$ of LC, $OPT(\mathcal{U})$ = fraction of edges satisfied by an optimal labeling of $\mathcal{U}$.

Label Cover Problem (LC)

Input:

- A bipartite graph $((V, W), E)$
- Two sets of labels M and N
- $\forall\ (v, w) \in E,\ \pi_{v,w} : M \rightarrow N$



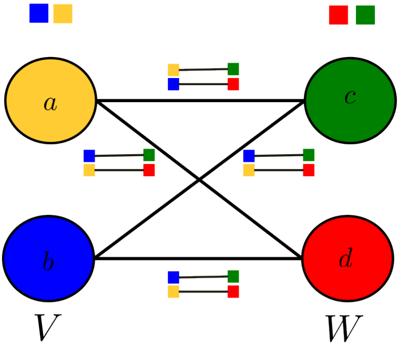
Output: An (optimal) labeling which maximizes the number of satisfied edges

For an instance $\mathcal{U}$ of LC, $OPT(\mathcal{U}) =$ fraction of edges satisfied by an optimal labeling of $\mathcal{U}$.

Label Cover Problem (LC)

Input:

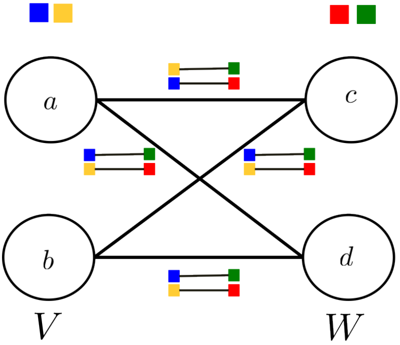
- A bipartite graph $((V, W), E)$
- Two sets of labels M and N
- $\forall\ (v, w) \in E,\ \pi_{v,w} : M \to N$



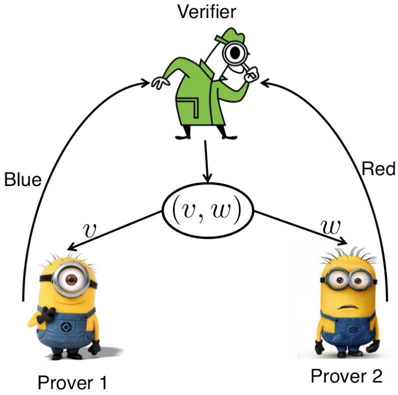
Output: An (optimal) labeling which maximizes the number of satisfied edges

For an instance $\mathcal{U}$ of LC, $OPT(\mathcal{U}) =$ fraction of edges satisfied by an optimal labeling of $\mathcal{U}$.

Where is the game?



Label Cover



2-Prover 1-Round Game

Unique Label Cover (ULC)

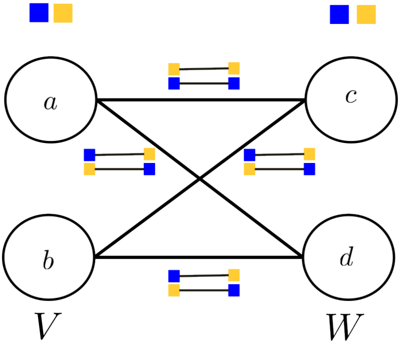
The label cover problem $\mathcal{L} = (G, M, N, \pi_{vw})$ is called unique if:

- $M = N$
- $\forall (v, w) \in E$, $\pi_{v,w}$ is a bijection (permutation)

Unique Label Cover (ULC)

The label cover problem $\mathcal{L} = (G, M, N, \pi_{vw})$ is called unique if:

- $M = N$
- $\forall (v, w) \in E$, $\pi_{v,w}$ is a bijection (permutation)



How is Label Cover (LC) useful?

It is all due to the following theorem:

Theorem

For $\epsilon > 0$, it is NP-hard to decide whether a Label Cover problem:

- *satisfies all edges ($OPT = 1$)*
- *satisfies at most a fraction ϵ of the edges ($OPT \leq \epsilon$)*

Proved with PCP theorem [AS98] + Raz's Parallel Repetition Lemma [Raz98]

- Reductions from $LC(1, \epsilon)$ have allowed to prove inapproximability results for many problems.
- Where does this problem fall short? $\rightarrow$ 2-CSPs
- Mostly because of the "many-to-one"ess of the constraints.
- How about having a stronger result? the same inapproximability theorem for Unique Label Cover?

Plan

The Unique Games Conjecture[Kho02]

Conjecture

For $\epsilon, \delta > 0$, it is NP-hard to decide whether a Unique Label Cover problem:

- *satisfies at least $1 - \epsilon$ fraction of the edges (OPT $\geq 1 - \epsilon$)*
- *satisfies at most a fraction δ of the edges (OPT $\leq \delta$)*

The Unique Games Conjecture[Kho02]

Conjecture

For $\epsilon, \delta > 0$, it is NP-hard to decide whether a Unique Label Cover problem:

- *satisfies at least $1 - \epsilon$ fraction of the edges (OPT $\geq 1 - \epsilon$)*
- *satisfies at most a fraction δ of the edges (OPT $\leq \delta$)*

Why $1 - \epsilon$ and not 1?

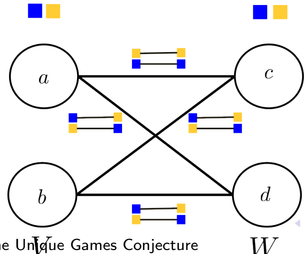
The Unique Games Conjecture[Kho02]

Conjecture

For $\epsilon, \delta > 0$, it is NP-hard to decide whether a Unique Label Cover problem:

- *satisfies at least $1 - \epsilon$ fraction of the edges (OPT $\geq 1 - \epsilon$)*
- *satisfies at most a fraction δ of the edges (OPT $\leq \delta$)*

Why $1 - \epsilon$ and not 1?
$\rightarrow$ Deciding if all edges can be satisfied is easy.



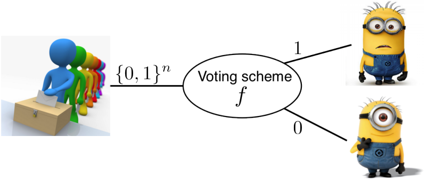
Plan

1. Game, what game?
   - Label cover
   - Why Label cover?
2. The conjecture
3. Implications of UGC
   - Analysis of boolean functions
   - Metric embeddings
   - Inapproximability
     - MaxCut
     - UGC and SDP
4. UGC: True or False?

To the Unique Games Conjecture are associated:

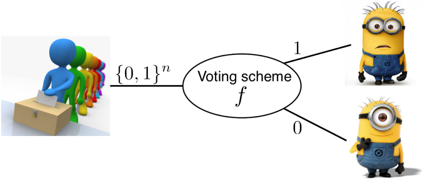
- Non-conditional results
  - Analysis of boolean functions
  - Metric embeddings
- Conditional results: Inapproximability

Majority is Stablest [MOO05]



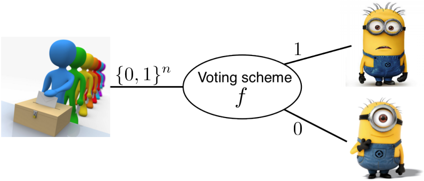
- $f : \{0,1\}^n \rightarrow \{0,1\} \rightarrow$ voting scheme.

Majority is Stablest [MOO05]



- $f : \{0,1\}^n \to \{0,1\} \to$ voting scheme.
- Dictatorship $\to f(x_1, \ldots, x_n) = x_i$ for some i

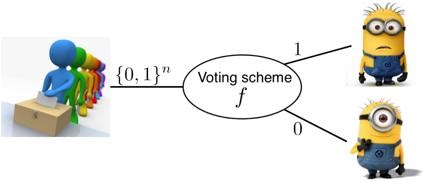
Majority is Stablest [MOO05]



- $f : \{0,1\}^n \to \{0,1\} \to$ voting scheme.
- Dictatorship $\to f(x_1, \ldots, x_n) = x_i$ for some i
- Influence of voter i in a scheme f:

$$Pr_{x \in \{1,-1\}^n}(f(x_1, \ldots, x_i, \ldots, x_n) \neq f(x_1, \ldots, -x_i, \ldots, x_n))$$

Majority is Stablest [MOO05]



- $f : \{0,1\}^n \rightarrow \{0,1\} \rightarrow$ voting scheme.
- Dictatorship $\rightarrow f(x_1, \ldots, x_n) = x_i$ for some i
- Influence of voter i in a scheme f:

$$Pr_{x \in \{1,-1\}^n}(f(x_1, \ldots, x_i, \ldots, x_n) \neq f(x_1, \ldots, -x_i, \ldots, x_n))$$

- Noise stability$_\rho$ of a scheme f: Probability that the result does not change if a random fraction ρ of voters flip their votes.

Studying problems under UGC a distinction was made between:

- Dictatorships
- Schemes that are far from being dictatorships

Studying problems under UGC a distinction was made between:

- Dictatorships
- Schemes that are far from being dictatorships

A question has arised: between schemes that are far from being dictatorships, what is the stablest scheme?

Studying problems under UGC a distinction was made between:

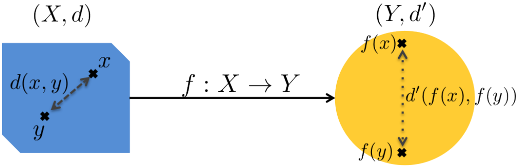
- Dictatorships
- Schemes that are far from being dictatorships

A question has arised: between schemes that are far from being dictatorships, what is the stablest scheme?

The answer: Majority is Stablest

The "Majority is Stablest" (MIS) theorem [MOO05] states that the Majority function maximizes noise stability among balanced boolean functions on the discrete cube with "small" influences.

Metric embedding[KV05]



$$L \times d(x, y) \leq d'(f(x), f(y)) \leq C \times L \times d(x, y)$$

$$\implies f \text{ has distortion } C$$

Metric Embedding: Goemans-Linial Conjecture

- **Goemans Linial Conjecture:** Every negative type metric embeds into l_1 with constant distortion.
  (d is a negative type metric if $\sqrt{d}$ is isometrically embeddable in l_2.)
-

Metric Embedding: Goemans-Linial Conjecture

- **Goemans Linial Conjecture:** Every negative type metric embeds into l_1 with constant distortion.
(*d* is a negative type metric if $\sqrt{d}$ is isometrically embeddable in l_2.)
- Insights from UGC have helped constructing a negative metric that embeds in l_1 with distortion at least $log(log(n))$ [KV05]
$\rightarrow$ The conjecture is false

Metric Embedding: Goemans-Linial Conjecture

Goemans-Linial Conjecture is true $\rightarrow$ $O(1)$-approximation for a graph partitioning problem $\mathcal{P}$ (sparsest cut) with some SDP relaxation $\mathcal{S}$.

Metric Embedding: Goemans-Linial Conjecture

Goemans-Linial Conjecture is true $\rightarrow$ $O(1)$-approximation for a graph partitioning problem $\mathcal{P}$ (sparsest cut) with some SDP relaxation $\mathcal{S}$.

- Reduction from ULC to $\mathcal{P}$ to prove inapproximability for $\mathcal{P}$ under UGC

Metric Embedding: Goemans-Linial Conjecture

Goemans-Linial Conjecture is true $\rightarrow$ $O(1)$-approximation for a graph partitioning problem $\mathcal{P}$ (sparsest cut) with some SDP relaxation $\mathcal{S}$.

- Reduction from ULC to $\mathcal{P}$ to prove inapproximability for $\mathcal{P}$ under UGC
- Results on approximation ULC with some SDP relaxation

Metric Embedding: Goemans-Linial Conjecture

Goemans-Linial Conjecture is true $\rightarrow$ $O(1)$-approximation for a graph partitioning problem $\mathcal{P}$ (sparsest cut) with some SDP relaxation $\mathcal{S}$.

- Reduction from ULC to $\mathcal{P}$ to prove inapproximability for $\mathcal{P}$ under UGC
- Results on approximation ULC with some SDP relaxation

$\Rightarrow$ The ratio of the approximation of $\mathcal{P}$ with $\mathcal{S}$ is not constant

Metric Embedding: Goemans-Linial Conjecture

Goemans-Linial Conjecture is true $\rightarrow$ $O(1)$-approximation for a graph partitioning problem $\mathcal{P}$ (sparsest cut) with some SDP relaxation $\mathcal{S}$.

- Reduction from ULC to $\mathcal{P}$ to prove inapproximability for $\mathcal{P}$ under UGC
- Results on approximation ULC with some SDP relaxation

$\Rightarrow$ The ratio of the approximation of $\mathcal{P}$ with $\mathcal{S}$ is not constant
$\Rightarrow$ The Goemans-Linial Conjecture is false

Some UGC inapproximability results [Kho10]

Problem	Best Approx. Known	Best In-approx known	Best Inap-prox. known under UGC
Vertex Cover	2	1.36	$2 - \epsilon$
MaxCut	0.878	$\frac{16}{17} + \epsilon$	$0.878 + \epsilon$
Max Acyclic Sub-graph	0.5	$\frac{65}{66} + \epsilon$	$0.5 + \epsilon$
Any CSP $\mathcal{C}$ with integrality gap $\alpha_{\mathcal{C}}$	$\alpha_{\mathcal{C}}$		$\alpha_{\mathcal{C}} + \epsilon$

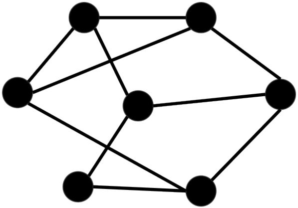
Max-Cut: definition

Input:

- A graph $G = (V, E)$

Output:

-



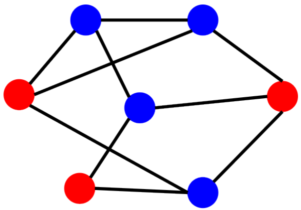
Max-Cut: definition

Input:

- A graph $G = (V, E)$

Output:

- A partition (V_1, V_2) which maximizes the size of the set $(V_1, V_2) \cap E$



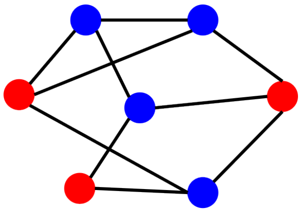
Max-Cut: definition

Input:

- A graph $G = (V, E)$

Output:

- A partition (V_1, V_2) which maximizes the size of the set $(V_1, V_2) \cap E$



MaxCut is NP-hard and hard to approximate within $\frac{16}{17}$ [Hås01].

MaxCut: Goemans Williamson algorithm [GW95]

Quadratic Program

$$\text{max:} \quad \sum_{(i,j)\in E} \frac{1 - x_i x_j}{2}$$

$$\text{s.t.:} \quad x_i^2 = 1 \quad \forall i \in V$$

SDP relaxation

$$\text{max:} \quad \sum_{(i,j)\in E} \frac{1 - v_i . v_j}{2}$$

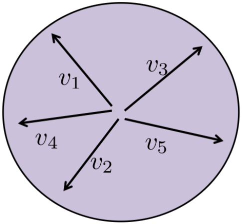
$$\text{s.t:} \quad v_i . v_i = 1 \quad \forall i \in V$$

$$v_i \in \mathbb{R}^n \quad \forall i \in V$$

MaxCut: Goemans Williamson algorithm[GW95]

The algorithm

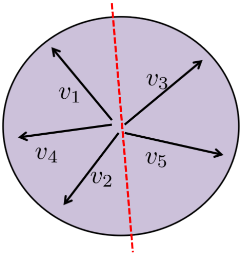
- Solve the relaxed SDP
-
-



MaxCut: Goemans Williamson algorithm[GW95]

The algorithm

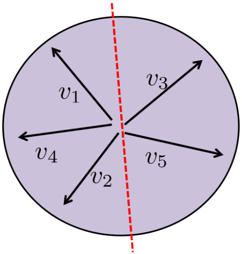
- Solve the relaxed SDP
- Cut the sphere with a random hyperplane
-



MaxCut: Goemans Williamson algorithm[GW95]

The algorithm

- Solve the relaxed SDP
- Cut the sphere with a random hyperplane
- The cut $\rightarrow$ the two sets of vectors cut by the hyperplane

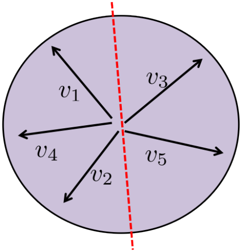


$\{1, 4, 2\} \cup \{3, 5\}$

MaxCut: Goemans Williamson algorithm[GW95]

The algorithm

- Solve the relaxed SDP
- Cut the sphere with a random hyperplane
- The cut $\rightarrow$ the two sets of vectors cut by the hyperplane

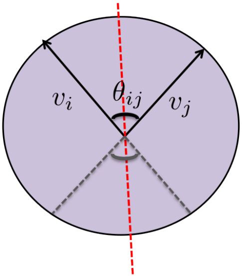


$\{1, 4, 2\} \cup \{3, 5\}$

What is the ratio achieved by the algorithm?

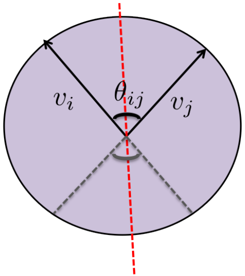
MaxCut: ratio of the Goemans Williamson algorithm[GW95]

- OPT:= value of the MaxCut
- OPT_{SDP}:= value obtained by the SDP relaxation
- $\mathbb{E}(C)$:= Expectation of the value of the cut obtained by the algorithm



MaxCut: ratio of the Goemans Williamson algorithm[GW95]

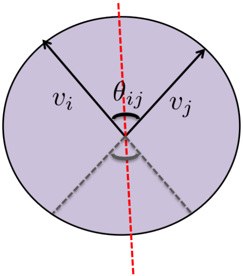
- $OPT_{SDP} = \sum_{(i,j)\in E} \frac{1-v_i.v_j}{2}$
-



MaxCut: ratio of the Goemans Williamson algorithm[GW95]

- $OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - v_i.v_j}{2}$

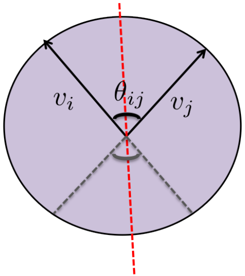
  $OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - \cos(\theta_{ij})}{2}$
-



MaxCut: ratio of the Goemans Williamson algorithm[GW95]

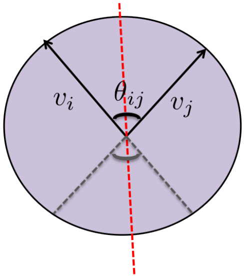
- $OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - v_i . v_j}{2}$

  $OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - \cos(\theta_{ij})}{2}$
- $\mathbb{E}(C) = \sum_{(i,j)\in E} Pr(v_i, v_j \text{ separated})$



MaxCut: ratio of the Goemans Williamson algorithm[GW95]

- $$OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - v_i . v_j}{2}$$
  $$OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - \cos(\theta_{ij})}{2}$$
- $$\mathbb{E}(C) = \sum_{(i,j)\in E} Pr(v_i, v_j \text{ separated})$$
  $$\mathbb{E}(C) = \sum_{(i,j)\in E} \frac{\theta_{ij}}{\pi}$$



MaxCut: ratio of the Goemans Williamson algorithm[GW95]

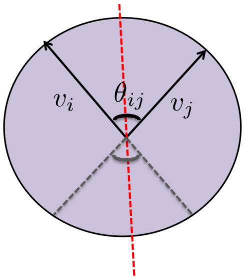
- $OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - v_i . v_j}{2}$

  $OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - \cos(\theta_{ij})}{2}$

- $\mathbb{E}(C) = \sum_{(i,j)\in E} Pr(v_i, v_j \text{ separated})$

  $\mathbb{E}(C) = \sum_{(i,j)\in E} \frac{\theta_{ij}}{\pi}$

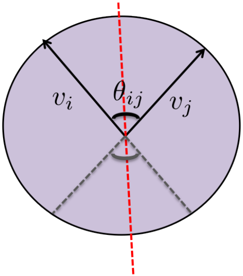
  $\mathbb{E}(C) = \sum_{(i,j)\in E} \frac{\theta_{ij}}{\pi} \frac{2}{1 - \cos(\theta_{ij})} \frac{1 - \cos(\theta_{ij})}{2}$



MaxCut: ratio of the Goemans Williamson algorithm[GW95]

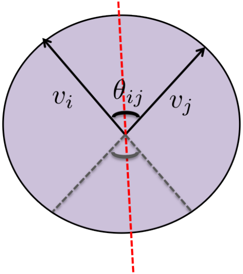
- $$OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - v_i.v_j}{2}$$
  $$OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - \cos(\theta_{ij})}{2}$$
- $$\mathbb{E}(C) = \sum_{(i,j)\in E} Pr(v_i, v_j \text{ separated})$$
  $$\mathbb{E}(C) = \sum_{(i,j)\in E} \frac{\theta_{ij}}{\pi}$$
  $$\mathbb{E}(C) = \sum_{(i,j)\in E} \frac{\theta_{ij}}{\pi} \frac{2}{1 - \cos(\theta_{ij})} \frac{1 - \cos(\theta_{ij})}{2}$$

Let $\alpha_{GW} = \min_{0\leq\theta\leq\pi} \frac{2}{\pi} \frac{\theta_{ij}}{1 - \cos(\theta_{ij})} \simeq 0.878$



MaxCut: ratio of the Goemans Williamson algorithm[GW95]

- $$OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - v_i . v_j}{2}$$
  $$OPT_{SDP} = \sum_{(i,j)\in E} \frac{1 - \cos(\theta_{ij})}{2}$$
- $$\mathbb{E}(C) = \sum_{(i,j)\in E} Pr(v_i, v_j \text{ separated})$$
  $$\mathbb{E}(C) = \sum_{(i,j)\in E} \frac{\theta_{ij}}{\pi}$$
  $$\mathbb{E}(C) = \sum_{(i,j)\in E} \frac{\theta_{ij}}{\pi} \frac{2}{1-\cos(\theta_{ij})} \frac{1-\cos(\theta_{ij})}{2}$$



Let $\alpha_{GW} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta_{ij}}{1-\cos(\theta_{ij})} \simeq 0.878$ then

$$\mathbb{E}(C) \geq \alpha_{GW} OPT_{SDP} \geq \alpha_{GW} OPT$$

From Unique Label Cover to Max-Cut [KKMO04]

$ULC(\delta) \rightarrow$ distinguishing between the cases:

- $OPT \geq 1 - \delta$
- $OPT \leq \delta$

From Unique Label Cover to Max-Cut [KKMO04]

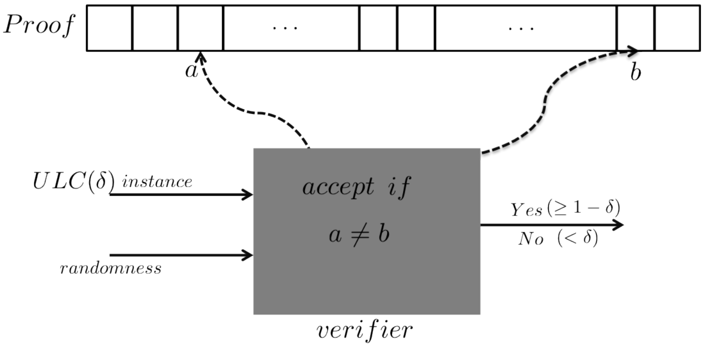
$ULC(\delta) \rightarrow$ distinguishing between the cases:

- $OPT \geq 1 - \delta$
- $OPT \leq \delta$

Theorem

For every $\epsilon > 0$ there exists δ such that there is a PCP for $ULC(\delta)$ in which the verifier reads two bits from the proof and accepts iff they are unequal, and which has completeness c and soundness s such that $\frac{s}{c} = \alpha_{GW} + \epsilon$

2-bit PCP for ULC(δ)



- Completeness: If $OPT(ULC) \geq 1-\delta$, then there is a proof that the verifier accepts with probability $\geq c$.
- Soundness: If $OPT(ULC) \leq \delta$, then all proofs are accepted with probability $\leq s$.

Corollary

Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

Corollary

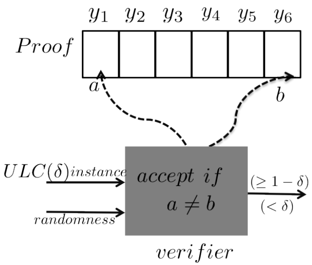
Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

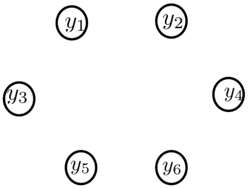
For $\epsilon > 0$, let $ULC(\delta)$ be an instance with a 2-bit PCP.

Corollary

Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

For $\epsilon > 0$, let $ULC(\delta)$ be an instance with a 2-bit PCP.

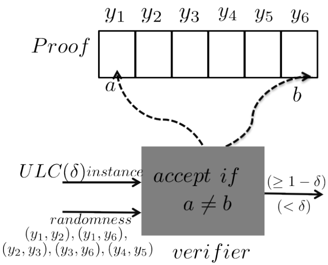


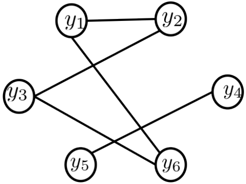


Corollary

Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

For $\epsilon > 0$, let $ULC(\delta)$ be an instance with a 2-bit PCP.

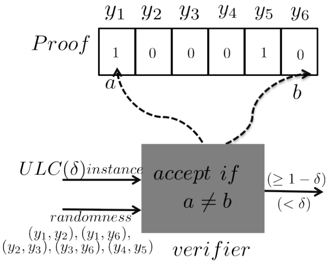


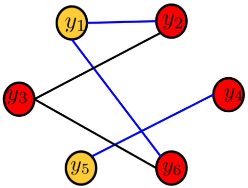


Corollary

Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

For $\epsilon > 0$, let $ULC(\delta)$ be an instance with a 2-bit PCP.



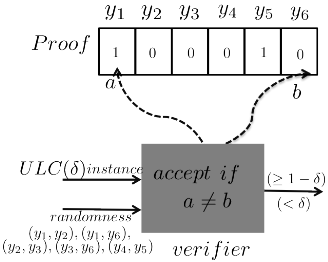


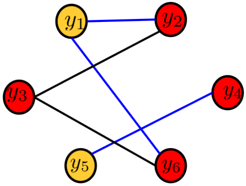
Probability of passing the test of a proof = fraction of edges of the corresponding cut

Corollary

Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

For $\epsilon > 0$, let $ULC(\delta)$ be an instance with a 2-bit PCP.





There is a proof that the verifier accepts with probability $\geq c \rightarrow$ there is a cut with value $\geq c|E|$

All proofs are accepted with probability $\leq s \rightarrow$ all cuts have value $\leq s|E|$

Corollary

Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

- $OPT(ULC) \geq 1 - \delta \Rightarrow MaxCut \geq c|E|$
- $OPT(ULC) \leq \delta \Rightarrow MaxCut \leq s|E|$

Corollary

Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

- $OPT(ULC) \geq 1 - \delta \Rightarrow MaxCut \geq c|E|$
- $OPT(ULC) \leq \delta \Rightarrow MaxCut \leq s|E|$

$$UGC \Rightarrow MaxCut(s, c) \text{ is NP-hard}$$

Corollary

Assuming UGC, MaxCut is hard to approximate within $\alpha_{GW} + \epsilon$.

- $OPT(ULC) \geq 1 - \delta \Rightarrow MaxCut \geq c|E|$
- $OPT(ULC) \leq \delta \Rightarrow MaxCut \leq s|E|$

$$UGC \Rightarrow MaxCut(s, c) \text{ is NP-hard}$$

$$\Rightarrow \text{MaxCut is hard to approximate within } \frac{s}{c} = \alpha_{GW} + \epsilon$$

How to build a 2-bit PCP for ULC?

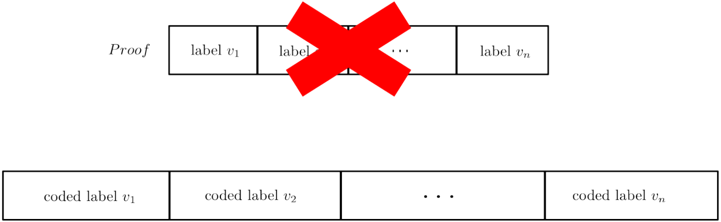
Proof	label v_1	label v_2	$\cdots$	label v_n

How to build a 2-bit PCP for ULC?

Proof	label v_1	label	$\cdots$	label v_n

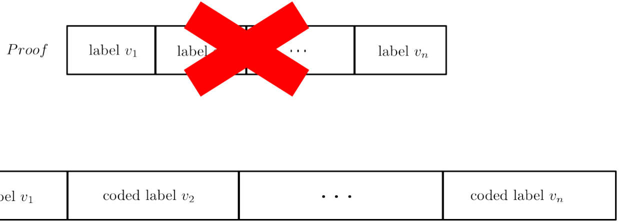
coded label v_1	coded label v_2	$\cdots$	coded label v_n

How to build a 2-bit PCP for ULC?



- The verifier expects codewords of the labels

How to build a 2-bit PCP for ULC?



- The verifier expects codewords of the labels
- The verifier has to have completeness c ans soundness s

How to build a 2-bit PCP for ULC?

Completeness: $OPT(ULC) \geq 1 - \delta$, then there is a proof that the verifier accepts with probability at least c

How to build a 2-bit PCP for ULC?

Completeness: $OPT(ULC) \geq 1 - \delta$, then there is a proof that the verifier accepts with probability at least c
$\rightarrow$ Code a good labeling into a proof that can be accepted with good propability (c)

How to build a 2-bit PCP for ULC?

Completeness: $OPT(ULC) \geq 1 - \delta$, then there is a proof that the verifier accepts with probability at least c

$\rightarrow$ Code a good labeling into a proof that can be accepted with good propability (c)

$\rightarrow$ Use "Long code" to code labels

Definition

The long code of label $i \in [1, n]$ is the truth table of the function $f : \{0,1\}^n \rightarrow \{0,1\}$ such that $f(x) = x_i$.

How to build a 2-bit PCP for ULC?

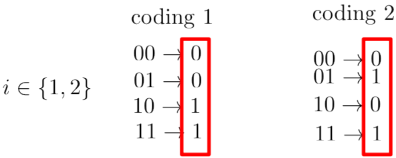
Completeness: $OPT(ULC) \geq 1 - \delta$, then there is a proof that the verifier accepts with probability at least c

$\rightarrow$ Code a good labeling into a proof that can be accepted with good propability (c)

$\rightarrow$ Use "Long code" to code labels

Definition

The long code of label $i \in [1, n]$ is the truth table of the function $f : \{0,1\}^n \rightarrow \{0,1\}$ such that $f(x) = x_i$.



Soundness: If $OPT(ULC) \leq \delta$, then all proofs are accepted with probability at most s.

Soundness: If $OPT(ULC) \leq \delta$, then all proofs are accepted with probability at most s.

$\rightarrow$ If a proof is accepted with probability $\geq s$ then $OPT(ULC) > \delta$

Soundness: If $OPT(ULC) \leq \delta$, then all proofs are accepted with probability at most s.
$\rightarrow$ If a proof is accepted with probability $\geq s$ then $OPT(ULC) > \delta$
$\rightarrow$ Decode a proof into labels

Soundness: If $OPT(ULC) \leq \delta$, then all proofs are accepted with probability at most s.
$\rightarrow$ If a proof is accepted with probability $\geq s$ then $OPT(ULC) > \delta$
$\rightarrow$ Decode a proof into labels
$\rightarrow$ Distinguish dictatorships from functions far from being dictatorships

Soundness: If $OPT(ULC) \leq \delta$, then all proofs are accepted with probability at most s.

$\rightarrow$ If a proof is accepted with probability $\geq s$ then $OPT(ULC) > \delta$

$\rightarrow$ Decode a proof into labels

$\rightarrow$ Distinguish dictatorships from functions far from being dictatorships

- A dictatorship depending on coordinate i can be decoded into label i.
- Functions far from dictatorships cannot be decoded

How to build a 2-bit PCP for ULC?

The PCP with only two bits can be designed thanks to:

- Unique games (permutations)
- Majority is stablest

UGC and SDP

- Under UGC, the SDP-based algorithm provides the best approximation for MaxCut.
-

UGC and SDP

- Under UGC, the SDP-based algorithm provides the best approximation for MaxCut.
- A stronger result [Rag08]: UGC $\rightarrow$ for every MAX-CSP, the simplest SDP relaxation is the best possible poly-time approximation.

UGC and SDP [Rag08]

- For every MAX-CSP there is a semi-definite programming relaxation $\mathcal{S}$
- Assuming UGC, no other polynomial time algorithm can provide a better approximation than $\mathcal{S}$

Plan

UGC: True or False?

True?

- Validates the quality of SDP relaxations

False?

UGC: True or False?

True?

- Validates the quality of SDP relaxations
- It provides very "neat" inapproximability results

False?

- The results can still hold even if the conjecture is false

UGC: True or False?

True?

- Validates the quality of SDP relaxations
- It provides very "neat" inapproximability results
- There is no algorithm to refute it

False?

- The results can still hold even if the conjecture is false
- A sub-exponential time algorithm has been designed [ABS10]

UGC: True or False?

True?

- Validates the quality of SDP relaxations
- It provides very "neat" inapproximability results
- There is no algorithm to refute it
- $GapULC_{C(\delta)\delta,\delta}$ is NP-hard [FR04]

False?

- The results can still hold even if the conjecture is false
- A sub-exponential time algorithm has been designed [ABS10]
- $C(\delta)\delta \to 0$ as $\delta \to 0$

Thank you

Sanjeev Arora, Boaz Barak, and David Steurer.
Subexponential algorithms for unique games and related problems.
In *In 51 st IEEE FOCS*, 2010.

Sanjeev Arora and Shmuel Safra.
Probabilistic checking of proofs: A new characterization of np.
J. ACM, 45(1):70–122, January 1998.

Uriel Feige and Daniel Reichman.
On systems of linear equations with two variables per equation.
In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 3122 of *Lecture Notes in Computer Science*, pages 117–127. Springer Berlin Heidelberg, 2004.

Michel X. Goemans and David P. Williamson.

Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.
J. ACM, 42(6):1115–1145, November 1995.

Johan Håstad.
Some optimal inapproximability results.
J. ACM, 48(4):798–859, July 2001.

Subhash Khot.
On the power of unique 2-prover 1-round games.
In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 767–775, New York, NY, USA, 2002. ACM.

Subhash Khot.
On the unique games conjecture.
Technical report, Courant Institute of Mathematical Sciences, NYU, 2010.

S. Khot, G. Kindler, E. Mossel, and R. O'Donnell.
Optimal inapproximability results for max-cut and other 2-variable csps?
In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 146–154, Oct 2004.

Subhash A. Khot and Nisheeth K. Vishnoi.
The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into l 1.
In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 53–62, Washington, DC, USA, 2005. IEEE Computer Society.

Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz.

Noise stability of functions with low influences: Invariance and optimality.

In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 21–30, Washington, DC, USA, 2005. IEEE Computer Society.

Limits of approximation algorithms: Pcps and unique games (dimacs tutorial lecture notes).

Prasad Raghavendra.
Optimal algorithms and inapproximability results for every csp.
In *In Proc. 40 th ACM STOC*, pages 245–254, 2008.

Ran Raz.
A parallel repetition theorem.
SIAM Journal on Computing, 27(3):763–803, 1998.

Luca Trevisan.
On khots unique games conjecture, 2011.