CS 598: Spectral Graph Theory. Lecture 7

The Unique Games Conjecture and SDP Duality.

Alexandra Kolla
Today

• The Unique Games Conjecture
• Unique Games and Graphs
• SDP for Unique Games
• Duality proof that Random Unique Games are easy.
The MAX CUT Problem

- **Input**: $G = (V, E)$
The MAX CUT Problem

- **Input:** $G = (V,E)$
- **Objective:** Partition $G$ in $(S,S')$ as to **MAXIMIZE** number of edges cut

- **[Karp ‘72]:** MAX CUT is NP-complete
- What about approximating MAX CUT?
The MAX CUT Problem

• **Input:** G = (V,E)

• **Objective:** Partition G in (S,S') as to MAXIMIZE number of edges cut

Approximation algorithms:

• Random cut (trivial): half of optimal

• [GW’94]: $\alpha_{GW} = 0.878$ approximation algorithm of MAX

How many of you bet this is best we can do?
The MAX CUT Problem

• **Input:** \( G = (V, E) \)
• **Objective:** Partition \( G \) in \( (S, S') \) as to MAXIMIZE number of edges cut

**Approximation algorithms:**

• **Random cut (trivial):** half of optimal
• **[GW’94]:** \( \alpha_{GW} = 0.878 \) approximation algorithm of MAX

If Unique Games Conjecture true, then it is!
Can We Hope for Better Approximation Algorithms in P?

Previous inapproximability not a coincidence! Unique Games Conjecture (UGC) captures **exact** inapproximability of many more problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Best Approximation Algorithm Known</th>
<th>UGC-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxCut</td>
<td>0.878[GW94]</td>
<td>0.878 [KKMO07]</td>
</tr>
<tr>
<td>Vertex Cover</td>
<td>2</td>
<td>2-(\varepsilon) [KR06]</td>
</tr>
<tr>
<td>Max (k)-CSP</td>
<td>(\Omega(k/2^k))[CMM07]</td>
<td>(O(k/2^k))[ST,AM,GR]</td>
</tr>
</tbody>
</table>
What are Unique Games?

1. Unique Games are popular not only among computer scientist!
What are Unique Games?

1. Unique Games are popular not only among computer scientists!

2. We can purchase Unique Games online!
What are Unique Games?

1. Unique Games are popular not only among computer scientists!

2. We can purchase Unique Games on-line!

3. Unique Games are related to the Unique Games Conjecture...
Unique Games = Unique Label Cover Problem

Given: set of constraints

Linear Equations mod $k$:

$$x_i - x_j = c_{ij} \mod k$$

**GOAL**

$k = \text{“alphabet” size}

Find labeling that satisfies **maximum** number of constraints.

**EXAMPLE**

- $x_1 - x_2 = 0 \pmod{3}$
- $x_2 - x_3 = 0 \pmod{3}$
- $x_1 - x_3 = 1 \pmod{3}$

The constraint graph

$$x_1 - x_3 = 1 \pmod{3}$$

$$x_1 - x_2 = 0 \pmod{3}$$

$$x_2 - x_3 = 0 \pmod{3}$$
Unique Games, an Example

Given: set of constraints

Linear Equations mod $k$:

$\begin{align*}
  x_i - x_j &= c_{ij} \mod k
\end{align*}$

**GOAL**

$k =$ “alphabet” size

Find labeling that satisfies **maximum** number of constraints.

**EXAMPLE**

$x_1 - x_2 = 0 \pmod 3$ \hspace{1cm} \checkmark

$x_2 - x_3 = 0 \pmod 3$ \hspace{1cm} \checkmark

$x_1 - x_3 = 1 \pmod 3$ \hspace{1cm} \times

The constraint graph

Satisfy 2/3 constraints
Unique Games, an Example

Given: set of constraints

Linear Equations mod $k$:

$$x_i - x_j = c_{ij} \mod k$$

**GOAL**

Find labeling that satisfies **maximum** number of constraints.

**EXAMPLE**

$$x_1 - x_2 = 0 \mod 3$$  
$$x_2 - x_3 = 0 \mod 3$$  
$$x_1 - x_3 = 1 \mod 3$$

The constraint graph

$k =$ “alphabet” size

Rest of the talk: $d$-regular graphs
Unique Games Conjecture

• [Khot’02] For every positive $\varepsilon$ and $\delta$ there is a large enough $k$ s.t. for some instance of Unique Games with alphabet size $k$ and $\text{OPT} > 1 - \varepsilon$, it is NP hard to satisfy a $\delta$ fraction of all constraints.

• Implies: many known algorithms are optimal: MAX CUT, Vertex Cover, $k$-CSP, ...

• [Raghavendra’08] SDP is the best method for any CSP problem assuming UGC!

Is Unique Games Conjecture True?
Unique Games Conjecture

• [Khot’02] For every positive $\varepsilon$ and $\delta$ there is a large enough $k$ s.t. for some instance of Unique Games with alphabet size $k$ and $\text{OPT} > 1 - \varepsilon$, it is NP hard to satisfy a $\delta$ fraction of all constraints.

• Implies: many known algorithms are optimal: MAX CUT, Vertex Cover, k-CSP,…

• [Raghavendra’08] SDP is the best method for any CSP problem assuming UGC!

Really embarassing to not know, since we can solve systems of linear equations!
## Summary: Algorithmic Results for UG

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>On 1-(\varepsilon) instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Khot</td>
<td>(1-O(k^2 \varepsilon^{1/5} \sqrt{\log(1/\varepsilon)}))</td>
</tr>
<tr>
<td>Trevisan</td>
<td>(1-O(3\sqrt{\varepsilon \log n}))</td>
</tr>
<tr>
<td>Gupta-Talwar</td>
<td>(1-O(\varepsilon \log n))</td>
</tr>
<tr>
<td>CMM1</td>
<td>(k^{-\varepsilon/2})</td>
</tr>
<tr>
<td>CMM2</td>
<td>(1-O(\varepsilon \sqrt{\log n / \log k}))</td>
</tr>
</tbody>
</table>

### General Graphs
- **Expander**
  - AKKTSV’08, KT’08, MM’10
    - Constant, depends on conductance
- **Local expander**
  - AIMS’09, SR’09
    - Constant, depends on local expansion

SDP/LP based
- Almost all above approaches were LP or SDP based

Tight for SDP, there is a counterexample
## Summary: Algorithmic Results for UG

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>On 1-(\varepsilon) instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Khot</td>
<td>1-O(k^2 \varepsilon^{1/5} \log(1/\varepsilon))</td>
</tr>
<tr>
<td>Trevisan</td>
<td>1-O(3\sqrt{\varepsilon \log n})</td>
</tr>
<tr>
<td>Gupta-Talwar</td>
<td>1-O((\varepsilon \log n))</td>
</tr>
<tr>
<td>CMM1</td>
<td>(k^{-\varepsilon/2})</td>
</tr>
<tr>
<td>CMM2</td>
<td>1-O((\varepsilon \sqrt{\log n} \sqrt{\log k}))</td>
</tr>
</tbody>
</table>

### General Graphs
- **Expander**
  - AKKTSV’08, KT’08, MM’10: Constant, depends on conductance
  - AIMS’09, SR’09: Constant, depends on local expansion

### Special Graphs
- **Local expander**
- **Few large eigenvalues**

- **SDP/LP based**
  - Tight for SDP, there is counterexample
  - Purely SPECTRAL Approach “beats” SDP
### Summary: Algorithmic Results for UG

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>On 1-(\varepsilon) instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Khot</td>
<td>1-(O(k^2 \varepsilon^{1/5} \sqrt{\log(1/\varepsilon)}))</td>
</tr>
<tr>
<td>Trevisan</td>
<td>1-(O(3\sqrt{\varepsilon \log n}))</td>
</tr>
<tr>
<td>Gupta-Talwar</td>
<td>1-(O(\varepsilon \log n))</td>
</tr>
<tr>
<td>CMM1</td>
<td>(k^{-\varepsilon/2})</td>
</tr>
<tr>
<td>CMM2</td>
<td>1-(O(\varepsilon \log n \sqrt{\log k}))</td>
</tr>
</tbody>
</table>

**General Graphs**

- **Expander**
  - AKKTSV’08, KT’08, MM’10: Constant, depends on conductance

**Special Graphs**

- **Local expander**
  - AIMS’09, SR’09: Constant, depends on local expansion

- **Few large eigenvalues**
  - K’10: Quality and running time depends on eigenspace

**ABS’10**: Subexponential time algorithm for ANY instance
### Summary: Algorithmic Results for UG

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>On 1-(\varepsilon) instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Khot</td>
<td>1-(O(k^2 \varepsilon^{1/5} \sqrt{\log(1/\varepsilon)}))</td>
</tr>
<tr>
<td>Trevisan</td>
<td>1-(O(3\sqrt{\varepsilon \log n}))</td>
</tr>
<tr>
<td>Gupta-Talwar</td>
<td>1-(O(\varepsilon \log n))</td>
</tr>
<tr>
<td>CMM1</td>
<td>(k^{-\varepsilon/2})</td>
</tr>
<tr>
<td>CMM2</td>
<td>1-(O(\varepsilon \sqrt{\log(n \sqrt{\log k})}))</td>
</tr>
</tbody>
</table>

**General Graphs**

- Expander
  - AKKTSV’08
  - KT’08, MM’10
  - AKKTSV’08
  - KT’08, MM’10
  - Constant, depends on conductance

**Special Graphs**

- Local expander
  - AIMS’09, SR’09
  - Constant, depends on local expansion

- Few large eigenvalues
  - K’10
  - Quality and running time depends on eigenspace

## Key Ingredient

ABS’10: Subexponential time algorithm for ANY instance
## Summary: Algorithmic Results for UG

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>On $1-\epsilon$ instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Khot</td>
<td>$1-O(k^2 \epsilon^{1/5} \sqrt{\log(1/\epsilon)})$</td>
</tr>
<tr>
<td>Trevisan</td>
<td>$1-O(3\sqrt{\epsilon \log n})$</td>
</tr>
<tr>
<td>Gupta-Talwar</td>
<td>$1-O(\epsilon \log n)$</td>
</tr>
<tr>
<td>CMM1</td>
<td>$k^{-\epsilon/2-\epsilon}$</td>
</tr>
<tr>
<td>CMM2</td>
<td>$1-O(\epsilon \sqrt{\log n} \sqrt{\log k})$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General Graphs</th>
<th>Expander</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKKTSV’08</td>
<td>Constant, depends on conductance</td>
</tr>
<tr>
<td>KT’08,MM’10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Special Graphs</th>
<th>Local expander</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIMS’09, SR’09</td>
<td>Constant, depends on local expansion</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Few large eigenvalues</th>
<th>K’10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quality and running time depends on eigenspace</td>
</tr>
</tbody>
</table>

KMM’10: Semi-Random instances are easy
Unique Games and Graphs

2. The “label-extended” graph

1. The “constraint graph”

- Replace each vertex with $k$ vertices - one for each label

$x_1 - x_2 = 0 \pmod{3}$
$x_2 - x_3 = 0 \pmod{3}$
$x_1 - x_3 = 1 \pmod{3}$
$x_1 - x_2 = 0 \pmod{3}$
$x_2 - x_3 = 0 \pmod{3}$
$x_1 - x_3 = 1 \pmod{3}$
$x_2 - x_3 = 0 \pmod{3}$
Unique Games and Graphs

1. The “constraint graph”

0 1 2

x

x

x

0 1 2

x

x

x

x

2. The “label-extended” graph

• Replace each vertex with k vertices— one for each label

• Replace each edge with the “permutation matching”
Unique Games and Graphs

2. The “label-extended” graph

1. The “constraint graph”

- Replace each edge with the “permutation matching”
- Replace each vertex with \( k \) vertices - one for each label
Unique Games and Graphs

2. The “label-extended” graph

- Replace each vertex with \( k \) vertices - one for each label

1. The “constraint graph”

- Replace each edge with the “permutation matching”

\[
\begin{align*}
x_1 - x_3 &= 1 \pmod{3} \\
x_1 - x_2 &= 0 \pmod{3} \\
x_2 - x_3 &= 0 \pmod{3}
\end{align*}
\]
M has each non-zero entry replaced by a block corresponding to the permutation on edge.

**GRAPH THEORY?**

It’s a graph, it has adjacency matrix!
“Old eigenvalues” of original graph are still eigenvalues. What other eigenvalues are there?
From Labelings to Spectra

Set $S$ that contains **exactly one** “small” node from each node group = labeling
From Labelings to Spectra

• Set $S$ that contains exactly one “small” node from each node group = labeling

• Corresponds to a “characteristic vector” (cut vector)

$$\chi_{(0,0,0)} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$
Perfect Satisfiability and Largest Eigenvalues

Let’s look at a perfectly satisfiable game for intuition...

- Corresponds to a perfect labeling, cut that cuts no edges
- Is eigenvector of $M$ with eigenvalue $d$
Perfect Satisfiability and Largest Eigenvalues

Let's look at a perfectly satisfiable game for intuition...

• Corresponds to a perfect labeling, cut that cuts no edges

• Is eigenvector of \( L \) with eigenvalue 0

\[ L = dI - M \]
On the Opposite Side: Very Unsatisfiable Instances

We similarly expect that very unsatisfiable Instances on “well-connected” graphs will have second Adjacency matrix eigenvalue is very small (far from d), or the second Laplacian eigenvalue is very large, far from 0.

“Well-connected” for us will mean that, if graph is d-regular, then the second Adjacency Matrix eigenvalue is $\sim O(\sqrt{d})$ and the second Laplacian eigenvalue is $\sim d - \Omega(\sqrt{d})$.
SDP Relaxation

- Introduce an “indicator vector” $u_i$ for each vertex $u$ and color $i$
  
  $u_i = \begin{cases} 
  e, & \text{if } u \text{ is colored with } i \\
  0, & \text{otherwise}
  \end{cases}$

- Number of unsatisfied constraints equals

$$\sum_{uv \in E} \sum_{i} \left\| u_i - v_{\pi_{uv}(i)} \right\|^2$$
SDP Relaxation

- Introduce an “indicator vector” \( u_i \) for each vertex \( u \) and color \( i \).

  - Minimize:
    \[
    \sum_{uv \in E} \sum_i \left\| u_i - v_{\pi_{uv}(i)} \right\|^2
    \]
    Subject to: \( \|u_1\|^2 + \ldots + \|u_k\|^2 = 1 \) for all \( u \)
    \[
    \langle u_i | u_i \rangle = 0, \text{ for all } u, i \neq j
    \]

- We next use dual to bound the primal (blackboard). Led to UG are easy on expanders