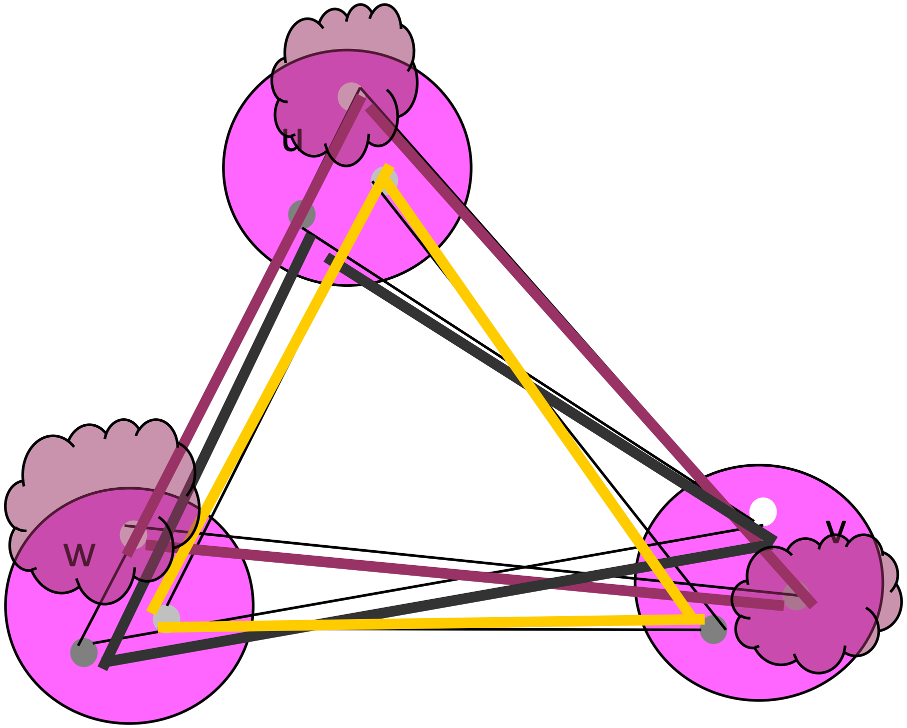
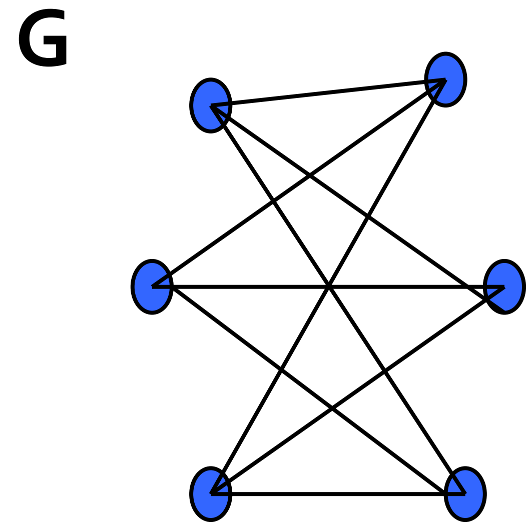
Towards Refuting UGC



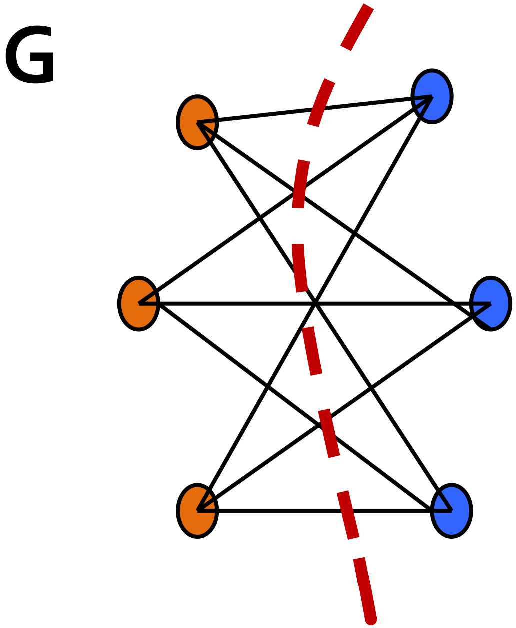
The MAX CUT Problem

- **Input:** G = (V,E)



The MAX CUT Problem

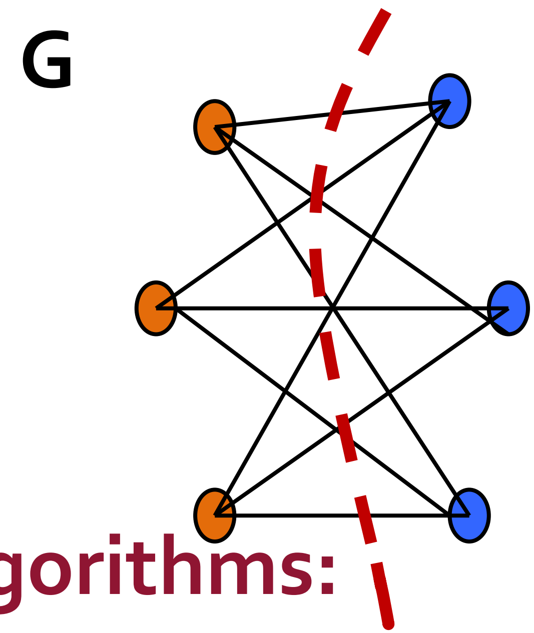
- **Input:** G = (V,E)
- **Objective :** Partition G in (S,S') as to MAXIMIZE number of edges cut



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

The MAX CUT Problem

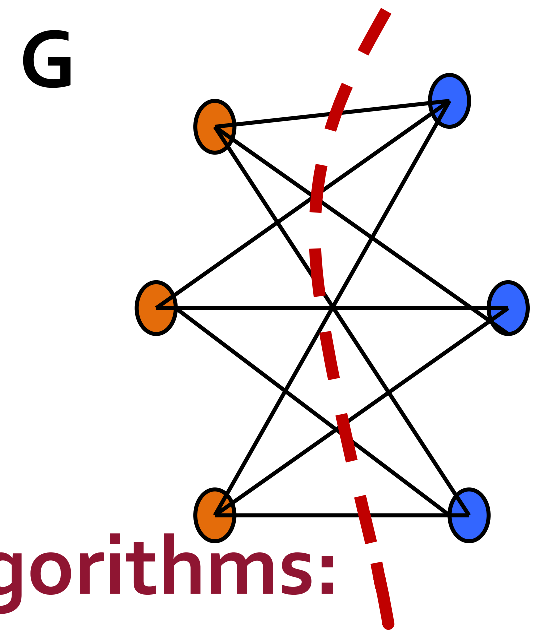
- **Input:** G = (V,E)
- **Objective :** Partition G in (S,S') as to MAXIMIZE number of edges cut



Approximation algorithms:

- **Random cut (trivial):** half of optimal
- **[GW'94]:** α_{GW}=0.878 approximation algorithm of M…

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]:** α_{GW}=0.878 approximation algorithm of M

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

Problem	Best Approximation Algorithm Known	UGC-Hardness
MaxCut	0.878[GW94]	0.878 [KKMO07]
Vertex Cover	2	2-ε [KR06]
Max k-CSP	$\Omega(k/2^k)$[CMM07]	$O(k/2^k)$[ST,AM,GR]

Plan for Today

1. Unique Games Conjecture(UGC)
2. Spectra of Graphs
3. Towards Refuting UGC on almost-all Graphs
4. Open Questions

Plan for Today

1**. Unique Games Conjecture(UGC)**

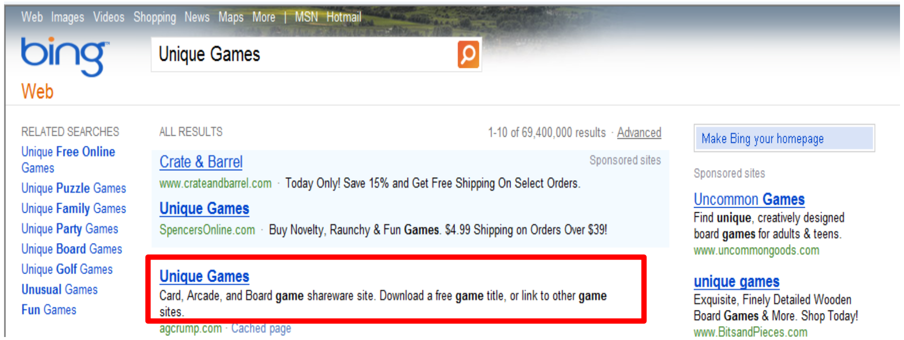
2. Spectra of Graphs

3. Towards Refuting UGC on almost-all Graphs

4. Open Questions

What are Unique Games?

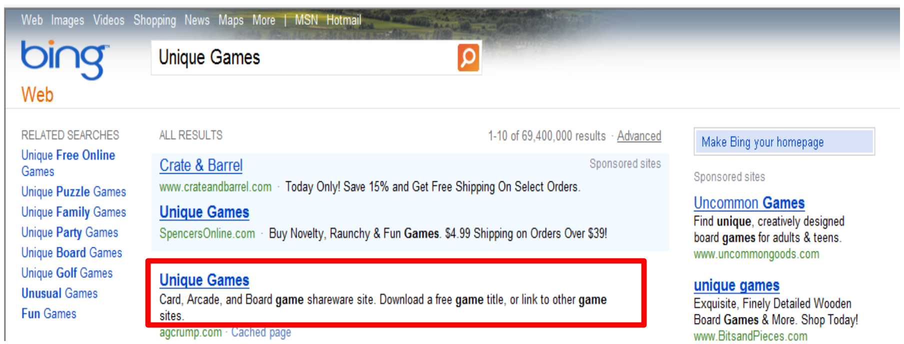
1. Unique Games are popular not only among computer scientist!

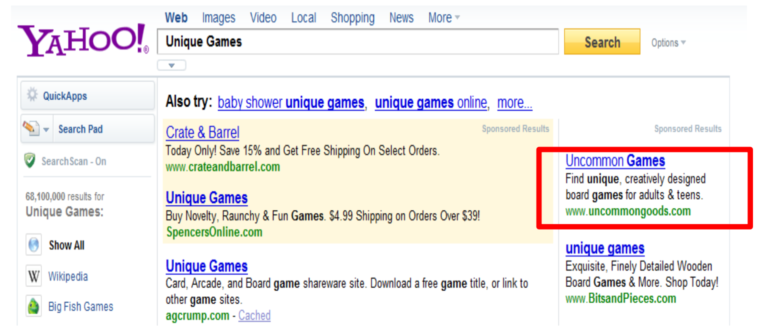


bing: 70 million pages

What are Unique Games?

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





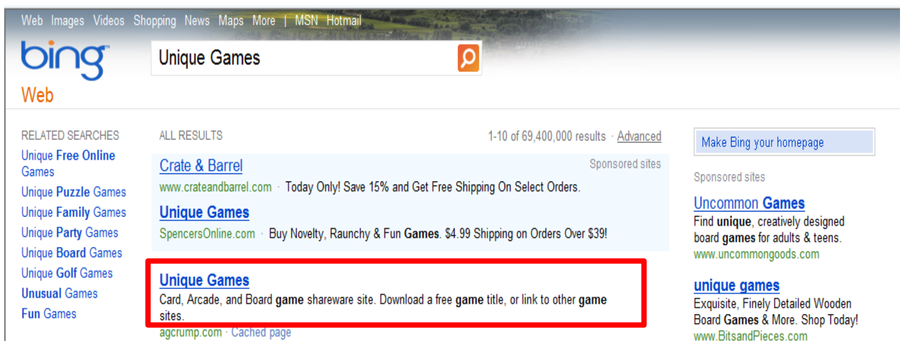
Yahoo!: 69 million pages

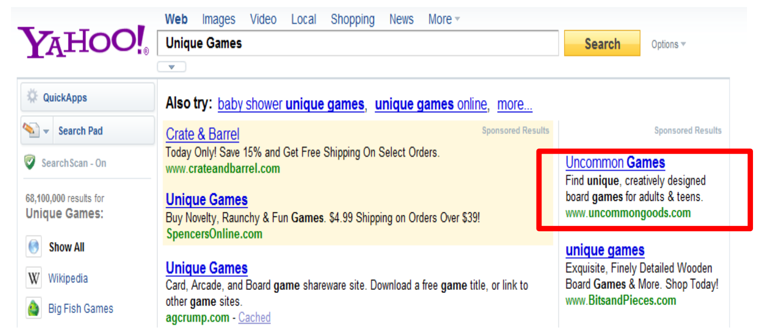
2. We can purchase Unique Games on-line!

bing: 70 million pages

What are Unique Games?

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



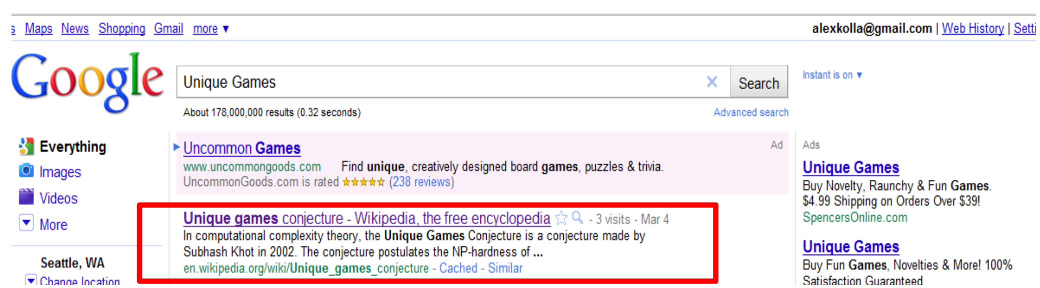


Yahoo!: 69 million pages

2. We can purchase Unique Games on-line!

bing: 70 million pages

3. Unique Games are related to the Unique Games Conjecture...



Google: 178 million pages

Unique Games = Unique Label Cover Problem

Given: set of constraints

Linear Equations mod k :
$x_i - x_j = c_{ij} \bmod k$

GOAL

k =“alphabet” size

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

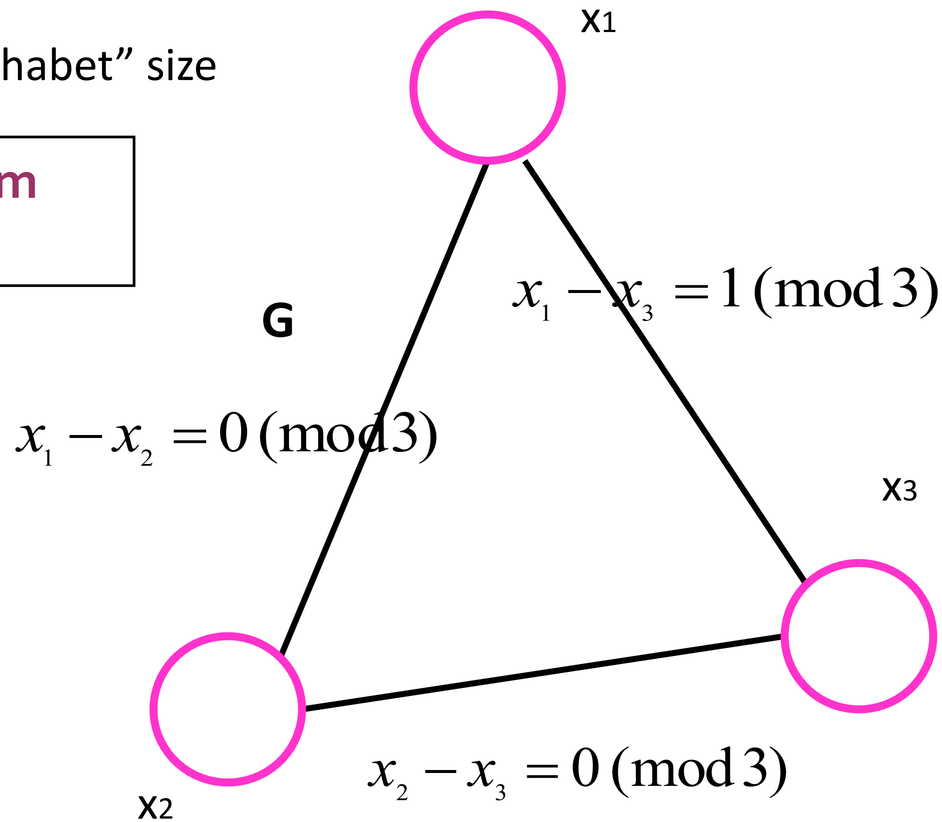
EXAMPLE

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

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

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

The constraint graph



Unique Games , an Example

Given: set of constraints

Linear Equations mod k :

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

GOAL

k =“alphabet” size

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

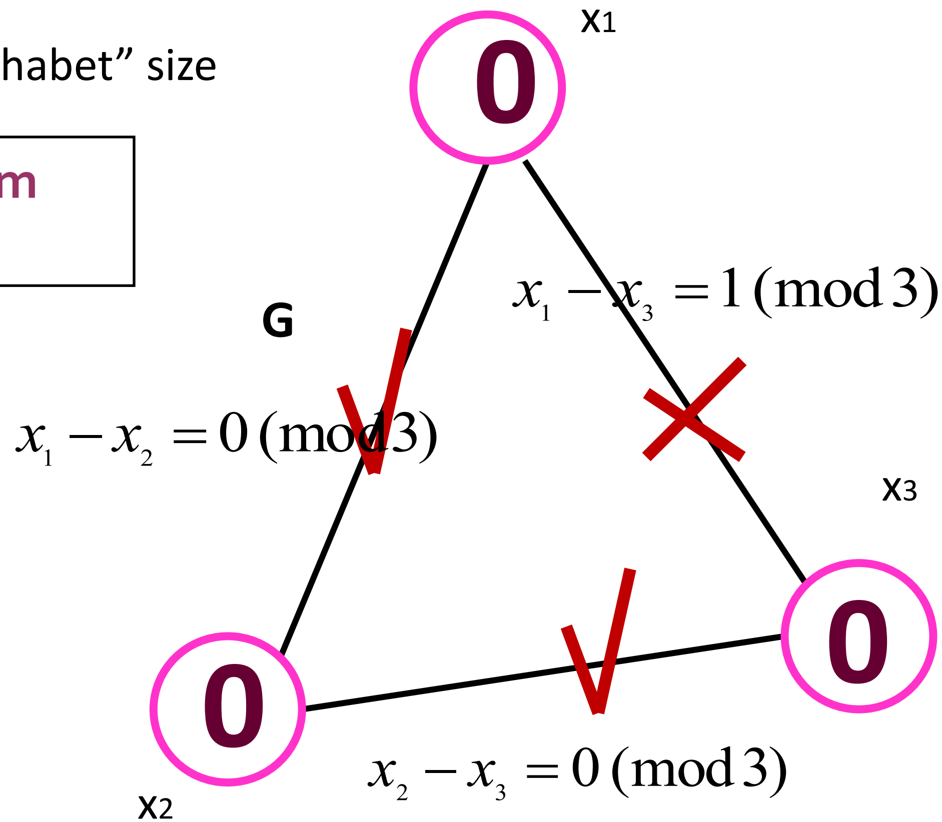
EXAMPLE

$$x_1 - x_2 = 0\ (\mathrm{mod}\ 3)$$ ✓

$$x_2 - x_3 = 0\ (\mathrm{mod}\ 3)$$ ✓

$$x_1 - x_3 = 1\ (\mathrm{mod}\ 3)$$ ✗

The constraint graph



Satisfy 2/3 constraints

Unique Games , an Example

Given: set of constraints

Linear Equations mod k :

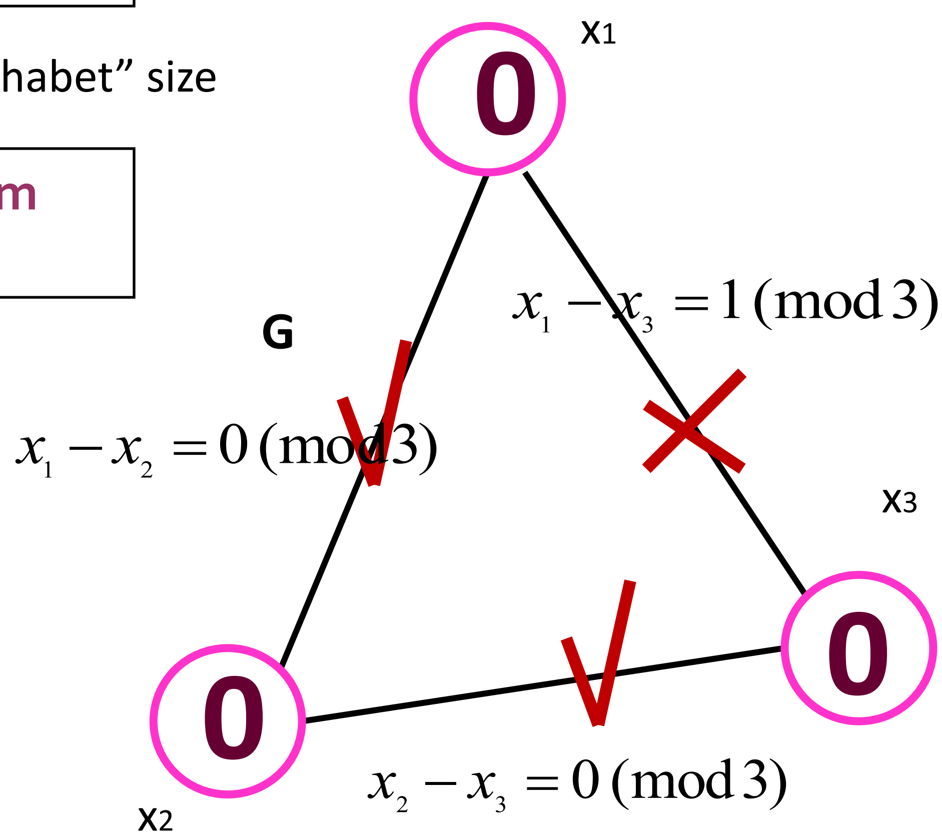
$x_i - x_j = c_{ij} \bmod k$

GOAL

k =“alphabet” size

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

The constraint graph



EXAMPLE

$x_1 - x_2 = 0 \pmod 3$ ✓

$x_2 - x_3 = 0 \pmod 3$ ✓

$x_1 - x_3 = 1 \pmod 3$ ✗

Rest of the talk: d-regular graphs

Unique Games Conjecture

- [Khot'02] For every positive ε and δ there is a large enough k s.t. for some instance of Unique Games with alphabet size k and $OPT > 1 - \varepsilon$, it is NP hard to satisfy a δ fraction of all constraints.
- Given a UG instance (graph and set of constraints over alphabet of size k) with the guarantee that it is 99% satisfiable, it is NP-hard to find an assignment that satisfies more than ½ of the constraints (for some 99% and some ½).

Is Unique Games Conjecture True?

Unique Games Conjecture

- UGC: given a UG instance (graph and set of constraints over alphabet of size k) with the guarantee that it is 99% satisfiable, it is NP-hard to find an assignment that satisfies more than ½ of the constraints (for some 99% and some ½).

Really embarrassing not to know, since solving systems of linear equations (exactly) is very easy!

Where to begin if we want to refute UGC?

- Several attempts in recent years to refute or prove UGC.
- Lot of progress but still no consensus.

Plan of attack: start ruling out cases.

Easy Instances

- Classify graphs according to their “spectral profile” (eigenvalues)
- Expanders [AKKTSV’08,KT’08],
- Local expanders, graphs with relatively few large eigenvalues [AIMS’09,SR’09,K’10]

Easy Distributions

- Find distributions that are hard?
  - Random Instances : NO! Follows from expander result.
  - Quasi-Random Instances? [KMM’10] NO!

Summary: Algorithmic Results for UG

Algorithm	On 1-ε instances
Khot	$1-O(k^2\,\varepsilon^{1/5}\sqrt{\log(1/\varepsilon)})$
Trevisan	$1-O(\sqrt[3]{\varepsilon \log n})$
Gupta-Talwar	$1-O(\varepsilon \log n)$
CMM1	$k^{-\varepsilon/2-\varepsilon}$
CMM2	$1-O(\varepsilon \sqrt{\log n}\sqrt{\log k})$

General Graphs

SDP/LP based

Special Graphs

Expander

AKKTSV'08 KT'08,MM'10	Constant, depend on conductance

Local expander

AIMS'09, SR'09	Constant, depends on local expansion

Tight for SDP, there is counterexample

Almost all above approaches were LP or SDP based

Summary: Algorithmic Results for UG

General Graphs

Algorithm	On 1-ε instances
Khot	$1-O(k^2\,\varepsilon^{1/5}\sqrt{\log(1/\varepsilon)})$
Trevisan	$1-O(\sqrt[3]{\varepsilon \log n})$
Gupta-Talwar	$1-O(\varepsilon \log n)$
CMM1	$k^{-\varepsilon/2-\varepsilon}$
CMM2	$1-O(\varepsilon \sqrt{\log n}\,\sqrt{\log k})$

SDP/LP based

Tight for SDP, there is counterexample

Special Graphs

Expander	AKKTSV'08 KT'08,MM'10	Constant, depend on conductance
Local expander	AIMS'09, SR'09	Constant, depends on local expansion
Few large eigenvalues	K'10	Quality and running time depends on eigenspace

Purely SPECTRAL Approach "beats" SDP

Summary: Algorithmic Results for UG

General Graphs

Algorithm	On 1-ε instances
Khot	$1-O(k^2 \varepsilon^{1/5}\sqrt{\log(1/\varepsilon)})$
Trevisan	$1-O(\sqrt[3]{\varepsilon \log n})$
Gupta-Talwar	$1-O(\varepsilon \log n)$
CMM1	$k^{-\varepsilon/2-\varepsilon}$
CMM2	$1-O(\varepsilon \sqrt{\log n} \sqrt{\log k})$

Special Graphs

Special graph	Reference	Result
Expander	AKKTSV'08 KT'08,MM'10	Constant, depends on conductance
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

General Graphs

Algorithm	On 1-ε instances
Khot	$1-O(k^2 \varepsilon^{1/5}\sqrt{\log(1/\varepsilon)})$
Trevisan	$1-O(\sqrt[3]{\varepsilon \log n})$
Gupta-Talwar	$1-O(\varepsilon \log n)$
CMM1	$k^{-\varepsilon/2-\varepsilon}$
CMM2	$1-O(\varepsilon \sqrt{\log n} \sqrt{\log k})$

Special Graphs

Expander	AKKTSV'08 KT'08,MM'10	Constant, depends on conductance
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

General Graphs

Algorithm	On 1-ε instances
Khot	$1-O(k^2 \varepsilon^{1/5}\sqrt{\log(1/\varepsilon)})$
Trevisan	$1-O(\sqrt[3]{\varepsilon \log n})$
Gupta-Talwar	$1-O(\varepsilon \log n)$
CMM1	$k^{-\varepsilon/2-\varepsilon}$
CMM2	$1-O(\varepsilon \sqrt{\log n} \sqrt{\log k})$

Special Graphs

Expander	AKKTSV'08 KT'08,MM'10	Constant, depends on conductance
Local expander	AIMS'09, SR'09	Constant, depends on local expansion
Few large eigenvalues	K'10	Quality and running time depends on eigenspace

KMM'10: Semi-Random instances are easy

Plan for Today

1. Unique Games Conjecture(UGC)

2**. Spectra of Graphs**

3. Towards Refuting UGC on almost-all Graphs

4. Open Questions

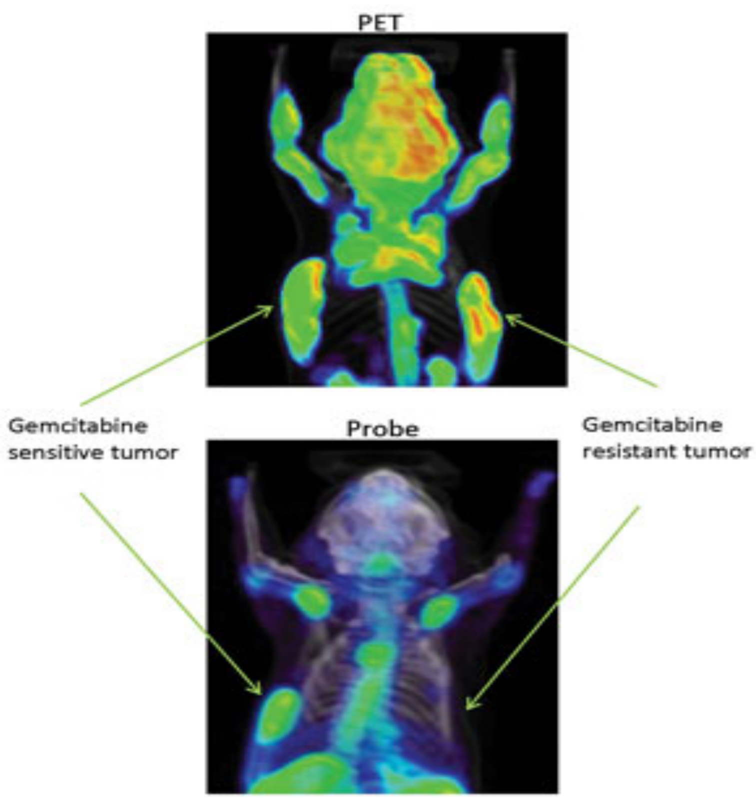
Spectral Graph Theory and Applications

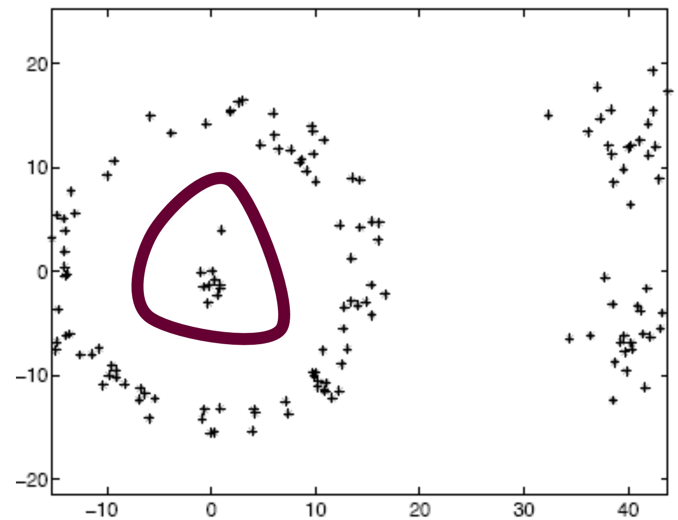
- Image Segmentation

How to pick the right segmentation?

Spectral Graph Theory and Applications

- Data clustering:
find points of similarity



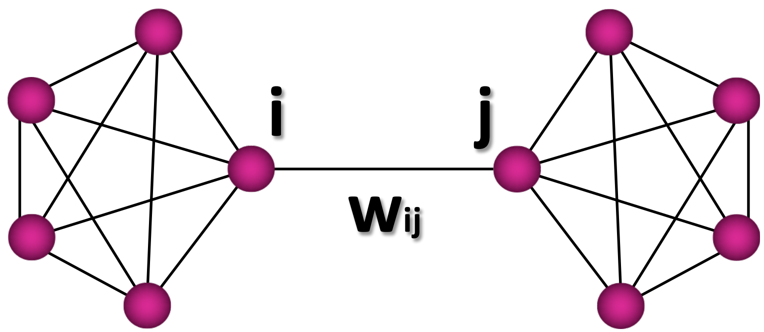


Many more :

-Coding Theory
-Network Security
-Convex Optimization
-...

Representing Graphs

Obviously, we can represent a graph with an nxn matrix

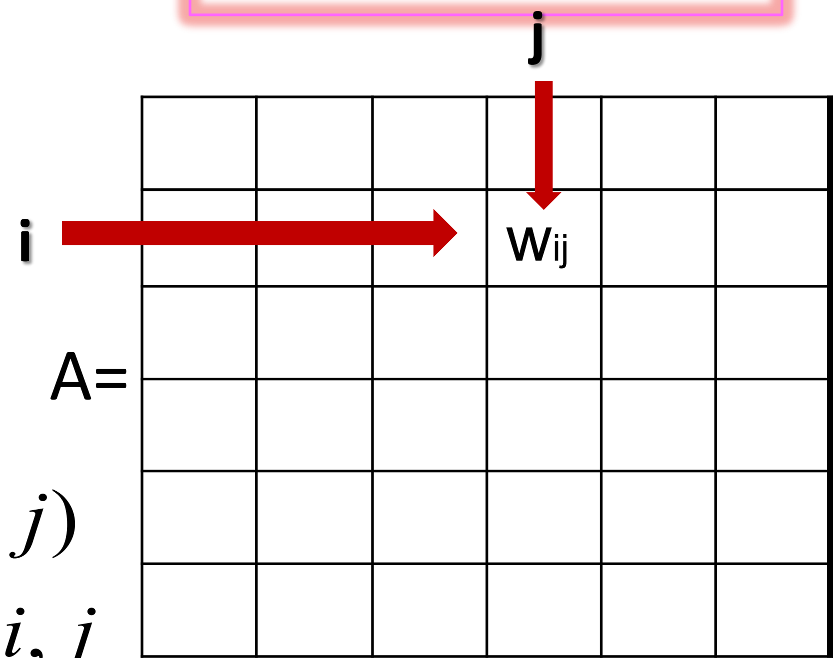


V: n nodes
E: m edges

G = {V,E}

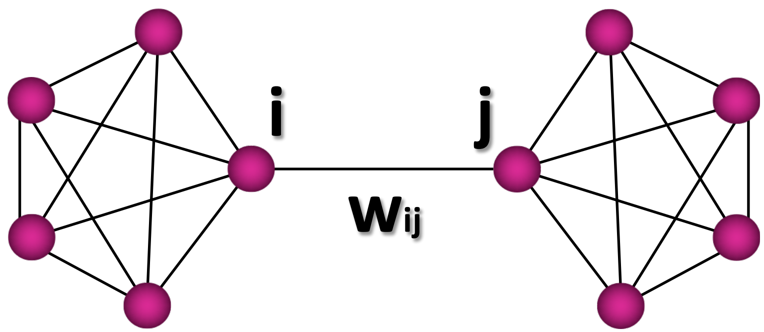
Adjacency matrix

A=



$$A_{ij} = \begin{cases} w_{ij} & \text{weight of edge } (i,j) \\ 0 & \text{if no edge between } i, j \end{cases}$$

Representing Graphs

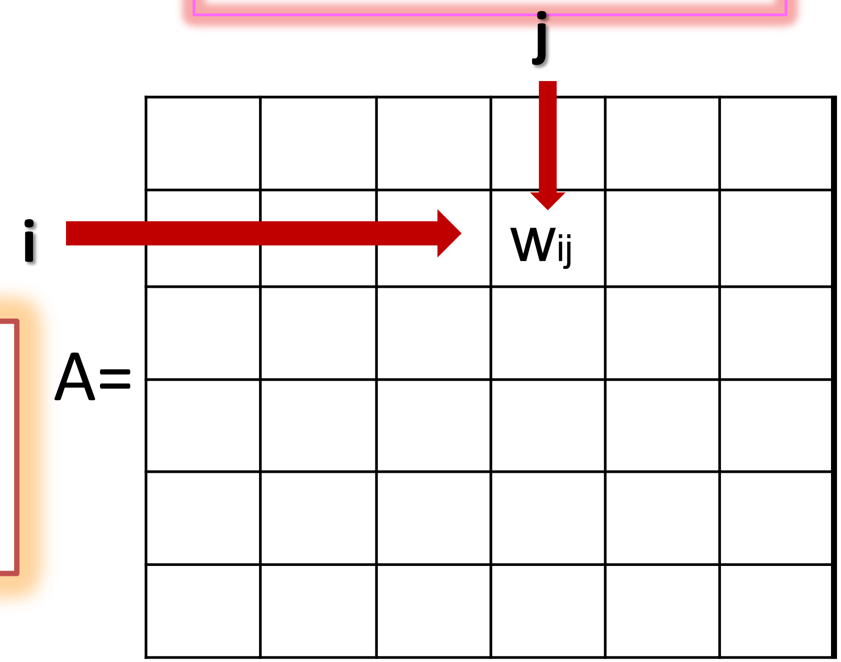


V: n nodes
E: m edges

G = {V,E}

Obviously, we can represent a graph with an nxn matrix

Adjacency matrix



Not-so-obvious:
Once we have matrix representation view graph as **linear operator**

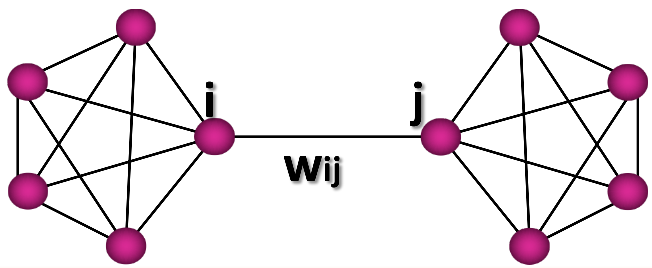
$$A : \Re^n \rightarrow \Re^n$$

Can be used to multiply vectors

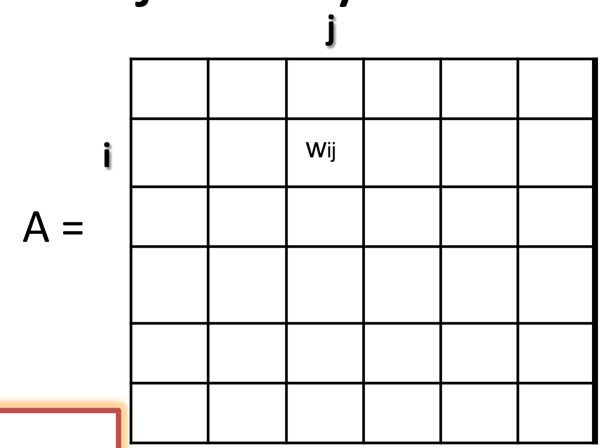
$$y = Ax$$

Amazing how this point of view gives information about graph

Graph Spectrum



Adjacency matrix



$$A : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$$

Well-known:
spectrum of linear operators
gives information about them

Already know:
A multiplies vectors

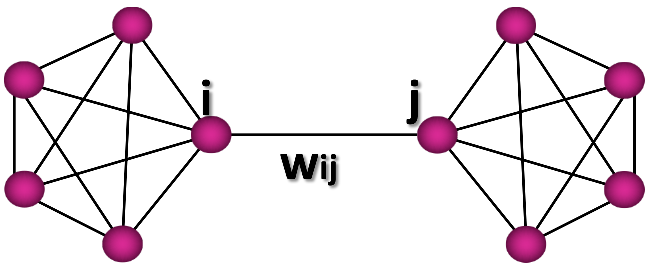
There are "special" vectors that
don't "rotate" just scale:

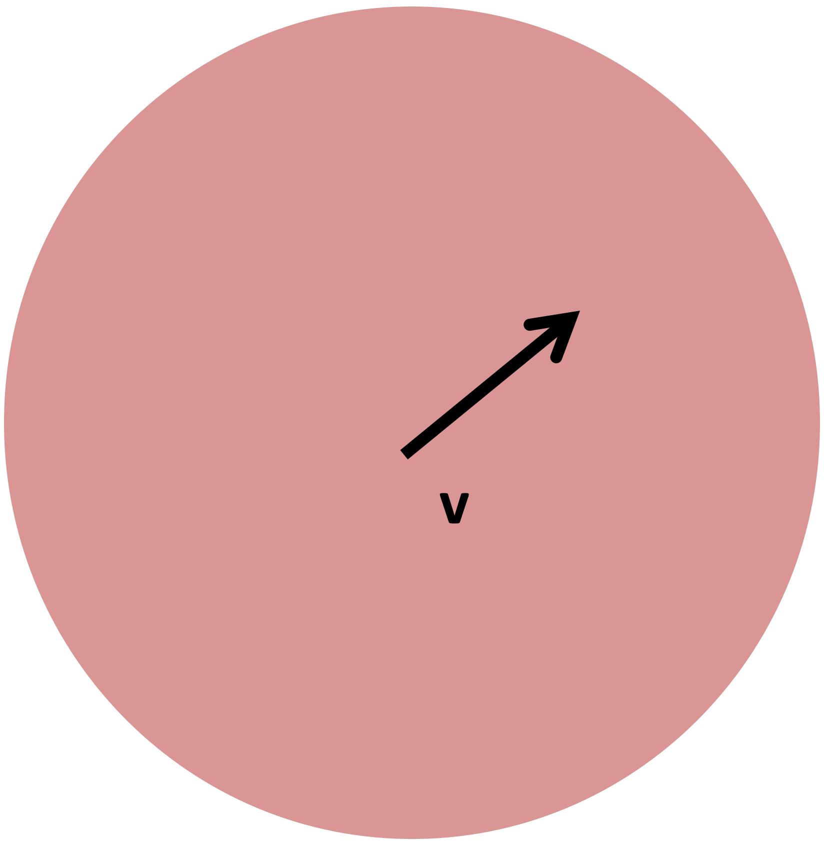
eigenvectors

$$Av = \lambda v$$

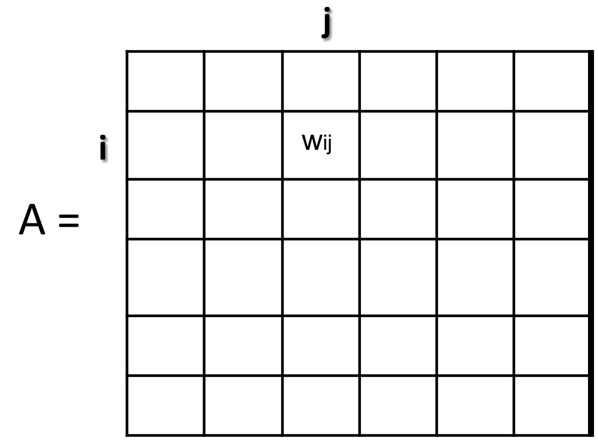
v eigenvector,
λ eigenvalue ("scaling" factor)

Graph Spectrum





Adjacency matrix

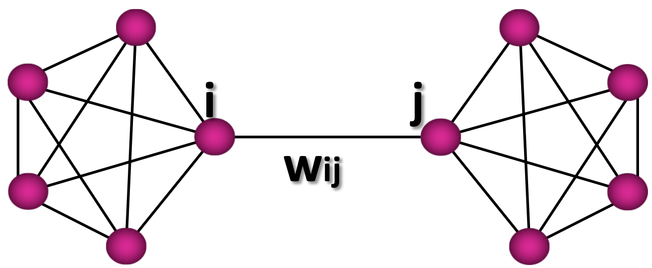


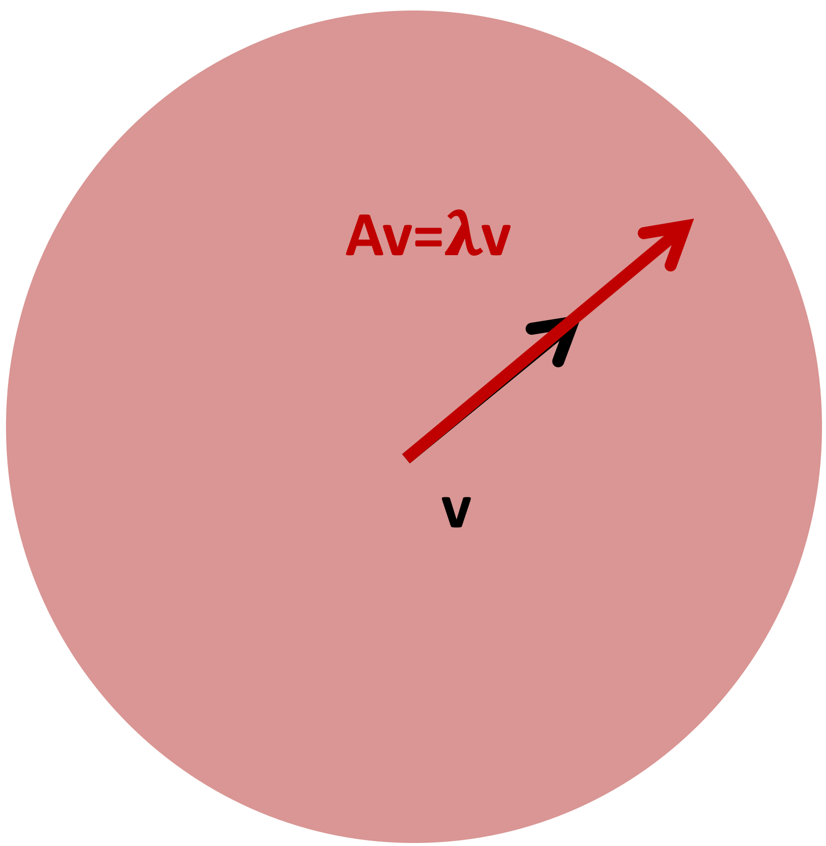
$$A : \Re^n \to \Re^n$$

$$Av = \lambda v$$

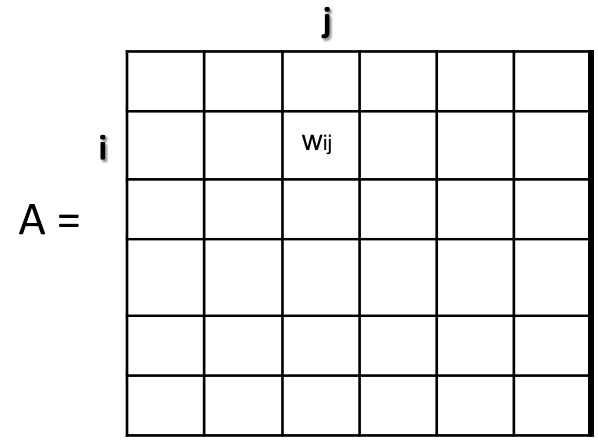
v eigenvector,
λ eigenvalue

Graph Spectrum





Adjacency matrix

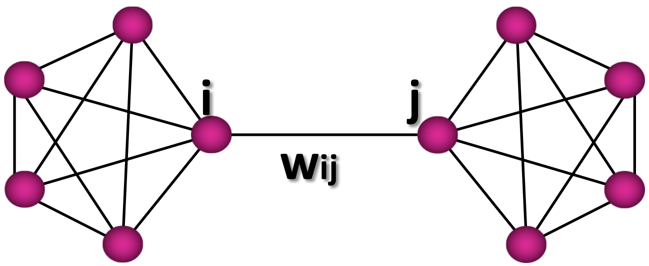


$$A : \Re^n \rightarrow \Re^n$$

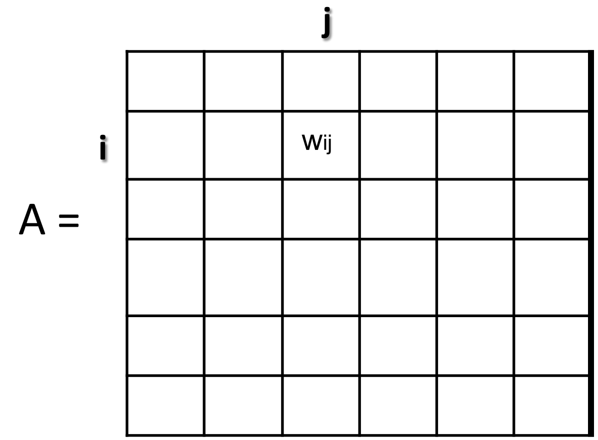
$$Av = \lambda v$$

v eigenvector,
λ eigenvalue

Graph Spectrum



Adjacency matrix



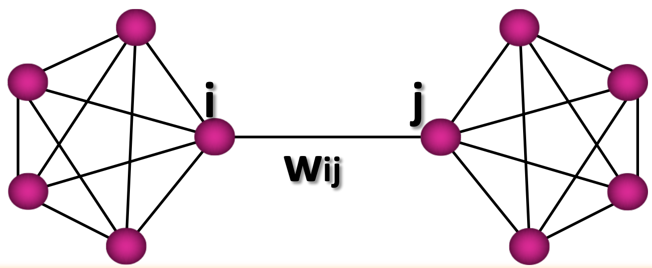
$$A : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$$

$$Av = \lambda v$$

Graph SPECTRUM =
List of eigenvalues $\{\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n\}$

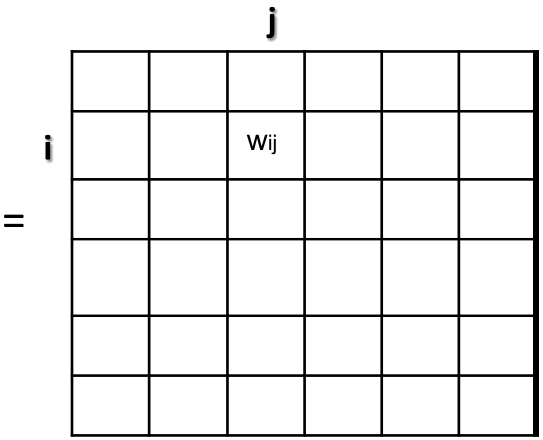
v eigenvector,
λ eigenvalue

"Listen" to the Graph



Adjacency matrix

A =



List of eigenvalues

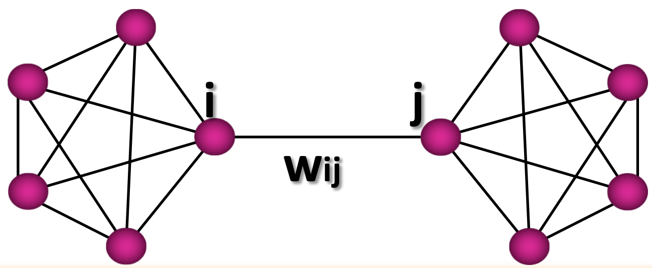
$\{\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n\}$:graph SPECTRUM

Eigenvalues reveal global graph properties not apparent from edge structure

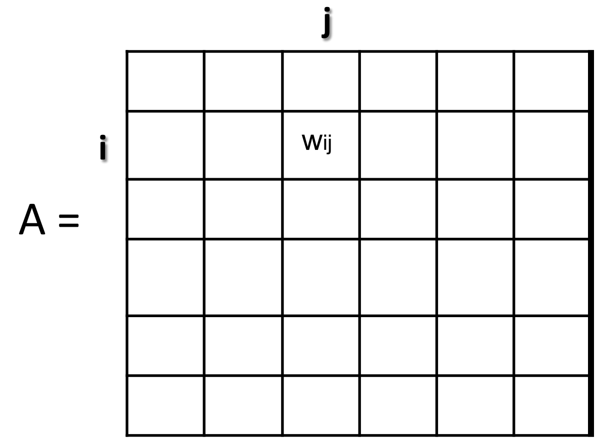
Hear shape of the drum

A drum:

"Listen" to the Graph



Adjacency matrix



List of eigenvalues

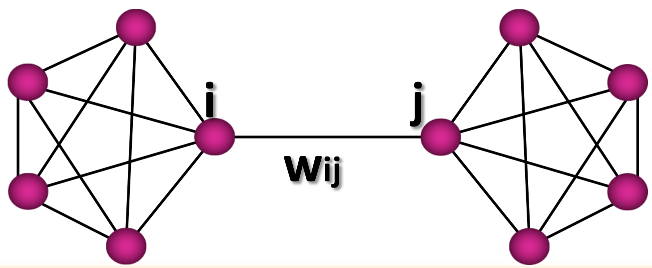
$\{\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n\}$:graph SPECTRUM

Eigenvalues reveal global graph properties not apparent from edge structure

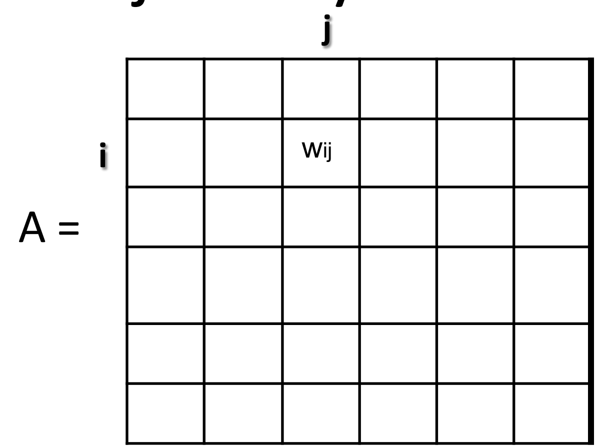
Hear shape of the drum

Its sound:

"Listen" to the Graph



Adjacency matrix



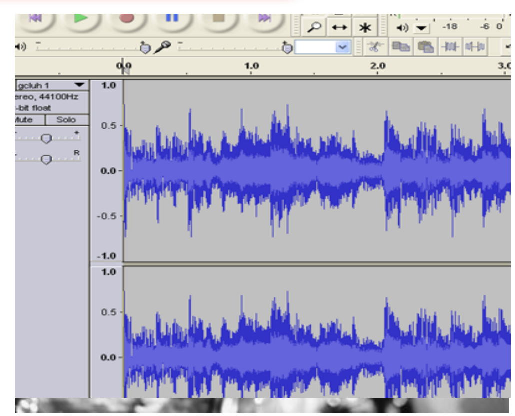
List of eigenvalues

$\{\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n\}$:graph SPECTRUM

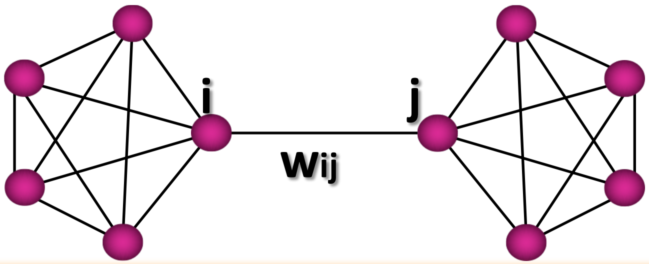
Eigenvalues reveal global graph properties not apparent from edge structure

Hear shape of the drum

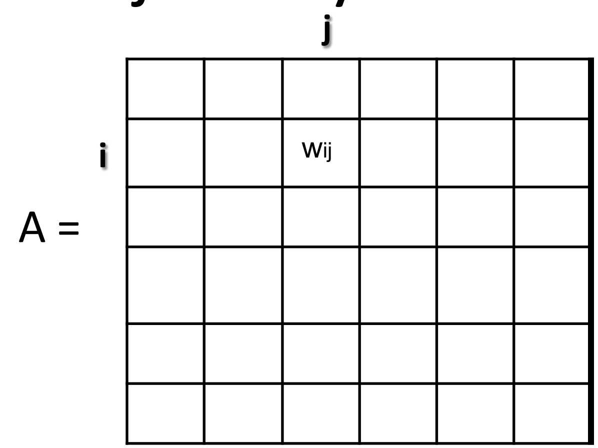
Its sound (eigenfrequenies):



“Listen” to the Graph



Adjacency matrix

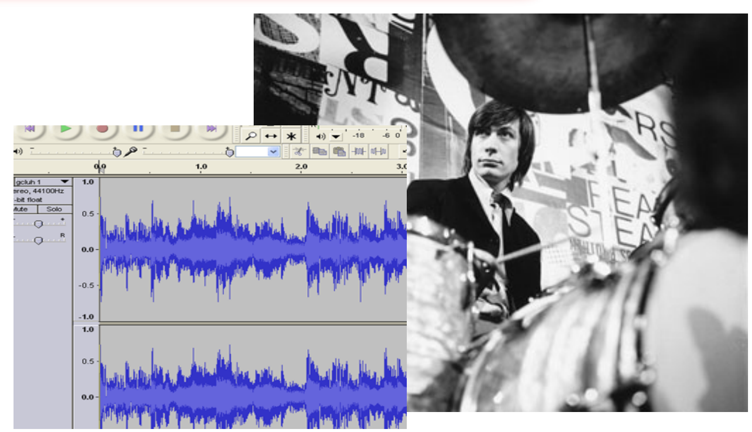


List of eigenvalues

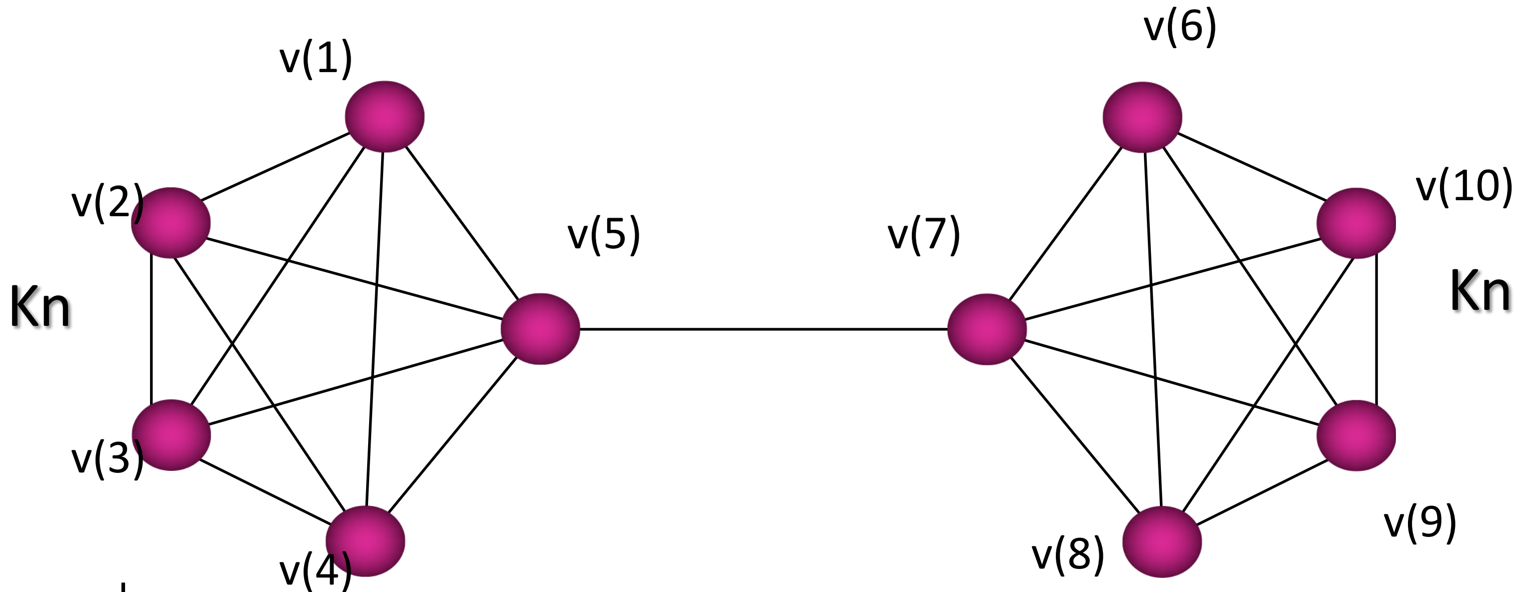
$\{\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n\}$:graph SPECTRUM

Eigenvalues reveal global graph properties not apparent from edge structure

If graph was a drum, spectrum would be its sound



Eigenvectors are Functions on Graph

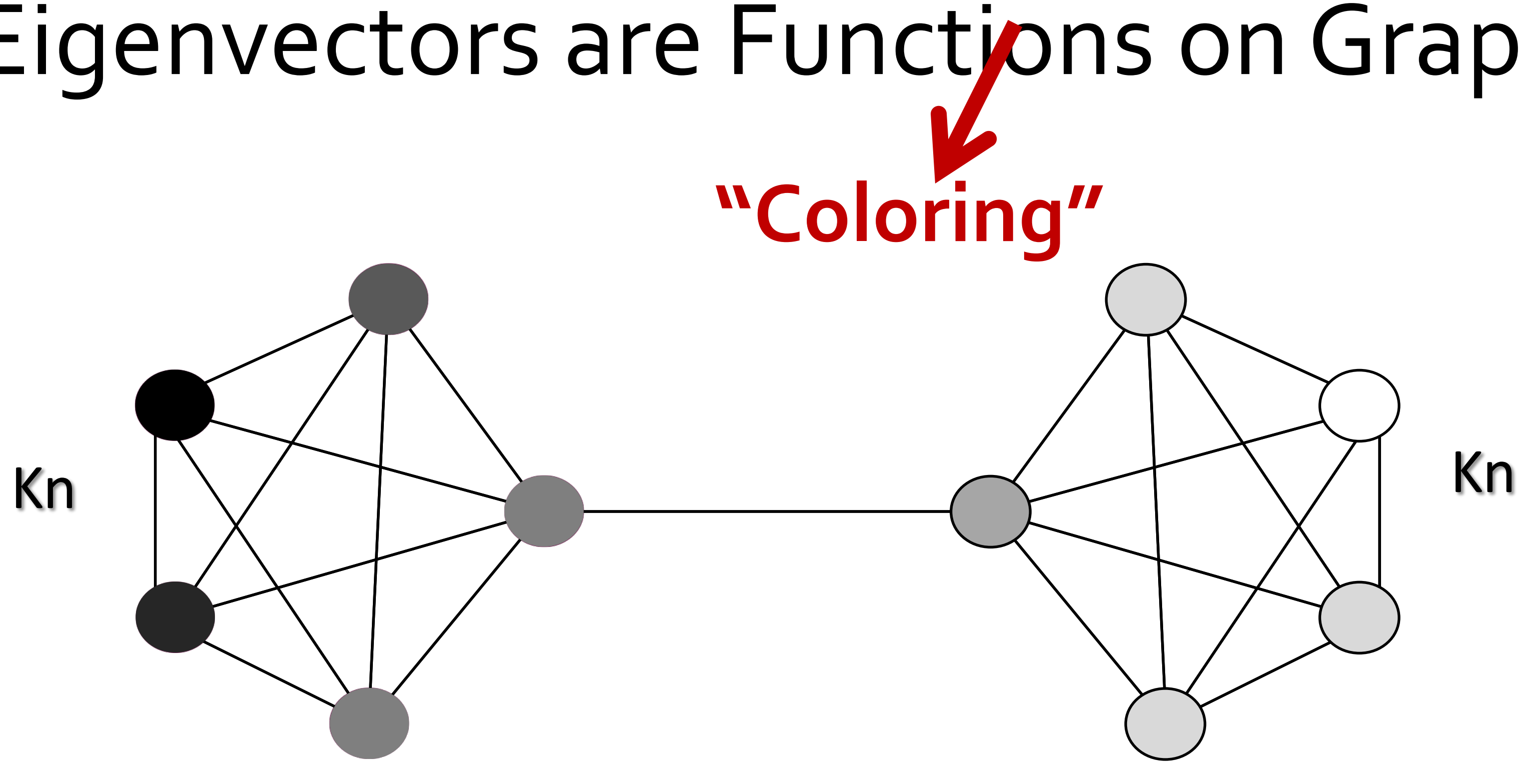


V: 2n nodes

$$v \in \Re^n, \quad v : V \to \Re \qquad Av = \lambda v$$

$v(i) =$ value at node i

Eigenvectors are Functions on Graph



V: 2n nodes

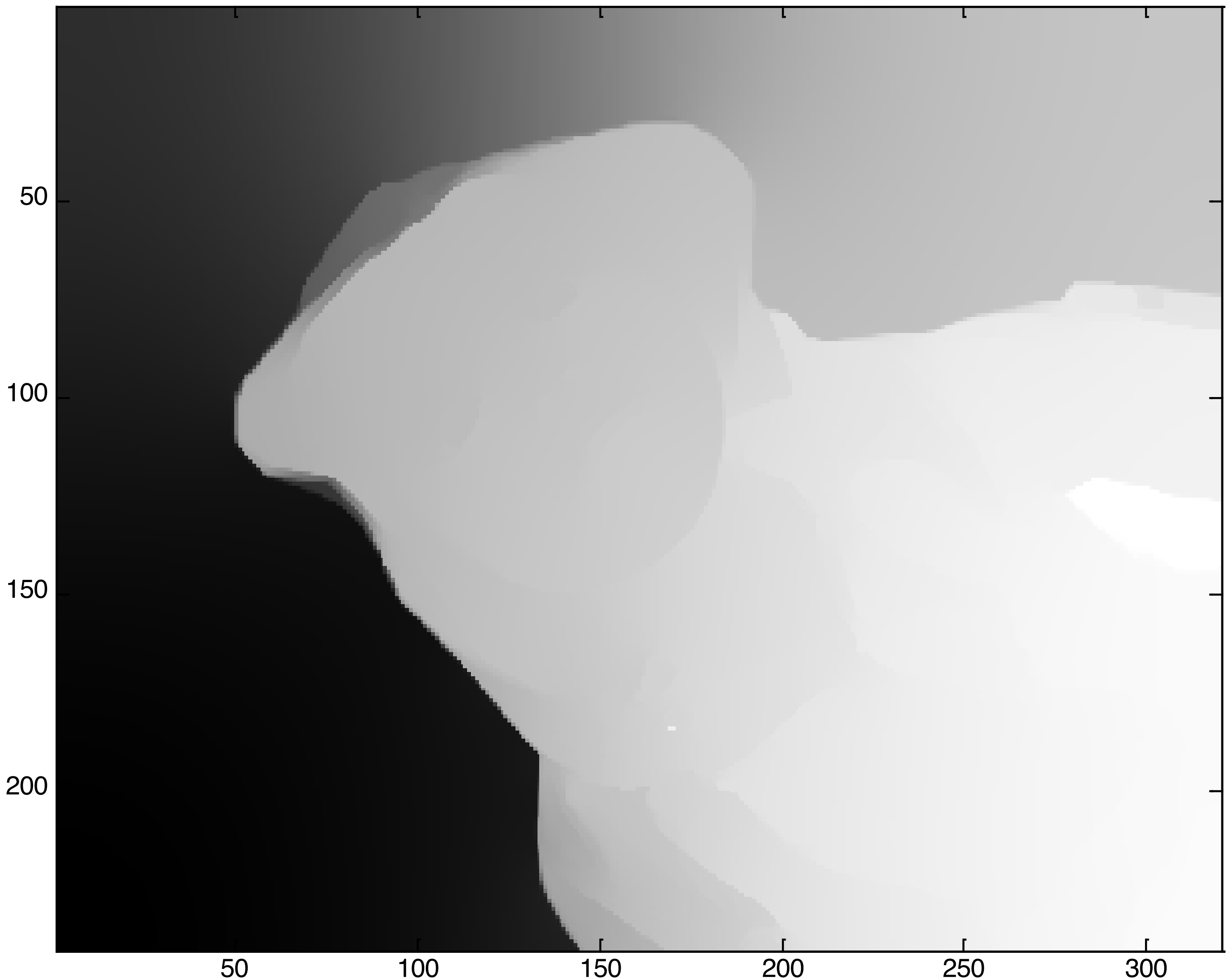
$$v \in \Re^n, \quad v : V \to \Re \qquad Av = \lambda v$$

$v(i) =$ value at node i = different shade of grey

So, let's See the Eigenvectors

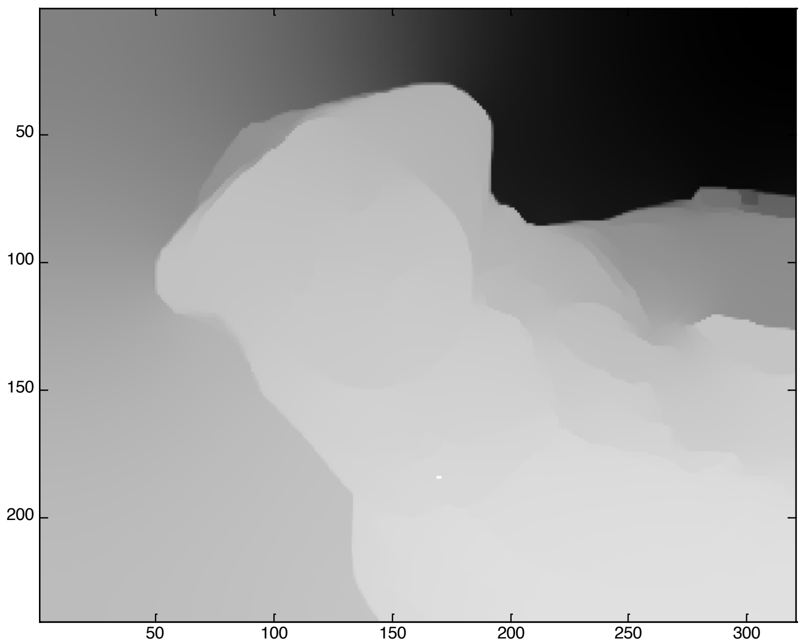
* Slides from Dan Spielman

The second eigenvector



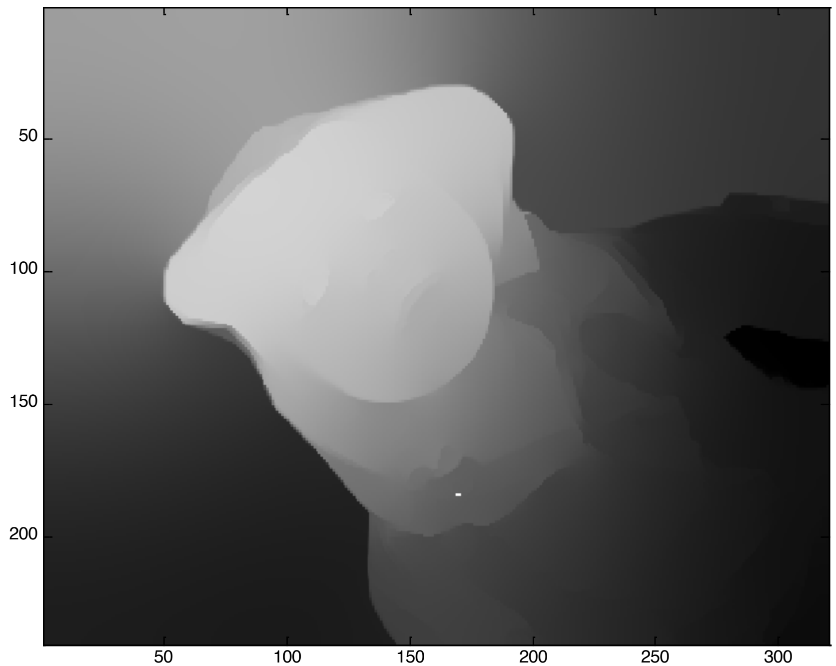
* Slides from Dan Spielman

Third Eigenvector



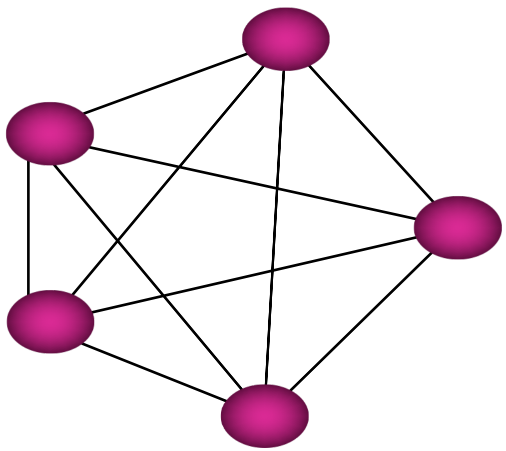
* Slides from Dan Spielman

Fourth Eigenvector

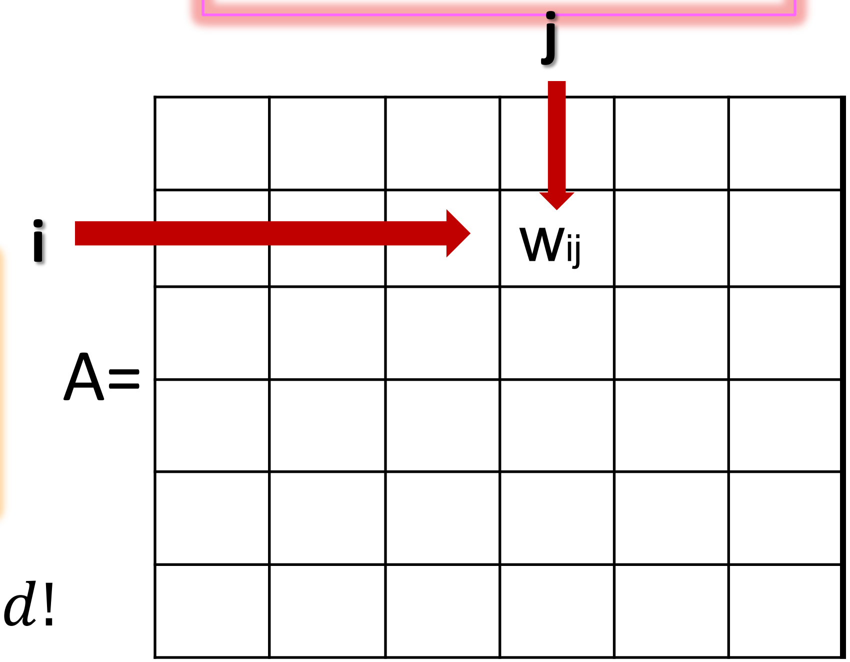


* Slides from Dan Spielman

Representing Graphs (d-regular)



Adjacency matrix



List of eigenvalues
$\{d=\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n\}$:graph
SPECTRUM

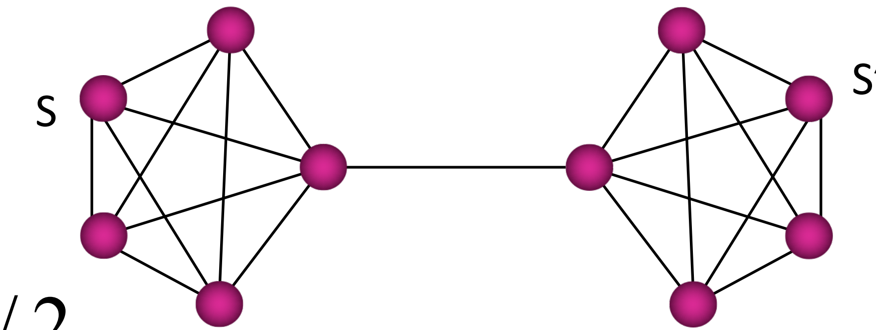
$\lambda \equiv \lambda_2 < d \Leftrightarrow$ *Graph connected*!

$d - \lambda_2$ also called "algebraic connectivity"

The further from 0, the more connected

Cuts and Algebraic Connectivity

Cuts in a graph:



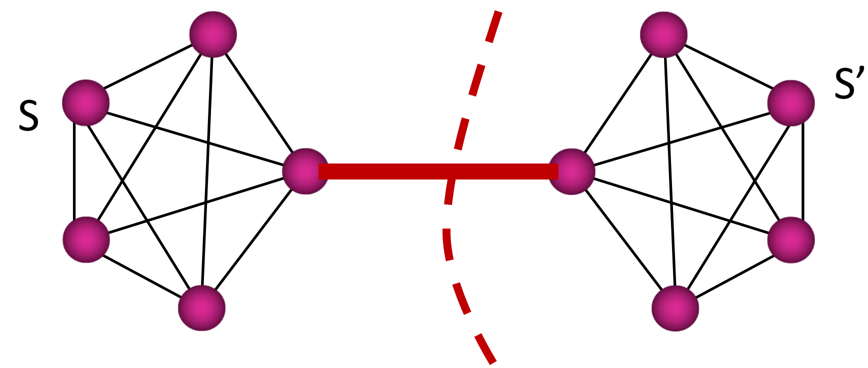
$$cut(S,S') = \frac{E(S,S')}{|S|}, |S| \leq n/2$$

Graph not well-connected when “easily” cut in two pieces

Cuts and Algebraic Connectivity

Sparsest Cut:

$$h(G) = \min_{S:|S| \le n/2} \frac{E(S,\bar{S})}{|S|}$$



Graph not well-connected when “easily” cut in two pieces

Would like to know Sparsest Cut but NP hard to find

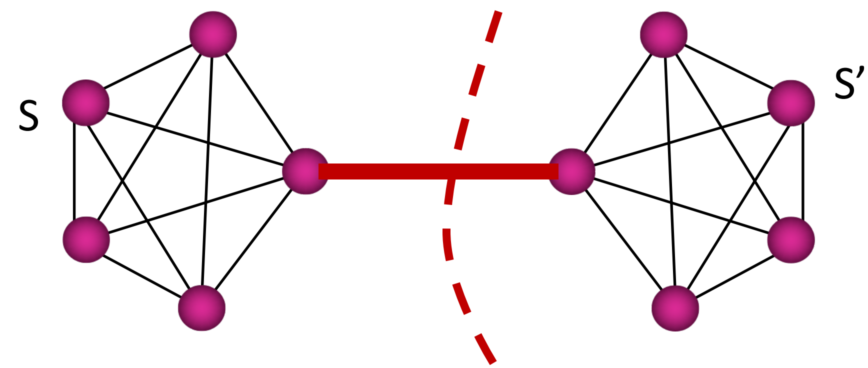
How does algebraic connectivity relate to standard connectivity?

Theorem(Cheeger-Alon-Milman): $\frac{d-\lambda}{2} \le h(G) \le \sqrt{2d}\,\sqrt{d-\lambda}$

Cuts and Algebraic Connectivity

Sparsest Cut:

$$h(G) = \min_{S:|S| \leq n/2} \frac{E(S,\bar{S})}{|S|}$$



Graph not well-connected when "easily" cut in two pieces

Would like to know Sparsest Cut but NP hard to find

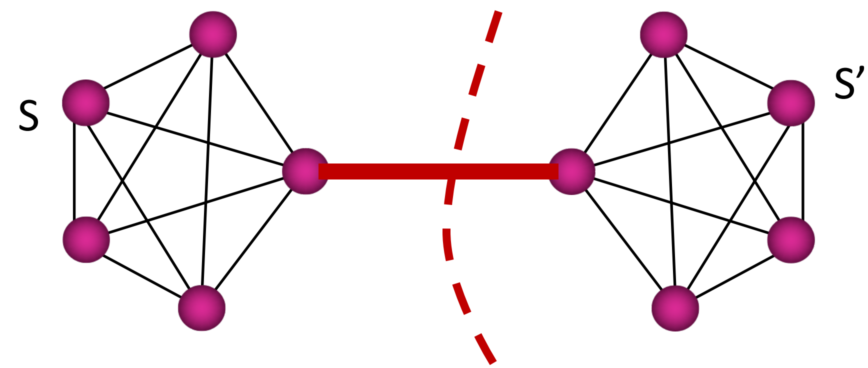
How does algebraic connectivity relate to standard connectivity?

Algebraic connectivity large ⟺ Graph well-connected

Cuts and Algebraic Connectivity

Sparsest Cut:

$$h(G) = \min_{S:|S| \le n/2} \frac{E(S,\bar{S})}{|S|}$$

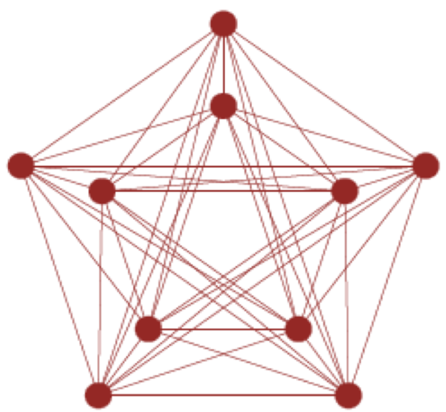


In fact, we can find a cut with the guarantee below, from the second eigenvector (and from all the eigenvectors)

$$\frac{d-\lambda}{2} \le h(G) \le \sqrt{2d}\,\sqrt{d-\lambda}$$

Graphs with no Small Cuts

Certain graphs have no small cuts: **Expanders**

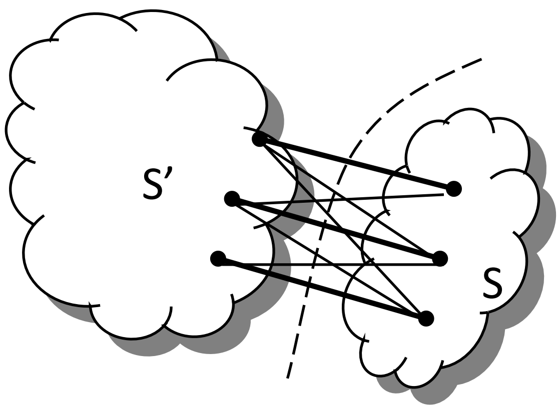


Very useful for applications

- Constructing robust networks.
- Routing.
- Maximizing throughput with fixed network topology.
- Error-correcting codes.
- Complexity theory.

Expanders in a Nutshell

Edge expansion: $$h(G) = \min_{S:|S|\leq n/2} \frac{E(S,\overline{S})}{|S|}$$



(Spectral Gap): d-λ=γd

Cheeger : $$\frac{d-\lambda}{2} \leq h(G) \leq \sqrt{2d(d-\lambda)}$$

Plan for Today

1. Unique Games Conjecture(UGC)
2. Spectra of Graphs
3. **Towards Refuting UGC on almost-all Graphs**
4. Open Questions

Unique Games = Unique Label Cover Problem

Given: set of constraints

Linear Equations mod k :

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

GOAL

k =“alphabet” size

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

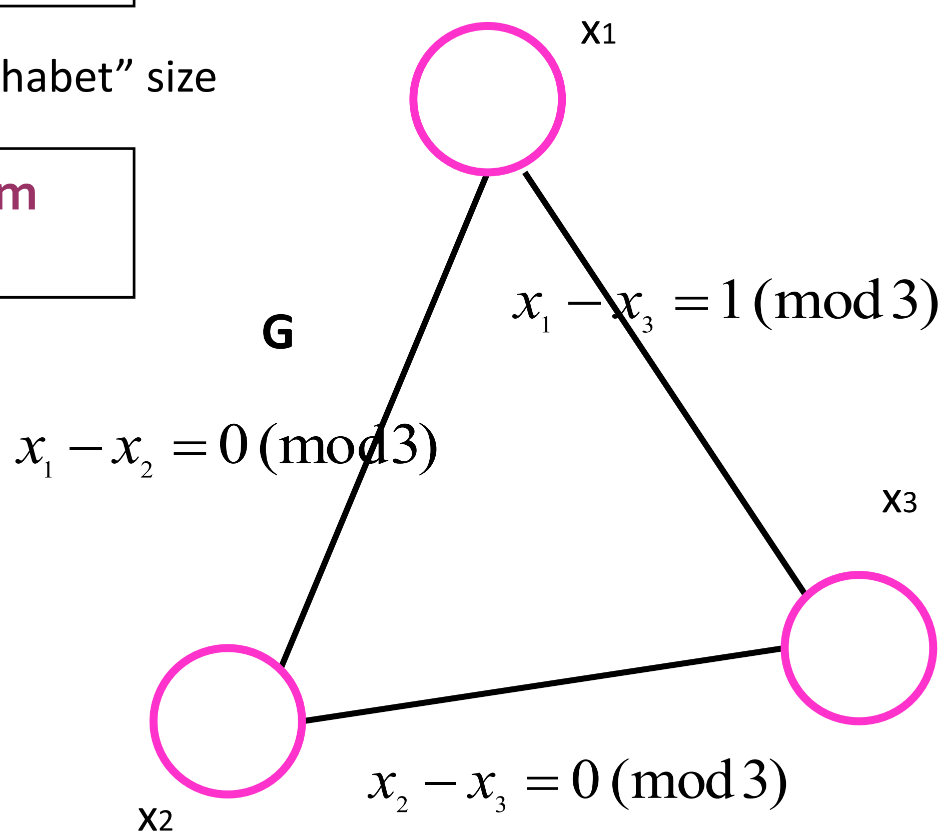
EXAMPLE

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

$$x_2 - x_3 = 0 \pmod 3$$

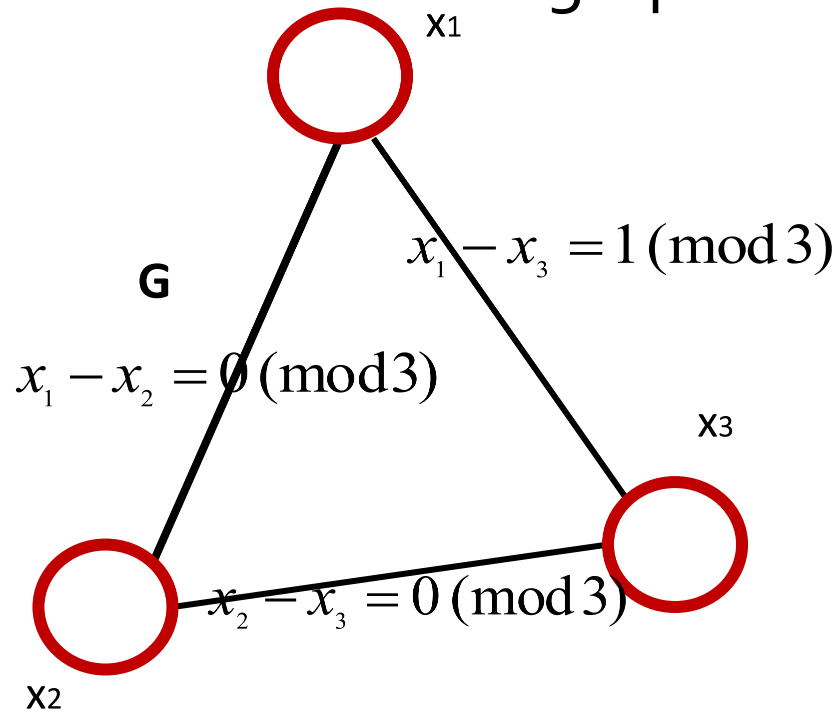
$$x_1 - x_3 = 1 \pmod 3$$

The constraint graph

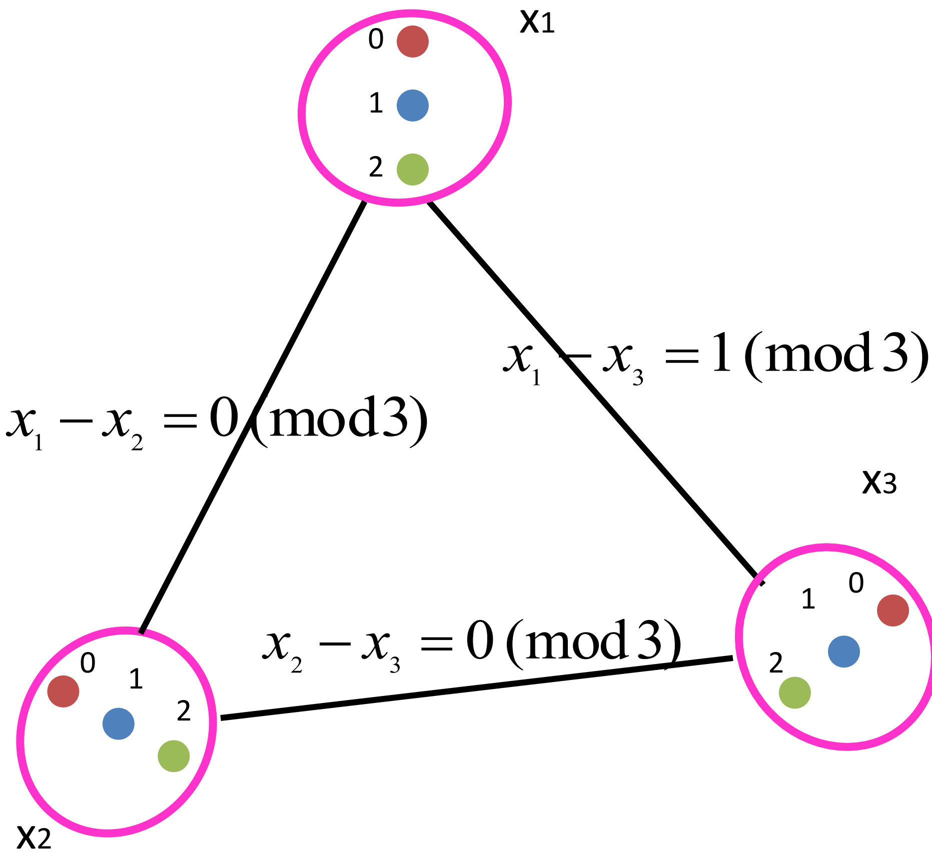


Unique Games and Graphs

1. The "constraint graph"



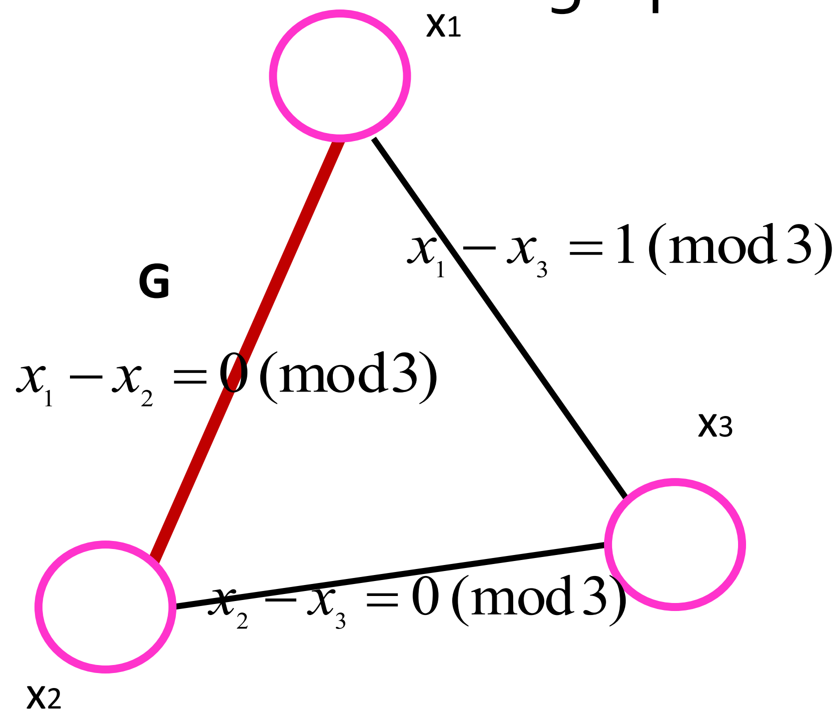
2. The "label-extended" graph



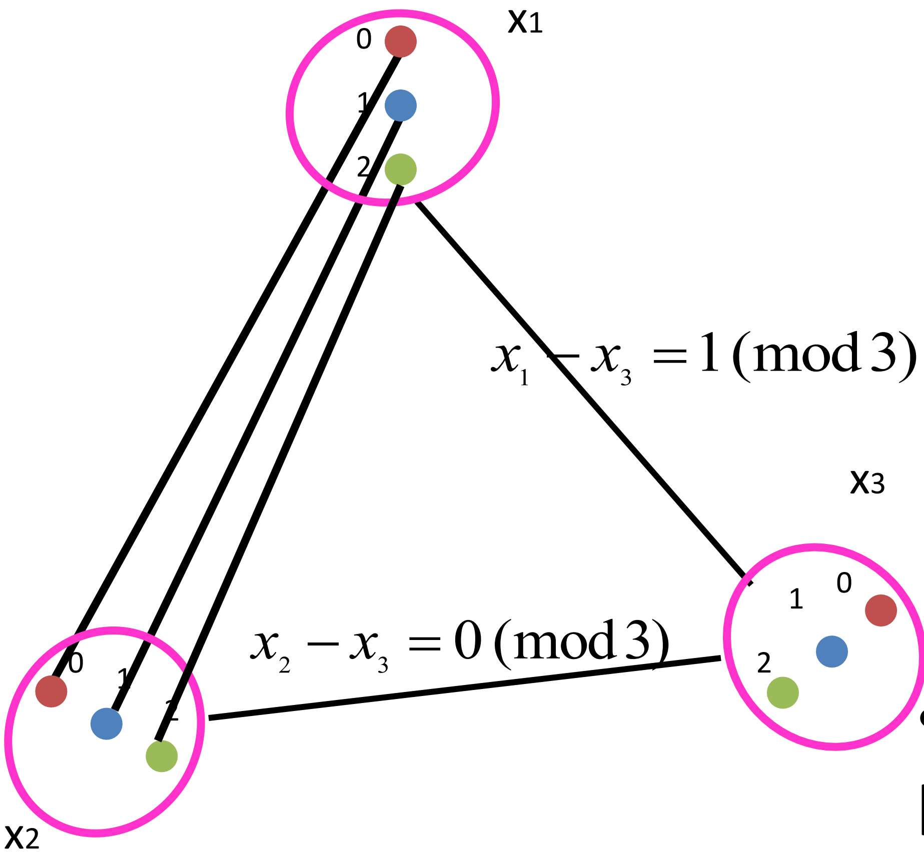
- Replace each vertex with k vertices- one for each label

Unique Games and Graphs

1. The "constraint graph"



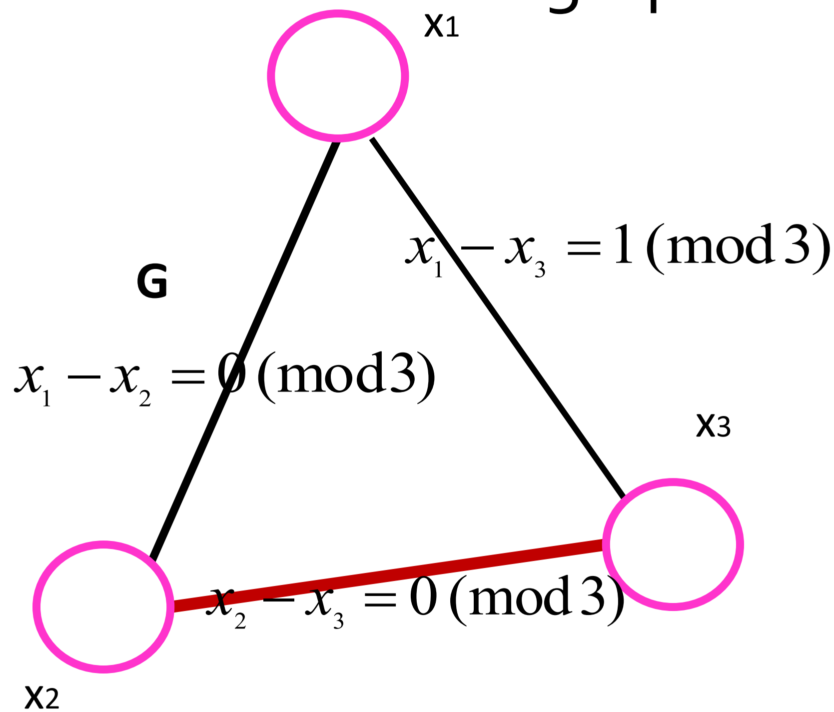
2. The "label-extended" graph



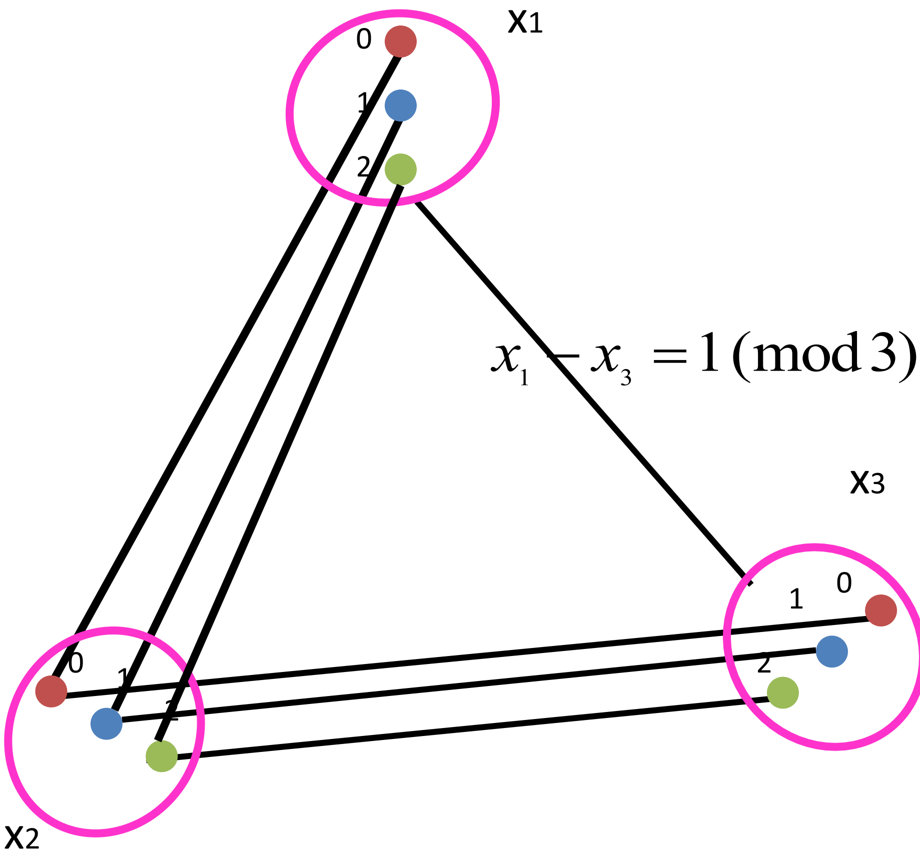
- Replace each vertex with k vertices- one for each label
- Replace each edge with the "permutation matching"

Unique Games and Graphs

1. The "constraint graph"



2. The "label-extended" graph

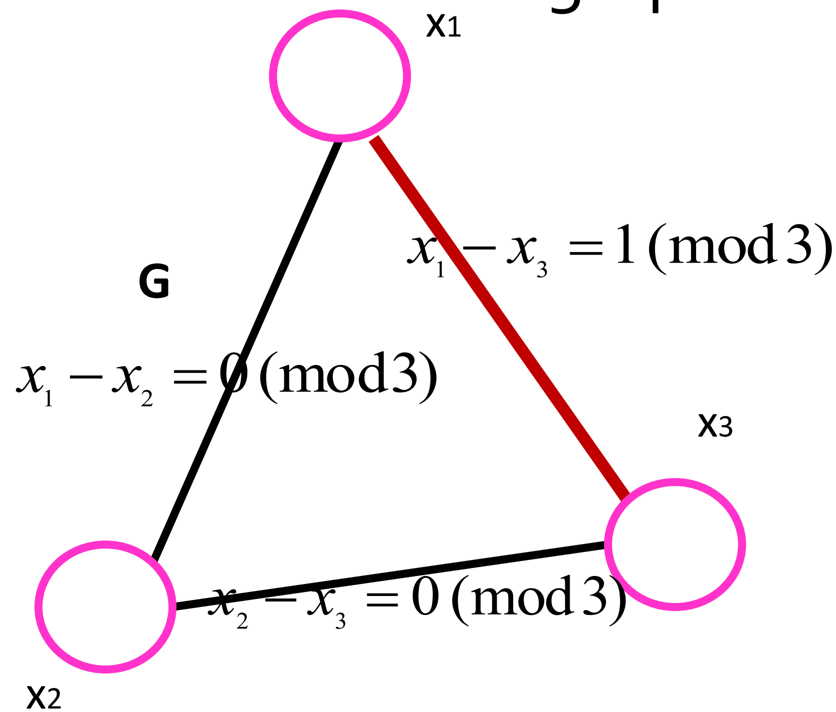


- Replace each vertex with k vertices- one for each label

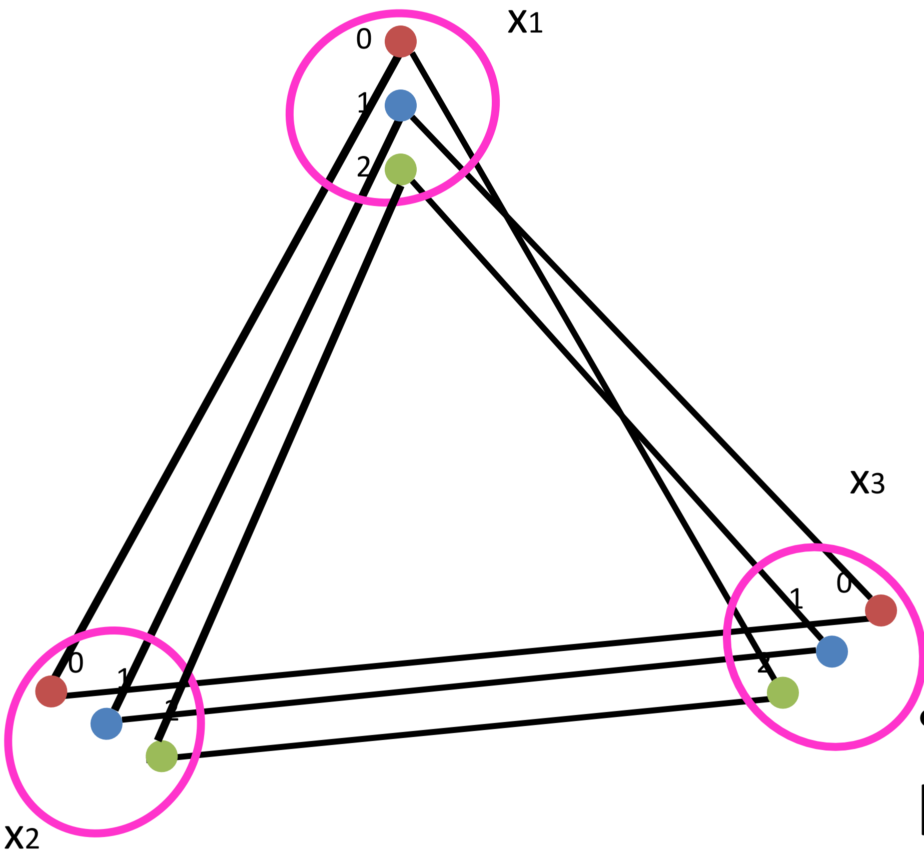
- Replace each edge with the "permutation matching"

Unique Games and Graphs

1. The “constraint graph”



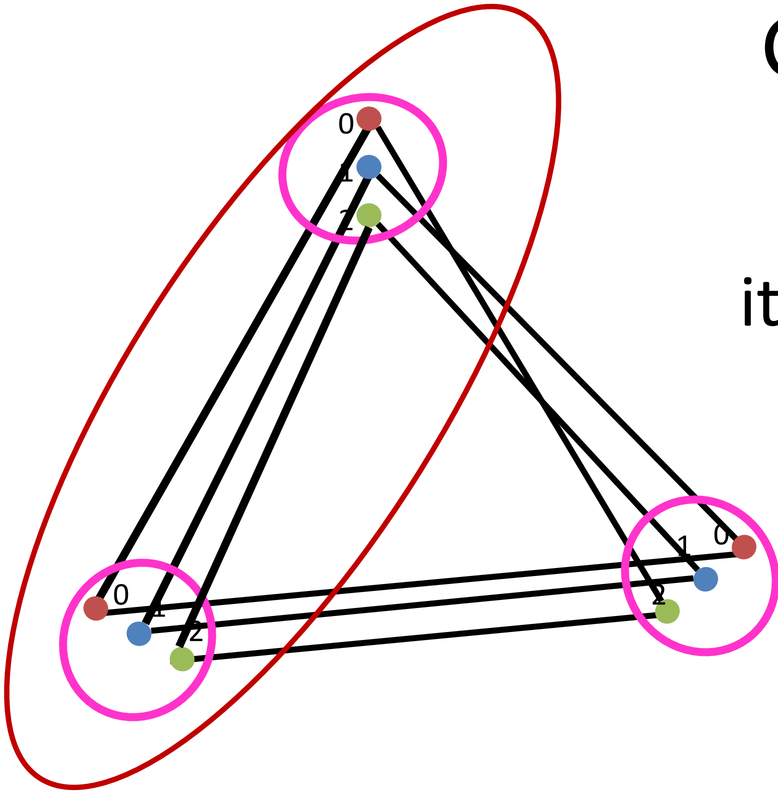
2. The “label-extended” graph



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

•Replace each edge with the “permutation matching”

More Graph Theory: The Label-Extended Graph



GRAPH THEORY?

it's a graph, it has adjacency matrix!

0	0	0	1	0	0			
0	0	0	0	1	0			
0	0	0	0	0	1			

M has each non – zero entry (u,w) replaced by a block corresponding to the permutation on edge

Sketch UGC False on Expanders

UGC FALSE on expanders[AKKTSV'08,KT'08 MM'10]:

When UG instance highly satisfiable and graph is expander, ptime algorithm finds labeling that satisfies 99% of the constraints

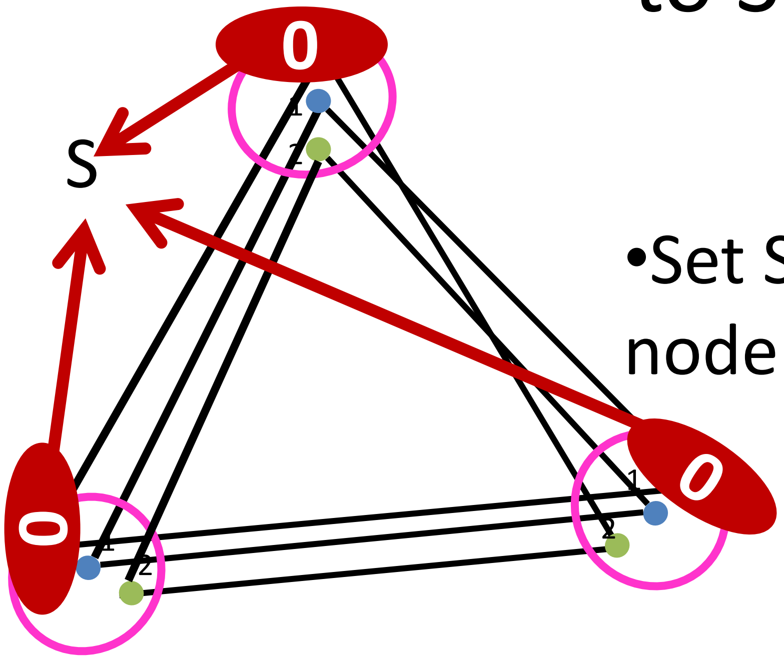
Why Expanders? Expansion of Unique Games and Sparsest Cut

Problem	Best Approximation Algorithm Known	UGC-Hardness
MaxCut	0.878[GW94]	0.878 [KKMO07]
Vertex Cover	2	2-ε [KR06]
Max k-CSP	$\Omega(k/2^k)$[CMM07]	$O(k/2^k)$[ST,AM,GR]

Uniform Sparsest

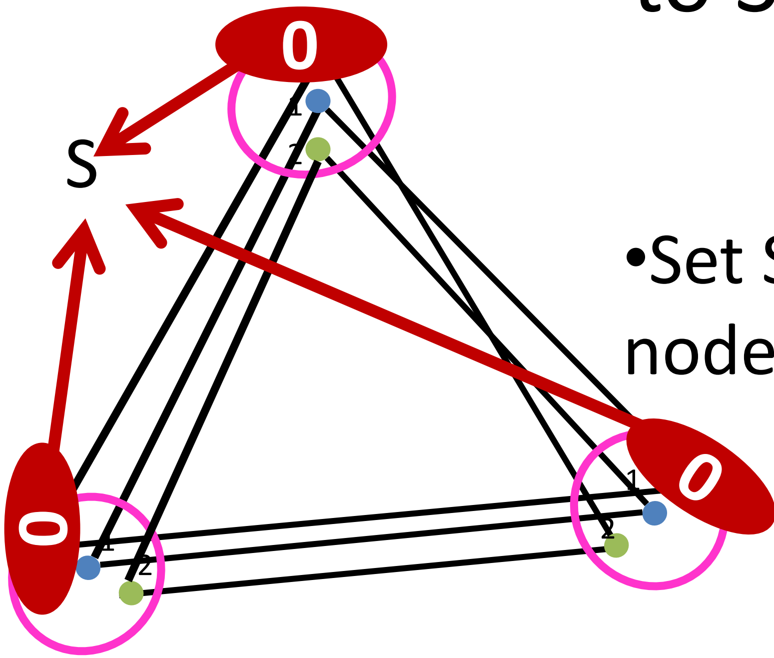
No hardness even assuming UGC unless expansion

Proof with Graph Theory: From Labelings to Spectra



- Set S that contains exactly one "small" node from each node group = labeling

Proof with Graph Theory: From Labelings to Spectra



•Set S that contains exactly one "small" node from each node group = labeling

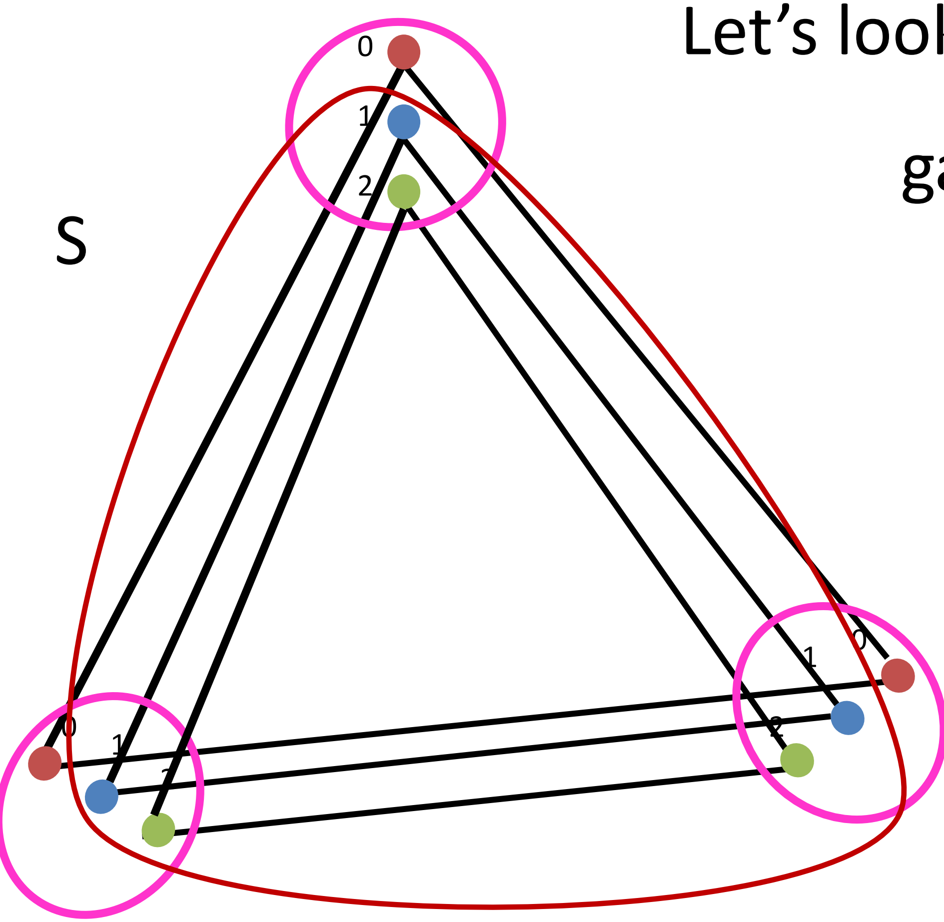
•Corresponds to a cut (S,S').

•Corresponds to a "characteristic vector".

$$\chi_{(0,0,0)} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Proof Intuition: a Perfect Game

S



Let's look at a perfectly satisfiable

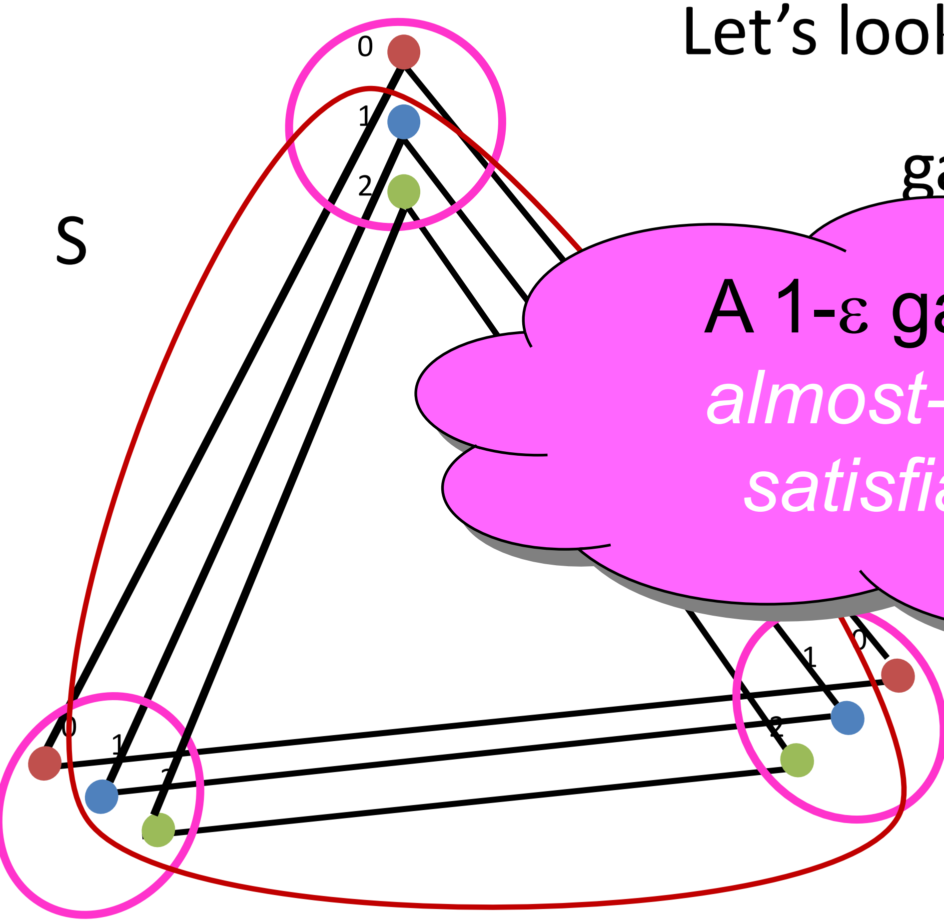
game for intuition…

Graph is disconnected,
it has second eigenvalue λ = d
(in fact, it has k eigenvalues = d)

As mentioned earlier, we can find cuts from those eigenvectors that cut zero edges. (d- λ =0)

If graph G was originally connected, those are the only "sparsest cuts". They correspond to perfect labelings.

Proof Intuition: a Perfect Game



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

A 1-ε game is an *almost-perfectly-satisfiable* one

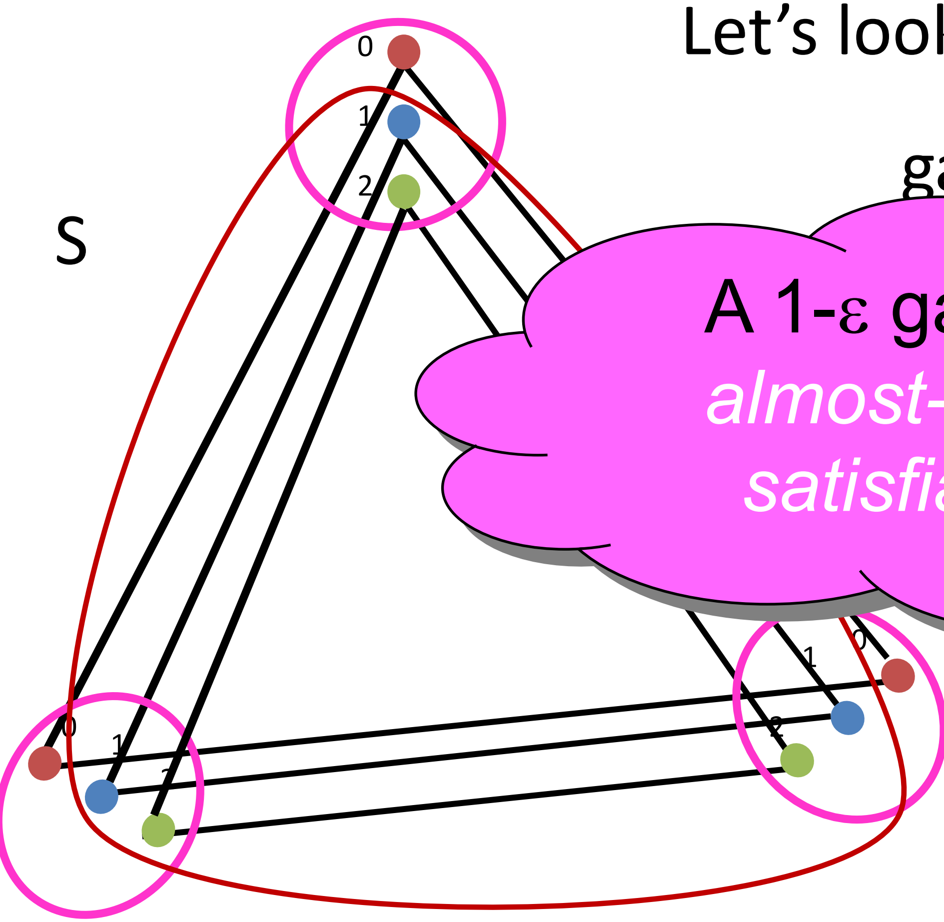
= d

…values = d)

As mentioned earlier, we can find cuts from those eigenvectors that cut zero edges. (d- λ =0)

If graph G was originally connected, those are the only "sparsest cuts". They correspond to perfect labelings.

Proof Intuition: a Perfect Game



Let's look at a perfectly satisfiable

game for intuition…

A 1-ε game is an *almost-perfectly-satisfiable* one

= d

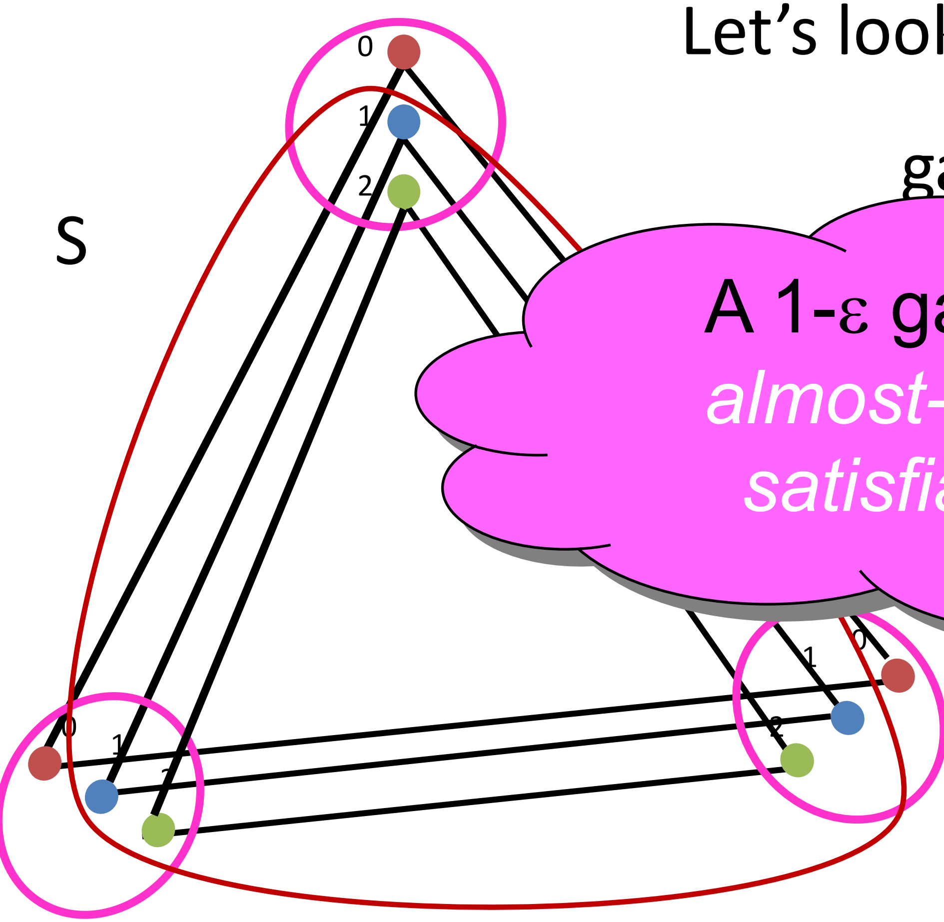
values = d)

As mentioned earlier, we can find cuts from those eigenvectors that cut zero edges. (d- λ =0)

expander

If graph G was originally ~~connected~~, those are the only "sparsest cuts". They correspond to perfect labelings.

Proof Intuition: a Perfect Game



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

A 1-ε game is an *almost-perfectly-satisfiable* one

= d

…values = d)

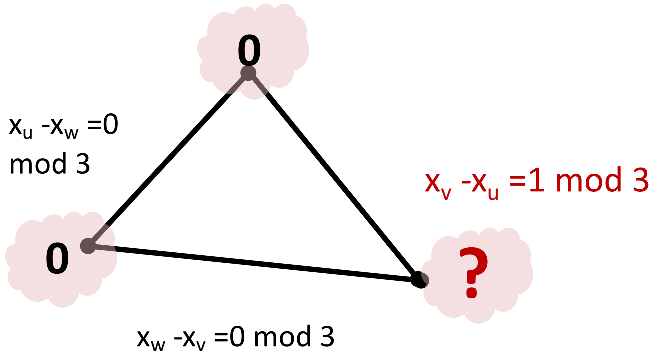
As mentioned earlier, we can find cuts from those eigenvectors that cut zero edges. (d- λ =0)

If graph G was originally ~~connected~~ expander, those are the only "sparsest cuts".

They correspond to almost-perfect labelings

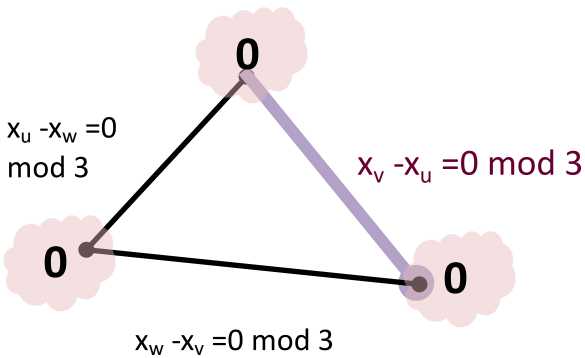
Proof: Reverse Engineering + Graph Spectra

1- ε Game

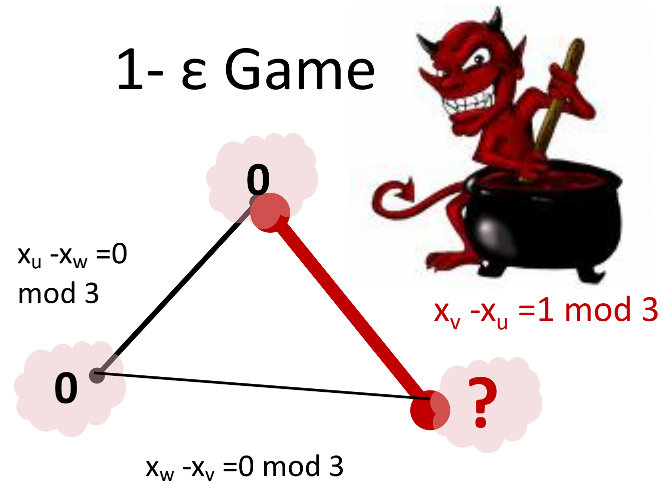


Proof: Reverse Engineering + Graph Spectra

Perfect Game:



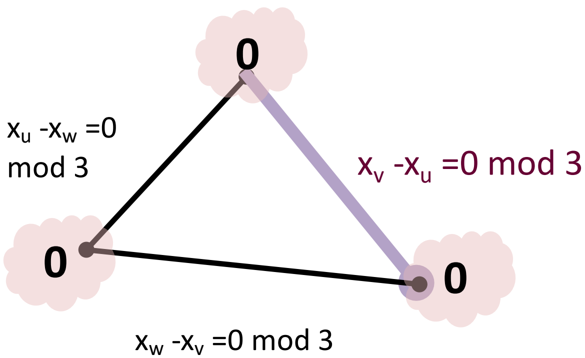
1- ε Game



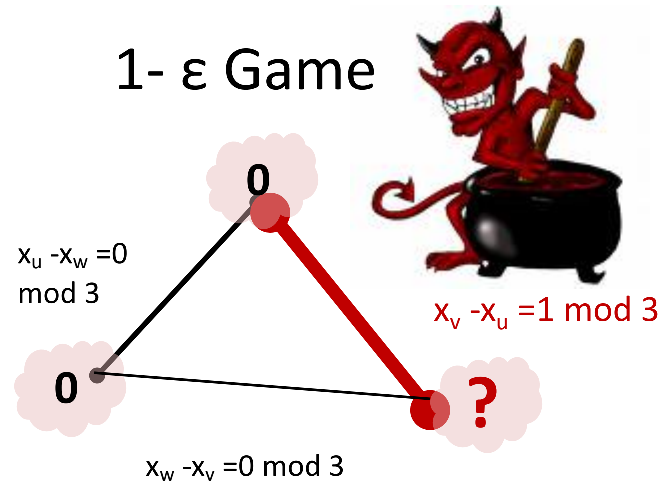
Think of it as “coming from” adversarialy perturbed completely satisfiable game

Proof: Reverse Engineering + Graph Spectra

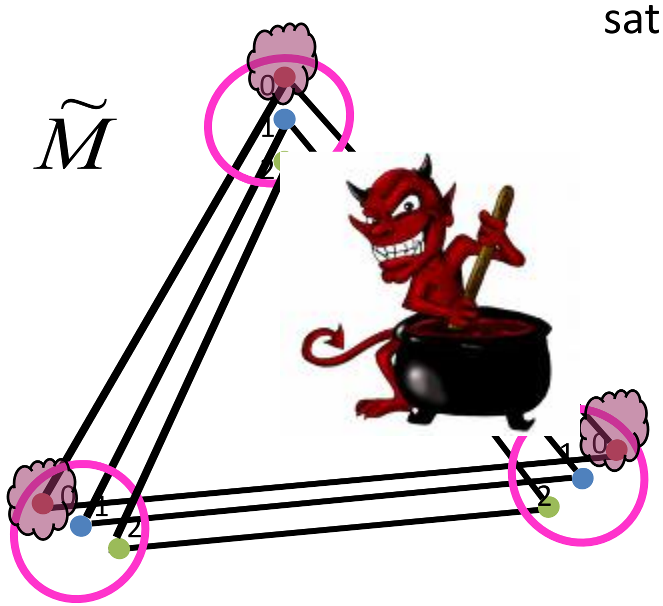
Perfect Game:

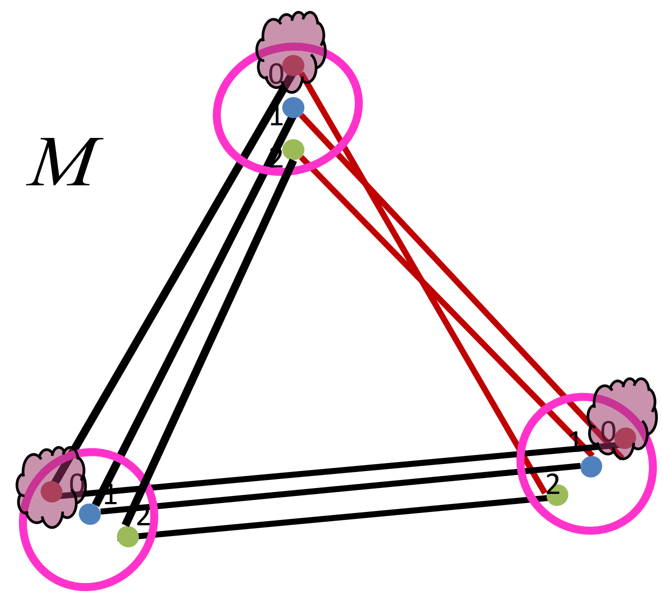


1- ε Game

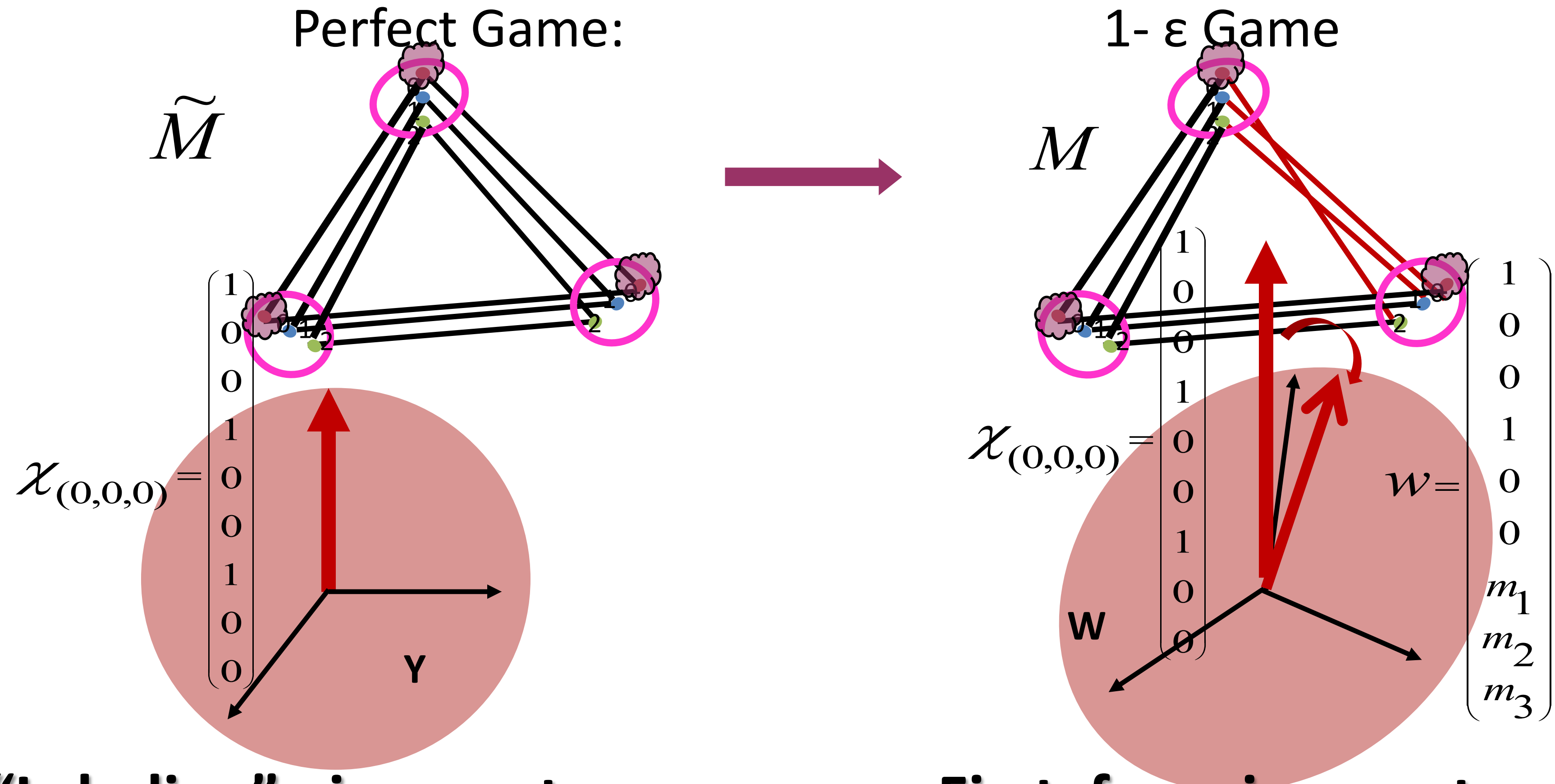


Think of it as “coming from” adversarialy perturbed completely satisfiable game





Proof: Reverse Engineering + Graph Spectra



"Labeling" eigenvectors:

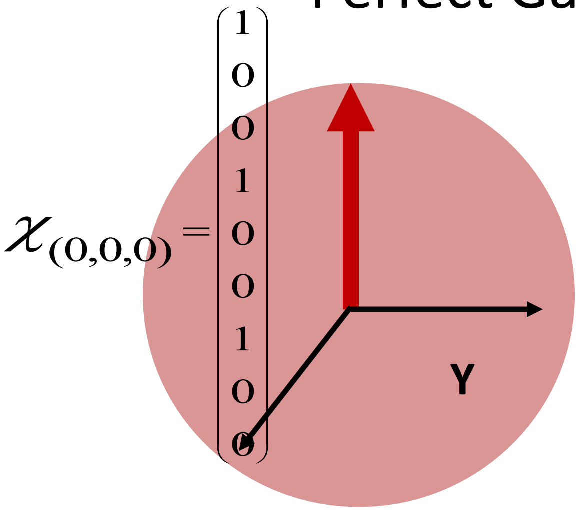
The k-dimensional espace Y of evalues equal to d contains all the information for the best labeling

First few eigenvectors:

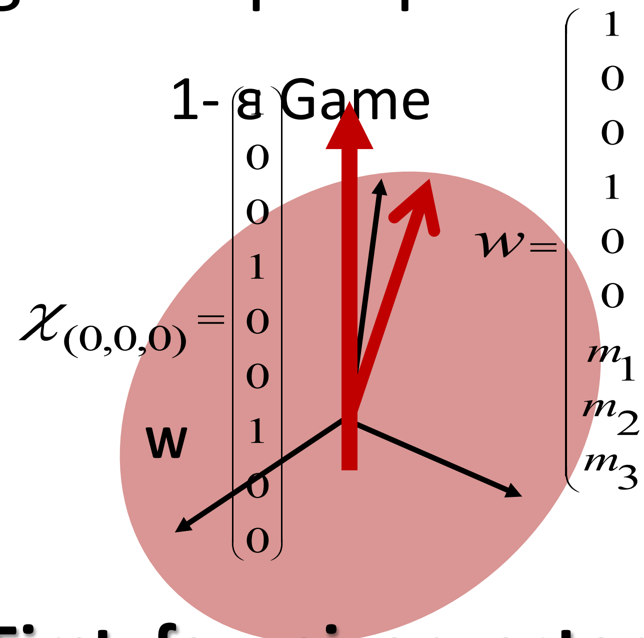
The k "labeling vectors" have large projection onto espace W with evalues >(1- 200ε)d

Proof: Reverse Engineering + Graph Spectra

Perfect Game:



1- ε Game



"Labeling" eigenvectors:

The k-dimensional espace Y of evalues equal to d contains all the information for the best labeling

First few eigenvectors:

The k "labeling vectors" have large projection onto espace W with evalues >(1- 200ε)d

$$for \|\chi\|=1, \quad \chi^T \widetilde{M} \chi = d$$

$$\chi^T M \chi \geq (1-2\varepsilon)d$$

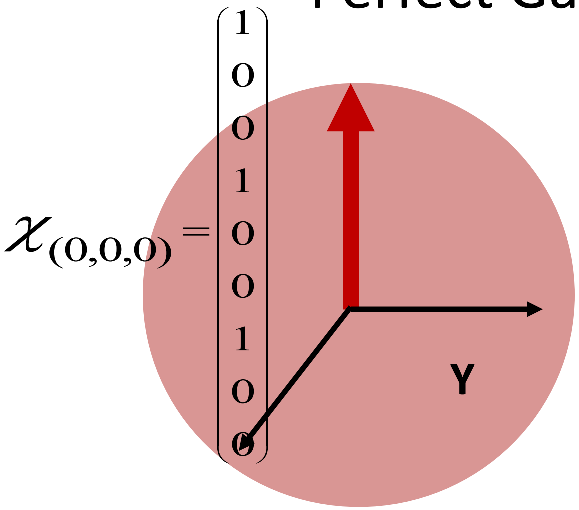
Write: $\chi = \alpha w + \beta w_\perp$

$$(1-2\varepsilon)d \leq \chi^T M \chi = a^2 w^T M w + \beta^2 w_\perp^T M w_\perp$$

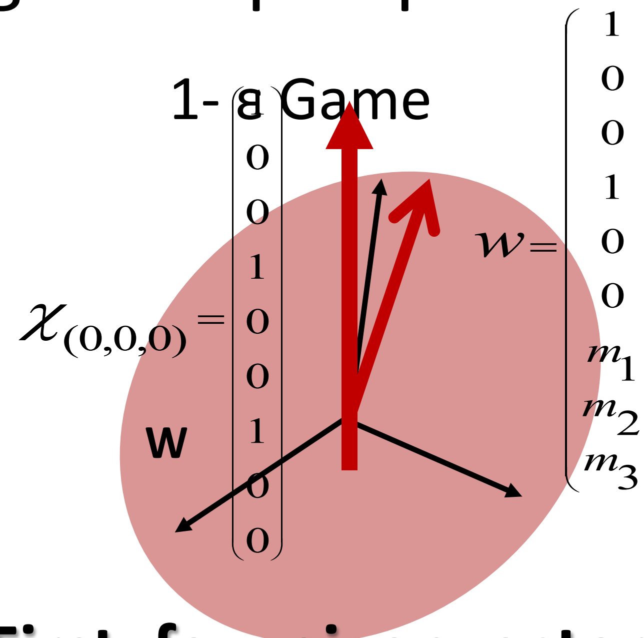
$$\leq a^2 d + \beta^2 (1-200\varepsilon)d \Rightarrow |\beta| \leq \frac{1}{10}$$

Proof: Reverse Engineering + Graph Spectra

Perfect Game:



1- ε Game



"Labeling" eigenvectors:

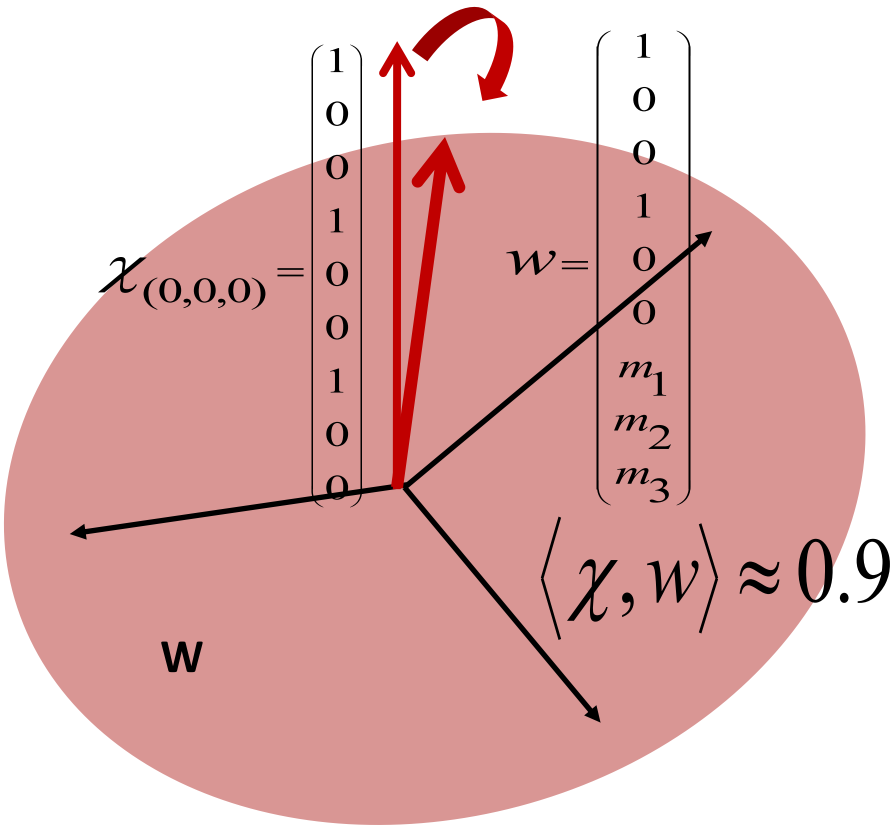
The k-dimensional espace Y of evalues equal to d contains all the information for the best labeling

First few eigenvectors:

The k "labeling vectors" have large projection onto espace W with evalues >(1- 200ε)d

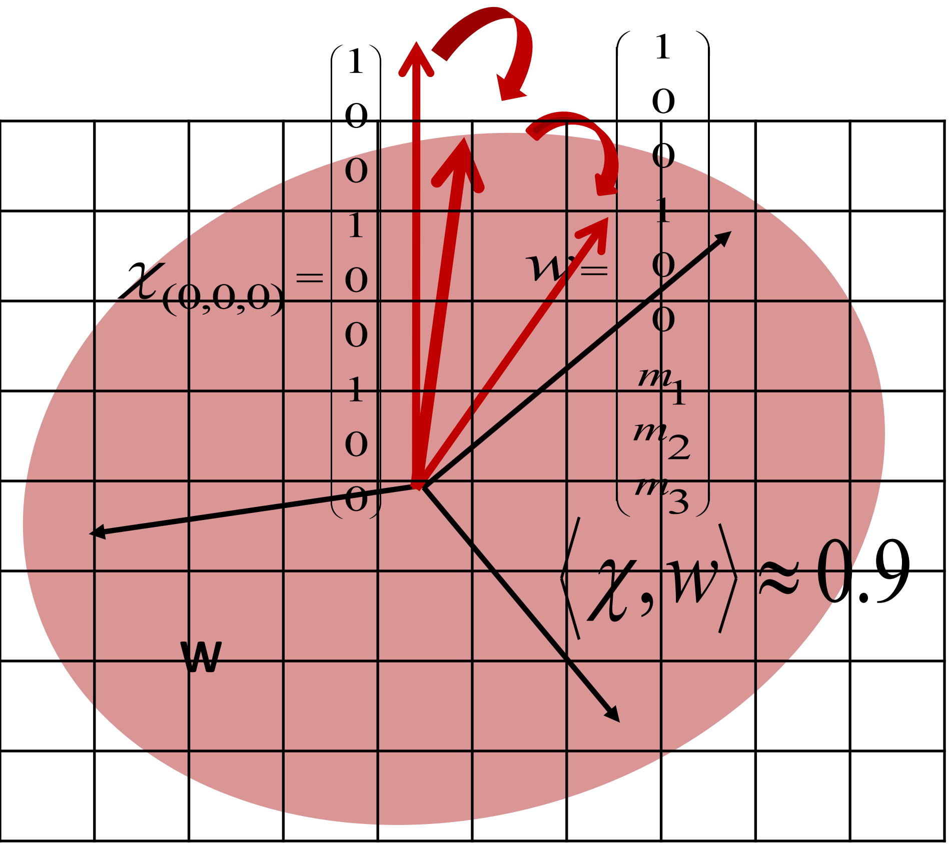
If we knew the projection w of χ then we could just "read off" a good labeling

Searching for a Needle in a Haystack?



But we need to find a particular vector in this whole space W!

Searching for a Needle, but "Efficiently"



But we need to find a particular vector in this whole space W!

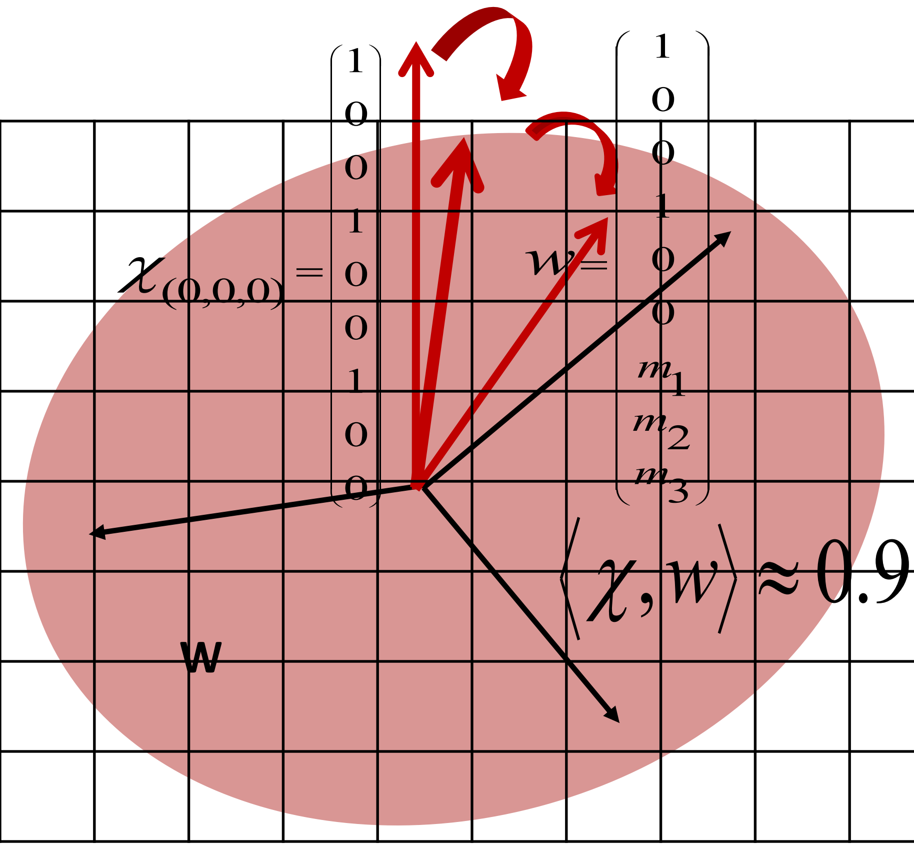
Idea: Discretize the space by net!

One point of the net is close to the vector we want

We find this vector and then "read off" the coordinates

Most blocks have (unique) maximum entry in the position that corresponds to the original value of node u

Searching for a Needle, but "Efficiently"



But we need to find a particular vector in this whole space W!

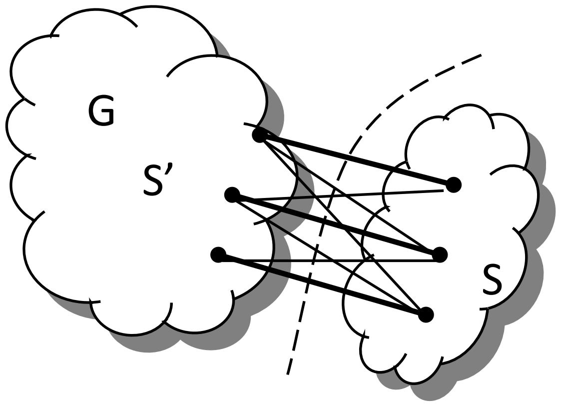
Idea:
Discretize the space by net!

Algorithm runs in time ~ #points in the net
=
exponential in the dimension of eigenspace W

The Dimension of W for Expanders

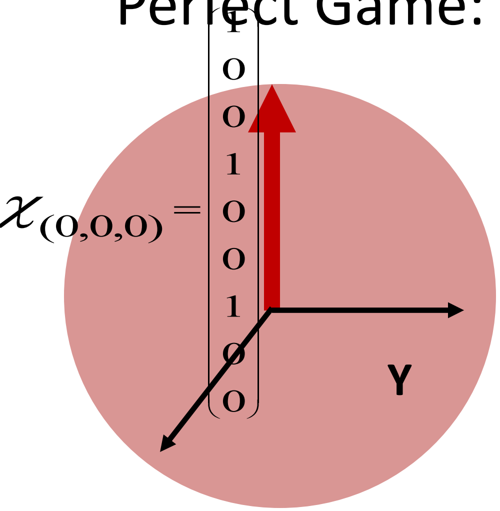
(Spectral Gap)=

$d - \lambda = \gamma d$

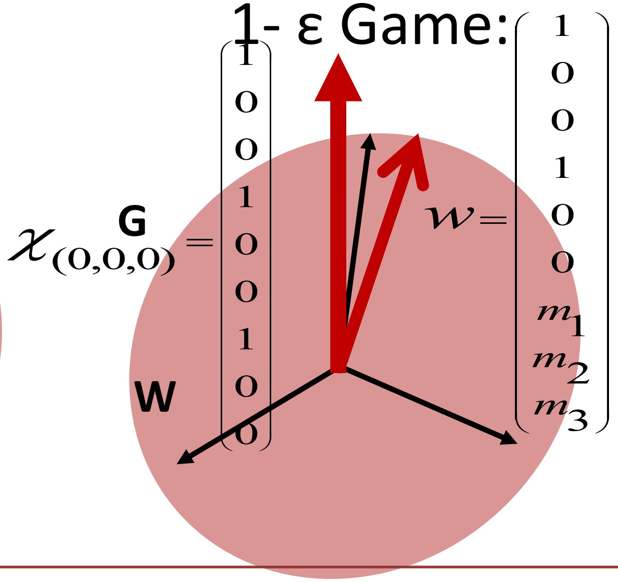


The Dimension of W for Expanders

Perfect Game:

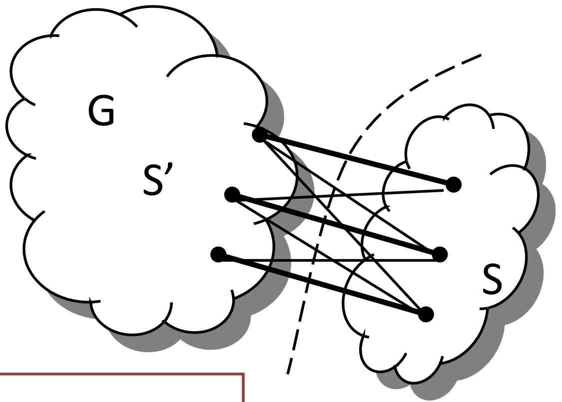


1- ε Game:



(Spectral Gap)=

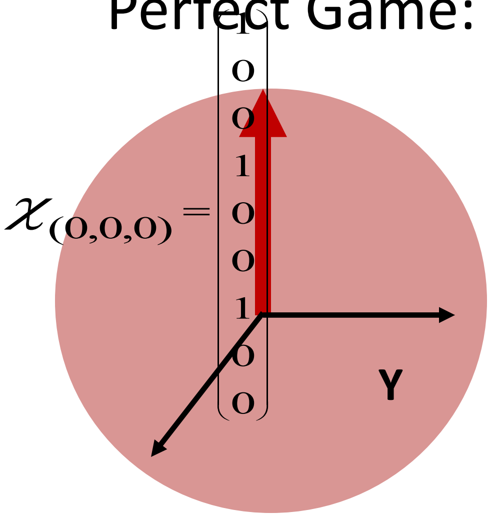
$d - \lambda = \gamma d$



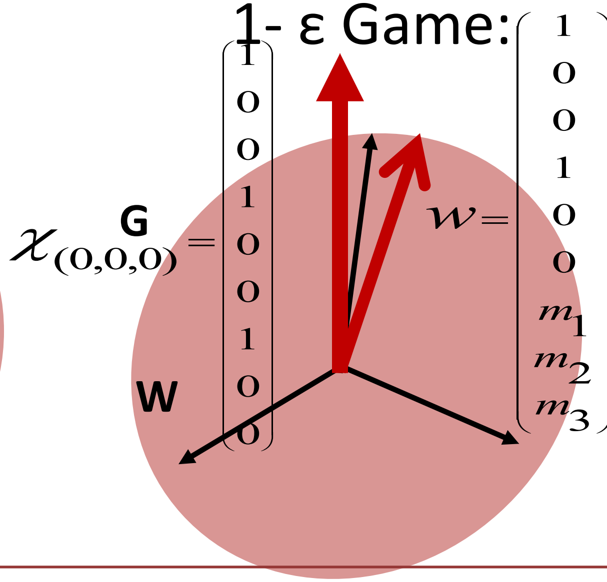
(Spectral gap between Y,Y$_\perp$)= absgap= γd

The Dimension of W for Expanders

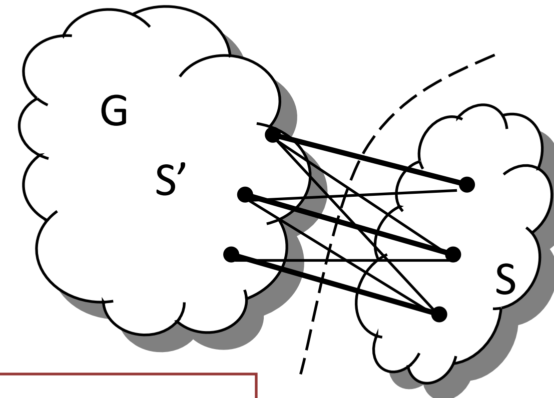
Perfect Game:



1- ε Game:



(Spectral Gap)=
d -λ = γ d



(Spectral gap between Y,Y⊥)= absgap= γd

W is "perturbed analog" of Y

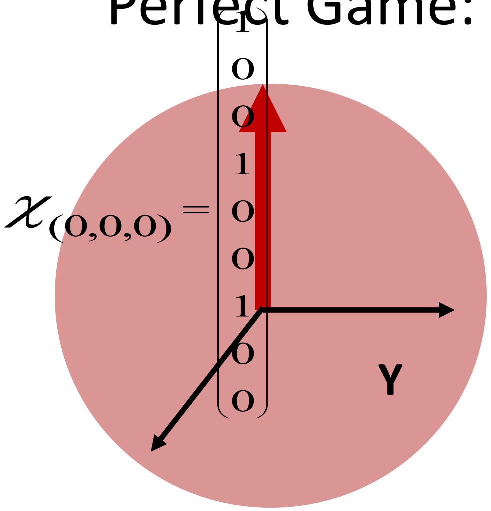
"The sin μ" Theorem [DK'70] : Angle between Y and "perturbed analog of Y" small

Equivalently, we can write every vector w in W as w = α y + β y⊥, y in Y

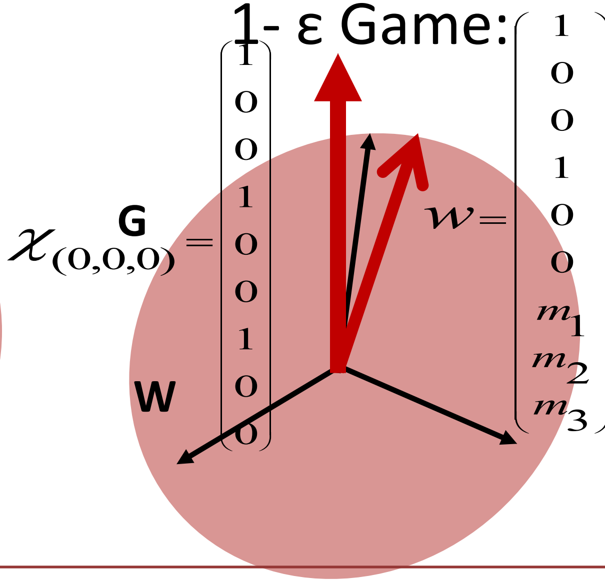
$$|\beta| \leq \frac{||(M - M_\epsilon)w||}{absgap} \leq O(\sqrt{\frac{\epsilon}{\gamma^3}})$$

The Dimension of W for Expanders

Perfect Game:

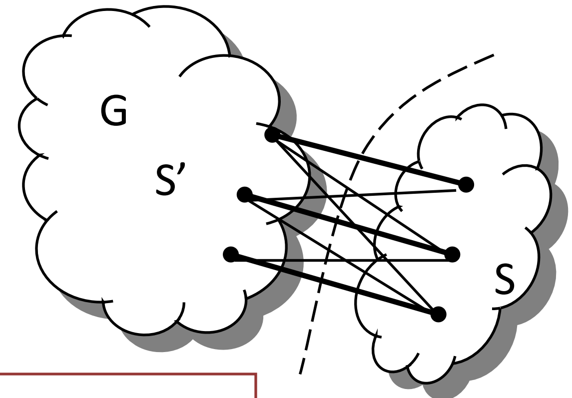


1- ε Game:



(Spectral Gap)=

$d - \lambda = \gamma d$



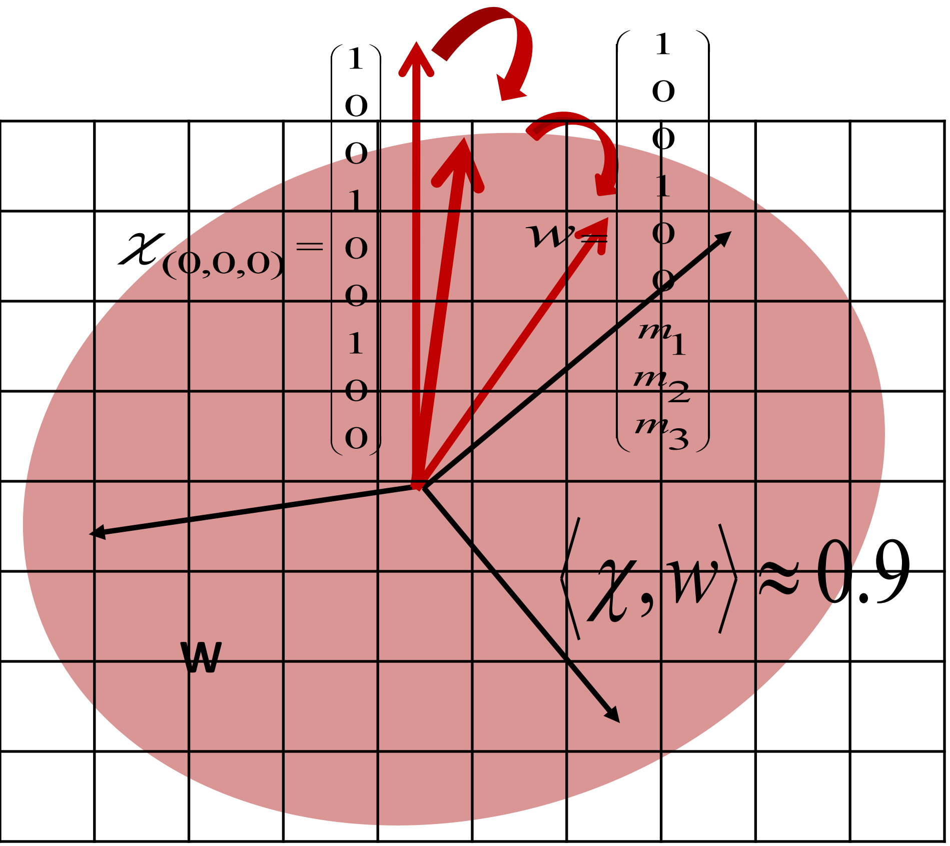
(Spectral gap between $Y, Y_\perp$)= absgap= γd

W is "perturbed analog" of Y

"The sin μ" Theorem [DK'70] : Angle between Y and "perturbed analog of Y" small

⇨ W is close to Y so **dim(W) ≤dim(Y) =k**

A General Algorithm

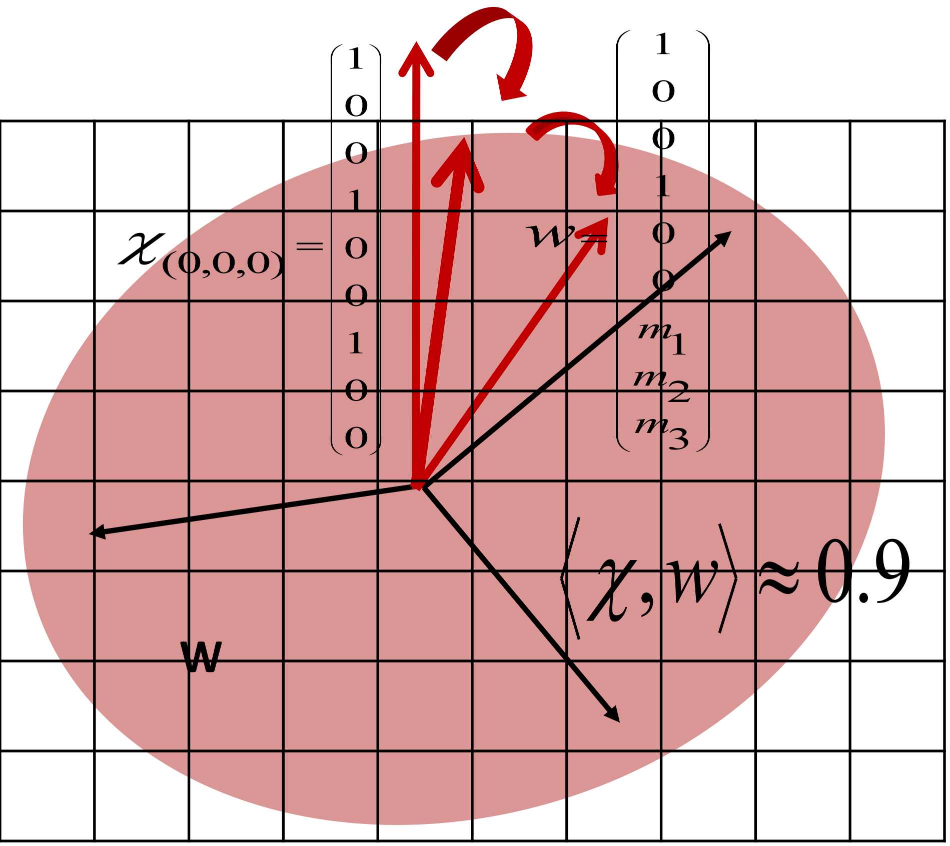


For expanders,
W is close to Y so
dim(W) ≤dim(Y) =k

Running time is
$2^k \approx 2^{\log n} \approx$ poly(n)

Algorithm runs in time ~ #points in the net
=
exponential in the dimension of eigenspace W

A General Algorithm

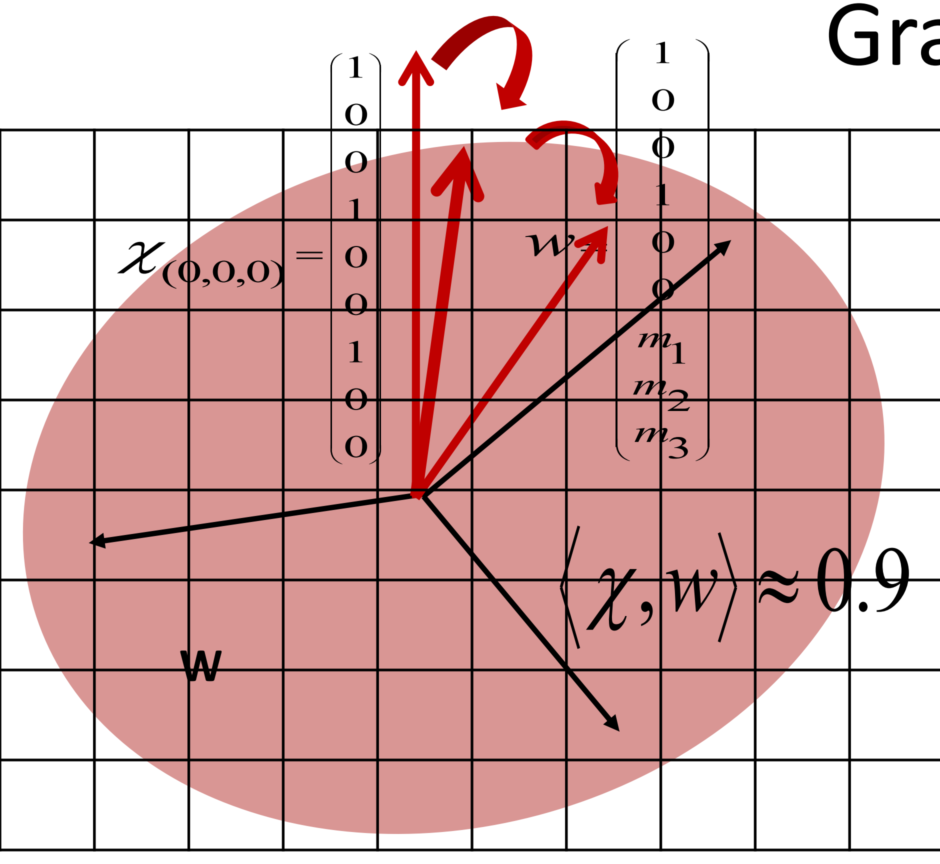


Algorithm runs in time ~ #points in the net

=

exponential in the dimension of eigenspace W

Another Special Case: The "Khot-Vishnoi" Graph



Graph that "cheats" a canonical semidefinite program for UG

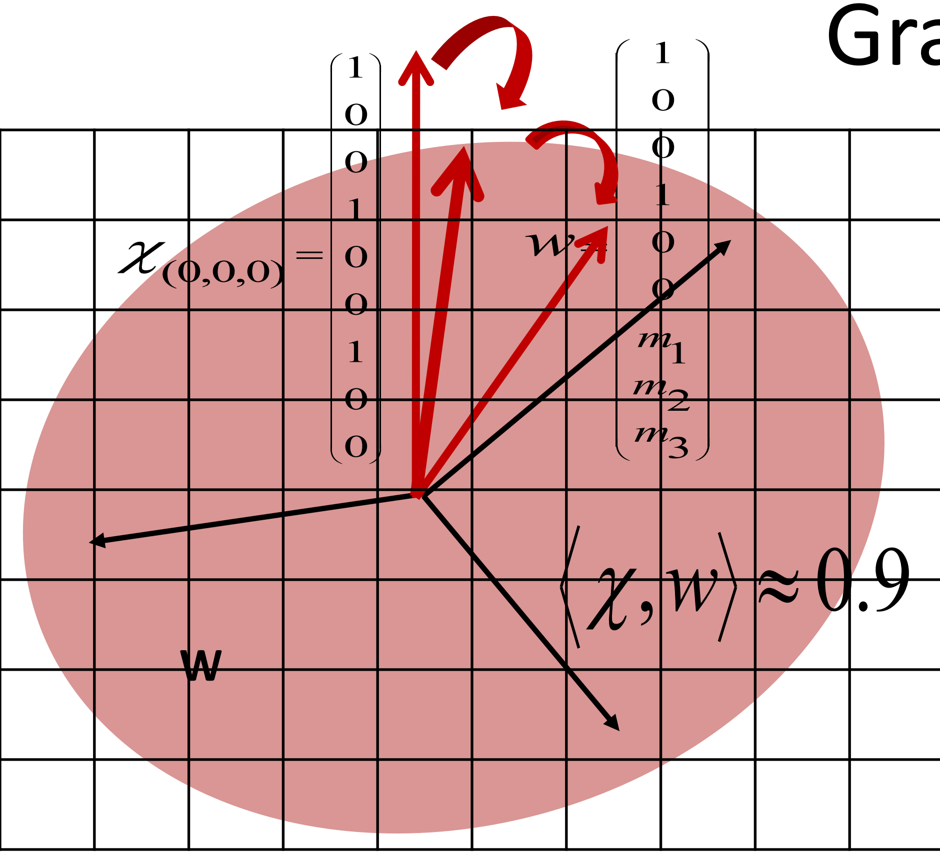
We show: Eigenspace in question has poly-logarithmic dimension

Algorithm runs in time ~ #points in the net

=

exponential in the dimension of eigenspace

Another Special Case: The “Khot-Vishnoi” Graph



Graph that “cheats” a canonical semidefinite program for UG

We show: Eigenspace in question has poly-logarithmic dimension

Algorithm runs in time ~ #points in the net

=

quasi-polynomial

UGC and the Spectrum of General Graphs

- After expanders, we realized that other constraint graphs are easy for UGC.
- How "easy" the graph is, depends on the number of large (close to d) eigenvalues of the adjacency matrix of the label-extended graph.
- Could solve previously "hardest" cases, where all Other techniques failed.
- Essentially only one case left, reflected by the Boolean Hypercube!! (?)

Plan for Today

1. Unique Games Conjecture(UGC)

2. Spectra of Graphs

3. Towards Refuting UGC on almost-all Graphs

4. **Open Questions**

Open Questions

Disprove the Unique Games Conjecture

- Can we argue about UGC on the cube?
- About 2 years ago a group of Quantum Computing Theorists came together and tried to find a quantum algorithm…
- Proved Maximal Inequality on the Cube, failed for UGC.
- What is the quantum complexity of UGC?

THANK YOU!