

# Lecture 16: Subexponential Time Algorithm for Small Set Expansion and Unique Games

# Lecture Outline

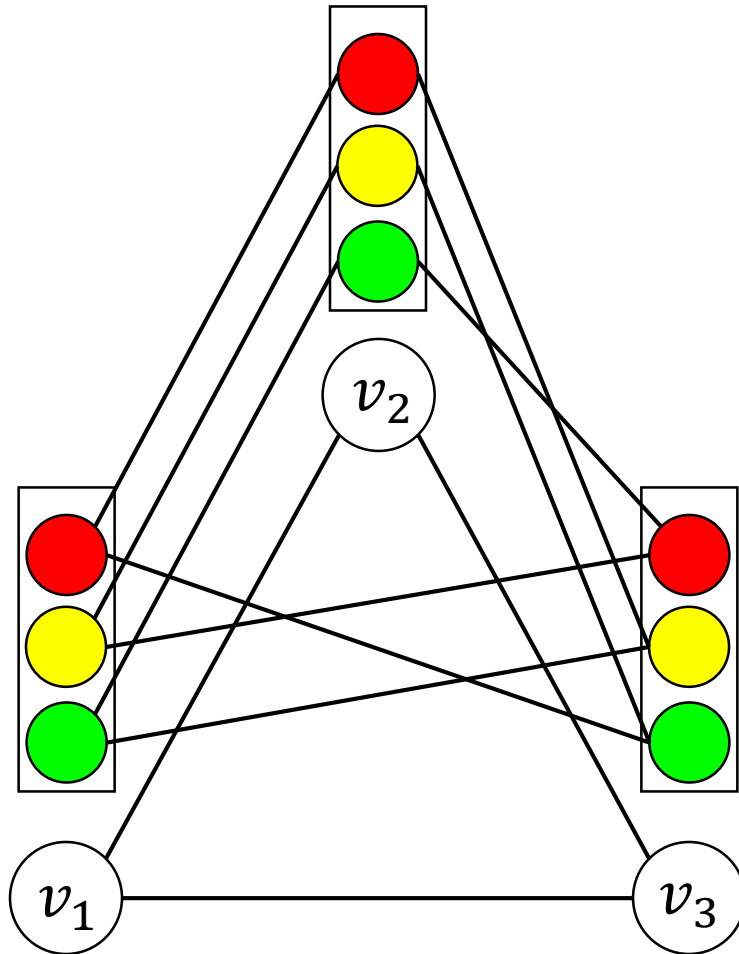
- Part I: Unique Games and Small Set Expansion
- Part II: Cheeger's Inequality and Threshold Rank
- Part III: Low Threshold Rank Case
- Part IV: High Threshold Rank Case
- Part V: Sketch of the Extension to Unique Games
- Part VI: Open Problems

# Part I: Unique Games and Small Set Expansion

# Review: Unique Games Problem

- **Unique Games:** Have a graph  $G$  where we wish to assign each vertex  $v$  of  $G$  a label  $l_v \in [1, k]$  where  $k$  is a large constant.
- For each edge  $(v, w)$  in  $G$ , we have a constraint specifying that  $l_w = \sigma(l_v)$  where  $\sigma$  is a permutation of  $[1, k]$ .
- Goal: Maximize the number of satisfied constraints.

# Unique Games Picture



In this example, we can satisfy two of the three constraints.

# Review: Unique Games Conjecture

- **Unique Games Conjecture (UGC)**: For all  $\epsilon > 0$ , there exists a constant  $k$  such that it is NP-hard to distinguish between the case when at most  $\epsilon$  of the constraints can be satisfied and the case when at least  $(1 - \epsilon)$  of the constraints can be satisfied
- UGC is a central open problem in theoretical computer science
- If true, implies optimal inapproximability results for MAX CUT and other problems

# Expansion of a Graph

- Definition: If  $G$  is a  $d$ -regular graph on a set of  $n$  vertices  $V$  and  $S \subseteq V$  is a subset of size at most  $\frac{n}{2}$ , the **expansion**  $\Phi_G(S)$  of  $S$  is  $\Phi_G(S) = \frac{|E(S, V \setminus S)|}{d|S|}$  where  $E(S, V \setminus S)$  is the set of edges between  $S$  and  $V \setminus S$ .
- Definition: the **expansion** of a graph  $G$  is  $\Phi_G = \min_{S: 0 < |S| \leq \frac{n}{2}} \Phi_G(S)$

# Small Set Expansion Problem

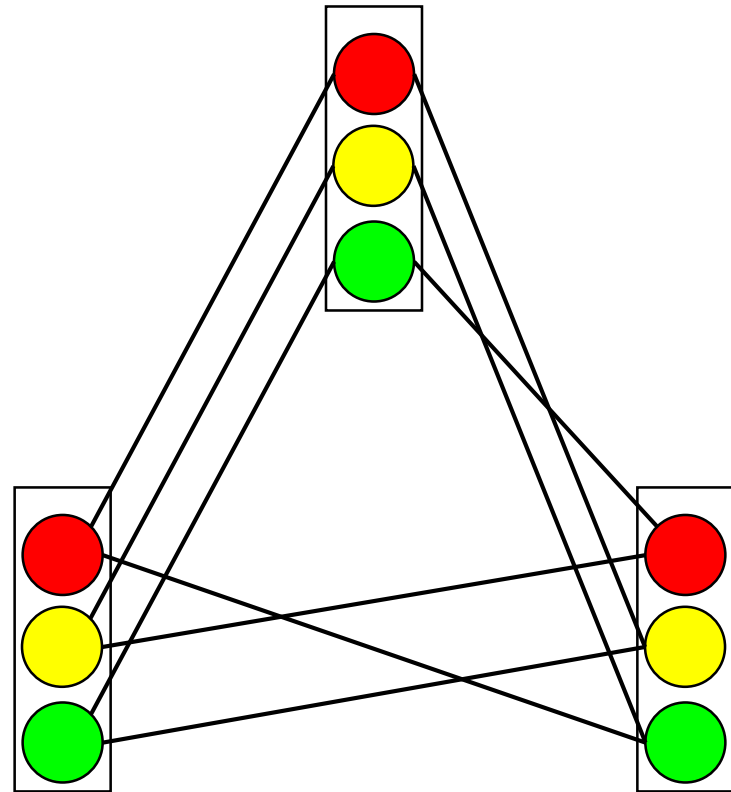
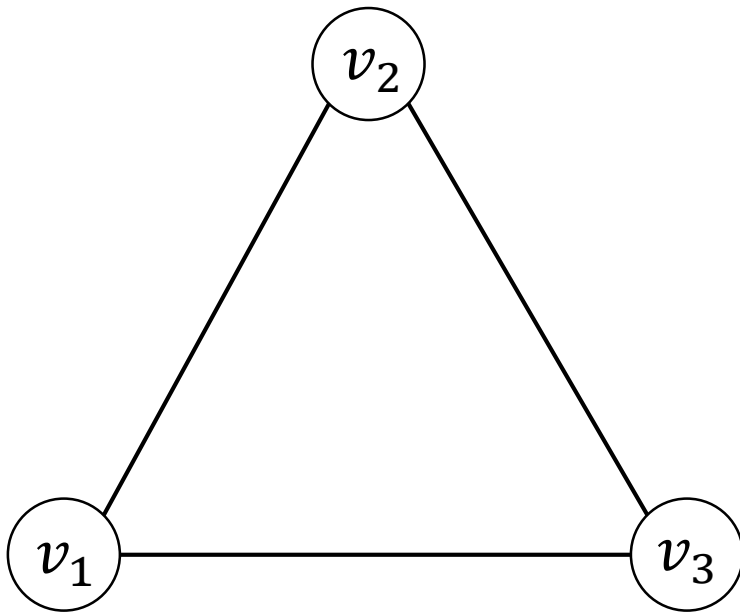
- What if we want to restrict ourselves to subsets of a certain small density  $\delta$ ?
- Definition: Define  $\Phi_G(\delta) = \min_{S: \frac{|S|}{n} = \delta} \Phi_G(S)$
- **Gap small set expansion problem (SSE)**: Given small constants  $\eta, \delta > 0$  and a graph  $G$  on  $n$  vertices, distinguish whether  $\Phi_G(\delta) \geq 1 - \eta$  or  $\Phi_G(\delta) \leq \eta$ .



# Relation between UG and SSE

- One direction: Given a unique games instance  $G$  where each vertex is involved in the same number of constraints, we can form the graph  $\widehat{G}$  whose vertices correspond to pairs  $(v, i)$  where  $v \in V(G)$  and  $i \in [1, k]$  and whose edges correspond to satisfied constraints.
- Call  $\widehat{G}$  the **label extended graph** of  $G$
- A solution to the unique games instance satisfying almost all constraints gives a subset of vertices of density  $\delta = \frac{1}{k}$  with small expansion.

# Label Extended Graph Picture



# Relation between UG and SSE

- Unfortunately, there could be other sets of the same size which have small expansion.
- For example, we could take a subset of  $n/k$  vertices  $\{v_j\}$  and then take all of the pairs  $(v_j, i)$ .
- Still, this suggests that UG and SSE are closely related.

# Reduction from SSE to UG

- Theorem [RS10]: There is a reduction from SSE to UG.
- Idea: Consider the following game with a verifier and two provers. Given a  $d$ -regular graph  $G$ :
  1. the verifier chooses  $k = \left\lceil \frac{1}{\delta} \right\rceil$  edges  $(u_1, v_1), \dots, (u_k, v_k)$  at random, sends the permuted set  $(u_1, \dots, u_k)$  to one prover, and sends the permuted set  $(v_1, \dots, v_k)$  to the other prover.
  2. Each prover chooses one vertex from their set
  3. The provers win if they selected some edge  $(u_i, v_i)$

# Unique Games Instance

- This corresponds to a unique games instance:
- The vertices are possible subsets of  $k$  vertices sent to a prover.
- Each randomly chosen set of  $k$  edges  $(u_1, v_1), \dots, (u_k, v_k)$  gives a constraint between the vertices  $(u_1, \dots, u_k)$  and  $(v_1, \dots, v_k)$

# Unique Games Partial Strategy

- Key idea: If there is a set  $S$  of size  $\delta n$  which has small expansion, the provers can use the following **partial strategy**:
- If they are given a set which contains precisely one vertex in  $S$ , take that vertex. Otherwise, do not answer.
- Because of the small expansion of  $S$ , when one prover answers, with high probability the other prover answers as well and they will be correct.

# From Partial to Full Strategies

- In a unique game, we must select a choice for every vertex.
- Idea: Play the game multiple times independently and allow the provers to choose one game which they will play. To win, the provers must choose the same game and win it.
- Repeating the game a constant number of times, with high probability the provers' partial strategy will work for at least one game (and they can choose the first such game)

# Soundness

- Also need to show that this unique game is sound, i.e. if there is no set of size  $\delta n$  with small expansion then the provers have no strategy to succeed.
- We won't discuss this here, see [RS10] for details.



# Subexponential Time Algorithm

- Theorem [ABS10]: There is an absolute constant  $c$  such that
  1. There is a  $2^{O(kn^\epsilon)}$  time algorithm that takes a unique games instance with alphabet size  $k$  which has a solution satisfying  $1 - \epsilon^c$  of its constraints and outputs a solution satisfying  $1 - \epsilon$  of its constraints.
  2. There is a  $2^{O\left(\frac{n^\epsilon}{\delta}\right)}$  time algorithm that takes a  $d$ -regular graph  $G$  such that  $\Phi_G(\delta) \leq \epsilon^c$  and outputs a set of vertices  $S'$  such that  $|S'| \leq \delta n$  and  $\Phi_G(S') \leq \epsilon$

# Subexponential Time Algorithm

- For most of the remainder of this lecture, we will focus on the subexponential time algorithm for SSE
- The subexponential time algorithm for UG is an extension of this algorithm.

# Part II: Cheeger's Inequality and Threshold Rank

# Review: Cheeger's Inequality

- **Cheeger's inequality:** Let  $G$  be a  $d$ -regular graph, let  $A$  be its adjacency matrix, and let  $1 = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  be the eigenvalues of  $\frac{A}{d}$ . Then  $\frac{1-\lambda_2}{2} \leq \Phi_G \leq \sqrt{2(1-\lambda_2)}$
- The subexponential time algorithm for SSE can be thought of as an analogue of Cheeger's inequality which looks at many top eigenvalues, not just the second.

# Easy Direction of Cheeger's Inequality

- Proof that  $\Phi_G \geq \frac{1-\lambda_2}{2}$ :
- Let  $S$  be the subset of size  $\leq \frac{n}{2}$  such that  $\Phi_G(S) = \frac{|E(S, V \setminus S)|}{d|S|} = \Phi_G$  and take  $v$  to be the vector  $v_i = n - |S|$  if  $i \in S$  and  $v_i = -|S|$  if  $i \notin S$ . Note that  $\|v\|^2 = n(n - |S|)|S|$
- $v \perp 1$ , so  $\lambda_2 \geq \frac{v^T \left(\frac{A}{d}\right) v}{\|v\|^2}$

# Calculation

- $v^T \begin{pmatrix} A \\ d \end{pmatrix} v = \frac{2}{d} \sum_{(i,j) \in E(G)} v_i v_j$
- If  $|E(S, V \setminus S)|$  were 0, we would have that  $v^T \begin{pmatrix} A \\ d \end{pmatrix} v = |S|(n - |S|)^2 + (n - |S|)|S|^2 = \|v\|^2$
- Each edge between  $S$  and  $S \setminus V$  reduces the number of edges within  $S$  and the number of edges within  $S \setminus V$  by  $\frac{1}{2}$ , which creates a difference of  $\frac{1}{d} (-2(n - |S|)|S| - |S|^2 - (n - |S|)^2) = -\frac{n^2}{d}$

# Calculation Continued

$$\begin{aligned} \mathbf{v}^T \left( \frac{A}{d} \right) \mathbf{v} &= \|\mathbf{v}\|^2 - \frac{n^2 |E(S, V \setminus S)|}{d} = \|\mathbf{v}\|^2 - \frac{n}{n - |S|} \cdot \\ \frac{2n(n - |S|)|S||E(S, V \setminus S)|}{d|S|} &= \|\mathbf{v}\|^2 \left( 1 - \frac{n}{n - |S|} \Phi_G(S) \right) \end{aligned}$$

- $\mathbf{v} \perp \mathbf{1}$ , so  $\lambda_2 \geq 1 - \frac{n}{n - |S|} \Phi_G(S) \geq 1 - 2\Phi_G(S)$

# Hard Direction of Cheeger's Inequality

- Want to show that  $\Phi_G \leq \sqrt{2(1 - \lambda_2)}$
- Proof idea: Let  $v$  be the eigenvector with eigenvalue  $\lambda_2$ . Show that there exists a cutoff value  $c$  such that if we take  $S = \{i: v_i \leq c\}$  then  $\Phi_G(S) \leq \sqrt{2(1 - \lambda_2)}$



# Threshold Rank

- **Definition:** Let  $G$  be a  $d$ -regular graph, let  $A$  be its adjacency matrix, and let  $1 = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  be the eigenvalues of  $\frac{A}{d}$ . Given  $\tau \in [0,1)$ , the **threshold rank** is defined to be  $rank_\tau(G) = |\{i: \lambda_i > \tau\}|$
- Example 1:  $\lambda_0$  is the usual rank of  $A$
- Example 2: For all  $\tau > 0$ , with high probability  $rank_\tau(G) = 1$  for a random graph if there are sufficiently many vertices.

# Theorem Cases

- Given a small  $\eta > 0$ , either  $\text{rank}_{1-\eta}(G) \leq n^\epsilon$  or  $\text{rank}_{1-\eta}(G) > n^\epsilon$
- Case I (analogue of the easy direction of Cheeger's inequality): For any set  $S$  with small expansion, there is a corresponding vector  $v$  which is close to being in the subspace of eigenvectors with eigenvalue  $> 1 - \eta$ . Since this subspace has dimension  $\leq n^\epsilon$ , we can search for an approximation to  $v$  in subexponential time.

# Theorem Cases

- Case II (analogue of the hard direction of Cheeger's inequality): If  $\text{rank}_{1-\eta}(G) > n^\epsilon$  then we can find a set of vertices  $S$  of size at most  $\delta n$  (but it could be much smaller) which has small expansion.

# Part III: Low Threshold Rank Case

# Low Threshold Rank Case

- Theorem 2.2 of [ABS10]: There is a  $2^{O(\text{rank}_{1-\eta}(G))} \text{poly}(n)$  time algorithm which given  $\epsilon > 0$  and a graph  $G$  containing a set  $S$  such that  $\Phi_G(S) \leq \epsilon$ , outputs a sequence of sets, one of which has symmetric difference of size at most  $8(\epsilon/\eta)|S|$  with the set  $S$

# Low Threshold Rank Case

- Let  $U$  be the subspace of eigenvectors of  $\frac{A}{d}$  with eigenvalue  $> 1 - \eta$
- Let  $S$  be a set of vertices of size  $\delta n$  such that  $\Phi_G(S) \leq \epsilon$ . Take  $v$  to be the same vector as before except normalized so that  $\|v\| = 1$ . In other words,  $v_i = \frac{(n-|S|)}{\sqrt{n|S|(n-|S|)}}$  if  $i \in S$  and  $v_i = \frac{-|S|}{\sqrt{n|S|(n-|S|)}}$  if  $i \notin S$
- Want to find a vector  $v'$  in  $U$  which  $v$  is close to.

# Low Threshold Rank Case

- Write  $v = \sqrt{1 - \gamma} u + \sqrt{\gamma} u^\perp$  where  $u \in U$  and  $u^\perp \in U^\perp$ .
- $v^T \begin{pmatrix} A \\ d \end{pmatrix} v = (1 - \frac{n}{n - |S|} \epsilon)$
- $v^T \begin{pmatrix} A \\ d \end{pmatrix} v = (1 - \gamma) u^T \begin{pmatrix} A \\ d \end{pmatrix} u + \gamma (u^\perp)^T \begin{pmatrix} A \\ d \end{pmatrix} u \leq 1 - \gamma + \gamma(1 - \eta) = 1 - \gamma\eta$
- Thus,  $\gamma \leq \frac{n}{n - |S|} \epsilon / \eta$ .

# Low Threshold Rank Case

- We can find a vector  $v'$  such that  $d^2(v, v') \leq \frac{2n\epsilon}{\eta(n-|S|)}$  using **epsilon nets** (see next few slides)
- Once we have such a vector  $v'$ , we can obtain a set  $S'$  by taking  $i \in S'$  if  $v'_i \geq \frac{\frac{n}{2} - |S|}{\sqrt{n(n-|S|)|S|}}$  and taking  $i \notin S'$  if  $v'_i < \frac{\frac{n}{2} - |S|}{\sqrt{n(n-|S|)|S|}}$

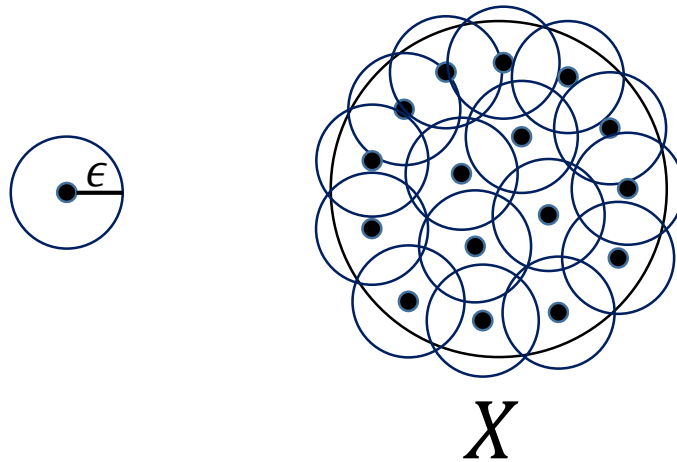


# Low Threshold Rank Case

- Each coordinate where  $S, S'$  differ contributes at least  $\frac{n}{4(n-|S|)|S|}$  to  $d^2(v, v')$
- $d^2(v, v') \leq \frac{2n\epsilon}{\eta(n-|S|)}$  so there are at most  $\frac{8\epsilon}{\eta} |S|$  such coordinates.

# Epsilon Nets

- Definition: An  $\epsilon$ -net for a set  $X$  is a set of points  $\{p_i\} \subseteq X$  such that  $\forall x \in X \exists i: d(x, p_i) \leq \epsilon$



# Epsilon Net Existence

- Lemma: For any set  $X$ , there is an epsilon net for  $X$  of size at most  $\frac{V(X+B_{\epsilon/2})}{V(B_{\epsilon/2})}$  where  $B_{\epsilon/2}$  is the ball of radius  $\epsilon/2$  and  $X + B_{\epsilon/2} = \{p: \exists x \in X: d(p, x) \leq \epsilon/2\}$
- Proof: We can construct our  $\epsilon$ -net greedily. As long as there is a point  $x \in X$  which is not yet covered, take  $p_{i+1} = x$ . When we are done, the balls of radius  $\epsilon/2$  around each  $p_i$  have zero intersection so there are at most  $\frac{V(X+B_{\epsilon/2})}{V(B_{\epsilon/2})}$  points in our  $\epsilon$ -net.

# Finding Epsilon Nets

- How can we find  $\epsilon$ -nets?
- If we can sample  $X + B_{\epsilon/2}$  at random, the probabilistic method gives us a  $2\epsilon$ -net with high probability (which is just as good as  $\epsilon$  is arbitrary).
- In particular, choose each point by sampling a point  $q'_i$  randomly from  $X + B_{\epsilon/2}$  and then locating an arbitrary point  $q_i \in X$  which is within distance  $\frac{\epsilon}{2}$  of  $p'_i$ .

# Finding Epsilon Nets Continued

- Let  $\{p_i\}$  be an arbitrary  $\epsilon$ -net of  $X$  of size  $m$ . If  $\forall i \exists j: d(q'_j, p_i) \leq \frac{\epsilon}{2}$  then  $q_j$  will be within distance  $\epsilon$  of  $p_i$  and thus the ball of radius  $2\epsilon$  around  $q_j$  contains the ball of radius  $\epsilon$  around  $p_i$  and we have a  $2\epsilon$ -net
- With high probability, sampling  $O\left(\frac{V(X+B_{\epsilon/2})}{V(B_{\epsilon/2})} \log m\right)$  points is sufficient

# Summary

- Upshot: We can find and enumerate over an  $\epsilon$ -net for the unit ball in dimension  $d = \text{rank}_{1-\eta}(G)$  in time  $\left(\frac{2}{\epsilon}\right)^{O(d)}$  which is  $2^{O(d \log(\epsilon))}$

# Finding $v'$

- If  $v$  is the vector we wish to approximate and we know  $v$  is distance at most  $\epsilon' = \sqrt{\frac{n\epsilon}{\eta(n-|S|)}}$  from  $U$  then take an  $\epsilon'$ -net  $\{p_i\}$  of  $U$ . Letting  $u \in U$  be the closest point in  $U$  to  $v$ , take  $v'$  to be an arbitrary  $p_i$  of distance  $\leq \epsilon'$  from  $u$ .
- $d^2(v, v') \leq 2(\epsilon')^2 = \frac{2n\epsilon}{\eta(n-|S|)}$  because  $u - v$  and  $v' - u$  are orthogonal.

# Part IV: High Threshold Rank Case



# High Threshold Rank Case

- Theorem 2.3 of [ABS10]: Let  $G$  be a regular graph on  $n$  vertices such that  $\text{rank}_{1-\eta}(G) \geq n^{100\eta/\gamma}$ . Then there exists a set of vertices  $S$  of size at most  $n^{1-\eta/\gamma}$  such that  $\Phi_G(S) \leq \sqrt{\gamma}$ . Moreover,  $S$  is the level set of a column of  $\left(\frac{Id}{2} + \frac{A}{2d}\right)^j$  for some  $j \leq O(\log n)$

# High Threshold Case Intuition

- Want to show that  $G$  cannot satisfy both:
  1.  $\frac{A}{d}$  has many large eigenvalues
  2. All sets  $S$  of size at most  $\delta n$  have expansion which is not too small

- Will analyze  $tr \left( \left( \frac{A}{2d} + \frac{Id}{2} \right)^k \right)$

- Idea #1: If  $\frac{A}{d}$  has  $m$  eigenvalues which are all at least  $1 - \eta$  then  $tr \left( \left( \frac{A}{2d} + \frac{Id}{2} \right)^k \right) \geq m \left( 1 - \frac{\eta}{2} \right)^k$

# High Threshold Case Intuition

- Idea #2: Applying  $\left(\frac{A}{2d} + \frac{Id}{2}\right)^{\frac{k}{2}}$  is equivalent to taking  $\frac{k}{2}$  steps in a **lazy random walk** where we stay put with probability  $\frac{1}{2}$  and take a step with probability  $\frac{1}{2}$ .

$\frac{1}{n} \text{tr} \left( \left( \frac{A}{2d} + \frac{Id}{2} \right)^k \right) = \frac{1}{n} \sum_i e_i^T \left( \frac{A}{2d} + \frac{Id}{2} \right)^{\frac{k}{2}} \left( \frac{A}{2d} + \frac{Id}{2} \right)^{\frac{k}{2}} e_i$  is the probability that two lazy random walks of  $\frac{k}{2}$  steps collide.

# High Threshold Case Intuition

- Intuition: If every set of size of vertices of size  $\leq \delta n$  expands by at least  $\epsilon$ , then we expect a lazy random walk of length  $\frac{k}{2}$  to reach a set of size at least  $\min\{(1 + \epsilon)^{\frac{k}{4}}, \delta n\}$ . Thus, we expect  $n$  times the collision probability to be at most  $\max\left\{\frac{n}{(1+\epsilon)^{\frac{k}{4}}}, \frac{1}{\delta}\right\}$
- If so, choosing the right value of  $k$  gives a contradiction.

# High Threshold Case Intuition

- The intuition is essentially correct, but considerable technical work is required to obtain a proof.
- For details, see [ABS10]

# Part V: Sketch of the Extension to Unique Games

# Extension to Unique Games

- How can this algorithm be extended to unique games?
- If  $G$  is a unique games instance with low threshold rank, consider the label extended graph  $\hat{G}$ .
- It can be shown that  $\hat{G}$  has relatively low threshold rank. Letting  $S$  be the true solution, we can apply the subexponential time algorithm to find a subset  $S' \approx S$ , which lets us recover an almost optimal solution.

# High Threshold Rank Case

- What if  $G$  has high threshold rank?
- Idea: Decompose  $G$  into pieces so that each piece is either small or has low threshold rank and there are few edges between different pieces. We can then apply the algorithm to each piece.
- For details, see [ABS10]



# Part V: Open Problems

# Open Problems

- What is the exact relationship between UG (unique games) and SSE (small set expansion)?
- **Major open problem:** How well does SOS do on UG and SSE?
- Is there a subexponential time algorithm for max cut and/or other problems?

# References

- [ABS10] S. Arora, B. Barak, and D. Steurer. Subexponential Algorithms for Unique Games and Related Problems. FOCS 2010
- [RS10] P. Raghavendra and D. Steurer. Graph Expansion and the Unique Games Conjecture. STOC 2010