**Limits of Approximation Algorithms** 26 May, 2010 (IMSc)

# Lec. 13: Approximation Algorithms for Unique Games

*Lecturer: Prahladh Harsha* *Scribe: Prajakta Nimbhorkar*

In the last lecture, we proved an inapproximability result for the MAX-CUT problem. We also introduced the unique label cover problem and Khot's unique games conjecture.

Today, we will see how well unique games can be approximated. In particular, we will see Trevisan's algorithm for approximating unique games on low-diameter graphs [Tre08]. We will then briefly discuss the algorithm of Arora et al. for approximating unique games on expanders [AKK$^+$08], and a recent sub-exponential time algorithm for unique games on general graphs due to Arora et al [ABS10]. The references for this lecture include the above cited papers and Lecture 8 of the DIMACS tutorial on Limits of approximation [HC09].

## 13.1 Recap: Unique Games and the Unique Games Conjecture

An instance of unique games consists of a bipartite graph $G = (U, V, E)$, a label set $[m]$, and a set of permutations $\pi = \{\pi_e | e \in E\}$ on the label set. Thus $\forall e = (u, v) \in E$ $\pi_{(u,v)} : [m] \to [m]$ is a permutation on $[m]$. The desired output consists of a labeling $A : U \cup V \to [m]$ that maximizes the number of satisfied edges. An edge $e = (u, v)$ is said to be *satisfied* if $label(v) = \pi_{(u,v)}(label(u))$. It can be seen that, if all the edges can be satisfied, it is trivial to find such a labeling: pick a vertex $u$ and choose its label arbitrarily. This gives unique labels to vertices which are reachable from $u$ in $G$. Cycle through all the $m$ labels for $u$.

Therefore, given $m$, and completeness and soundness parameters $1-\varepsilon$ and $\delta$ respectively, a gap-version of unique label cover $GAP_{1-\varepsilon,\delta}ULC(m)$ is defined as follows:

$$\begin{aligned} YES &= \{(G, \pi) | \exists \text{ labeling } A \text{ such that the number of edges satisfied } \geq (1-\varepsilon)|E|\} \\ NO &= \{(G, \pi) | \forall \text{ labelings } A \text{ the number of edges satisfied } \leq \delta|E|\} \end{aligned}$$

The goal is to distinguish between the $YES$ and $NO$ instances.

**Unique Games Conjecture** ([Kho02])**.** *For all $\varepsilon, \delta$, there exists $m$ such that* $\text{GAP}_{1-\varepsilon,\delta}\text{ULC}(m)$ *does not have a polynomial-time algorithm.*

## 13.2 Approximation Algorithms for Unique Games

The purpose behind designing approximation algorithms is to refute UGC. Thus, for some $\varepsilon, \delta$, and for each $m$, we want an algorithm $\mathcal{A}$ such that, if the input ULC instance is $(1-\varepsilon)$-satisfiable, then $\mathcal{A}$ outputs a labeling that satisfies at least $\delta|E|$ edges. Thus $\mathcal{A}$ works on highly satisfiable instances and it separates YES and NO instances. (Recall that complete satisfiability is trivial.)

We consider a special case viz. MAX-CUT, where the label set has size $m = 2$. Thus the instance consists of a graph $G = (V, E)$, and the promise that it either has a cut that contains at least $(1 - \varepsilon)$-fraction of edges or all the cuts in $G$ contain less than $\delta$-fraction of edges. We have seen a randomized algorithm for MAX-CUT that puts the vertices into one of the two partitions with equal probability and gives $1/2 + \varepsilon$-approximation. Goemans-Williamson's algorithm gives $\alpha_{gw} \approx 0.87856$ approximation. The question is to determine whether we can do better when we have the above promise. We will see that Goemans-Williamson's algorithm works better on large cuts.

**Lemma 13.2.1.** *There exist constants $\varepsilon_0 \in (0, 1)$ and $c$ such that for all $\varepsilon < \varepsilon_0$ if* MAX-CUT$(G) \geq (1 - \varepsilon)|E|$*, then Goemans-Williamson's algorithm outputs a cut that contains at least $(1 - \Omega(\sqrt{\varepsilon}))|E|$ edges.*

*Proof.* Recall that the algorithm involves solving the following SDP relaxation:

$$\begin{aligned} \text{Maximize } Z &= \frac{1}{2}\mathbb{E}[1 - \langle v_i, v_j\rangle] \\ \text{such that} & \quad \forall i \|v_i\|^2 = 1 \end{aligned}$$

where $v_i \in \mathbb{R}^n$. If MAX-CUT$(G) \geq (1 - \varepsilon)|E|$ then $Z \geq (1 - \varepsilon)$. We show that the Goemans-Williamson's rounding gives a solution that satisfies the condition in the lemma. Recall that the rounding involves picking a random hyperplane passing through origin and partitioning the vectors in the solution depending on which side of the hyperplane they lie. This gives that

$$\mathbb{E}[cut_{GW}(G)] = \mathbb{E}\Big[\frac{\cos^{-1}(\langle v_i, v_j\rangle)}{\pi}\Big]$$

Let $x_e = \frac{1 - \langle v_i, v_j\rangle}{2}$ and $y = h(x) = \frac{\cos^{-1}(\rho)}{\pi}$, where $\rho = \langle v_i, v_j\rangle$. Consider the function $h$ as defined above. It is easy to check there exist constants $\varepsilon_0' \in (0, 1)$ and $c$ such that for all $\varepsilon < \varepsilon_0$ if $x \geq 1 - \varepsilon$, then $y = h(x) \geq 1 - c\sqrt{\varepsilon}$. Thus, if each of the $x_e$'s in the expectation satisfied $x_e \geq 1 - \varepsilon$, we would be done.

However, we only have the promise that $\mathbb{E}[x_e] \geq 1 - \varepsilon$. If the function $h$ were convex, we could get $\mathbb{E}[h(x_e)] \geq h(\mathbb{E}[x_e])$ which gives the desired bound. Unfortunately this is not the case. So let $\tilde{h}$ be the largest convex function under $h$. It is easy to check that there exists an $\varepsilon_0''$ such that for all $x \geq 1 - \varepsilon_0''$, we have $h(x) = \tilde{h}(x)$ (see Figure 13.2). Let $\varepsilon_0 = \min\{\varepsilon_0', \varepsilon_0''\}$. We then have

$$\mathbb{E}[\text{cut}] = \mathbb{E}[h(x_e)] \geq \mathbb{E}[\tilde{h}(x_e)] \geq \tilde{h}(\mathbb{E}[x_e]) = h(\mathbb{E}[x_e])$$

where the last equality holds because $h$ and $\tilde{h}$ are equal in the *large-cut region*, that is, the region close to 1 (see Figure 13.2). ☐

Thus Goemans-Williamson's algorithm can find a cut of size at least $1 - c\sqrt{\varepsilon}$ if there is a cut of size at least $1 - \varepsilon$. We know that $\forall \rho, \varepsilon$ $GAP_{\frac{1-\rho}{2} - \varepsilon', \frac{\cos^{-1}(\rho)}{\pi} + \varepsilon'}$- MAX-CUT is UG-hard, whereas here we have a polynomial-time algorithm for $GAP_{1-\varepsilon, 1-c\sqrt{\varepsilon}}$. Thus a slight improvement over Goemans-Williamson's algorithm implies refutation of the unique games conjecture.

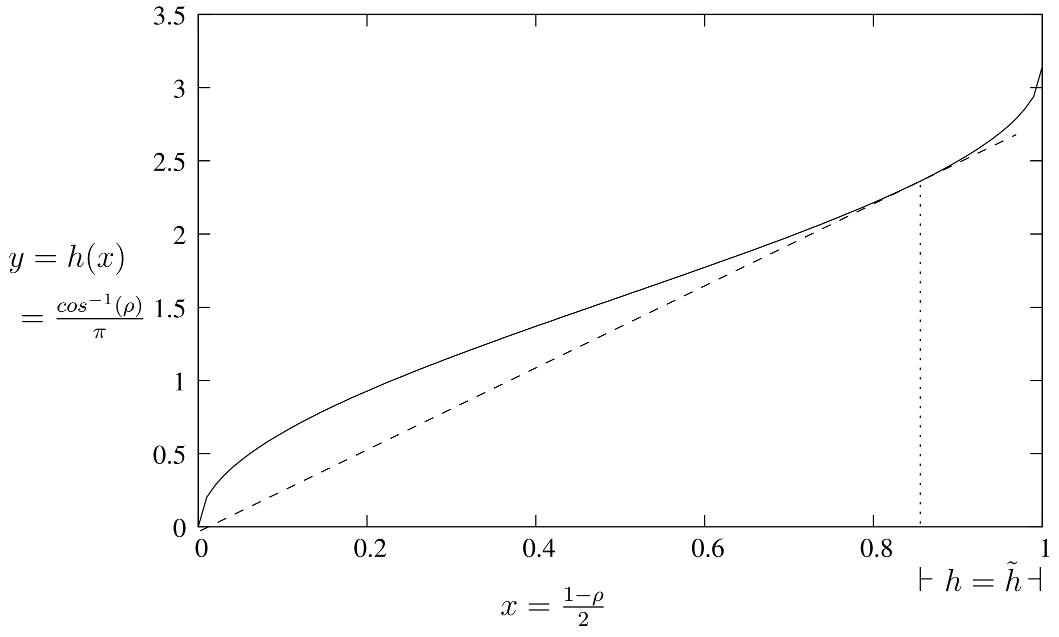

**Figure 1**: Plot of $h(x) = \frac{\cos^{-1}(1-2x)}{\pi}$ and $\tilde{h}(x)$, the largest convex function less than or equal to $h$, pointwise

## 13.3 Trevisan's Algorithm for Approximating Unique Games

Now we will see Trevisan's algorithm for approximating unique games [Tre08]. This algorithm gives a constant fraction approximation for sub-constant values of $\varepsilon$. It works by solving and rounding an SDP relation of unique games.

**SDP relaxation for unique games:** The SDP consists of variables $v_i$ for each vertex $v$ in the unique games instance and for each label $i \in [m]$. A variable $v_i = 1$ if the vertex $v$ is labelled $i$. The integer program is given by

$$\begin{array}{ll} \text{Maximize} & \displaystyle\sum_{e=(u,v)\in E} \sum_{i\in[m]} u_i v_{\pi_e(i)} \\ \text{subject to} & v_i \in \{0,1\} \\ & \forall v \quad \forall i \neq j \in [m] \quad v_i v_j = 0 \\ & \displaystyle\sum_{i\in[m]} v_i = 1 \end{array}$$

The corresponding SDP relaxation is

$$\begin{array}{ll} \text{Maximize} & \displaystyle\sum_{e=(u,v)\in E} \sum_{i\in[m]} \langle u_i, v_{\pi_e(i)}\rangle \\ \text{subject to} & \forall v \quad \forall i \neq j \quad \langle v_i, v_j\rangle = 0 \\ & \forall v \quad \forall i \neq j \in [m] \quad \langle v_i, v_j\rangle = 0 \\ & \forall v \quad \displaystyle\sum_{i\in[m]} \|v_i\|^2 = 1 \\ & \forall u, v, i, j \quad \langle u_i, v_j\rangle \geq 0 \end{array}$$

Thus for each vertex, we have $m$ mutually orthogonal vectors, such that their squared $\ell_2$ norms sum up to 1.

Note that we can tighten the above SDP by adding any constraint as long as the feasible solutions of the integer program satisfy them. Trevisan, in particular, adds the triangle inequalities which can be easily verified for the feasible integer solutions.

$$\begin{array}{rcl} \|w_h - u_i\|^2 & \leq & \|w_h - v_j\|^2 + \|v_j - u_i\|^2 \\ \|v_j - u_i\|^2 & \geq & \|v_j\|^2 - \|u_i\|^2 \end{array}$$

for each $u, v, w, i, j, h$.

Using the identity $\|a-b\|^2 = \|a\|^2 + \|b\|^2 - 2\langle a, b\rangle$, the objective function becomes

$$\sum_{e=(u,v)\in E} \left(1 - \frac{1}{2}\sum_{i\in[m]} \|u_i - v_{\pi_e(i)}\|^2\right)$$

### 13.3.1 Rounding the SDP

We now need to round the SDP to get an integral solution. Observe that $\sum_i \|u_i\|^2 = 1$, i.e., the squared $\ell_2$ norms of $u_i$'s define a probability distribution on the set of labels for $u$. This gives one natural rounding algorithm: for each vertex $u$ picking label $i$ for $u$ with probability $\|u_i\|^2$.

However, the above rounding procedure ignores possible correlations between edges. For instance, suppose an edge $e = (u, v)$ contributes 1 to the SDP solution, or equivalently $u_i = v_{\pi_e(i)}$ for all $i$. Note that this means, that the set of vectors is the same for $u$ and $v$, however in a permuted order. Thus, if we had chosen the label $i$ for $u$ (this happens with probability $\|u_i\|^2$), it makes sense to choose label $j$ for $v$ such that $u_i = v_j$ as this would ensure that $j = \pi_e(i)$ and edge $e$ is satisfied.

Let us see how we can generalize this idea even when the edge $e = (u, v) \in E$ contributes $1 - \varepsilon$ to SDP solution. In this case, we have that $\sum_{i\in[m]} \|u_i - v_{\pi_e(i)}\|^2 \leq 2\varepsilon$. In this case, even though the bundle of vectors for $u$ and $v$ are not the same, the two bundles are "close". Thus, if we pick a vector $u_i$ for $u$, we would like to pick the vector $v_j$ for $v$ such that $\|u_i - v_j\|^2$ is minimized. More formally, we do the following.

**Edge-level rounding for edge $e = (u, v)$**

1. For $u$, pick $i$ with probability $\|u_i\|^2$ and label $u$ with $i$.

2. For $v$, $(u, v) \in E$, pick $j$ such that $\|v_j - u_i\|^2$ is minimized. Label $v$ by $j$ (Break ties arbitrarily.)

However, we need to prove that this satisfies $\pi_e$. This is shown by the following claim.

**Claim 13.3.1.** $\sum_{i\in[m]} \|u_i - v_{\pi_e(i)}\|^2 \le 2\varepsilon \Rightarrow Pr[label(v) \neq \pi_e(label(u))] \le 4\varepsilon$.

*Proof.* Note that the randomness is only in Step 1. We define BAD $\subseteq [m]$ such that $i \in$BAD if $\exists j \neq \pi_e(i)$ such that $\|u_i - v_j\|^2 \le \|u_i - v_{\pi_e(i)}\|^2$. We will show that for all $i \in$ BAD, we have $\|u_i\|^2 \le 2\|u_i - v_{\pi_e(i)}\|^2$. Assuming this we can prove the claim as follows.

$$\begin{aligned} Pr[label(v) \neq \pi_e(label(u))] &= \sum_{i\in BAD} \|u_i\|^2 \\ &\le 2 \sum_{i\in BAD} \|u_i - v_{\pi_e(i)}\|^2 \le 4\varepsilon \end{aligned}$$

Now to prove the assumption. Denote $u_i = a$, $v_{\pi_e(i)} = b_1, v_j = b_2$. We have $\|a - b_2\|^2 \le \|a - b_1\|^2, \langle b_1, b_2\rangle = 0$. We want to conclude that $\|a\|^2 \le 2\|a - b_1\|^2$.

Case 1 : $\|b_1\|^2 \le \frac{1}{2}\|a\|^2$. In this case, the inequality follows since $\|a - b_1\|^2 \ge \|a\|^2 - \|b_1\|^2$.

Case 2 : $\|b_2\|^2 \le \frac{1}{2}\|a\|^2$. In this case, the inequality follows since $\|a - b_1\|^2 \ge \|a - b_2\|^2$.

Case 3 : $\|b_1\|^2, \|b_2\|^2 \ge \frac{1}{2}\|a\|^2$. As $b_1, b_2$ are mutually orthogonal, $\|b_1 - b_2\|^2 = \|b_1\|^2 + \|b_2\|^2 \ge \|a\|^2$. But by traingle inequality, $\|b_1 - b_2\|^2 \le \|a - b_1\|^2 + \|a - b_2\|^2 \le 2\|a - b_1\|^2$.

This completes the proof of the claim. □

However, the above tells us how to round such that a particular edge is satisfied. We need a more global rounding procedure. We will first see how the above idea easily generalizes to low-diameter graphs and then show how any graph can be decomposed to a low-diameter graph by discarding a small fraction of the edges.

### 13.3.2 Approximating unique games on low-diameter graphs

Now we will see an approximation algorithm for unique games when the underlying graph has low-diameter (in fact low-radius) [Tre08]. Assume that there is a vertex $r \in V$ such that for each vertex $u$, $d_G(u, r) \le d$ where $d_G$ denotes the distance in graph $G$ and $d$ is the radius of $G$. The algorithm involves solving the SDP and then using appropriate rounding scheme to get an approximate solution for unique games instance. The idea is to choose a randomized rounding scheme for $r$ and propagate it to all the other vertices.

**Rounding:**

1. Choose label $i$ for $r$ with probability $\|r_i\|^2$.
2. For each $v \in V$, label $v$ with $j$ so as to minimize $\|v_j - r_i\|^2$ (breaking ties arbitrarily).

We will now show that for each edge $e = (u, v)$ if the contribution of the edge $e$ to the SDP solution is large, then the probability that the above rounding satisfies the constraint $\pi_e$ will also be large.

**Lemma 13.3.2.** *If each edge contributes $1-\frac{\varepsilon}{8(d+1)}$ to the SDP solution, then $Pr_{(u,v)=e\in E}[e$ is satisfied $] \geq 1-\varepsilon$.*

*Proof.* Let $v_0 = r, v_1, \ldots, v_t = u$ be a path from $r$ to $u$ of length at most $d$. Define the permutation $\pi_u$ to be the composition of permutations along the path (i.e., $\pi_u = \pi_{(u_{t-1},u_t)} \circ \pi_{(u_{t-2},u_{t-1})} \circ \ldots \pi_{(u_0,u_1)}$ and $\pi_v = \pi_{(u,v)} \circ \pi_u$. Now, clearly edge $e$ is satisfied if $label(u) = \pi_u(label(r))$ and $label(v) = \pi_v(lable(r))$.

For all edges $e=(u,v)$, $\sum_{i\in[m]} \|u_i - v_{\pi_e(i)}\|^2 \leq \frac{\varepsilon}{4(d+1)}$. Hence, by triangle inequality, we have

$$\sum_{i\in[m]} \|r - u_{\pi_u(i)}\|^2 \leq \frac{\varepsilon}{4} \tag{13.3.1}$$

$$\sum_{i\in[m]} \|r_{-}v_{\pi_v(i)}\|^2 \leq \frac{\varepsilon}{4} \tag{13.3.2}$$

From 13.3.1, $Pr[\pi_u(label(r)) \neq label(u)] \leq \frac{\varepsilon}{2}$. From 13.3.2, $Pr[\pi_v(label(r)) \neq label(v)] \leq \frac{\varepsilon}{2}$. Now the lemma follows. ☐

### 13.3.3 Approximating unique games on general graphs

If $G$ is a general graph (not a low-diameter graph), we decompose it into low-radius components without discarding too many edges. In particular, we discard at most $O(\varepsilon)$-fraction of edges.

**Graph decomposition:** Given a graph $G$, it is decomposed into low-radius components as follows. Each component is created as follows.

1. Start with a vertex $u$.
2. Define $G_0 = \{u\} \cup N(u)$. The idea is to expand by one BFS layer at a time and look at the number of newly added edges.
3. Fix some $\alpha > 1$. While $E(G_i \cup N(G_i)) > \alpha E(G_i)$, $G_{i+1} = G_i \cup N(G_i)$, $i \leftarrow i+1$.
4. Output $G_i$

Thus a component is expanded as long as more than a constant fraction of new edges are added. If the radius of the component is $d$, then the number of edges in the component is at least $\alpha^d$. Since, this can be at most $|E|$, we must have $d \leq \frac{\log|E|}{\log\alpha}$. Thus, the radius of the component is at most $\frac{\log|E|}{\log\alpha} \approx O(\log n)$ for constant $\alpha$.

Let $E_i$ be the number of edges cut while forming $i$th component and $F_i$ be the number of edges in the component. We must have $(E_i + F_i) \leq \alpha F_i$. Hence, the component has at least $F_i \geq \frac{E_i}{\alpha-1}$ edges. The total number of edges is $|E| \geq \sum F_i$ which by the above argument is at least $\frac{1}{\alpha-1}\sum_i E_i$. Thus, the number of edges cut is at most $\frac{|E|}{1/(\alpha-1)}$. We choose $\alpha = 1+\varepsilon'$. Thus we have decomposition of graph into components of radius $\frac{\log|E|}{\log(1+\varepsilon')} = O(\frac{\log|E|}{\varepsilon'})$, with at most $\varepsilon'|E|$ inter-component edges.

Now the algorithm is given in the following steps: (Assume SDP solution $\geq (1-\eta)|E|$.)

1. Remove all the edges that contribute $< (1 - \frac{3\eta}{\varepsilon})$ to the SDP solution. (There are at most $\frac{\varepsilon}{3}|E|$ such edges.)

2. Decompose the graph into components of radius $O(\frac{\log n}{\varepsilon})$ by discarding at most $\frac{\varepsilon}{3}|E|$ edges.

3. Run Trevisan's algorithm.

The algorithm works if $1 - \frac{3\eta}{\varepsilon} \geq 1 - \frac{\varepsilon}{8(d+1)} \geq 1 - \frac{\varepsilon^2}{C \log n}$. Thus, if the unique label cover instance is $(1-\eta)$-satisfiable, we can find labeling that satisfies $(1-\varepsilon)$-fraction of edges, where $\eta = \frac{c\varepsilon^3}{\log n}$ for some constant $c$. Note that for expanders, we get $\frac{c\varepsilon^2}{\log n}$.

## 13.4 Approximating unique games on expanders

There are better approximations known for unique games, when the underlying graph is an expander [AKK+08, KT07, Kol10]. We give an overview of these results.

**Lemma 13.4.1** ([AKK+08]). *If a ULC instance $(G, \pi)$ is $(1-\varepsilon)$-satisfiable, and $G$ has spectral expansion $\lambda$, then there is a polynomial-time algorithm that finds a labeling satisfying $(1 - \tilde{O}(\frac{\varepsilon}{\lambda}))$-fraction of edges.*

Let $\lambda_2$ be the second eigenvalue of the Laplacian of $G$. Then $\lambda_2 = \min_{x \perp \mathbf{1}} \frac{\|x^T L x\|}{\|x^T x\|}$, where $L$ is the Laplacian matrix of $G$. Let $v_1, \ldots, v_n$ be the vertices in $G$. Associate vectors $z_1, \ldots, z_n \in \mathbb{R}^n$ respectively with each of the vertices. It is known that $\lambda_2 = \min_{z_1,\ldots,z_n \in \mathbb{R}^n} \frac{E_{(i,j)\in E}[\|z_i - z_j\|^2]}{E_{(i,j)\in V^2}[\|z_i - z_j\|^2]}$. We will use this characterization of $\lambda_2$.

The algorithm consists of solving an SDP and then rounding the solution to get labels for the vertices. The SDP is same as in the previous section, with the exception that only the triangle inequalities $\langle u_i, v_j \rangle \geq 0$ are used.

We describe the rounding here. Given a set of vectors for $u, v$, we want to determine a permutation $\sigma_{uv} : [m] \to [m]$ that minimizes $\rho_{uv} = \sum_{i \in [m]} \|u_i - v_{\sigma_{uv}(i)}\|^2$. This is the best permutation that aligns the $u_i$s and $v_j$s. This is a simple minimum weight bipartite matching problem, and hence can be exactly solved. The rounding procedure thus has following steps:

1. Pick $u \in V$ at random. Label $u$ by $i$ with probability $\|u_i\|^2$.

2. For each $v \in V$, label $v$ by $\sigma_{uv}(i)$.

The analysis involves showing that the SDP solution is large if and only if $E_{e=(u,v)\in E}[\sum_{i\in[m]} \|u_i - v_{\pi_e(i)}\|^2] \leq \varepsilon$.

**Improvements and generalizations:**

1. [KT07, Kol10]: This is a generalization of the algorithm of [AKK+08] to the case when there are $k$ eigenvalues of (the normalized adjacency matrix of )$G$ that lie between 1 and some constant $\lambda$. Note that $k = 1$ in the above algorithm. Define $rank^*_\lambda(G) =$ number of eigenvalues of $G$ between 1 and $\lambda$. If $k = n^\varepsilon$, [Kol10] gives a sub-exponential time algorithm.

2. [ABS10]: A few weeks ago, Arora, Barak and Steurer gave a sub-exponential time algorithm for UG building on the above ideas. This paper describes a way to decompose a graph into components, where each component has $rank^*_\lambda = n^\varepsilon$, by discarding at most an $\varepsilon$-fraction of edges. The algorithm is iterative, where an SDP for the $i+1$st largest eigenvalue is written from the solution of the SDP for the $i$th largest eigenvalue. The result is given in the following lemma:

   **Lemma 13.4.2** ([ABS10])**.** *If $G$ is $(1-\varepsilon)$-satisfiable, then the algorithm finds a labeling that satisfies at least $\frac{1}{2}$-fraction of edges. Moreover, the algorithm runs in time $2^{n^\varepsilon}$.*

## References

[ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. *Subexponential algorithms for unique games and related problems*, 2010. (manuscript).

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. *Unique games on expanding constraint graphs are easy: extended abstract.* In *Proc. 40th ACM Symp. on Theory of Computing (STOC)*, pages 21–28. 2008. doi:10.1145/1374376.1374380.

[HC09] Prahladh Harsha and Moses Charikar. *Limits of approximation algorithms: PCPs and unique games*, 2009. (DIMACS Tutorial, July 20-21, 2009). arXiv:1002.3864.

[Kho02] Subhash Khot. *On the power of unique 2-prover 1-round games.* In *Proc. 34th ACM Symp. on Theory of Computing (STOC)*, pages 767–775. 2002. doi:10.1145/509907.510017.

[Kol10] Alexandra Kolla. *Spectral algorithms for unique games.* In *Proc. 25th IEEE Conference on Computational Complexity*, pages 122–130. 2010. eccc:TR10-029, doi:10.1109/CCC.2010.20.

[KT07] Alexandra Kolla and Madhur Tulsiani. *Playing random and expanding unique games*, 2007. (manuscript).

[Tre08] Luca Trevisan. *Approximation algorithms for unique games.* Theory of Computing, 4(1):111–128, 2008. (Preliminary version in *46th FOCS*, 2005). eccc:TR05-034, doi:10.4086/toc.2008.v004a005.