Spectral Algorithms for Unique Games

Alexandra Kolla*

Abstract

We give a new algorithm for Unique Games which is based on purely *spectral* techniques, in contrast to previous work in the area, which relies heavily on semidefinite programming (SDP). Given a highly satisfiable instance of Unique Games, our algorithm is able to recover a good assignment. The approximation guarantee depends only on the completeness of the game, and not on the alphabet size, while the running time depends on spectral properties of the *Label-Extended* graph associated with the instance of Unique Games.

We further show that on input the integrality gap instance of Khot and Vishnoi, our algorithm runs in quasi-polynomial time and decides that the instance if highly unsatisfiable. Notably, when run on this instance, the standard SDP relaxation of Unique Games *fails*. As a special case, we also re-derive a polynomial time algorithm for Unique Games on expander constraint graphs.

The main ingredient of our algorithm is a technique to effectively use the full spectrum of the underlying graph instead of just the second eigenvalue, which is of independent interest. The question of how to take advantage of the full spectrum of a graph in the design of algorithms has been often studied, but no significant progress was made prior to this work.

---

*Microsoft Research.

1 Introduction

A Unique Game is defined in terms of a constraint graph $G = (V, E)$, an integer k which is called the alphabet, a set of variables $\{x_u\}_{u \in V}$, one for each vertex u, and a set of permutations (constraints) $\pi_{uv} : [k] \to [k]$, one for each edge $(u, v) \in E$. An assignment to the variables (labeling) is said to satisfy the constraint on the edge $(u, v) \in E$ if $\pi_{uv}(x_u) = x_v$. The edges are taken to be undirected and hence $\pi_{uv} = (\pi_{vu})^{-1}$. The goal is to assign a value from the set $[k]$ to each variable x_u so as to maximize the number of satisfied constraints.

Khot [Kho02] conjectured that it is NP-hard to distinguish between the cases when almost all the constraints of a Unique Game are satisfiable and when very few of the constraints are satisfiable:

Conjecture 1. *(Unique Games Conjecture-*UGC*) For any constants $\epsilon, \delta > 0$, there is a $k(\epsilon, \delta)$ such that for any $k > k(\epsilon, \delta)$, it is NP-hard to distinguish between instances of Unique Games with alphabet size k where at least $1 - \epsilon$ fraction of constraints are satisfiable and those where at most δ fraction of constraints are satisfiable.*

There are two reasons which make UGC particularly intriguing. First, it is a well-balanced question. Despite continuous efforts to prove or disprove it, there is still no consensus regarding its validity. This seems to indicate that UGC is more likely to be resolved in the near future in contrast to the P vs. NP problem for example, for which it is widely believed that P$\neq$NP but current techniques have been unable to prove it. Second, as seen by a series of works, the truth of UGC implies that the currently best known approximation algorithms for many important computational problems have optimal approximation ratios. Since its origin, UGC has been successfully used to prove often optimal hardness of approximation results for several important NP-hard problems such as MIN-2SAT-DELETION [Kho02], VERTEX COVER [KR03], MAXIMUM CUT [KKMO04] and non-uniform SPARSEST CUT [CKK$^+$06,KV05]. In addition, in recent years, UGC has also proved to be intimately connected to the limitations of semidefinite programming (SDP). Making this connection precise, the authors in [Aus10] and [Rag08] show that if UGC is true, then for every constraint satisfaction problem (CSP) the best approximation ratio is given by a certain simple SDP.

Arguably, a seemingly strong reason for belief in UGC is the failure of several attempts to design efficient algorithms for Unique Games using current state-of-the-art techniques, such as linear and semidefinite programming (LP/SDP). Indeed, several works [KV05, RS09] show limitations of LPs and SDPs in solving Unique Games by exhibiting the existence of *itegrality gap* instances. Those are Unique Games instances that *fool* the SDP since, even though they have no good satisfying assignment, the respective SDP solution is high. The existence of such instances implies, in particular, that there is no hope for a "good" approximation algorithm based on those SDPs. Moreover, recently it was shown that solving Unique Games is at least as hard as another seemingly hard problem called the small set expansion problem [RS10]. Our work presents evidence that a different set of techniques might be more powerful when it comes to designing algorithms for Unique Games thus giving hope that such techniques can be used to potentially disprove UGC.

We present a purely *spectral* algorithm for Unique Games that finds highly satisfying assignments when they exist. The running time of our algorithm depends on spectral properties of the *Label-Extended* graph or the constraint graph associated with the instance of Unique Games. Our algorithm runs in polynomial or quasi-polynomial time for a large class of instances, including those where the standard SDP provably fails as well as instances where the constraint graph is an expander. At a high level, we show that given $\epsilon > 0$, there is a $\delta = \delta(\epsilon)$ such that the algorithm is able to distinguish between the following two cases:

- **YES** case: There exists an assignment that satisfies $(1 - \epsilon)$ fraction of the constraints.

- **NO** case: Every assignment satisfies less that δ fraction of the constraints.

In particular, our algorithm runs in quasi-polynomial time on input the SDP integrality gap instance as it appears in [KV05] (hereafter referred to as $\mathcal{KV}$). We note that the authors in [KV05] roughly showed that, when run on a certain highly unsatisfiable instance of Unique Games, the standard SDP relaxation has very high objective value and consequently fails to distinguish between the two cases above. As another special case, our algorithm runs in polynomial time when the constraint graph is an expander (therefore re-deriving results similar to [AKK+08] and [KT07]). Moreover, similarly to [AKK+08, MM09, KT07, AIMS09], the performance of the algorithm does not depend on the alphabet size k.

Our main result is the following:

Theorem. *(Main) Let $\mathcal{U} = (G, M, k)$ be a $(1 - \epsilon)$ satisfiable instance of Unique Games on a d-regular graph G with alphabet size k. Let M be the adjacency matrix of the label-extended graph of G. Let W be the space spanned by eigenvectors of M that have eigenvalue greater than $(1 - \gamma)d$, for $\gamma > 8\epsilon$. There is an algorithm that runs in time $2^{O(\frac{\gamma}{\epsilon} \dim(W))} + poly(n \cdot k)$ and finds an assignment that satisfies at least $(1 - O(\frac{\epsilon}{\gamma - 8\epsilon} + \epsilon))$ fraction of the constraints.*

We also show a similar theorem for the special case where all the constraints are of the form Γ-Max-Lin and the graph satisfies some special properties. That theorem is then used to derive the above-mentioned algorithm for expanders.

Our key contribution is a technique of effectively using the full spectrum and eigenvectors of the relevant graph rather than just the first few eigenvalues. Interestingly, prior to this work, researchers had often attempted to take advantage of the full spectrum of a graph in the design of approximation algorithms, but no significant progress was made. We believe that our techniques are of independent interest and could contribute to developing better algorithms for a number of other problems. In a recent breakthrough paper, Arora, Barak and Steurer [ABS10] used our main technique as a crucial building block in their sub-exponential time algorithm for arbitrary instances of Unique Games. Consequently, we believe that our results give evidence that spectral techniques might be a more powerful tool than SDPs for attacking and, potentially even disproving, the Unique Games Conjecture.

1.1 Comparison to Other Work on Unique Games

Several polynomial time approximation algorithms using linear and semidefinite programming have been developed for Unique Games on arbitrary graphs (see [Kho02], [Tre05], [GT06], [CMM06a], [CMM06b]). These algorithms start with an instance where the value of the SDP or LP relaxation is $1 - \epsilon$ and round it to a solution with value ν. Here, value of the game refers to the maximum fraction of satisfiable constraints. For $\nu > \delta$, this would give an algorithm to distinguish between the two cases. However, most of these algorithms give good approximations only when ϵ is very small ($\epsilon = O(1/\log n)$ or $\epsilon = O(1/\log k)$) and their approximation guarantee depends on the alphabet size k [1]. For constant ϵ, only the algorithm in [CMM06a] gives interesting parameters with $\nu \approx k^{-\epsilon/(2-\epsilon)}$. We refer the reader to [CMM06a] for a comparison of parameters of various algorithms.

---

[1] It might be good to think of k as $O(\log n)$ since this is the range of interest for most reductions.

For some special cases of graphs, it has been shown that there are efficient algorithms that solve Unique Games. One such example is when the constraint graph is a *spectral expander*. In that case, the authors in [AKK+08] (and later [MM09] with improved parameters) showed that one can find a highly satisfying assignment (with, say, $\nu \geq 90\%$) in polynomial time. Notably, the approximation guarantee of their algorithm depends on the expansion parameters of the graph rather than the alphabet size. As another example, the authors in [AIMS09] presented efficient algorithms for constraint graphs with large *local expansion*.

Compared to the previous papers, our algorithm uses purely spectral methods as opposed to linear or semidefinite programming and runs in time that depends on spectral properties of the constraint graph or the label-extended graph. The approximation guarantee does not depend on the size of the graph or the size of the alphabet and thus our algorithm always finds a good assignment when one exists. We note that, in an unpublished manuscript, Kolla and Tulsiani [KT07] developed a *spectral* algorithm that runs in polynomial time and finds highly satisfying assignments on expander constraint graphs.

1.2   Overview of Our Techniques: Recovering Solutions by Spectral Methods

Our basic approach for the Unique Games algorithm is exhaustive search in a subspace spanned by several eigenvectors of a graph associated with the Unique Games instance. We identify a "good" subspace which contains a vector that "encodes" a highly satisfying assignment and then exhaustively search for this vector in (a discretization of) the subspace as we explain below.

Let $\mathcal{U}$ be our Unique Games instance which is $(1 - \epsilon)$ satisfiable. One can think of this instance as "coming from" a completely satisfiable instance $\tilde{\mathcal{U}}$ as follows: an adversary on input $\tilde{\mathcal{U}}$, picks an ϵ fraction of edges and changes the constraints on those edges so that the resulting instance $\mathcal{U}$ becomes $(1 - \epsilon)$ satisfiable. Let M be (the adjacency matrix of) the graph corresponding to $\mathcal{U}$ and $\widetilde{M}$ (the adjacency matrix of) the graph corresponding to $\tilde{\mathcal{U}}$. Assume, for simplicity, that these graphs are d-regular. We first observe that a characteristic vector y_α of a perfectly satisfying assignment α for $\tilde{\mathcal{U}}$ is an eigenvector of $\widetilde{M}$ with eigenvalue d.

We next consider a space W which is the span of the eigenvectors of M with eigenvalue very close to d and show that every characteristic vector y_α is close to some vector in W (in ℓ_2 norm).

Our algorithm simply looks at a set $\mathcal{S} \subseteq W$ of appropriately many candidate vectors and "reads-off" an assignment.

Recover-Solution$_{\mathcal{S}}(\mathcal{U})$

- For each $x \in \mathcal{S}$, construct an assignment L_x by assigning to each vertex u, the index corresponding to the largest entry in the block $(x_{u1}, \ldots, x_{uk})$ i.e. $L_x(u) = \arg\max_i x_{ui}$.

- Out of all assignments L_x for $x \in \mathcal{S}$, choose the one satisfying the maximum number of constraints.

To choose $\mathcal{S}$, we identify a set of *nice* vectors $\mathcal{N} \subseteq W$ such that the above algorithm works for any vector v close to some vector in $\mathcal{N}$. These are going to be the vectors that are close to the assignment eigenvectors. We then construct a set $\mathcal{S} \subseteq W$ of test vectors such that at least one vector $v \in \mathcal{S}$ is close to a vector in $\mathcal{N}$. $\mathcal{S}$ is simply an epsilon-net for W. Lastly, we go over all vectors in $\mathcal{S}$ until we find v. The running time of the algorithm will depend on the size of $\mathcal{S}$ which, for an appropriately defined epsilon-net, it is exponential in the dimension of W.

Certain "simple" graphs, including expander graphs and the Khot-Vishnoi instance, have only a few eigenvalues close to d and thus the dimension of W is small. Consequently, exhaustive enumeration in the subspace spanned by the corresponding eigenvalues will quickly find a good-enough assignment.

1.3 Organization of the Paper

The rest of the paper is organized as follows: in section 2 we give some preliminary notation and definitions that will be used in the rest of the paper. Section 3 contains the proofs of the main theorem above as well as of a theorem for the Γ-Max-Lin case. More specifically, we prove the main theorem in subsection 3.1. The proof of the theorem for the Γ-Max-Lin case appears in subsection 3.2. Subsection 3.3 contains the generalization of the main theorem for non-regular graphs. In section 4, we analyze our algorithm when run on the Khot-Vishnoi instance.

2 Preliminaries

2.1 Spectra of Graphs

We remind the reader that for a graph G, the adjacency matrix $A = A_G$ is defined as

$$A_G(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{if } (u, v) \notin E \end{cases}$$

We will also be dealing with weighted graphs, where edge (u, v) has weight $w_{uv} \geq 0$ and the adjacency matrix is defined as $A_G(u, v) = w_{uv}$. We will also assume, w.l.o.g. that $w_{uv} \leq 1$ (if not, we just re-scale the weights of the edges by the maximum weight). If the graph has n vertices, A_G has n real eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. We can always choose n eigenvectors $\gamma_1, \cdots, \gamma_n$ such that γ_i has eigenvalue λ_i which form an orthonormal basis of $\mathbb{R}^n$. We note that if the graph is d-regular (the total weight of edges adjacent to every node is d) then the largest eigenvalue is equal to d and the corresponding unit length eigenvector is the (normalized) all-one's vector.

For a graph G, let D be the diagonal matrix with diagonal entry $D(u, u) = \sum_{v:(u,v)\in E} w_{uv}$ equal to the degree of node u, namely the sum of the weights of edges adjacent to u. The Laplacian of G is defined as follows:

$$L_G = D - A_G$$

If the graph has n vertices, L_G has n real eigenvalues $0 = \mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$. We can always choose n eigenvectors $\gamma_1, \cdots, \gamma_n$ such that γ_i has eigenvalue μ_i which form an orthonormal basis of $\mathbb{R}^n$. We note that 0 is always an eigenvalue with corresponding unit length eigenvector the (normalized) all-one's vector. Moreover, if and only if the graph has k connected components, then L_G has k eigenvalues equal to zero.

2.2 The Label-Extended Graph

For a given instance of Unique Games on a constraint graph $G = (V, E)$, let A be the adjacency matrix of G and let M denote the $nk \times nk$ matrix such that the $k \times k$ block M_{uv} is equal to $w_{uv}\Pi_{uv}$. We use Π_{uv} to denote the $k \times k$ matrix of the permutation π_{uv}. Namely,

$$\Pi_{uv}(i,j) = \begin{cases} 1 & \text{if } \pi_{uv}(i) = j \\ 0 & \text{otherwise} \end{cases}$$

Note that $\pi_{uv} = (\pi_{vu})^{-1}$ implies $\Pi_{uv} = \Pi_{vu}{}^T$ therefore M is symmetric. We can view M as the adjacency matrix of the *Label-Extended* graph for that instance of Unique Games. Let k denote the size of the alphabet. We will denote this instance by $\mathcal{U} = (G, M, k)$.

Definition 2. *(Characteristic vector of a labeling) For any labeling of the vertices $L = \{L_u\}_{u \in V}$ with $L_u \in [k]$ we define the characteristic vector of that labeling as the nk dimensional vector y^L with*

$$y^L(u,i) = \begin{cases} 1 & \text{if } i = L_u \\ 0 & \text{otherwise} \end{cases}$$

We will often normalize such vectors to make them unit vectors. In this case, we have

$$\tilde{y}^L(u,i) = \begin{cases} \frac{1}{\sqrt{n}} & \text{if } i = L_u \\ 0 & \text{otherwise} \end{cases}$$

With some abuse of notation we will refer to both vectors as characteristic vectors of labelings whenever it is clear from the context.

The following is an important observation:

Observation 3. *Assume that a certain instance $\mathcal{U}$ is completely satisfiable. Namely, assume that there is a labeling $L = \{L_u\}_{u \in V}$ with $L_u \in [k]$ such that, for all $(u,v) \in E$ we have $\pi_{uv}(L_u) = L_v$. Then the characteristic vector y^L is a eigenvector of M with eigenvalue d, if G is a d-regular graph. It is also an eigenvector of the laplacian L_M with eigenvalue 0.*

2.3 Γ-Max-Lin Instances

Let Γ be an abelian group. A Γ-Max-Lin instance of Unique Games on graph $G(V,E)$ has for each edge $(u,v) \in E$ a constraint of the form $x_u - x_v = c_{uv}$, where x_u, x_v are variables taking values in Γ and $c_{uv} \in \Gamma$. The alphabet k is the size of the group Γ. We note that if a pair of labels $(L_u, L_v) \in \Gamma \times \Gamma$ satisfies a constraint $x_u - x_v = c_{uv}$ of the above form, then for all $i \in \Gamma$, the pair $(L_u + i, L_v + i) \in \Gamma \times \Gamma$ also satisfies the constraint. This implies that any labeling $\{L_u\}_{u \in V}$ of the vertices of G is shift-invariant. Namely, for all $i \in \Gamma$, the labelings $\{L_u\}_{u \in V}$ and $\{L_u + i\}_{u \in V}$ satisfy the same set of constraints.

2.4 The Spectrum of Cayley Graphs

For the purposes of this paper, we will use a generalized definition for Cayley graphs. For background on the standard definition and properties of Cayley graphs see, for example, [Kas02, Spi09].

Definition 4. *(Adjacency matrix of a Cayley graph) Let Ω be a group with n elements. The vertex set of a Cayley graph $C(\Omega)$ is the group Ω. The adjacency matrix of $C(\Omega)$ is the $n \times n$ matrix $A_{C(\Omega)}$ defined as follows: For $g_1, g_2 \in \Omega$, $A_{C(\Omega)}(g_1, g_2) = f(g_1 - g_2)$, where "$-$" refers to the inverse group operation and $f : \Omega \to \mathbb{R}_+$. Note that the above definition allows $C(\Omega)$ to have weighted edges, with the only constraint being that the weight of an edge between g_1 and g_2 depends on their group-theoretic difference.*

We note that the standard definition of a Cayley graph, given a set S of generators, is a special case of the above, where $f(g_1 - g_2) = 1$ if $g_1 - g_2 \in S$ and 0 otherwise.

For what follows, we will only be interested in Ω being abelian. In the rest of this section, for simplicity, we will identify a vector f in $\mathbb{R}^n$ with the function $f : \Omega \to \mathbb{R}$ and we will use the two interchangeably.

It is well known, that for the abelian case $A_{C(\Omega)}$ has an eigenbasis that consists of the n group-theoretic *characters* $\chi_g, g \in \Omega$. We remind the reader that every function $f : \Omega \to \mathbb{R}$ can be written as follows: $f(x) = \sum_{g \in \Omega} \hat{f}(g) \chi_g(x)$ where $\hat{f}(g)$ is the *fourier coefficient* that corresponds to character χ_g. For more background on characters and fourier analysis we refer the reader to standard algebra textbooks and surveys see, for example [Art91], [Lin07], [Kas02].

Since definition 4 is slightly different than the usual, we present a proof of the following lemma for completeness.

Lemma 5. *Let $C(\Omega)$ be Cayley graph of an abelian group Ω, as in definition 4, and $A_{C(\Omega)}$ be its adjacency matrix. Then for every $g \in \Omega$, $A_{C(\Omega)} \chi_g = |\Omega| \hat{f}(g) \chi_g$. Namely, every character χ_g is an eigenvector of $A_{C(\Omega)}$ and $|\Omega| \hat{f}(g)$ is its corresponding eigenvalue. Here, with $\hat{f}$ we denote fourier coefficients.*

Proof. It is enough to prove the equality for each entry of $A_{C(\Omega)} \chi_g$, that is for every vertex $x \in \Omega$ we will show $\{A_{C(\Omega)} \chi_g\}_x = |\Omega| \hat{f}(g) \chi_g(x)$. We calculate:

$$\sum_{y \in \Omega} A_{C(\Omega)}(x, y) \chi_g(y) = \sum_{y \in \Omega} f(x - y) \chi_g(y) = \sum_{z \in \Omega} f(z) \chi_g(x) \chi_g(-z) = |\Omega| \hat{f}(g) \chi_g(x)$$

$\square$

2.5 The Khot-Vishnoi Graph

In [KV05], the authors considered the following family of graphs:

For parameters n and ϵ, let $N = 2^n$ and $n = 2^k$. Denote by $\mathcal{F}$ the family of all boolean functions on $\{-1, 1\}^k$. For $f, g \in \mathcal{F}$ define the product fg as $(fg)(x) = f(x)g(x)$. Let $H = \{\chi_S | S \subseteq [k]\}$ be the set of characters of the group $\mathbf{F}_2^n$. Consider the equivalence relation $\equiv$ on $\mathcal{F}$ defined as $f \equiv g$ iff there is an $S \subseteq [k]$ such that $f = g\chi_S$. This relation partitions $\mathcal{F}$ into equivalence classes $\mathcal{P}_1, \cdots, \mathcal{P}_m$, where $m = \frac{N}{n} = \frac{2^n}{n}$. For each equivalence class $\mathcal{P}_i$, we could pick an arbitrary representative $p_i \in \mathcal{P}_i$, so that

$$\mathcal{P}_i = \{p_i \chi_S | S \subseteq [k]\}$$

For $0 < \epsilon \le 1$ let $f \in_\epsilon \mathcal{F}$ denote a random boolean function on $\{-1, 1\}^k$ where for every $x \in \{-1, 1\}^k$, independently, $f(x) = 1$ with probability $1 - \epsilon$ and $f(x) = -1$ with probability ϵ. For the given parameter ϵ and boolean functions $f, g \in \mathcal{F}$ let

$$\mathbf{wt}_\epsilon(\{f, g\}) := \mathbf{Pr}_{f' \in_{1/2} \mathcal{F}, \mu \in_\epsilon \mathcal{F}}\big[(\{f = f'\} \wedge \{g = f'\mu\}) \vee (\{g = f'\} \wedge \{f = f'\mu\})\big]$$

The $\mathcal{KV}_{n,\epsilon}$ constraint graph with parameters n and ϵ can now be defined to have vertex set $V = \{\mathcal{P}_1, \cdots, \mathcal{P}_m\}$. For every $f \in \mathcal{P}_i$ and $g \in \mathcal{P}_j$, there is an edge between the vertices $\mathcal{P}_i$ and $\mathcal{P}_j$ with weight $\mathbf{wt}_\epsilon(\{f, g\})$

We next describe the set of constraints. The set of labels for the instance will be $2^{[k]} := \{S : S \subseteq [k]\}$, i.e. the set of labels $[n]$ is identified with the set $2^{[k]}$ (and, thus, $n = 2^k$). This identification will be used from now on. Let $f \in \mathcal{P}_i$ with $f = p_i \chi_S$ and $g \in \mathcal{P}_j$ with $g = p_i \chi_T$ for some $S, T \subseteq [k]$. The constraint π_e for the edge $e\{f, g\}$ corresponding to the pair of functions f, g can now be defined as

$$\pi_e^{\mathcal{P}_i}(T \triangle U) = S \triangle U, \quad \forall U \subseteq [k]$$

Here, $\triangle$ is the symmetric difference operator on sets.

We denote the label-extended graph of the instance described above with $\widetilde{\mathcal{KV}}_{n,\epsilon}$. The graph $\widetilde{\mathcal{KV}}_{n,\epsilon}$ has vertex set $\{-1, 1\}^n$. Between two vertices $f, g \in \{-1, 1\}^n$ there is an edge with weight $n \cdot \mathbf{wt}_\epsilon(\{f, g\})$.

It will be useful, for the purposes of this paper, to consider an equivalent definition of $\mathcal{KV}_{n,\epsilon}$ and $\widetilde{\mathcal{KV}}_{n,\epsilon}$, by translating to the $\{0, 1\}$ language. Let $H_n = \{0, 1\}^n$ be the group $\mathbf{F}_2^n$ with addition (modulo 2) as group operation. Let $n = 2^k$. Then the set H is the *Hadamard* code on n bits. Namely,

$$H = \left\{ h_y \in \{0, 1\}^n, y \in \{0, 1\}^k | \text{s.t. } h_y(x) = \langle x, y \rangle, x \in \{0, 1\}^k \right\}$$

Note that $|H| = n$ and H is a subgroup of H_n.

The graph $\mathcal{KV}_{n,\epsilon}$ is just a Cayley graph of the quotient group of H_n by H, namely of the group $Q = H_n / H$. The vertex set $V = \{\mathcal{P}_1, \cdots, \mathcal{P}_m\}$ consists of the $\frac{N}{n}$ cosets of H. For every f in the coset $\mathcal{P}_i$ and for every g in the coset $\mathcal{P}_j$, there is an edge between the vertices $\mathcal{P}_i$ and $\mathcal{P}_j$ with weight $\mathbf{wt}_\epsilon(\{f, g\})$. For simplicity of notation, we will identify the cosets with their representatives, i.e. if $x = \mathcal{P}_i$ is a vertex of $\mathcal{KV}_{n,\epsilon}$, then we will use x to refer both to the coset $\mathcal{P}_i$ as well as to its representative p_i.

The total weight of all the edges between two cosets $x = \mathcal{P}_i$ and $y = \mathcal{P}_j$ is

$$A(x, y) = \sum_{f \in \mathcal{P}_i, g \in \mathcal{P}_j} \epsilon^{|f+g|}(1 - \epsilon)^{n-|f+g|} = \sum_{h_1, h_2 \in H} \epsilon^{|x+h_1+y+h_2|}(1 - \epsilon)^{n-|x+h_1+y+h_2|}$$

Note that the graph is n-regular, namely the total weight of edges adjacent to any node x is

$$\sum_{y \in Q} \sum_{h_1, h_2 \in H} \epsilon^{|x+h_1+y+h_2|}(1 - \epsilon)^{n-|x+h_1+y+h_2|} = n \sum_{y \in Q} \sum_{h_2 \in H} \epsilon^{|x+y+h_2|}(1 - \epsilon)^{n-|x+y+h_2|}$$

$$= n \sum_{z \in H_n} \epsilon^{|z|}(1 - \epsilon)^{n-|z|} = n \sum_{i=0}^n \binom{n}{i} \epsilon^i (1 - \epsilon)^{n-i} = n(1 - \epsilon + \epsilon)^n = n$$

The label-extended graph $\widetilde{\mathcal{KV}}_{n,\epsilon}$ is a Cayley graph of H_n, since for every two vertices $u, v \in \{0, 1\}^n$ we have that the corresponding entry of the adjacency matrix of $\widetilde{\mathcal{KV}}_{n,\epsilon}$ depends only on the hamming distance of u and v, or, in other words, on $u - v = u + v$ (modulo 2). Moreover, $\widetilde{\mathcal{KV}}_{n,\epsilon}$ can be

described as an "ε-perturbed" version of the standard hypercube graph: the weight between two vertices $u, v \in \{0,1\}^n$ is proportional to the probability that if one starts from the n-bit string u and flips each one of its bits independently with probability ϵ, then one gets the n-bit string v. The following appears in [KV05]. We give the informal statement here, for simplicity. We refer the reader to [KV05] for the full statement and details.

Claim 6. *(Integrality Gap for Unique Games, Informal Statement) Let n be an integer and $\epsilon > 0$ be a parameter. Then for the instance of Unique Games described above, with label-extended graph $\widetilde{\mathcal{KV}}_{n,\epsilon}$ the following hold:*

- *Every labeling $L : V \to [n]$ satisfies at most a $\frac{1}{n^\epsilon}$ fraction of the total weight of the edges.*

- *The standard SDP relaxation for Unique Games has objective value greater than $1 - 9\epsilon$.*

2.6 Additional Notation

For simplicity of the presentation, we will assume that G is a regular graph, namely $\sum_v w_{uv} = d$ for all nodes u. At the end of the next section, we show that the results can easily be generalized to non-regular graphs.

For the sake of convenience, in the rest of the paper we will often use the term *eigenspace* to refer to the space spanned by a set of eigenvectors that don't necessarily have the same eigenvalue. The relevant set of eigenvectors that span this space will always be clear from the context. We use poly(n) to refer to some polynomial function in n.

We also use the following notation for the time needed to compute eigenvalues and eigenvectors:

Definition 7. *Let $\mathcal{U} = (G, M, k)$ be an instance of Unique Games on a graph G on n nodes and with alphabet size k. Let M be the adjacency matrix of the label-extended graph of G. Let W be some space spanned by eigenvectors of M. We denote by $T_W(M)$ the time needed to compute an eigenbasis and the corresponding eigenvalues of W. Note that $T_W(M)$ is polynomial in the dimension of M, namely $T_W(M) = poly(n \cdot k)$.*

We also note that in the rest of this paper we use the terms "assignment" and "labeling" interchangeably.

3 Recovering Solutions by Spectral Methods

In this section, we will show how, given a $(1-\epsilon)$ satisfiable instance of Unique Games $\mathcal{U} = (G, M, k)$ on graph G and with alphabet size k, the eigenvectors of the label-extended graph M may be used to recover good assignments. Specifically, we show the following:

Theorem 8. *(Main) Let $\mathcal{U} = (G, M, k)$ be a $(1 - \epsilon)$ satisfiable instance of Unique Games on a d-regular graph G on n nodes with alphabet size k. Let M be the adjacency matrix of the label-extended graph of G as above. Let W be the space spanned by eigenvectors of M that have eigenvalue greater than $(1 - \gamma)d$, for $\gamma > 8\epsilon$. There is an algorithm that runs in time $2^{O(\frac{\gamma}{\epsilon} dim(W))} + poly(n \cdot k)$ and finds an assignment that satisfies at least $(1 - O(\frac{\epsilon}{\gamma - 8\epsilon} + \epsilon))$ fraction of the constraints.*

In particular, the theorem implies that for small enough ϵ, for every $1 - \epsilon$ satisfiable instance of Unique Games that satisfies the assumptions of the theorem, one can find an assignment that satisfies more than 90 percent of the constraints in time that depends on the spectral profile of M. We also remark the following:

Remark: If the algorithm of theorem 8 fails to find an assignment that satisfies at least $(1 - O(\frac{\epsilon}{\gamma - 8\epsilon} + \epsilon))$ fraction of the constraints, then for the input $\mathcal{U}$, every assignment satisfies less than $(1 - \epsilon)$ fraction of the constraints. Here, the constant in the $O(\cdot)$ notation is the same as the one guaranteed by the theorem above.

We also consider the special case where the constraints are arbitrary Γ-Max-Lin. For such constraints, we prove theorem 9 below.

Theorem 9. *Let $\mathcal{U} = (G, M, k)$ be a $(1 - \epsilon)$ satisfiable Γ-Max-Lin instance of Unique Games on a d-regular graph G on n nodes with alphabet size k. Let $S_{(1-\gamma)}$ be the space spanned by eigenvectors of G that have eigenvalue greater than $(1 - \gamma)d$. Assume moreover that every unit-length vector $\phi \in S_{(1-\gamma)}$ has $\|\phi\|_\infty \leq \frac{C}{\sqrt{n}}$ for some constant C. Then, for $\gamma = \Omega(\sqrt{\epsilon})$, there is an algorithm that runs in time $2^{O(k \cdot D_S)} + poly(n \cdot k)$ and finds an assignment that satisfies at least $(1 - \zeta)$ fraction of the constraints for some $\zeta = O(\sqrt{\epsilon})$. Here D_S denotes the dimension of $S_{(1-\gamma)}$ and the constant in the $\Omega(\cdot)$ notation depends on C.*

Assuming that $T_{S_{(1-\gamma)}}(M) < 2^{O(k \cdot D_S)}$ the algorithm in theorem 9 has running time that solely depends on the spectral profile of G, since roughly, the space of eigenvectors of M with large eigenvalue has dimension equal to k times the dimension of the corresponding eigenspace of G.

The result for expander graphs as it appears in [KT07] can be derived as a corollary of theorem 9.

Corollary 10. *(Unique Games are Easy on Expanders) Let $\mathcal{U} = (G, M, k)$ be a $(1 - \epsilon)$ satisfiable Γ-Max-Lin instance of Unique Games. Assume moreover that G is a d-regular spectral expander on n nodes. Namely, the second eigenvalue of the adjacency matrix of G is $\lambda \leq (1 - \gamma)d$, for $\gamma = \Omega(\sqrt{\epsilon})$. Then, there is a polynomial time algorithm that finds an assignment that satisfies at least $(1 - \zeta)$ fraction of the constraints for some $\zeta = O(\sqrt{\epsilon})$.*

Proof. The eigenspace $S_{(1-\gamma)}$ of theorem 9 consists solely of the all 1's vector. The ℓ_∞ norm assumption is trivially satisfied with $C = 1$. Then, the conclusion of the theorem implies that for $\gamma = \Omega(\sqrt{\epsilon})$, there is an algorithm that runs in time $2^{O(k \cdot 1)} + T_{S_{(1-\gamma)}}(M) = 2^{O(k)} + poly(n \cdot k)$ and finds an assignment that satisfies at least an $1 - O(\sqrt{\epsilon})$ fraction of the constraints. By the statement of the Unique Games Conjecture, it is enough to consider k to be at most logarithmic in n. Therefore the algorithm runs in $2^{O(\log n)} + poly(n \cdot k) = poly(n)$ time. $\square$

We remark that the Γ-Max-Lin requirement is necessary for corollary 10. The proof fails to produce similar guarantee for the general case. This is due to the fact that the spectral properties of label-extended graphs that correspond to arbitrary Unique Games are poorly understood.

3.1 Proof of Theorem 8

We prove theorem 8 by combining two facts. First we show that characteristic vectors of good labelings have large projection onto the eigenspace of M spanned by eigenvectors with large eigenvalues. Then we show that if M has at most D large eigenvalues then we can find a vector close

(in ℓ_2-norm) to such a characteristic vector of a labeling in time exponential in D, by looking at an appropriate epsilon-net for this eigenspace. Our proof is inspired by the approach that the authors in the unpublished manuscript [KT07] used in order to recover satisfying assignments on expanders.

Proof Overview. Assume we are given a $(1 - \epsilon)$ satisfiable instance $\mathcal{U} = (G, M, k)$ that satisfies the assumptions of theorem 8. We define a completely satisfiable game $\widetilde{\mathcal{U}} = (G, \widetilde{M}, k)$ (with value 1) that "corresponds" to $\mathcal{U}$ as follows:

Definition 11. *(Completion of a Game) Let $\mathcal{U} = (G, M, k)$ be as above and let $\mathcal{L} = \{L(u)\}$ be an assignment that satisfies $(1 - \epsilon)$ fraction of the constraints. Let $e = (u, v)$ be an edge with constraint $x_v = \pi_{uv}(x_u)$ that is not satisfied by this assignment, with labels $L(u)$, $L(v)$ on its endpoints. Construct a new game by changing the constraint on every such edge $e = (u, v)$ by replacing π_{uv} with a permutation $\tilde{\pi}_{uv}$ for which $\tilde{\pi}_{uv}(L(u)) = L(v)$. Let $\widetilde{\mathcal{U}} = (G, \widetilde{M}, k)$ be this new completely satisfiable game. We say that $\widetilde{\mathcal{U}}$ is a "completion" of $\mathcal{U}$.*

For the sake of the proof, we will assume that an *almost* satisfiable instance $\mathcal{U} = (G, M, k)$ is constructed as follows:

- Let $\widetilde{\mathcal{U}} = (G, \widetilde{M}, k)$ be a completion of $\mathcal{U}$.

- Let an adversary pick the ϵ fraction of edges that were unsatisfied in $\mathcal{U}$ and change their constraints back to the original ones. We can now think of M as a "perturbation" of $\widetilde{M}$ and $\mathcal{U}$ as a "perturbed" game of $\widetilde{\mathcal{U}}$.

Let W be the span of the eigenvectors of M with eigenvalue at least $(1 - \gamma)d$, for some $\gamma > 8\epsilon$. The algorithm simply looks at a set $\mathcal{S} \subseteq W$ of appropriately many candidate vectors and "reads-off" an assignment. We describe later on how to choose the set $\mathcal{S}$.

```
Recover-Solution_S(U)
```

- For each $x \in \mathcal{S}$, construct an assignment L_x by assigning to each vertex u, the index corresponding to the largest entry in the block $(x_{u1}, \ldots, x_{uk})$ i.e. $L_x(u) = \arg\max_i x_{ui}$.

- Out of all assignments L_x for $x \in \mathcal{S}$, choose the one satisfying the maximum number of constraints.

To choose $\mathcal{S}$, we look at the highest eigenvectors for the matrix $\widetilde{M}$. Those are the *assignment* eigenvectors, namely the characteristic vectors of the (perfectly) satisfying assignments of $\widetilde{\mathcal{U}}$ (since they all have eigenvalue equal to d). We will first observe that every such eigenvector y is close to some vector in W, and the length of the projection of y onto W depends on ϵ, γ. We then identify a set of *nice* vectors $\mathcal{N} \subseteq W$ such that the above algorithm works for any vector v close to some vector in $\mathcal{N}$. These are going to be the vectors that are close to the assignment eigenvectors. We then construct a set $\mathcal{S} \subseteq W$ of test vectors such that at least one vector $v \in \mathcal{S}$ is close to a vector in $\mathcal{N}$. $\mathcal{S}$ is simply an epsilon-net for W. Lastly, we go over all vectors in $\mathcal{S}$ until we find v. The running time of the algorithm will depend on the size of $\mathcal{S}$ which, for an appropriately defined epsilon-net, it is exponential in the dimension of W.

3.1.1 Eigenspaces and Labelings

Our first step is to identify an appropriate eigenspace W of M such that characteristic vectors of good labelings have large projection onto W.

For the matrix $\widetilde{M}$, let $\mathcal{L} = \{L(u)\}_{u \in V(G)}$ be a completely satisfying assignment. Define vector $y^{(\mathcal{L})}$ as

$$y_{ui}^{(\mathcal{L})} = \begin{cases} \frac{1}{\sqrt{n}} & \text{if} \quad i = L(u) \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that $y^{(\mathcal{L})}$ is an eigenvector of $\widetilde{M}$ with eigenvalue d. Since $y^{(\mathcal{L})}$ corresponds to the satisfying assignment $\mathcal{L}$, it can be seen as the characteristic vector of the assignment. We refer to such vectors as the "assignment" eigenvectors. Our next goal is to show that, for some appropriate choice of γ, the eigenspace W contains vectors close to an assignment vector.

Claim 12. *(Closeness) For every completely satisfying assignment $\mathcal{L}$ there is a unit vector $v_{\mathcal{L}} \in W$ such that $v_{\mathcal{L}} = \alpha y^{(\mathcal{L})} + \beta y^{(\mathcal{L})}{}_{\perp}$, with $\alpha > 0$ and $|\beta| \leq \sqrt{\frac{2\epsilon}{\gamma}}$. Here both $y^{(\mathcal{L})}$ and $y^{(\mathcal{L})}{}_{\perp}$ are unit vectors and $y^{(\mathcal{L})}{}_{\perp} \perp y^{(\mathcal{L})}$. By taking, for example, $\gamma \geq 200\epsilon$, we have $|\beta| \leq \frac{1}{10}$.*

Proof. We can easily see that $(y^{(\mathcal{L})})^T M y^{(\mathcal{L})} \geq d(1 - 2\epsilon)$. We can now write $y^{(\mathcal{L})} = a v_{\mathcal{L}} + b(v_{\mathcal{L}})_{\perp}$, with $a > 0$, $v_{\mathcal{L}} \in W$ and $(v_{\mathcal{L}})_{\perp} \in W^{\perp}$. We calculate:

$$(1 - 2\epsilon)d \leq (y^{(\mathcal{L})})^T M y^{(\mathcal{L})} = a^2 (v_{\mathcal{L}})^T M v_{\mathcal{L}} + b^2 ((v_{\mathcal{L}})_{\perp})^T M (v_{\mathcal{L}})_{\perp} \leq a^2 d + b^2(1 - \gamma)d$$

from which we get that $|b| \leq \sqrt{\frac{2\epsilon}{\gamma}}$.

Now, we can in turn express $v_{\mathcal{L}} = \alpha y^{(\mathcal{L})} + \beta y^{(\mathcal{L})}{}_{\perp}$, where $\alpha = \langle v_{\mathcal{L}}, y^{(\mathcal{L})} \rangle = a = \sqrt{1 - b^2} \geq \sqrt{1 - \frac{2\epsilon}{\gamma}}$ or, equivalently, $|\beta| = \sqrt{1 - \alpha^2} \leq \sqrt{\frac{2\epsilon}{\gamma}}$. $\qquad\square$

We have now managed to show that there exists a set of "nice" vectors $\mathcal{N} = \{v_{\mathcal{L}}\} \subseteq W$. The next claim shows that, if we knew the $v_{\mathcal{L}}$'s then we could set $\mathcal{S} = \mathcal{N}$ and the algorithm `Recover-Solution`$_{\mathcal{N}}(\mathcal{U})$ would return a highly satisfying assignment for $\mathcal{U}$.

Claim 13. *If x is a vector such that $x = \alpha y^{(\mathcal{L})} + \beta y^{(\mathcal{L})}{}_{\perp}$ for some $y^{(\mathcal{L})}$ with $\alpha > 0$ and $y^{(\mathcal{L})}{}_{\perp} \perp y^{(\mathcal{L})}$, then the coordinate $x_{uL(u)}$ is maximum in absolute value in at least $(1 - \frac{2\beta^2}{\alpha^2})n$ blocks.*

Proof. Within each block u, in order for coordinate $L(u)$ to be no longer the maximum one, it must happen that for some j

$$\alpha \frac{1}{\sqrt{n}} + \beta(y^{(\mathcal{L})}{}_{\perp})_{uL(u)} \leq \beta(y^{(\mathcal{L})}{}_{\perp})_{uj}$$

However, this gives

$$[(y^{(\mathcal{L})}{}_{\perp})_{uj} - (y^{(\mathcal{L})}{}_{\perp})_{uL(u)}]^2 \geq \frac{\alpha^2}{\beta^2 n} \Rightarrow$$

$$2(y^{(\mathcal{L})}{}_{\perp})_{uj}^2 + 2(y^{(\mathcal{L})}{}_{\perp})_{uL(u)}^2 \geq \frac{\alpha^2}{\beta^2 n} \Rightarrow$$

$$\left\| (y^{(\mathcal{L})}{}_{\perp})_u \right\|^2 \geq (y^{(\mathcal{L})}{}_{\perp})_{uj}^2 + (y^{(\mathcal{L})}{}_{\perp})_{uL(u)}^2 \geq \frac{\alpha^2}{2\beta^2 n}$$

Since $\left\| y^{(\mathcal{L})}{}_{\perp} \right\| = 1$, this can only happen for at most $\frac{2\beta^2}{\alpha^2}n$ blocks. $\qquad\square$

12

3.1.2 Eigenspace Enumeration: Finding the Set $\mathcal{S}$ of Test Vectors

Our next step is to show that the search for a good assignment can be done in time exponential to the number of large eigenvalues of M, namely the dimension of W.

Since we don't know the $y^{(\mathcal{L})}$ (if we did, we would be done), and the space W contains infinitely many unit vectors, we cannot identify $\mathcal{N}$. To get around this, we *discretize* W with an appropriate epsilon-net. If we let $w^{(0)}, \ldots, w^{(\dim(W)-1)}$ be an eigenbasis for W. We define the set $\mathcal{S}$ as

$$
\mathcal{S} \;=\; \left\{ v = \sum_{s=0}^{(\dim(W)-1)} \alpha_s w^{(s)} \;\mid\; \alpha_s \in \sqrt{\frac{2\epsilon}{\gamma\dim(W)}}\mathbb{Z}, \;\; \|v\| \le 1 \right\}
$$

It can be calculated (see, for instance [FO05]) that the number of points in the set $\mathcal{S}$ is at most $2^{O(\frac{\gamma}{\epsilon}\dim(W))}$.

To conclude the proof of theorem 8, it remains to show that $\mathcal{S}$ has a vector close to a nice vector $v_{\mathcal{L}} \in \mathcal{N}$. By construction, $\mathcal{S}$ contains at least one vector close to every vector in $\mathcal{N}$ and thus it also contains at least one vector v such that $v = \alpha v_{\mathcal{L}} + \beta(v_{\mathcal{L}})_{\perp}$ for some $\mathcal{L}$ and $|\beta| \le \sqrt{\frac{2\epsilon}{\gamma}}$. Together with claim 12, this implies that we can also write $v = ay^{(\mathcal{L})} + by^{(\mathcal{L})}_{\perp}$, with $|b| \le \sqrt{\frac{2\epsilon}{\gamma}} + \sqrt{\frac{2\epsilon}{\gamma}} = 2\sqrt{\frac{2\epsilon}{\gamma}}$. Thus, by claim 13, for this vector v, $\texttt{Recover-Solution}_{\mathcal{S}}(\mathcal{U})$ recovers an assignment which agrees with $y^{(\mathcal{L})}$ in $(1 - O(\frac{\epsilon}{\gamma-8\epsilon}))$ fraction of the blocks.

We will consider that a constrain is violated if one of the two following events happen: either the assignment we recovered does not agree with the initial perfectly satisfying assignment or the constraint is one of the (at most) ϵ fraction of constraints that were changed. It follows that the assignment which we recovered, violates constraints on edges that have total weight at most $O(\frac{\epsilon}{\gamma-8\epsilon})nd + \epsilon nd$. Since the total weight of constraints is $nd/2$, theorem 8 follows. The algorithm runs in $2^{O(\frac{\gamma}{\epsilon}\dim(W))} + T_W(M) = 2^{O(\frac{\gamma}{\epsilon}\dim(W))} + \text{poly}(n \cdot k)$ time.

3.2 Proof of Theorem 9

Let $\theta > 0$ such that $\gamma \ge \theta \ge \Omega(\epsilon\gamma)$. Let W be the span of eigenvectors of M with eigenvalue at least $(1-\theta)d$. Let Y be the span of eigenvectors of $\widetilde{M}$ with eigenvalue at least $(1-\gamma)d$. In particular, Y contains all the *assignment* eigenvectors, namely the characteristic vectors of the (perfectly) satisfying assignments of $\widetilde{\mathcal{U}}$ (since they all have eigenvalue equal to d). Let $Y_{\perp}$ be the orthogonal complement of Y.

For the proof, we will use theorem 8 together with a bound on the dimension of W. Roughly we show that the dimension of W is at most as large as the dimension of Y which, in turn, is at most k times the dimension of $S_{(1-\gamma)}$. To conclude, we apply theorem 8 for the eigenspace W.

We will assume w.l.o.g that the graphs we are dealing with are connected. Otherwise, we apply the results to each connected component individually.

Definition 14. *Let Φ be an eigenbasis for G. For every eigenvector $\phi = (\phi_u)_{u \in V} \in \Phi$ let $\widetilde{\phi}$ to be the kn-dimensional vector $\widetilde{\phi}_{ui} = \phi_u$, for $i = 1, \cdots, k$. Due to shift invariance, there exist k satisfying labelings of the unique games instance which we denote by $L_i, i = 1, \cdots, k$. Let $y^{(L_i)}$ be the characteristic vector of labeling L_i and let $\mathcal{E}(\phi) = \{\widetilde{\phi} \cdot y^{(L_i)}| \quad for \quad i = 0, \cdots, k-1\}$. Here $\cdot$ denotes entry-wise vector product. We define the following set of $k \times n$ vectors: $\widetilde{\Phi} = \bigcup_{\phi \in \Phi} \mathcal{E}(\phi).$*

13

We note that the set $\tilde{\Phi}$ consists of orthogonal unit vectors, since every two vectors in $\tilde{\Phi}$ have disjoint support. The next observation immediately follows from the fact that, due to the shift-invariance property of Γ-Max-Lin instances, $\widetilde{M}$ is the adjacency matrix of k disconnected copies of G.

Observation 15. $\tilde{\Phi}$ *is an eigenbasis of* $\widetilde{M}$. *Consequently, the dimension of* Y *is at most* $k \cdot D_S$, *where* D_S *is the dimension of* $S_{(1-\gamma)}$, *as per the assumption of the theorem.*

Lemma 16. *Assume* $\gamma \geq \theta \geq \Omega(\epsilon\gamma)$ *for some appropriately large constant. Then any unit-length vector* $w \in W$ *can be expressed as* $\alpha y + \beta y_\perp$ *where* $y \in Y$ *and* $y_\perp \in Y_\perp$ *are both unit length vectors, and* $|\beta| \leq O(\sqrt{\frac{\theta}{\gamma^3}})$. *By taking, for example,* $\gamma \geq \Omega(\sqrt[3]{\theta})$ *and, consequently,* $\gamma \geq \Omega(\sqrt{\epsilon})$, *we get* $|\beta| \leq \frac{1}{8}$.

Combining claims 15 and 16, we can proceed as follows: From claim 16 we obtain that W has dimension $\dim(W) \leq \dim(Y)$. Otherwise, we would find a vector orthogonal to all the vectors in Y which cannot be close to their span. From claim 15 we obtain $\dim(Y) = k \cdot D_S$.

To conclude the proof of theorem 9, we apply theorem 8 with W being the (at most) $k \cdot D_S$ dimensional eigenspace of M with eigenvalues $\geq (1 - \theta)d$.

Finally, it remains to argue about the running time of the algorithm. Since $\dim(W) \leq k \cdot D_S$, the number of points is exponential in kD_S. Hence, the algorithm runs in $2^{O(k \cdot D_S)} + T_W(M) = 2^{O(k \cdot D_S)} + \text{poly}(n \cdot k)$ time.

3.2.1 Perturbation of Eigenspaces: Proof of Claim 16

We next prove claim 16. Towards this end, we will apply some results from matrix perturbation theory. To find appropriate γ and θ such that the eigenspaces W and Y are close, we use the following claim which essentially appears in [DK70] as the $\sin\theta$ theorem and was used in [KT07].

Lemma 17. *Let* w *be a unit length eigenvector of* M *with eigenvalue* $\lambda \geq (1 - \theta)d$, *and let* λ_s *denote the largest eigenvalue of* $\widetilde{M}$ *which is smaller than* $(1 - \gamma)d$. *Then,* w *can be written as* $\alpha y + \beta y_\perp$ *with* $|\beta| \leq \frac{\|(\widetilde{M}-M)w\|}{(\lambda-\lambda_s)}$. *Here* $y \in Y$ *and* $y_\perp \in Y^\perp$ *are unit length vectors.*

As seen by the previous lemma, in order to prove that the space Y does not change by much due to the perturbation, we simply need to bound $\left\|(\widetilde{M} - M)w\right\|$. We will need the fact that w is somewhat "uniform" across blocks. To formalize this, let $\bar{w}$ be the n-dimensional vector such that $\bar{w}_u = \|w_u\|$ where w_u is the k-dimensional vector $(w_{u1}, \ldots, w_{uk})^T$. We then show that $\bar{w}$ is very close to a vector in $S_{(1-\gamma)}$.

Claim 18. *If* w *is a unit-length eigenvector of* M *with eigenvalue more than* $(1 - \theta)d$ *and* $\bar{w}$ *as above, then we can write* $\bar{w}$ *as* $\bar{w} = a\phi + b\phi_\perp$ *with* $|b| \leq \sqrt{\frac{\theta}{\gamma}}$ *and* $\phi \in S_{(1-\gamma)}$, $\phi_\perp \in S_{(1-\gamma)}^\perp$, *both having unit length.*

Proof. Since, w corresponds to a large eigenvalue, we have that

$$(1 - \theta)d \leq (w)^T M w = \sum_{u,v} w_u{}^T A_{uv} \cdot \Pi_{uv} w_v \leq \sum_{u,v} \|w_u\| A_{uv} \|w_v\| = (\bar{w})^T A \bar{w}$$

14

Here the second inequality follows from the fact that the operator norm of Π_{uv} is at most 1. Writing $\bar{w}$ as $a\phi + b\phi_\perp$, we get

$$(\bar{w})^T A\bar{w} \;\leq\; a^2 d + b^2(1-\gamma)d$$

$$\Rightarrow (1-\theta)d \;\leq a^2 d + b^2(1-\gamma)d \;\Rightarrow\; |b| \leq \sqrt{\frac{\theta}{\gamma}}$$

$\square$

Using the above, and the fact that the matrix $\widetilde{M}$ is only perturbed in ϵ fraction of the edges, we can now bound $\left\|(\widetilde{M} - M)w\right\|$ as follows.

Claim 19. $\left\|(\widetilde{M} - M)w\right\| \leq O\left(\sqrt{\frac{\theta}{\gamma}}\right) d$

Proof. Define the $n \times n$ matrix R as $R_{uv} = w_{uv} \leq 1$ when the block $(\widetilde{M} - M)_{uv}$ has any non-zero entry, and $R_{uv} = 0$ otherwise. Note that if $(\widetilde{M} - M)_{uv}$ is non-zero, then it must be the (scaled) difference of two permutation matrices. Thus, for all v we have $\left\|(\widetilde{M} - M)_{uv} w_v\right\| \leq 2R_{uv} \|w_v\|$. We calculate:

$$\left\|(\widetilde{M} - M)w\right\| \;=\; \sqrt{\sum_u \left\|\sum_v (\widetilde{M} - M)_{uv} w_v\right\|^2} \;\leq\; \sqrt{\sum_u \left(\sum_v \left\|(\widetilde{M} - M)_{uv} w_v\right\|\right)^2}$$

$$\leq\; \sqrt{\sum_u \left(\sum_v 2R_{uv} \|w_v\|\right)^2}$$

$$=\; 2\|R\bar{w}\|$$

To estimate $\|R\bar{w}\|$, we break it up as

$$\|R\bar{w}\| \leq |a| \|R\phi\| + |b| \|R\phi_\perp\|$$

Since each row of R has total sum of entries at most d, $|b|\|R\phi_\perp\| \leq \sqrt{\frac{\theta}{\gamma}}d$. Also,

$$\|R\phi\| = \sqrt{\sum_u |\sum_v R_{uv}\phi_v|^2} \leq \sqrt{\sum_u \left(\sum_v R_{uv}|\phi_v|\right)^2} \leq \frac{C}{\sqrt{n}}\sqrt{\sum_u \left(\sum_v R_{uv}\right)^2}$$

Since R has a total sum of entries at most $\epsilon n d$, this expression is maximized when it has d 1s in ϵn rows. This gives $\|R\phi\| \leq C\sqrt{\epsilon}d$. And, putting everything together we obtain

$$\left\|(\widetilde{M} - M)w\right\| \;\leq\; 2C\sqrt{\epsilon}d + 2\sqrt{\frac{\theta}{\gamma}}d \;\leq\; O\left(\sqrt{\frac{\theta}{\gamma}}\right) d$$

We note that the $O(\cdot)$ in the above expression depends on C. We obtained the last inequality by using the assumption of the lemma $\theta \geq \Omega(\epsilon\gamma)$. $\square$

15

Combining the above bound with claim 17, we get that any unit-length vector $w \in W$ can be expressed as $\alpha y + \beta y_\perp$ where $y \in Y$ and $|\beta| \leq O(\sqrt{\frac{\theta}{\gamma}} d \cdot \frac{1}{(1-\theta)d - \lambda_s})$. Recall that λ_s was smaller than $(1-\gamma)d$, which implies

$$|\beta| \leq O\left(\sqrt{\frac{\theta}{\gamma}} \cdot \frac{1}{\gamma - \theta}\right) \tag{1}$$

To conclude the proof of claim 16 we take $\theta << \gamma$. We calculate:

$$|\beta| \leq O\left(\sqrt{\frac{\theta}{\gamma}} \cdot \frac{1}{\gamma - \theta}\right) \leq O\left(\sqrt{\frac{\theta}{\gamma^3}}\right) \tag{2}$$

It is sufficient to consider $\gamma \geq \Omega(\sqrt[3]{\theta})$ and observe that this also implies, by the assumption $\theta \geq \Omega(\epsilon\gamma)$ that $\gamma \geq \Omega(\sqrt{\epsilon})$.

3.3   Generalizing to Non-Regular Graphs

It remains to show that the above results hold for non-regular graphs. We will consider eigenvectors and eigenvalues of the Laplacian matrix of the label-extended graph. Let G and M as before. Let D be the $nk \times nk$ diagonal matrix with block $D_{uu} = deg(u) \cdot I_k$, where I_k is the $k \times k$ identity matrix. The Laplacian of the label-extended graph is the matrix $L_M = D - M$. Let d be the average degree of G, and therefore of M, namely $d = \frac{\sum_{u \in V} d_u}{n}$.
We re-state the main theorem 8 to capture the non-regular case.

Theorem 20. *Let $\mathcal{U} = (G, M, k)$ be a $(1 - \epsilon)$ satisfiable instance of Unique Games and W the eigenspace of L_M with eigenvalues less than γd, for $\gamma \geq 8\epsilon$. There is an algorithm that runs in time $2^{O(\frac{\gamma}{\epsilon} dim(W))} + poly(n \cdot k)$ and finds an assignment that satisfies at least $(1 - O(\frac{\epsilon}{\gamma - 8\epsilon} + \epsilon))$ fraction of the constraints.*

Proof. Let W be the span of the eigenvectors of L_M with eigenvalue less than γd, for some $\gamma \geq 8\epsilon$. We again recover highly satisfying assignments using our algorithm Recover-Solution$_{\mathcal{S}}(\mathcal{U})$. Similarly to the previous section, the set $\mathcal{S} \subseteq W$ will be a discretization of our new W.

All we need to show is the following analog of claim 12. We use the same notation as in section 3.1.

Claim 21. *For every completely satisfying assignment $\mathcal{L}$ there is a unit vector $v_{\mathcal{L}} \in W$ such that $v_{\mathcal{L}} = \alpha y^{(\mathcal{L})} + \beta y^{(\mathcal{L})}_\perp$, with $|\beta| \leq \sqrt{\frac{2\epsilon}{\gamma}}$. Here $y^{(\mathcal{L})}$ and $y^{(\mathcal{L})}_\perp$ are both unit length vectors and $y^{(\mathcal{L})}_\perp \perp y^{(\mathcal{L})}$. By taking, for example, $\gamma \geq 200\epsilon$, we have $|\beta| \leq \frac{1}{10}$.*

Proof. (Of Claim)

We first observe that the *assignment* eigenvectors $y^{\mathcal{L}}$ are eigenvectors of $L_{\widetilde{M}}$ with eigenvalue 0.

We next see that

$$
\begin{aligned}
(y^{(\mathcal{L})})^T L_M y^{(\mathcal{L})} &= (y^{(\mathcal{L})})^T (D - M) y^{(\mathcal{L})} = (y^{(\mathcal{L})})^T (D - \widetilde{M} + \widetilde{M} - M) y^{(\mathcal{L})} \\
&= (y^{(\mathcal{L})})^T (\widetilde{M} - M) y^{(\mathcal{L})} \leq \frac{2\epsilon 2|E|}{n} = 2\epsilon d
\end{aligned}
$$

16

We can now write $y^{(\mathcal{L})} = av_{\mathcal{L}} + b(v_{\mathcal{L}})_{\perp}$, with $v_{\mathcal{L}} \in W$ and $(v_{\mathcal{L}})_{\perp} \in W^{\perp}$. We calculate:

$$2\epsilon d \geq (y^{(\mathcal{L})})^T L_M y^{(\mathcal{L})} = a^2 (v_{\mathcal{L}})^T L_M v_{\mathcal{L}} + b^2 ((v_{\mathcal{L}})_{\perp})^T L_M (v_{\mathcal{L}})_{\perp} \geq b^2 \gamma d$$

from which we get that $|b| \leq \sqrt{\frac{2\epsilon}{\gamma}}$.

Now, we can in turn express $v_{\mathcal{L}} = \alpha y^{(\mathcal{L})} + \beta y^{(\mathcal{L})}_{\perp}$, where $\alpha = \langle v_{\mathcal{L}}, y^{(\mathcal{L})} \rangle = |a| = \sqrt{1 - b^2} \geq \sqrt{1 - \frac{2\epsilon}{\gamma}}$
or, equivalently, $|\beta| = \sqrt{1 - \alpha^2} \leq \sqrt{\frac{2\epsilon}{\gamma}}$. $\qquad\square$

To conclude the proof of the theorem, we observe that the analog of claim 13 follows immediately. We define $\mathcal{S}$ as in section 3.1.2, namely

$$\mathcal{S} \;=\; \left\{ v = \sum_{s=0}^{(\dim(W)-1)} \alpha_s w^{(s)} \;\;\middle|\;\; \alpha_s \in \sqrt{\frac{2\epsilon}{\gamma \dim(W)}} \mathbb{Z}, \;\; \|v\| \leq 1 \right\}$$

where $w^{(i)}$ are now eigenvectors of our new W. The theorem follows.

$\qquad\square$

4   Solving Unique Games on the Khot-Vishnoi Instance

In this section, we show that, when run on the Khot-Vishnoi integrality gap instance, our main algorithm runs in quasi-polynomial time and correctly decides the (un)satisfiability of the instance. Namely, we show the following:

Claim 22. *Our main algorithm (as in theorem 8) when given input the integrality gap instance of Khot and Vishnoi as described in the preliminaries section, with label-extended graph $\widetilde{\mathcal{KV}}_{n,\epsilon}$, runs in time $n^{poly(\log n)}$ and correctly decides that the instance is highly unsatisfiable. Here $N = 2^n$ the number of nodes of H_n and n is the alphabet size.*

The proof goes by showing that the eigenspace W as per theorem 8 has relatively low dimension. The theorem guarantees that exhaustive search in W would find a good assignment if such assignment existed and consequently, failure to find such an assignment, will correctly classify the Khot-Vishnoi instance as unsatisfiable.

The rest of this section is devoted to the proof of claim 22. We use the equivalent definition of $\widetilde{\mathcal{KV}}_{n,\epsilon}$ as a Cayley graph of H_n which is an "ϵ-perturbed" version of the hypercube graph. We will need the following claim for the spectrum of $\widetilde{\mathcal{KV}}_{n,\epsilon}$. We denote the adjacency matrix of $\widetilde{\mathcal{KV}}_{n,\epsilon}$ by $M_{\mathcal{KV}}$.

Claim 23. *The following is true for the spectrum of $\widetilde{\mathcal{KV}}_{n,\epsilon}$.*

- *The eigenvectors are the characters of the group H_n.*

- *The eigenvalue that corresponds to χ_ω is $(1 - 2\epsilon)^r n$, where $r = |\omega|$ is the hamming weight of ω, and appears with multiplicity $C_r = \binom{n}{r}$.*

Proof. The first item above is immediate from the definition of the graph as it appears in the preliminaries section.

For the second item, we just need to calculate the eigenvalues corresponding to χ_ω or, equivalently, the quadratic form

$$\frac{1}{\sqrt{N}}(\chi_\omega)^T M_{\mathcal{KV}}(\frac{1}{\sqrt{N}}\chi_\omega) = \frac{1}{N}\sum_{x,y\in H_n}\chi_\omega(x)M_{\mathcal{KV}}(x,y)\chi_\omega(y)$$

$$= \frac{1}{2^n}\sum_{x,y\in H_n}\chi_\omega(x)n\epsilon^{|x+y|}(1-\epsilon)^{n-|x+y|}\chi_\omega(y)$$

$$= \frac{n}{2^n}\sum_{x,y\in H_n}\chi_\omega(x+y)\epsilon^{|x+y|}(1-\epsilon)^{n-|x+y|}$$

$$= \frac{n}{2^n}2^n\sum_{z\in H_n}\chi_\omega(z)\epsilon^{|z|}(1-\epsilon)^{n-|z|}$$

$$= n\mathbf{E}_{z\in\epsilon\mathcal{F}}\big[\chi_\omega(z)\big] = \prod_{i=1}^{n}\mathbf{E}_{z_i\in\{0,1\}}[(-1)^{\omega_i\cdot z_i}] = n(1-2\epsilon)^{|\omega|}$$

To conclude the proof, we need to argue about the multiplicity of each eigenvalue or, equivalently, the number of characters χ_ω of a given hamming weight r. It is easily seen that there are $C_r = \binom{n}{r}$ characters of H_n that correspond to hamming weight r. □

We will directly apply theorem 8. For this purpose we need to calculate the dimension of the eigenspace W of $\widetilde{\mathcal{KV}}_{n,\epsilon}$ for some appropriate $\gamma \geq \Omega(\sqrt{\epsilon})$.

Lemma 24. *Let* $\gamma = C\sqrt{\epsilon}$ *for some sufficiently large constant C. The dimension of the eigenspace W of the graph $\widetilde{\mathcal{KV}}_{n,\epsilon}$ with eigenvalues* $\geq (1-\gamma)$ *is at most* $D_W \leq \frac{1}{\epsilon}n^{O(\frac{1}{\epsilon})} = poly(n) = poly(\log N)$.

Proof. We first need to identify an upperbound on r such that $(1-2\epsilon)^r \leq (1-\gamma)$, for $\gamma = C\sqrt{\epsilon}$. It is enough to take $r = \frac{\log 1-\gamma}{\log 1-2\epsilon}$. In order to approximate r, we use the Taylor series expansion:

$$\log\frac{1}{1-x} = \sum_{i=1}^{\infty}\frac{x^i}{i}$$

And for ϵ, γ small enough we get

$$r = \frac{\log 1-\gamma}{\log 1-2\epsilon} = \frac{\log\frac{1}{1-\gamma}}{\log\frac{1}{1-2\epsilon}} \approx \frac{\gamma}{\epsilon} \leq C\frac{1}{\sqrt{\epsilon}} \leq \tilde{C}\frac{1}{\epsilon}$$

for some appropriately large constant $\tilde{C}$. The dimension of W can now be bounded, using claim 23, as follows: $D_W \leq \sum_{r=0}^{\tilde{C}1/\epsilon}\binom{n}{r} \leq \frac{1}{\epsilon}n^{O(\frac{1}{\epsilon})}$. □

We conclude by arguing, according to theorem 8, that our algorithm will run in time bounded by $2^{D_W} = 2^{\frac{1}{\epsilon}\cdot n^{O(\frac{1}{\epsilon})}} = N^{\frac{1}{\epsilon}\cdot(\log N)^{O(\frac{1}{\epsilon})}} = N^{(poly(\log N))}$. It will fail to find a highly satisfying assignment and thus decide that the above instance is highly unsatisfiable. Here we used the fact that for the graph in question, $T_{S_W}(M) \leq 2^{D_W}$.

Acknowledgements

References

[ABS10]   Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Proceedings of the 51st IEEE Symposium on Foundations of Computer Science*, 2010.

[AIMS09]   Sanjeev Arora, Russell Impagliazzo, William Matthews, and David Steurer. Improved algorithms for unique games via divide and conquer, 2009. manuscript.

[AKK+08]   Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth Vishnoi. Unique Games on expanding constraint graphs are easy. In *Proceedings of the ACM Symposium on the Theory of Computing*, volume 40, pages 21–28, 2008.

[Art91]   Michael Artin. *Algebra*. Prentice Hall, 1991.

[Aus10]   Per Austrin. Towards sharp inapproximability for any 2-CSP. *SIAM J. Comput.*, 39(6):2430–2463, 2010.

[CKK+06]   Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. In *Computational Complexity*, volume 15(2), pages 94–114, 2006.

[CMM06a]   Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the ACM Symposium on the Theory of Computing*, volume 38, pages 205–214, 2006.

[CMM06b]   Eden Chlamtac, Konstantin Makarychev, and Yury Makarychev. How to play unique games using embeddings. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, volume 48, pages 687–696, 2006.

[DK70]   Chandler Davis and W.M. Kahan. The rotation of eigenvectors by a perturbation III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[FO05]   Uriel Feige and Eran Ofek. Spectral techniques applied to sparse random graphs. *Random Structures and Algorithms*, 27(2):251–275, 2005.

[GT06]   Anupam Gupta and Kunal Talwar. Approximating unique games. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, volume 24, pages 99–106, 2006.

[Kas02]   Petteri Kaski. Eigenvectors and spectra of cayley graphs, 2002. Available from: http://www.tcs.hut.fi/Studies/T-79.300/2002S/esitelmat/kaski-paper-020506.pdf.

[Kho02]   Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the ACM Symposium on the Theory of Computing*, volume 34, pages 767–775, 2002.

[KKMO04] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable CSPs? In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, volume 45, pages 146–154, 2004.

[KR03] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. In *Proceedings of the IEEE Conference on Computational Complexity*, volume 18, pages 379–386, 2003.

[KT07] Alexandra Kolla and Madhur Tulsiani. Playing Unique Games using graph spectra, 2007. Manuscript.

[KV05] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, volume 46, pages 53–62, 2005.

[Lin07] Nati Linial. Introduction to fourier analysis, 2007. Available from: http://www.cs.huji.ac.il/ nati/PAPERS/uw/lect01.pdf.

[MM09] Yury Makarychev and Konstantin Makarychev. How to play Unique Games on expanders, 2009. ECCC Report No. 21.

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the ACM Symposium on the Theory of Computing*, volume 40, pages 245–254, 2008.

[RS09] Prasad Raghavendra and David Steurer. Integrality gaps for strong SDP relaxations of Unique Games. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science*, pages 575–585, 2009.

[RS10] P. Raghavendra and D. Steurer. Graph expansion and the Unique Games conjecture. In *Proceedings of the 42nd ACM Symposium On The Theory Of Computing*, volume 42, pages 755–764, 2010.

[Spi09] Daniel Spielman. Cayley graphs, 2009. Letcure Notes.

[Tre05] Luca Trevisan. Approximation algorithms for unique games. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, volume 46, pages 197–205, 2005.