Spectral Algorithms and Unique Games

Jeremy McMahan

December 17, 2019

1 Introduction

Graph problems are many of the most interesting and most difficult problems in computer science. Many graph-theoretic techniques have been developed to tackle these challenging problems, though there are other useful ways to solve them. In particular, we can use linear algebraic ideas from spectral graph theory to develop efficient and elegant solutions to many such problems. Spectral Graph Theory boils down to choosing a matrix associated with a graph that models properties of the graph thatwe care about, and then using the linear algebraic structure of this matrix, usually its spectra, to derive solutions to the problem at hand. We will show many examples of how spectral techniques can lead to interesting algorithms and new insights on the structure of these problems. Most notably, we will discuss spectral techniques for approximating the UniqeGames problem. In fact, the current best known algorithm for approximating UniqueGames is based on spectral techniques.

The Unique Games Conjecture is a core open problem in theoretical computer science. It involves the hardness of approximating a gap version of UniqueGames. An instance of UniqueGames consists of a graph $G = (V, E)$ along with bijections for each edge, $\pi_{(u,v)} : [k] \to [k]$, that we can think of as constraints on each edge. We call the number k the alphabet size of the game. The goal of the game is to construct a labeling function $\ell : V \to [k]$ that satisfies as many constraints as possible. Specifically, we say that the constraint on edge (u, v) is satisfied if $\pi_{(u,v)}(\ell(u)) = \ell(v)$ and the goal is to maximize the total fraction of constraints satisfied. If the game is completely satisfiable, i.e. there is a labeling that satisfies every constraint, then finding such a labeling is easy. Thus, the Unique Games Conjecture deals with a gap version of the problem. Namely, if the game is only $(1 - \epsilon)$ satisfiable, then it is conjectured that it is NP-hard to compute a labeling that satisfies a δ fraction of the constraints. There are many equivalent formulations of the conjecture, but we will mostly focus on the following formulation [4]:

Unique Games Conjecture: For any $\epsilon, \delta > 0$ there is some function $k(\epsilon, \delta)$ so that for any $k > k(\epsilon, \delta)$ it is NP-hard to distinguish between instances of UniqueGames with alphabet size k that are at least $(1 - \epsilon)$-satisfiable from those that are at most δ-satisfiable.

The Unique Games Conjecture is so important because its truth would immediately imply hardness of approximation results for many famous NP-hard problems [3]. For many of these problems, it further shows that the current best known approximation algorithm is in fact optimal. The conjecture also would demonstrate integrality gap instances for many LP or SDP approaches problems which implies such approaches won't work for those problems. However, in this paper, we will consider promising spectral techniques that could lead to an effecient approximation algorithm for UniqueGames, which would refute this conjecture. In particular, the techniques we illustrate avoid many of the issues of standard SDP approaches to UniqueGames and even work in quasi-polynomial time on the classical integrality gap instance for those SDPs [4].

2 Preliminaries

Let $G = (V, E)$ be an undirected graph. When G is weighted we associate with G a function $w_G : E \to R$ that outputs the weight of each edge. We will always use n to denote the total number of vertices of the graph. Since we will only consider undirected graphs, we will denote an edge by (u, v) for convenience of notation with the understanding that there is no direction associated with the edge. Whenever we need to discuss ordered pairs of vertices (which need not be edges) we will use the notation $u, v \in V$. We assume familiarity with common combinatorial properties of graphs such as connectivity. We define an independent set in a graph as any subset of nodes having no edges between them and define a clique to be any subset of nodes having all possible edge connections between them. Then, we define the independence number of G, $\alpha(G)$, to be the size of a largest independent set in G and the clique number, $\omega(G)$, to be the size of a largest clique in G.

We also assume a familiarity with linear algebra and eigenvalue theory. Given a real symmetric $n \times n$ matrix, M, M has n real-valued eigenvalues and a set of n mutually orthonormal eigenvectors. A vector $0 \neq x \in R^n$ is an eigenvector for M if $Mx = \lambda x$ for some $\lambda \in R$, which is the corresponding eigenvalue to x. The quanitity $x^T Mx = \sum_{i,j} M_{i,j} x_i x_j$ is called a quadratic form of M with respect to x, and $\frac{x^T Mx}{x^T x}$ is the Rayleigh quotient of M with respect to x. We note that if x is an eigenvector of M of eigenvalue λ, then $\frac{x^T Mx}{x^T x} = \lambda$. More generally, if λ_i is the ith smallest eigenvalue of M, we have that $\lambda_i = \min_{x \perp S_{i-1}} \frac{x^T Mx}{x^T x}$, where S_{i-1} is the set of eigenvectors corresponding to $\lambda_1, \ldots, \lambda_{i-1}$. Also, we have that $\lambda_n = \max_x \frac{x^T Mx}{x^T x}$. Lastly, we note that for any subspace $W \subseteq R^n$, that $R^n = W + W^\perp$ where addition denotes direct sum, which is the set of all sums of vectors from the first set with vectors from the second, and $W^\perp$ is the orthogonal complement of W containing all vectors that are orthogonal to every vector in W. If $x \in W$, we will typically denote by $x_\perp$ any vector that is orthogonal to x and so is an element of $W^\perp$. Lastly, we assume all norms are the Euclidean norm.

We will also use the Iverson bracket notation where appropriate to convert a boolean predicate, P, into a number in $\{0, 1\}$. Specifically, define

$$[P] = \begin{cases} 1 \text{ if } P \text{ is true} \\ 0 \text{ o.w.} \end{cases}$$

3 Spectral Graph Theory

Many important combinatorial properties of a graph can be revealed through the structure of different matrices associated with the graph. In particular, the spectra of the adjacency and Laplacian matrices can tell us about the connectivity, cuts, and even chromatic number of the graph. Thus, spectral algorithms that exploit the eigenvalues of these matrices can lead to elegant solutions to many complicated graph problems. In this section we will look at linear algebraic properties of these matrices and their relation to the graph's combinatorial properties. Furthermore, we will use some of these techniques to develop algorithms for fundamental graph problems. Later in the paper, we will give a more sophisticated spectral algorithm that approximates UniqueGames.

3.1 Spectral Basics

The most natural matrix associated with a graph $G = (V, E)$ is its adjacency matrix that models which edges exist in the graph. Formally, A_G is a matrix with rows and columns indexed by the vertices of the graph and entries defined by

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

However, an arguably more useful matrix associated with the graph is the Laplacian matrix, L_G, which naturally models the connectivity and structure of cuts the graph [5]. Formally, we define $L_G = D_G - A_G$ where D_G is a diagonal matrix of the degrees of each vertex. Equivalently, we can specify the Laplacian by

$$L_G(u, v) = \begin{cases} d_G(u) & \text{if } u = v \\ -1 & \text{if } (u, v) \in E \\ 0 & \text{o.w.} \end{cases}$$

Note each of these matrices can be easily generalized for the case of weighted graphs. In that case, we can simply replace each 1 in the above definitions with the weight of the edge $w_G(u, v)$ and replace the degree with the weighted degree, $d_G(u) = \sum_{(u,v)\in E} w_G(u, v)$. Most of the results we will discuss apply equally well to weighted graphs though we will focus on the unweighted case for simplicity.

We will commonly construe matrices and vectors with functions and consequently use function notation when describing these quantities as commonplace

in literature about spectral graph theory. In particular, we will sometimes consider these matrices as linear operators mapping a vector to another vector, $M : x \to Mx$, or as a function mapping a vector to a real number through its quadratic form, $M : x \to x^T Mx$. Below, we summarize a few useful formulas for the matrices we will consider using this perspective. Let $x \in R^V$ be a real valued vector and $u \in V$ an arbitrary vertex of G. Then,

$$(A_G x)(u) = \sum_{(u,v)\in E} w_G(u,v)x(v)$$

$$(L_G x)(u) = \sum_{(u,v)\in E} w_G(u,v)(x(u) - x(v))$$

$$x^T L_G x = \sum_{(u,v)\in E} w_G(u,v)(x(u) - x(v))^2$$

From the last equation, we see the quadratic form is always non-negative as it is a sum of squares, and so L_G is positive semi-definite. Consequently, all the eigenvalues of L_G are at least 0. On the other hand, A_G may have negative eigenvalues.

Since A_G and L_G are $n \times n$ real symmetric matrices, we know by the spectral theorem that they both have n real eigenvalues and a set of n mutually orthogonal unit eigenvectors. In particular, let $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of L_G with corresponding mutually orthogonal unit eigenvectors $\nu_1, \ldots, \nu_n$, and $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$ be the eigenvalues of A_G with corresponding mutually orthogonal unit eigenvectors $\phi_1, \ldots, \phi_n$.

In general, the eigenvalues of both matrices are related. This is most easily seen when G is d-regular since then we have $L_G = D_G - A_G = dI - A_G$. This implies that $\lambda_i = d - \mu_i$ for all i. So, the largest eigenvalue of the Laplacian corresponds to the smallest eigenvalue of the adjacency and vice versa.

To develop intuition on why the Laplacian is important, lets consider how its quadratic form behaves on cuts in the graph. In particular, define the characteristic vector of a subset of nodes, $S \subseteq V$, as χ_S where

$$\chi_S(u) = \begin{cases} 1 & \text{if } u \in S \\ 0 & \text{o.w.} \end{cases}$$

we see that

$$\begin{aligned} \chi_S^T L_G \chi_S &= \sum_{(u,v)\in E} (\chi_S(u) - \chi_S(v))^2 \\ &= \sum_{\substack{(u,v)\in E \\ u,v\in S}} (1-1)^2 + \sum_{\substack{(u,v)\in E \\ u,v\notin S}} (0-0)^2 + \sum_{\substack{(u,v)\in E \\ u\in S, v\notin S}} (1-0)^2 \\ &= |\partial(S)| \end{aligned}$$

where $\partial(S) = \{(u,v) \in E | u \in S, v \notin S\}$ is the boundary of S, which contains exactly the edges crossing the cut induced by S.

3.2 Connectivity and Cuts

To start, we will show how these matrices can give us information on the connectivity properties of the graph. One important aspect of connectivity is determining the number of walks between two vertices. If we want to consider walks of length 0, the problem is trivial as the only vertices that can be reached from a vertex, s, in no steps is s itself. Similarly, if we allow only walks of length 1, the only vertices s can reach are those it has edges to, which is exactly encoded in the adjacency matrix. In particular, if we want to know the number of walks between s and t that take 1 step, we merely need to compute $A_G(s,t)$. Now, let us consider what $A_G^2(s,t)$ represents.

$$A_G^2(s,t) = \sum_{w \in V} A_G(s,w) A_G(w,t) = \sum_{w \in V} [(s,w) \in E \wedge (w,t) \in E]$$

The last quantity counts the number of intermediate vertices that could be on a $s-t$ path of length 2, which is exactly the number of length 2 $s-t$ walks. If we assume inductively that $A_G^{k-1}(s,t)$ is the number of $s-t$ paths that take exactly $k-1$ steps, we see that

$$A_G^k(s,t) = \sum_{w \in V} A_G^{k-1}(s,w) A_G(w,t) = \sum_{w \in V} |\{s{-}t \text{ walks of length } k{-}1\}| [(w,t) \in E]$$

which is exactly counting the number of $s-t$ walks of length k as every such path must consist of a $s-w$ walk of length $k-1$ for some w followed by the edge (w,t). Thus, we see that $A_G^k(s,t)$ is the number of $s-t$ walks in G of length exactly k. Also, if we add the identity matrix to A_G, we give a walk the option of staying at the current vertex (since we are essentially adding self loops to the graph). Consequently, an analogous argument shows that $(I + A_G)^k(s,t)$ is the number of $s-t$ walks in G of length at most k. Since any path in a graph can have length at most $n-1$, we can determine whether s and t are connected by checking that $A_G^{n-1}(s,t) > 0$. Using repeated squaring, this gives a fairly efficient algorithm for determining whether two vertices are connected.

Another important problem involving connectivity is counting the number of minimal sized connected subgraphs in a graph. In other words, we want to count the number of spanning trees in the graph. This problem, can also be easily solved using spectral methods, which is encapsulated in the famous matrix tree theorem. In particular, we consider a version of this theorem for connected graphs that immediately gives us a way to compute the number of spanning trees from the spectra of the Laplacian matrix.

Theorem 3.1 (Matrix Tree Theorem)**.** *The number of spanning trees contained in a connected graph G is exactly $\frac{1}{n}\lambda_2 \dots \lambda_n$*

This Theorem highlights the surprising power of spectral techniques as the eigenvalues of the Laplacian matrix are intimately related and in fact exactly capture the number of spanning trees in the graph. We will not prove this theorem though we will show in detail why the eigenvalues of these matrices can be related to several other properties of graphs.

Let's turn to the simplest question of whether a graph is connected or not. Recall that the smallest eigenvalue of the Laplacian matrix, λ_1, is defined to be $\min_x \frac{x^T L_G x}{x^T x} \geq 0$. Suppose G has components $G_1, \ldots, G_k$, and consider the vectors x_i that assign 1 to every vertex in G_i and 0 otherwise. In other words, $x_i = \chi_{V_{G_i}}$. We see these vectors give $x_i^T L_G x_i = 0$ since the boundary of V_{G_i} is 0 being that it is a connected component and so has no edges to the rest of the graph. Consequently, the Rayleigh quotient is 0 for these x_i. Also, each of these x_i are orthogonal since no component shares any vertex by definition. Hence, we have $\lambda_1 = \ldots = \lambda_k = 0$. Also, any vector, x, orthogonal to all the vectors that are constant on a single component must set a different value for at least two vertices, u, v, in the same component. Since u, v are connected but have different x value, it must be at some point along some $u - v$ path that the value of x on the endpoints of an edge are different and so have a positive difference. This implies that the Rayleigh Quotient is positive for any such x and so $\lambda_{k+1} > 0$. Thus, if G has k components, the eigenvalue 0 has multiplicity exactly k for the Laplacian matrix. The converse also holds. In particular, this means G is connected if and only if $\lambda_2 > 0$. In general, λ_2 is called the algebraic connectivity for connected graphs and measures how well the graph is connected.

Another useful graph property we can consider is the isoperimetric number, which is also called edge expansion, of the graph [8]. This number models expansion properties of the graph, namely, how large is the ratio of edges leaving some subgraph to the number of vertices in that subgraph. This notion is very useful for expanders and many other clustering and cut like problems on graphs. Formally, the isoperimetric ratio of a subset of nodes is $\theta(S) = \frac{|\partial(S)|}{|S|}$ and the isoperimetric number for a graph is the minimal such ratio over strict non-empty subsets of nodes $\theta_G = \min_{\emptyset \neq S \subset V} \theta(S)$. Since each set implicitly also includes its complement in the boundary we can further simplify the isopermetric number to be $\theta_G = \min_{|S| \leq \frac{n}{2}} \theta(S)$. We will show a bound on this number using the algebraic connectivity. This gives insight on the relationship between the algebraic connectivity of the graph and its expansion properties while also giving a simple approximation algorithm for bounding the isoperimetric number.

Theorem 3.2. $\theta_G \geq \frac{\lambda_2}{2}$

Proof. We will in fact prove the stronger statement that for any strict non-empty subset of nodes, S, $\theta(S) \geq \lambda_2(1 - s)$ where $s = \frac{|S|}{n}$. By definition of λ_2 as $\min_{x \perp 1} \frac{x^T L_G x}{x^T x}$, we know that $x^T L_G x \geq \lambda_2 x^T x$ for any x orthogonal to the constant vectors. The Rayleigh quotient of L_G with respect to the characteristic vector, χ_S, is exactly the isoperimetric ratio of S, $\theta(S)$. This vector need not be orthogonal to the constant vectors, but we can perturb it slightly to make

accomplish this. Specifically, if we consider $y = x_S - s$ defined by

$$y(u) = \begin{cases} 1 - s \text{ if } u \in S \\ -s \text{ o.w.} \end{cases}$$

we have that

$$y^T L_G y = \sum_{\substack{(u,v)\in E \\ u\in S, v\notin S}} ((1-s)-(-s))^2 = \sum_{\substack{(u,v)\in E \\ u\in S, v\notin S}} 1 = |\partial(S)|$$

Also, we have that

$$y^T y = |S|(1-s)^2 + (n-|S|)s^2 = |S| - 2|S|s + |S|s^2 - |S|s^2 + ns^2 = |S|(1-s)$$

Putting these equalities together gives a Rayleigh quotient of value $\frac{|\partial(S)|}{|S|(1-s)}$. Lastly, we show y is orthogonal to the constant vectors.

$$y^T 1 = |S|(1-s) - (n-|S|)s = |S| - s|S| + s|S| - ns = |S| - |S| = 0$$

Since y is orthogonal to the constant vectors, we then know $\frac{|\partial(S)|}{|S|(1-s)} \geq \lambda_2$. Multiplying both sides by $(1-s)$ gives the desired bound. Using the fact that $|S| \leq \frac{n}{2}$ and so $(1-s) \geq \frac{1}{2}$ gives the theorem. □

Computing the edge expansion is also closely related to computing a bisection of minimum size [1]. A bisection is a partition of the graph into two equal sized subsets, and its size is the size of the cut this partition induces. Finding a minimum size bisection is hard, but since we can capture information about cuts using the Laplacian eigenvalues, there is hope for a spectral approach. In particular, if G is connected, we know the constant vectors are eigenvectors of the smallest eigenvalue. Thus, it must be that ν_2 is orthogonal to all constant vectors and so must be positive on some vertices and negative on others. Also, since ν_2 minimizes the Rayleigh quotient over all non-constant vectors, we know that the change of ν_2 over endpoints of an edge are not too large in general. In particular, vertices clustered together with many edges between them likely have the same sign under ν_2, so using the sign of ν_2 to determine a bisection is an effective spectral approach. For the best case scenario, consider the dumbbell graph. In this case, ν_2 could be a small positive constant on all of the left clique and the same constant though negative on the right clique. This would minimize total change of ν_2 over the edges and clearly gives the smallest size bisection when using its sign to determine the cut. This same intuition can be refined by using more sophisticated heuristics on the coordinates of ν_2 to get even better approximations for general graphs. Thus, we see the eigenvalue approximates the value of the solution, in this case the isoperimetric number, which is closely related to the size of a bisection, and the second eigenvalue coordinates can be used to construct the actual solution. This combination of results is common in spectral methods and all the algorithms we present in this paper will roughly follow this strategy.

There is also an upper bound on the isoperimetric number known as Cheeger's inequality. In fact, we can upper bound a seemingly stronger quantity called the conductance of the graph, where the conductance of a subset of nodes, S, is $\phi(S) = \frac{|\partial(S)|}{\min(d(S),d(V-S))}$. Notice the similarity to the definition of the sparsest cut which we know is even hard to approximate within a constant factor under the assumption of UGC. Thus, spectral techniques can even be used to get approximate bounds on very hard problems.

3.3 Coloring

In this section, we will show how spectral algorithms can be used to tackle other hard problems with our primary example being graph colorability [6]. We will need to use the following well-known theorem that we will not prove.

Theorem 3.3 (Perron-Frobenius Theorem)**.** *Let G be a connected weighted graph, then the following hold:*

1. $\mu_1 \geq -\mu_n$,
2. $\mu_1 > \mu_2$, *and*
3. μ_1 *has a strictly positive eigenvector.*

Overall, coloring a graph can be viewed as partitioning the graph into several sections. One simple partitioning question we can ask is whether it can be split into two parts that only have edges between these parts. In other words, is the graph bipartite? The general k-partite question is equivalent to the k-coloring problem, so we start with bipartiteness.

Theorem 3.4. *A connected graph G is bipartite if and only if $\mu_1 = -\mu_n$*

Proof.

- [$\implies$] If G is bipartite, let $L \cup R = V$ be the bipartition. Let ϕ_1 be a strictly positive eigenvector of A_G corresponding to μ_1 guaranteed to exist by the Perron-Frobenius Theorem and define

$$x(u) = \begin{cases} \phi_1(u) \text{ if } u \in L \\ -\phi_1(u) \text{ if } u \in R \end{cases}$$

We then see that for any $u \in L$,

$$(A_G x)(u) = \sum_{(u,v)\in E} x(v) = -\sum_{(u,v)\in E} \phi_1(v) = -(A_G\phi_1)(u) = -\mu_1\phi_1(u) = -\mu_1 x(u)$$

A symmetrical argument holds in the case when $u \in R$, so we see that x is an eigenvector of A_G with eigenvalue $-\mu_1$. Now by the Perron-Frobenius theorem we know $\mu_1 \geq -\mu_n$. If $-\mu_1$ were not the smallest eigenvalue, then it must be that $-\mu_1 > \mu_n$ which is implies that $\mu_1 < -\mu_n$, a contradiction. Hence, $\mu_1 = -\mu_n$.

- [$\Longleftarrow$] If $\mu_1 = -\mu_n$, construct vector y satisfying $y(u) = |\phi_n(u)|$, where ϕ_n is an eigenvector for A_G with eigenvalue μ_n. By the triangle inequality, we have that

$$|\mu_n| = |\phi_n^T A_G \phi_n| = |\sum_{u,v \in V} A_G(u,v)\phi_n(u)\phi_n(v)| \leq \sum_{u,v \in V} A_G(u,v)|\phi_n(u)||\phi_n(v)|$$

$$= \sum_{u,v \in V} A_G(u,v)y(u)y(v) = y^T A_G y \leq \mu_1 y^T y = \mu_1$$

the last equality follows from the fact that $y^T y = \phi_n^T \phi = 1$ since ϕ_n is a unit vector. Now, since $\mu_1 = -\mu_n$ we know all the inqualities must hold. In particular, for the first inequality to hold, we need the sign of every product in the sum to be the same in order for it to equal the sum of the products of absolute values. Also, since the total sum is negative as $\mu_n < 0$ (note this holds since G is connected and so $-\mu_n = \mu_1 > 0$) it must be that whenever $A_G(u,v) = 1$ that $\phi_n(u), \phi_n(v)$ have opposite signs (or they are both are both 0, but The Perron-Frobenius theorem again prohibits this since the eigenvector for μ_1 is strictly positive). In other words, every edge goes between a vertex with positive sign under ϕ_n and a vertex with negative sign under ϕ_n. Thus, grouping vertices based on their sign under ϕ_n is a valid bipartition, so G is bipartite.

□

Before tackling the general coloring case, we will need to know how μ_1 relates to the degree sequence of the graph.

Theorem 3.5. $d_{avg} \leq \mu_1 \leq d_{max}$

Proof. We simply need to consider what the Rayleigh Quotient of the Laplacian looks like under the all 1s vector.

$$\mu_1 = \max_x \frac{x^T A_G x}{x^T x} \geq \frac{1^T A_G 1}{1^T 1} = \frac{\sum_{u,v \in V} A_G(u,v)}{n} = \frac{\sum_{(u,v) \in E} 2}{n} = \frac{2|E|}{n} = d_{avg}$$

Now, to show the other inequality we need to consider how ϕ_1 interacts with A_G. Suppose $\phi_1(v)$ is the largest entry of ϕ_1. Suppose WLOG $\phi_1(v) \neq 0$.

$$\mu_1 = \frac{(A_G\phi_1)(v)}{\phi_1(v)} = \frac{\sum_{(v,u) \in E} \phi_1(u)}{\phi_1(v)} = \sum_{(v,u) \in E} \frac{\phi_1(u)}{\phi_1(v)} \leq \sum_{(v,u) \in E} 1 = d(v) \leq d_{max}$$

where the inequality follows from the fact that v is the maximizer for ϕ_1.

□

In fact, if G is a connected graph, we see that if $\mu_1 = d_{max}$ then it must be the case that v is a vertex of maximum degree and each neighbor u of v

must also be a maximizer for ϕ_1. Hence, we can apply the same argument to the neighbors of v to show they are all of maximum degree. Since the graph is connected, we can inductively apply this argument (on the length of a shortest u to v path) to see that every vertex must be maximum degree. On the other hand, if G is d_{max}-regular, then $\mu_1 = d_{max}$ can be achieved through the all 1s vector. So we have just shown the following corollary.

Corollary 3.5.1. *If G is connected then $\mu_1 = d_{max}$ if and only if G is d_{max}-regular.*

In fact, we can strengthen the previous theorem's lower bound further.

Lemma 3.6. *For every $S \subseteq V$, we have $d_{avg}(S) \leq \mu_1$ where $d_{avg}(S)$ is the average degree of the induced subgraph of S.*

Given these tools, we can give bounds on the chromatic number of the graph. Recall that a (proper) k-coloring of a graph G is a function $c : V \to [k]$ such that $\forall (u, v) \in E, c(u) \neq c(v)$. The chromatic number $\chi(G)$ is the smallest k for which G is k-colorable. In general, k-colorable is equivalent to being k-partite since there are no edges between vertices of the same color. Given an arbitrary ordering of the vertices, we continually color a vertex by the least color available that is different from its neighbors that have already been colored. This clearly gives a proper coloring using total number of colors $1 + \max_u |\{v | v < u \wedge (u, v) \in E\}|$ where $v < u$ means v appears before u in the ordering. Since any vertex is adjacent to at most d_{max} many other nodes, this immediately implies that $\chi(G) \leq d_{max} + 1$. We can improve this bound on the chromatic number by finding better orderings of the vertices to use in this simply greedy coloring scheme. In particular, we can use μ_1 to choose a good ordering. This leads to the following result.

Theorem 3.7 (Wilf's Theorem)**.** $\chi(G) \leq \lfloor \mu_1 \rfloor + 1$

Proof. We will construct a good ordering for the greedy coloring as follows. First, by Theorem 3.5, we know that $d_{avg} \leq \mu_1$ and hence there exists some vertex u with $d(u) \leq \mu_1$. Since the degree is an integer it must be that $d(u) \leq \lfloor \mu_1 \rfloor$. Let u be the last vertex in our ordering. Now, we know the largest eigenvalue of A_{G-u} is at most that of A_G which is μ_1. Thus, we can inductively order the vertices of $G - u$ so that each vertex has at most $\lfloor \mu_1 \rfloor$ neighbors appearing before it. Placing u at the end of this ordering then ensures that every vertex of G has at most $\lfloor \mu_1 \rfloor$ neighbors appearing before it and so G is $\lfloor \mu_1 \rfloor + 1$-colorable using the greedy coloring. □

In fact, we can get a lower bound too by using multiple eigenvalues.

Theorem 3.8 (Hoffman's Bound)**.** $\chi(G) \geq 1 - \frac{\mu_1}{\mu_n}$

Hoffman's bound follows from a bound on the indepedence number of the graph, $\alpha(G)$, which is also classically NP-complete to compute [7].

Theorem 3.9. *If G is d-regular, $\alpha(G) \leq n \frac{-\mu_n}{d - \mu_n}$*

Proof. We will prove this using the correspondence between the eigenvalues of the Laplacian and the Adjacency matrix in the case of regular graphs. We show for any independent set S, that $|S| \leq n(1 - \frac{d_{avg}(S)}{\lambda_n})$. Then, since $\lambda_n = d - \mu_n$ and $d_{avg} = d$ in our case, we have $|S| \leq n(\frac{\lambda_n - d}{\lambda_n}) = n(\frac{-\mu_n}{d-\mu_n})$.

Again we will look at the Rayleigh Quotient with respect to the vector $y = \chi_S - s$. We know $y^T L_G y = |\partial(S)| = d(S) = d_{avg}(S)|S|$ since S has all its incident edges on the boundary by independence. Recall that $y^T y = |S|(1-s) = n(s-s^2)$. Thus, $\lambda_n = \max_x \frac{x^T L_G x}{x^T x} \geq \frac{y^T L_G y}{y^T y} = \frac{d_{avg}(S)|S|}{n(s-s^2)} = \frac{d_{avg}(S)}{1-s}$. Rearranging gives $s \leq 1 - \frac{d_{avg}(S)}{\lambda_n}$, so $|S| \leq n(1 - \frac{d_{avg}(S)}{\lambda_n})$. □

4 Unique Games

In this section, we discuss the best known algorithm for UniqueGames, which is uses spectral techniques [2].

Theorem 4.1. *Let $U = (G, M, k)$ 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 the eigenvectors of M that have eigenvalue greater than $(1-\gamma)d$, for $\gamma > 8\epsilon$. There is an algorithm that runs in $O(2^{\frac{\gamma}{\epsilon} dim(W)}) + poly(nk)$ time and finds an assignment that satisfies at least a $(1 - O(\frac{\epsilon}{\gamma - 8\epsilon} + \epsilon))$ fraction of the constraints [4].*

For an instance of UniqueGames, U, define the label-extended graph of U to be the graph of matchings corresponding to the possible labelings of the endpoints of each edge. In particular, we replace each edge $(u, v) \in E$ by a bipartite subgraph where we have an edge $((u, i), (v, j))$ if $\pi_{(u,v)}(i) = j$. In other words, $((u, i), (v, j))$ is an edge of the label extended graph if the labeling that maps u to i and v to j would satisfy the constraint on the original edge (u, v). Formally, we define the adjacency matrix of the label-extended graph of the game as $M(u, v, i, j) = [\pi_{(u,v)}(i) = j]$ if $(u, v) \in E$ and $M(u, v, i, j) = 0$ otherwise. Alternatively, we can define the $k \times k$ block of M corresponding to an edge (u, v) by the permutation matrix $\Pi_{(u,v)}$ where $\Pi_{(u,v)}(i, j) = [\pi_{(u,v)}(i) = j]$.

Given an instance of UniqueGames, U, that is $(1 - \epsilon)$-satisfiable, we will consider a completion of the game, $\tilde{U}$, that is the original game where we replace each of the ϵ constraints that are unsatisfiable in some maximal assignment, L, by any other constraints that will make the instance completely satisfiable. Then, we will view an almost satisfiable unique game as being derived from a completely satisfying one where an adversary picked an ϵ fraction of the constraints to change as to potentially make the whole instance $(1 - \epsilon)$-satisfiable. Thus, we will commonly consider some completion of a game, $\tilde{U}$, when discussing solutions to the original game. In particular, if L is a labeling that satisfies every constraint of the completion game, then L will satisfy an $(1 - \epsilon)$ fraction of the constraints of the original graph since they share all but an ϵ fraction of constraints. Thus, if we knew which ϵ constraints were modified from

some completion game, we could easily compute a good labeling in polynomial time. However, finding such a completion seems difficult in general.

Next, if L is a labeling that satisfies all the constraints for $\tilde{U}$, we define the characteristic vector of L as y^L, where

$$y^L(u,i) = \frac{1}{\sqrt{n}}[L(u) = i]$$

we note that if $\tilde{M}$ is the adjacency matrix of the label-extended graph of $\tilde{U}$, then y^L is a unit eigenvector of $\tilde{M}$ of maximum eigenvalue, which is d. Specifically,

$$\begin{aligned}(y^L)^T \tilde{M} y^L &= \sum_{u,v,i,j} \tilde{M}(u,v,i,j) y^L(u,i) y^L(v,j) \\ &= \frac{1}{n} \sum_{u,v,i,j} [\pi_{(u,v)}(i) = j][L(u) = i][L(v) = j] \\ &= \frac{1}{n} \sum_{(u,v)\in E} 1 = \frac{nd}{n} = d\end{aligned}$$

since G is d-regular so the total number of edges is nd. In general,

Lemma 4.2. *If U is δ-satisfiable and L is a labeling satisfying at least a δ fraction of U's constraints, then $(y^L)^T M y^L \geq \delta d$*

Proof. $(y^L)^T M y^L = \sum_{u,v,i,j} = \frac{1}{n} *$number of edges satisfied $\geq \frac{\delta n d}{n} = \delta d$. □

As mentioned previously, any labeling, L, satisfying all the constraints of some completion will be a good labeling for the given game. Consequently, the characteristic vector of L will have a large Rayleigh Quotient with respect to M similarly to how this vector maximizes the Rayleigh Quotient for $\tilde{M}$. Thus, we could hope that a subspace of eigenvectors for M with large eigenvalues, W, would be close to the characteristic vectors of perfect labelings for some completion of the game and so could be used to get good labelings. In particular, we will construct a set of nice vectors, N, contained in this subspace that are all guaranteed to be close to the characteristic vectors of the completely satisfying assignments. Since W will be infinite, finding this set N will be impossible using an exhaustive search, so we will need to consider a discretized approximation to W that is guaranteed to have some vectors close to the nice vectors. Then, we use the following algorithm for computing a good labeling from this set S.

GetLabeling(S,U):

- For each $x \in S$, construct a labeling L_x defined by $L_x(u) = \arg\max_{i\in[k]} x(u,i)$.
- Return the labeling L_x that maximizes the fraction of constraints of U satisfied out of all labelings constructed in the previous part.

In our case, we will consider the subspace W that is the span of all eigenvctors of M that have eigenvalue greater than $(1-\gamma)d$ for some $\gamma > 8\epsilon$. The following claim will help us identify the set of nice vectors, N.

Lemma 4.3. *For every completely satisfing assingment, L, there is a unit vector, v_L, such that $v_L = \alpha y^L + \beta y^L_\perp$ for some $\alpha \geq \sqrt{1 - \frac{2\epsilon}{\gamma}} > 0$ and $|\beta| < \sqrt{\frac{2\epsilon}{\gamma}}$*

Proof. First, recall by lemma 4.2, $(y^L)^T M y^L \geq d(1-\epsilon) \geq d(1-2\epsilon)$. Now, we will write $y^L = av_L + b(v_L)_\perp$ for some $a > 0$, $v_L \in W$, and $(v_L)_\perp \in W^\perp$ and derive conditions on b. Note we can always write y^L as a sum of a vector from some subspace and a vector from the orthogonal complement of the subspace by well known results of linear algebra. Also, by absorbing the inverse of the norm of the vectors, $v_L, (v_L)_\perp$ into their corresponding multiplicative factors, a, b, we can assume that v_L and $(v_L)_\perp$ are unit vectors. We have that

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

where last inequality follows from the fact that d is the largest eigenvalue of d so no quadratic form of a unit vector can be more than d and $(v_L)_\perp$ is orthogonal to every eigenvector in W and so cannot have Rayleigh Quotient larger than $(1-\gamma)d$ otherwise it would be in W. From these inequalities, we see that $1 - 2\epsilon \leq a^2 + b^2(1-\gamma)$. This implies that $2\epsilon - 1 \geq -a^2 + b^2(\gamma - 1)$. Rearranging the inequality gives $|b| \leq \sqrt{\frac{2\epsilon + a^2 - 1}{\gamma - 1}}$. Since y^L is a unit vector and $v_L, (v_L)_\perp$ are orthogonal to each other, the triangle inequality implies that $1 = ||av_L|| + ||b(v_L)_\perp|| = a^2 + b^2$. From this we see that $a^2 - 1 \leq 0$ and so $|b| \leq \sqrt{\frac{2\epsilon}{\gamma}}$. Now, we can write $v_L = \alpha y^L + \beta y^L_\perp$ where $\alpha =< v_L, y^L >$. This follows since any vector can be written as a linear combination of eigenvectors where the scaling factor of each eigenvector is its inner product with the given vector. Note here $y^L_\perp$ absorbs the terms of all the other eigenvectors besides y^L as they are all orthogonal to y^L. Since $a > 0$, $< v_L, y^L >= a = \sqrt{1 - b^2} \geq \sqrt{1 - \frac{2\epsilon}{\gamma}}$. Since v_L is also a unit vector, we can repeat a similar argument as before to conclude that $|\beta| = \sqrt{1 - \alpha^2} \leq \sqrt{\frac{2\epsilon}{\gamma}}$. □

If we knew the completely satisfying the assignments, we could simply use $N = \{v_L\}$ as input to GetLabeling.

Lemma 4.4. *If x is a vector such that $x = \alpha y^L + \beta y^L_\perp$ for some y^L with $\alpha > 0$, then $L(u) = \arg\max_{i \in [k]} x(u,i)$ in at least $(1 - \frac{2\beta^2}{\alpha^2})n$ blocks of M.*

This lemma implies that the approach GetLabeling(N,U) uses to constructing a labeling by reading it off from the nice vectors will work for many of the vertices.

Lastly, we need to compute the discretized approximation to W, S, that will have vectors close to some nice vector. Suppose $w_1, ..., w_{dim(W)}$ is an eigenbasis for W. Then, we will consider

$$S = \{v = \sum_{i=1}^{dim(W)} \alpha_i w_i | \alpha_s \in \sqrt{\frac{2\epsilon}{\gamma dim(W)}} Z, ||v|| \leq 1\}$$

It has been show that $|S| \leq 2^{O(\frac{\gamma}{\epsilon} dim(W))}$. By construction, S contains some vector v for which $v = \alpha v_L + \beta (v_L)_\perp$ for some nice vector v_L and some $\beta \leq \sqrt{\frac{2\epsilon}{\gamma}}$. Thus, $v = ay^L + by^L_\perp$ with $|b| \leq 2\sqrt{\frac{2\epsilon}{\gamma}}$. So, GetLabeling(S,U) must return a labeling that satisfies a $1 - \frac{2b^2}{a^2}$ fraction of the blocks. Since the norm of v is at most 1 we can again argue that $a^2 \geq 1 - b^2 \geq 1 - \frac{8\epsilon}{\gamma}$, so $1 - \frac{2b^2}{a^2} \geq 1 - 2\frac{8\epsilon}{\gamma - 8\epsilon}$. Thus, this labeling will satisfy a $1 - O(\frac{\epsilon}{\gamma - 8\epsilon})$ fraction of the blocks. We need to analyze how many constraints are violated by the labeling returned by this procedure. First, we know there are an ϵ fraction of the constraints that were changed by the adversary and so cannot possibly be satisfied. Also, there are an $O(\frac{\epsilon}{\gamma - 8\epsilon})$ many constraints that disagree with the perfect labeling. Thus, there are a total of at most $\epsilon + O(\frac{\epsilon}{\gamma - 8\epsilon})$ fraction of the edges that are unsatisfied by our labeling, and so $1 - O(\epsilon + \frac{\epsilon}{\gamma - 8\epsilon})$ are satisfied. Lastly, we note that the algorithm runs in time exponential in the dimension of W. Specifically, we first have to construct the eigenbasis that defines S, which takes $poly(nk)$ time. Then, the procedure GetLabeling will take time to do an exhaustive search over S. Since S has at most $2^{O(\frac{\gamma}{\epsilon} dim(W))}$, this step will take $2^{O(\frac{\gamma}{\epsilon} dim(W))}$ time. Overall, the algorithm runs in time $2^{O(\frac{\gamma}{\epsilon} dim(W))} + poly(nk)$ as desired.

We now discuss the consequences of this algorithm. First, this algorithm runs in polynomial time and satisfies enough constraints for expander graphs in order to disprove the conjecture on expanders. In addition, this algorithm works in subexponential time on the game that achieves the integrality gap for the standard SDP formulation for the unique games problem. Thus, this spectral approach seems to be more powerful. In fact, the best know algorithm for unique games uses the above algorithm as a subroutine along with a partitioning procedure for the graph [2]. This best known algorithm runs in subexponential time on all graphs and satisfies a very large fraction of the constraints. These spectral approaches could still possibly be improved further to disprove the Unique Games Conjecture.

5 Conclusion

Graph problems are incredibly difficult and prevalent throughout computer science. Of all of these, the Unique Games Conjecture has profound implications if true. However, using the power of spectral techniques, we can design efficient approximation algorithms for many NP-hard problems, including UniqueGames. Further investigation of spectral approaches could potentially lead to the refutation of the Unique Games Conjecture, and so the most interesting open problem would be to extend the spectral techniques presented in this paper to solve UniqueGames once and for all or perhaps develop completely new spectral techniques that could give a completely different perspective on the conjecture. In particular, the algorithm presented in this paper used only the adjacency matrix, which is usually not the most natural matrix to use with spectral techniques. Thus, it would be interesting to investigate whether other choices of

standard matrix such as the Laplacian or even a completely new matrix could be used to get better results.

References

[1] Noga Alon. "Spectral Techniques in Graph Algorithms". In: *Proceedings of the Third Latin American Symposium on Theoretical Informatics*. LATIN '98. Berlin, Heidelberg: Springer-Verlag, 1998, pp. 206–215. ISBN: 3-540-64275-7. URL: http://dl.acm.org/citation.cfm?id=646387.690180.

[2] Sanjeev Arora, Boaz Barak, and David Steurer. "Subexponential Algorithms for Unique Games and Related Problems". In: *J. ACM* 62.5 (Nov. 2015), 42:1–42:25. ISSN: 0004-5411. DOI: 10.1145/2775105. URL: http://doi.acm.org/10.1145/2775105.

[3] S. Khot. "On the Unique Games Conjecture (Invited Survey)". In: *2010 IEEE 25th Annual Conference on Computational Complexity*. June 2010, pp. 99–121. DOI: 10.1109/CCC.2010.19.

[4] Alexandra Kolla. "Spectral Algorithms for Unique Games". In: *CoRR* abs/1102.2300 (2011). arXiv: 1102.2300. URL: http://arxiv.org/abs/1102.2300.

[5] Daniel Spielman. "Introduction". In: *Spectral Graph Theory Lecture Notes* (2015). URL: http://www.cs.yale.edu/homes/spielman/561/lect01-15.pdf.

[6] Daniel Spielman. "The Adjacency Matrix and Graph Coloring". In: *Spectral Graph Theory Lecture Notes* (2015). URL: http://www.cs.yale.edu/homes/spielman/561/lect03-15.pdf.

[7] Daniel Spielman. "The Adjacency matrix and the nth eigenvalue". In: *Spectral Graph Theory Lecture Notes* (2012). URL: http://www.cs.yale.edu/homes/spielman/561/2012/lect03-12.pdf.

[8] Daniel Spielman. "The Laplacian". In: *Spectral Graph Theory Lecture Notes* (2015). URL: http://www.cs.yale.edu/homes/spielman/561/lect02-15.pdf.