

Spectral Partitioning of Random Graphs

Frank M^cSherry
Dept. of Computer Science and Engineering
University of Washington
Seattle, WA 98105
mcsberry@cs.washington.edu

Abstract

Problems such as bisection, graph coloring, and clique are generally believed hard in the worst case. However, they can be solved if the input data is drawn randomly from a distribution over graphs containing acceptable solutions.

In this paper we show that a simple spectral algorithm can solve all three problems above in the average case, as well as a more general problem of partitioning graphs based on edge density. In nearly all cases our approach meets or exceeds previous parameters, while introducing substantial generality. We apply spectral techniques, using foremost the observation that in all of these problems, the expected adjacency matrix is a low rank matrix wherein the structure of the solution is evident.

1 Introduction

The problems of bisection, graph coloring, and clique are well known to be NP-hard. As such, much literature has emerged on the average-case performance of algorithms for each problem, the belief being that perhaps the worst-case instances are not representative of problem instances as a whole. Average-case analysis is equivalent to considering the probability of success on a random graph, however choosing problem instances *too uniformly* from the collection of all graphs is not particularly illuminating. A very small fraction of all n node graphs have small colorings, large cliques, or bisections of merit. Trivial algorithms are thus able to compete with good algorithms.

One resolution to this issue is to restrict the input distribution to graphs which are likely to have a specific non-trivial solution. Each average-case graph problem is therefore associated with a particular model of random graph. A graph is generated by including each edge independently with an associated probability. The probability associated with each edge is tailored so that with high probability the graph includes a particular combinatorial object.

Three problems that have been studied extensively are bisection, k -coloring, and clique, where the associated graph models are:

- **Bisection:** Fix a bisection. Include each intra-part edge with probability q , and each inter-part edge with probability $p < q$.
- **k -Coloring:** Fix a k -coloring. Include each edge with probability p , and then removes all intra-color edges.
- **Clique:** Fix a set of nodes for inclusion in the clique. Include each edge with probability p and then complete the clique.

The problem of graph bisection on random graphs has been studied for some time. Bui et al. [7] and Dyer and Frieze [10] have presented algorithms for bisecting dense graphs when $p < (1 - \epsilon)q$. Jerrum and Sorkin [14] consider an approach based on simulated annealing, but using a single constant temperature. Boppana [6] presents a spectral algorithm which succeeds for a large range of parameters (we will see them shortly), but his approach requires the solution to a convex optimization problem. Recently, Condon and Karp [8] analyzed a linear time combinatorial algorithm for partitioning which nearly achieves the same range of parameters as [6].

Similarly, many researchers have worked on the problem of coloring random graphs which have k -colorings. Kucera [18], Turner [22], and Dyer and Frieze [10] present algorithms that optimally color k -colorable graphs for fixed k , with high probability. However, most graphs are dense, and therefore easier to color than sparse graphs. Blum and Spencer [5] and Alon and Kahale [2] demonstrate algorithms that color random sparse graphs properly with high probability, the latter using a spectral algorithm.

The problem of finding a large clique in a random graph was suggested by Karp in [16]. Kucera [19] observes that when the size of the clique is $\omega(\sqrt{n \log n})$ and $p = 1/2$, the clique members are simply the vertices of highest degree. Alon, Krivelevich, and Sudakov [3] showed that a planted

clique of size $\Omega(\sqrt{n})$ can be found through spectral techniques when $p = 1/2$.

1.1 Graph Partition Model

The graph models above each generate a graph by including each edge independently with an associated probability. Further, in each model we can partition the nodes of the graph so that the probability associated with each edge depends only on the parts to which its endpoint belongs. Based on this observation we introduce the following general model of “structured” random graphs.

$\mathcal{G}(\psi, P)$: Let $\psi : \{1, \dots, n\} \rightarrow \{1, \dots, k\}$ be a partition of n nodes into k classes. Let P be a $k \times k$ matrix where $P_{ij} \in [0, 1]$ for all i, j . Include edge (u, v) with probability $P_{\psi(u)\psi(v)}$.

For a particular distribution $\mathcal{G}(\psi, P)$ we let \widehat{G} refer to the matrix of random variables corresponding to the adjacency matrix of the random graph. We also let G refer to the matrix of expectations, where $G_{uv} = P_{\psi(u)\psi(v)}$.

We now specialize this model into three models which are equivalent to the models for planted bisection/multisection, coloring, and clique presented in the literature.

- **Planted Multisection** (ψ, p, q) : ψ is the multisection. P is p everywhere, except the diagonal where it is q .
- **Planted k-Coloring** (ψ, p) : ψ is the coloring. P is p everywhere except the diagonal, where it is 0.
- **Planted Clique** (ψ, p) : Let $\psi(v) = 1$ iff v is in the clique. P is p everywhere, except P_{11} , which is 1.

Our model of a random graph with a planted partition leads naturally to the following graph partitioning problem.

Planted Partition Problem: Given a graph \widehat{G} drawn from the distribution $\mathcal{G}(\psi, P)$, produce a partition $\widehat{\psi}$ so that

$$\widehat{\psi}(u) = \widehat{\psi}(v) \quad \text{iff} \quad \psi(u) = \psi(v)$$

As ψ encodes the solution to each of the problems above, recovering a partition equivalent to ψ generalizes the problems of finding planted multisections, cliques, and colorings in their respective models.

It is important to disassociate this problem from traditional related optimization problems. The goal is not to find the largest clique or min cost bisection, but rather to recover the planted object. In many cases these two will be the same, but if the optimal solution is not equivalent to the planted object our goal is to find the latter.

1.2 Our Approach

Given the matrix G , it is easy to reconstruct ψ by clustering the columns G_u of G . Unfortunately, we have instead a matrix \widehat{G} which is a highly perturbed version of G , and the columns \widehat{G}_u are nowhere near the G_u . It is perhaps natural to ask: “Why can we hope to recover ψ at all?” The answer to this question, and our approach to this problem, is based on the following observation:

For any ψ and P , the matrix G has rank k .

If P_G is the projection on the column space of G

- $|P_G(G_u) - G_u|$ is zero.
- $|P_G(G_u - \widehat{G}_u)|$ is small.

By the triangle inequality, $P_G(\widehat{G}_u)$ equals G_u , plus a “small” error.

Of course, we do not have access to P_G either. A result from matrix perturbation theory saves us, in that the equivalent projection for \widehat{G} is close to P_G . Our approach is now to find a projection P_X so that

- | | |
|--------------------|---------------------------------------|
| Data is Preserved: | $ P_X(G_u) - G_u $ is small |
| Noise is Removed: | $ P_X(G_u - \widehat{G}_u) $ is small |

If so, then $P_X(\widehat{G}_u)$ equals G_u up to a “small” error. If when $\psi(u) \neq \psi(v)$, $|G_u - G_v|$ is much larger than this error, we can apply a simple greedy clustering process to the $P_X(\widehat{G}_u)$.

With a minor modification, this is the algorithm we analyze in this paper. Let τ be a threshold parameter, and let **CProj** be a function which computes an “appropriate” projection matrix.

Partition (\widehat{G}, τ)

1. Randomly divide $\{1, \dots, n\}$ into two parts.
Let this division split the columns of \widehat{G} as $\left[\widehat{A} \mid \widehat{B} \right]$
2. Let $P_1 = \mathbf{CProj}(\widehat{B})$; let $P_2 = \mathbf{CProj}(\widehat{A})$
3. Let $\widehat{H} = \left[P_1(\widehat{A}) \mid P_2(\widehat{B}) \right]$
4. While there are unpartitioned nodes
 - (a) Choose an unpartitioned node u_i arbitrarily
 - (b) For each v , set $\widehat{\psi}(v) = i$ if $|\widehat{H}_{u_i} - \widehat{H}_v| \leq \tau$
5. Return the partition $\widehat{\psi}$.

Notice that we split the matrix \widehat{G} into two parts. This is done to avoid the conditioning that would otherwise exist between the error $G - \widehat{G}$ and the computed projection, a function of \widehat{G} .

1.3 Results

The main result of this paper is an analysis of the algorithm **Partition**. Appropriate choices of **CProj** and τ result in perfect classification for a large range of (ψ, P) . Before considering the general result, we examine the specializations to the three problems we have mentioned. For these corollaries we require that at least one entry in G has variance at least $\log^6(n)/n$, and that the failure probability δ is no smaller than $\exp(-\log^6 n)$.

Corollary 1 *Let (ψ, p, q) be an instance of the planted multisection problem with k parts. There is a constant c so that for sufficiently large n if*

$$\frac{q-p}{q} > c\sqrt{\frac{\log(n/\delta)}{qn}}$$

then we can recover ψ with probability $1 - \delta$.

This range of parameters is equivalent to the range in [6], up to constant factors. It is worth emphasizing that Boppana produces the optimal bisection, whereas we recover the planted multisection.

Corollary 2 *Let (ψ, p) be an instance of the planted k -coloring problem, where the size of each color class is linear in n . There is a constant c such that for sufficiently large n if*

$$p > c\log^3(n/\delta)/n$$

then we can recover ψ with probability $1 - \delta$.

This result is simultaneously weaker and more general than that of [2]. Here we admit color classes of differing sizes, and can further generalize to cover the case where the sizes of the color classes are asymptotically different. On the other hand, this result covers a smaller range of p than [2] who show that the problem can be solved even when $p = c/n$, for some large constant c .

Corollary 3 *Let (ψ, p) be an instance of the planted clique problem, where the clique size is s . There is a constant c such that for sufficiently large n if*

$$\frac{1-p}{p} > c\left(\frac{n}{s^2} + \frac{\log(n/\delta)}{s}\right)$$

then we can recover ψ with probability $1 - \delta$.

This result subsumes the spectral result of [3] where they show that cliques of size $\Omega(\sqrt{n})$ can be found when $p = 1/2$. Note that this theorem allows for variable p and s . The restriction to a single clique is also not necessary. The general theorem addresses graphs with several hidden cliques and hidden independent sets, each of varying size.

To describe the performance of **Partition** in the general context of the Planted Partition Problem we must describe the range of (ψ, P) for which the algorithm succeeds. This range is best described by a requisite lower bound on $|G_u - G_v|$ when $\psi(u) \neq \psi(v)$.

Theorem 4 *Let (ψ, P) be an instance of the planted partition problem. Let $\sigma^2 \gg \log^6 n/n$ be an upper bound on the variance of the entries in G , and let s_m be the size of the smallest part of ψ .*

There is a constant c such that for sufficiently large n if when $\psi(u) \neq \psi(v)$

$$|G_u - G_v|^2 > ck\sigma^2 \left(\frac{n}{s_m} + \log\left(\frac{n}{\delta}\right) \right)$$

then given \hat{G} we can efficiently recover ψ with probability $1 - \delta$ over the random graph \hat{G} , and probability k^{-1} over the random bits of the algorithm.

1.4 Additional Related Work

As well as theoretical success in average case analysis, spectral algorithms have been successfully used in practice as a heuristic for data partitioning. While there is no single spectral approach, most examine the eigenvectors of the adjacency matrix of a graph (or of the Laplacian of this matrix). In particular, the second eigenvector is typically used as a classifier, partitioning nodes based on the sign of their coordinate. The special cases of bounded degree planar graphs and d -dimensional meshes (cases which occur frequently in practice) were analyzed successfully by Spielman and Teng in [21]. Recently, Kannan, Vempala, and Vetta [15] gave a compelling clustering bi-criteria and a spectral algorithm which produces clusterings of quality similar to the optimal clustering.

The perturbation theory of a matrix's spectrum has been around for some time. Davis and Kahan [9] present a classic analysis of the perturbation of Hermitian operators. Through a fairly simple transformation, the results also lead to perturbation bounds for the singular value decomposition. Papadimitriou et al. [20] and Azar et al. [4] have applied perturbation theory to analyze and give justification for spectral techniques. In particular, the observed insensitivity of low rank approximations to random noise is closely related to this work.

Feige and Kilian [11] consider an alternate model for describing the performance of "empirical" algorithms. For the problems of bisection, coloring, and clique, a random graph is produced as before. However, they now allow an adversary to "help" the algorithm, perhaps by including additional edges between color classes, or removing non-clique edges. [11] give algorithms that address these problems when the objects are of linear size. While this model is

interesting in these three domains, it does not seem to generalize to planted partition problem.

1.5 Paper Outline

In section 2 we review a few important facts from linear algebra. In section 3 we analyze two types of projection matrices. Finally, in section 4 we conclude with several extensions to and observations about our spectral approach.

2 Linear Algebra

We will need to introduce some notions from linear algebra before we can proceed too far. The text of Golub and Van Loan [13] is an excellent reference for this material. Notationally speaking, capital letters (M) will represent matrices, single subscripts (M_i) will represent a column of the matrix (or a vector), and double subscripts (M_{ij}) will index elements of a matrix first by row, then column.

2.1 Vector and Matrix Norms

We will frequently use the vector L2 norm, defined as

$$|M_i|^2 = \sum_j M_{ij}^2$$

We will use two matrix norms in this paper. They are

$$|M| = \max_{|x|=1} |Mx| \quad \text{Spectral Norm}$$

$$|M|_F^2 = \sum_{i,j} M_{ij}^2 \quad \text{Frobenius Norm}$$

One important relation between the two norms is:

Fact 5 *If M is rank k , then $|M|_F^2 \leq k|M|^2$.*

If we associate the Frobenius summation by column we see

Fact 6 $\sum_i |M_i|^2 = |M|_F^2$

A counting argument limits the number of columns that have significant length

Claim 7 *The set $\{i : |M_i|^2 > |M|_F^2/c\}$ has size at most c .*

2.2 Subspaces and Projections

A subspace of R^n is a set of vectors closed under addition and scalar multiplication. Any subspace S of R^n can be represented by a collection of orthonormal vectors $\{q_i\}$ such that S is exactly the set of linear combinations of those vectors $\{q_i\}$. If we let Q be the matrix whose columns are the $\{q_i\}$, then the matrix

$$P_S = QQ^T$$

is the orthogonal projection onto S . For any vector x , $P_S x$ is the vector in S closest to x . If one were to stand at x and drop a ball, with gravity oriented towards the hyper-plane S describes, it would bounce exactly at $P_S x$.

2.3 Singular Vectors

One very important class of projections with respect to a matrix M is the projection onto the span of its first k left singular vectors (computed through the singular value decomposition). If P_M denotes this projection (k will be “understood”), then $P_M M$ is the optimal rank k approximation to M in the following sense

Fact 8 *For any rank k matrix X , $|M - P_M M| \leq |M - X|$*

These facts (and the triangle inequality) demonstrate that

Lemma 9 *For any matrix \widehat{M} and rank k matrix M ,*

$$|M - P_{\widehat{M}} \widehat{M}|_F^2 \leq 8k|M - \widehat{M}|^2$$

Proof:

$$\begin{aligned} |M - P_{\widehat{M}} \widehat{M}|_F^2 &\leq 2k|M - P_{\widehat{M}} \widehat{M}|^2 \\ &\leq 2k(|M - \widehat{M}| + |\widehat{M} - P_{\widehat{M}} \widehat{M}|)^2 \\ &\leq 2k(|M - \widehat{M}| + |\widehat{M} - M|)^2 \\ &= 8k|M - \widehat{M}|^2 \end{aligned}$$

□

It is worth noting that this fairly simple observation plays a central role in our result. By bounding $|G - P_{\widehat{G}} \widehat{G}|_F^2$, we have bounded the total entry-wise error in using $P_{\widehat{G}} \widehat{G}$, a computable quantity, to approximate G .

2.4 Random Matrices

Random matrices play an important role in our result, as it is a random matrix that separates the observed data from what we aim to recover. In particular, as we saw just above, the spectral norm of $G - \widehat{G}$ is quite relevant. The following is a classic expectation result of Furedi and Komlos [12], combined with a recent concentration result of Krivelevich and Vu [17].

Theorem 10 *Let \widehat{M} be a matrix generated by randomly rounding the entries of a matrix of probabilities M preserving symmetry. Let σ be the largest deviation of an entry in \widehat{M} . If $\sigma^2 \gg \log^6 n/n$, then*

$$|\widehat{M} - M| \leq 4\sigma\sqrt{n}$$

with probability at least $1 - 2e^{-\sigma^2 n/8}$

For the range of σ allowed in the theorem, this probability is always $1 - O(\exp(-\log^6 n))$. For the rest of this paper, we will assume that this bound holds. Furthermore, unless explicitly stated otherwise, we will assume that σ satisfies the condition of the theorem.

3 Spectral Graph Partitioning

It is now time to identify projections and delve into their particulars. Recall that **Partition** works when the projected columns $P(\widehat{G}_u)$ are close to the original columns G_u , and the columns in G from different parts are distant. We can codify this in the following observation.

Observation 11 *Assume that for all u*

$$\begin{aligned} |P_1(A_u) - A_u| &\leq \gamma_1 & \text{and} & & |P_1(A_u - \widehat{A}_u)| &\leq \gamma_2 \\ |P_2(B_u) - B_u| &\leq \gamma_1 & \text{and} & & |P_2(B_u - \widehat{B}_u)| &\leq \gamma_2 \end{aligned}$$

If when $\psi(u) \neq \psi(v)$

$$|G_u - G_v| \geq 4(\gamma_1 + \gamma_2)$$

then **Partition** $(\widehat{G}, 2(\gamma_1 + \gamma_2))$ is equivalent to ψ .

Naturally, the challenging question is: ‘‘Given \widehat{A} and \widehat{B} , can we compute projections P_1 and P_2 with small values of γ_1 and γ_2 ?’’ Theorems 12 and 14 bound these values for two different computable projections. Theorem 14 involves **CProj** and largely surpasses the results of Theorem 12, but the latter is pleasingly simple and (in the author’s opinion) more natural than Theorem 14.

Before we barge into any proofs or other discussions, we need to define a few terms. With high probability, the size of each part of ψ when restricted to A (or B) is close to half the original size of the part. To facilitate our discussion, we will assume (wrongly) that this size is exactly half. Let s_i denote the size of part i of ψ in each of A and B . Let $s_m = \min_i s_i$ be a lower bound on these sizes. The variables i, j, ℓ all lie in $\{1, \dots, k\}$, whereas the variables u, v lie in $\{1, \dots, n\}$. We will occasionally use the notation like G_i to refer to the column of G that corresponds to nodes in part i . Likewise, we use s_u to refer to the size of the part containing u .

3.1 A Traditional Spectral Result

A natural projection to consider for P_2 is the projection onto the first k left singular vectors of \widehat{A} . As they are the best basis to describe \widehat{A} , we might imagine that they will capture the structure of B as well (the columns of A and B are the same).

Theorem 12 *With probability at least $1 - \delta$*

$$\begin{aligned} |P_{\widehat{A}}(B_u) - B_u| &\leq 8\sigma\sqrt{nk/s_u} \\ |P_{\widehat{A}}(B_u - \widehat{B}_u)| &\leq \sqrt{2k \log(n/\delta)} \end{aligned}$$

Proof: Note that

$$\begin{aligned} (I - P_{\widehat{A}})B &= (I - P_{\widehat{A}})\widehat{A} - (I - P_{\widehat{A}})(\widehat{A} - B) \\ &= (\widehat{A} - P_{\widehat{A}}\widehat{A}) - (I - P_{\widehat{A}})(\widehat{A} - A) \end{aligned}$$

$|\widehat{A} - P_{\widehat{A}}\widehat{A}|$ can be bounded as $|\widehat{A} - A|$, as it is $\lambda_{k+1}(\widehat{A})$, and $\lambda_{k+1}(A) = 0$. The magnitude of any eigenvalue can change by at most $|A - \widehat{A}|$, which give us that bound. We take norms of the above equation and conclude

$$|(P_{\widehat{A}} - I)B| \leq 2|\widehat{A} - A|$$

Now, observe that for any column u , there are s_u identical columns in $(I - P_{\widehat{A}})B$. As such, the length of the column in $(P_{\widehat{A}} - I)B$ can be no more than our L2 bound, divided by $\sqrt{s_u}$. Otherwise we will violate the L2 bound we just derived (that column would witness a higher L2 norm).

We now consider the second inequality. If $\{Q_j\}$ are the left singular vectors of \widehat{A} , then

$$|P_{\widehat{A}}(B_u - \widehat{B}_u)|^2 = \sum_j \langle (B_u - \widehat{B}_u), Q_j \rangle^2$$

Observe that for any unit vector Q_j

$$\langle (B_u - \widehat{B}_u), Q_j \rangle = \sum_v (B_{vu} - \widehat{B}_{vu})Q_{vj}$$

Each of these terms in the sum is an independent zero mean random variable. Furthermore, we can bound

$$\sum_v |(B_{vu} - \widehat{B}_{vu})Q_{vj}|^2 \leq 1$$

We can apply Azuma’s inequality (Theorem 13) to bound $|\langle (B_u - \widehat{B}_u), Q_j \rangle|$. Furthermore, we can apply a union bound to bound $|P_{\widehat{A}}(B_u - \widehat{B}_u)|$ for all u . \square

Theorem 13 (Azuma) *Let $\{X_i\}$ be independent zero mean random variables such that $|X_i| \leq c_i$ where $\sum_i c_i^2 \leq 1$. Then for any $\lambda > 0$,*

$$Pr\left[\left|\sum_i X_i\right| \geq \lambda\right] \leq 2e^{-\lambda^2/2}$$

3.2 Combinatorial Projections

While Theorem 12 is acceptable when σ is large (or s_m is small) it does not give competitive results for very small values of σ . The absence of a σ in the second inequality stops attempts to prove strong bounds. We would like to argue that if the vector $B_i - \widehat{B}_i$ has very low variance in its entries, its projection will be that much smaller.

We now discuss the algorithm **CProj** and analyze its performance. In this algorithm, and in its discussion, we use

the notation \widehat{A}_v^T . This is the v th column of the transpose of \widehat{A} , which is a vector of roughly $n/2$ coordinates. Note that as each part of ψ is assumed bisected, A_v^T is like a small copy of G_v^T .

CProj($\widehat{A}, k, s_m, \tau$)

1. While there are at least $s_m/2$ unclassified nodes
 - (a) Choose an unclassified node v_i randomly.
 - (b) Let $T_i = \{u : |P_{\widehat{A}^T}(\widehat{A}_{v_i}^T - \widehat{A}_u^T)| \leq \tau\}$
 - (c) Mark each $u \in T_i$ as classified.
2. Assign each remaining node to the T_i with closest projected v_i .
3. Let \widehat{c}_i be the characteristic vector of T_i
4. Return $P_{\widehat{c}}$, the projection onto the span of the \widehat{c}_i .

If the \widehat{c}_i were the characteristic vectors of ψ , this projection would be exactly P_A . Instead, we will see that the \widehat{c}_i are not unlike the characteristic vectors of ψ .

Theorem 14 *If when $\psi(u) \neq \psi(v)$*

$$|B_u - B_v| > 64\sigma\sqrt{nk/s_m}$$

then with probability $1 - \delta$, if we let

$$P_{\widehat{c}} = \mathbf{CProj}(A, k, s_m, 32\sigma\sqrt{nk/s_m})$$

then

$$\begin{aligned} |P_{\widehat{c}}(B_u) - B_u| &\leq 64\sigma\sqrt{nk/s_m} \\ |P_{\widehat{c}}(B_u - \widehat{B}_u)| &\leq \sigma\sqrt{2k \log(n/\delta)} \end{aligned}$$

with probability at least $k^{-1/2}$.

Proof: Lemma 16 proves the first bound, but we must first read Lemma 15 for an important observation. A Chernoff-type argument proves the second bound, though the details are omitted from this abstract. \square

Combining this theorem with Observation 11 allows us to conclude Theorem 4. We now proceed to the proofs of the supporting lemmas. The first is a structural property about the sets produced in **CProj**. We assume (without loss of generality, as we will see) that $\psi(v_i) = i$.

Lemma 15 *For the conditions of Theorem 14, with probability at least $k^{-1/2}$ **CProj**($\widehat{A}, k, s_m, 32\sigma\sqrt{nk/s_m}$) produces k sets T_i such that if we define*

$$\alpha_{\ell i} = |\{v : \psi(v) = \ell \text{ and } v \in T_i\}|$$

then for any j

$$\sum_{i,\ell} \alpha_{\ell i} (P_{\ell j} - P_{ij})^2 \leq 128k|A - \widehat{A}|^2/s_j$$

Proof: Let us call a node v good if

$$\begin{aligned} |A_v^T - P_{\widehat{A}}\widehat{A}_v^T| &\leq |A - P_{\widehat{A}}\widehat{A}|_F / \sqrt{s_m/2} \\ &\leq 16\sigma\sqrt{nk/s_m} \end{aligned}$$

Claim 7 bounds the number of non-good nodes as fewer than $s_m/2$. The probability we pick a good node for each of the first k nodes can be crudely bounded as at least 2^{-k} . A more refined analysis gives us a lower bound of $k^{-1/2}$. Note that for each round, if we select a good v_i , T_i will include all other good nodes from v_i 's part in ψ , and no good nodes from other parts of ψ . With probability $k^{-1/2}$, after the k th iteration we have removed all the good nodes, leaving fewer than $s_m/2$ nodes.

At this point, we return to the inequality we are trying to prove. Observe that as all T_i are centered around good nodes (the v_i), every v that contributes to $\alpha_{\ell i}$ must satisfy

$$|A_v^T - P_{\widehat{A}^T}\widehat{A}_v^T| \geq |A_\ell^T - A_i^T|/4$$

We will now consider a series of inequalities.

$$\begin{aligned} \sum_{i,\ell} \alpha_{\ell i} (P_{\ell j} - P_{ij})^2 &\leq \sum_{i,\ell} \alpha_{\ell i} |A_\ell^T - A_i^T|^2 / s_j \\ &\leq \sum_v 16|A_v^T - P_{\widehat{A}^T}\widehat{A}_v^T|^2 / s_j \\ &= 16|A^T - P_{\widehat{A}^T}\widehat{A}^T|_F^2 / s_j \\ &\leq 128k|A - \widehat{A}|^2 / s_j \end{aligned}$$

which concludes the proof. \square

Lemma 16 *For the terms defined above, and letting*

$$P_{\widehat{c}} = \mathbf{CProj}(\widehat{A}, k, s_m, 32\sigma\sqrt{nk/s_m})$$

it is the case that

$$|P_{\widehat{c}}B_i - B_i| \leq 64\sigma\sqrt{nk/s_i}$$

Proof: If we let \widehat{s}_i be the size of set T_i , then we can write

$$\widehat{c}_i^T B_j = \widehat{s}_i P_{ij} + \sum_{\ell} \alpha_{\ell i} (P_{\ell j} - P_{ij})$$

And so if $v \in T_i$ the v th coordinate of $P_{\widehat{c}}B_j$ is

$$(P_{\widehat{c}}B_j)_v = P_{ij} + \sum_{\ell} \alpha_{\ell i} (P_{\ell j} - P_{ij}) / \widehat{s}_i$$

Let us now break the vector $P_{\widehat{c}}B_j$ along these lines. Assuming $v \in T_i$, we define E_j and F_j by defining their values at each coordinate v .

$$\begin{aligned} E_{vj} &= P_{ij} \\ F_{vj} &= \sum_{i,\ell} \alpha_{\ell i} (P_{\ell j} - P_{ij}) / \widehat{s}_i \end{aligned}$$

It should be clear that $P_{\widehat{c}}B_j = E_j + F_j$. Observe that

$$|B_j - E_j|^2 = \sum_{i,\ell} \alpha_{\ell i} (P_{\ell j} - P_{ij})^2$$

and that (letting i vary with v)

$$\begin{aligned} |F_j|^2 &= \sum_v \sum_{\ell} \alpha_{\ell i}^2 (P_{\ell j} - P_{ij})^2 / \widehat{s}_i^2 \\ &= \sum_i \widehat{s}_i \sum_{\ell} \alpha_{\ell i}^2 (P_{\ell j} - P_{ij})^2 / \widehat{s}_i^2 \\ &\leq \sum_{i,\ell} \alpha_{\ell i} (P_{\ell j} - P_{ij})^2 \end{aligned}$$

We conclude, through the help of Lemma 15, that

$$\begin{aligned} |P_{\widehat{c}}B_j - B_j| &\leq |F_j| + |E_j - B_j| \\ &\leq 64\sigma \sqrt{nk/s_i} \end{aligned}$$

□

4 Observations and Extensions

4.1 Implementation Details

For the most part, the question of average case multisection, coloring, and clique is: “for what range of parameters can we find a solution in poly-time?” The algorithm presented in this paper also has the desirable property that it is not slow. Aside from the computation of the matrix $P_{\widehat{A}}$, we require $O(nk^2 + mk)$ time to partition and classify nodes.

Several recent papers have begun to address the problem of efficiently computing approximations to $P_{\widehat{A}}\widehat{A}$. The optimality of $P_{\widehat{A}}\widehat{A}$ is not truly required; if one were to produce a rank k matrix X instead of $P_{\widehat{A}}\widehat{A}_k$, the term $|\widehat{A} - X|$ could be introduced into the bound, replacing occurrences of $|A - \widehat{A}|$. Achlioptas and McSherry [1] examine the problem of quickly computing approximations which have bounded spectral norm. Their technique adds random noise to each entry, increasing σ but having no other effect on our algorithm. Through a trivial analysis, their technique enables partitioning in sub-linear time for a certain (non-trivial) range of parameters.

4.2 General Graph Partitioning

Our restriction to unweighted graphs is purely artificial. The bound of Theorem 10 applies equally well to weighted, non-symmetric matrices (at the expense of a $\sqrt{2}$ term). Perhaps even more interesting, at no point have we actually required that our matrices be square. Our analyses have tacitly assumed this, but they can easily be rewritten in terms of n_1 and n_2 , should the input matrix be $n_1 \times n_2$ dimensional.

We can also analyze the case where G is not of low rank, but is well represented by a low rank matrix. In other words, we can view it as a low rank matrix plus a small norm *deterministic* error. Our results do not immediately translate, due principally to the arbitrary projection this error may have. If this quantity is bounded our proofs will hold.

4.3 Parameterless Partitioning

The principal open questions in [8] involve the problem of partitioning a graph when either the part sizes or number of parts is unknown. In this paper we need only a lower bound on the sizes, and an upper bound on the number of parts. These two bounds do occur in the requisite lower bound on $|G_u - G_v|$, and if they are too loose we risk not satisfying this bound. Otherwise, the algorithm performs properly, even without precise information about the size and number of parts.

4.4 Future Directions

One significant open question with respect to this work is: “Is **CProj** necessary?” Our analysis of $P_{\widehat{A}}$ was defeated by our inability to analyze the projection of a low variance random vector onto the perturbed singular vectors. It seems reasonable to believe that with probability $1 - \delta$

$$|P_{\widehat{A}}(B_u - \widehat{B}_u)| \in O(\sigma \sqrt{k \log(n/\delta)})$$

At present, this is only speculation.

5 Acknowledgements

The author would like to thank the many people who read and commented on this paper, including particularly Dimitris Achlioptas, Amos Fiat, and Anna Karlin. The author would also like to thank Kaustubh Deshmukh, Ashish Sabharwal, and Pradeep Shenoy for their helpful comments on the proof of Lemma 15.

References

- [1] Dimitris Achlioptas and Frank McSherry, *Fast computation of low rank matrix approximations*, Proceedings of the thirty-third annual ACM symposium on the theory of computing, 2001, pp. 611 – 618.
- [2] Noga Alon and Nabil Kahale, *A spectral technique for coloring random 3-colorable graphs*, SIAM Journal on Computing **26** (1997), no. 6, 1733–1748.
- [3] Noga Alon, Michael Krivelevich, and Benny Sudakov, *Finding a large hidden clique in a random graph*, Random Structures and Algorithms **13** (1998), 457–466.

- [4] Yossi Azar, Amos Fiat, Anna Karlin, Frank McSherry, and Jared Saia, *Data mining through spectral analysis*, Symposium on the Theory of Computation (2001), 619 – 626.
- [5] Avrim Blum and Joel Spencer, *Coloring random and semi-random k -colorable graphs*, Journal of Algorithm **19** (1995), 204 – 234.
- [6] Ravi B. Boppana, *Eigenvalues and graph bisection: An average-case analysis*, Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, 1985, pp. 280 – 285.
- [7] Thang N. Bui, Soma Chaudhuri, F. Thomas Leighton, and Michael Sipser, *Graph bisection algorithms with good average case behavior*, Proceedings of the 25th IEEE Symposium on the Foundations of Computer Science, 1984, pp. 181–192.
- [8] Anne Condon and Richard Karp, *Algorithms for graph partitioning on the planted partition model*, Random Structure and Algorithms **8** (1999), no. 2, 116 – 140.
- [9] Chandler Davis and William Kahan, *The rotation of eigenvectors by a perturbation 3*, SIAM Journal on Numerical Analysis **7** (1970), 1–46.
- [10] Martin E. Dyer and Alan M. Frieze, *Fast solution of some random NP-hard problems*, IEEE Symposium on Foundations of Computer Science, 1986, pp. 331–336.
- [11] Uriel Feige and Joe Kilian, *Heuristics for finding large independent sets, with applications to coloring semi-random graphs*, Proceedings of the 39th Annual Symposium on Foundations of Computer Science (1998), 674 – 683.
- [12] Zoltán Füredi and János Komlós, *The eigenvalues of random symmetric matrices*, Combinatorica **1** (1981), no. 3, 233–241.
- [13] Gene Golub and Charles Van Loan, *Matrix computations*, third ed., The Johns Hopkins University Press, Baltimore and London, 1996.
- [14] Mark Jerrum and Gregory B. Sorkin, *Simulated annealing for graph bisection*, IEEE Symposium on Foundations of Computer Science, 1993, pp. 94–103.
- [15] Ravi Kannan, Santosh Vempala, and Adrian Vetta, *On clusterings: good, bad and spectral*, IEEE Symposium on Foundations of Computer Science, 2000.
- [16] Richard Karp, *The probabilistic analysis of some combinatorial search algorithms*, Algorithms and Complexity, New Directions and Recent Results (New York) (J. F. Traub, ed.), Academic Press, 1976, pp. 1–20.
- [17] Michael Krivelevich and Van H. Vu, *On the concentration of eigenvalues of random symmetric matrices*, Microsoft Technical Report, no. 60, 2000.
- [18] Ludek Kucera, *Expected behavior of graph colouring algorithms*, Lecture Notes Comput. Sci **56**, 1977, pp. 447 – 451.
- [19] Ludek Kucera *Expected complexity of graph partitioning problems*, DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science **57** (1995).
- [20] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala, *Latent semantic indexing: A probabilistic analysis*, 1998, pp. 159–168.
- [21] Daniel Spielman and Shang Hua Teng, *Spectral partitioning works: Planar graphs and finite element meshes*, Proc. 37th Conf. on Foundations of Computer Science, 1996, pp. 96–105.
- [22] Jonathan Turner, *Almost all k -colorable graphs are easy to color*, Journal of Algorithms **9** (1988), no. 1, 63–82.