Spectral Clustering by Recursive Partitioning

Anirban Dasgupta[⋆1], John Hopcroft[2], Ravi Kannan[3], and Pradipta Mitra[⋆⋆3]

[1] Yahoo! Research Labs
[2] Department of Computer Science, Cornell University
[3] Department of Computer Science, Yale University

In this paper, we analyze the second eigenvector technique of spectral partitioning on the planted partition random graph model, by constructing a recursive algorithm using the second eigenvectors in order to learn the planted partitions. The correctness of our algorithm is not based on the ratio-cut interpretation of the second eigenvector, but exploits instead the stability of the eigenvector subspace. As a result, we get an improved cluster separation bound in terms of dependence on the maximum variance. We also extend our results for a clustering problem in the case of sparse graphs.

1 Introduction

Clustering of graphs is an extremely general framework that captures a number of important problems on graphs. In a general setting, the clustering problem is to partition the vertex set of a graph into "clusters", where each cluster contains vertices of only "one type". The exact notion of what the vertex "type" represents is dependent on the particular application of the clustering framework. We will deal with the clustering problem on graphs generated by the versatile planted partition model (See [18, 5]). In this probabilistic model, the vertex set of the graph is partitioned into k subsets $T_1, T_2, \ldots, T_k$. Each edge (u, v) is then a random variable that is independently chosen to be present with a probability A_{uv}, and absent otherwise. The probabilities A_{uv} depend only on the parts to which the two endpoints u and v belong. The adjacency matrix $\widehat{A}$ of the random graph so generated is presented as input. Our task then is to identify the latent clusters $T_1, T_2, \ldots, T_k$ from $\widehat{A}$.

Spectral methods have been widely used for clustering problems, both for theoretical analysis as well as empirical and application areas. The underlying idea is to use information about the eigenvectors of $\hat{A}$ to extract structure. There are different variations to this basic theme of spectral clustering, which can be essentially divided into 2 classes of algorithms.

1. Projection heuristics, in which the top few eigenvectors of the adjacency matrix $\widehat{A}$ are used to construct a low-dimensional representation of the data, which is then clustered.
2. The second eigenvector heuristic, in which the coordinates of the second eigenvector of $\widehat{A}$ is used to find a split of the vertex set into two parts. This technique is then applied recursively to each of the parts obtained.

⋆ Work done when author was at Cornell University
⋆⋆ Supported by NSF's ITR program under grant number 0331548

Experimental results claiming the goodness of both spectral heuristics abound. Relatively fewer are results that strive to demonstrate provable guarantees about the heuristics. Perhaps more importantly, the worst case guarantees [17] that have been obtained do not seem to match the stellar performance of spectral methods on most inputs, and thus it is still an open question to characterize the class of inputs for which spectral heuristics do work well. In order to be able to formalize the average case behavior of spectral analysis, researchers have analyzed its performance on graphs generated by random models with latent structure [4, 18]. These graphs are generated by zero-one entries from a independently chosen according to a low-rank probability matrix. The low rank of the probability matrix reflects the small number of vertex types present in the unperturbed data. The intuition developed by Azar et al. [4] is that in such models, the random perturbations may cause the individual eigenvectors to vary significantly, but the subspace spanned by the top few eigenvectors remains stable. From this perspective, however, the second eigenvector technique does not seem to be well motivated, and it remains an open question as to whether we can claim anything better than the worst case bounds for the second eigenvector heuristic in this setting.

In this paper, we prove the goodness of the second eigenvector partitioning for the planted partition random graph model [11, 4, 18, 10]. We demonstrate that in spite of the fact that the second eigenvector itself is not stable, we can use it to recover the embedded structure.

Our main aim in analyzing the planted partition model using the second eigenvector technique is to try to bridge the gap between the worst case analysis and the actual performance. However, in doing so, we achieve a number of other goals too. The most significant among these is that we can get tighter guarantees than [18] in terms of the dependence on the maximum variance. The required separation between the columns clusters T_r and T_s can now be in terms of $\sigma_r + \sigma_s$, the maximum variances in each of these two clusters, instead of the maximum variance $\sigma_{\max}$ in the entire matrix. This gain could be significant if the maximum variance $\sigma_{\max}$ is due to only one cluster, and thus can potentially lead to identification of a finer structure in the data. Our separation bounds are however worse than [18, 1] in terms of dependence on the number of clusters. Another contribution of the paper is to model and solve a restricted clustering problem for sparse (constant degree) graphs. Graphs clustered in practice are often "sparse", even of very low constant degree. A concern about analysis of many heuristics on random models [18, 10] is that they don't cover sparse graphs. In this paper, we propose a model motivated by random regular graphs (see [14, 6], for example) for the clustering problem that allows us to use strong concentration results which are available in that setting. We will use some extra assumptions on the degrees of the vertices and finally show that expansion properties of the model will allow us to achieve a clean clustering through a simple algorithm.

2 Model and our results

A is a matrix of probabilities where the entry A_{uv} is the probability of an edge being present between the vertices u and v. The vertices are partitioned into k clusters T_1, $T_2, \ldots, T_k$. The size of the r^{th} cluster T_r is n_r and the minimum size is denoted by $n_{\min} = \min_r\{n_r\}$. Let, $w_{\min} = n_{\min}/n$. We assume that the minimum size $n_{\min} \in \Omega(n/k)$. The characteristic vector of the cluster T_r is denoted by $\boldsymbol{g}^{(r)}$ defined as $\boldsymbol{g}^{(r)}(i) = 1/\sqrt{n_r}$ for $i \in T_r$ and 0 elsewhere. The probability A_{uv} depends only on the two clusters in which the vertices u and v belong to. Given the probability matrix A, the random graph $\widehat{A}$ is then generated by independently setting each $\widehat{A}_{uv} (= \widehat{A}_{vu})$ to 1 with probability A_{uv} and 0 otherwise. Thus, the expectation of the random variable $\widehat{A}_{uv}$ is equal to A_{uv}. The variance of $\widehat{A}_{uv}$ is thus $A_{uv}(1 - A_{uv})$. The maximum variance of any entry of $\widehat{A}$ is denoted σ^2, and the maximum variance for all vertices belonging to a cluster T_r as denoted as σ_r^2. We usually denote a matrix of random variables by $\widehat{X}$ and the expectation of $\widehat{X}$ as $X = \mathbf{E}\left[\widehat{X}\right]$. We will also denote vectors by boldface (e.g. $\boldsymbol{x}$). $\boldsymbol{x}$ has the i^{th} coordinate $\boldsymbol{x}(i)$. For a matrix X, X_i denotes the column i. The number of vertices is n. We will assume the following separation condition.

Separation Condition. Each of the variances σ_r satisfies $\sigma_r^2 \geq \log^6 n/n$. Furthermore, there exists a large enough constant c such that for vertices $u \in T_r$ and $v \in T_s$, the columns A_u and A_v of the probability matrix A corresponding to different clusters T_r and T_s satisfy

$$\|A_u - A_v\|_2^2 \geq 64c^2k^5 \left(\sigma_r + \sigma_s\right)^2 \frac{\log(n)}{w_{\min}} \tag{1}$$

For clarity of exposition, we will make no attempt to optimize the constants or exponents of k. Similarly, we will ignore the term $w_{\min}$ for the most part. We say that a partitioning $(S_1, \ldots, S_l)$, respects the original clustering if the vertices of each T_r lie wholly in any one of the S_j. We will refer to the parts S_j as super-clusters, being the union of one or more clusters T_r. We say that a partitioning $(S_1, \ldots, S_l)$ agrees with the underlying clusters if each S_i is exactly equal to some T_r (i.e. $l = k$). The aim is to prove the following theorem.

Theorem 1. *Given $\widehat{A}$ that is generated as above, i.e. $A = \mathbf{E}\left[\widehat{A}\right]$ satisfies condition 1, we can cluster the vertices such that the partitioning agrees with the underlying clusters with probability at least $1 - \frac{1}{n^\delta}$, for suitably large δ.*

3 Related Work

The second eigenvector technique has been analyzed before, but mostly from the viewpoint of constructing cuts in the graph that have a small ratio of edges cut to vertices separated. There has been a series of results [13, 2, 5, 19] relating

the gap between the first and second eigenvalues, known as the Fiedler gap, to the quality of the cut induced by the second eigenvector. Spielman and Teng [20] demonstrated that the second eigenvector partitioning heuristic is good for meshes and planar graphs. Kannan et al. [17] gave a bicriteria approximation for clustering using the second eigenvector method. Cheng et al. [7] showed how to use the second eigenvector method combined with a particular cluster objective function in order to devise a divide and merge algorithm for spectral clustering. In the random graph setting, there has been results by Alon et al. [3], and Cojaoghlan [8] in using the coordinates of the second eigenvector in order to perform coloring, bisection and other problems. In each of these algorithms, however, the cleanup phase is very specific to the particular clustering task at hand.

Experimental studies done on the relative benefits of the two heuristics often show that the two techniques outperform each other on different data sets [21]. In fact results by Meila et al.[21] demonstrate that the recursive methods using the second eigenvector are actually more stable than the multiway spectral clustering methods if the noise is high. Anopther paper by Zhao et al. [23] shows that recursive clustering using the second eigenvector performs better than a number of other hierarchical clustering algorithms.

4 Algorithm

For the sake of simplicity, in most of the paper, we will be discussing the basic bipartitioning step that is at the core of of our algorithm. In Section 4.2 we will describe how to apply it recursively to learn all the k clusters. Define the matrix $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$. Note that for any vector $\boldsymbol{z}$ such that $\sum_i \boldsymbol{z}_i = 0$, $J\boldsymbol{z} = \boldsymbol{z}$. Given the original matrix $\widehat{A}$ we will create $\Theta(n/(k \log n))$ submatrices by partitioning the set of rows into $\Theta(n/(k \log n))$ parts randomly. Suppose $\widehat{C}$ denotes any one of these parts. Given the matrix $\widehat{C}$ as input, we will first find the top right singular vector $\boldsymbol{u}$ of the matrix $\widehat{C}J$. The coordinates of this vector will induce a mapping from the columns (vertices) of $\widehat{C}$ to the real numbers. We will find a large "gap" such that substantial number of vertices are mapped to both sides of the gap. This gives us a natural bipartition of the set of vertices of $\widehat{C}$. We will prove that this classifies all vertices correctly, except possibly a small fraction. This will be shown in Lemmas 2 to 6. We next need to "clean up" this bi-partitioning, and this will be done using a correlation graph construction along with a Chernoff bound. The algorithm and a proof will be furnished in Lemma 7. This completes one stage of recursion in which we create a number of superclusters all of which respect the original clustering. Subsequent stages proceed similarly. In what follows, we will be using the terms "column" and "vertex" interchangeably, noting that vertex x corresponds to column $\widehat{C}_x$.

4.1 Proof

For the standard linear algebraic techniques used in this section, we refer the reader to [16]. Recall that each $\widehat{C}$ is a $\frac{n}{2k \log n} \times n$ matrix where the rows are

chosen randomly. Denote the expectation of $\widehat{C}$ by $\mathbf{E}\left[\widehat{C}\right] = C$, and by $\boldsymbol{u}$ the top right singular vector of $\widehat{C}J$, i.e. the top eigenvector of $(\widehat{C}J)^T\widehat{C}J$. In what follows, we demonstrate that for each of random submatrices $\widehat{C}$, we can utilize the second right singular vector $\boldsymbol{u}$ to create a partitioning of the columns of $\widehat{C}J$ that respects the original clustering. The following fact is intuitive and will be proven later in lemma 8, when we illustrate the full algorithm.

Fact 1 *$\widehat{C}$ has at least $\frac{n_r}{2k\log n}$ rows for each cluster T_r.*

Let $\sigma = \max_r\{\sigma_r\}$, where the maximum is taken only over clusters present in $\widehat{C}$ (and therefore, potentially much smaller than $\sigma_{\max}$). We also denote $C(r,s)$ for the entries of C corresponding to vertices of T_r and T_s. The following result is from Furedi-Komlos and more recently, Vu [22, 15] claiming that a matrix of i.i.d. random variables is close to its expectation in the spectral norm.

Lemma 2. (Furedi, Komlos; Vu) *If $\widehat{X}$ is a 0/1 random matrix with expectation $X = \mathbf{E}\left[\widehat{X}\right]$, and the maximum variance of the entries of $\widehat{X}$ is σ^2 which satisfies $\sigma^2 \geq \log^6 n/n$,[4] then with probability $1 - o(1)$,*

$$\|X - \widehat{X}\|_2 < 3\sigma\sqrt{n}$$

In particular, we have $\|C - \widehat{C}\|_2 < 3\sigma\sqrt{n}$.

The following lemmas will show that the top right singular vector $\boldsymbol{u}$ of $\widehat{C}J$ gives us an approximately good bi-partition.

Lemma 3. *The first singular value λ_1 of the expected matrix CJ satisfies $\lambda_1(CJ) \geq 2c(\sigma_r + \sigma_s)k^2\sqrt{n}$ for each pair of clusters r and s that belong to C. Thus, in particular, $\lambda_1(CJ) \geq 2c\sigma k^2\sqrt{n}$.*

Proof. Suppose $\widehat{C}$ has the clusters T_r and T_s, $r \neq s$. Assume $n_r \leq n_s$. Consider the vector $\boldsymbol{z}$ defined as :

$$\boldsymbol{z}_x = \begin{cases} \frac{1}{\sqrt{2n_r}} & \text{if } x \in T_r \\ -\frac{\sqrt{n_r}}{n_s\sqrt{2}} & \text{if } x \in T_s \\ 0 & \text{otherwise} \end{cases}$$

Now, $\sum_x \boldsymbol{z}(x) = \frac{n_r}{\sqrt{2n_r}} - \frac{\sqrt{n_r}}{\sqrt{2}n_s}n_s = 0$. Also, $\|\boldsymbol{z}\|^2 = \frac{n_r}{2n_r} + \frac{n_r n_s}{2n_s^2} = \frac{1}{2} + \frac{1}{2}\frac{n_r}{n_s} \leq 1$. Clearly, $\|\boldsymbol{z}\| \leq 1$. For any row C^j from a cluster T_t, it can be shown that $C^j \cdot \boldsymbol{z} = \sqrt{\frac{n_r}{2}}(C(r,t) - C(s,t))$. We also know from fact 1 that there are at least

[4] In fact, in light of recent results in [12] this holds for $\sigma^2 \geq C' \log n/n$, with a different constant in the concentration bound.

$n_t/(2k\log n)$ such rows. Now,

$$\begin{aligned}\|CJ\boldsymbol{z}\|^2 &\geq \sum_j (C^j\cdot\boldsymbol{z})^2 = \sum_t\sum_{j\in T_t}(C^j\cdot\boldsymbol{z})^2 \geq \sum_t \frac{n_t}{2k\log n}\frac{n_r}{2}(C(r,t)-C(s,t))^2\\ &= \frac{n_r}{4k\log n}\sum_t n_t(C(r,t)-C(s,t))^2\frac{n_r}{4k\log n}\|C_r-C_s\|_2^2\\ &\geq 64\frac{n_r}{4k\log n}c^2k^5\left(\sigma_r+\sigma_s\right)^2\log(n)/w_{\min}\geq 16c^2nk^4(\sigma_r+\sigma_s)^2\end{aligned}$$

using the separation condition and the fact that n_r is at least $w_{\min}n$. And thus $\lambda_1(CJ)$ is at least $4c(\sigma_r+\sigma_s)k^2\sqrt{n}$. Note that the 4th step uses the separation condition (1). □

The above result, combined with the fact the the spectral norm of the random perturbation being small immediately implies that the norm of the matrix $\widehat{C}J$ is large too. Thus,

Lemma 4. *The top singular value of $\widehat{C}J$ is at least $c\sigma k^2\sqrt{n}$.*

Proof. Proof omitted.

Lemma 5. *The vector $\boldsymbol{u}$, the top right singular vector of $\widehat{C}J$ can be written as $\boldsymbol{u}=\boldsymbol{v}+\boldsymbol{w}$ where both $\boldsymbol{v},\boldsymbol{w}$ are orthogonal to $\mathbf{1}$ and further, $\boldsymbol{v}$ is a linear combination of the indicator vectors $\boldsymbol{g}^{(1)},\boldsymbol{g}^{(2)},\ldots$ for clusters T_r that have vertices in the columns of $\widehat{C}$. Also, $\boldsymbol{w}$ sums to zero on each T_r. Moreover,*

$$\|\boldsymbol{w}\|\leq\frac{4}{ck^2} \tag{2}$$

Proof. We may define the two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ as follows: $\boldsymbol{v}=\sum_r(\boldsymbol{g}^{(r)}\cdot\boldsymbol{u})\boldsymbol{g}^{(r)},\boldsymbol{w}=\boldsymbol{u}-\boldsymbol{v}$.

It is easy to check that $\boldsymbol{w}$ is orthogonal to $\boldsymbol{v}$, and that $\sum_{x\in T_r}\boldsymbol{w}(x)=0$ on every cluster T_r. Thus both $\boldsymbol{v}$ and $\boldsymbol{w}$ are orthogonal to $\mathbf{1}$. As $\boldsymbol{v}$ is orthogonal to $\boldsymbol{w}$, $\|\boldsymbol{v}\|^2+\|\boldsymbol{w}\|^2=\|\boldsymbol{u}\|^2=1$. Now,

$$\begin{aligned}\lambda_1(\widehat{C}J)=\|\widehat{C}J\boldsymbol{u}\|&\leq\|\widehat{C}J\boldsymbol{v}\|+\|\widehat{C}J\boldsymbol{w}\|\leq\lambda_1(\widehat{C}J)\|\boldsymbol{v}\|+\|CJ\boldsymbol{w}\|+\|CJ-\widehat{C}J\|\|\boldsymbol{w}\|\\ &\leq\lambda_1(\widehat{C}J)(1-\|\boldsymbol{w}\|^2/2)+\|C-\widehat{C}\|\|\boldsymbol{w}\|\end{aligned}$$

using the fact that $(1-x)^{1/2}\leq 1-\frac{x}{2}$ for $0\leq x\leq 1$, and also noting that $J\boldsymbol{w}=\boldsymbol{w}$, and therefore $CJ\boldsymbol{w}=C\boldsymbol{w}=0$. Thus, from the above, $\|\boldsymbol{w}\|\leq\frac{2\|C-\widehat{C}\|}{\lambda_1(\widehat{C}J)}\leq\frac{4\sigma\sqrt{n}}{c\sigma k^2\sqrt{n}}\leq\frac{4}{ck^2}$ using Lemma 2 and Lemma 4. □

We now show that in bi-partitioning each $\widehat{C}$ using the vector $\boldsymbol{u}$, we only make mistakes for a small fraction of the columns.

Lemma 6. *Given the top right singular vector $\boldsymbol{u}$ of $\widehat{C}$, there is a way to bipartition the columns of $\widehat{C}$ based on $\boldsymbol{u}$, such that all but $\frac{n_{\min}}{ck}$ columns respect the underlying clustering of the probability matrix C.*

Proof. Consider the following algorithm. Consider the real values $\boldsymbol{u}(x)$ corresponding to the columns $\widehat{C}_x$.

1. Find β such that at most $\frac{n}{ck^2}$ of the $\boldsymbol{u}(x)$ lies in $(\beta, \beta + \frac{2}{k\sqrt{n}})$. Moreover, define $L = \{x : \boldsymbol{u}(x) < \beta + \frac{1}{k\sqrt{n}}\}; R = \{x : \boldsymbol{u}(x) \geq \beta + \frac{1}{k\sqrt{n}}\}$. It must be that both $|L|$ and $|R|$ are at least $n_{\min}/2$. Note that $\widehat{C} = L \cup R$. If we cannot find any such gap, don't proceed (a cluster has been found that can't be partitioned further).
2. Take $L \cup R$ as the bipartition.

We must show that, if the vertices contain at least two clusters, a gap of $\frac{2}{k\sqrt{n}}$ exists with at least $n_{\min}/2$ vertices on each side. For simplicity, for this proof we assume that all clusters are of equal size (the general case will be quite similar). Let $\boldsymbol{v} = \sum_{r=1}^{k} \alpha_r \boldsymbol{g}^{(r)}$. Recall that $\boldsymbol{v}$ is orthogonal to $\mathbf{1}$, and thus $\sum_{r=1}^{k} \alpha_r \sqrt{\frac{k}{n}} = 0$. Now note that $1 = \|\boldsymbol{u}\|^2 = \|\boldsymbol{v}\|^2 + \|\boldsymbol{w}\|^2$. This and lemma 5 gives us

$$\sum_{r=1}^{k} \alpha_r^2 \geq 1 - \frac{16}{c^2 k} \geq \frac{1}{2} \tag{3}$$

We claim that there is an interval of $\Theta\left(\frac{1}{k\sqrt{k}}\right)$ on the real line such that no α_r lies in this interval and at least one α_r lies on each side of the interval. We will call such a gap a "proper gap". Note that a proper gap will partition the set of vertices into two parts such that there are at least $n_{\min}/2$ vertices on each side of it.

The above claim can be proved using basic algebra. We will omit details here. Thus, it can been seen that for some constant c, there will be a proper gap of $\frac{1}{ck\sqrt{n}}$ in the vector $\boldsymbol{v}$. We then argue that most of the coordinates of $\boldsymbol{w}$ are small and do not spoil the gap. Since the norm of $\|\boldsymbol{w}\|^2$ is bounded by $16/(c^2k^4)$, it is straightforward to show that at most $\frac{n}{ck^2}$ vertices x can have $\boldsymbol{w}(x)$ over $\frac{4}{k\sqrt{cn}}$. This shows that for most vertices $\boldsymbol{w}(x)$ is small and will not "spoil" the proper gap in $\boldsymbol{v}$. Thus, with high probability, the above algorithm of finding a gap in $\boldsymbol{u}$ always succeeds. Next we have to show that any such gap that is found from $\boldsymbol{u}$ actually corresponds to a proper gap in $\boldsymbol{v}$. Since there must be at least $n_{\min}/2$ vertices on each side of the gap in $\boldsymbol{u}$, and since the values $\boldsymbol{u}(x)$ and $\boldsymbol{v}(x)$ are close (i.e. $\boldsymbol{w}(x) = \boldsymbol{u}(x) - \boldsymbol{v}(x)$ is smaller than $1/(2k\sqrt{n})$) except for $\frac{n}{ck}$ vertices, it follows that a proper gap found in $\boldsymbol{u}$ must correspond to a proper gap in $\boldsymbol{v}$. Thus the only vertices that can be misclassified using this bi-partition are the vertices that are either in the gap, or have $\boldsymbol{w}(x)$ larger than $\frac{1}{k\sqrt{n}}$. Given this claim, it can be seen a using a proper gap a bi-partition of the vertices can be found with at most $\frac{n}{2ck^2} \approx \Theta\left(\frac{n_{\min}}{ck}\right)$ vertices on the wrong side of the gap. □

A natural idea for the "clean up" phase would be to use $\log n$ independent samples of $\widehat{C}$ (thus requiring the $\log n$ factor in the separation) and try to use a Chernoff bound argument. This argument doesn't work, unfortunately, the

reason being that the singular vector can induce different bi-partitions for each of the $\widehat{C}$'s. For instance, if there are 3 clusters in the original data, then in the first step we could split any one of the three clusters from the other two. This means a naive approach will need to account for all possible bi-partitionings and hence require an extra 2^k in the separation condition. The following lemma deals with this problem:

Lemma 7. *Suppose we are given set V that is the union of a number of clusters $T_1 \cup \ldots \cup T_t$. Given $p = ck \log n$ independent bi-partitions of the set of columns V, such that each bi-partition agrees with the underlying clusters for all but $\frac{n_{\min}}{4ck}$ vertices, there exists an algorithm that, with high probability, will compute a partitioning of the set V such that*

- *The partitioning respects the underlying clusters of the set V.*
- *The partitioning is non-trivial, that is, if the set V contains at least two clusters, then the algorithm finds at least two partitions.*

Proof. Consider the following algorithm. Denote $\varepsilon = \frac{1}{4ck}$.

1. Construct a (correlation) graph H over the vertex set V.
2. Two vertices x and y are adjacent if they are on the same L or R for at least $(1 - 2\varepsilon)$ fraction of the bi-partitions.
3. Let $N_1, \ldots, N_l$ be the connected components of this graph. Return $N_1, \ldots, N_l$.

We now need to prove that the following claims hold with high probability : 1. N_j respects the cluster boundary, i.e. each cluster T_r that is present in V satisfies $T_r \subseteq N_{j_r}$ for some j_r; and 2. If there are at least two clusters present in V, i.e. $t \geq 2$, then there are at least two components in H. For two vertices $x, y \in H$, let the **support** $s(x, y)$ equal the fraction of tests such that x and y are on the same side of the bi-partition. For the first claim, we define a vertex x to be a "bad" vertex for the i^{th} test if $|w(x)| > \frac{1}{k\sqrt{cn}}$. From lemma 6 the number of bad vertices is clearly at most $\frac{1}{ck} n_{\min}$. It is clear that a misclassified vertex x must either lie in the gap $(\beta, \beta + \frac{2}{k\sqrt{n}})$ or it must be a bad one. So for any vertex x, the probability that x is misclassified in the i^{th} test is at most $\varepsilon = 1/(4ck)$. If there are p tests, then the expected times that a vertex x is misclassified is at most εp. Supposing Y_x^i is the indicator random variable for the vertex x being misclassified in the i^{th} test. Thus, $\mathbf{Pr}\left[\sum_i Y_x^i > 2\varepsilon p\right] < \exp\left(-\frac{16p}{ck}\right) < \frac{1}{n^3}$ since $p = ck \log n$. Thus, each pair of vertices in a cluster, are on the same side of the bipartition for at least $(1 - 2\varepsilon)$ fraction of the tests. Clearly, the components N_j always obey the cluster partitions.

Next, we have to prove the second claim. For contradiction, assume there is only one connected component. We know, that if $x, y \in T_r$ for some r, the fraction of tests on which they landed on same side of partition is $s(x, y) \geq (1 - 2\varepsilon)$. Hence the subgraph induced by each T_r is complete. With at most k clusters in V, this means that any two vertices x, y (not necessarily in the same cluster) are separated by a path of length at most k. Clearly $s(x, y) \geq (1 - 2k\varepsilon)$. Hence, the

total support of inter-cluster vertex pairs is

$$\sum_{r\neq s}\sum_{x\in T_r, y\in T_s} s(x,y) \geq (1-2k\varepsilon)\sum_{r\neq s} n_r n_s \geq \sum_{r\neq s} n_r n_s - 2k\varepsilon \sum_{r\neq s} n_r n_s. \quad (4)$$

Let us count this same quantity by another method. From Lemma 6, it is clear that for each test at least one cluster was separated from the rest (apart from small errors). Since by the above argument, all but ε vertices are good, we have that, at least $n_{\min}(1-\varepsilon)$ vertices were separated from the rest. Hence the total support is

$$\sum_{r\neq s}\sum_{x\in T_r, y\in T_s} s(x,y) \leq \sum_{r\neq s} n_r n_s - n_{\min}(1-\varepsilon)(n - n_{\min}(1-\varepsilon)) < \sum_{r\neq s} n_r n_s - n_{\min} n/2$$

But this contradicts equation 4 if $2k\varepsilon \sum_{r\neq s} n_r n_s < n_{\min} n/2$ i.e. $\varepsilon < \frac{n_{\min} n/4}{k\sum_{r\neq s} n_r n_s} < \frac{n_{\min} n/2}{kn^2} \leq \frac{1}{2ck}$. With the choice of $\varepsilon = 1/(4ck)$, we get a contradiction. Hence the correlation graph satisfies the properties claimed. □

4.2 Final Algorithm

We now describe the complete algorithm. Basically, it is the bi-partitioning technique presented in the previous section repeated (at most) k times applied to the matrix $\widehat{A}$.

Algorithm 1 Cluster $(\widehat{A}, k)$

Partition the set of rows into k random equal parts, each part to be used in the corresponding step of recursion. Name the i^{th} part to be $\widehat{B}_i$.
Let $(S_1, \ldots, S_l)$ =**Bi-Partition** $(\widehat{B}_1, k)$.
Recursively call **Bi-Partition** on each of S_i, and on each of the results, using the appropriate columns of a separate $\widehat{B}_j$ for each call. The recursion ends when the current call returns only one S_i. Let $\widehat{T}_1, \ldots, \widehat{T}_k$ be the final groups.

As the split in every level is "clean", as we have shown above, the whole analysis goes through for recursive steps without any problems. In order to de-condition the steps of the recursion, we have to first create k independent instances of the data by partitioning the rows of the matrix $\widehat{A}$ into a k equally sized randomly chosen sets. This creates a collection of rectangular matrices $\widehat{B}_1, \ldots, \widehat{B}_k$. The module **Bi-Partition**$(\widehat{X}, k)$ on being invoked with the matrix $\widehat{X}$ and the cluster parameter k consists of two phases: an approximate partitioning by the singular vector, followed by a clean-up phase. The rows of matrix $\widehat{X}$ are further subdivided to create a number of rectangular matrices $\widehat{C}^{(i)}$, which correspond to C that we used in our analysis of the bi-partitioning phase. One thing we still need to prove that the fact 1 made for $\hat{C}$ in the beginning of section 4.1 is valid for $C^{(i)}$

Algorithm 2 Bi-Partition $(\widehat{X}, k)$

Partition the set of rows into $c_1 \log n$ equal parts randomly. The i^{th} set of rows forms the matrix $\widehat{C}^{(i)}$.
For each $\widehat{C}^{(i)}$, find the right singular vector of $\widehat{C}^{(i)}J$ and call it $\boldsymbol{u}_i$.
Split:
Find a proper gap β, such that $(\beta, \beta + \frac{2}{k\sqrt{n}})$ has at most $\frac{n}{c_2 k}$ vertices and define

$$L_i = \{x : u_i(x) < \beta + \frac{1}{k\sqrt{n}}\}$$

$$R_i = \{x : u_i(x) \geq \beta + \frac{1}{k\sqrt{n}}\}$$

$$|L_i| \geq n_{\min}/2; |R_i| \geq n_{\min}/2$$

$$\widehat{C}^{(i)} = L_i \cup R_i$$

If no such gap exists, return.
Cleanup:
Construct a (correlation) graph with the columns of $\widehat{X}$ as the vertices.
Connect two vertices x and y if they are on the same L_i or R_i for at least $(1 - \frac{1}{2ck}) \log n$ times. Let $N_1, \ldots, N_l$ be the connected components of this graph. Return $N_1, \ldots, N_l$.

Lemma 8. *Consider each matrix* $C^{(i)} = \mathbf{E}\left[\widehat{C}^{(i)}\right]$. *W.h.p. there are at least* $\frac{n_r}{2c_1 k \log n}$ *rows in* $C^{(i)}$ *corresponding to* T_r.

Proof. In each $\widehat{C}^{(i)}$, the expected number of rows from each T_r is $\frac{n_j}{k \times c_1 \log n}$. Using Chernoff bound, the number of rows contributed by each cluster T_r to the matrix $\widehat{C}^{(i)}$ is at least $\frac{n_j}{2c_1 k \log n}$ with probability $1 - \exp[-\frac{n_j}{2c_1 k \log n}] \geq 1 - \frac{1}{n^3}$. Thus, over the all random partitions, w.p. $1 - \frac{1}{n^2}$, the statement is true.

5 Sparse Graphs

5.1 Our model and related work

The input $\hat{A}$ is a n-vertex undirected graph. There will be k clusters in the graph, with $n_r = \Omega(n)$ being the size of cluster T_r. Let x $\in T_r$. Then we assume that the number of edges from x to vertices of T_s:

$$e(x, T_s) = d_{rs} \tag{5}$$

For some constant d_{rs}. We assume that these constants satisfy $n_r d_{rs} = n_s d_{sr}$.

Let $\hat{A}^{(rs)}$ be the submatrix of $\hat{A}$ containing rows corresponding to T_r and columns corresponding to T_s. Then $\hat{A}^{(rs)}$ is a matrix randomly chosen from all matrices satisfying equation 5 (to account for symmetry $\hat{A}^{(rs)} = (\hat{A}^{(rs)})^T$).

Let $A = \mathbf{E}\left[\hat{A}\right]$. If vertex $x \in T_r$, let $A_x = \mu_r$. Note that $\mu_r(x) = \mu_s(y)$ where $y \in T_s; x \in T_r$ due to symmetry. Let d be a upper bound for vertex degree in

the graph. We will assume, for all r, and some constant c_0, $d_{rr} \geq \frac{1}{2}d + c_0\sqrt{dk}$ Which will now imply something we need: $\|\mu_r - \mu_s\|_2^2 \geq c_0^2 k^2 \frac{d}{n}$.

Among previous works on "sparse" graphs are results by Alon and Kahale [3] (3 coloring) and Coja-Oghlan [8, 9] (bisection, clustering). Our results are not comparable to theirs as those models are only sparse "on average". Nevertheless, the gap required in [8], improving on [5], is $np' - np = \Theta(\sqrt{np' \log np'})$ in a $G(n, p')$ model, which put in our terminology is $\Theta(\sqrt{d \log d})$, similar to our separation (in fact we don't need the $\log d$ factor). The separation in [3] is d. It should be emphasized again that both settings are quite different from ours. A d-regular model for bisection was studied by Bui et. al. [6]. They present an algorithm that finds bisections of width (cardinality of the bisection) $o(n^{1-1/(d/2)})$ from a graph that is randomly chosen from d-regular graphs having such a bisection. We depend on having different d_{rr}'s for different clusters for a notion of partitioning, and in any case we seek to solve a more general problem.

5.2 Algorithm

For sets (of vertices) U and W, let $e(U, W)$ be the the number of edges between U and W. Here we only present the "clean up" phase as everything else remains essentially the same. Our result is that if the separation condition holds, this

Algorithm 3 SparseCleanup(P_1, P_2, d)

loop
  find a vertex v in P_2' such that $e(v, P_1') > e(v, P_2')(1 + \frac{1}{2\sqrt{d}})$
  if no vertex can be found, end loop.
  move v to P_1'
end loop
loop
  find a vertex v in P_1' such that $e(v, P_2') > e(v, P_1')(1 + \frac{1}{2\sqrt{d}})$
  if no vertex can be found, end loop.
  move v to P_2'
end loop

algirthm will successfully cluster the vertices. We omit the proof of this fact here.

References

1. Dimitris Achlioptas and Frank McSherry, *On spectral learning of mixtures of distributions*, Conference on Learning Theory (COLT) 2005, 458-469.
2. Noga Alon, *Eigenvalues and expanders*, Combinatorica, **6(2)**, (1986) , 83-96.
3. Noga Alon and Nabil Kahale, *A spectral technique for coloring random 3-colorable graphs*, SIAM Journal on Computing **26** (1997), n. 6. 1733-1748.

4. Yossi Azar, Amos Fiat, Anna R. Karlin, Frank McSherry and Jared Saia, *Spectral analysis of data*, Proceedings of the 32^{nd} annual ACM Symposium on Theory of computing (2001), 619-626.
5. Ravi Boppana, *Eigenvalues and graph bisection: an average case analysis*, Proceedings of the 28^{th} IEEE Symposium on Foundations of Computer Science (1987).
6. Thang Bui, Soma Chaudhuri, Tom Leighton and Mike Sipser, *Graph bisection algorithms with good average case behavior*, Combinatorica, **7**, 1987, 171-191.
7. David Cheng, Ravi Kannan, Santosh Vempala and Grant Wang, *A Divide-and-Merge methodology for Clustering*, Proc. of the 24^th ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems (PODS), 196 - 205.
8. Amin Coja-Oghlan, *A spectral heuristic for bisecting random graphs*, Proceedings of the 16^th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005.
9. Amin Coja-Oghlan, *An adaptive spectral heuristic for partitioning random graphs*, Automata, Languages and Programming, 33^{rd} International Colloquium, ICALP, Lecture Notes in Computer Science 4051 Springer 2006.
10. Anirban Dasgupta, John Hopcroft and Frank McSherry, *Spectral analysis of random Graphs with skewed degree distributions*, Proceedings of the 42^{nd} IEEE Symposium on Foundations of Computer Science (2004), 602-610.
11. Martin Dyer and Alan Frieze, *Fast Solution of Some Random NP-Hard Problems*, Proceedings of the 27^{th} IEEE Symposium on Foundations of Computer Science (1986), 331-336
12. Uriel Feige and Eran Ofek, *Spectral techniques applied to sparse random graphs*, Random Structures and Algorithms, **27(2)**, 251–275, September 2005.
13. M Fiedler, *Algebraic connectibility of graphs*, Czechoslovak Mathematical Journal, **23(98)**, (1973), 298-305.
14. Joel Friedman, Jeff Kahn and Endre Szemeredi, *On the second eigenvalue of random regular graphs*, Proceedings of the 21^{st} annual ACM Symposium on Theory of computing (1989), 587 - 598.
15. Zoltan Furedi and Janos Komlos, *The eigenvalues of random symmetric matrices*, Combinatorica 1, **3**, (1981), 233–241.
16. G. Golub, C. Van Loan (1996), *Matrix computations, third edition*, *The Johns Hopkins University Press Ltd., London*.
17. Ravi Kannan, Santosh Vempala and Adrian Vetta, *On Clusterings : Good, bad and spectral*, Proceedings of the Symposium on Foundations of Computer Science (2000), 497 - 515.
18. Frank McSherry, *Spectral partitioning of random graphs*, Proceedings of the 42^{nd} IEEE Symposium on Foundations of Computer Science (2001), 529-537.
19. Alistair Sinclair and Mark Jerrum, *Conductance and the mixing property of markov chains*, the approximation of the permenant resolved, Proc. of the 20^{th} annual ACM Symposium on Theory of computing (1988), 235-244.
20. Daniel Spielman and Shang-hua Teng, *Spectral Partitioning Works: Planar graphs and finite element meshes*, Proc. of the 37th Annual Symposium on Foundations of Computer Science (FOCS '96), 96 - 105.
21. Deepak Verma and Marina Meila, *A comparison of spectral clustering algorithms*, TR UW-CSE-03-05-01, Department of Computer Science and Engineering, University of Washington (2005).
22. Van Vu, *Spectral norm of random matrices*, Proc. of the 36^{th} annual ACM Symposium on Theory of computing (2005), 619-626.
23. Ying Zhao and George Karypis, *Evaluation of hierarchical clustering algorithms for document datasets*, Proc. of the 11 International Conference on Information and Knowledge Management (2002), 515 - 524.