On Clusterings: Good, Bad and Spectral

Ravi Kannan*
Department of Computer Science,
Yale University, New Haven.
Email: kannan@cs.yale.edu

Santosh Vempala† and Adrian Vetta‡
Department of Mathematics,
M.I.T., Cambridge.
Email: {vempala, avetta}@math.mit.edu

Abstract. We motivate and develop a natural bicriteria measure for assessing the quality of a clustering which avoids the drawbacks of existing measures. A simple recursive heuristic is shown to have poly-logarithmic worst-case guarantees under the new measure. The main result of the paper is the analysis of a popular *spectral* algorithm. One variant of spectral clustering turns out to have effective worst-case guarantees; another finds a "good" clustering, if one exists.

*Supported in part by NSF grant CCR-9820850.
†Supported by NSF CAREER award CCR-9875024.
‡Supported in part by NSF CAREER award CCR-9875024 and in part by a Wolfe fellowship.

1 Introduction

Clustering, or partitioning into dissimilar groups of similar items, is a problem with many variants in mathematics and the applied sciences. The availability of vast amounts of data has revitalized research on the problem. Over the years, several clever heuristics have been invented for clustering. While many of these heuristics are problem-specific, the method known as *spectral clustering* has been applied successfully in a variety of different situations. Roughly speaking, spectral clustering is the technique of partitioning the rows of a matrix according to their components in the top few singular vectors of the matrix (see section 1.1 for a detailed description). The main motivation of this paper was to analyze the performance of spectral clustering. Such an evaluation, however, is inextricably linked to the question of how to measure the quality of a clustering. The justification provided by practitioners is typically case-by-case and experimental ("it works well on my data"). Theoreticians, meanwhile, have been busy studying quality measures that are seductively simple to define (e.g. k-median, minimum sum, minimum diameter, etc.). The measures thus far analyzed by theoreticians are easy to fool, i.e. there are simple examples where the "right" clustering is obvious but optimizing these measures produces undesirable solutions (see section 2). Thus, neither approach has been entirely satisfactory.

In this paper we propose a new bicriteria measure of the quality of a clustering, based on expansion-like properties of the underlying pairwise similarity graph. The quality of a clustering is given by two parameters: α, the minimum *conductance*[1] of the clusters and ϵ, the ratio of the weight of inter-cluster edges to the total weight of all edges. The objective is to find an (α, ϵ)-clustering that maximizes α and minimizes ϵ. Note that the conductance provides a measure of the quality of an individual cluster (and thus of the overall clustering) whilst the weight of the inter-cluster edges provides a measure of the cost of the clustering. Hence, imposing a lower bound, α, on the quality of each individual cluster we strive to minimize the cost, ϵ, of the clustering; or conversely, imposing an upper bound on the cost of the clustering we strive to maximize its quality. In section 2, we motivate the use of this more complex, bicriteria measure by showing that it does not have the obvious drawbacks of the simpler quality measures.

While the new measure might be qualitatively attractive, it would be of little use if optimizing it were computationally intractable. In section 3 we study a recursive heuristic designed to optimize the new measure. Although finding an exact solution is NP-hard, the algorithm is shown to have simultaneous poly-logarithmic approximations guarantees for the two parameters in the bicriteria measure (corollary 2).

In section 4 we turn to spectral algorithms for clustering. These algorithms are popular in part because of their speed (see section 5) and applicability in a variety of contexts [1, 7, 16, 20, 21]. However, while performing quite well in practice, they had hitherto eluded a rigorous worst-case analysis. This could be attributed to the fact that existing measures of quality have serious drawbacks, and did not capture the quality of the algorithm; even an exact algorithm for optimizing these measures might do poorly in many practical settings. We show that a simple recursive variant of spectral clustering has effective worst-case approximation guarantees with respect to the bicriteria measure (corollary 4). It is worth noting that both our worst-case guarantees follow from the same general theorem (see theorem 1 in section 3).

[1] Conductance will be defined precisely in Section 2; it measures how well-knit a graph is.

Another variant of spectral clustering has the following guarantee: if the input data has a rather good clustering (i.e., α is large and ϵ is small), then the spectral algorithm will find a clustering that is "close" to the optimal clustering (theorem 5).

1.1 Spectral Clustering Algorithms

Spectral clustering refers to the general technique of partitioning the rows of a matrix according to their components in the top few singular vectors of the matrix. The underlying problem, that of clustering the rows of a matrix, is ubiquitous. We mention three special cases that are all of independent interest:

- The matrix encodes the pairwise similarities of vertices of a graph.
- The rows of the matrix are points in a d-dimensional Euclidean space. The columns are the coordinates.
- The rows of the matrix are documents of a corpus. The columns are terms. The (i, j) entry encodes information about the occurrence of the jth term in the ith document.

Given a matrix A, the spectral algorithm for clustering the rows of A is given below.

```
Spectral Algorithm I
Find the top k right singular vectors v_1, v_2, ..., v_k.
Let C be the matrix whose jth column is given by Av_j.
Place row i in cluster j if C_ij is the largest entry in the ith row of C.
```

The algorithm has the following interpretation[2]. Suppose the rows of A are points in a high-dimensional space. Then the subspace defined by the top k right singular vectors of A is the rank-k subspace that best approximates A. The spectral algorithm projects all the points onto this subspace. Each singular vector then defines a cluster; to obtain a clustering we map each projected point to the (cluster defined by the) singular vector that is closest to it in angle.

In section 4, we study a recursive variant of this algorithm.

2 What is a Good Clustering?

How good is the spectral algorithm? Intuitively, a clustering algorithm performs well if points that are similar are assigned the same cluster and points that are dissimilar are assigned to different clusters. Of course, this may not be possible to do for every pair of points, and so we compare the clustering found by the algorithm to the *optimal* one for the given matrix. This, though, leads to another question: what exactly is an optimal clustering? To provide a quantitative answer, we first need to define a measure of the quality of a clustering. In recent years several combinatorial measures of clustering quality have been investigated in detail. These include *minimum diameter, k-center, k-median,* and *minimum sum* (for example: [4], [5], [9], [12], [13], etc.).

[2] Computationally, it is useful to note that the jth column of C is also given by $\lambda_j u_j$; here λ_j is the jth singular value of A and u_j is the jth left singular vector.

All these measures, although mathematically attractive due to their simplicity, are easy to fool. That is, one can construct examples with the property that the "best" clustering is obvious and yet an algorithm that optimizes one of these measures finds a clustering that is substantially different (and therefore unsatisfactory). Such examples are presented in Figures 1 and 2, where the goal is to partition the points into two clusters. Observe that all the measures given above seek to minimize some objective function. In the figures, nearby points (which represent highly similar points) induce low cost edges; points that are farther apart (and represent dissimilar points) induce high cost edges.

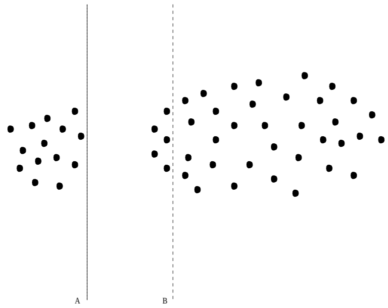

Figure 1: Optimizing the diameter produces B while A is clearly more desirable.

Consider a clustering that minimizes the maximum diameter of the clusters; the diameter of a cluster being the largest distance, say, between two points in a cluster. It is NP-hard to find such a clustering, but this is not our main concern. What is worrisome about the example shown in Figure 1 is that the optimal solution (B) produces a cluster which contains points that should have been separated. Clustering with respect to the minimum sum and k-center measures will produce the same result. The reason such a poor cluster is produced is that although we have minimized the maximum dissimilarity between points in a cluster, this was at the expense of creating a cluster with many dissimilar points. The clustering (A) on the other hand, although it leads to a larger maximum diameter, say, is desirable since it better satisfies the goal of "similar points together and dissimilar points apart". This problem also arises for the k-median measure (see, for example, the case shown in Figure 2); it may produce clusters of poor quality.

We will find it more convenient to model the input as a similarity graph rather than as a distance graph. This is indeed often the case in practice. Thus the input is an edge-weighted complete graph whose vertices need to be partitioned. The weight of an edge a_{ij} represents the similarity of the vertices (points) i and j. Thus, the graph for points in space would have high edge weights for points that are close together and low edge weights for points that are far apart. So the graph is associated with an $n \times n$ symmetric matrix A with entries a_{ij}; here we assume that the a_{ij} are non-negative.

Let us now return to the question of what a good clustering is. The quality of a cluster should be determined by how similar the points within a cluster are. Note that each cluster is represented by a subgraph. In particular, if there is a cut of small weight that divides the cluster into two pieces of comparable size then the cluster has lots of pairs of vertices that are dissimilar and hence it is of low quality. This might suggest that the quality of a subgraph as a cluster is the minimum cut of the subgraph. However, this is misleading as is illustrated by

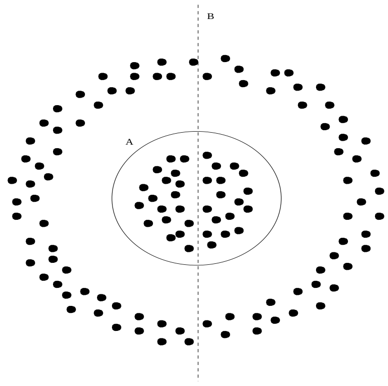

Figure 2: The inferior clustering B is found by optimizing the 2-median measure.

Figure 3. In this example edges represent high-similarity pairs and non-edges represent pairs that are highly dissimilar. The minimum cut of the first subgraph is larger than that of the second subgraph. This is because the second subgraph has low degree vertices. However, the second subgraph is a higher quality cluster. This can be attributed to the fact that in the first subgraph there is a cut whose weight is small *relative to the sizes of the pieces it creates.* A quantity that measures the relative cut size is the *expansion*. The expansion of a graph is the minimum ratio over all cuts of the graph of the total weight of edges of the cut to the number of vertices in the smaller part created by the cut. Formally, we denote the expansion of a cut $(S, \bar{S})$ by:

$$\psi(S) = \frac{\sum_{i \in S, j \notin S} a_{ij}}{\min(|S|, |\bar{S}|)}$$

We say that the expansion of a graph is the minimum expansion over all the cuts of the graph. Our first measure of quality of a cluster is the expansion of the subgraph corresponding to it. The expansion of a clustering is the minimum expansion of one of the clusters.

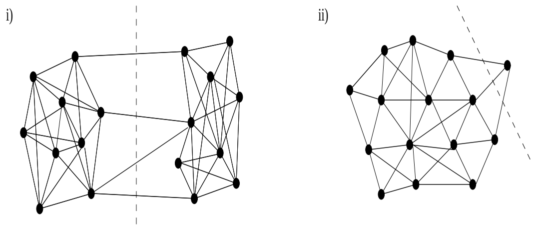

Figure 3: The second subgraph is of higher quality as a cluster even though it has a smaller minimum cut.

The measure defined above gives equal importance to all the vertices of the given graph. This, though, may lead to a rather taxing requirement. For example, in order to accommodate a vertex i with very little similarity to all the other vertices combined (i.e., $\sum_j a_{ij}$ is small), then α will have to be very low. Arguably, it is more prudent to give greater importance to vertices that have many similar neighbors and lesser importance to vertices that have few similar neighbors. This can be done by a direct generalization of the expansion, called the *conductance*, in which subsets of vertices are weighted to reflect their importance.

The conductance of a cut $(S, \bar{S})$ in G is denoted by:

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} a_{ij}}{\min(a(S), a(\bar{S}))}$$

Here $a(S) = a(S,V) = \sum_{i \in S} \sum_{j \in V} a_{ij}$. The conductance of a graph is the minimum conductance over all the cuts of the graph; $\phi(G) = \min_{S \subseteq V} \phi(S)$. In order to quantify the quality of a clustering we generalize the definition of conductance further. Take a cluster $C \subseteq V$ and a cut $(S, C \backslash S)$ within C, where $S \subseteq C$. Then we say that the *conductance* of S in C is:

$$\phi(S, C) = \frac{\sum_{i \in S, j \in C \backslash S} a_{ij}}{\min(a(S), a(C \setminus S))}$$

The conductance $\phi(C)$ of a cluster C will then be the smallest conductance of a cut within the cluster. The conductance of a clustering is the minimum conductance of its clusters. This conductance measure seems extremely well suited to achieve our intuitive goal i.e. clustering similar points and separating dissimilar points. We then obtain the following optimization problem: given a graph and an integer k, find a k-clustering with the maximum conductance. Notice that optimizing the expansion/conductance gives the right clustering in the examples of Figures 1 and 2. To see this assume, for example, that the points induce an unweighted graph (i.e. zero-one edge weights). Thus, a pair of vertices induces an edge if and only if the two vertices are close together. Clustering (A) will then be obtained in each example.

There is still a problem with the above clustering measure. The graph might consist mostly of clusters of high quality and maybe a few points that create clusters of very poor quality, so that any clustering necessarily has a poor overall quality (since we have defined the quality of a clustering to be the minimum over all the clusters). In fact, to boost the overall quality, the best clustering might create many clusters of relatively low quality so that the minimum is as large as possible. Such an example is shown in Figure 4.

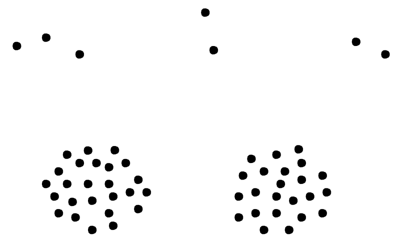

Figure 4: Assigning the outliers leads to poor quality clusters.

One way to handle the problem might be to avoid restricting the number of clusters. But this could lead to a situation where many points are in singleton (or extremely small) clusters. Instead we measure the quality of a clustering using two criteria, the first is the minimum quality of the clusters (called α), and the second is the fraction of the total weight of edges that are not covered by the clusters (called ϵ).

Definition 1. *We call a partition $\{C_1, C_2, \ldots, C_l\}$ of V an (α, ϵ)-clustering if :*
1. The conductance of each C_i is at least α.
2. The total weight of inter-cluster edges is at most an ϵ fraction of the total edge weight.

Thus we obtain a bicriteria measure of the quality of a clustering. Associated with this bicriteria measure is the following optimization problem (note that the number of clusters is not restricted).

Problem 1. *Given α, find an (α, ϵ)-clustering that minimizes ϵ (alternatively, given ϵ, find an (α, ϵ)-clustering that maximizes α).*

It is not hard to see that optimizing this measure of cluster quality, does well on the earlier "bad" examples. While it is impossible for any measure to be universally the *right* measure, an important question is to find the class of applications for which the proposed measure is suitable. Empirical results suggest that the bicriteria measure seems natural for a variety of applications. The focus of the rest of this paper, however, is to consider the measure from a theoretical standpoint and to examine in detail the performance of spectral clustering algorithms.

It may be noted that there is a monotonic function f that represents the optimal (α, ϵ) pairings. For example, for each α there is a minimum value of ϵ, equal to $f(\alpha)$, such that an (α, ϵ)-clustering exists. In the following sections we present two approximation algorithms for the clustering property. One nice characteristic of these algorithms is that in a single application they can be used to obtain an approximation f' for the entire function f, not just for f evaluated at a single point. Thus the user need not specify a desired value of α or ϵ a priori. Rather, the desired conductance/cost trade-off may be determined after consideration of the approximation function f'.

3 Approximation Algorithms

Problem 1 is NP-hard. To see this, consider maximizing α whilst setting ϵ to zero. This problem is equivalent to finding the conductance of a given graph which is well known to be NP-hard [11]. Here we present a simple heuristic and provide worst-case approximation guarantees for it.

```
Approximate-Cluster Algorithm
Find a cut that approximates the minimum conductance cut in G.
Recurse on the pieces induced by the cut.
```

The idea behind our algorithm is simple. Given G, find a cut $(S, \bar{S})$ of minimum conductance. Then recurse on the subgraphs induced by S and $\bar{S}$. Finding a cut of minimum conductance is hard, and hence we need to use an approximately minimum cut. There are two well-known approximations for the minimum conductance cut, one is based on a linear programming relaxation and the other is derived from the second eigenvector of the graph. Before we discuss these approximations, we prove a general theorem for general approximation heuristics.

Let $\mathcal{A}$ be an approximation algorithm that produces a cut of conductance at most Kx^{ν} if the minimum conductance is x, where K is independent of x (K could be a function of n, for example) and ν is a fixed constant between between 0 and 1. The following theorem (which is the main theorem of the paper) provides a guarantee for the approximate-cluster algorithm using $\mathcal{A}$ as a subroutine.

Theorem 1. *If G has an (α, ϵ)-clustering, then the approximate-cluster algorithm will find a clustering of quality*

$$\left(\left(\frac{\alpha}{6K \log \frac{n}{\epsilon}}\right)^{1/\nu}, (12K+2)\epsilon^{\nu} \log \frac{n}{\epsilon}\right).$$

Proof. Let the cuts produced by the algorithm be $(S_1, T_1), (S_2, T_2), \ldots$, where we adopt the convention that S_j is the "smaller" side (i.e., $a(S_j) \leq a(T_j)$). Let $C_1, C_2, \ldots C_l$ be an (α, ϵ)-clustering. We use the termination condition of $\alpha^* = \frac{\alpha}{6 \log \frac{n}{\epsilon}}$. We will assume that we apply the recursive step in the algorithm only if the conductance of a given piece as detected by the heuristic for the minimum conductance cut is less than α^*. In addition, purely for the sake of analysis we consider a slightly modified algorithm. If at any point we have a cluster C_t with the property that $a(C_t) < \frac{\epsilon}{n} a(V)$ then we split C_t into singletons. The conductance of singletons is defined to be 1. Then, upon termination, each cluster has conductance at least

$$\left(\frac{\alpha^*}{K}\right)^{1/\nu} = \left(\frac{\alpha}{6K \log \frac{n}{\epsilon}}\right)^{1/\nu}$$

Thus it remains to bound the weight of the inter-cluster edges. Observe that $a(V)$ is twice the total edge weight in the graph, and so $W = \frac{\epsilon}{2} a(V)$ is the weight of the inter-cluster edges in this optimal solution.

Now we divide the cuts into two groups. The first group, H, consists of cuts with "high" conductance within clusters. The second group consists of the remaining cuts. We will use the notation $w(S_j, T_j) = \sum_{u \in S_j, v \in T_j} a_{uv}$. In addition, we denote by $w_{\mathrm{I}}(S_j, T_j)$ the sum of the weights of the intra-cluster edges of the cut (S_j, T_j), i.e., $w_{\mathrm{I}}(S_j, T_j) = \sum_{i=1}^{l} w(S_j \cap C_i, T_j \cap C_i)$. We then set

$$H = \left\{ j : w_{\mathrm{I}}(S_j, T_j) \geq 2\alpha^* \sum_{i=1}^{l} \min(a(S_j \cap C_i), a(T_j \cap C_i)) \right\}$$

We now bound the cost of the high conductance group. For all $j \in H$, we have,

$$\alpha^* a(S_j) \quad \geq \quad w(S_j, T_j) \geq w_{\mathrm{I}}(S_j, T_j) \geq 2\alpha^* \sum_i \min(a(S_j \cap C_i), a(T_j \cap C_i))$$

Consequently we observe that

$$\sum_i \min(a(S_j \cap C_i), a(T_j \cap C_i)) \leq \frac{1}{2} a(S_j)$$

From the algorithm's cuts, $\{(S_j, T_j)\}$, and the optimal clustering, $\{C_i\}$, we define a new clustering via a set of cuts $\{(S'_j, T'_j)\}$ as follows. For each $j \in H$, we define a cluster-avoiding cut (S'_j, T'_j) in $S_j \cup T_j$ in the following manner. For each $i, 1 \leq i \leq l$, if $a(S_j \cap C_i) \geq a(T_j \cap C_i)$, then place all of $(S_j \cup T_j) \cap C_i$ into S'_j. If $a(S_j \cap C_i) < a(T_j \cap C_i)$, then place all of $(S_j \cup T_j) \cap C_i$

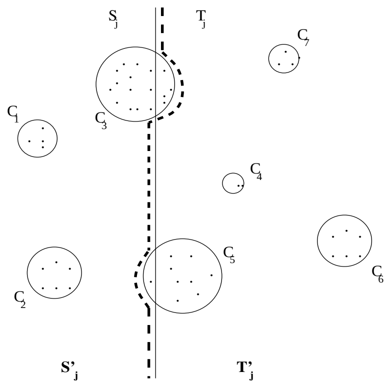

Figure 5: The proof of Theorem 1.

into T_j'. An example is given in Figure 5, where the original cut is shown by the solid line and the cluster-avoiding cut by the dashed line. Notice that, since $|a(S_j) - a(S_j')| \le \frac{1}{2}a(S_j)$, we have that $\min(a(S_j'), a(T_j')) \ge \frac{1}{2}a(S_j)$. Now we will use the approximation guarantee for the cut procedure to get an upper bound on $w(S_j, T_j)$ in terms of $w(S_j', T_j')$.

$$\begin{aligned} \frac{w(S_j, T_j)}{a(S_j)} &\le K\left(\frac{w(S_j', T_j')}{\min\{a(S_j'), a(T_j')\}}\right)^{\nu} \\ &\le K\left(\frac{2w(S_j', T_j')}{a(S_j)}\right)^{\nu} \end{aligned}$$

Hence we have bounded the overall cost of the high conductance cuts with respect to the cost of the cluster-avoiding cuts. We now bound the cost of these cluster-avoiding cuts. Let $P(S)$ denote the set of inter-cluster edges incident at a vertex in S, for any subset S of V. Also, for a set of edges F, let $w(F)$ denote the sum of their weights. Then, $w(S_j', T_j') \le w(P(S_j'))$, since every edge in (S_j', T_j') is an inter-cluster edge. So we have,

$$w(S_j, T_j) \le K\big(2w(P(S_j'))\big)^{\nu} a(S_j)^{1-\nu} \tag{1}$$

Next we prove the following claim.

Claim 1. For each vertex $u \in V$, there are at most $\log \frac{n}{\epsilon}$ values of j such that u belongs to S_j. Further, there are at most $2\log\frac{n}{\epsilon}$ values of j such that u belongs to S_j'.

To prove the claim, fix a vertex $u \in V$. Let

$$I_u = \{j : u \in S_j\} \qquad\qquad J_u = \{j : u \in S_j' \setminus S_j\}$$

Clearly if $u \in S_j \cap S_k$ (with $k > j$), then (S_k, T_k) must be a partition of S_j or a subset of S_j. Now we have, $a(S_k) \le \frac{1}{2}a(S_k \cup T_k) \le \frac{1}{2}a(S_j)$. So $a(S_j)$ reduces by a factor of 2 or greater between two successive times u belongs to S_j. The maximum value of $a(S_j)$ is at most $a(V)$ and the minimum value is at least $\frac{\epsilon}{n}a(V)$, so the first statement of the claim follows.

Now suppose $j, k \in J_u; j < k$. Suppose also $u \in C_i$. Then $u \in T_j \cap C_i$. Also, later, T_j (or a subset of T_j) is partitioned into (S_k, T_k) and, since $u \in S_k' \setminus S_k$, we have $a(T_k \cap C_i) \le a(S_k \cap C_i)$.

Thus $a(T_k \cap C_i) \leq \frac{1}{2}a(S_k \cup T_k) \leq \frac{1}{2}a(T_j \cap C_i)$. Thus $a(T_j \cap C_i)$ halves between two successive times that $j \in J_u$. So, $|J_u| \leq \log \frac{n}{\epsilon}$. This proves the second statement in the claim (since $u \in S'_j$ implies that $u \in S_j$ or $u \in S'_j \setminus S_j$). These concepts are shown pictorially in Figure 6, where the cuts (S_j, T_j) and (S_k, T_k) are represented by solid lines and the cuts (S'_j, T'_j) and (S'_k, T'_k) by dashed lines.

Using this claim, we can bound the overall cost of the group of cuts with high conductance within clusters with respect to the cost of the optimal clustering as follows:

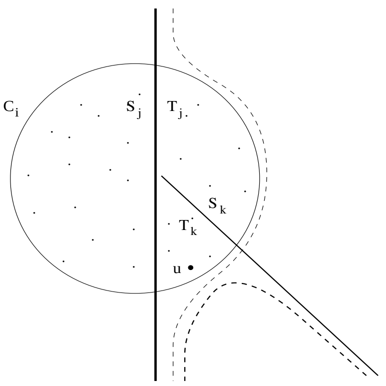

Figure 6: Proof of Claim 1.

$$\begin{aligned}
\sum_{j \in H} w(S_j, T_j) &\leq \sum_{\text{all } j} K\big(2w(P(S'_j))\big)^{\nu} a(S_j)^{1-\nu} \\
&\leq K\Big(2\sum_{\text{all } j} w(P(S'_j))\Big)^{\nu} \Big(\sum_j a(S_j)\Big)^{1-\nu} \\
&\leq K\left(2\epsilon \log \frac{n}{\epsilon}\, a(V)\right)^{\nu} \left(2 \log \frac{n}{\epsilon}\, a(V)\right)^{1-\nu} \\
&\leq 2K\epsilon^{\nu} \log \frac{n}{\epsilon}\, a(V) \qquad (2)
\end{aligned}$$

Here we used Hölder's inequality.

Next we deal with the group of cuts with low conductance within clusters i.e., those j not in H. First, suppose that all the cuts together induce a partition of C_i into $P^i_1, P^i_2, \ldots P^i_{r_i}$. Every edge between two vertices in C_i which belong to different sets of the partition must be cut by some cut (S_j, T_j) and, conversely, every edge of every cut $(S_j \cap C_i, T_j \cap C_i)$ must have its two end points in different sets of the partition. So, given that C_i has conductance α, we obtain

$$\sum_{\text{all } j} w_{\text{I}}(S_j \cap C_i, T_j \cap C_i) = \frac{1}{2}\sum_{s=1}^{r_i} w(P^i_s, C_i \setminus P^i_s) \geq \frac{1}{2}\alpha \sum_s \min(a(P^i_s), a(C_i \setminus P^i_s))$$

For each vertex $u \in C_i$ there can be at most $\log \frac{n}{\epsilon}$ values of j such that u belongs to the smaller (according to $a(\cdot)$) of the two sets $S_j \cap C_i$ and $T_j \cap C_i$. So, we have that

$$\sum_{s=1}^{r_i} \min(a(P_s^i), a(C_i \setminus P_s^i)) \quad \geq \quad \frac{1}{\log \frac{n}{\epsilon}} \sum_j \min(a(S_j \cap C_i), a(T_j \cap C_i))$$

Thus,

$$\sum_{\text{all } j} w_{\rm I}(S_j, T_j) \geq \frac{\alpha}{2 \log \frac{n}{\epsilon}} \sum_{i=1}^{l} \sum_j \min(a(S_j \cap C_i), a(T_j \cap C_i))$$

Therefore, from the definition of H, we have

$$\sum_{j \notin H} w_{\rm I}(S_j, T_j) \leq 2\alpha^* \sum_{\text{all } j} \sum_{i=1}^{l} \min(a(S_j \cap C_i), a(T_j \cap C_i)) \leq \frac{2}{3} \sum_{\text{all } j} w_{\rm I}(S_j, T_j)$$

Thus, we are able to bound the intra-cluster cost of the low conductance group of cuts in terms of the intra-cluster cost of the high conductance group. Applying (2) then gives

$$\sum_{j \notin H} w_{\rm I}(S_j, T_j) \leq 2 \sum_{j \in H} w_{\rm I}(S_j, T_j) \leq 4K\epsilon^{\nu} \log \frac{n}{\epsilon} a(V) \tag{3}$$

In addition, since each inter-cluster edge belongs to at most one cut S_j, T_j, we have that

$$\sum_{j \notin H} (w(S_j, T_j) - w_{\rm I}(S_j, T_j)) \quad \leq \quad \frac{\epsilon}{2} a(V) \tag{4}$$

We then sum up (2), (3) and (4). To get the total cost we note that splitting up all the V_t with $a(V_t) \leq \frac{\epsilon}{n} a(V)$ into singletons costs us at most $\frac{\epsilon}{2} a(V)$ on the whole. Substituting $a(V)$ as twice the total sum of edge weights gives the bound on the cost of inter-cluster edge weights. This completes the proof of Theorem 1. $\square$

The Leighton-Rao algorithm for approximating the conductance finds a cut of conductance at most $2 \log n$ times the minimum [14]. In our terminology, it is an approximation algorithm with $K = 2 \log n$ and $\nu = 1$. Applying theorem 1 leads to the following guarantee.

Corollary 2. *If the input has an (α, ϵ)-clustering, then, using the Leighton-Rao heuristic, the approximate-cluster algorithm finds an*

$$\left(\frac{\alpha}{12 \log n \log \frac{n}{\epsilon}}, 26\epsilon \log n \log \frac{n}{\epsilon} \right) \text{-}\mathit{clustering}. \quad \square$$

We now assess the running time of the algorithm using this heuristic. The fastest implementation for this heuristic, due to Benczur and Karger [3], runs in $\tilde{O}(n^2)$ time (where the $\tilde{O}$ notation suppresses factors of $\log n$). Since the algorithm makes less than n cuts, the total running time is $\tilde{O}(n^3)$. This might be slow for some real-world applications. We discuss a potentially more practical algorithm in the next section.

4 Performance Guarantees for Spectral Clustering

In this section, we describe and analyse a recursive variant of the spectral algorithm. This algorithm, outlined below, has been used in the field of computer vision [16] and also in the field of web search engines [21]. Note that the algorithm is a special case of the approximate-cluster algorithm described in the previous section; here we use a spectral heuristic to approximate the minimum conductance cut.

```
Spectral Algorithm II
Normalize A and find its 2nd right eigenvector v.
Find the best ratio cut wrt v.
Recurse on the pieces induced by the cut.
```

Thus, we find a clustering by repeatedly solving a one-dimensional clustering problem. Since the latter is easy to solve, the algorithm is efficient. The fact that it also has worst-case quality guarantees is less obvious.

We now elaborate upon the basic description of this variant of the spectral algorithm. Initially, we normalize our matrix A by scaling the rows so that the row sums are all equal to one. At any later stage in the algorithm we have a partition $\{C_1, C_2, \ldots, C_s\}$. For each C_t, we consider the $|C_t| \times |C_t|$ submatrix B of A restricted to C_t. We normalize B by setting b_{ii} to $1 - \sum_{j \in C_t, j \neq i} b_{ij}$. As a result, B is also non-negative with row sums equal to one.

Observe that upon normalization of the matrix, our conductance measure corresponds to the familiar Markov Chain conductance measure i.e.

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} a_{ij}}{\min(a(S), a(\bar{S}))} = \frac{\sum_{i \in S, j \notin S} \pi_i b_{ij}}{\min(\pi(S), \pi(\bar{S}))}$$

where π is the stationary distribution of the Markov Chain.

We then find the second eigenvector of B. This is the right eigenvector v corresponding to the second largest eigenvalue λ_2, i.e. $Bv = \lambda_2 v$. Then order the elements (rows) of C_t decreasingly with respect to their component in the direction of v. Given this ordering, say $\{u_1, u_2, \ldots u_r\}$, find the minimum *ratio cut* in C_t. This is the cut that minimises $\phi(\{u_1, u_2, \ldots u_j\}, C_t)$ for some j, $1 \leq j \leq r - 1$. We then recurse on the pieces $\{u_1, \ldots, u_j\}$ and $C_t \setminus \{u_1, \ldots, u_j\}$.

4.1 Worst-case guarantees

We will use the following theorem to prove a worst-case guarantee for the algorithm. This result was essentially proved by Sinclair and Jerrum (in their proof of Lemma 3.3 in [17], although not mentioned in the statement of the lemma). For completeness, and due to the fact that Theorem 3 is usually not explicitly stated in the Markov Chain literature (or usually includes some other conditions which are not relevant here), we include a proof of this result. Observe that, via the use of the second eigenvalue, the theorem bounds the conductance of the cut found by the heuristic with respect to that of the optimal cut.

Theorem 3. *Suppose B is a $N \times N$ matrix with non-negative entries with each row sum equal to 1 and suppose there are positive real numbers $\pi_1, \pi_2, \ldots \pi_N$ summing to 1 such that*

$\pi_i b_{ij} = \pi_j b_{ji}$ for all i, j. If v is the right eigenvector of B corresponding to the second largest eigenvalue λ_2, and $i_1, i_2, \ldots i_N$ is an ordering of $1, 2, \ldots N$ so that $v_{i_1} \geq v_{i_2} \ldots \geq v_{i_N}$, then

$$\min_{S \subseteq \{1,2,\ldots N\}} \frac{\sum_{i \in S, j \notin S} \pi_i b_{ij}}{\min(\sum_{i \in S} \pi_i, \sum_{j \notin S} \pi_j)} \geq 1 - \lambda_2 \geq \frac{1}{2} \left(\min_{l, 1 \leq l \leq N} \frac{\sum_{1 \leq u \leq l; l+1 \leq v \leq N} \pi_{i_u} b_{i_u i_v}}{\min(\sum_{1 \leq u \leq l} \pi_{i_u}, \sum_{l+1 \leq v \leq N} \pi_{i_v})} \right)^2$$

Before proving this theorem, let us use it along with Theorem 1 to get a worst-case guarantee for spectral algorithm II. In our terminology, the above theorem says that the spectral heuristic for minimum conductance is an approximation algorithm with $K = \sqrt{2}$ and $\nu = 1/2$.

Corollary 4. *If the input has an (α, ϵ)-clustering, then, using the spectral heuristic, the approximate-cluster algorithm finds an*

$$\left(\frac{\alpha^2}{72 \log^2 \frac{n}{\epsilon}}, 20\sqrt{\epsilon} \log \frac{n}{\epsilon} \right) \text{-clustering.} \quad \square$$

Proof (of Theorem 3). We first evaluate the second eigenvalue. Towards this end, let $D^2 = \text{diag}(\pi)$. Then, from the time-reversibility property of B, we have $D^2 B = B^T D^2$. Hence $Q = DBD^{-1}$ is symmetric. The eigenvalues of B and Q are the same, with their largest eigenvalue equal to 1. In addition, $\pi^T D^{-1} Q = \pi^T D^{-1}$ and therefore $\pi^T D^{-1}$ is the left eigenvector of Q corresponding to the eigenvalue 1. So we have,

$$\lambda_2 = \max_{\pi^T D^{-1} x = 0} \frac{x^T DBD^{-1} x}{x^T x}$$

Thus, substituting $y = D^{-1}x$, we obtain

$$1 - \lambda_2 = \min_{\pi^T D^{-1} x = 0} \frac{x^T D(I - B) D^{-1} x}{x^T x} = \min_{\pi^T y = 0} \frac{y^T D^2 (I - B) y}{y^T D^2 y}$$

The numerator can be rewritten:

$$\begin{aligned} y^T D^2 (I - B) y &= -\sum_{i \neq j} y_i y_j \pi_i b_{ij} + \sum_i \pi_i (1 - b_{ii}) y_i^2 \\ &= -\sum_{i \neq j} y_i y_j \pi_i b_{ij} + \sum_{i \neq j} \pi_i b_{ij} \frac{y_i^2 + y_j^2}{2} \\ &= \sum_{i < j} \pi_i b_{ij} (y_i - y_j)^2 \end{aligned}$$

Denote this final term by $\mathcal{E}(y, y)$. Then

$$1 - \lambda_2 = \min_{\pi^T y = 0} \frac{\mathcal{E}(y, y)}{\sum_i \pi_i y_i^2}$$

To prove the first inequality of the theorem, let $(S, \bar{S})$ be the cut with the minimum conductance. Define a vector w as follows

$$w_i = \begin{cases} \sqrt{\frac{1}{\sum_u a(u)} \frac{\pi(\bar{S})}{\pi(S)}} & \text{if } i \in S \\ -\sqrt{\frac{1}{\sum_u a(u)} \frac{\pi(S)}{\pi(\bar{S})}} & \text{if } i \in \bar{S} \end{cases}$$

It is then easy to check that $\sum_i \pi_i w_i = 0$ and that

$$\phi(S) \geq \frac{\mathcal{E}(w,w)}{\sum_i \pi_i w_i^2} \geq 1 - \lambda_2$$

Hence we obtain the desired lower bound on the conductance.

We will now prove the second inequality. Suppose that the minimum above is attained when y is equal to v. Then Dv is the eigenvector of Q corresponding to the eigenvalue λ_2 and, v is the right eigenvector of B corresponding to λ_2. Our ordering is then with respect to v in accordance with the statement of the theorem. Assume that, for simplicity of notation, the indices are reordered (i.e. the rows and corresponding columns of B and D are reordered) so that $v_1 \geq v_2 \geq \cdots \geq v_N$. Now define r to satisfy $\pi_1 + \pi_2 + \cdots + \pi_{r-1} \leq \frac{1}{2} < \pi_1 + \pi_2 + \cdots + \pi_r$, and let $z_i = v_i - v_r$ for $i = 1, \ldots, n$. Then $z_1 \geq z_2 \geq \cdots \geq z_r = 0 \geq z_{r+1} \geq \cdots \geq z_n$, and

$$\begin{aligned}
\frac{\mathcal{E}(v,v)}{\sum_i \pi_i v_i^2} = \frac{\mathcal{E}(z,z)}{-v_r^2 + \sum_i \pi_i z_i^2} &\geq \frac{\mathcal{E}(z,z)}{\sum_i \pi_i z_i^2} \\
&= \frac{\left(\sum\limits_{i<j} \pi_i b_{ij} (z_i - z_j)^2\right) \left(\sum\limits_{i<j} \pi_i b_{ij} (|z_i| + |z_j|)^2\right)}{\left(\sum\limits_{i} \pi_i z_i^2\right) \left(\sum\limits_{i<j} \pi_i b_{ij} (|z_i| + |z_j|)^2\right)}
\end{aligned}$$

Consider the numerator of this final term. By Cauchy-Schwartz

$$\begin{aligned}
\left(\sum_{i<j} \pi_i b_{ij} (z_i - z_j)^2\right) \left(\sum_{i<j} \pi_i b_{ij} (|z_i| + |z_j|)^2\right) &\geq \left(\sum_{i<j} \pi_i b_{ij} |z_i - z_j| (|z_i| + |z_j|)\right)^2 \\
&\geq \left(\sum_{i<j} \pi_i b_{ij} \sum_{k=i}^{j-1} |z_{k+1}^2 - z_k^2|\right)^2 \qquad (5)
\end{aligned}$$

Here the second inequality follows from the fact that if $i < j$ then $|z_i - z_j|(|z_i| + |z_j|) \geq \sum_{k=i}^{j-1} |z_{k+1}^2 - z_k^2|$. This follows from observations that

i) If z_i and z_j have the same sign (i.e. $r \notin \{i, i+1, \ldots, j\}$) then $|z_i - z_j|(|z_i| + |z_j|) = |z_i^2 - z_j^2|$.

ii) Otherwise, if z_i and z_j have different signs then $|z_i - z_j|(|z_i| + |z_j|) = (|z_i| + |z_j|)^2 > z_i^2 + z_j^2$.

Also,

$$\sum_{i<j} \pi_i b_{ij} (|z_i| + |z_j|)^2 \leq 2 \sum_{i<j} \pi_i b_{ij} (z_i^2 + z_j^2) \leq 2 \sum_i \pi_i z_i^2$$

As a result we have,

$$\begin{aligned}
\frac{\mathcal{E}(v,v)}{\sum_i \pi_i v_i^2} &\geq \frac{\left(\sum\limits_{i<j} \pi_i b_{ij} (z_i - z_j)^2\right) \left(\sum\limits_{i<j} \pi_i b_{ij} (|z_i| + |z_j|)^2\right)}{\left(\sum\limits_{i} \pi_i z_i^2\right) \left(\sum\limits_{i<j} \pi_i b_{ij} (|z_i| + |z_j|)^2\right)} \\
&\geq \frac{\left(\sum_{i<j} \pi_i b_{ij} \sum_{k=i}^{j-1} |z_{k+1}^2 - z_k^2|\right)^2}{2 \left(\sum_i \pi_i z_i^2\right)^2}
\end{aligned}$$

Set $S_k = \{1, 2, \ldots, k\}$, $C_k = \{(i,j) : i \le k < j\}$ and

$$\hat{\alpha} = \min_{k, 1 \le k \le N} \frac{\sum_{(i,j) \in C_k} \pi_i b_{ij}}{\min(\sum_{i: i \le k} \pi_i, \sum_{i: i > k} \pi_i)}$$

Since $z_r = 0$, we obtain

$$\begin{aligned}
\sum_{i<j} \pi_i b_{ij} \sum_{k=i}^{j-1} |z_{k+1}^2 - z_k^2| &= \sum_{k=1}^{N-1} |z_{k+1}^2 - z_k^2| \sum_{(i,j) \in C_k} \pi_i b_{ij} \\
&\ge \hat{\alpha} \left(\sum_{k=1}^{r-1} (z_k^2 - z_{k+1}^2) \pi(S_k) + \sum_{k=r}^{N-1} (z_{k+1}^2 - z_k^2)(1 - \pi(S_k)) \right) \\
&= \hat{\alpha} \left(\sum_{k=1}^{N-1} (z_k^2 - z_{k+1}^2) \pi(S_k) + (z_N^2 - z_r^2) \right) \\
&= \hat{\alpha} \sum_{k=1}^{N} \pi_k z_k^2
\end{aligned}$$

Consequently, if $\pi^T y = 0$ then

$$1 - \lambda_2 = \frac{\mathcal{E}(v,v)}{\sum_i \pi_i v_i^2} \ge \frac{\hat{\alpha}^2}{2}$$

□

4.2 In the presence of a good balanced clustering

In this section, we consider the situation in which a given input matrix A has a particularly good clustering. Here the matrix can be partitioned into blocks such that the conductance of each block as a cluster is high and the total weight of inter-cluster edges is small. We present a result which shows that, in such a circumstance, the spectral algorithm will find a clustering that is close to the optimal clustering; i.e., only a small number of rows will be placed in the incorrect cluster.

First we show how to model this situation. We will use the following terminology. Denote by $|x|$ the 2-norm (length) of a vector x. The 2-norm of an $n \times m$ matrix A is

$$\max_{x \in \mathbb{R}^m, |x|=1} \|Ax\|$$

We assume that A can be written as $B + E$, where B is a block diagonal matrix with row sums equal to 1. The blocks of B, say $B_1, B_2, \ldots, B_k$, induce the clusters of the optimal clustering, and E corresponds to the set of edges that run between clusters.

Rather than conductance, it will be easier to state the result in terms of the minimum *eigenvalue gap* of the blocks $B_1, B_2, \ldots, B_k$. The eigenvalue gap of a matrix is $\beta = 1 - \frac{\lambda_2}{\lambda_1}$. and is closely related to the conductance ($\frac{\phi^2}{2} \le \beta \le 2\phi$). Ideally, we would like to show that if $A = B + E$ where β is large for each block in B and the total weight of edges in E is small then the spectral algorithm works. While this might be expected for a typical input, it is possible to construct examples where applying spectral algorithm I does not work. So we consider the following slightly modified version of spectral algorithm I. As before, project the points onto the space spanned by the top k singular right vectors (eigenvectors). Let C be the $n \times k$ matrix obtained. For each row c_i of C, we define a set S_i as follows:

$$S_i = \left\{ j \, : \, \frac{c_i \cdot c_j}{|c_i| \, |c_j|} \geq \frac{3}{4} \right\}$$

Observe that these sets are clearly not disjoint. Now, to define the final clustering, pick the largest S_i, then delete all its elements from C. Repeat in this greedy manner on the remaining sets until all the elements are covered.

Theorem 5 shows that if the 2-norm of E and the $k+1$st eigenvalue of B are small then this spectral algorithm finds a clustering very close to the optimal one. For analyses in a similar spirit, see Papadimitriou et al. [15] and Fiat et al. [2]. The condition below requires a gap between the kth and $k+1$st eigenvalues of the input matrix. Intuitively, this also corresponds to the gap between the top two eigenvalues of any block, and thus captures the fact that each block has high conductance.

Theorem 5. *Suppose the input matrix A can be written as $B + E$, where B satisfies the following conditions: (1) it is block-diagonal matrix with k blocks, (2) the largest block size is $O(\frac{n}{k})$, (3) it has all row sums equal to 1, and (4) $\lambda_{k+1}(B) + ||E|| \leq \delta < 1/2$. Then the spectral clustering algorithm applied to A misclassifies $O(\delta^2 n)$ rows.*

Proof. Since B is block-diagonal with k blocks, its top k eigenvalues are all 1 (the top eigenvalue of each block is 1). Let the blocks of B be $B_1, B_2, \ldots, B_k$ and let their sizes be $n_1, n_2, \ldots, n_k$. The i'th eigenvector of B has support corresponding to the rows of B_i, and each entry in its support has value $1/\sqrt{n_i}$. Let Y_k be the $n \times k$ matrix whose columns are these eigenvectors. So $B = B_k + B_{n-k}$ where $B_k = Y_k Y_k^T$. Therefore,

$$A = B_k + B_{n-k} + E = B_k + E'$$

where, by assumption,

$$||E'|| \leq ||B_{n-k}|| + ||E|| = \lambda_{k+1}(B_n) + ||E|| \leq \delta.$$

Let X_k be the matrix whose columns are the top k eigenvectors of A. By Stewart's theorem ([19], Theorem 4.11, page 745) applied to A, B_k and E',

$$\bar{X}_k = Y_k (I + P^T P)^{-\frac{1}{2}}$$

is an invariant subspace for A, i.e., $X_k = \bar{X}_k U$ for some orthonormal matrix U, and

$$||P|| \leq \frac{2||E'||}{1 - 2||E'||}$$

It follows that

$$X_k = Y_k U + F$$

where

$$||F|| \leq \frac{2\delta}{1 - 2\delta}$$

The matrix we use for clustering is $C = AX_k$. Let us compare this with Y_kU.

$$\begin{aligned}
||C - Y_kU|| &= ||(B_k + E')(Y_kU + F) - Y_kU|| \\
&= ||B_kF + E'Y_kU + E'F|| \\
&\leq ||F|| + ||E'|| + ||E'||\,||F|| \\
&\leq \frac{2\delta}{1-2\delta} + \delta + \frac{2\delta^2}{1-2\delta} \\
&= \frac{3\delta}{1-2\delta}
\end{aligned}$$

This means that C is close to a rotated version of Y_k!

The bound on the 2-norm also implies a bound on the Frobenius norm:

$$||C - Y_kU||_F^2 \leq k||C - Y_kU||^2 \leq \frac{9\delta^2 k}{(1-2\delta)^2} \tag{6}$$

Let $y_1, y_2, \ldots y_n$ be the rows of Y_kU. Each y_i that belongs to block j has $|y_i| = 1/\sqrt{n_j}$ (rotation preserves lengths). Note that for any i, j from the same block of B, $y_i = y_j$ and for i, j from different blocks, $y_i \cdot y_j = 0$. Let $c_1, c_2, \ldots c_n$ be the rows of C. We will next show that for most i, c_i is close to y_i.

Call an element i *distorted* if $|c_i - y_i| \geq |y_i|/9$. Let the set of distorted rows be D. Suppose m rows are distorted. Then

$$\sum_{i \in D} |c_i - y_i|^2 \geq \frac{1}{81} \sum_{i \in D} |y_i|^2 \geq \frac{m}{81n'}$$

where n' is the size of the largest block. However, using (6),

$$\frac{m}{n'} \leq 81 \sum_{i \in D} |c_i - y_i|^2 \leq 81||C - Y_kU||_F^2 \leq \frac{3^6\delta^2 k}{(1-2\delta)^2}$$

Hence,

$$m \leq \frac{3^6\delta^2 kn'}{(1-2\delta)^2} = O(\delta^2 n)$$

with our assumptions that $n' = \Omega(n/k)$ and $\delta < 1/2$.

Next, consider the set that is not distorted. For any i, j such that $i, j \notin D$,

$$\begin{aligned}
c_i \cdot c_j &= (y_i + c_i - y_i) \cdot (y_j + c_j - y_j) \\
&= y_i \cdot y_j + (c_i - y_i) \cdot y_j + (c_j - y_j) \cdot y_i + (c_i - y_i) \cdot (c_j - y_j).
\end{aligned}$$

Hence,

$$\begin{aligned}
|c_i \cdot c_j - y_i \cdot y_j| &= |(c_i - y_i) \cdot y_j + (c_j - y_j) \cdot y_i + (c_i - y_i) \cdot (c_j - y_j)| \\
&\leq |(c_i - y_i) \cdot y_j| + |(c_j - y_j) \cdot y_i| + |(c_i - y_i) \cdot (c_j - y_j)| \\
&\leq \frac{2}{9}|y_i|\,|y_j| + \frac{1}{81}|y_i|\,|y_j| \\
&\leq \frac{19}{80}|c_i|\,|c_j| \\
&< \frac{1}{4}|c_i|\,|c_j|
\end{aligned}$$

Here we have used the fact that for $i \notin D$,

$$|y_i|^2 \le |c_i|^2 + |y_i - c_i|^2 \le |c_i|^2 + \frac{1}{81}|y_i|^2$$

and hence $|y_i|^2 \le \frac{81}{80}|c_i|^2$. This implies that for two non-distorted rows i,j, we have $c_i \cdot c_j > \frac{3}{4}|c_i|\,|c_j|$ if i,j are from the same block of B, and $c_i \cdot c_j < \frac{1}{4}|c_i|\,|c_j|$ otherwise. The theorem follows (the key observation is that two undistorted rows from different clusters stay in different clusters). □

5 Conclusion

There are two basic aspects to analyzing a clustering algorithm

- Quality: how good is the clustering produced?
- Speed: how fast can it be found?

In this paper we have mostly dealt with the former issue while taking care that the algorithms are polynomial time. The spectral algorithms depend on the time it takes to find the top (or top k) singular vector(s). While this can be done in polynomial time, it might still too expensive for applications such as information retrieval. The work of [10] and [8] on randomized algorithms for low-rank approximation addresses this problem. The running time of their first algorithm depends only on the quality of the desired approximation and not on the size of the matrix, but it assumes that the entries of the matrix can be sampled in a specific manner. Their second algorithm needs no assumptions and has a running time that is linear in the number of non-zero entries. More recently, [6] describes an efficient implementation of spectral algorithm II that maintains sparsity, and also gives experimental evidence that it performs favorably compared to other well-known clustering algorithms.

Acknowledgment. We thank Anna Karlin for pointing out an error in an earlier version.

References

[1] C. Alpert, A Kahng and Z. Yao, "Spectral partitioning: the more eigenvectors the better", *Discrete Applied Mathematics*, **90**, pp3-26, 1999.

[2] Y. Azar, A. Fiat, A. Karlin, F. McSherry and J. Saia, "Spectral analysis of data", *Proceedings of 33rd Symposium on Theory of Computing*, pp619-626, 2001.

[3] A. Benczur and D. Karger, "Approximate $s-t$ min-cuts in $O(n^2)$ time", *Proceedings of 28th Symposium on Theory of Computing*, pp47-55, 1996.

[4] M. Charikar, C. Chekuri, T. Feder and R. Motwani, "Incremental clustering and dynamic information retrieval", *Proceedings of 29th Symposium on Theory of Computing*, 1997.

[5] M. Charikar, S. Guha, D. Shmoys and E. Tardos. "A constant-factor approximation for the k-median problem." *Proceedings of 31st Symposium on Theory of Computing*, pp1-10, 1999.

[6] D. Cheng, R. Kannan, S. Vempala and G. Wang, "On a recursive spectral algorithm for clustering from pairwise similarities", *MIT LCS Technical Report MIT-LCS-TR-906*, 2003.

[7] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning", *Knowledge Discovery and Data Mining*, pp269-274, 2001.

[8] P. Drineas, A. Frieze, R. Kannan, S. Vempala and V. Vinay, "Clustering in large graphs and matrices", *Proceedings of 10th Symposium on Discrete Algorithms*, pp291-299, 1999.

[9] M. Dyer and A. Frieze, "A simple heuristic for the p-center problem", *Operations Research Letters*, **3(6)**, pp285-288, 1985.

[10] A. Frieze, R. Kannan and S. Vempala, "Fast Monte-Carlo algorithms for finding low-rank approximations", *Proceedings of 39th Symposium on Foundations of Computer Science*, pp370-378, 1998.

[11] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[12] P. Indyk, "A sublinear time approximation scheme for clustering in metric spaces", *Proceedings of 40th Symposium on Foundations of Computer Science*, pp154-59, 1999.

[13] K. Jain and V. Vazirani, "Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation", *Journal of the ACM*, **48(2)**, pp274-296, 2001.

[14] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms", *Journal of the ACM*, **46(6)**, pp787-832, 1999.

[15] C. Papadimitriou, P. Raghavan, H. Tamaki and S. Vempala, "Latent semantic indexing: a probabilistic analysis", *Journal of Computer and System Sciences*, **61**, pp217-235, 2000.

[16] J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22(8)**, pp888-905, 2000. See `http://www-2.cs.cmu.edu/~jshi/Grouping/`

[17] A. Sinclair and M. Jerrum, "Approximate counting, uniform generation and rapidly mixing Markov chains", *Information and Computation*, **82**, pp93-133, 1989.

[18] D. Spielman and S. Teng, "Spectral partitioning works: planar graphs and finite element meshes", *Proceedings of 37th Symposium on Foundations of Computer Science*, pp349-358, 1996.

[19] G. Stewart, "Error and perturbation bounds for subspaces associated with certain eigenvalue problems", *SIAM Review*, **15(4)**, pp727-64, 1973.

[20] Y. Weiss, "Segmentation using eigenvectors: a unifying view", *Proceedings of IEEE International Conference on Computer Vision*, pp975-982, 1999.

[21] See `http://www-math.mit.edu/cluster/`