

## Simple, direct and efficient multi-way spectral clustering

ANIL DAMLE<sup>†</sup>

*Department of Computer Science, Cornell University, 423 Gates Hall, Ithaca, NY 14853*

<sup>†</sup>Corresponding author. Email: damle@cornell.edu

VICTOR MINDEN

*Center for Computational Biology, Flatiron Institute, 162 Fifth Avenue, New York, NY 10010*

vminden@flatironinstitute.org

AND

LEXING YING

*Department of Mathematics and Institute for Computational & Mathematical Engineering,  
Stanford University, 450 Serra Mall, Bldg 380, Stanford, CA 94305*

lexing@stanford.edu

[Received on 15 April 2017; revised on 4 December 2017; accepted on 13 April 2018]

We present a new algorithm for spectral clustering based on a column-pivoted QR factorization that may be directly used for cluster assignment or to provide an initial guess for  $k$ -means. Our algorithm is simple to implement, direct and requires no initial guess. Furthermore, it scales linearly in the number of nodes of the graph and a randomized variant provides significant computational gains. Provided the subspace spanned by the eigenvectors used for clustering contains a basis that resembles the set of indicator vectors on the clusters, we prove that both our deterministic and randomized algorithms recover a basis close to the indicators in Frobenius norm. We also experimentally demonstrate that the performance of our algorithm tracks recent information theoretic bounds for exact recovery in the stochastic block model. Finally, we explore the performance of our algorithm when applied to a real-world graph.

*Keywords:* spectral clustering; column-pivoted QR; stochastic block model.

Spectral clustering has found extensive use as a mechanism for detecting well-connected subgraphs of a network. Typically, this procedure involves computing an appropriate number of eigenvectors of the (normalized) Laplacian and subsequently applying a clustering algorithm to the embedding of the nodes defined by the eigenvectors. Currently, one of the most popular algorithms is  $k$ -means++ [6], the standard iterative  $k$ -means algorithm [32] applied to an initial clustering chosen via a specified random sampling procedure. Due to the non-convex nature of the  $k$ -means objective, however, this initialization does not preclude convergence to local minima, which can be poor clusterings.

We provide an alternative, direct (non-iterative) procedure for clustering the nodes in their eigenvector embedding. It is important to note that our procedure is not a substitute for  $k$ -means++ when tackling general (i.e., non-spectral) clustering problems. For spectral embeddings of graphs with



Symbol indicates reproducible data.

community structure, however, we take advantage of additional geometric structure of the embedding to build a more robust clustering procedure. Furthermore, our algorithm is built out of a simple column-pivoted QR factorization (CPQR), making it easy to implement and use. Finally, a simple randomized acceleration of our algorithm substantially reduces the cost of cluster assignment, making it feasible for large problems of practical interest.

## 1. Background and setup

Given a simple undirected graph  $\mathcal{G}$  with adjacency matrix  $A \in \{0, 1\}^{n \times n}$ , we consider the multi-way clustering problem of partitioning the vertices of  $\mathcal{G}$  into  $k$  disjoint clusters. A common (albeit unrealistic) generative model for graphs exhibiting this sort of cluster structure is the  $k$ -way *stochastic block model* (SBM).

**DEFINITION 1 (SBM [27])** Partition  $[n]$  into  $k$  mutually disjoint and non-empty clusters  $\mathcal{C}_1, \dots, \mathcal{C}_k$ . Given probabilities  $p$  and  $q$  such that  $p > q$ , let  $M \in [0, 1]^{n \times n}$  have entries  $M_{ii} \equiv 0$  and, for  $i \neq j$ ,

$$M_{ij} \equiv \begin{cases} p, & \{i, j\} \subset \mathcal{C}_k \text{ for some } k, \\ q, & \text{else.} \end{cases}$$

A symmetric adjacency matrix  $A \in \{0, 1\}^{n \times n}$  with  $A_{ij} \sim \text{Bernoulli}(M_{ij})$  for  $i < j$  and  $A_{ii} \equiv 0$  for all  $i$  is said to be distributed according to the  $k$ -way SBM with clusters  $\{\mathcal{C}_i\}_{i=1}^k$ , within-cluster probability  $p$  and between-cluster probability  $q$ .

For an SBM with equisized clusters, the maximum-likelihood estimate for the clusters can be phrased in terms of maximizing the number of within-cluster edges; that is, given  $A$ , find a matrix  $X$  whose columns are indicator vectors for cluster membership such that  $X$  attains the optimal value of the combinatorial optimization problem

$$\begin{aligned} & \underset{X}{\text{maximize}} && \text{Tr}(X^T A X) \\ & \text{subject to} && X \in \{0, 1\}^{n \times k}, \quad X^T X = \frac{n}{k} I_k. \end{aligned} \tag{1.1}$$

If  $A$  is not assumed to be a random sample from the SBM, then the above problem does not have the interpretation of maximum-likelihood estimation, though it remains a common starting point for clustering. Given that the combinatorial optimization problem is NP-hard, it is typical to relax (1.1) to a computationally tractable convex formulation.

A common relaxation of (1.1) is to remove the restriction that  $X \in \{0, 1\}^{n \times k}$  and instead optimize over real-valued matrices,

$$\begin{aligned} & \underset{X}{\text{maximize}} && \text{Tr}(X^T A X) \\ & \text{subject to} && X \in \mathbf{R}^{n \times k}, \quad X^T X = \frac{n}{k} I_k. \end{aligned} \tag{1.2}$$

While this optimization problem is still non-convex, it follows from the Courant–Fischer–Weyl min-max principle that an optimal point  $X_*$  is given as  $X_* = V_k Q$ , where  $V_k \in \mathbf{R}^{n \times k}$  contains the eigenvectors of  $A$  corresponding to the  $k$  largest eigenvalues and  $Q \in \mathbf{O}_k$  is an arbitrary orthogonal transformation.

Because the solution  $X_*$  is no longer discrete, the canonical spectral clustering approach uses the rows of  $X_*$  as coordinates in a standard point-cloud clustering procedure such as `k-means`.

We propose an algorithm based on a CPQR of the matrix  $V_k^T$  that can be used either as a stand-alone clustering algorithm or to initialize iterative algorithms such as `k-means`. Our approach stems from the computational quantum chemistry literature, where the same basic procedure has been developed as a method for constructing spatially localized bases of an eigenspace associated with the nonlinear eigenproblem of Kohn–Sham density functional theory [13,14]. Our numerical experiments show that our approach closely tracks the information theoretic limit in terms of exact clustering of the SBM, whereas `k-means++` on its own does not.

### 1.1 Related work

An algorithm using a column-pivoted QR for cluster assignment of general point-cloud data—with an assumption of orthogonality amongst the clusters—has previously appeared in the literature in the context of spectral relaxation of the `k-means` objective [43]. Curiously, though we find the basic idea to be powerful, this algorithmic approach seems to have been ignored and we can find no further reference to it. We expand upon this work, taking advantage of algorithmic improvements appearing in the computational chemistry literature for greater efficiency. Further, by addressing the sparse adjacency matrix case instead of the Gram matrix of point-cloud data, we are able to strongly motivate our approach based on proven spectral properties of model graphs and graphs with assumed underlying cluster structure.

Spectral methods as we discuss here stem from work on the Fiedler vector [16,18] for the two-block case and spectral embeddings coupled with `k-means` clustering [37] in the multi-block case, for a broad overview see, e.g., [42]. For a more comprehensive overview of initialization procedures for `k-means` see, e.g., Celebi *et al.* [11]. Recent analysis of these methods applied to the SBM [20,39] focuses on misclassification of nodes. Another line of work considers matrices besides the adjacency matrix or normalized Laplacian to achieve theoretical detection thresholds [28,34]. Other recent work [35] demonstrates where spectral methods break down and argues for the use of semidefinite programming (SDP)-based methods.

Recently, there has been significant work on understanding when it is possible to exactly recover communities in the sparse SBM, wherein the probability of connections between nodes is  $\Theta(\log n/n)$ . Specifically, the information theoretic bound for when exact recovery is possible with two blocks was developed by Abbe *et al.* [1] and an SDP-based algorithm achieving the bound was proposed. Additional recent works [3,24,25] have extended this theory to the multi-block case (for a constant or slowly growing number of clusters) and generalized the SDP approach. We will return to this example in the numerical results section.

## 2. Algorithm

### 2.1 Preliminaries

Suppose  $\mathcal{G}$  is a simple undirected graph with symmetric adjacency matrix  $A \in \{0, 1\}^{n \times n}$ . Let the eigendecomposition of  $A$  be given by

$$A = [V_k \mid V_{n-k}] \begin{bmatrix} A_k & 0 \\ 0 & A_{n-k} \end{bmatrix} [V_k \mid V_{n-k}]^T,$$

where  $V_k \in \mathbf{R}^{n \times k}$  and  $V_{n-k} \in \mathbf{R}^{n \times (n-k)}$  contain pairwise orthonormal columns of eigenvectors, and  $\Lambda_k = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_k)$  contains the  $k$  largest eigenvalues of  $A$  sorted as  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$ . Of interest in spectral clustering is the structure of the eigenspace spanned by the columns of  $V_k$  or a related matrix, as we make explicit below.

**DEFINITION 2** Suppose  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$  is a clustering of the vertex set  $[n]$ . We say the matrix  $W' \in \{0, 1\}^{n \times k}$  with entries given by

$$W'_{ij} \equiv \begin{cases} 1, & \text{if vertex } i \text{ belongs to cluster } j, \\ 0, & \text{otherwise,} \end{cases}$$

is an *indicator matrix* for the underlying clustering. If  $W'$  is an indicator matrix and  $W \in \mathbf{R}_+^{n \times k}$  is given by scaling the columns of  $W'$  to have unit  $\ell_2$  norm, then we call  $W$  a *normalized indicator matrix* for the underlying clustering.

Suppose  $A$  is sampled from the  $k$ -way SBM on  $n$  nodes with within- and between-cluster connection probabilities  $p$  and  $q$ . Letting  $W$  be a normalized indicator matrix for the underlying clustering and provided that  $p$  and  $q$  do not decay too quickly asymptotically, standard matrix concentration inequalities [8,41] coupled with the Davis–Kahan theorem [15] give convergence of the spectral projector  $V_k V_k^T$  in  $n$ , i.e., for fixed  $k$  there exists  $\delta_n = o(1)$  such that

$$\left\| WW^T - V_k V_k^T \right\|_2 \leq \delta_n$$

with high probability. The exact nature of these concentration results depends strongly on the asymptotic behavior of  $p$  and  $q$ , however, the general idea is that with high probability any given  $A$  will be close to the average adjacency matrix. For example, if  $k$  is fixed, the clusters are equisized, and  $p, q \sim \frac{\log n}{n}$  then  $\delta_n \sim \frac{1}{\sqrt{\log n}}$  as we will elaborate on in Section 2.4,

More generally, we would like to consider the case where  $A$  is a symmetric adjacency matrix not necessarily coming from the SBM. In this setting it is more difficult to talk about asymptotic convergence. Therefore, we will generically consider the case where there exists a normalized indicator matrix  $W$  and  $\varepsilon > 0$  such that the eigenvector matrix  $V_k$  satisfies

$$\left\| WW^T - V_k V_k^T \right\|_2 \leq \varepsilon.$$

From this bound on the distance between spectral projectors, we can conclude the existence of  $Q \in \mathbf{O}_k$  such that

$$\|W - V_k Q\|_F \leq \sqrt{k}\varepsilon + \mathcal{O}(\varepsilon^2). \quad (2.1)$$

As we will observe, for these relations to be useful in the construction of our algorithms  $\varepsilon$  must be sufficiently small in a manner that depends on  $k$  and  $n$ .

Concisely, we expect the eigenvector matrix  $V_k$  to look like the normalized indicator matrix  $W$ , under a suitable orthogonal transformation. Alternatively,  $V_k Q$  provides an approximately sparse basis for the range of  $V_k$ . Thus, while any orthogonal transformation of the (appropriately scaled) eigenvectors is a solution to (1.2), there exists a specific solution that directly reflects the underlying cluster structure. If  $W$  and  $V_k$  in (2.1) were both known, finding an orthogonal transformation to make  $V_k$  resemble  $W$  is

not a difficult problem. However, we seek to find this transformation without knowing the clustering *a priori*.

In general, when handed a graph that may not come from the SBM, we cannot leverage explicit knowledge of eigenvectors. However, for graphs that admit a  $k$ -way clustering, there is a slightly more general geometric structure we can appeal to—so-called orthogonal cone structure (OCS). Motivation for this model and the nomenclature is borrowed from Schiebinger *et al.* [40]. However, they address a slightly different (though related) problem and we do not use their model exactly. Additional justification for finding OCS in graphs with community structure may be found in Gharan & Trevisan [21] and Benzi *et al.* [9].

To aid in our discussion, for a given orthonormal vector  $q \in \mathbf{R}^k$  and scalar  $\mu \in [0, 1]$  we define the cone  $\mathcal{K}_\mu(q)$  as

$$\mathcal{K}_\mu(q) \equiv \left\{ x \in \mathbf{R}^k \mid \frac{x^T q}{\|x\|_2} \geq 1 - \mu \right\}.$$

Given this simple definition of a cone, we may now define OCS rigorously.

**DEFINITION 3** A set of points  $\{x_i\}_{i=1}^n$  in  $\mathbf{R}^k$  exhibits OCS with parameters  $\eta \in (0, 1]$ ,  $\mu \in [0, 1]$  and  $\delta > 0$  if the following hold:

1. There exists a set of orthonormal vectors  $\{q_i\}_{i=1}^k$  such that at least  $\eta n$  of the  $x_i$  satisfy

$$x_i \in \mathcal{K}_\mu(q_j) \tag{2.2}$$

for exactly one  $j = 1, \dots, k$ . Furthermore, each of the  $\mathcal{K}_\mu(q_j)$  contains at least one  $x_i$ .

2. Let  $\mathcal{I}$  denote the set of  $x_i$  that satisfies (2.2) for some  $j$ , then

$$\|x_i\|_2 \leq \delta \min_{l \in [k]} \max_{x_j \in \mathcal{K}_\mu(q_l)} \|x_j\|_2 \quad \forall x_i \notin \mathcal{I}.$$

When these two conditions are satisfied for a set of points in  $\mathbf{R}^k$  we say  $\{x_i\}_{i=1}^n \in \text{OCS}_k(\eta, \mu, \delta)$ .

This definition essentially says that given  $n$  points, a fraction  $\eta$  of them can be assigned uniquely to a set of cones with orthogonal centers, while the remaining points have small norm relative to the largest point in each cone. Importantly, given the cone centers  $\{q_j\}_{j=1}^k$  it is easy to correctly cluster the points that lie in the cones—under the assumption that the cones reflect the desired cluster structure. For each point this is accomplished by checking which  $q_j$  it has largest magnitude inner product with and assigning it to that cluster. Those points that lie outside the cones are assumed to not be particularly well suited to any of the clusters and any assignment is considered acceptable.

In both OCS and SBM we have  $n$  points contained in  $k$  cones, the centers of which are assumed to be orthogonal. However, they are represented in an arbitrary coordinate system using the eigenvectors, which means the embedding coordinates of each point do not make it easy to determine which points belong to which cone. If we can find a coordinate system roughly aligned with the centers of the cones, we can then rotate the points into this system and read off the cluster assignment based off the largest magnitude entry. This idea is illustrated in Fig. 1.

We accomplish this task primarily with the use of a CPQR. We define this factorization notationally for matrices that have more columns than rows, which is the setting we will require later.

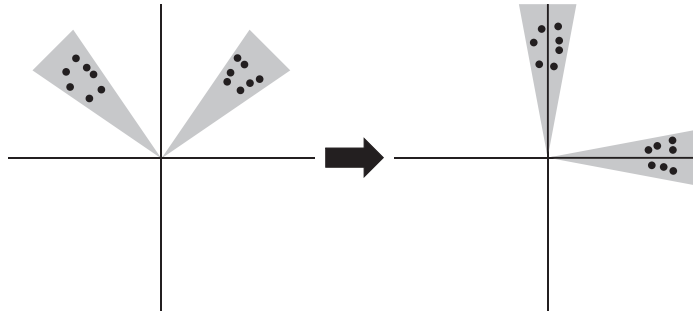


FIG. 1. Two-dimensional demonstration of the expected OCS. While the computed eigenvectors (left) may not readily allow for coordinate-based clustering, there is a rotation of the coordinate system (right) that does.

DEFINITION 4 Let  $B \in \mathbf{R}^{k \times n}$  with  $k \leq n$  have rank  $k$ . The CPQR of  $B$  is the factorization

$$B\Pi = Q[R_1 \ R_2]$$

as computed via the Businger–Golub algorithm [10] (see also Chapter 5 of Golub & Van Loan [22]) where  $\Pi \in \mathbf{R}^{n \times n}$  is a permutation matrix,  $Q \in \mathbf{O}_k$  is an orthogonal matrix,  $R_1 \in \mathbf{R}^{k \times k}$  is upper triangular and  $R_2 \in \mathbf{R}^{k \times (n-k)}$ .

Ideally, the permutation  $\Pi$  is chosen to make  $R_1$  as well conditioned as possible given  $B$ . The Golub–Businger algorithm chooses this permutation using a greedy heuristic, wherein at each step of the algorithm the column with largest norm is picked as the pivot and the remaining columns are orthogonalized against it.

Lastly, we briefly define a slightly less common matrix factorization we leverage in our algorithm—the polar factorization.

DEFINITION 5 For a matrix  $B \in \mathbf{R}^{k \times k}$ , the *polar factorization* of  $B$  is the factorization

$$B = UH,$$

where  $U \in \mathbf{R}^{k \times k}$  is orthogonal and  $H \in \mathbf{R}^{k \times k}$  is positive semi-definite.

Such a factorization always exists and, if  $B$  has full rank,  $H$  is guaranteed to be positive definite. Importantly, in any orthogonally invariant norm  $U$  is the closest orthogonal matrix to  $B$  [17]. The polar factorization is also relatively insensitive to perturbations. For example, if  $B = UH$  and  $\tilde{B} = \tilde{U}\tilde{H}$  are two polar factorizations and both  $B$  and  $\tilde{B}$  are non-singular, then

$$\|U - \tilde{U}\|_F \leq \frac{2}{\sigma_k(B) + \sigma_k(\tilde{B})} \|B - \tilde{B}\|_F, \quad (2.3)$$

see, e.g., Theorem 1 [31]. Finally, the polar factorization may be computed easily from the orthogonal factors from the singular value decomposition of  $B$  with computational cost  $\mathcal{O}(k^3)$  or by a specialized routine. Further details may be found in, e.g., Higham [26].

## 2.2 Algorithm statement

Given  $V_k$  we propose the following algorithm for finding  $k$  clusters in the graph:

Note that  $|\cdot|$  is used to represent the elementwise absolute value operation when applied to a matrix, and here we assume that every node belongs to exactly one cluster. For real-world applications, if all entries in a column are small we may optionally not assign the corresponding node to any cluster, but we do not explore that further at this time.

Algorithm 1 is strikingly simple, there is no dependence on any sort of initial guess, it is naturally invariant to the data ordering, and it is constructed out of common and efficient matrix factorizations. From an implementation standpoint the CPQR may be computed using, e.g., the `qr` function in MATLAB<sup>®</sup> or the LAPACK [5] routine `DGEQP3` [38]. Overall, the cost of this algorithm is linear in the number of nodes of the graph and quadratic in the number of clusters, i.e.,  $\mathcal{O}(nk^2)$ . The lack of an initial guess is a particularly important distinction from the generally used `k-means` algorithm.

---

### Algorithm 1 CPQR-based spectral clustering

---

- 1: **input:** Invariant subspace  $V_k$
- 2: Compute the CPQR factorization

$$V_k^T \Pi = QR.$$

- 3: Let  $\mathcal{C}$  denote the  $k$  columns selected as the first  $k$  columns of  $\Pi$ .
- 4: Compute the polar factorization

$$\left( V_k^T \right)_{:, \mathcal{C}} = UH.$$

- 5: For each  $j \in [n]$  assign node  $j$  to cluster

$$c_j \equiv \arg \max_i \left( \left| U^T V_k^T \right|_{i,j} \right).$$

- 6: **return** Cluster assignment and  $U$
- 

A simple randomized variant of our algorithm allows us to more efficiently compute a matrix  $U$  such that  $U^T V_k$  may be used for cluster assignment with high probability. To do this, we define the probability distribution  $\rho$  over  $[n]$  as

$$\rho_j \equiv \Pr(\{j\}) = \frac{\left\| \left( V_k^T \right)_{:,j} \right\|_2}{k}. \quad (2.4)$$

Sampling strategies based on  $\rho$  are not new and correspond to sampling via the so-called leverage scores [33] of the spectral projector  $V_k V_k^T$ . Generically, similar strategies are common in the construction of low-rank factorizations [19]. However, here we have a different goal. The spectral projector is always exactly rank  $k$ , and we are not concerned with sufficiently sampling its range.

Given  $V_k$ , an oversampling factor  $\gamma$  and a failure probability  $\delta$ , the following algorithm computes the cluster assignments:

---

**Algorithm 2** Randomized CPQR-based spectral clustering
 

---

- 1: **input:** Invariant subspace  $V_k$ , oversampling factor  $\gamma$  and failure probability  $\delta$
- 2: Sample a set  $\mathcal{J}$  of  $\gamma k \log(\frac{k}{\delta})$  nodes from  $[n]$  based on the discrete distribution  $\rho$  in (2.4).
- 3: Compute the CPQR factorization

$$\begin{pmatrix} V_k^T \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}_{:, \mathcal{J}} \Pi = QR.$$

- 4: Let  $\mathcal{C}$  denote the  $k$  columns selected as the first  $k$  columns of  $\Pi$ .
- 5: Compute the polar factorization

$$\left( \begin{pmatrix} V_k^T \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}_{:, \mathcal{J}} \right)_{:, \mathcal{C}} = UH.$$

- 6: For each  $j \in [n]$  assign node  $j$  to cluster

$$c_j \equiv \arg \max_i \left( \left| U^T V_k^T \right|_{i,j} \right).$$

- 7: **return** Cluster assignment and  $U$
- 

The cost of the CPQR is now  $\mathcal{O}(k^3 \log k)$ , yielding a cluster assignment operator in time independent of  $n$ . While computing  $U^T V_k^T$  in step 3 formally scales as  $\mathcal{O}(nk^2)$ , in practice the dominant cost of the deterministic algorithm is the CPQR. If one only wishes to cluster a subset of the nodes or check if two nodes are in the same cluster, then  $U^T$  need only be applied to some columns of  $V_k^T$ . Furthermore,  $U^T V_k^T$  scales well when computed on parallel architectures and the assumed structure, specifically the locality, of the product could also potentially be used to reduce the cost of its computation. The other dependence on  $n$  in the form of  $\mathcal{O}(nk)$  is in computing the column norms of  $V_k^T$  to build  $\rho$ . This operation also parallelizes well and asymptotically costs no more than writing down  $V_k$  itself. To truly make the cost of constructing  $U$  independent of  $n$ , however, one may sample nodes uniformly to build  $\mathcal{J}$ . In this case one must sample slightly more than  $k \log k$ , in a manner that depends on the size of the smallest cluster relative to  $n$ , to ensure the same failure probability.

Prior to embarking on the proofs, we provide a brief high-level rationale for the use of a CPQR. Assuming every point belongs to exactly one cluster, the  $k$ -means algorithm applied to the spectral embedding may be expected to work in principle given a sufficiently close initial guess for the cluster centers. Equivalently, given one node from each cluster we may use it as a proxy for a cone center. It turns out the CPQR procedure achieves this goal when we are working with the spectral embedding for a graph with community structure.

Heuristically, consider picking an initial node at random. It belongs to one of the clusters, and thus might be a reasonable surrogate for a cone center that represents that cluster. Now, we need a procedure to pick a point that is guaranteed to be in a different cluster to use as another center. For general clustering problems this may be difficult. However, here we observed that the clusters have additional structure: we expect them to be orthogonal. This motivates the use of the CPQR. Once we orthogonalize all of the



points against the center point of the first cluster, the norms of other points in the same cluster become small while the norms of points in different clusters remain more or less the same. Therefore, picking the point with largest remaining norm should yield a viable center for a different cluster. By repeating this procedure  $k$  times we hope to get a good set of  $k$  different cluster centers. This procedure is codified by the CPQR factorization introduced earlier.

We can also motivate the development of Algorithm 2 from this viewpoint. If the key goal of the CPQR is to identify one point (column) per cone, we do not need to consider all of the columns of  $V_k^T$ . Our sub-sampling strategy is designed to produce a matrix  $(V_k^T)_{:, \mathcal{J}}$  that contains at least one point per cone. In particular, the use of leverage scores ensures that the probabilities of selecting a column from a given cluster are all (roughly) equal. We may then proceed as before and utilize the CPQR to identify the set of  $k$  points. Failing to sample one point per cluster will result in failure of the remainder of the algorithm as we will not be able to identify the cluster we failed to draw a sample from. However, if we successfully sample one point per cluster then the algorithm will proceed in the same manner as the deterministic variant.

The preceding procedure ideally gives us  $k$  points, each of which represents one of the underlying cones. However, we expect the cones to exhibit orthogonal structure and the selected cluster centers may not be quite orthogonal. Hence, we compute the ‘closest’ orthogonal matrix to  $(V_k^T)_{:, \mathcal{C}}$  via the polar factorization. This orthogonal matrix defines a coordinate system that aligns with the expected structure of the points. Once viewed in this coordinate system (by applying the appropriate rotation) we may simply cluster the points based on their largest magnitude entry.

**REMARK 1** One may, alternatively, simply use the  $Q$  factor from the CPQR as the new coordinate system, which corresponds to a greedy orthogonalization of the  $k$  points. In the later examples, because  $R_1$  ends up being quite well conditioned there is not a significant difference between these two strategies. The polar factorization has the potential advantage of not giving preferential treatment to any of coordinate directions based on the order in which they were selected.

### 2.3 Analysis

To make our analysis concrete, we assume the indicator structure inherent to the SBM. The justification for OCS is the same. In particular, one may relax the model for  $W$  to represent more general OCS. As long as  $W$  is assumed to only contain one non-zero per row the following analysis still holds (with slightly larger constants to account for the fact that each column of  $W$  may contain more than one non-zero value). This is reasonable because the problem is fundamentally the same for both the SBM and more general OCS: given a collection of orthogonal indicator vectors that have been arbitrarily rotated and perturbed, how do we recover the underlying structure?

Theorem 7.2 of Gu & Eisenstat [23] proves that for a CPQR as defined above

$$\sigma_k(R_1) \geq \frac{\sigma_k(B)}{f(n, k)}, \quad (2.5)$$

where  $\sigma_j$  denotes the  $j^{\text{th}}$  singular value of a matrix and  $f(n, k) = 2^k \sqrt{n}$ . This bound ensures that the singular values of  $R_1$  are not too much smaller than the singular values of  $B$ . The CPQR defined here is related to a broader class of rank-revealing QR (RRQR) factorizations [12, 23] that seek a permutation  $\Pi$  such that (2.5) holds for different forms of the denominator  $f(n, k)$ , e.g.,  $\sqrt{k(n-k)} + \min(n, n-k)$ . Because the matrices we deal with are quite far from the pathological worst-case examples, we find the standard CPQR algorithm sufficient for our needs.

We will show our algorithm finds a matrix  $U$  such that  $V_k U$  is close to  $W$  (up to a signed column permutation) without explicit knowledge of the sparsity structure of  $W$ . The potentially different column order inherited from the signed permutation simply corresponds to a relabeling of the clusters and, since we take the largest entry in absolute value to do cluster assignment, the sign does not matter.

We prove two lemmas that ensure the CPQR pivoting strategy applied to  $V_k^T$  (or a randomly selected subset of its columns) implicitly identifies  $k$  linearly independent columns of  $WW^T$ .

LEMMA 1 Let  $W \in \mathbf{R}^{n \times k}$  and  $V \in \mathbf{R}^{n \times k}$  have orthonormal columns and satisfy

$$\|WW^T - VV^T\|_2 \leq \varepsilon$$

with  $\varepsilon < 2^{-k}/\sqrt{n}$ . If  $V^T \Pi = QR$  is a CPQR factorization and  $\mathcal{C}$  denotes the original indices of the first  $k$  columns selected by the permutation  $\Pi$ , then

$$\text{range} \left\{ (WW^T)_{:, \mathcal{C}} \right\} = \text{range} \{W\}.$$

*Proof.* Theorem 7.2 of Gu & Eisenstat [23] coupled with the fact that  $\sigma_i(V) = 1$  for  $i = 1, \dots, k$  implies that  $\sigma_{\min}(R_1) \geq 2^{-k}/\sqrt{n}$ . Substituting in the CPQR factorization yields

$$(WW^T + E)\Pi = VQ \begin{bmatrix} R_1 & R_2 \end{bmatrix}$$

with  $\|E\|_2 \leq \varepsilon$ . Now,  $\sigma_{\min}(R_1) \geq \varepsilon$  implies that  $(W^T)_{:, \mathcal{C}}$  is non-singular, as otherwise the distance from  $VQR_1$  to the nearest singular matrix is less than  $\varepsilon$ , which is a contradiction. The result follows.  $\square$

We now consider a lemma necessary for Algorithm 2. Here, the input  $V$  is fixed and not random; randomness arises because we only compute a CPQR of a randomly selected subset of the columns of  $V^T$ . To ensure that our algorithm will be successful, we need to ensure that the random subsample contains at least one column associated with each cluster. If the sub-sampling strategy fails, it becomes impossible for the CPQR to identify one point per cluster, and thus the algorithm will fail with some controllable probability.

LEMMA 2 Let  $W \in \mathbf{R}^{n \times k}$  be a normalized indicator matrix and  $V \in \mathbf{R}^{n \times k}$  have orthonormal columns and satisfy

$$\|WW^T - VV^T\|_2 \leq \varepsilon$$

with  $\varepsilon < 2^{-k}/\sqrt{n}$  and assume there exists  $\gamma > 0$  such that

$$\frac{1}{\gamma} \leq \sum_{i \in \mathcal{C}_j} \|(V^T)_{:, i}\|_2^2$$

for  $j \in [k]$ . Let  $\mathcal{J}$  denote  $\gamma k \log \frac{k}{\delta}$  samples with replacement from the discrete distribution  $\rho$  over  $[n]$ . If  $(V^T)_{:, \mathcal{J}} \Pi = QR$  is a CPQR factorization and  $\mathcal{C}$  denotes the original indices of the first  $k$  columns selected by the permutation  $\Pi$ , then with probability at least  $1 - \delta$

$$\text{range} \left\{ (WW^T)_{:, \mathcal{C}} \right\} = \text{range} \{W\}.$$

*Proof.* Because

$$\frac{1}{\gamma} \leq \sum_{i \in \mathcal{C}_j} \left\| (V^T)_{:,i} \right\|_2^2$$

for  $j = 1 \in [k]$ , we can conclude that

$$\frac{1}{\gamma k} \leq \sum_{i \in \mathcal{C}_j} \rho_j$$

for  $j = 1 \in [k]$ . Now, a simple coupon-collecting argument [36] over  $k$  bins, namely  $\{\mathcal{C}_i\}_{i=1}^k$ , where the smallest mass a bin can have is  $1/(\gamma k)$  ensures that  $\mathcal{J}$  contains at least one point from each set  $\mathcal{C}_i$  with probability at least  $1 - \delta$ . If this event occurs, then  $(W^T)_{:, \mathcal{J}}$  is non-singular and the result follows from the same argument as Lemma 1.  $\square$

**REMARK 2** In either lemma, one may substitute the CPQR for a different RRQR [12] or strong RRQR [23] and inherit the appropriate upper bound on  $\varepsilon$  based on the corresponding  $f(n, k)$ .

By assuming that  $\|WW^T - VV^T\| \leq \varepsilon$ , (2.1) implies that  $W \approx VZ$  for some unknown orthogonal matrix  $Z$ . The key aspect of Lemmas 1 and 2 is that the CPQR identifies sufficient information to allow the construction of an orthogonal matrix  $U$  that is close to  $Z$ .

**THEOREM 1** Let  $W \in \mathbf{R}^{n \times k}$  be a normalized indicator matrix and suppose  $n > 4$ . Let  $V \in \mathbf{R}^{n \times k}$  with orthonormal columns satisfy

$$\|WW^T - VV^T\|_2 \leq \varepsilon$$

with  $\varepsilon < 2^{-k}/\sqrt{n}$ . If  $U$  is computed by Algorithm 1 with  $V$  as input, then there exists a permutation matrix  $\widehat{\Pi}$  such that

$$\|W\widehat{\Pi} - VU\|_F \leq 2\varepsilon\sqrt{kn} + \mathcal{O}(\varepsilon^2).$$

*Proof.* Let  $\mathcal{C}$  denote the original indices of the first  $k$  columns selected by the permutation  $\Pi$ . Based on the non-zero structure of  $W$  and Lemma 1, there exists a permutation matrix  $\widehat{\Pi}$  such that  $R_I \equiv \widehat{\Pi}^T (W^T)_{:, \mathcal{C}}$  is diagonal with positive diagonal entries. Our assumptions on  $V$  and  $W$  in conjunction with (2.1) allow us to assert the existence of an orthogonal matrix  $Z$  such that

$$\|Z\widehat{\Pi}R_I - (V^T)_{:, \mathcal{C}}\|_F \leq \sqrt{k}\varepsilon + \mathcal{O}(\varepsilon^2).$$

Now, observe that  $(Z\widehat{\Pi})R_I$  is the polar factorization of a matrix with orthogonal factor  $Z\widehat{\Pi}$  and symmetric positive definite factor  $R_I$ . Therefore, we can use (2.3) to bound the distance between  $Z\widehat{\Pi}$  and the orthogonal factor associated with the polar factorization of  $(V^T)_{:, \mathcal{C}}$  as

$$\|Z\widehat{\Pi} - U\|_F \leq 2\varepsilon\sqrt{kn} + \mathcal{O}(\varepsilon^2),$$

where we have used that the entries of  $R_I$  are bounded from below by  $1/\sqrt{n}$ . We may also infer from equation (2.1) that

$$\|W\widehat{\Pi} - VZ\widehat{\Pi}\|_F \leq \sqrt{k}\varepsilon + \mathcal{O}(\varepsilon^2).$$

Substituting  $U$  for  $Z\widehat{\Pi}$  and using the bound of their difference allows us to conclude the desired result.  $\square$

**THEOREM 2** Let  $W \in \mathbf{R}^{n \times k}$  be a normalized indicator matrix and suppose  $n > 4$ . Let  $V \in \mathbf{R}^{n \times k}$  with orthonormal columns satisfy

$$\|WW^T - VV^T\|_2 \leq \varepsilon$$

with  $\varepsilon < 2^{-k}/\sqrt{n}$  and assume there exists  $\gamma > 0$  such that

$$\frac{1}{\gamma} \leq \sum_{i \in \mathcal{C}_j} \|V_{:,i}\|_2^2$$

for  $j \in [k]$ . If  $U$  is computed by Algorithm 2 with  $V$  as input, then with probability at least  $1 - \delta$  there exists a permutation matrix  $\widehat{\Pi}$  such that

$$\|W\widehat{\Pi} - VU\|_F \leq 2\varepsilon\sqrt{kn} + \mathcal{O}(\varepsilon^2).$$

*Proof.* The proof mirrors that of Theorem 1, we simply use Lemma 2 in place of Lemma 1 to ensure that the permutation  $\widehat{\Pi}$  exists.  $\square$

**REMARK 3** The probabilistic characterization of Theorem 2 is entirely a consequence of the column sub-sampling in Algorithm 2 and its analysis via Lemma 2. If the column sub-sampling strategy is successful, the algorithm has the same properties as the deterministic variant. If the sub-sampling strategy fails, we cannot assert that  $R_j$  has positive diagonal entries and are unable to proceed with the analysis.

Theorems 1 and 2 show that if the subspace spanned by  $V_k$  is close to one containing indicator vectors on the clusters, then both of the proposed algorithms yield an orthogonal transformation of  $V_k$  approximating the cluster indicators. This representation of the subspace may then be directly used for cluster assignment. If desired, one may also use the computed clustering to seed the  $k$ -means algorithm and attempt to further improve the results with respect to the  $k$ -means objective function. This can either be accomplished by computing the cluster centers, or by using the points associated with the set  $\mathcal{C}$  as the initial cluster centers.

#### 2.4 Connections to the SBM

The preceding theory is rather general, as it only relies on normalized indicator structure of  $W$ . For example, in the case of the SBM with  $k$  clusters of  $m$  nodes each we let  $W$  be a normalized indicator matrix representing the range of  $\mathbb{E}A = M$ —a rank  $k$  matrix. In this setting, Corollary 8.1.11 of Golub & Van Loan [22] immediately implies for a given instance of the SBM

$$\|WW^T - V_k V_k^T\|_2 \leq \frac{4}{m(p-q) - p} \|A - M\|_2. \quad (2.6)$$

We now restrict ourselves to the regime where  $p, q \sim \frac{\log n}{n}$  and introduce the parametrization  $p \equiv \alpha \frac{\log m}{m}$  and  $q \equiv \beta \frac{\log m}{m}$  as in Abbe *et al.* [1]. Note that here we assume a constant number of clusters  $k$  and therefore  $n \rightarrow \infty$  implies that  $m \rightarrow \infty$ . In this regime,  $\|A - M\|_2 \lesssim \sqrt{\log m}$  with probability tending to 1 as  $m \rightarrow \infty$ .

**THEOREM 3** There exists a universal constant  $C$  such that

$$P \left\{ \|A - M\|_2 \geq (3\sqrt{2}\sqrt{\alpha + (k-1)\beta} + C)\sqrt{\log m} \right\} \rightarrow 0$$

as  $n \rightarrow \infty$ .

*Proof.* The result is a direct consequence of Corollary 3.12 and Remark 3.13 of Bandeira & Van Handel [8] applied to  $A - M$ . Using the notation of that reference,  $C$  is selected such that  $C^2 > \tilde{c}_{1/2}$ .  $\square$

**COROLLARY 1** There exists a constant depending only on  $\alpha$  and  $\beta$ , denoted  $C_{\alpha,\beta}$ , such that

$$P \left\{ \|WW^T - VV^T\|_2 \geq \frac{C_{\alpha,\beta}}{(\alpha(1 - 1/m) - \beta)\sqrt{\log m}} \right\} \rightarrow 0$$

as  $n \rightarrow \infty$ .

*Proof.* The result is obtained by combining Theorem 3 with (2.6) and letting

$$C_{\alpha,\beta} = 3\sqrt{2}\sqrt{\alpha + (k-1)\beta} + C.$$

$\square$

Corollary 1 gives an upper bound on  $\|WW^T - VV^T\|_2$  that holds with high probability and decays asymptotically as  $\frac{1}{\sqrt{\log m}}$ . It is important to note that this rate of decay is not sufficient to satisfy the assumptions required for Lemmas 1 and 2. Nevertheless, as evidenced by our numerical results our algorithms perform quite well in this sparse regime. In fact, under the assumption that our algorithm correctly identifies one point per cluster, our results say that the average error  $\frac{1}{n}\|W\hat{\Pi} - VU\|_F$  decays like  $\frac{1}{\sqrt{n \log n}}$ . We elaborate on this in Section 4, where we discuss possible extensions to the SBM analysis.

## 2.5 Connections to OCS

The preceding theory may also be connected to OCS. In particular, we may view the collection of orthogonal cone centers  $\{q_j\}_{j=1}^k$  as an arbitrary rotation of the canonical basis vectors. This means that the indicator matrix  $W$  is also relevant for the OCS model. However, now we need to assume that a fraction  $1 - \eta$  of the rows of  $W$  are exactly zero, corresponding to nodes that do not belong to any cone. In practice, this motivates a potential modification of our algorithm to avoid assigning a cluster to any column of  $|U^T V_K^T|$  where all of the entries are uniformly small.

In the OCS framework, the prior theory carries through in a similar manner with a few small modifications we discuss here. In particular, Lemmas 1 and 2 must be modified to ensure that one column per cone is selected. We accomplish this in the following lemma by using the same conditioning argument as before and assuming that  $\mu$  and  $\delta$  are sufficiently small.

**LEMMA 3** Let  $V \in \mathbf{R}^{n \times k}$  have orthonormal columns with the columns  $\{v_i\}_{i=1}^n$  of  $V^T$  exhibiting OCS, specifically  $\{v_i\}_{i=1}^n \in \text{OCS}(\eta, \mu, \delta)$  with cone centers  $\{q_i\}_{i=1}^k$ . Furthermore, let  $c_M \equiv \max_{i \in [n]} \|v_i\|_2$  and assume that

$$\delta c_M < \frac{2^{-k}}{\sqrt{n}} \quad \text{and} \quad \mu c_M^2 < \frac{2^{-2k-4}}{n}.$$

If  $V^T \Pi = QR$  is a CPQR factorization and  $\mathcal{C}$  denotes the original indices of the first  $k$  columns selected by the permutation  $\Pi$ , then for each  $j \in [k]$  the matrix  $(V^T)_{:, \mathcal{C}}$  has exactly one column contained in  $\mathcal{K}_\mu(q_j)$ .

*Proof.* As before, the smallest singular value of  $(V^T)_{:, \mathcal{C}}$  is bounded from below by  $\frac{2^{-k}}{\sqrt{n}}$ . This precludes any column of  $(V^T)_{:, \mathcal{C}}$  from having norm less than  $\frac{2^{-k}}{\sqrt{n}}$  since  $c_M < 1$ . In particular, this implies that no column is outside the cones  $\mathcal{K}_\mu(q_j)$  from the definition of OCS.

It remains to prove that no two columns of  $(V^T)_{:, \mathcal{C}}$  come from the same cone. We proceed by contradiction. Assume  $i_1, i_2 \in \mathcal{C}$  and  $v_{i_1}, v_{i_2} \in \mathcal{K}_\mu(q_j)$  for some  $j$ . Using our upper bound on  $\mu c_M$  we may conclude that

$$\|v_{i_1} - q_j q_j^T v_{i_1}\|_2 \leq \frac{2^{-k-2}}{\sqrt{n}}$$

and the same inequality holds for  $v_{i_2}$ . Therefore, there exists a perturbation of  $(V^T)_{:, \mathcal{C}}$  with  $\ell_2$  norm bounded by  $\frac{2^{-k}}{\sqrt{n}}$  that makes two columns co-linear, which contradicts our lower bound on the smallest singular value. Finally, since  $|\mathcal{C}| = k$  the result follows.  $\square$

Asserting that we can select one point per cone via Lemma 3 yields the following recovery theorem.

**THEOREM 4** Let  $V \in \mathbf{R}^{n \times k}$  have orthonormal columns, and assume the columns of  $V^T$ , denoted  $\{v_i\}_{i=1}^n \in \text{OCS}(\eta, \mu, \delta)$  with cone centers  $\{q_i\}_{i=1}^k$ . Furthermore, let  $c_M \equiv \max_{i \in [n]} \|v_i\|_2$  and assume that

$$\delta c_M < \frac{2^{-k}}{\sqrt{n}} \quad \text{and} \quad \mu c_M^2 < \frac{2^{-2k-4}}{n}.$$

If  $U$  is computed by Algorithm 1 with  $V$  as input, then there exists a permutation matrix  $\widehat{\Pi}$  such that  $U$  satisfies

$$\frac{\|\widehat{\Pi} U^T v_i - Q^T v_i\|_2}{\|v_i\|_2} \leq \frac{(2 + \sqrt{2})k^{3/2} c_M \mu}{c_m} \quad \forall i \in [n],$$

where the columns of  $Q$  are the underlying cone centers  $\{q_i\}_{i=1}^k$  and  $c_m \equiv \min_{j \in \mathcal{C}} \|v_j\|_2$ .

*Proof.* Based on Lemma 3 there exists a permutation  $\widehat{\Pi}$  such that

$$\|(V^T)_{:, \mathcal{C}} \widehat{\Pi} - QD\|_F \leq \sqrt{2k} c_M \mu,$$

where  $D$  is a diagonal matrix and  $D_{ii}$  is equal to the two norm of the  $i^{\text{th}}$  column of  $(V^T)_{:, \mathcal{C}}$ . Equivalently we may assert that

$$\|(V^T)_{:, \mathcal{C}} - Q\widehat{\Pi}^T \widehat{\Pi} D \widehat{\Pi}^T\|_F \leq \sqrt{2k} c_M \mu.$$

Thinking of  $(Q\widehat{\Pi}^T)(\widehat{\Pi} D \widehat{\Pi}^T)$  as the polar factorization of some matrix divided into its orthogonal and positive definite parts, we use perturbation theory for polar factorizations [26] to bound the distance between  $U$  and  $Q\widehat{\Pi}^T$  as

$$\|U - Q\widehat{\Pi}^T\|_F \leq \frac{(2 + \sqrt{2})k^{3/2} c_M \mu}{c_m}.$$

Using

$$\|\widehat{\Pi}U^T v_i - Q^T v_i\|_2 \leq \|U - Q\widehat{\Pi}\|_F \|v_i\|_2$$

allows us to conclude the desired result.  $\square$

This result implies that if  $\mu$  and  $\delta$  are small enough our algorithm can correctly assign the  $\eta n$  nodes that belong to clusters, up to an arbitrary labeling encoded in  $\widehat{\Pi}$ . While our CPQR analysis yields a lower bound on  $\min_{j \in \mathcal{C}} \|v_j\|_2$  in terms of  $k$  and  $n$  it is often pessimistic in practice, so we leave our result in terms of the smallest column norm. Adding some assumptions about the total mass of nodes per cone allows Lemma 3 and Theorem 4 to be modified in a similar manner as before to address the randomized variant of our algorithm.

Ultimately, these results are somewhat unwieldy and appear pessimistic when compared with the observed performance of our algorithm. It is also difficult to assert that a graph's eigenvectors will obey such structure (especially if we require small  $\delta$  and  $\mu$ ). Later, we demonstrate the behavior of our algorithm on real graphs and its favorable performance when compared with standard spectral clustering methods.

## 2.6 Connected components

A limiting, and somewhat cleaner, scenario of the graph clustering problem is the graph partitioning problem. Here, we are given a graph that is comprised of  $k$  disjoint connected components. A relatively simple problem is to partition the graph into these  $k$  components. It turns out, given any orthonormal basis for the  $k$ -dimensional eigenspace associated with the zero eigenvalue of the normalized Laplacian Algorithm 1 exactly recovers the partition (with high probability in the case of Algorithm 2).<sup>1</sup> While there are many algorithms for finding connected components, we find this to be an interesting property of our algorithm.

Given a matrix  $W \in \mathbf{R}^{n \times k}$  with orthonormal columns and at most one non-zero in each row, let  $V \in \mathbf{R}^{n \times k}$  differ from  $W$  by some rotation  $Z \in \mathbf{O}_k$ . In this case, it is simple to modify the analysis of Section 2.3 to show that our algorithm applied to  $V$  will exactly recover  $W$  up to a permutation of the columns.

A more interesting observation is that given any  $\tilde{k} < k$  dimensional subspace of the  $k$ -dimensional subspace associated with the zero eigenvalue of the normalized Laplacian Algorithm 1 will partition the nodes into  $\tilde{k}$  connected components. We prove this by showing that our algorithm necessarily places each connected component in the same cluster by virtue of its assignment step.

**THEOREM 5** Let  $V \in \mathbf{R}^{n \times \tilde{k}}$  have orthonormal columns, let  $W \in \mathbf{R}^{n \times k}$  have orthonormal columns with exactly one non-zero per row and assume  $\tilde{k} \leq k$ . If  $V = WZ$  for some  $Z \in \mathbf{R}^{k \times \tilde{k}}$  with orthonormal columns, then Algorithm 1 partitions  $[n]$  into  $\tilde{k}$  clusters in a manner such that no two clusters contain rows of  $W$  with the same sparsity pattern.

*Proof.* Let  $\mathcal{S}_i \subset [n]$  denote the support of column  $i$  of  $W$ . Using this notation, we see that the columns of  $(V^T)_{:, \mathcal{S}_i}$  are co-linear—they are all proportional to the  $i^{\text{th}}$  column of  $Z^T$ . This implies that, given any orthogonal matrix  $U$ , each column of  $|U^T(V^T)_{:, \mathcal{S}_i}|$  attains its maximum in the same row. Therefore, we see that each connected component is assigned to a single cluster using our algorithm.  $\square$

<sup>1</sup> Note that here, we use the normalized Laplacian since its eigenvectors contain the desired normalized indicator structure analogous to Definition 2.

**REMARK 4** In general the use of the CPQR should ensure that all of the output clusters will be non-empty, however, this depends on properties of  $Z$ . Intuitively, because  $U$  is the closest orthogonal matrix to  $k$  different well-conditioned columns of  $Z^T$ , at least one column of  $Z^T$  will align closely with one column of  $U$ .

Depending on how the problem is formulated this property may not be shared by `k-means`. In the presence of degree heterogeneity, given an initial guess that properly partitions the nodes into disconnected sets, `k-means` may move to a local minimum of its objective function that does not respect the proper partition. In this setting such a shortcoming may be easily fixed by normalizing the embedding points before running `k-means`. Nevertheless, we believe this simple conceptual experiment provides additional motivation for considering angular relationships between embedded nodes rather than a distance-based clustering.

### 3. Numerical results

We now present numerical simulations to validate the performance of our algorithm through examination of its behavior for multi-way spectral clustering. All of our experiments were conducted using MATLAB<sup>®</sup> and the included `k-means++` implementation was used as a point of comparison. If `k-means` failed to converge after 100 iterations (the default maximum) we simply took the output as is. Code implementing our methodology and these experiments may be found at <https://github.com/asdamle/QR-spectral-clustering>.

#### 3.1 SBM

We first consider the SBM with  $k = 9$  equisized clusters and show via simulation that our algorithm, in contrast to `k-means++`, recovers the phase transition behavior near the information theoretic threshold [1,3,4,24]. We also demonstrate similar phase transition behavior when there are  $k = 7$  unequized clusters.

For our tests with the SBM we compare the behavior of three different clustering schemes: our algorithm, `k-means` using the columns  $\mathcal{C}$  from our algorithm as an initial guess for cluster centers and `k-means++` itself. We exclusively utilize Algorithm 2 with an oversampling factor of  $\gamma = 5$ . For more general problems, the non-random method may perform better, however, in this context they performed almost identically, and hence the deterministic results are omitted.

Interestingly, in the regime where  $p$  and  $q$  are both  $\Theta(\log n/n)$  we do not necessarily expect rapid enough asymptotic concentration of  $A$  for our results to theoretically guarantee recovery to the information theoretic limit. Nevertheless, we do observe good behavior of our algorithm when applied in this setting and, as we will see, the same behavior is not observed with `k-means++`.

One major avenue of work in recent years has been in the area of SDP relaxation for clustering [1,3,25]. Broadly speaking, such relaxations recast (1.1) in terms of  $XX^T$  and then relax  $XX^T$  to a semidefinite matrix  $Z$ . These SDP relaxations often enjoy strong consistency results on recovery down to the information theoretic limit in the case of the SBM, in which setting the optimal solution  $Z_*$  can be used to recover the true clusters exactly with high probability. However, these algorithms become computationally intractable as the size of the graph grows.

**3.1.1 Equisized clusters** First, we consider  $k = 9$  equisized clusters each with size  $m = 150$  (i.e.,  $n = 1350$ ) and within-cluster and between-cluster connection probabilities  $p = \alpha \log m/m$  and  $q = \beta \log m/m$ , respectively. Defining a grid of 80 equispaced values of  $\alpha$  and 40 equispaced values



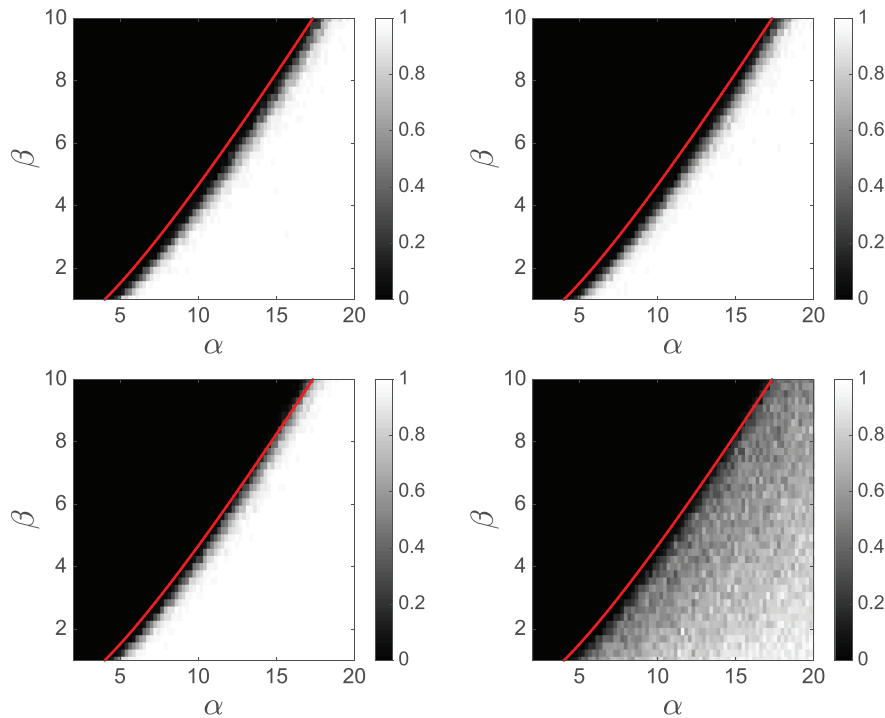


FIG. 2. The phase plots here show the fraction of trials exhibiting exact recovery in the case of equisized clusters, and the asymptotic transition curve  $\sqrt{\alpha} - \sqrt{\beta} = 1$  is given by the overlaid red line (color online). The top row of phase plots corresponds to Algorithm 2 applied to eigenvectors  $V_k$  of  $A$  (top left) and of  $A_N$  (top right). We also give results based on seeding  $k$ -means with the clustering  $\mathcal{C}$  from our algorithm applied to eigenvectors from  $A_N$  (bottom left) and results for vanilla  $k$ -means++ (bottom right).

of  $\beta$ , we generated 50 instances of the SBM for each  $(\alpha, \beta)$  pair (redrawing the adjacency matrix if it was disconnected). Then, for both the adjacency matrix  $A$  and the degree-normalized adjacency matrix  $A_N$  with entries  $(A_N)_{ij} \equiv A_{ij}/\sqrt{d_i d_j}$ , where  $d_i$  is the degree of node  $i$ , we computed the top nine eigenvectors and used them as input to each of the three algorithms mentioned before. While for degree-regular graphs we do not expect to observe differences in the results between  $A$  and  $A_N$ , when applied to graphs not coming from the SBM or that are not as degree regular, we anticipate better performance from degree normalization.

In the plots of Fig. 2 we color each point according to the fraction of trials resulting in successful recovery for each  $(\alpha, \beta)$  pair, i.e., the fraction of trials in which all nodes were correctly clustered. These phase diagrams show the primary advantage of our algorithm over  $k$ -means++: robust recovery (where possible). Whether used to explicitly compute the clustering directly, or to seed  $k$ -means, we cleanly recover a sharp phase transition that hews to the theory line. In contrast,  $k$ -means++ fails to exactly recover the clusters a substantial portion of the time, even far from the phase transition.

Theoretically, if we scale  $k$  as  $k = o(\log m)$  and take the asymptotic limit as  $m \rightarrow \infty$ , then a sharp phase transition between where recovery is possible almost surely and where it does not occur at the threshold curve  $\sqrt{\alpha} - \sqrt{\beta} = 1$  [4]. (Note that similar results exist elsewhere [24], albeit with a slightly different definition of  $\alpha$  and  $\beta$ .) In Fig. 2 we observe that our phase transition does not quite match

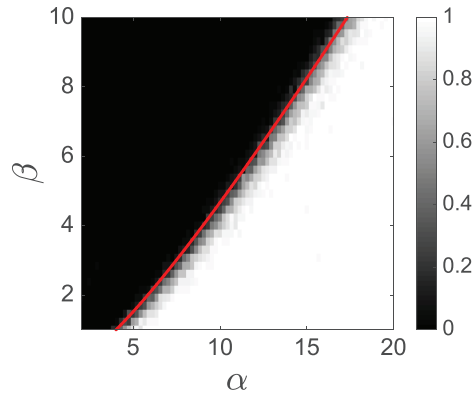


Fig. 3. The phase plot shows the fraction of trials (for equisized clusters) where application of our algorithm to eigenvectors  $V_k$  of  $A$  yields a multi-way cut metric (3.1) less than or equal to that of the true underlying clusters. The asymptotic transition curve  $\sqrt{\alpha} - \sqrt{\beta} = 1$  is given by the overlaid red line (color online).

the asymptotically expected behavior. However, based on results for the two-block case [1], various arguments [4] show that we expect the location of the recovery threshold in the finite case to deviate from the asymptotic threshold with error that decays only very slowly, i.e.,  $\Omega(1/\log m)$ .

To further explore the discrepancy between our algorithms' performance and the asymptotic theory, we consider a slightly different criteria than exact recovery. A natural metric for a good graph clustering is to minimize the number of edges between clusters while normalizing for cluster size. Specifically, we define the multi-way cut metric for a  $k$ -way partition of nodes into non-empty sets  $S_1, S_2, \dots, S_k$  as

$$\max_{i=1, \dots, k} \frac{\#\{\text{Edges between } S_i \text{ and } \overline{S_i}\}}{|S_i|}. \quad (3.1)$$

Figure 3 shows the fraction of times that our algorithm yields as good or better a multi-way cut metric than the true clustering. Near the threshold we may not be recovering the underlying clustering, but perhaps we should not expect to since we are often finding a better clustering under a slightly different metric.

**3.1.2 Unequisized clusters** Secondly, we consider the case where there are  $k = 7$  *unequisized* clusters with sizes  $m = 70, 80, 90, 100, 110, 120$  and  $130$ . In this case we define the within- and between-cluster connection probabilities as  $p = \alpha \log 70/70$  and  $q = \beta \log 70/70$ . As before, we test the algorithms for exact recovery and present the phase plots in Fig. 4. As before, our algorithm shows a sharp phase transition, whereas  $k$ -means++ does not. Such behavior is still expected [3], though the characterization of its location becomes significantly more complicated. Importantly, our algorithm seamlessly deals with unknown cluster sizes.

### 3.2 A real-world graph

We complement our experiments using the SBM by demonstrating the application of our algorithm to a real-world graph and comparing its performance with that of  $k$ -means++. To measure performance we consider both the  $k$ -means objective function—the sum of squared distances from points to their

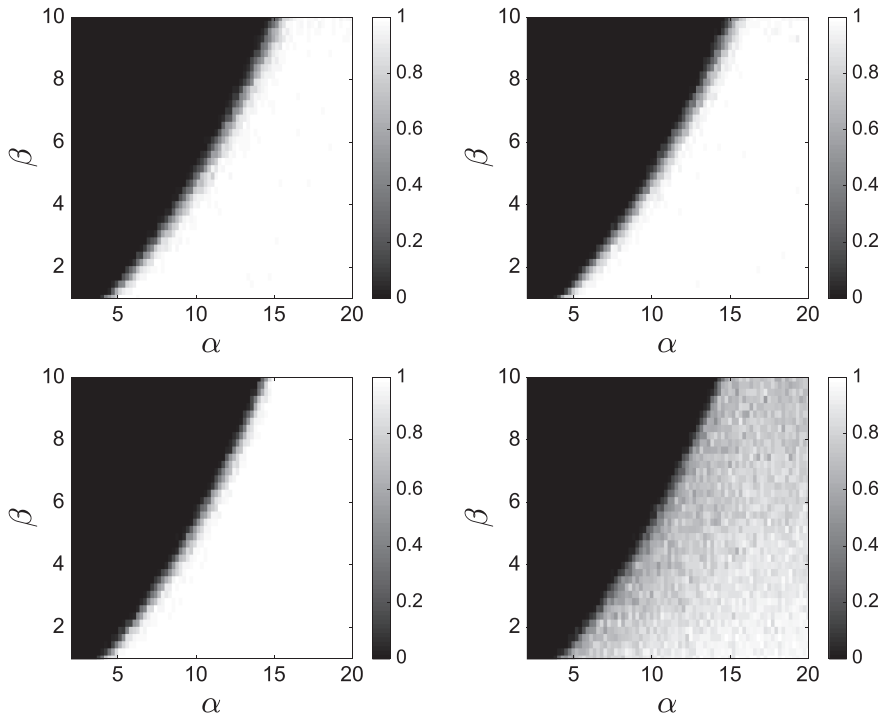


FIG. 4. The phase plots here show the fraction of trials exhibiting exact recovery in the case of unequized clusters. The top row of phase plots corresponds to Algorithm 2 applied to eigenvectors  $V_k$  of  $A$  (top left) and of  $A_N$  (top right). We also give results based on seeding  $k$ -means with the clustering  $\mathcal{E}$  from our algorithm applied to eigenvectors from  $A_N$  (bottom left) and results for vanilla  $k$ -means++ (bottom right).

cluster centers—and the multi-way cut metric (3.1). Since it is a metric on the graph itself (as opposed to the embedding), minimization of the multi-way cut metric over possible partitions is the more interpretable metric. Furthermore, as we will observe, a smaller  $k$ -means objective value does not necessarily translate into a smaller multi-way cut metric.

The graph we consider is the collaboration network for the arXiv Astrophysics category [30]. This graph consists of 18,772 nodes and 198,110 undirected edges corresponding to authors and co-authorship, respectively. The graph is partitioned into 290 connected components, the largest of which contains 17,903 nodes. For all of the experiments conducted here, we used  $k$  eigenvectors corresponding to the  $k$  largest eigenvalues of a degree-normalized adjacency matrix  $A_N$  as in Section 3.1.1.

We begin with a simple experiment conducted using the whole graph. Because the graph has 290 connected components, it is possible to partition the nodes into  $k$  disconnected subgraphs using any subspace of the invariant space associated with the first 290 eigenvalues. As outlined in Theorem 5 our algorithm theoretically accomplishes this task without fail. Our experiments validate this fact, looking for a 10-way partition our deterministic algorithm achieved a multi-way cut metric of zero. In contrast, over 50 trials the smallest multi-way cut metric found using  $k$ -means++ was 0.08. Seeding  $k$ -means using our clustering resulted in multi-way cut metric of 11.03, though the  $k$ -means objective was decreased from 6.31 to 4.69. In particular, this demonstrates that the  $k$ -means objective function is not necessarily a good proxy for the multi-way cut metric.

TABLE 1 Comparison of deterministic CPQR-based clustering and  $k$ -means++

|                                    | $k$ -means objective | multi-way cut (3.1) |
|------------------------------------|----------------------|---------------------|
| $k$ -means++ mean                  | 1.36                 | 8.48                |
| $k$ -means++ median                | 1.46                 | 10.21               |
| $k$ -means++ minimum               | 0.76                 | 1.86                |
| $k$ -means++ maximum               | 2.52                 | 42.03               |
| Algorithm 1                        | 2.52                 | 1.92                |
| $k$ -means seeded with Algorithm 1 | 0.76                 | 1.86                |

Next, we take the largest connected component of the graph and seek to partition it six ways. Admittedly, as with many real-world graphs it is not an easy problem to determine a value for  $k$  that leads to a natural clustering. Nevertheless, here we choose  $k = 6$  and note that there is a slightly larger gap in the spectrum between the sixth and seventh eigenvalues than others in the surrounding area.

Table 1 summarizes the results comparing Algorithm 1 with 50 trials of  $k$ -means++. While our algorithm does not find the best possible multi-way cut, it gets close in a deterministic manner and serves as a very good seeding for  $k$ -means. Interestingly, this is accomplished with complete disregard for the  $k$ -means objective function: our algorithm results in a sum of squared distances larger than any of the local minima found by  $k$ -means++. However, by seeding  $k$ -means with the clustering performed by our algorithm we find as good a local minima of the  $k$ -means objective as any found by  $k$ -means++.

#### 4. Open problems

While the development of our algorithms is motivated by several theoretical considerations and their efficacy is demonstrated on several well-understood model problems, there is a gap between the theoretical performance that can be guaranteed and that which is observed. We highlight some relevant open questions.

**Entry-wise perturbation results for eigenvectors** To obtain provably good clusterings with Algorithms 1 and 2, Theorems 1 and 2 require stringent conditions on the distance between the subspaces  $V$  and  $W$ . In fact, a tighter *entry-wise* error bound on  $WW^T - VV^T$  would allow for the development of stronger results for our algorithm. An example of this is clustering the SBM, where the performance of our algorithm exceeds that guaranteed by our theorems (which assume small subspace distance). In particular, Lemmas 1 and 2 are used to ensure that one column associated with each cluster is found, which can be refined with entry-wise bounds. We do not know of any sufficiently strong results in the SBM literature (though substantial recent progress has been made by, e.g., Abbe *et al.* [2]) and are working to utilize and further develop such bounds.

**Pervasiveness of OCS** Under the assumption that a set of points exhibits OCS, our clustering algorithms have strong theoretical motivation and guarantees. While the existence of this structure has been identified in certain settings (e.g., Schiebinger *et al.* [40]), exploring the pervasiveness of this phenomenon for more general point sets is an important area of ongoing research. For example, while our analysis of graph clustering is tailored to the SBM, OCS may prove fruitful as a lens for analyzing the applicability of our methods to more general graph families [7,29].

## 5. Discussion and conclusion

We have presented a new efficient (particularly the randomized variant) algorithm for spectral clustering of graphs with community structure. In contrast to the traditionally used  $k$ -means algorithm, our method requires no initial guess for cluster centers and achieves the theoretically expected recovery results for the SBM. Given that a bad initial guess can mean the  $k$ -means algorithm does not achieve the desired result, this is a particularly important feature. Furthermore, we can always use our algorithm to generate an initial seed for  $k$ -means and observe in our experiments that this can provide small gains in recovery near the phase transition boundary. When considering a real-world graph, our algorithm compares favorably as a means for determining clusters that achieve a small multi-way cut metric.

Recent results have yielded important understanding of what can be possible for recovery in the SBM. However, the SDP-based methods that achieve these results do not scale to large problems of practical interest. Conversely, the traditionally used  $k$ -means algorithm, while scalable to large problems, fails to achieve the best possible behavior on these small scale SBMs due to its dependence on an initial guess. Our algorithm is both scalable to large problems and matches the behavior of SDPs on smaller model problems. These two properties make it attractive for general use.

Here we have explored the behavior of our algorithm on the SBM and provided theoretical justification for its use. To motivate its use for more general problems, we have discussed its connections to the more broadly applicable OCS of clusters arising naturally in various problems. Furthermore, Theorems 1 and 2 may be extensible to cluster indicators with a more general structure, though at the expense of weaker results. We intend to further explore the behavior of our algorithms on a wider range of real-world graphs in future work.

## Acknowledgements

The authors thank Austin Benson, Sven Schmit, Nolan Skochdopole and Yuekai Sun for useful discussion, Stanford University and the Stanford Research Computing Center for providing computational resources and support that have contributed to these research results, and the anonymous reviewers for their many insightful and helpful comments.

## Funding

National Science Foundation [DMS-1606277 to A.D.]; Stanford [Stanford Graduate Fellowship to V.M.]; and the United States Department of Energy [DE-FG02-97ER25308 to V.M., DE-FC02-13ER26134 to L.Y. and DE-SC0009409 to L.Y.].

## REFERENCES

1. ABBE, E., BANDEIRA, A. S. & HALL, G. (2016) Exact recovery in the Stochastic Block Model. *IEEE Trans. Inf. Theory*, **62**, 471–487.
2. ABBE, E., FAN, J., WANG, K. & ZHONG, Y. (2017) Entrywise eigenvector analysis of random matrices with low expected rank. arXiv preprint arXiv:1709.09565.
3. ABBE, E. & SANDON, C. (2015) Community detection in general stochastic block models: fundamental limits and efficient algorithms for recovery. In *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS) 2015*, pp. 670–688.
4. AGARWAL, N., BANDEIRA, A. S., KOILIARIS, K. & KOLLA, A. (2015) Multisection in the stochastic block model using semidefinite programming. arXiv preprint arXiv:1507.02323.

5. ANDERSON, E., BAI, Z., BISCHOF, C., BLACKFORD, S., DEMMEL, J., DONGARRA, J., DU CROZ, J., GREENBAUM, A., HAMMARLING, S., MCKENNEY, A. & SORENSEN, D. (1999) *LAPACK Users' Guide*, 3rd edn. Philadelphia, PA: SIAM.
6. ARTHUR, D. & VASSILVITSKII, S. (2007) k-means++: The advantages of careful seeding. *Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA: SIAM, pp. 1027–1035.
7. ATHREYA, A., FISHKIND, D. E., LEVIN, K., LYZINSKI, V., PARK, Y., QIN, Y., SUSSMAN, D. L., TANG, M., VOGELSTEIN, J. T. & PRIEBE, C. E. (2017) *Statistical inference on random dot product graphs: a survey*. ArXiv e-prints.
8. BANDEIRA, A. S. & VAN HANDEL, R. (2016) Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *Ann. Probab.*, **44**, 2479–2506.
9. BENZI, M., BOITO, P. & RAZOUK, N. (2013) Decay properties of spectral projectors with applications to electronic structure. *SIAM Review*, **55**, 3–64.
10. BUSINGER, P. & GOLUB, G. H. (1965) Linear least squares solutions by Householder transformations. *Numer. Math.*, **7**, 269–276.
11. CELEBI, M. E., KINGRAVI, H. A. & VELA, P. A. (2013) A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl.*, **40**, 200–210.
12. CHANDRASEKARAN, S. & IPSEN, I. C. (1994) On rank-revealing factorisations. *SIAM J. Matrix Anal. Appl.*, **15**, 592–622.
13. DAMLE, A., LIN, L. & YING, L. (2015) Compressed representation of Kohn–Sham orbitals via selected columns of the density matrix. *J. Chem. Theory Comput.*, **11**, 1463–1469, PMID:26574357.
14. DAMLE, A., LIN, L. & YING, L. (2017) Computing localized representations of the Kohn–Sham subspace via randomization and refinement. *SIAM J. Sci. Comput.*, **39**, B1178–B1198.
15. DAVIS, C. & KAHAN, W. M. (1970) The rotation of eigenvectors by a perturbation. III. *SIAM J. Numer. Anal.*, **7**, 1–46.
16. DONATH, W. E. & HOFFMAN, A. J. (1973) Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, **17**, 420–425.
17. FAN, K. & HOFFMAN, A. J. (1955) Some metric inequalities in the space of matrices. *Proc. Am. Math. Soc.*, **6**, 111–116.
18. FIEDLER, M. (1973) Algebraic connectivity of graphs. *Czechoslovak Math. J.*, **23**, 298–305.
19. FRIEZE, A., KANNAN, R. & VEMPALA, S. (2004) Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, **51**, 1025–1041.
20. GAO, C., MA, Z., ZHANG, A. Y. & ZHOU, H. H. (2015) *Achieving optimal misclassification proportion in stochastic block model*. arXiv preprint arXiv:1505.03772.
21. GHARAN, S. O. & TREVISAN, L. (2014) Partitioning into expanders. *Proc. of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA: SIAM, pp. 1256–1266.
22. GOLUB, G. H. & VAN LOAN, C. F. (1996) *Matrix Computations*, 3rd edn. Baltimore, MD, USA: Johns Hopkins University Press.
23. GU, M. & EISENSTAT, S. C. (1996) Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, **17**, 848–869.
24. HAJEK, B., WU, Y. & XU, J. (2015) Achieving exact cluster recovery threshold via semidefinite programming: extensions. arXiv preprint arXiv:1502.07738.
25. HAJEK, B., WU, Y. & XU, J. (2016) Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Trans. Inf. Theory*, **62**, 2788–2797.
26. HIGHAM, N. J. (1986) Computing the polar decomposition—with applications. *SIAM J. Sci. Stat. Comput.*, **7**, 1160–1174.
27. HOLLAND, P. W., LASKEY, K. B. & LEINHARDT, S. (1983) Stochastic blockmodels: first steps. *Soc. Networks*, **5**, 109–137.
28. KRZAKALA, F., MOORE, C., MOSSEL, E., NEEMAN, J., SLY, A., ZDEBOROVÁ, L. & ZHANG, P. (2013) Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci. USA*, **110**, 20935–20940.

29. LEE, J. R., GHARAN, S. O. & TREVISAN, L. (2014) Multiway spectral partitioning and higher-order Cheeger inequalities. *J. ACM*, **61**, 37:1–37:30.
30. LESKOVEC, J. & KREVL, A. (2014) *SNAP Datasets: Stanford Large Network Dataset Collection*. <http://snap.stanford.edu/data>.
31. LI, R.-C. (1995) New perturbation bounds for the unitary polar factor. *SIAM J. Matrix Anal. Appl.*, **16**, 327–332.
32. LLOYD, S. (1982) Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, **28**, 129–137.
33. MAHONEY, M. W. & DRINEAS, P. (2009) CUR matrix decompositions for improved data analysis. *Proc. Natl. Acad. Sci. USA*, **106**, 697–702.
34. MASSOULIÉ, L. (2014) Community detection thresholds and the weak Ramanujan property. *Proc. of the 46th Annual ACM Symposium on Theory of Computing*. New York, NY: ACM, pp. 694–703.
35. MONTANARI, A. & SEN, S. (2016) Semidefinite Programs on Sparse Random Graphs and Their Application to Community Detection. *Proc. of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*. New York, NY, USA: ACM, pp. 814–827.
36. MOTWANI, R. & RAGHAVAN, P. (2010) Randomized Algorithms. In *Algorithms and theory of computation handbook (2 ed.)*, Mikhail J. Atallah and Marina Blanton (Eds.), Boca Raton, FL: Chapman & Hall/CRC 12-12.
37. NG, A. Y., JORDAN, M. I. & WEISS, Y. (2001) On spectral clustering: analysis and an algorithm. *Adv. in Neural Info. Proc. Sys.* Cambridge, MA: MIT Press, pp. 849–856.
38. QUINTANA-ORTÍ, G., SUN, X. & BISCHOF, C. H. (1998) A BLAS-3 version of the QR factorization with column pivoting. *SIAM J. Sci. Comput.*, **19**, 1486–1494.
39. ROHE, K., CHATTERJEE, S. & YU, B. (2011) Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Stat.*, **39**, 1878–1915.
40. SCHIEBINGER, G., WAINWRIGHT, M. J. & YU, B. (2015) The geometry of kernelized spectral clustering. *Ann. Stat.*, **43**, 819–846.
41. TROPP, J. A. (2015) An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.*, **8**, 1–230.
42. VON LUXBURG, U. (2007) A tutorial on spectral clustering. *Stat. Comput.*, **17**, 395–416.
43. ZHA, H., HE, X., DING, C., GU, M. & SIMON, H. D. (2001) Spectral relaxation for k-means clustering. *Adv. in Neural Info. Proc. Sys.* MIT Press, pp. 1057–1064.