Combinatorics, Probability and Computing (2018) 27, 124–140.
doi:10.1017/S0963548317000463

A Simple SVD Algorithm for Finding Hidden Partitions

VAN VU

Department of Mathematics, Yale, New Haven, CT 06520, USA
(e-mail van.vu@yale.edu)

Received 17 April 2014; revised 21 November 2016; first published online 9 October 2017

Finding a hidden partition in a random environment is a general and important problem which contains as subproblems many important questions, such as finding a hidden clique, finding a hidden colouring, finding a hidden bipartition, *etc.*

In this paper we provide a simple SVD algorithm for this purpose, addressing a question of McSherry. This algorithm is easy to implement and works for sparse graphs under optimal density assumptions. We also consider an approximating algorithm, which on one hand works under very mild assumptions, but on other hand can sometimes be upgraded to give the exact solution.

2010 *Mathematics subject classification*: Primary 68Q87
Secondary 68W20, 60C05

1. The problem and a new algorithm

The Hidden Partition problem is the following. Let X be a set of n vertices with a partition $X = \cup_{i=1}^{k} X_i$; for all $1 \leqslant i \leqslant j \leqslant k$ and any $x \in X_i, y \in X_j$, put a random edge between x and y with probability p_{ij}. Given one such random graph, one has to recover the sets X_i. This problem is of importance in computer science and statistics, and contains as special cases several well-studied problems such as Hidden Clique, Hidden Bisection, Hidden Colouring, Clustering, *etc.* (see *e.g.* [1, 2, 3, 6, 7, 8, 11, 13, 14, 15, 16, 18, 21, 23, 24] and the references therein). In what follows, we refer to X_i as clusters.

In an influential paper [27], McSherry provided a (randomized) polynomial-time algorithm that solves the general Hidden Partition problem for a large range of parameters. As corollary, he derived several earlier results obtained for special cases. We refer the reader to this paper for a detailed discussion of results prior to [27].

The general idea of [27] (and many earlier works on clustering) is to find a good geometric representation of the vertices. We say that a representation is *perfect* if there is a number $r > 0$ such that:

- vertices in the same cluster have distance at most r from each other,
- vertices from different clusters have distance at least $4r$ from each other.

Once a perfect representation is obtained, it is easy to find the clusters. If r is known, then the solution is obvious. If r is not known, then there are several simple algorithms. For instance, one can create a minimal spanning tree (with respect to the distances) on the vertices and then remove the largest $k-1$ edges. In what follows, we put all these simple algorithms under a subroutine called *Clustering by Distances* and the reader can choose his/her favourite to implement.

In the rest of the paper, $s_u := |X_i|$ if $u \in X_i$ and $s := \min_{u \in X} s_u = \min_i |X_i|$. We assume that n is sufficiently large, whenever needed. Asymptotic notation is used under the assumption $n \to \infty$. All explicit constants (such as the 4 above) are *ad hoc* and we make no attempt to optimize them.

A popular way to find a perfect representation is to project the points of X (seen as vectors in $\mathbb{R}^n$) onto a properly chosen low-dimensional subspace. The main technical part of McSherry's algorithm is a subroutine called CProj (Combinatorial Projection), which creates this projection in a combinatorial way. The inputs in this subroutine are a matrix $\hat{A}$, parameters k, s, and a properly chosen threshold τ. For a matrix M, P_M denotes the orthogonal projection onto the column space of M, and M_v is the column indexed by v.

Algorithm 1: Combinatorial Projection (CProj)

(1) While there are at least $s/2$ unclassified nodes, choose an unclassified node v_i randomly and define $T_i := \{u | \|P_{\hat{A}^T}(\hat{A}^T_{v_i} - \hat{A}^T_u)\| \leqslant \tau\}$, where u ranges over the set of unclassified nodes. Mark each $u \in T_i$ as classified.
(2) Assign each remaining node to the T_i with the closest projected v_i.
(3) Let $\hat{c}_i$ be the characteristic vector of T_i.
(4) Return $P_{\hat{c}}$, the orthogonal projection matrix onto the span of the $\hat{c}_i$.

Algorithm 2: McSherry's algorithm

(1) Randomly partition the set $\{1, \ldots, n\}$ into two parts A and B. Let $\hat{A}, \hat{B}$ be the submatrices of the adjacency matrix formed by columns from A and B. (One next uses these two matrices to produce two projections using CProj, thinking of their columns as nodes.)
(2) Let $P_1 = \text{CProj}(\hat{B}), P_2 = \text{CProj}(\hat{A})$ and compute $\hat{H} = [P_1(\hat{A})|P_2(\hat{B})]$.
(3) Run *Clustering by Distances* on the projected points.

For more details about this algorithm (such as how the parameters are chosen) we refer the reader to [27].

Let P be the probability matrix $(p_{ij})_{1 \leqslant i,j \leqslant k}$. For a vertex $u \in X$, $\mathbf{u}$ denotes the corresponding column in P. Define

$$\Delta := \min \|\mathbf{u} - \mathbf{v}\|_2,$$

where the minimum is taken over all pairs u, v belonging to different clusters. Furthermore, define

$$\sigma^2 := \max_{i,j} p_{ij}(1 - p_{ij}).$$

McSherry proved the following theorem [27].

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

Theorem 1.1. *There is a constant $C > 0$ such that if $\sigma^2 \gg \log^6 n/n$ and*

$$\Delta \geqslant C\sigma k^{1/2}\left(\sqrt{\frac{n}{s}} + \sqrt{\log \frac{n}{\varepsilon}}\right), \tag{1.1}$$

the above algorithm (with a proper choice of the threshold τ) recovers the partition with probability $1-\varepsilon$ with respect to the random graph and k^{-1} with respect to the auxiliary random bits.

In this paper, we present a new spectral algorithm for the problem. The key features of this algorithm are as follows.

- It works under an optimal density assumption and a different condition on Δ.
- Both the algorithm and the analysis are simple (the proof is only a few pages).
- The algorithm is easy to implement. Its main operation is to compute the leading few eigenvectors of a matrix, a task for which many software packages are available.
- It addresses a question raised by McSherry [27] about the possibility of avoiding the combinatorial subroutine CProj.

We pushed the bound $\sigma^2 \gg \log^6 n/n$ to $\sigma^2 \gg \log n/n$, which is optimal. In fact, in certain settings, one can go below this density; see the discussion below and also [9], where the ideas introduced here are further developed to deal with very sparse graphs. The key technical ingredient in our analysis is Lemma 2.1, concerning the magnitude of the orthogonal projection of a random vector onto a deterministic subspace. This lemma seems to have a wide range of potential applications.

As we focus on complete recovery, the density bound $\log n/n$ is necessary. If one's goal is to obtain an approximate recovery, then there are many earlier works considering density as small as c/n, which we are going to discuss in the paper. We are going to discuss approximate recovery in Theorem 1.3 below.

Let us mention an essential point that for approximation we can work with assumptions much weaker than those needed for full recovery (in both McSherry's algorithm and our algorithm). Furthermore, in certain settings, one can upgrade an approximate solution to an exact one using an extra (fast) subroutine. Thus, in these cases, we obtain a new algorithm for full recovery under weaker assumptions.

To this end, M_k denotes the subspace spanned by the first k left singular vectors of a matrix M. Let $\hat{P}$ be our input, namely the adjacency matrix of a random graph generated by P. Arguably, the most natural choice for H, the subspace we would like to project on, is $\hat{P}_k$ (SVD), which leads to Algorithm 3.

Algorithm 3: SVD I

(1) Project the columns of $\hat{P}$ onto $\hat{P}_k$.
(2) Run *Clustering by Distances* on the projected points.

While SVD I could well win the contest for being the simplest algorithm, and perhaps the first one that most practitioners of the spectral method would think of, it is hard to analyse in the general case. In what follows, we analyse a slightly more technical alternative, SVD II (Algorithm 4), which is a variant of an algorithm proposed in [27, Section 1].

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

Algorithm 4: SVD II

(0) Randomly partition X into two subsets Y and Z. Let B be the adjacency matrix of the bipartite graph between Y and Z. Let Y_1 be a random subset of Y obtained by selecting each element with probability $1/2$ independently, and let $\hat{A}$ be the submatrix of B formed by the columns indexed by Y_1.
(1) Project the columns of B indexed by $Y_2 := Y \backslash Y_1$ on $\hat{A}_k$.
(2) Run *Clustering by Distances* on the projected points.

Compared to SVD I, the extra steps in SVD II are the random partitions in step (0), done in order to reduce correlation. This is not entirely unexpected. A careful reading of [27] reveals that one also needs an extra partition in Algorithm 2 to make the analysis go through; in particular, the proof of [27, Theorem 12] needs a further refinement to be complete, since $\hat{A}$ and $\hat{B}$ are not independent.

For simplicity, we assume that P has rank k. If the rank is $k' < k$, then in step (1) we project onto $\hat{A}_{k'}$; the analysis remains the same.

Notice that SVD II gives a partition of Y_2, not X. There are many ways to extend it to a partition of X. For instance, we can run the algorithm l times (for some small l) and find partitions of $Y_2^1, \ldots, Y_2^l$, where Y_2^i are random subsets of X with density $1/4$ (the input graph is the same, only the random partitions are different). If a cluster C in Y_2^i and a cluster C' in $Y_2^{i'}$ intersect, then they must belong to the same cluster in X and we can merge them. If we choose $l = 3\log n$, say, then with probability $1 - o(n^{-1})$, all vertices of X must belong to some Y_2^i, and we recover the clusters $X_1, \ldots, X_k$ at the end. We omit the details of this merging part.

Let us now analyse SVD II. Let $\lambda_1(P) \geqslant \cdots \geqslant \lambda_k(P) := \lambda$ be the non-trivial singular values of P. In particular $\lambda := \lambda_k(P)$ is the least singular value of P.

Theorem 1.2. *There is a constant $C > 0$ such that the following holds. Assume that $\sigma^2 \geqslant C\log n/n$ and $s \geqslant C\log n, k = o((n/\log n)^{1/2})$. Then SVD II clusters Y_2 correctly with probability $1 - o(n^{-1})$ if one of the following two conditions is satisfied.*

- *Condition 1:*

$$\Delta \geqslant C\left(\sigma\sqrt{\frac{n}{s}} + \sqrt{\log n}\right).$$

- *Condition 2:*

$$\Delta \geqslant C\left(\sigma\sqrt{\frac{n}{s}} + \sqrt{k}\left(\sigma\sqrt{\log n} + \frac{\log n}{\sqrt{s}} + \frac{\sigma\sqrt{n}\log n}{\lambda}\right)\right).$$

If we omit the assumption $s \geqslant C\log n$, the statement still holds but with probability

$$1 - o(n^{-1}) - c\sum_{i=1}^{k} e^{-|X_i|/c}$$

for some constant c.

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

The conditions on Δ in Theorems 1.1 and 1.2 are incomparable (see also Theorem 1.5 below for a comparison). The lower bound $\sigma^2 \geqslant C \log n/n$ is optimal, up to the value of C. If $\sigma^2 < (1-\varepsilon)\log n/n$, then with high probability there are many isolated points which can be assigned to any cluster. On the other hand, if one's goal is to find an optimal solution (regardless of whether it comes from the hidden structures), then one can go below $\log n/n$; see for instance [1, 10]. We can reduce the failure probability $o(n^{-1})$ to $o(n^{-K})$ for any constant K at the cost of increasing the constant C.

In practice, one is often satisfied with an *approximate solution*. We say that a partition $X = \cup_{i=1}^k X_i'$ is ε-correct if $|X_i \backslash X_i'| \leqslant \varepsilon |X_i|$. Similarly, we say that a geometric representation of X is ε-perfect if there are points $x_1, \ldots, x_k$ with distance at least $8r$ from each other, so that at least $(1-\varepsilon)|X_i|$ points from X_i has distance at most r to x_i.

Theorem 1.3. *Given $\varepsilon > 0$, there is a constant $C > 0$ such that the following holds. If $\sigma^2 \geqslant C \log n/n, s \geqslant C \log n$ and*

$$\Delta \geqslant C\sigma \sqrt{\frac{n}{s}},$$

then with probability at least $1-\varepsilon$ the projection in SVD II produces an $(1-\varepsilon)$-perfect representation of the points in Y_2.

We say that $X_1, \ldots, X_k$ are γ-balanced if $|X_i| \leqslant (1+\gamma)s$ where $s = \min_j |X_j|$.

Lemma 1.4. *For arbitrary positive constants ε, γ, k, let $\delta := \varepsilon/(k+(k-1)(1+\gamma))$. Given an δ-perfect representation of γ-balanced sets $X_1, \ldots, X_k$, we can find an ε-correct partition by a fast randomized algorithm which succeeds with probability $1-o(n^{-1})$.*

For the description of the algorithm and the proof of Lemma 1.4, see Appendix B. In what follows, we refer to this algorithm as Approximate Clustering. Combining Theorem 1.3 and Lemma 1.4, we obtain Algorithm 5 and Theorem 1.5.

Algorithm 5: SVD III

(0) Randomly partition X into two subsets Y and Z. Let B be the adjacency matrix of the bipartite graph between Y and Z. Let Y_1 be a random subset of Y by selecting each element with probability $1/2$ independently and let $\hat{A}$ be the submatrix of B formed by the columns indexed by Y_1.
(1) Project the columns of B indexed by $Y_2 := Y \backslash Y_1$ on $\hat{A}_k$.
(2) Run *Approximate Clustering* on the projected points.

Theorem 1.5. *Given constants $\varepsilon, \gamma, k > 0$, there is a constant $C > 0$ such that the following holds for any hidden γ balanced partition $X = \cup_{i=1}^k X_i$. If $\sigma^2 \geqslant C \log n/n, s \geqslant C \log n$ and*

$$\Delta \geqslant C\sigma \sqrt{\frac{n}{s}},$$

then with probability at least $1-\varepsilon$ SVD III finds an ε-correct partition of Y_2.

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

The advantage of Theorem 1.5 is that its assumption on Δ is both simpler and stronger than that of Theorems 1.1 and 1.2. Of course, the partition it outputs is only ε-correct. There is an interesting point here: It turns out that in many cases one can easily upgrade an ε-correct partition to an exact one. We will discuss this idea in the Hidden Bipartition problem below.

An important result that overlaps ours is that of Coja-Oghlan [10, Theorem 1.1], which also improves upon Theorem 1.1, using an adaptive algorithm. The setting of [10, Theorem 1.1] is more general than ours, allowing both very small and very large densities. Its purpose is to recover an approximate partition, under certain assumptions. Assumption $R1$ in this theorem requires the maximum expected degree to be at least $\log^2(n/s)$; we do not have this assumption. Assumption $R2$ requires $s \geqslant \log^{30} n$; we require $s \geqslant C \log n$ (in fact, $s \geqslant C$ is sufficient if we are satisfied with success probability 0.99 instead of $1 - o(n^{-1})$). The main assumption $R3$ is a lower bound on Δ of the form

$$\Delta \geqslant Ck^{3/2}\sigma\sqrt{\frac{n}{s}} + C\log\left(D + \frac{n}{s}\right) \max_{1 \leqslant i \leqslant k} \sum_{j=1}^{k} p_{ij}(1 - p_{ij}).$$

This assumption and the corresponding assumption in Theorem 1.2 are incomparable. In the case when the first term is dominating, our assumption does not require the $k^{3/2}$ factor. If one aims for approximate recovery, the assumption on Δ in Theorem 1.5 is the weakest. The proofs in [10] also used spectral techniques, but seem more delicate and longer than ours; see [10] for more details.

Let us now consider the performance of SVD II and SVD III on a few subproblems. We allow the value of C to be flexible in order to omit smaller order terms for convenience.

Hidden Clique. In this problem, $k = 2$, s is the size of the clique, and $\Delta = (1 - p)\sqrt{s}$, where p is the density of the random graph, which is assume to be bounded away from 1 ($p < 0.99$, say). Condition 1 becomes

$$(1 - p)s^{1/2} \geqslant C\left(p^{1/2}\sqrt{\frac{n}{s}} + \sqrt{\log n}\right)$$

which is satisfied if $s \geqslant C(\sqrt{np} + \log n)$.

Corollary 1.6. *There is a constant C such that, for any $0.99 > p \geqslant C \log n/n$ and $s \geqslant C(\sqrt{np} + \log n)$, SVD II finds the hidden clique of size s with probability $1 - o(1)$.*

This result is comparable to [27, Corollary 3]. The first polynomial-time algorithm for hidden cliques of size $C\sqrt{n}$, for a large constant C, was provided by Alon, Krivelevich and Sudakov [2]. In fact, they showed that one can reduce C to any constant $\varepsilon > 0$, at the cost of pushing the running time to $n^{f(\varepsilon)}$, where $f(\varepsilon)$ tends to infinity as ε tends to zero. The constants C in all these works are often large (and implicit). If one needs a really fast algorithm, then the best current C is e^{-1}, obtained by Deshpande and Montanary in a recent paper [14].

Hidden Bipartition. Let the two densities be $0.99 \geqslant p > q > 0$. We have $k = 2$, $\Delta = |p - q|n^{1/2}$, $s = n/2$, $\sigma^2 = \Theta(p)$. The two singular values of P are $(p + q)n$ and $(p - q)n$. Condition 2 of Theorem 1.2 requires $(p - q)/p^{1/4} \geqslant C\sqrt{\log n/n}$.

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

Corollary 1.7. *There is a constant C such that the following holds. Let $0.99 > p > q \geqslant C \log n/n$ be edge densities such that $(p-q)/p^{1/4} \geqslant C\sqrt{\log n/n}$. Then SVD II finds the hidden bipartition with probability $1 - o(n^{-1})$.*

The best known condition on Δ is $(p-q)/\sqrt{p} \geqslant C\sqrt{\log n/n}$, under stronger density assumptions; see [7, 11, 27]. We can obtain this bound on Δ with Theorem 1.5 and an extra idea. Let us first apply Theorem 1.5. The condition on Δ has become $(p-q)/\sqrt{p} \geqslant C\sqrt{\log n/n}$. Thus, we have the following result.

Corollary 1.8. *For any $\varepsilon > 0$ there is a constant C such that the following holds. Let $0.99 > p > q \geqslant C \log n/n$ be edge densities such that $(p-q)/\sqrt{p} \geqslant C\sqrt{\log n/n}$. Then SVD III finds an ε-correct partition with probability at least $1 - \varepsilon$.*

We next upgrade an ε-correct partition to an exact one using the following general idea. At the beginning we randomly split the input graph into two parts, Red and Blue, by colouring each edge Red or Blue with probability 1/2, independently (see Algorithm 6). First use the Red part as input to recover an ε-correct partition for some small ε (say $\varepsilon = 0.1$). Next, reveal the Blue graph and use information about edge distribution of this graph to correct the misclassified vertices.

Algorithm 6: Hidden Bipartition

(0) Randomly colour the edges of the input graph Red and Blue with probability $1/2$ each.
(1) Use SVD III on the Red graph to produce an 0.1-correct partition $X_1' \cup X_2'$.
(2) Reveal of Blue graph. For $u \in X_i'$, label it misclassified if the number of Blue neighbours (of u) in X_i' is less than the number of Blue neighbours (of u) in X_{3-i}'. Otherwise u is well-classified.
(3) Output X_i as the union of well-classified vertices in X_i' and misclassified vertices in X_{3-i}'.

Corollary 1.9. *For any $\varepsilon > 0$ there is a constant C such that the following holds. Let $0.99 > p > q \geqslant C \log n/n$ be edge densities such that $(p-q)/\sqrt{p} \geqslant C\sqrt{\log n/n}$. Then algorithm Hidden Bipartition solves the Hidden Bipartition problem with probability at least $1 - \varepsilon$.*

We prove Corollary 1.9 in Section 4. This corollary is comparable with [27, Corollary 1], but with a better (optimal) density assumption. The first result on Hidden Bipartition was obtained by Bui, Chaudhuri, Leighton and Sipser [8] and Dyer and Frieze [15] under the condition $q < (1-c)p$. For a related problem of finding the optimal bisection (which may not come from the hidden one), Boppana [7] presents a spectral algorithm which succeeds for a large range of parameters, using convex optimization. Condon and Karp [11] analysed a linear-time combinatorial algorithm for partitioning which nearly achieves the same range of parameters as [7]. For works concerning densities as small as c/n, we refer to Coja-Oghlan [10] and a recent paper by Chin, Rao and Vu [9]. In Section 4 we will also discuss an analogue of Corollary 1.9 for the Hidden Colouring problem.

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

The rest of the paper is organized as follows. In the next section we present a few technical lemmas, including Lemma 2.1 mentioned above. Then we prove Theorem 1.2 in Section 3. In Section 4 we prove Theorem 1.3 and Corollary 1.9 and discuss related results. The proofs of Lemma 2.1 and Lemma 1.4 will be presented in the Appendix.

2. Technical lemmas

Lemma 2.1 (projection of a random vector). *There are constants C_1, C_2 such that the following holds. Let $X = (\xi_1, \dots, \xi_n)$ be a random vector in $\mathbb{R}^n$ whose coordinates ξ_i are independent random variables with mean 0 and variance at most $\sigma^2 \leqslant 1$. Assume furthermore that the ξ_i are, with probability 1, bounded by 1 in absolute value. Let H be a subspace of dimension d and let $\Pi_H \xi$ be the length of the orthogonal projection of ξ onto H. Then*

$$\mathbb{P}(\Pi_H X \geqslant \sigma\sqrt{d} + C_1\sqrt{\log n}) \leqslant n^{-3}.$$

Furthermore, if H has an orthonormal basis $v_1, \dots, v_d$ such that $\max_{1 \leqslant i \leqslant d} \|v_i\|_\infty \leqslant \alpha$, then

$$\mathbb{P}(\Pi_H X \geqslant C_2\sqrt{d}(\sigma\sqrt{\log n} + \alpha \log n)) \leqslant n^{-3}.$$

We prove Lemma 2.1 in the Appendix.

Lemma 2.2 (norm of a random matrix). *There is a constant $C_0 > 0$ such that the following holds. Let E be a symmetric matrix whose upper diagonal entries e_{ij} are independent random variables where $e_{ij} = 1 - p_{ij}$ or $-p_{ij}$ with probabilities p_{ij} and $1 - p_{ij}$, respectively, where $0 \leqslant p_{ij} \leqslant 1$. Let $\sigma^2 := \max_{ij} p_{ij}(1 - p_{ij})$. If $\sigma^2 \geqslant C_0 \log n/n$, then*

$$\mathbb{P}(\|E\| \geqslant C_0 \sigma n^{1/2}) \leqslant n^{-3}.$$

If $\sigma^2 \geqslant \log^4 n/n$, the statement is a corollary of [30, Theorem 1.4]. For smaller σ, one can prove this lemma using the ε-net approach by Kahn and Szemerédi [22]. We omit the details, which are very similar to the proof of Feige and Ofek for [17, Theorem 1.1].

Lemma 2.3 (perturbation bound). *Let M, N be matrices and $\delta := \lambda_k(M) - \lambda_{k+1}(M) > 0$. Then*

$$\sin \angle(M_k, (M+N)_k) \leqslant \delta^{-1}\|N\|.$$

This lemma is a well-known result in numerical linear algebra, known as the Davis–Kahan–Wedin theorem; see [5, 12, 20, 32].

3. Proof of Theorem 1.2

Recall that in the first step of the algorithm, we randomly partition the vertex set X into two subsets Y and Z. Let B be the adjacency matrix of the bipartite graph between Y and Z. Let Y_1 be

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

a random subset of Y by selecting each element with probability $1/2$ independently and let $\hat{A}$ be the submatrix of B formed by the columns indexed by Y_1.

Let A be the probability matrix p_{ij} corresponding to $\hat{A}$. As A is a large random submatrix of P, it is not hard to show that $\lambda_k(A) \geqslant \frac{1}{8}\lambda_k(P)$ with high probability (we provide a verification of this fact at the end of the proof).

We view the adjacency matrix $\hat{A}$ (between Y_1 and Z) as a random perturbation of A, $\hat{A} := A + E$, where the entries e_{ij} of E are independent and $e_{ij} = 1 - p_{ij}$ with probability p_{ij} and $-p_{ij}$ with probability $1 - p_{ij}$. We let $\hat{\mathbf{u}}, \mathbf{u}, e_u$ denote the columns corresponding to a vertex u in $\hat{A}, A, E$, respectively. All matrices are of size approximately $n/2 \times n/4$ by the definitions of Y, Z and Y_1, Y_2.

Our leading idea is that the random perturbation E does not change A_k too much, thus hopefully the projections onto $\hat{A}_k$ and A_k differ by only a small amount. The heart of the matter, of course, is to bound this error term. While tempting, a straightforward application of Lemma 2.3 is too crude in the general case (it does lead to some simple solution for some subproblems in certain range of parameters). We will still make use of this lemma, but for a quite different purpose.

For simplicity, we assume in the rest of the proof that $s \geqslant C \log n$. For a sufficiently large C, this implies that with probability $1 - o(n^{-1})$, each cluster X_i intersects Z in at least $|X_i|/3$ elements. Thus, the distance between two columns (belonging to different clusters) in A is at least $\Delta/3$. We aim to show that with high probability $\|P_{\hat{A}_k}\hat{\mathbf{u}} - \mathbf{u}\| < \Delta/30$ for all $u \in Y_2$. It is easy to check that this provides a perfect geometric representation with $r = \Delta/15$. If there is no lower bound on s, then the probability that the random partition has this property is at least $1 - c\sum_{i=1}^{k} e^{-|X_i|/c}$ for some constant $c > 0$.

For a fixed u, by the triangle inequality,

$$\|P_{\hat{A}_k}\hat{\mathbf{u}} - \mathbf{u}\| \leqslant \|P_{\hat{A}_k}(\hat{\mathbf{u}} - \mathbf{u})\| + \|(P_{\hat{A}_k} - I)\mathbf{u}\| = \|P_{\hat{A}_k} e_u\| + \|(P_{\hat{A}_k} - I)\mathbf{u}\|.$$

To bound the second term, we follow an argument from [27] and consider

$$(P_{\hat{A}_k} - I)A = (P_{\hat{A}_k} - I)\hat{A} - (P_{\hat{A}_k} - I)E.$$

The spectral norm of the first term is $\lambda_{k+1}(\hat{A}_k)$, which is at most $\lambda_{k+1}(A) + \|E\|$ by Weyl's inequality. On the other hand, $\lambda_{k+1}(A) + \|E\| = \|E\|$, as A has rank at most k. The spectral norm of the second term is also at most $\|E\|$, by the basic fact that $\|MN\| \leqslant \|M\|\|N\|$. Thus, by Lemma 2.2, with probability at least $1 - n^{-3}$,

$$\|(P_{\hat{A}_k} - I)A\| \leqslant 2\|E\| \leqslant C_0\sigma n^{1/2},$$

for some constant C_0. (In what follows, we allow ourselves to adjust the value of C_0 slightly in order to absorb negligible terms.)

Let χ_u be the vector $s_u^{-1/2}\mathbf{I}_u$, where $\mathbf{I}_u$ is the indicator vector for the cluster containing u. As χ_u has unit length, for any matrix M we have $\|M\| \geqslant \|M\chi_u\|$; thus

$$\|(P_{\hat{A}_k} - I)A\| \geqslant \|(P_{\hat{A}_k} - I)A\chi_u\| = s_u^{1/2}\|(P_{\hat{A}_k} - I)\mathbf{u}\|.$$

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

Combining the last two inequalities, we conclude that with probability at least $1-n^{-3}$

$$\|(P_{\hat{A}_k}-I)u\| \leqslant C_0\sigma\sqrt{\frac{n}{s_u}},$$

for all $u \in X$.

Now we take care of the first term, whose analysis is more involved. By the first part of Lemma 2.1,

$$\| P_{\hat{A}_k} e_u\| \leqslant \sigma k^{1/2} + C_1\sqrt{\log n}$$

with probability $1-o(n^{-2})$, for a properly chosen constant C_1. As $sk \leqslant n$, the term $\sigma k^{1/2}$ is at most $\sigma\sqrt{n/s}$ and can be omitted. This yields that if

$$\Delta \geqslant C_0\sigma\sqrt{n/s}+C_1\sqrt{\log n}$$

then the algorithm succeeds with probability at least $1-o(n^{-1})$. This proves the first part of the theorem concerning Condition 1.

To prove the second part (Condition 2), we find a different way to bound the distance $P_{\hat{A}_k}e_u$. Rewrite $\hat{A}=A+E$ and let v be a column vector of A, normalized to have unit length. Recall that $|X_i\cap Z| \geqslant \frac{1}{3}|X_i| = s_i/3$ for all i. It follows that each coordinate in v is repeated at least $s/3$ times. The sets where the coordinates are repeated are the same for different vectors v. It follows for any vector w in Span(A) that each coordinates of w is repeated at least $s/3$ times. Consequently, $\|w\|_\infty \leqslant 2s^{-1/2}$. Furthermore, by Lemma 2.3 and Lemma 2.2, we have with probability $1-o(n^{-2})$ that

$$\sin(A_k,\hat{A}_k) \leqslant C_0\frac{\sigma\sqrt{n}}{\lambda},$$

which implies that, for any unit vector $v \in \hat{A}_k$,

$$\|v\|_\infty \leqslant 2s^{-1/2}+C_0\frac{\sigma\sqrt{n}}{\lambda} := \alpha.$$

Using the second part of Lemma 2.1, we conclude that with probability $1-o(n^{-2})$

$$\|P_{\hat{A}_k}e_u\| \leqslant C\sqrt{k}(\sigma\sqrt{\log n}+\alpha\log n) = C\sqrt{k}\left(\sigma\sqrt{\log n}+\frac{\log n}{\sqrt{s}}+\frac{\sigma\sqrt{n}\log n}{\lambda}\right),$$

for all u and some properly chosen constant C, concluding the proof.

Let us now show that with high probability, $\lambda_k(A) \geqslant \frac{1}{8}\lambda_k(P)$. We first compare the singular values of P with the singular values of $\tilde{P}$, the probability matrix of the bipartite graph spanned by Y and X. Using Chernoff's bound, one can easily show that with probability at least $1-n^{-2}$

$$||X_i\cap Y|-|X_i|/2| \leqslant 5\sqrt{|X_i|\log n}, \tag{3.1}$$

for all $1 \leqslant i \leqslant k$.

We use the fact that for any matrix M of rank k $\lambda_k(M) = \inf_{\text{rank}(M')=k-1}\|M-M'\|_F$. For simplicity, let us assume for a moment that $|X_i\cap Y| = |X_i|/2$. Let $\tilde{P}'$ be the matrix that defines $\lambda_k(\tilde{P})$. We define P', a rank $(k-1)$ approximation of P, by extending $\tilde{P}'$ as follows. For the block indexed by $X_i\backslash Y$, simply copy the block of $\tilde{P}'$ corresponding to $X_i\cap Y$. It is trivial that P' has rank

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

$k-1$ and

$$\|P-P'\|_F^2 = 2\|\tilde{P}-\tilde{P}'\|_F^2,$$

which implies $\lambda_k \leqslant \sqrt{2}\lambda_k(\tilde{P})$. With the same argument, we can compare $\lambda_k(\tilde{P})$ with $\lambda_k(B)$ and the latter with $\lambda_k(A)$, each time losing a factor of $\sqrt{2}$. At the end it would give $\lambda_k(P) \leqslant 2^{3/2}\lambda_k(A)$.

To make the argument precise, we need to remove the assumption $|X_i \cap Y| = |X_i|/2$. Using (3.1) instead of this assumption, we can create a matrix P' such that

$$\|P-P'\|_F^2 \leqslant 2\|\tilde{P}-\tilde{P}'\|_F^2 + 5\sum_{i=1}^{k}\sqrt{|X_i|\log n}\sigma^4.$$

On the other hand, the extra term $5\sum_{i=1}^{k}\sqrt{|X_i|\log n}\sigma^4$ is less than $\frac{1}{4}\lambda_k(P)^2$ by the assumption of the theorem. Thus, we can use the above estimate to get a slightly weaker bound $\lambda_k(P) \leqslant 2\lambda_k(\tilde{P})$, which leads to $\lambda_k(P) \leqslant 8\lambda_k(A)$, as desired.

4. Approximate solutions and upgrading

4.1. Proof of Theorem 1.3

We follow the proof of Theorem 1.2. The key is to bound $\|P_{\hat{A}_k} e_u\|$. Recall that

$$\mathbb{E}\|P_{\hat{A}_k} e_u\|^2 \leqslant \sigma^2 k.$$

By Markov's inequality, it follows that $\mathbb{P}(\|P_{\hat{A}_k} e_u\| \geqslant K\sigma k^{1/2}) \leqslant K^{-2}$. We call a vertex u *good* if $\|P_{\hat{A}_k} e_u\| \leqslant K\sigma k^{1/2}$. For a sufficiently large C (depending on K), all good vertices will be clustered correctly. Moreover, choosing $K \geqslant 2\varepsilon^{-1/2}$, the probability for u being good is at least $1-\varepsilon/4$, thus the expectation of the number of good elements in X_i is at least $|X_i|(1-\varepsilon/4)$. As the good events are independent, an easy application of Chernoff's bound yields that with probability $1-n^{-2}$, at least $|X_i|(1-\varepsilon)$ points from X_i are good. This completes the proof.

4.2. Hidden Bipartition: Proof of Corollary 1.9

We can assume, without generality, that $1/100 > p > q > 0$. (To obtain the upper bound on p, one can randomly sparsify the input graph if necessary.) Notice that the densities in the Red graph are $p/2$ and $q/2$. By Theorem 1.5, we obtain an 0.1-correct partition with probability at least $1-\varepsilon/2$, provided that

$$\frac{p/2-q/2}{\sqrt{p/2}} \geqslant C\sqrt{\frac{\log n}{n}}$$

for some sufficiently large constant C. By Chernoff's bound, one can prove that with probability $1-o(n^{-1})$, all degrees in the Red graph are at most $n/50 = 0.02n$. In what follows, we condition on this event.

Consider the Blue graph. Intuitively, this graph is also random with densities $p/2$ and $q/2$. We have to be a bit careful, however, since the Blue graph is not entirely independent from the Red graph.

- If e is an edge in the Red graph, then it cannot be an edge in the Blue graph.

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

- If e is not an edge in the Red graph and both end points are in X_1 (or X_2), then e is an edge in the Blue graph with probability $p_1 := (p/2)/(1-p/2)$.
- If e is not an edge in the Red graph and one end point in X_1 and the other in X_2, then e is an edge in the Blue graph with probability $q_1 := (q/2)/(1-q/2)$.
- Conditioned on the Red graph, the events of non-edges in the Red graph becoming edges in the Blue graph are mutually independent.

Let $X_1' \cup X_2'$ be the 0.1-correct partition obtained by SVD III with the Red graph as input. Let $X_i' := A_i \cup B_i$, where $A_i := X_i \cap X_i'$ and $B_i := X_i' \cap X_{3-i}$. By the definition of ε-correctness (in our case $\varepsilon = 0.1$), A_i have size at least $\frac{1}{2}(1-\varepsilon)n = 0.45n$, for $i = 1,2$. It follows that $0.55n \geqslant |X_i'| \geqslant 0.45n$ and $|B_i| \leqslant 0.1n$.

Consider $u \in X_1'$. Let $N_R(u)$ be the set of neighbours of u in the Red graph and let $d_i(u)$ be the number of neighbours of u in X_i' in the Blue graph.

If $u \in X_1$ (*i.e.* u is well-classified), then

$$d_1(u) := D_1 = \sum_{x \in A_1 \setminus N_R(u)} \chi(x) + \sum_{y \in B_1 \setminus N_R(u)} \mu(y),$$

where $\chi(x)$ are i.i.d. indicator variables with mean p_1, and $\mu(y)$ are i.i.d. indicator variables with mean q_1. Furthermore,

$$d_2(u) := D_2 = \sum_{x \in B_1 \setminus N_R(u)} \chi(x) + \sum_{y \in A_1 \setminus N_R(u)} \mu(y).$$

We have

$$D := D_1 - D_2 = \sum_{x \in A_1 \setminus N_R(u)} (\chi(x) - \mu(x)) - \sum_{y \in B_1 \setminus N_R(u)} (\chi(y) - \mu(y)).$$

As $N_R(u) \leqslant 0.02n$ and $|A_i| \geqslant 0.45n$, $B_i| \geqslant 0.1n$, it follows that

$$\mathbb{E}D \geqslant (p_1 - q_1)(|A_1 \setminus N_R(u)| - |B_1 \setminus N_R(u)|),$$

where the left-hand side is at least

$$(p_1 - q_1)(0.45n - 0.02n - 0.1n) \geqslant 0.3n(p_1 - q_1) \geqslant 0.15n(p - q) \geqslant 5 \log n,$$

provided that the constant C in Corollary 1.9 is sufficiently large. Applying Chernoff's bound, it is easy to show that with probability at least $1 - n^{-3}$, $D > 0$ or $d_1(u) > d_2(u)$. A similar argument shows that if $u \in X_2$ (misclassified), then $d_1(u) < d_2(u)$. By the union bound, we conclude that the algorithm Hidden Bipartition succeeds with probability at least

$$1 - \varepsilon/2 - o(n^{-1}) - n \times n^{-3} \geqslant 1 - \varepsilon,$$

concluding the proof.

4.3. Hidden Colouring

We can obtain the following analogue of Corollary 1.9.

Corollary 4.1. *For any constant $\varepsilon > 0$ there is a constant C such that the following holds. Let $0.99 > p \geqslant C \log n/n$. Then algorithm Hidden Colouring (see Algorithm 7) solves the Hidden Colouring problem with probability $1 - \varepsilon$.*

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

Algorithm 7: Hidden Colouring

(0) Randomly colour the edges of the input graph Red, Blue, Green with probability $1/3$ each.
(1) Use SVD III on the Red graph to produce an $0.1/k$-correct partition $X_1' \cup X_2' \cdots \cup X_k'$.
(2) Reveal the Blue graph. For $u \in X_i'$, label u misclassified if the number of neighbours of u in X_i' is at least $0.5/(k-1)$ its total degree. Let Y_i be the set obtained from X_i' by deleting the misclassified vertices.
(3) Output X_i as the set of all vertices with no neighbour in Y_i in the Green graph.

Many researchers have worked on the problem of colouring random graphs which have k-colourings. Kučera [24], Turner [29] and Dyer and Frieze [12] have presented effective algorithms for dense graphs. Prior to McSherry's paper [27], Blum and Spencer [4] and Alon and Kahale [1] demonstrated algorithms that colour random sparse graphs properly with high probability. Corollary 4.1 is comparable to [27, Corollary 2], with a better (optimal) density bound. If we aim for an approximate recovery (or an optimal colouring which may not come from the hidden one), then there are algorithms which work for lower density $\Omega(1/n)$; see [1, 10], in particular the discussion in [10, Section 2.1].

The proof for the misclassified part follows the same idea as in the previous section; we omit the details. After step (2), we receive sets Y_i which are big subsets of X_i. (One can easily show that $|Y_i| \geqslant |X_i|/2$ with high probability.) It is easy to see (again by Chernoff's bound) that the only vertices which have no neighbours in Y_i (in the Green graph) are the vertices of X_i. This concludes the proof.

Using the same idea, one can handle a common generalization of Hidden Bipartition and Hidden Colouring. Let $X_1, \ldots, X_k$ be sets of size n/k. Draw edges within each X_i with probability p and between X_i and X_j with probability q.

Corollary 4.2. *For any $\varepsilon > 0$ there is a constant C such that the following holds. Let $0.99 > p, q \geqslant C \log n/n$ be edge densities such that $|p-q|/\sqrt{p} \geqslant C\sqrt{\log n/n}$. Then one can recover the partition with probability at least $1-\varepsilon$ by an efficient algorithm.*

This corollary is a variant of [27, Corollary 1], again with a superior density bound; we omit the details.

Appendix A: Proof of Lemma 2.1

Notice that the function $\Pi_H(X)$ is 1-Lipschitz and convex; thus by Talagrand's inequality [28], for any $t > 0$,

$$\mathbb{P}(\Pi_H X \geqslant \mu + t) \leqslant 2\exp(-t^2/4)$$

where μ is the median of $\Pi_H(X)$. We do not know μ; however, we can bound from above. Slightly abusing the notation, let $\Pi := (\pi_{ij})$ denote the projection matrix onto H; then, by

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

independence,

$$\mathbb{E}|\Pi_H X|^2 = \mathbb{E}X^T \Pi X = \sum_{i=1}^{n} \pi_{ii} \mathbb{E}\xi_i^2 \leqslant \sigma^2 \sum_{i=1}^{n} \pi_{ii} = d\sigma^2.$$

Combining this with the concentration inequality, it is not hard to show that $\mu \leqslant \sigma d^{1/2} + O(1)$, concluding the proof of the first part of the lemma. The reader can also check [31] for a detailed discussion of inequalities of this type.

To prove the second part, notice that if $v_1, \ldots, v_d$ form an orthonormal basis of H, then

$$\|\Pi_H X\|^2 = \sum_{i=1}^{d} |X \cdot v_i|^2.$$

Thus, our statement is a direct consequence of the following claim.

Claim A.1. *Let $(a_1, \ldots, a_n)$ be real numbers such that $\sum_i a_i^2 = 1$ and $|a_i| \leqslant \alpha$ for all i. Let ξ_i be independent random variables with mean 0 and $\mathbb{E}|\xi_i|^k \leqslant \sigma^2$ for all $k \geqslant 2$. Let $S := \sum_{i=1}^n a_i \xi_i$. Then*

$$\mathbb{P}(|S| \geqslant 4(\sigma\sqrt{\log n} + \alpha \log n)) \leqslant 2n^{-3}.$$

To prove Claim A.1, notice that for any $0 < t \leqslant \alpha^{-1}$ we have

$$\mathbb{E}\exp(tS) = \prod_i \mathbb{E}\exp(t a_i \xi_i) \leqslant \prod_i \left(1 + \frac{\sigma^2 a_i^2 t^2}{2!} + \frac{t^3 a_i^3 \mathbb{E}\xi_i^3}{6!} + \cdots\right).$$

(We have inequality here as σ^2 is an upper bound on the variance.) Since $\mathbb{E}\xi_i^k \leqslant \sigma^2$ for all $k \geqslant 2$ and $t|a_i| \leqslant 1$, the rightmost formula is

$$\leqslant \prod_i (1 + \sigma^2 t^2 a_i^2) \leqslant \exp(\sigma^2 t^2).$$

Markov's inequality yields

$$\mathbb{P}(S \geqslant T) \leqslant \exp(-tT + t^2\sigma^2).$$

To optimize the right-hand side, let us consider two cases

Case 1. $\sigma \geqslant \alpha\sqrt{\log n}$. Take $T = 4\sigma\sqrt{\log n}$ and $t = \sqrt{\log n}/\sigma \leqslant \alpha^{-1}$. With this setting $-tT + t^2\sigma^2 = -3\log n$.

Case 2. $\sigma < \alpha\sqrt{\log n}$. Take $T = 4\alpha \log n$ and $t = \alpha^{-1}$. In this setting, $-tT + t^2\sigma^2 \leqslant -4\log n + \log n = -3\log n$.

One can bound $\mathbb{P}(-S \leqslant T)$ the same way.

Appendix B: Approximate Clustering

To analyse Algorithm 8, let us first consider the case that the clusters X_i have the same size. In this case $s = n/k$ and $\gamma = 0$.

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

Algorithm 8: Approximate Clustering

The input is a δ-perfect set X.

(0) Set $S_0 := X$.

(1) For $i = 0, \dots, k-1$, choose a random point w from S_i. Find a set $\tilde{X}_{i+1}$ of $(1-\delta)s$ points of distance at most $2r$ to w. Set $S_{i+1} = S_i \backslash \tilde{X}_{i+1}$.

(2) Partition S_k into k parts $\tilde{X}'_i, 1 \leqslant i \leqslant k$, of size $|X_i| - (1-\delta)s$, respectively. Output $X'_i := \tilde{X}_i \cup \tilde{X}'_i$; $i = 1, \dots, k$.

(3) If in (1) one cannot find X_{i+1} for some i, go back to (0) and repeat the cycle.

Let W be a δ-perfect representation of a set of size N. Call a point $w \in W$ *good* if it has distance at most r to one of the centres $x_1, \dots, x_k$. If a point w is of distance at most r to x_j, then the ball of radius $2r$ around w contains at least $(1-\delta)s$ points from X_j and at most $(k-1)\delta s$ points from other X_l. Thus, if we take any set $\tilde{X}_j$ of $(1-\delta)s$ points in this ball, then at least $(1-\delta k)s$ of them belongs to X_j. For $\delta := \varepsilon/k$, $X'_i s$ satisfies $|X_i \backslash X'_i| \leqslant \varepsilon s$ and we obtain an ε-perfect partition.

The probability that step (1) goes through successfully is the probability that we can choose k consecutive good points. Notice that in S_i the number of good points is at least

$$(1-\delta)n - (1-\delta)is = (1-\delta)ks - (1-\delta)is = (1-\delta)(k-i)s.$$

On the other hand,

$$|S_i| = n - (1-\delta)is = ks - (1-\delta)is = (k - (1-\delta)i)s.$$

So the chance that we pick up a good point in S_i is at least

$$\frac{(1-\delta)(k-i)}{k-(1-\delta)i} := p_i.$$

The probability that step (1) goes through is at least

$$\rho := \prod_{i=0}^{k-1} p_i.$$

The analysis for the case when the X_i is γ-balanced ($|X_i| \leqslant (1+\gamma)s$) is similar. If a point w is of distance at most r to x_j, then the ball of radius $2r$ around w contains at least $(1-\delta)s$ points from X_j and at most $(k-1)\delta(1+\gamma)s$ points from other X_l. Thus, if we take any set $\tilde{X}_j$ of $(1-\delta)s$ points in this ball, then at least

$$(1-\delta)s - (k-1)\delta(1+\gamma)s = (1 - k\delta - (k-1)\delta(1+\gamma))s$$

of them belong to X_j. So for $\delta := \varepsilon/(k + (k-1)(1+\gamma))$, $X'_j s$ satisfies $|X_j \backslash X'_j| \leqslant \varepsilon s$ and we obtain an ε-perfect partition.

The values of p_i and ρ change slightly compared to the case $\gamma = 0$. The size of S_i now is

$$|S_i| = n - (1-\delta)is \geqslant (1+\gamma)ks - (1-\delta)is = ((1+\gamma)k - (1-\delta)i)s.$$

Thus

$$p_i := \frac{(1-\delta)(k-i)}{(1+\gamma)k - (1-\delta)i}$$

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

and

$$\rho := \prod_{i=0}^{k-1} p_i \geqslant \prod_{i=0}^{k-1} \frac{(1-\delta)(k-i)}{k(1+\gamma)} = k! \left(\frac{1-\delta}{k(1+\gamma)}\right)^k \geqslant \left(\frac{1-\delta}{e(1+\gamma)}\right)^k.$$

As a consequence, we obtain the following lemma, which implies Lemma 1.4.

Lemma B.1. *With probability at least* $1 - n^{-2}$*, Approximate Clustering produces an* ε*-correct partition after at most*

$$3\rho^{-1} \log n \leqslant 3 \left(\frac{e(1+\gamma)}{1-\delta}\right)^k \log n$$

cycles, given $\delta = \varepsilon/(k + (k-1)(1+\gamma))$.

Acknowledgement

V. Vu is supported by research grants DMS-1500944 and FA9550-12-1-0083. He would like to thank the referee for his/her careful reading and numerous suggestions which greatly improved the quality of the paper.

References

[1] Alon, N. and Kahale, N. (1997) A spectral technique for coloring random 3-colorable graphs. *SIAM J. Comput.* **26** 1733–1748.

[2] Alon, N., Krivelevich, M. and Sudakov, B. (1998) Finding a large hidden clique in a random graph. *Random Struct. Alg.* **13** 457–466.

[3] Azar, Y., Fiat, A., Karlin, A., McSherry, F. and Saia, J. (2001) Spectral analysis of data. In *STOC '1: Proc. 33rd Annual ACM Symposium on Theory of Computing*, ACM, pp. 619–626.

[4] Bai, Z. and Silverstein, J. W. (2009) *Spectral Analysis of Large Dimensional Random Matrices*, Springer.

[5] Bhatia, R. (1997) *Matrix Analysis*, Vol. 169 of Graduate Texts in Mathematics, Springer.

[6] Blum, A. and Spencer, J. (1995) Coloring random and semi-random k-colorable graphs. *J. Algorithms* **19** 204–234.

[7] Boppana, R. B. (1987) Eigenvalues and graph bisection: An average-case analysis. In *SFCS '87: Proc. 28th Annual Symposium on Foundations of Computer Science*, ACM, pp. 280–285.

[8] Bui, T. N., Chaudhuri, S., Leighton, F. T. and Sipser, M. (1987) Graph bisection algorithms with good average case behavior. *Combinatorica* **7** 171–191.

[9] Chin, P., Rao, A. and Vu, V. (2015) Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery. *Proc. of Machine Learning Research* **40** 391–423.

[10] Coja-Oghlan, A. (2010) Graph partitioning via adaptive spectral techniques. *Combin. Probab. Comput.* **19** 227–284.

[11] Condon, A. and Karp, R. M. (2001) Algorithms for graph partitioning on the planted partition model. *Random Struct. Alg.* **18** 116–140.

[12] Davis, C. and Kahan, W. M. (1970) The rotation of eigenvectors by a perturbation III. *SIAM J. Numer. Anal.* **7** 1–46.

[13] Dekel, Y., Gurel-Gurevich, O. and Peres, Y. (2011) Finding hidden cliques in linear time with high probability. In *ANALCO11: Proc. Eighth Workshop on Analytic Algorithmics and Combinatorics*, SIAM, pp. 67–75.

https://doi.org/10.1017/S0963548317000463 Published online by Cambridge University Press

[14] Deshpande, Y. and Montanari, A. (2015) Finding hidden cliques of size $\sqrt{N/e}$ in nearly linear time. *Found. Comput. Math.* **15** 1069–1128.
[15] Dyer, M. E. and Frieze, A. M. (1986) Fast solution of some random NP-hard problems. In *27th Annual Symposium on Foundations of Computer Science*, IEEE, pp. 221–336.
[16] Feige, U. and Krauthgamer, R. (2000) Finding and certifying a large hidden clique in a semirandom graph. *Random Struct. Alg.* **16** 195–208.
[17] Feige, U. and Ofek, E. (2005) Spectral techniques applied to sparse random graphs. *Random Struct. Alg.* **27** 251–275.
[18] Feige, U. and Ron, D. (2010) Finding hidden cliques in linear time. In *AofA'10: 21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms*, DMTCS Proc. AM, pp. 189–204.
[19] Füredi, Z. and Komlós, J. (1981) The eigenvalues of random symmetric matrices. *Combinatorica* **1** 233–241.
[20] Golub, G. H. and Van Loan, C. F. (1996) *Matrix Computations*, Johns Hopkins University Press.
[21] Jerrum, M. and Sorkin, G. B. (1993) Simulated annealing for graph bisection. In *Proc. 34th Annual IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 94–103.
Friedman, J., Kahn, J. and Szemeredi, E. (1989) On the second eigenvalue of random regular graphs, STOC '89 Proceedings of the twenty-first annual ACM symposium on Theory of computing Pages 587–598.
[22] Friedman, J., Kahn, J. and Szemerédi, E. (1989) On the second eigenvalue of random regular graphs. In *Proc. of the twenty first annual ACM symposium on theory of computing: STOC*, pp. 587–598.
[23] Kannan, R. and Vempala, S. (2009) *Spectral Algorithms*, Now Publishers Inc.
[24] Kučera, L. (1977) Expected behavior of graph coloring algorithms. In *Fundamentals of Computation Theory: Proc. 1977 International FCT-Conference*, Springer, pp. 447–451.
[25] Matula, D. W. (1972) The employee party problem. *Notices Amer. Math. Soc.* **19** A-382.
[26] McDiarmid, C. (1998) Concentration. In *Probabilistic Methods for Algorithmic Discrete Mathematics* (M. Habib *et al.*, eds), Springer, pp. 195–248.
[27] McSherry, F. (2001) Spectral partitioning of random graphs. In *FOCS 2001: Proc. 42nd IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 529–537.
[28] Talagrand, M. (1996) A new look at independence. *Ann. Probab.* **24** 1–34.
[29] Turner, J. (1988) Almost all k-colorable graphs are easy to color. *J. Algorithms* **9** 63–82.
[30] Vu, V. (2007) Spectral norm of random matrices. *Combinatorica* **27** 721–736.
[31] Vu, V. and Wang, K. (2015) Random weighted projections, random quadratic forms and random eigenvectors. *Random Struct. Alg.* **47** 792–821.
[32] Wedin, P.-Å. (1972) Perturbation bounds in connection with singular value decomposition. *BIT Numer. Math.* **12** 99–111.
[33] Xu, R. and Wunsch, D. (2014) *Clustering*, Wiley.