A SIMPLE SVD ALGORITHM FOR FINDING HIDDEN PARTITIONS

VAN VU

Abstract. 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 coloring, finding a hidden bipartition etc.

In this paper, we provide a simple SVD algorithm for this purpose, answering a question of McSherry. This algorithm is easy to implement and works for sparse graphs under optimal density assumptions.

1. The problem and a new algorithm

The hidden partition problem is the following. L et X be a set of n vertices with a partition $X = \cup_{i=1}^{k} X_i$; for all $1 \leq i \leq j \leq n$ 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 coloring, clustering etc (see, for instance, [1, 2, 3, 6, 7, 8, 10, 12, 13, 16, 18, 20, 14, 23, 21] and the references therein). In what follows, we refer to X_i as clusters.

In an influential paper [26], Mc Sherry 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 [26].

The general idea of [26] (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 favorite to implement.

1991 *Mathematics Subject Classification.* 68Q87, 68W20, 60C05.

V. Vu is supported by research grants DMS-0901216 and AFOSAR-FA-9550-09-1-0167.

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 are used under the assumption $n \to \infty$. All explicit constants (such as the 4 above) are adhoc 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 H. The main technical part of Mc Sherry's algorithm is a subroutine called *CProj* (Combinatorial Projection), which creates H 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 $T_i := \{u | \|P_{\hat{A}^T}(\hat{A}^T_{v_i} - \hat{A}^T_u)\| \le \tau\}$, 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: Mc Sherry's algorithm

```
(1) Randomly partition the set $\{1, \dots, n\}$ 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 this two matrices to produce two projections using CPROJ, thinking of their columns as nodes.)
(2) Let $P_1 = CProj(\hat{B}), P_2 = CProj(\hat{A})$ and compute $\bar{H} = [P_1(\hat{A}) | P_2(\hat{B})]$.
(3) Run Clustering by Distances on the projected points.
```

For more details about this algorithm, we refer the reader to [26].

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

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

where the minimum is taken over all pairs u, v belonging to different clusters. Mc Sherry proved [26]

Theorem 1. *Assume that $\sigma^2 \gg \log^6 n/n$ is an upper bound on the variances of the entries. There is a constant $C > 0$ such that if*

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

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

The main open question raised by Mc Sherry in [26] is to find a more natural and simpler algorithm, which does not involve the subroutine CPROJ (see [26, Section 4.4]). In this paper, we address this problem by presenting and analyzing a spectral algorithm, which, in many cases, work under optimal density assumption. Both the algorithm and the analysis are simple (the proof is only few pages). Furthermore, our 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.

We managed to push the bound $\sigma^2 \gg \log^6 n/n$ to $\sigma^2 \gg \log n/n$, which is optimal.The bound $\log^6 n/n$ in Theorem 1 comes from a technical result concerning random matrices by Füredi and Komlos [17] and the analysis in [26] is sharp with respect to this result. It seems hard to improve upon F?̈redi-Komlos' result to get the $\log n$ bound (such an improvement would be interesting on its own right), and our new density bound was obtained via a different rout.

The main new technical ingredient in our analysis is Lemma 10 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. Our algorithm can be adjusted to work at this density as well, but the detail is involved and we defer it to a future paper.

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 would be $\hat{P}_k$ (SVD), which leads to the algorithm below

```
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 most practitioners of the spectral method would think of, it is hard to analyze in the general case. In what follows, we analyze a slightly more technical alternative, SVD II, which is a variant of an algorithm proposed in [26, Section 1].

```
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 $Y$ 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 the correlation. (A careful reading of [26] reveals that one also need an extra partition in Algorithm 2 to make the analysis go through; in particular the proof of [26, Theorem 12] needs modification, as $\hat{A}$ 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 $Y_2^1, \dots, Y_2^l$, 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 $X_1, \dots, X_k$ at the end. We omit the details.

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

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

- *Condition 1.* $\Delta \geq C(\sigma\sqrt{\frac{n}{s}} + \sqrt{\log n})$.
- *Condition 2.* $\Delta \geq C\Big(\sigma\sqrt{\frac{n}{s}} + \sqrt{k}\big(\sigma\sqrt{\log n} + \frac{\log n}{\sqrt{s}} + \frac{\sigma\sqrt{n}\log n}{\lambda}\big)\Big)$.

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

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

for some constant c.

The conditions on Δ in Theorems 1 and 2 are incomparable (see also Theorem 5 below for a comparison). The lower bound $\sigma^2 \geq C\log n/n$ is optimal, up to the value of C. If $\sigma^2 < (1-\epsilon)\log n/n$, then with high probability, there are linearly 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 that it comes from the hidden structures or not), then one can go below $\log n/n$; see for instance [1, 9]. 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 $|X_i \backslash X_i'| \leq \epsilon |X_i|$. Similarly, we say that a geometric representation of X is ϵ-perfect if there are points $x_1, \dots, x_k$ with distance at least $8r$ from each other so that at least $(1-\epsilon)|X_i|$ points from X_i has distance at most r to x_i.

Theorem 3. *Given $\epsilon > 0$, there is a constant $C > 0$ such that the following holds. If $\sigma^2 \geq C\frac{\log n}{n}, s \geq C\log n$ and*

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

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

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

Lemma 4. *For arbitrary positive constants ϵ, γ, k, let $\delta := \frac{\epsilon}{k + (k-1)(1+\gamma)}$. Given an δ-perfect representation of γ-balanced sets $X_1, \dots, X_k$, 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 4, see Appendix B. In what follows, we refer to this algorithm as Approximate Clustering. Combining Theorem 3 and Lemma 4, we have the following algorithm and theorem.

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 5. *Given constants $\epsilon, \gamma, k > 0$, 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 $\sigma^2 \geq C\frac{\log n}{n}$, $s \geq C \log n$ and*

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

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

The advantage of Theorem 5 is that its assumption on Δ is both simpler and stronger than that of Theorems 1 and 2. The caveat here is that the partition is only ϵ-correct. It has turned out, however, 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 [9, Theorem 1.1], which also improves upon Theorem 1, using an adaptive algorithm. The setting of [9, 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 $\log^2 \frac{n}{s}$; we do not have this assumption. Assumption $R2$ requires $s \geq \log^{30} n$; we require $s \geq C \log n$ (in fact, $s \geq C$ is sufficient if we are satisfied with success probability .99 instead of $1 - o(n^{-1})$). The main assumption $R3$ is a lower bound on Δ of the form

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

This assumption and the corresponding assumption in Theorem 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 5 is the weakest. The proofs in [9] also used spectral technique, but seem more delicate and longer than ours; see [9] for more details.

Let us now consider the performance of SVD II and SVD III on 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. Condition 1 becomes

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

which is satisfied if $s \geq C(\sqrt{np} + \sqrt{\log n})$. As $np = \Theta(\sigma^2 n) = \Omega(\log n)$, this simplifies to $s \geq C\sqrt{np}$.

Corollary 6. *There is a constant C such that for any $p \geq C\frac{\log n}{n}$ and $s \geq C\sqrt{np}$, SVD II finds the hidden clique of size s with probability $1-o(1)$.*

This result is comparable to [26, Corollary 3]. The first polynomial time algorithm for hidden clique 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 $\epsilon > 0$, at the cost of pushing the running time to $n^{f(\epsilon)}$, where $f(\epsilon)$ 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 [13].

Hidden Bipartition. Let the two densities be $.99 \geq p > q > 0$. 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 2 requires $\frac{p-q}{p^{1/4}} \geq C\sqrt{\frac{\log n}{n}}$.

Corollary 7. *There is a constant C such that the following holds Let $.99 > p > q \geq C\log n/n$ be edge densities such that $\frac{p-q}{p^{1/4}} \geq C\sqrt{\frac{\log n}{n}}$ then SVD II finds the hidden bipartition with probability $1-o(n^{-1})$.*

The best known condition on Δ is $\frac{p-q}{\sqrt{p}} \geq C\sqrt{\frac{\log n}{n}}$, under stronger density assumptions; see [7, 10, 26]. We can obtain this bound on Δ with Theorem 5 and an extra idea. Let us first apply Theorem 5. The condition on Δ has become $\frac{p-q}{\sqrt{p}} \geq C\sqrt{\frac{\log n}{n}}$. Thus, we have

Corollary 8. *For any $\epsilon > 0$ there is a constant C such that the following holds Let $.99 > p > q \geq C\log n/n$ be edge densities such that $\frac{p-q}{\sqrt{p}} \geq C\sqrt{\frac{\log n}{n}}$ then SVD III finds an ϵ-correct partition with probability at least $1-\epsilon$.*

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 coloring each edge Red or Blue with probability half, independently. First use the Red part as input to recover an ϵ-correct partition for some small ϵ (say $\epsilon = .1$). 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 color 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 .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 neighbors (of $u$) in $X'_i$ is less than the number of Blue neighbors (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 9. *For any $\epsilon > 0$ there is a constant C such that the following holds Let $.99 > p > q \geq C \log n/n$ be edge densities such that $\frac{p-q}{\sqrt{p}} \geq C\sqrt{\frac{\log n}{n}}$. Then algorithm Hidden Bipartition solves the hidden bipartition problem with probability at least $1 - \epsilon$.*

We prove Corollary 9 in Section 4. This corollary is comparable with [26, Corollary 1], but with a better (optimal) density assumption. The first result on Hidden Bipartition was obtained by Bui et. al. [8] and Dyer and Frieze [14] 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 [10] analyzed 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's paper [9] where the author found approximate partitions. In Section 4, we will also discuss an analogue of Corollary 9 for the hidden coloring problem.

The rest of the paper is organized as follows. In the next section, we present a few technical lemmas, including Lemma 10 mentioned above. Then we prove Theorem 2 in Section 3. In Section 4, we prove Theorem 3 and Corollary 9 and discuss related results. The proofs of Lemma 10 and Lemma 4 will be presented in the appendix.

2. Technical lemmas

Lemma 10 (Projection of a Random Vector)**.** *There are constants C_1, C_2 such that the following holds. Let $X = (\xi_1, \ldots, \xi_n)$ be a random vector in $\mathbb{R}^n$ whose coordinates ξ_i are independent random variables with mean 0 and variance at most $\sigma^2 \leq 1$. 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 $\Pi_H \xi$ be the length of the orthogonal projection of ξ onto H. Then*

$$\mathbf{P}(\Pi_H X \geq \sigma\sqrt{d} + C_1\sqrt{\log n}) \leq n^{-3}.$$

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

$$\mathbf{P}(\Pi_H X \geq C_2\sqrt{d}(\sigma\sqrt{\log n} + \alpha \log n)) \leq n^{-3}.$$

We prove Lemma 10 in the appendix.

Lemma 11 (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 $0 \leq p_{ij} \leq 1$. Let $\sigma^2 := \max_{ij} p_{ij}(1 - p_{ij})$. If $\sigma^2 \geq C_0 \log n/n$, then*

$$\mathbf{P}(\|E\| \geq C_0\sigma n^{1/2}) \leq n^{-3}.$$

If $\sigma^2 \geq \frac{\log^4 n}{n}$, the statement is a corollary of [29, Theorem 1.4]. For smaller σ, one can prove this lemma using the ϵ-net approach by Kahn and Szemeredi [22]. We omit the details, which is very similar to the proof of Feige and Ofek for [15, Theorem 1.1].

Lemma 12 (Perturbation bound)**.** *Let M, N be matrices where $\delta := \lambda_k(M) - \lambda_{k+1}(M) > 0$. Then*

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

This lemma is a well known result in numerical linear algebra, known as Davis-Kahan-Wedin theorem; see [5, 11, 32, 19].

3. Proof of Theorems 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 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 $\lambda_k(A) \geq \frac{1}{8}\lambda_k(P)$ 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 denote by $\hat{\mathbf{u}}, \mathbf{u}, e_u$ 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 inviting, a straightforward application of Lemma 12 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 $s \geq C \log n$. 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

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

To bound the second term, we follow an argument from [26] 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 $\lambda_{k+1}(A_k) \leq \lambda_{k+1}(A) + \|E\| = \|E\|$, as A has rank at most k. The spectral norm of the second term is also at most $\|E\|$. Thus, by Lemma 11, by probability at least $1 - n^{-3}$

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

for some constant C_0.

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 $\|M\| \geq \|M\chi_u\|$; thus

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

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

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

for all $u \in X$.

Now we tend to the first term, whose analysis is more involved. By the first part of Lemma 10,

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

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

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

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 $|X_i \cap Z| \geq \frac{1}{3}|X_i| = s_i/3$ for all i. By symmetry, each coordinate in v is repeated at least $s/3$ times, thus $\|v\|_\infty \leq 2s^{-1/2}$. It follows that for any unit vector $v \in \text{Span}(A)$, $\|v\|_\infty \leq 2s^{-1/2}$. Furthermore, by Lemma 12 and Lemma 11, we have with probability $1 - o(n^{-2})$ that

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

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

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

Using the second part of Lemma 10, we conclude that with probability $1 - o(n^{-2})$,

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

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

Let us now show that with high probability, $\lambda_k(A) \geq \frac{1}{8}\lambda_k(P)$. 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 $1 - n^{-2}$

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

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

We use the fact that for any matrix M of rank k $\lambda_k(M) = \inf_{rank(M')=k-1} \|M - M'\|_F$. For simplicity, let us assume for a moment that $|X_i \cap Y| = |X_i|/2$. Let $\tilde{P}'$ be the matrix that define $\lambda_k(\tilde{P})$. 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 $k-1$ and

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

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

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

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

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 $\lambda_k(P) \leq 2\lambda_k(\tilde{P})$, which leads to $\lambda_k(P) \leq 8\lambda_k(A)$, as desired.

4. Approximate Solutions and Upgrading

4.1. **Proof of Theorem 3.** We follow the proof of Theorem 2. The key is to bound $\|P_{\hat{A}_k} e_u\|$. Recall that

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

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

4.2. **Hidden bipartition: Proof of Corollary 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 5, we obtain an .1-correct partition with probability at least $1-\epsilon/2$, provided that

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

for some sufficiently large constant C. By Chernoff bound, one can prove that with probability $1-o(n^{-1})$, all degrees in the Red graph are at most $n/50 = .02n$. 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, however, have to be a bit careful since the Blue graph is not entirely independent from the Red one. Observe that

- If e is an edge in the Red graph, then it cannot be an edge in the Blue graph.
- If e is not an edge in the Red graph and both end points are in X_1 (or X_2), then e an edge in the Blue graph with probability $p_1 := \frac{p/2}{1-p/2}$.
- 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 an edge in the Blue graph with probability $q_1 := \frac{q/2}{1-q/2}$.
- Conditioned on the Red graph, the events of non-edges in the Red graph become edges in the Blue graph are mutually independent.

Let $X_1' \cup X_2'$ be the .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 $\epsilon = .1$), A_i have size at least $\frac{1}{2}(1-\epsilon)n = .45n$, for $i = 1, 2$. It follows that $.55n \geq |X_i'| \geq .45n$ and $|B_i| \leq .1n$.

Consider $u \in X_1'$. Let $N_R(u)$ be the set of neighbors of u in the Red graph and $d_i(u)$ be the number of neighbors 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 iid indicator variables with mean p_1 and $\mu(y)$ are iid indicator variables with mean q_1. Furthermore,

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

We have

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

As $N_R(u) \leq .02n$ and $|A_i| \geq .45n$, $B_i| \geq .1n$, it follows that

$$\mathbf{E}D \geq (p_1 - q_1)(|A_1 \backslash N_R(u)| - |B_1 \backslash N_R(u)|),$$

where the LHS is at least

$$(p_1 - q_1)(.45n - 0.02n - 0.1n) \geq .3n(p_1 - q_1) \geq .15n(p - q) \geq 5 \log n,$$

provided that the constant C in Corollary 9 is sufficiently large. Applying Chernoff 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 Biparittion succeeds with probability at least

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

concluding the proof.

Algorithm 7: Hidden Coloring

```
(0) Randomly color 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 $\frac{1}{k}$-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 neighbors of $u$ in $X'_i$ is at least $\frac{.5}{k-1}$ 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 neighbor in $Y_i$ in the Green graph.
```

4.3. **Hidden coloring.** We can obtain the following analogue of Corollary 9.

Corollary 13. *For any constant $\epsilon > 0$ there is a constant C such that the following holds Let $.99 > p \geq C \log n/n$. Then algorithm Hidden Coloring solves the hidden coloring problem with probability $1 - \epsilon$.*

Many researchers have worked on the problem of coloring random graphs which have k- colorings. Kucera [23], Turner [27], and Dyer and Frieze [11] presented effective algorithms for dense graphs. Prior to Mc Sherry's paper [26], Blum and Spencer [4] and Alon and Kahale [1] demonstrate algorithms that color random sparse graphs properly with high probability. Corollary 13 is comparable to [26, Corollary 2], with a better (optimal) density bound. If we aim for an approximate recovery (or an optimal coloring which may not comes from the hidden one) then there are algorithms which work for lower density $\Omega(1/n)$; see [1, 9], in particular the discussion in [9, 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 $|Y_i| \geq |X_i|/2$ with high probability.) It is easy to see (again by Chernoff's bound) that the only vertices which have no neighbors in Y_i (in the Green graph) are the vertices of X_i. This conclude the proof.

Using the same idea, one can handle a common generalization of Hidden Bipartition and Hidden Coloring. Let $X_1, \dots, X_k$ 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 14. *For any $\epsilon > 0$ there is a constant C such that the following holds Let $.99 > p, q \geq C \log n/n$ be edge densities such that $\frac{|p-q|}{\sqrt{p}} \geq C\sqrt{\frac{\log n}{n}}$. Then one can recovers the partition with probability at least $1 - \epsilon$ by an efficient algorithm.*

This corollary is a variant of [26, Corollary 1], again with a weaker density bound; we omit the details.

Appendix A. Proof of Lemma 10

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

$$\mathbf{P}(\Pi_H X \geq \mu + t) \leq 2\exp(-t^2/4)$$

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

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

Combining this with the concentration inequality, it is not hard to show that $\mu \leq \sigma d^{1/2} + O(1)$, concluding the proof of the first part of the lemma. The reader can also check [30] for a detailed discussion on inequalities of this type.

To prove the second part, notice that if $v_1, \dots, v_d$ 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 15. *Let $(a_1, \dots, a_n)$ be real numbers such that $\sum_i a_i^2 = 1$ and $|a_i| \leq \alpha$ for all i. Let ξ_i be independent random variables with mean 0 and $\mathbf{E}|\xi_i|^k \leq \sigma^2$ for all $k \geq 2$. Let $S := \sum_{i=1}^n a_i \xi_i$. Then*

$$\mathbf{P}(|S| \geq 4(\sigma\sqrt{\log n} + \alpha \log n) \leq 2n^{-3}.$$

To prove Claim 15, notice that for any $0 < t \leq \alpha^{-1}$ we have

$$\mathbf{E}\exp(tS) = \prod_i \mathbf{E}\exp(ta_i\xi_i) = \prod_i (1 + \frac{\sigma^2 a_i^2 t^2}{2!} + \frac{t^3 a_i^3 \mathbf{E}\xi_i^3}{6!} + \dots)$$

Since $\mathbf{E}\xi_i^k \leq \sigma^2$ for all $k \geq 2$ and $t|a_i| \leq 1$, the right most formula is

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

Markov's inequality yields

$$\mathbf{P}(S \geq T) \leq \exp(-tT + t^2\sigma^2).$$

To optimize the RHS, let us consider two cases

Case 1. $\sigma \geq \alpha\sqrt{\log n}$. Take $T = 4\sigma\sqrt{\log n}$ and $t = \frac{\sqrt{\log n}}{\sigma} \leq \alpha^{-1}$. 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, $-tT + t^2\sigma^2 \leq -4\log n + \log n = -3\log n$.

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

Appendix B. Approximate Clustering

Algorithm 8: Approximate Clustering

```
The input is a $\delta$-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 $\tilde{X}_i', 1 \leq i \leq k$, 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.
```

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

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 centers $x_1, \dots, x_k$. If a point w is of distance at most r to x_j, then the ball of radius $2r$ around w contains and at least $(1-\delta)s$ points from X_j and at most $(k-1)\delta s$ points from other X_l's. 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 $\delta := \epsilon/k$, $X_i's$ satisfies $|X_i \backslash X_i'| \leq \epsilon s$ 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 ($|X_i| \leq (1+\gamma)s$) 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's. 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 $\delta := \frac{\epsilon}{k+(k-1)(1+\gamma)}$, $X_j's$ satisfies $|X_j \backslash X_j'| \leq \epsilon s$ 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

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

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

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

As a consequence, we obtain the following lemma, which implies Lemma 4.

Lemma 16. *With probability at least $1-n^{-2}$, Approximate Clustering produces an ϵ-correct partition after at most $3\rho^{-1}\log n \leq 3(\frac{e(1+\gamma)}{1-\delta})^k \log n$ cycles, given $\delta = \frac{\epsilon}{k+(k-1)\gamma}$.*

Acknowledgement. The author would like to thank the referee for his careful reading and numerous suggestions which greatly improve the quality of the paper.

References

[1] N. Alon, N. Kahale. *A spectral technique for coloring random 3-colorable graphs.* SIAM Journal on Computing, 26(6):1733–1748, 1997.

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

[3] Y. Azar, A. Fiat, A. Karlin, F. McSherry, J. Saia. *Spectral analysis of data.* Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, 619–626, 2001.

[4] Z. Bai, J.W. Silverstein. *Spectral analysis of large dimensional random matrices.* Springer, 2009.

[5] R. Bhatia. *Matrix analysis*, volume 169. Springer Verlag, 1997.

[6] A. Blum, J. Spencer. *Coloring random and semi-random k-colorable graphs.* Journal of Algorithms, 19 no. 2, 204–234, 1995.

[7] R.B. Boppana. *Eigenvalues and graph bisection: An average-case analysis.* Proceedings of the 28th Annual Symposium on Foundations of Computer Science, 280–285, 1987.

[8] T.N. Bui, S. Chaudhuri, F.T. Leighton, M. Sipser. *Graph bisection algorithms with good average case behavior.* Combinatorica, 7 no. 2, 171–191, 1987.

[9] A. Coja-Oghlan, *Graph partitioning via adaptive spectral techniques,* Combinatorics, Probability and Computing (2010), 19, 227-284.

[10] A. Condon, R.M. Karp. *Algorithms for graph partitioning on the planted partition model.* Random Structures and Algorithms, 18 no. 2, 116–140, 2001.

[11] C. Davis, W.M. Kahan. *The rotation of eigenvectors by a perturbation. iii.* SIAM Journal on Numerical Analysis, 7(1):1–46, 1970.

[12] Y. Dekel, O. Gurel-Gurevich, Y. Peres. *Finding hidden cliques in linear time with high probability.* Proceedings of the Eighth Workshop on Analytic Algorithmics and Combinatorics, 67–75, 2011.

[13] Y. Deshpande, A. Montanari. *Finding hidden cliques of size $\sqrt{N/e}$ in nearly linear time.* Available at `arXiv:1304.7047 [math.PR]`.

[14] M.E. Dyer, A.M. Frieze. *Fast solution of some random NP-hard problems.* 27th Annual Symposium on Foundations of Computer Science, 221–336, 1986.

[15] U. Feige and E. Ofek, *Spectral techniques applied to sparse random graphs,* Random Structures Algorithms 27 (2005), no. 2, 251275.

[16] U. Feige, R. Krauthgamer. *Finding and certifying a large hidden clique in a semirandom graph.* Random Structures and Algorithms, 16 (2): 195–208, 2000.

[17] Z. Furedi and J. Komlos, *The eigenvalues of random symmetric matrices*, Combinatorica 1981, Volume 1, Issue 3, 233-241.

[18] U. Feige, D. Ron. *Finding hidden cliques in linear time.* DMTCS proc. AM, 189–204, 2010.

[19] G.H. Golub, C.F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 1996.

[20] M. Jerrum, G.B. Sorkin. *Simulated annealing for graph bisection.* IEEE Symposium on Foundations of Computer Science, 94–103, 1993.

[21] R. Kannan, S. Vempala. *Spectral algorithms.* Now Publishers Inc, 2009.

[22] J. Kahn and E. Szemerédi, STOC 1989.

[23] L. Kučera. *Expected behavior of graph coloring algorithms.* Fundamentals of computation theory, pages 447–451. Springer, 1977.

[24] D.W. Matula. *The employee party problem.* Notices Am. Math. Soc., 19, pp. A–382, Feb.1972.

[25] C. McDiarmid. *Concentration.* In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, B. Reed, eds.: *Probabilistic Methods for Algorithmic Discrete Mathematics*, Springer, New York, 195–248, 1998.

[26] F. McSherry. *Spectral partitioning of random graphs.* Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science - FOCS, 529–537, 2001.

[27] J. Turner. *Almost all k-colorable graphs are easy to color,* Journal of Algorithms 9 (1988), no. 1, 63?82.

[28] M. Talagrand. *A new look at independence.* Ann. Probab., 24 no. 1, 1–34, 1996.

[29] V. Vu. *Spectral norm of random matrices.* Combinatorica, 27(6):721–736, 2007.

[30] V. Vu and K. Wang, *Random weighted projections, random quadratic forms and random eigenvectors*, to appear in Random Structures and Algorithms.

[31] R. Xu and D. Wunsch, Clustering, Wiley 2014.

[32] P.-Å. Wedin. *Perturbation bounds in connection with singular value decomposition.* BIT Numerical Mathematics, 12(1):99–111, 1972.

Department of Mathematics, Yale, New Haven, CT 06520

E-mail address: van.vu@yale.edu