

Correlation Clustering with Noisy Input

Claire Mathieu*

Warren Schudy*

Abstract

Correlation clustering is a type of clustering that uses a basic form of input data: For every pair of data items, the input specifies whether they are similar (belonging to the same cluster) or dissimilar (belonging to different clusters). This information may be inconsistent, and the goal is to find a clustering (partition of the vertices) that disagrees with as few pieces of information as possible.

Correlation clustering is APX-hard for worst-case inputs. We study the following semi-random noisy model to generate the input: start from an *arbitrary* partition of the vertices into clusters. Then, for each pair of vertices, the similarity information is corrupted (noisy) independently with probability p . Finally, an adversary generates the input by choosing similarity/dissimilarity information arbitrarily for each corrupted pair of vertices.

In this model, our algorithm produces a clustering with cost at most $1 + O(n^{-1/6})$ times the cost of the optimal clustering, as long as $p \leq 1/2 - n^{-1/3}$. Moreover, if all clusters have size at least¹ $c_1\sqrt{n}$ then we can *exactly* reconstruct the planted clustering. If the noise p is small, that is, $p \leq n^{-\delta}/60$, then we can *exactly* reconstruct all clusters of the planted clustering that have size at least $3150/\delta$, and provide a certificate (witness) proving that those clusters are in any optimal clustering.

Among other techniques, we use the natural semi-definite programming relaxation followed by an interesting rounding phase. The analysis uses SDP duality and spectral properties of random matrices.

1 Introduction

Clustering is an important tool for analyzing large data sets. Correlation clustering [6, 5] is a type of clustering that uses a very basic form of input data: indications that certain pairs of vertices (pairs of data items) belong in the same cluster, and certain other pairs belong in different clusters [5]. Unfortunately the information is not necessarily consistent, possibly claiming for example that “cat” is similar to “dog” and “dog” is similar to “bog” but “cat” is not similar to “bog”. The goal is to

find a clustering, that is, a partition of the vertices, that agrees with as many pieces of information as possible (maximization) or disagrees with as few as possible (minimization). This has applications in data mining, e.g. the work of Cohen and Richman [11].

As correlation clustering is hard to solve exactly, many previous papers focused on approximation algorithms [5, 10, 12, 1, 25]. For the purpose of approximation, note that the maximization and minimization goals are not equivalent: minimizing is harder. How hard is it to minimize disagreements? The best known algorithm has approximation ratio $O(\log n)$ [12] (n is the number of items (vertices) being clustered). However, a constant approximation ratio (currently 2.5) is achievable in the restricted *complete information* setting that assumes that information is available for every pair of vertices [1, 26, 5]; this result is essentially best possible (except for lowering the value of the constant) since the complete information problem is APX-hard [10]. Yet, in this paper we concentrate on the minimization version with complete information.

How can one get around this APX-hardness result to provably get a clustering with value within $1 + \epsilon$ of optimal? Additional assumptions are needed. Thus, there is a PTAS that relies on the additional assumption that the number of clusters is a constant [16]. In this paper, motivated by the data mining context, our angle is to assume that the input comes from a noisy model defined as follows. To generate the input, start from an arbitrary partition \mathcal{B} of the vertices into clusters (*base clustering*). Then, each pair of vertices is perturbed independently with probability p . In the *fully-random* model, the input is generated from \mathcal{B} simply by switching every perturbed pair. In the *semi-random* variant, an adversary controls the perturbed pairs and decides whether or not to switch them. Such noisy models are hardly new: they have been studied for many graph problems such as complete information feedback arc set [9], maximum bisection [8], k -coloring, unique games [21], maximum clique [18, 22, 3, 14], and even for correlation clustering itself [6, 5, 23, 24, 13]. Indeed, studying the noisy model theoretically led [6] to a heuristic algorithm that they used to successfully cluster real gene expression data.

We note that, prior to this work, noisy correlation

*Department of Computer Science, Brown University. Supported in part by NSF Grant CCF-0728816. Email: {claire, ws}@cs.brown.edu.

¹Throughout this work c_1, c_2, c_3, \dots denote absolute constants.

clustering had only been studied in the fully-random model [6, 5, 23, 24]. Typically, results assume that all clusters have size bounded below, by $c_1 n$ [6] or by $c_2 \sqrt{n \log n} / (1/2 - p)^{1+\epsilon}$ [24]. From a slightly different angle, [5] makes no assumption on minimum input cluster size but finds all clusters of the base clustering that have size at least $c_3 p \sqrt{n \log n} / (1/2 - p)^2$.^{2,3} Their algorithm can be used to yield a $1 + o(1)$ approximation for $p = \omega((\log n)/n)$. (More precisely, the additive error is $O(pn^{3/2} \sqrt{\log n} / (1/2 - p)^2)$ for $\sqrt{\log n/n} < p < 1/2$ and $O(n \log n)$ for smaller p .)

In contrast to our work, none of those prior results on noisy correlation clustering *certify* optimality of the output clustering. Certification is a highly desirable feature, as explained by Feige and Krauthgamer [14]:

“since average case algorithms do not have an a priori guarantee on their performance, it is important to certify that the algorithm is indeed successful on the particular instance at hand.”

Our analysis relies on the validity of the noisy model, hence that is also important for us. All of our results address that issue and come with accompanying certificates.

Our results. We design a simple approximation algorithm. It uses a semi-definite programming relaxation and, for rounding, a constant factor approximation algorithm [26, 1]. Note that this semidefinite program was already used by [25], but our rounding algorithm is completely different from theirs. We then proceed to prove that it has several desirable features.

Our first result bounds the cost of the output (Theorem 2.1). In the worst-case model, that cost is a constant-factor approximation. In the noisy model,⁴ let OPT (resp. OPT_{full}) denote the optimum cost in the semi-random model (resp. fully-random model). In the fully random model, our algorithm is a PTAS. In the semi-random variant, our algorithm yields cost at most $OPT + O(n^{-1/6})OPT_{full}$ with high probability (over the random perturbation). Our algorithm also produces a lower-bound, proving that the random perturbation was “random enough” for this high probability event to occur.

Our second result (Theorem 2.2) shows one circumstance in which the base clustering can be reconstructed *exactly*. We show that if $p \leq 1/3$ and all clusters have

size at least $c_5 \sqrt{n}$ then the base clustering is the unique optimum solution of the natural SDP relaxation of correlation clustering, hence is the output of our algorithm.

Our third result (Theorem 2.3) analyzes the small noise regime, $p \leq n^{-\delta}/60$ for some δ . (Such a regime makes sense in applications in which perturbations are rare, for example if they correspond to mutations). Then, we have an algorithm which finds all clusters of size $\Omega(1/\delta)$, even in the semi-random model. It also produces a certificate (witness) proving that the clusters found are part of all optimal clusterings. For example, consider a base clustering consisting of clusters of (large enough) constant size, and let $p = n^{-1/10}/60$. For every vertex u , we can determine which $\Theta(1)$ pairs uv are correctly labeled as similar in the input, among the sea of $\Theta(n^{9/10})$ pairs uv incorrectly labeled as similar.

Our analysis uses probabilistic and combinatorial arguments, spectral properties of random matrices (Lemma 3.12), and semidefinite programming duality (Section 4).

Comparison with previous results.

1. We give the first algorithm that achieves a $1 + o(1)$ approximation even when there are arbitrarily few noisy edges (so that OPT is small) and clusters are arbitrarily small (so that reconstructing them exactly is impossible.) (Theorem 2.1)
2. In the semi-random model, as it turns out, for constant p our additive error is $O(n^{3/2})$; the best result previously known had additive error $O(\epsilon n^2)$ (in the worst-case model) [16].
3. In the fully random model, compared to previous work [5, 24] we find clusters a factor $\sqrt{\log n}$ smaller for constant p . (For example, when $p = 1/3$ we reconstruct the base clustering when all clusters are size at least $c_5 \sqrt{n}$ rather than $c_6 \sqrt{n \log n}$.) We improve additive error by the same factor.
4. Not only does the output have cost that, with high probability, is within $1 + o(1)$ of optimal, but our algorithm also produces a deterministic lower-bound witnessing that fact. In other words our algorithm knows whether or not its input is “sufficiently random.”
5. Let $\delta > 0$. If $p \leq n^{-\delta}/60$ we give the first algorithm that exactly reconstructs every cluster of size $3150/\delta$.

Our work is worse than previous work in two ways: First, the algorithm for item 5 has impractical runtime even for modest $\delta = 1/3$. Second, when p is a constant, we, like [22], can only find the base clusters exactly if *all*

²A straightforward variation works in the semi-random model but only finds clusters of size at least $c_4 np^2 \sqrt{\log n} / (1/2 - p)^2$.

³McSherry [23] also solved this problem, however their results depend on the number of clusters and are worse than [24] if the number of clusters is large.

⁴Assuming $p \leq 1/2 - n^{-1/3}$

Algorithm 1 MAINCLUSTER algorithm

Input: graph $G = (V, E)$ Output: clustering *Output*

Let \hat{E} be E with edges that are not part of triangles removed.

Call algorithm SDPCLUSTER on input (V, \hat{E}) , yielding clustering \mathcal{A} .

Let U denote the vertices in singleton clusters of \mathcal{A} .

Use maximum matching to compute an optimal clustering of U into clusters of size at most 2.

are size at least $c_5\sqrt{n}$. Bansal Blum Chawla [5]’s result is incomparable, since they find the clusters of size at least $c_7\sqrt{n\log n}$, even if the base clustering is a mixture of large and small clusters.

2 Algorithms

The input is an undirected graph $G = (V, E)$, where $\{u, v\} \in E$ iff we have information that u and v are similar, and $\{u, v\} \notin E$ iff we have information that u and v are dissimilar.⁵ A *clustering* is a partition of the vertices into *clusters*. To any clustering, we associate the induced graph, which has an edge between every pair of intra-cluster vertices (including self-loops), and an associated matrix, the adjacency matrix of the induced graph. We use the terminology *vertex pair* (VP) to avoid ambiguity with *edge*, which refers to vertex pairs that have an edge between them in some graph.

The *correlation clustering problem* consists of finding a clustering \mathcal{A} of V minimizing $d(E, \mathcal{A})$, where $d(\cdot, \cdot)$ denotes the Hamming distance between two graphs (viewed as sets of edges), or equivalently, half of the ℓ_1 distance between the associated adjacency matrices: for symmetric matrices M and N , $d(M, N) = \frac{1}{2} \sum_{u,v} |M_{uv} - N_{uv}|$.

Fix the error parameter p . Our noisy model assumes that the input graph $G = (V, E)$ is generated by perturbing some unknown base clustering \mathcal{B} as follows: identify \mathcal{B} with the associated graph. Then, in the *fully random model*, for each pair of vertices, the information is flipped independently with probability p (in other words, to go from \mathcal{B} to the input graph we flip every edge and every non-edge independently with probability p). In the *semi-random model*, for each pair of vertices, the information is corrupted (noisy) independently with probability p ; then, an adversary generates the input

⁵Our model requires complete information so we use absence of an edge to represent dissimilarity, not an edge labeled “-” as in previous work.

Algorithm 2 SDPCLUSTER algorithm

Input: graph $G = (V, F)$ Output: a clustering \mathcal{A}

Compute an optimal solution X^* to the following semi-definite program:

$$\min d(X, F) \text{ s.t. } \begin{cases} X_{ii} = 1 & \forall i \\ X_{ij} \geq 0 & \forall i \neq j \\ X_{ij} + X_{jk} - X_{ik} \leq 1 & \forall i \neq j \neq k \\ X \text{ pos. semi-definite} \end{cases}$$

Let $U \leftarrow V$.

while U is non-empty **do**

 Pick a pivot vertex v uniformly at random from U
 $A \leftarrow \{v\}$

 For each vertex u of $U \setminus \{v\}$, add u to A independently with probability X_{uv}^* .

 Output the resulting cluster A , and let $U \leftarrow U \setminus A$.

end while

graph by choosing similarity/dissimilarity information arbitrarily for each corrupted pair of vertices (when the adversary always flips the information of every corrupted pair, the resulting input is exactly the input generated in the fully random model).

We let \mathcal{B} denote the unknown base clustering, $G = (V, E)$ the input graph generated from \mathcal{B} in the semi-random model, and $G_{full} = (V, E_{full})$ the corresponding graph generated from \mathcal{B} in the fully-random model. OPT denotes the cost of the optimal solution for input G , and, in the semi-random model, OPT_{full} denotes the cost of the optimal solution for the associated fully-random input.

Throughout the paper, when we write that an event occurs *with high probability* (w.h.p.), we mean that it holds with probability at least $1 - n^{-\alpha}$ for some $\alpha > 0$ (over the randomness in E_{full}). Let $\mathbf{E}_{alg}[\cdot]$ denote expectation over the random choices made by algorithm MAINCLUSTER.

Our first theorem shows that the MAINCLUSTER algorithm has three desirable properties. It is a constant-factor approximation in the adversarial model, a $1+o(1)$ approximation in the planted model, and produces a lower bound certifying its approximation factor.

THEOREM 2.1. (MAIN THEOREM) *Algorithm MAINCLUSTER runs in polynomial time and is such that:*

1. *For any input graph G , $\mathbf{E}_{alg}[\text{Cost}(\text{OUT})] \leq 3.5 \text{ OPT}$.*
2. *In the semi-random model, if $p \leq 1/2 - n^{-1/3}$ then, with high probability over the noisy model, $\mathbf{E}_{alg}[\text{Cost}(\text{OUT})] \leq \text{OPT} + O(n^{-1/6})\text{OPT}_{full}$.*

3. One can compute a lower bound L on OPT . In the fully random model, if $p \leq 1/2 - n^{-1/3}$ then, with high probability over the noisy model, $E_{alg}[Cost(OUT)] \leq L + o(L)$.

A 2.5-approximation algorithm was already known so the first part of Theorem 2.1 is a contribution to the understanding of the MAINCLUSTER algorithm, not to the understanding of the correlation clustering problem.

The noisy model is motivated by the view that \mathcal{B} is the ground truth, so it makes sense to ask if we can, not just approximate the objective function, but actually recover \mathcal{B} . We give two results of this type. Our next theorem shows that SDPCLUSTER recovers the base clustering exactly whenever all base clusters are sufficiently large.

THEOREM 2.2. *In the semi-random model if $p \leq 1/3$ and all clusters have size at least $c_5\sqrt{n}$ then \mathcal{B} is with high probability the unique optimum solution of the SDP used in SDPCLUSTER.*

Our next theorem shows that when the noise is small it is possible to reconstruct all base clusters of super-constant size.

THEOREM 2.3. (LARGE CLUSTER THEOREM) *Assume that $p \leq n^{-\delta}/60$ for some δ . Then in the semi-random model Algorithm 3 outputs a set of clusters \mathcal{A} such that:*

1. *W.h.p. the output clusters \mathcal{A} are exactly the clusters of the base clustering that have size $\geq 3150/\delta$.*
2. *The algorithm certifies that for any optimal clustering, its clusters of size $\geq 3150/\delta$ are exactly the output clusters.*

Almost all of our proofs apply identically in the fully random and semi-random models. For simplicity of exposition we only emphasize the distinction between the models where important.

Remarks

1. A natural generalization of our planted noisy model has input generated by adding noise to an arbitrary graph G rather than a union of cliques \mathcal{B} . Unfortunately this “smoothed” model is hard to approximate; if the noise is less than $n^{-\delta}$ for some $\delta > 0$ this model has no PTAS unless $P=NP$. To see this, take an arbitrary correlation clustering instance with ℓ nodes and non-zero optimum cost, make $\ell^{1/\delta}$ copies of each, yielding a graph with $n = \ell^{1+1/\delta}$ nodes. Then add noise. The number of resulting noisy edges is $O(n^{2-\delta}) = O(\ell^{2/\delta+1-1/\delta}) =$

Algorithm 3 Large Cluster Algorithm (parameterized by integer $s = \lceil 70/\delta \rceil$).

Input: graph $G = (V, E)$.

Let $\mathcal{A} \leftarrow \emptyset$.

For every subset S of V of size s :

Let A be the set of vertices that have at least $|S|/2$ neighbors in S .

If all of the following conditions hold:

1. For all disjoint sets $T, U \subseteq A$, $|T| = |U| = s$, G has at least $.9|T||U|$ edges between T and U .
2. For all sets $T \subseteq A$, $U \subseteq V \setminus A$, $|T| = |U| = s$, G has at most $.1|T||U|$ edges between T and U .
3. $|A| \geq 3|S|$ and A is equal to the set of vertices that have at least $|A|/2$ neighbors in A .

Then add A to the collection \mathcal{A} .

If any of the following properties fails to hold then Output: “Failure”.

1. The sets in \mathcal{A} are disjoint.
2. For every cluster $A \in \mathcal{A}$:
 - (a) Every vertex in A has at least $.9|A|$ neighbors in A .
 - (b) Every vertex not in A has at most $.1|A|$ neighbors in A .
3. For every pair of disjoint vertex sets S, T with cardinalities $|S| = |T| = 6s$, such that no cluster $A \in \mathcal{A}$ intersects both S and T ($\forall A, A \cap S = \emptyset$ or $A \cap T = \emptyset$), there are at most $(.35)|S||T|$ edges between S and T .

Let $\mathcal{A}' = \{A \in \mathcal{A} : |A| \geq 45s\}$.

Output \mathcal{A}' .

$o(\ell^{2/\delta}) = O(\text{new optimum})$, so a PTAS for correlation clustering in this smoothed model would imply a PTAS for adversarial correlation clustering and hence $P=NP$ [10]. This smoothed model may however be interesting for relatively large p such as $p = 1/\text{polylog}(n)$.

2. A random graph $G(n, p)$ with $p = n^{-\delta}$ has max clique of size $\Theta(1/\delta)$ with high probability [7]. Therefore the size of clusters reconstructed by Algorithm 3 is optimal to within constant factors.
3. The *planted clique problem* is the special-case of our problem where all but one of the clusters of \mathcal{B} have size one and the noise only adds edges,

not removes them. The semi-random nature of our planted model allows our model to include the planted clique problem as a special case. The best known result [14] for planted clique and constant p requires the clique to be of size $\Omega(\sqrt{n})$, which matches Theorem 2.2. Theorem 2.2 does *not* subsume previous results for planted clique [14] since it requires that *all* clusters be that large. One interesting open question is how the correlation clustering SDP behaves when some clusters are large and other small.

4. Our results on reconstructing clusters exactly and on approximating the objective function do *not* imply each other. To see that even optimal results for the reconstruction problem are insufficient for approximation, consider when $p = n^{-3/2}$ and the base clusters have size 1 and 2. In this setting is clearly impossible to reconstruct the base clustering from the input data and such a reconstruction would not be a good approximation to the objective even if it were available. To see that approximation is insufficient for reconstruction note that omitting a vertex from a base cluster costs at most the size of that cluster, which is much less than OPT in most circumstances.

3 Proof of Theorem 2.1

3.1 Analysis of SDPCLUSTER Algorithm The following extension of an analysis from [1] forms the starting point of our worst-case analysis of section 3.2 and is also used in section 3.5.

THEOREM 3.1. [1] *Let $G = (V, F)$ be an instance of correlation clustering, and let \mathcal{A} be the resulting output of algorithm SDPCLUSTER. Then, for any clustering \mathcal{C}' , we have:*

$$\mathbf{E}_{alg} [d(\mathcal{A}, F)] \leq 2.5 d(\mathcal{C}', F).$$

Proof. (Proof Sketch) Algorithm SDPCLUSTER is almost identical to algorithm LP-KWIKCLUSTER from [1]. Our SDP (semi-definite program) includes all of the constraints in [1] ($X_{ij} \leq 1$ is implied by $X_{ii} = 1$, $X_{jj} = 1$, and positive semi-definiteness) as well as the positive semi-definite constraint. Integral clusterings satisfy the positive semi-definite constraint, so our SDP is a relaxation. The analysis from [1] applies unchanged to SDPCLUSTER.

The following theorem is very similar to Theorem 3.1, but shows an approximation factor relative to the fractional clustering X^* rather than relative to the input edge set F . The analysis, which uses techniques from [1], is deferred to the full version.

THEOREM 3.2. *Let $G = (V, F)$ be an instance of correlation clustering, and let \mathcal{A} be the resulting output of algorithm SDPCLUSTER. Then, for any clustering \mathcal{C}' , we have:*

$$\mathbf{E}_{alg} [d(\mathcal{A}, X^*)] \leq 3 d(\mathcal{C}', X^*).$$

LEMMA 3.1. *Let u be a vertex with no neighbors in the input (V, F) to algorithm SDPCLUSTER. Then u is in a singleton cluster of the clustering \mathcal{A} obtained by SDPCLUSTER.*

Proof. (Proof Sketch) Let u be a vertex with no neighbors in F . Assume that $X_{uv} > 0$ for some $v \neq u$ in some SDP feasible X . Consider the solution X' obtained from X by setting $X'_{uw} = 0$ for all w . It is easy to see that X' is also SDP feasible and has strictly better objective. So the optimum X^* must have $X^*_{uw} = 0$. Rounding therefore puts u in a singleton cluster.

3.2 Proof of Theorem 2.1 (1)

LEMMA 3.2. *For input $G = (V, E)$, there exists an optimal clustering \mathcal{C} such that for every input edge $\{u, v\}$ that is inside a cluster C of \mathcal{C} of size 3 or more, there exists a vertex $w \in C$ such that $\{u, v, w\}$ is a triangle in the input graph.*

Proof. Let \mathcal{C} be an optimal clustering with the most clusters. Assume, for a contradiction, that there is a cluster $C \in \mathcal{C}$ of size 3 or more, and an input edge $\{u, v\}$ in C such that u and v have no common neighbor in C . Consider the clustering \mathcal{C}' obtained from \mathcal{C} by replacing C with two clusters, $\{u, v\}$ and $C \setminus \{u, v\}$. It is easy to check that \mathcal{C}' is at least as good as \mathcal{C} , so \mathcal{C}' is also optimal but has more clusters, contradicting the definition of \mathcal{C} .

LEMMA 3.3. *Let \mathcal{C} be an optimal clustering for input $G = (V, E)$, let \mathcal{A} be the output of SDPCLUSTER on $\hat{G} = (V, \hat{E})$, and Output denote the clustering output by MAINCLUSTER. Then:*

$$d(\text{Output}, E) \leq d(\mathcal{C}, E) + d(\mathcal{A}, \hat{E}).$$

Proof. Let \mathcal{M} be the matching in the last step of algorithm MAINCLUSTER. Since \mathcal{M} only merges singletons of \mathcal{A} by using edges of G , we have $d(\text{Output}, E) = d(\mathcal{A}, E) - |\mathcal{M}|$. By the triangular inequality, $d(\mathcal{A}, E) \leq d(\mathcal{A}, \hat{E}) + d(\hat{E}, E)$.

Assume, without loss of generality, that \mathcal{C} satisfies Lemma 3.2, and let \mathcal{M}^* be the collection of clusters of \mathcal{C} of size 2. By definition of \hat{G} and by Lemma 3.2, the edges of $G \setminus \hat{G}$ are not in clusters on \mathcal{C} of size 3 or more. So they are either in \mathcal{M}^* or between different clusters of \mathcal{C} . Thus $d(\hat{E}, E) \leq |\mathcal{M}^*| + d_1(\mathcal{C}, E)$, where $d_1(\cdot)$ only takes into account edges that are in $G \setminus \hat{G}$.

Let U denote the vertices that are in singleton connected components of \widehat{G} and hence by Lemma 3.1 in \mathcal{A} . Let \mathcal{M}_1^* be the clusters of \mathcal{M}^* contained in U , and $\mathcal{M}_2^* = \mathcal{M}^* \setminus \mathcal{M}_1^*$. By maximality of \mathcal{M} , we have $|\mathcal{M}_1^*| \leq |\mathcal{M}|$.

Let $\{u_i, v_i\}$, $1 \leq i \leq |\mathcal{M}_2^*|$ denote the edges of \mathcal{M}_2^* , with $v_i \notin U$. Since v_i is in some non-singleton connected component \widehat{G} , there is an edge $\{v_i, w_i\}$ of \widehat{G} . By definition of \widehat{G} , there is a vertex z_i such that $\{v_i, w_i, z_i\}$ is a triangle of \widehat{G} . We mark the two edges $\{v_i, w_i\}$ and $\{v_i, z_i\}$. When we have done this for every i , it is easy to see that each edge of \widehat{G} has at most two marks, so $|\mathcal{M}_2^*|$ is less than or equal to the total number of marked edges.

By construction, the marked edges go between different clusters of \mathcal{C} , but they are edges of G which are also in \widehat{G} , so the number of marked edges is at most $d_2(\mathcal{C}, E)$, where $d_2(\cdot)$ only counts edges that are in \widehat{G} .

Putting all inequalities together and noticing that $d_1(\mathcal{C}, E) + d_2(\mathcal{C}, E) = d(\mathcal{C}, E)$ yields the lemma.

Proof. (Of Part 1 of Theorem 2.1) By Lemma 3.3 we have $d(\text{Output}, E) \leq OPT + d(\mathcal{A}, \widehat{E})$. Consider the clustering \mathcal{C}' obtained from \mathcal{C} by splitting clusters of size 2 whenever the corresponding edge is not in \widehat{E} . By Theorem 3.1 applied to $G = (V, \widehat{E})$ we have $d(\mathcal{A}, \widehat{E}) \leq 2.5d(\mathcal{C}', \widehat{E})$. Lemma 3.2 implies that $d(\mathcal{C}', \widehat{E}) \leq d(\mathcal{C}, E) = OPT$. Putting these inequalities together yields the theorem.

3.3 Lower Bound In this section we prove the following simple lower bound:

THEOREM 3.3. *There exists c_8 such that if $c_8/n \leq p \leq 1/3$ then w.h.p. $OPT_{\text{full}} = \Omega(n^2p)$. If $p \leq c_8/n$ then w.h.p. either $OPT_{\text{full}} = 0$ or $OPT_{\text{full}} = \widetilde{\Omega}(n^3p^2 + nn_2p)$, where n_2 is the number of vertices in base clusters of size 2 or more.*

LEMMA 3.4. (VARIANT OF [19]) $d(OPT, \mathcal{B}) \leq \frac{4n \log n}{-\log 4p}$ with high probability

Proof. Fix some clustering \mathcal{A} . Let $D = d(\mathcal{B}, \mathcal{A})$. Clearly \mathcal{A} is better than \mathcal{B} if and only if at least $D/2$ of the D pairs where they differ are noisy. This occurs with probability at most $\binom{D}{D/2} p^{D/2} \leq 2^D p^{D/2} = (4p)^{D/2}$.

Take a union bound over all $O(n^n)$ clusterings [19] \mathcal{A} with $D = d(\mathcal{B}, \mathcal{A}) \geq \frac{4n \log n}{-\log 4p}$, we bound the probability that $d(OPT, \mathcal{B}) \geq 4n \frac{\log n}{-\log 4p}$ by $n^n \cdot (4p)^{D/2} = \exp(n \log n + \frac{2n \log n}{-\log 4p} \log(4p)) = n^{-n}$.

Proof. (Of Theorem 3.3) First we prove that $OPT = \Omega(n^2p)$ when $c_8/n \leq p \leq 1/5$. In the upper portion of

this range where $1/\sqrt{n} \leq p \leq 1/5$ we see $d(OPT, \mathcal{B}) = O(n \log n)$ by Lemma 3.4. The triangle inequality and a trivial Chernoff bound yield $OPT = d(\mathcal{C}, E) \geq d(\mathcal{B}, E) - d(\mathcal{C}, \mathcal{B}) = \Omega(n^2p) - O(n \log n) = \Omega(n^2p)$. In the lower portion where $c_8/n \leq p \leq 1/\sqrt{n}$ Lemma 3.4 implies $d(OPT, \mathcal{B}) = O(n)$. As before we get $OPT = d(\mathcal{C}, E) \geq d(\mathcal{B}, E) - d(\mathcal{C}, \mathcal{B}) = \Omega(n^2p) - O(n) = \Omega(n^2p)$ for sufficiently large c_8 and hence p .

Second observe that if $n^3p^2 + nn_2p = \tilde{O}(1)$ then the Theorem is trivial. We henceforth assume that $np \leq c_8$ and $n^3p^2 + nn_2p = \omega(\log n)$. We will frequently use the fact that

$$(3.1) \quad \max(n^2p, n_2) \geq \frac{n^2p + n_2}{2} = \frac{n^3p^2 + nn_2p}{2np} = \omega(\log n).$$

A classic lower-bound on OPT is the size of any collection of VP-disjoint bad triplets (triplet packing) where a bad triplet is a set $\{u, v, w\}$ such that $E_{uv} = E_{vw} = 1$ but $E_{uw} = 0$. Consider the instance \mathcal{B}' obtained from \mathcal{B} , by flipping every vertex pair with probability $p/2$, analogously to how E is formed by flipping vertex pairs with probability p . Note that \widehat{E} can be expressed as applying noise of approximately $p/2$ to \mathcal{B}' . We will use \mathcal{B}' to construct a triplet packing lower bound.

We first construct a large matching M of \mathcal{B}' size $\Omega(n_2 + n^2p)$ as follows.

If $n_2 \geq \min(n^2p, n/2)$ note that a max matching of \mathcal{B} has size at least $n_2/3$, which is at least (the smaller of $n/6$ and) $\omega(\log n)$ using (3.1). With high probability only a $O(p)$ fraction of these edges are not in \mathcal{B}' so M has size $\Omega(n_2) = \Omega(n_2 + n^2p)$ w.h.p.

In the case that $n_2 < \min(n^2p, n/2)$ we find a matching among the at least $n/2$ vertices that are singletons in \mathcal{B} . We have $n^2p = \omega(\log n)$ by (3.1), so by a Chernoff bound the number of noisy edges is $\Omega(n^2p)$. Consider generating these noisy edges one by one, adding each to the matching M if possible. As long as $|M| \leq n/8 = \Omega(n^2p)$ (otherwise we are done) each new edge has probability at least $(1/2)^2$ of being added to the matching. These events are not independent but with a little technical effort one can create related events that are. Another Chernoff bound yields $|M| = \Omega(n^2p) = \Omega(n_2 + n^2p)$ w.h.p.

Label the vertices $V = \{v_1, v_2 \dots v_n\}$ so that (v_{2i-1}, v_{2i}) is in the matching M for all $1 \leq i \leq |M|$. Let $\alpha = \min(|M|, n/4)$. Note that $\alpha = \Omega(n_2 + n^2p)$. We construct a triplet packing as follows. For each iteration $1 \leq i \leq \alpha$ check if there exists some $n/2 < j \leq n$ such that (v_{2i-1}, v_{2i}, v_j) is a bad triplet. If so, pick an arbitrary such j and add triplet (v_{2i-1}, v_{2i}, v_j) to the triplet packing.

Consider the probability that iteration i produces

a triplet for some $1 \leq i \leq \alpha$. By construction v_{2i-1} and v_{2i} are in the same cluster of \mathcal{B}' . Each vertex v_j with $n/2 < j \leq n$, whether in the same base cluster or different, has probability $\Theta(p)$ of forming a bad triplet. Therefore the probability that iteration i produces a bad triplet is $\Theta(\min(np, 1)) = \Theta(np)$. The expected number of triplets packed is therefore $\Theta(np)|\alpha| = \Theta(nn_2p + n^3p^2) = \omega(\log n)$. The iterations are independent so a Chernoff bound implies $OPT = \Omega(n^3p^2 + nn_2p)$ with high probability as well.

3.4 Probabilistic Lemmas The following Lemma is a trivial application of Chernoff bounds and will be used frequently.

LEMMA 3.5. *Let Z be a sum of independent indicator random variables. We have $Z \leq 2\mathbf{E}[Z] + 9\ln n$ w.h.p. If $\mathbf{E}[Z] \geq 20\log n$ then $\frac{1}{2}\mathbf{E}[Z] \leq Z \leq 2\mathbf{E}[Z]$ with high probability.*

LEMMA 3.6. *W.h.p. $E_{uv} = \hat{E}_{uv}$ for all u, v within any base cluster of size at least $6\lg n$, where E and \hat{E} are as defined in Algorithm MAINCLUSTER.*

In other words, edges within base clusters of size at least $6\lg n$ are unaffected by the first line of MAINCLUSTER.

Proof. (Of Lemma 3.6) Fix edge u, v of E ,⁶ with u and v both within a single base cluster of size k . For any other vertex w in the same base cluster we have $E_{uw} = E_{vw} = 1$ with probability at most $2p - p^2 \leq 3/4$. These events are independent so the edge u, v is part of no triangles with probability at most $(3/4)^{k-2}$. When $k \geq 6\lg n$ this probability is at most $2n^{-3}$, so with high probability every edge within a cluster is part of a triangle, hence in \hat{E} .

LEMMA 3.7. *The expected number of edges u, v (of E) within base clusters of size at least 3 that are affected by the first line of MAINCLUSTER (i.e. $E_{uv} = 1$ and $\hat{E}_{uv} = 0$) is $O(n_3p)$, where n_3 is the number of vertices in base clusters of size 3 or more. A bound of $\tilde{O}(n_3p+1)$ holds with high probability.*

Proof. Each cluster of size $k \geq 3$ includes in expectation at most

$$\binom{k}{2}(2p - p^2)^{k-2} \leq \frac{k^2}{2}(2p - p^2)(3/4)^{k-3} = O(p)$$

edges that are not part of any triangles. Summing over $O(n)$ clusters proves the expectation part of the Lemma.

By Lemma 3.6 we can ignore clusters of size $\Omega(\log n)$. For cluster B_i of size between 3 and $\Theta(\log n)$ let ϵ_i denote the event that there is some edge within B_i that is not part of some triangle within B_i . These events are clearly independent and as noted above have probability $O(p)$, so by a Chernoff bound (Lemma 3.5) the number of ϵ_i that occur is $O(np + \log n)$ with high probability. Each ϵ_i contributes $O(\log^2 n)$ edges, completing the proof of the theorem.

LEMMA 3.8. *For sufficiently large constant c_9 we have with high probability all vertices are part of at most $\frac{c_9}{10}(np + \log n) - 1$ noisy pairs.*

Proof. (Proof Sketch) Semi-randomness can only help, so consider the fully random model. Fix vertex v . Each of the other vertices $u \in V$ has probability p of being a noisy pair with v , so the total number of noisy pairs is a sum of independent indicator random variables. Lemma 3.5 completes the proof.

We say that a triplet of vertices $\{u, v, w\}$ is an *unnatural triangle* if $E_{uv} = E_{vw} = E_{uw} = 1$ but $\{u, v, w\}$ is not contained within a single base cluster.

LEMMA 3.9. *The number of unnatural triangles whose vertices are all in base clusters of size at most k is $O((np)^3 + (np)^2k)$ in expectation. A bound of $O((np)^3 + (np)^2k + \log^6 n)$ holds with high probability.*

To prove Lemma 3.9 we use an elegant generalization of Chernoff bounds due to Kim and Vu [20, 4] to prove the high probability portion. We present a special case of their theorem in notation suitable for our use.

Let P denote the set of all $\binom{|V|}{2}$ possible pairs of vertices $u, v \in V$. Let \mathcal{T} denote a collection of triplets (sets of size 3) of vertex pairs from P . Let $\{E_p\}_{p \in P}$ denote a collection of independent indicator random variables. Let random variable Y denote the number of $T \in \mathcal{T}$ such that $E_e = 1$ for all $e \in T$. For any $A \subset P$ we have random variable Y_A equal the number of $T \in \mathcal{T}$ such that $A \subset T$ and $E_e = 1$ for all $e \in T \setminus A$.

Let $E_i = \max_{A \subset P: |A|=i} \mathbf{E}[Y_A]$. Observe that $Y = Y_{\emptyset}$ and $E_0 = \mathbf{E}[Y]$.

THEOREM 3.4. (KIM AND VU [20]) *In the scenario above if $E_1, E_2, E_3 \leq 1$ we have*

$$\Pr\left(|Y - \mathbf{E}[Y]| > a\sqrt{\max(\mathbf{E}[Y], 1)\lambda^3}\right) < d \cdot \exp(-\lambda + 2\ln n)$$

for absolute constants a and d .

COROLLARY 3.1. *In the scenario above if $E_1, E_2, E_3 \leq 1$ we have*

$$Y = O(\mathbf{E}[Y] + \log^6 n)$$

with high probability.

⁶Recall “edge” implies $E_{uv} = 1$

Proof. (Of Lemma 3.9) We classify unnatural triangles based on whether their vertices come from two or three distinct clusters. There are $O(n^3)$ possible triplets spanning three different clusters, and each has probability p^3 of being an unnatural triangle. Therefore the expected number of such unnatural triangles is $O(n^3 p^3)$. There are $O(n^2 k)$ possible triplets spanning two different clusters, and each has probability at most p^2 of being unnatural, yielding expectation $O(n^2 p^2 k)$.

Observe that the condition $E_3 \leq 1$ is trivially satisfied. In our applications every $T \in \mathcal{T}$ consists of the edges of a triangle, so $E_2 \leq 1$ is trivially satisfied as well.

For simplicity we assume the fully random model; the reader can readily verify that our arguments hold in the semi-random model as well. To apply Corollary 3.1 we let E be the edges in the fully random model.

Let $\mathcal{T}^{(1)}$ denote the collection of possible triangles with vertices in three distinct base clusters, where the triangles are represented by three vertex pairs from P . In order to apply Corollary 3.1 we need to bound E_1 and E_2 by 1. Recall E_1 is a maximization over sets $A \subset P$ of size 1 and fix $A = \{e\}$ for some $e \in P$. If e is within a single cluster then $Y_A = 0$. If e spans two clusters there are at most n possible $T \ni e$, each of which has probability p^2 , so $E_1 \leq np^2$. Therefore by Corollary 3.1 proves the Lemma with respect to the unnatural triangles with vertices in three distinct base clusters.

Let $\mathcal{T}^{(2)}$ denote the collection of possible triangles with vertices in two distinct base clusters, where the triangles are represented by three vertex pairs from P . Fix $A = \{e\}$ for some $e \in P$. If e is within a single cluster we bound $\mathbf{E}[Y_A]$ by np^2 as before. If e spans two distinct clusters, then the third vertex to form a triangle must be one of at most $2k$ vertices. Each possible triangle has probability p , so $E_1 \leq 2kp = o(1)$. Corollary 3.4 proves the Lemma w.r.t. the unnatural triangles with vertices in two distinct base clusters. This concludes the proof of Lemma 3.9.

3.5 Proof of Theorem 2.1 (2) when $p \leq n^{-2/3}$.

In this part, we analyze the algorithm in the semi-random model, assuming that $p \leq n^{-2/3}$. The case $p \geq n^{-2/3}$ is deferred to Section 3.6.

LEMMA 3.10. *There exists c_9 such that with high probability the semi-definite program finds all base clusters B of size at least $c_9(np + \log n)$ exactly. That is, $X_{uv}^* = 1$ if $u, v \in B$, and $X_{uv}^* = 0$ if $u \in B$ and $v \notin B$.*

Proof. We choose constant c_9 equal to the constant of the same name from Lemma 3.8. We say that vertex pair u, v is *B-incident* if at least one of u and v is in B .

Let X^* be an arbitrary optimal solution of the semi-definite program. Consider the solution X' obtained from X^* by modifying X_{uv}^* for B -incident vertex pairs as follows:

$$X'_{uv} = \begin{cases} X_{uv}^* & \text{if } uv \text{ not } B\text{-incident} \\ 0 & \text{if } |\{u, v\} \cap B| = 1 \\ 1 & \text{if } |\{u, v\} \cap B| = 2 \end{cases}$$

We will prove the Lemma by arguing that $X' = X^*$. Following terminology from [1], say that a triplet $\{u, v, w\}$ of vertices is a *bad triplet* if $\hat{E}_{uv} = \hat{E}_{vw} = 1$ and $\hat{E}_{uw} = 0$. We enrich the semidefinite program by adding the constraint that $X_{uv} = X_{uv}^*$ for every vertex pair uv that is not B -incident. This creates a new semidefinite program (P') that has the same value as the original (so X^* is also optimal for (P')). Then, we relax the constraints by removing the constraint that X be positive semidefinite, and by only writing the ijk constraint for $\{i, j, k\}$ bad triplet such that all three vertex pairs are B -incident. Moreover, we change variables by

$$\text{defining } y_{uv} = \begin{cases} X_{uv} & \text{if } \hat{E}_{uv} = 0 \\ 1 - X_{uv} & \text{if } \hat{E}_{uv} = 1 \end{cases} \quad \text{for } B\text{-incident}$$

vertex pairs. Finally, we translate the objective function by the quantity $\sum_{uv \text{ not } B\text{-incident}} (1 - X_{uv}^*) \hat{E}_{uv} + X_{uv}^* (1 - \hat{E}_{uv})$. We obtain the following linear program (P):

$$\begin{aligned} \text{(P)} \quad & \min \sum_{uv \text{ } B\text{-incident}} y_{uv} \quad \text{s.t.} \\ & y_{uv} + y_{vw} + y_{uw} \geq 1 \quad \text{for } uvw \text{ bad triplet, } |uvw \cap B| \geq 2 \\ & y_{uv} \geq 0 \quad \text{for } uv \text{ } B\text{-incident} \end{aligned}$$

We say that vertex pair u, v is *removed* if $\hat{E}_{uv} \neq \mathcal{B}_{uv}$. A feasible solution to (P) is obtained by setting y'_{uv} equal to 1 if $\hat{E}_{uv} \neq \mathcal{B}_{uv}$ and to 0 otherwise. It is clearly feasible, and its value is the number of B -incident removed pairs. We will argue that y' is the *unique* optimal solution of (P). What does that imply? Observe that y' is precisely the solution obtained from X' by our change of variables. Since it is straightforward to see that X' is feasible in (P'), and (P) is essentially a relaxation of (P'), this implies that X' is the unique optimal solution of (P'). Since X^* is optimal for (P'), we conclude that $X = X^*$, hence the Lemma.

It only remains to prove that y' is the unique optimal solution of (P). Consider the linear programming dual (D) of (P).

$$\begin{aligned} \text{(D)} \quad & \max \sum_T \pi_T \quad \text{s.t.} \\ & \sum_{T \ni \{u, v\}} \pi_T \leq 1 \quad \text{for } uv \text{ } B\text{-incident} \\ & \pi_T \geq 0 \quad \text{for } T \text{ bad triplet, } |T \cap B| \geq 2 \end{aligned}$$

A feasible solution to (D) is constructed as follows. We say that vertex pair u, v is *noisy* if $E_{uv} \neq \mathcal{B}_{uv}$. We say that a bad triplet of vertices is *broken* if at least two of the vertices from B and exactly one of the three vertex pairs are removed. We set

$$\pi_T = \begin{cases} 1/\#\{\text{broken } T' \text{ that share} \\ \text{their removed edge with } T\} & \text{if } T \text{ is broken} \\ 0 & \text{otherwise} \end{cases}$$

Let us prove that, with high probability, π is feasible.

If vertex pair $e = uv$ is removed then, by definition of π , we have $\sum_{T: e \in T} \pi_T = 1$.

If vertex pair $e = uv$ is not removed, then any broken triplet $T = uvw$ must have either uw or vw removed, hence noisy. Using Lemma 3.8, the packing constraint associated to e therefore has

$$\begin{aligned} \sum_{T: e \in T} \pi_T &\leq \sum_{w: uw \text{ noisy}} \pi_{uvw} + \sum_{w: vw \text{ noisy}} \pi_{uvw} \\ &\leq 2 \frac{c_9}{10} (np + \log n) \max_T \pi_T. \end{aligned}$$

To bound $\max_T \pi_T$, fix a broken T with removed edge $e = uv$. Observe for every vertex $w \in B \setminus \{u, v\}$, triplet $T' = uvw$ is broken, except when uw or vw is removed. By Lemma 3.6, every B -incident removed pair is also noisy, so by Lemma 3.8 we can write

$$\pi_T \leq \frac{1}{|B| - \frac{2c_9}{10}(np + \log n)} \leq \frac{10}{8c_9(np + \log n)}$$

using the assumption of Lemma 3.10 that $|B| \geq c_9(np + \log n)$. This implies $\sum_{T: e \in T} \pi_T \leq 1/4$, proving feasibility of π .

Now, observe that the value of π is exactly the number of B -incident removed pairs. This equals the value of y' , so y' and π are both optimal.

Moreover, consider a non-removed pair $e = uv$. Observe that for π , the packing constraint associated to e has positive slackness ($\geq 3/4$). By complementary slackness conditions (and optimality of π) this implies that *every* optimal primal solution y satisfies $y_{uv} = 0$. Now, consider a removed pair $e = uv$, and let $w \in B - \{u, v\}$ be such that neither uw nor vw are removed (such a w exists by Lemma 3.8, Lemma 3.6, and our assumption on $|B|$). Then (P) has a constraint associated to uvw , which, for an optimal solution, reads $y_{uv} + y_{vw} + y_{uw} = y_{uv} \geq 1$. By optimality again, we infer $y_{uv} = 1$. Therefore y' is the *unique* optimal solution of (P), as desired.

Proof. (Of Theorem 2.1 (2) when p is small) By Lemma 3.3,

$$\text{Cost}(\text{Output}) \leq \text{OPT} + d(\mathcal{A}, \hat{E}).$$

By Lemma 3.10, we know that the optimal solution X^* of the semi-definite program matches \mathcal{B} precisely on all vertex pairs incident to at least one cluster of size at least $c_{10}(np + \log n)$. By the design of the rounding algorithm, this implies that \mathcal{A} also matches \mathcal{B} precisely on for those clusters; moreover, X^* is also optimal in the subgraph induced by the vertices in the remaining clusters. So we can ignore the large clusters and assume that all clusters have size at most $k = c_9(np + \log n)$.

Consider the clustering \mathcal{B}' obtained from \mathcal{B} by splitting clusters of size 2 in two. By Theorem 3.1,

$$\mathbf{E}_{\text{alg}} [d(\mathcal{A}, \hat{E})] \leq 2.5 d(\mathcal{B}', \hat{E}).$$

We now proceed to compare $d(\mathcal{B}', \hat{E})$ to OPT.

First, consider the case $np \leq c_8$, where c_8 is the constant from Theorem 3.3. We will bound the expected value, over the noisy process, of $d(\mathcal{B}', \hat{E})$ and use Markov's inequality. Clearly a vertex pair u, v only contributes to $d(\mathcal{B}', \hat{E})$ if one of the following three conditions holds.

- $B'_{uv} = 0$ and $E_{uv} = \hat{E}_{uv} = 1$. We bound the expected number of such pairs by three times the expected number of unnatural triangles, i.e. $O((np)^3 + (np)^2 k)$ by Lemma 3.9.
- $B'_{uv} = 1$ and $E_{uv} = \hat{E}_{uv} = 0$. We bound the expected number of such pairs by $O(n_3 k p)$ trivially, where n_3 is the number of vertices in base clusters of size at least 3 and $k = c_9(np + \log n)$ is the upper-bound on cluster size.
- $B'_{uv} = 1$, $E_{uv} = 1$ and $\hat{E}_{uv} = 0$. We bound the number of such pairs by $O(n_3 p)$ using Lemma 3.7.

Altogether,

$$\begin{aligned} \mathbf{E}_{\text{graph}} [d(\mathcal{B}', \hat{E})] &= O([(np)^3 + (np)^2 k] + [n_3 k p] + [n_3 p]) \\ (3.2) \quad &= \tilde{O}((np)^2 + n_3 p) \end{aligned}$$

since $k = O(\log n)$ in this case. By Theorem 3.3 we have $\text{OPT}_{\text{full}} = \Omega(n_2 np + n^3 p^2)$ w.h.p.⁷ and so $\mathbf{E} [d(\mathcal{B}', \hat{E})] = \tilde{O}(\text{OPT}_{\text{full}}/n)$, hence by Markov's inequality, with high probability we have $d(\mathcal{B}', \hat{E}) = O(n^{-1/6})\text{OPT}_{\text{full}}$.

Second, consider the case $np > c_8$. Then $k = \tilde{O}(np)$. Lemmas 3.9, 3.5 and 3.7 allow us to redo the

⁷If $\text{OPT}_{\text{full}} = 0$ the present part 2 of Theorem 2.1 follows from part 1.

proof of (3.2), replacing expectations by high probability statements, to show that

$$(3.3) \quad d(\mathcal{B}', \hat{E}) = O([(np)^3 + (np)^2 k] + [n_3 k p] + [n_3 p] + \log^6 n)$$

with high probability. Now, (3.3) simplifies to $d(\mathcal{B}', \hat{E}) = \tilde{O}((np)^3) \leq \tilde{O}(n^2 p) np^2 \leq \tilde{O}(n^{-1/3}) OPT_{full}$ w.h.p. using Theorem 3.3. Together, the two cases completes the proof.

3.6 Proof of Theorem 2.1 (2) when $p \geq n^{-2/3}$
Define $M \bullet N \equiv \sum_{u,v} M_{uv} N_{uv}$, and for any $\{0, 1\}$ matrix M , let

$$\tilde{M}_{uv} = \begin{cases} -1 & \text{if } M_{uv} = 1 \\ 1 & \text{if } M_{uv} = 0. \end{cases}$$

This gives a way to rewrite the objective of our semi-definite program. The following Lemma is trivial.

LEMMA 3.11. *For any symmetric matrices M and N with $M_{uv} \in \{0, 1\}$ and $0 \leq N_{uv} \leq 1$, we have*

$$d(M, N) = (1/2)(\tilde{M} \bullet N - \tilde{M} \bullet M)$$

The next Lemma is key. It is eventually used for $X = X^*$, the optimal solution to the SDP.

LEMMA 3.12. *With high probability over the noisy model, the following holds: for every positive semi-definite matrix X with trace n ,*

$$|\tilde{E}_{full} \bullet X - \mathbf{E}[\tilde{E}_{full}] \bullet X| \leq 5\sqrt{pn}^{3/2}.$$

Proof. Let $M = \tilde{E}_{full} - \mathbf{E}[\tilde{E}_{full}]$. Since X is symmetric, we can write $X = \sum_{i=1}^n \lambda_i v_i v_i^T$ where λ_i are the eigenvalues of X and v_i are corresponding unit-length eigenvectors. By elementary linear algebra, (using the fact that $Tr(AB) = Tr(BA)$ for two (m, n) and (n, m) matrices),

$$\begin{aligned} M \bullet X &= Tr(M^T X) = \sum_i Tr(M^T \lambda_i v_i v_i^T) \\ &= \sum_i \lambda_i Tr(v_i^T M v_i) = \sum_i \lambda_i v_i^T M v_i. \end{aligned}$$

Since X is positive semi-definite, the λ_i s are non-negative and we get

$$\begin{aligned} |M \bullet X| &\leq \sum_i \lambda_i |v_i^T M v_i| \leq \sum_i \lambda_i \rho(M) \\ &= Tr(X) \rho(M) = n \rho(M), \end{aligned}$$

where $\rho(M)$ is the spectral radius of M . By [15] we have $\rho(M) \leq 5\sqrt{np}$, hence the Lemma.

Proof. (Of Part 2 of Theorem 2.1 when $p \geq n^{-2/3}$) Since the maximum matching of U can only improve the cost, the cost of the output is at most $d(\mathcal{A}, E)$, where \mathcal{A} denotes the output of SDPCLUSTER. By the triangle inequality and Theorem 3.2,

$$\begin{aligned} \mathbf{E}_{alg}[d(\mathcal{A}, E)] &\leq \mathbf{E}_{alg}[d(\mathcal{A}, X^*)] + d(X^*, E) \\ &\leq 3d(\mathcal{B}, X^*) + d(X^*, E). \end{aligned}$$

Let \mathcal{C} be an optimal clustering satisfying the conditions of Lemma 3.2. Since the semi-definite program is a relaxation, $d(X^*, \hat{E}) \leq d(\mathcal{C}, \hat{E})$. Lemma 3.2 implies that

$$(3.4) \quad d(X^*, E) \leq d(\mathcal{C}, E) + n/2 = OPT + O(n).$$

To see this, transform \hat{E} into E by switching vertex pairs one at a time. Each time vertex pair uv is switched, either u, v is a cluster of size 2 in \mathcal{C} or the cost of \mathcal{C} increases by 1. The cost of X^* , as that of any fractional clustering, increases by at most 1, hence Equation (3.4).

By Lemma 3.11, $d(\mathcal{B}, X^*) = (1/2)(\tilde{\mathcal{B}} \bullet X^* - \tilde{\mathcal{B}} \bullet \mathcal{B})$. In the fully random model, it is straightforward that $\mathbf{E}[\tilde{E}_{full}] = (1 - 2p)\tilde{\mathcal{B}}$, and so

$$(\tilde{\mathcal{B}} \bullet X^* - \tilde{\mathcal{B}} \bullet \mathcal{B}) = \frac{1}{1 - 2p}(\mathbf{E}[\tilde{E}_{full}] \bullet X^* - \mathbf{E}[\tilde{E}_{full}] \bullet \mathcal{B}).$$

Applying Lemma 3.12 to both $X = X^*$ and $X = \mathcal{B}$, we can write

$$d(\mathcal{B}, X^*) \leq \frac{1}{2(1 - 2p)}(\tilde{E}_{full} \bullet X^* - \tilde{E}_{full} \bullet \mathcal{B} - 10\sqrt{pn}^{3/2}).$$

Applying Lemma 3.11 again, once to $\tilde{E}_{full} \bullet X^*$ and once to $\tilde{E}_{full} \bullet \mathcal{B}$, we get

$$\tilde{E}_{full} \bullet X^* - \tilde{E}_{full} \bullet \mathcal{B} = 2(d(E_{full}, X^*) - d(E_{full}, \mathcal{B})).$$

We observe that

$$(3.5) \quad d(X^*, E_{full}) - d(\mathcal{B}, E_{full}) \leq d(X^*, E) - d(\mathcal{B}, E).$$

To see that, transform E_{full} into E by switching vertex pairs one at a time. Each time the adversary is “nice” (taking advantage of the semi-random model) and declines to add noise to vertex pair uv , the value of \mathcal{B} decreases by 1 whereas the value of X^* , as that of any fractional clustering, decreases by at most 1, hence Equation (3.5).

We further claim that

$$(3.6) \quad d(X^*, E) - d(\mathcal{B}, E) \leq d(X^*, \hat{E}) - d(\mathcal{B}, \hat{E}) + \tilde{O}(np + 1).$$

To see that, we transform E into \hat{E} by switching vertex pairs one at a time. Each time the algorithm removes

an edge of E that are between different clusters of \mathcal{B} , the value of \mathcal{B} decreases by 1 whereas the value of X^* , as that of any fractional clustering, decreases by at most 1. Each time the algorithm removes an edge of E that is inside a cluster of \mathcal{B} , the value of \mathcal{B} changes by 1 and the value of X^* , as that of any fractional clustering, changes by at most 1, so the difference changes by at most 2. Hence the difference from $d(X^*, E) - d(\mathcal{B}, E)$ to $d(X^*, \hat{E}) - d(\mathcal{B}, \hat{E})$ is at most twice the number of edges inside clusters of \mathcal{B} that are removed by the algorithm. By Lemma 3.7 there are $\tilde{O}(np + 1)$ such pairs, hence Equation (3.6).

By optimality of X^* , we have $d(\hat{E}, X^*) - d(\hat{E}, \mathcal{B}) \leq 0$. Clearly $O(n) + \tilde{O}(np + 1) = O(\sqrt{pn}^{3/2})$ for $p \geq n^{-2/3}$. Altogether this yields

$$\text{Cost}(\text{OUT}) \leq \text{OPT} + O\left(\frac{\sqrt{pn}^{3/2}}{1 - 2p}\right).$$

By Theorem 3.3, $\text{OPT}_{\text{full}} = \Omega(n^2 p)$, and so

$$(\text{Cost}(\text{OUT}) - \text{OPT}) / \text{OPT}_{\text{full}} = O\left(\frac{1}{\sqrt{np}(1/2 - p)}\right).$$

It is easy to check that for $n^{-2/3} \leq p \leq 1/2 - n^{-1/3}$, this quantity is $O(n^{-1/6})$.

3.7 Proof of Theorem 2.1 (3) Part (3) of Theorem 2.1 is proved using the same techniques as used to prove part (2). When p is large we choose lower bound $L = d(X^*, E) - n/2$, which is valid by Equation (3.4) and was implicitly shown to be sufficiently tight in the last section.

When p is small the argument is a bit more involved. First note that the proof of Lemma 3.10 can be readily converted into a polynomial-time algorithm for certifying that a particular cluster is in every optimal clustering. This algorithm succeeds with high probability on any base cluster B of size at least $c_9(np + \log n)$. We run this algorithm (solve an LP and check complementary slackness) on every output cluster $A \in \text{Output}$. We discard those clusters that this algorithm successfully certifies and compute initial lower bound L_1 equal to what Output (and hence \mathcal{C}) pays for the edges incident to the discarded clusters. Lemma 3.3 implies that $L_2 = d'(\text{Output}, E) - d'(\mathcal{A}, \hat{E})$ is a lower bound on the instance induced by the remaining vertices, where $d'(\cdot, \cdot)$ is like $d(\cdot, \cdot)$ but is limited to the subgraph induced by the remaining vertices. Our overall lower bound is $L = L_1 + L_2$.

4 Proof of Theorem 2.2

This proof is inspired by the analysis in [14] for the planted clique problem. Our key innovation is ex-

tending the results of Füredi and Komlós [15] to random matrices with entries only partially independent (Lemma 4.5).

4.1 Overall Proof First we give a brief outline of our proof. First we define a new SDP (4.7) which is (up to scaling) a relaxation of the SDP in our SDPCLUSTER algorithm. Then we write its dual (4.8). We define a dual solution (Π, Υ) in (4.9). Finally we analyze eigenvalues (Lemmas 4.1-4.4) to prove that our dual solution is feasible. Finally we use the fact that our dual feasible solution has the same objective value as \mathcal{B} in the primal, plus a few more arguments, to prove the theorem.

By Lemma 3.6 with high probability all edges within base clusters remain in \hat{E} . Edges between clusters that are not in \hat{E} can be accounted for as semi-randomness, so we can ignore the distinction between \hat{E} and E . Semi-randomness can be dealt with easily (using Equation (3.5) in Section 3.6) so we focus on the fully random case.

Throughout this section we assume that there are b clusters of size at least $\gamma \geq c_5 \sqrt{n}$ for some constant c_5 to be decided later. Let $\mu = 1 - 2p \geq 1/3$.

Let J^{uv} denote a matrix with u, v entry equal to 1 and all others 0. Following [2], we write $M \succeq N$ if $M - N$ is positive semi-definite. Recall the definition $\tilde{E}_{uv} = \begin{cases} -1 & \text{if } E_{uv} = 1 \\ 1 & \text{if } E_{uv} = 0 \end{cases}$. Observe that X^* is feasible in the following SDP:

$$(4.7) \quad \begin{aligned} & \max_X -\tilde{E} \bullet X \quad \text{s.t.} \\ & \begin{cases} J^{uu} \bullet X = 1 & \forall u \in V & (\text{Dual variable } \Pi_u) \\ -J^{uv} \bullet X \leq 0 & \forall u, v \in V & (\text{Dual variable } -\Upsilon_{uv}) \\ X \succeq 0 & & (\text{positive semi-definite}) \end{cases} \end{aligned}$$

Recall from Lemma 3.11 that $d(E, X) = (1/2)(\tilde{E} \bullet X - \tilde{E} \bullet E)$ so up to rescaling SDP (4.7) relaxes the SDP in SDPCLUSTER (by eliminating the triangle inequalities).

For convenience we package the dual variables Π_u and $-\Upsilon_{uv}$ (the minus sign will be convenient later) into diagonal matrix Π and symmetric matrix $-\Upsilon$. Here is the dual of SDP (4.7):

$$(4.8) \quad \begin{aligned} & \min_{\Pi, \Upsilon} \sum_v \Pi_v \quad \text{s.t.} \\ & \begin{cases} \Pi - (-\Upsilon) \succeq -\tilde{E} \\ \Pi \text{ diagonal} \\ -\Upsilon \succeq 0 \end{cases} \quad (\text{All entries non-negative}) \end{aligned}$$

We rewrite the first constraint in the dual as $M \equiv \tilde{E} + \Pi + \Upsilon \succeq 0$.

Recall that the vertex set V is partitioned into b base clusters B_1, \dots, B_b . For this proof we subdivide⁸ the clusters into *subclusters* S_1, \dots, S_r so that $\gamma/2 \leq |S_i| \leq \gamma$ for all i . For vertex u let $B(u)$ and $S(u)$ denote the cluster and subcluster u is in respectively. We now present our dual solution (Π, Υ) . We choose

$$(4.9) \quad \Pi_{uu} = - \sum_{v \in B(u)} \tilde{E}_{uv}.$$

Let Υ_{uv} be zero when $B(u) = B(v)$ and

$$\sum_{i \in S(u), j \in S(v)} \frac{\tilde{E}_{ij}}{|S(u)||S(v)|} - \sum_{i \in S(u)} \frac{\tilde{E}_{iv}}{|S(u)|} - \sum_{j \in S(v)} \frac{\tilde{E}_{uj}}{|S(v)|}$$

otherwise. This dual solution satisfies two key (and easily verified) properties:

1. The vector with all components in one base cluster equal to 1 and all others zero is an eigenvector of M with eigenvalue 0.
2. The dual objective value $\sum_v \Pi_v$ equals $-\tilde{E} \bullet \mathcal{B}$, the primal value of \mathcal{B} .

Let us first show that $-\Upsilon_{uv} > 0$ with high probability for any u, v with $B(u) \neq B(v)$. Observe that each of the three sums in the definition of Υ_{uv} are an average of $\Omega(\gamma)$ independent $-1/1$ random variables each with mean μ . Each of the three sums therefore equals its expectation, namely μ , plus or minus $O\left(\sqrt{\frac{\log n}{\gamma}}\right) < \mu/10$ by the Azuma-Hoeffding inequality. We conclude that

$$(4.10) \quad \Upsilon_{uv} < 0.$$

As noted above M has b orthogonal eigenvectors with eigenvalue 0. We next show that the $b+1$ smallest eigenvalue of M is strictly positive, which will imply that M is positive semi-definite. To do so we decompose M as

$$M = \tilde{E} + \Pi + \Upsilon \\ = \Pi + \underbrace{\left(\mathbf{E} \left[\tilde{E} + \Upsilon \right] \right)}_{\equiv M^{(1)}} + \underbrace{\left(\tilde{E} - \mathbf{E} \left[\tilde{E} \right] \right)}_{\equiv M^{(2)}} + \underbrace{\left(\Upsilon - \mathbf{E} \left[\Upsilon \right] \right)}_{\equiv M^{(3)}}$$

and analyze the eigenvalues of Π , $M^{(1)}$, $M^{(2)}$ and $M^{(3)}$ separately. In particular we will prove the following four Lemmas:

LEMMA 4.1. *All eigenvalues of Π are $\Omega(\sqrt{\gamma})$ with high probability.*

LEMMA 4.2. *The $b+1$ smallest eigenvalue of $M^{(1)}$ is μ .*

LEMMA 4.3. *$M^{(2)}$ has all eigenvalues at least $-\Theta(\sqrt{n})$ with high probability.*

LEMMA 4.4. *$M^{(3)}$ has all eigenvalues at least $-\Theta(\sqrt{n})$ with high probability.*

Before proving these Lemmas we show how they imply Theorem 2.2. We use the following corollary of the Courant-Fischer Theorem, due to Weyl:

THEOREM 4.1. (WEYL [17]) *For any $n \times n$ symmetric matrices A and B and integer $1 \leq k \leq n$ the k th smallest eigenvalue of $A+B$ is at least the k th smallest eigenvalue of A plus the smallest eigenvalue of B .*

We apply Theorem 4.1 three times with $k = b+1$ together with Lemmas 4.1, 4.2, 4.3 and 4.4. This shows that the $b+1$ smallest eigenvalue of M is at least $\Omega(\sqrt{\gamma}) - \mu - \Theta(\sqrt{n}) - \Theta(\sqrt{n}) > 0$ for sufficiently large $\gamma = c_5 \sqrt{n}$. We conclude $M \succeq 0$ and hence our dual assignment is feasible. As already remarked this dual assignment has the same objective value as \mathcal{B} in the primal, so we conclude that both are optimal.

We will now prove Theorem 2.2 by showing that \mathcal{B} is not only a primal optimal but the *unique* optimum.

Proof. (Of Theorem 2.2) Let X^* be an arbitrary optimal solution of the SDPCUSTER SDP. First consider some u, v in different clusters. We previously showed (4.10) that dual variable $-\Upsilon_{uv}$ is strictly positive, hence by the complementary slackness condition $X_{uv}^* \cdot -\Upsilon_{uv} = 0$ we conclude $X_{uv}^* = 0$.

For u, v in different clusters we use the complementary slackness condition $MX^* = 0$ [2]. This implies that any eigenvector of X^* with non-zero eigenvalue must be in the nullspace of M . In the previous paragraph we showed that X^* is block diagonal with one block per cluster so w.l.o.g. assume eigenvectors of X^* each have their support contained within a single cluster. Our previous analysis of the eigenspectrum of M implies that the nullspace of M is spanned by vectors x_1, x_2, \dots, x_b where x_i has components equal to 1 within base cluster B_i and 0 elsewhere. We can therefore write X^* as a linear combination of rank-one outer product matrices $x_i x_i^T$. The constraint $X_{uu} = 1$ imply that the linear combination has coefficients all 1, i.e. $X^* = \mathcal{B}$.

⁸On first reading consider the special case of all clusters having size exactly γ and $B_i = S_i$.

4.2 Eigenvalue analysis The following Lemma, proved shortly in Section 4.3, will be helpful for proving Lemmas 4.3 and 4.4. Füredi and Komlós [15] showed that a symmetric matrix with independent random entries, each with variance σ^2 , has spectral radius $(2 + o(1))\sigma\sqrt{n}$. Our Lemma generalizes their result to matrices whose entries are not completely independent, but independent outside certain roughly equal sized blocks.

LEMMA 4.5. (GENERALIZATION OF [15]) *Let set V , $|V| = n \geq 64$, be partitioned into r classes $S_1 \dots S_r$, all of size between $\beta/2$ and β for some $1 \leq \beta \leq n$. Let $S(v)$ denote the class that $v \in V$ is in. Let N be a matrix-valued random variable indexed by V where:*

- $N_{uv} = N_{vu}$ for all $u, v \in V$ (symmetric)
- $\mathbf{E}[N] = 0$
- The blocks of matrix N induced by the class structure are mutually independent. The block indexed by $1 \leq i, j \leq r$ is the set of all random variables N_{uv} with $\{S(u), S(v)\} = \{i, j\}$.
- $\mathbf{E}[N_{uv}^2] \leq \sigma^2$ and $|N_{uv}| \leq K$ for all $u, v \in V$ and some uniform K and σ with $K \geq \sigma > 0$.
- $\beta \cdot \frac{K^2}{\sigma^2} \leq n/(800 \lg^4 n) = \tilde{O}(n)$

Then with probability at least $1 - n^{-8}$ all eigenvalues of N have absolute value at most $20\sigma\sqrt{\beta n}$, i.e. the spectral radius of N is $O(\sigma\sqrt{\beta n})$.

Lemma 4.5 with $\beta = 1$ is equivalent up to constants to [15]. We use Lemma 4.5 twice, once with $\beta = 1$ and once with β equal to the cluster size lower-bound γ .

Proof. (Of Lemma 4.1) Matrix Π is diagonal so its eigenvalues are simply the entries on its diagonal. Consider entry Π_u for vertex u in cluster $B(u)$. Clearly Π_u consists of the sum of $|B(u)| - 1$ independent random variables each with mean μ . Therefore by the Azuma-Hoeffding inequality we have $\Pi_u = \mu(|B(u)| - 1) \pm O(\sqrt{|B(u)| - 1} \log n) = \Omega(\gamma)$ with high probability. A union bound over vertices completes the proof.

Proof. (Of Lemma 4.2) Note that $\mathbf{E}[\Upsilon_{ij}] = -\mathbf{E}[\tilde{E}_{ij}]$ for all i and j in different clusters hence $M^{(1)}$ is block diagonal with one block per cluster. Consider some block N corresponding to cluster B_i . Clearly $N = -\mu J + \mu I$, hence N has eigenvalue $-\mu|B_i| + \mu$ with multiplicity 1 and eigenvalue μ with multiplicity $|B_i| - 1$. Unioning these spectra over the clusters proves the Lemma.

Proof. (Of Lemma 4.3) Observe that the entries of $M^{(2)}$ are independent, have variance $O(p)$ and are bounded by 2. The result of Füredi and Komlós, i.e. Lemma 4.5 with $s = n$ and $\beta = 1$, proves the Lemma.

Proof. (Of Lemma 4.4) It is easy to verify that $M^{(3)}$ satisfies the conditions of Lemma 4.5 with $\beta = \gamma$, $\sigma = \Theta(\frac{1}{\sqrt{\gamma}})$ and $K = \Theta\left(\sqrt{\frac{\log n}{\gamma}}\right)$, the last by Azuma-Hoeffding inequality. Applying Lemma 4.5 proves the Lemma.

4.3 Proof of Lemma 4.5 We extend the techniques of Füredi and Komlós [15]. Let λ_i denote the i th eigenvalue of matrix N . Let $k = 10 \lceil \lg n \rceil$, an even integer. Clearly

$$\begin{aligned} & \Pr\left(\max_i |\lambda_i| \geq 20\sigma\sqrt{\beta n}\right) \\ & \leq \Pr\left(\sum_i \lambda_i^k \geq (20\sigma\sqrt{\beta n})^k\right) \\ (4.11) \quad & \leq \mathbf{E}\left[\sum_i \lambda_i^k\right] / (20\sigma\sqrt{\beta n})^k \quad (\text{Markov}). \end{aligned}$$

The sum of λ_i^k is the trace of N^k , which can alternatively be written as $\sum_q \prod_{e \in q} N_e$, where the sum is over walks q of length k whose ending point equals their starting point. Consider such a walk q .

If there are two classes S_i and S_j such that q has exactly one edge f between S_i and S_j , then

$$\mathbf{E}\left[\prod_{e \in q} N_e\right] = \mathbf{E}[N_f] \mathbf{E}\left[\prod_{e \in q, e \neq f} N_e\right] = 0$$

by independence and since $\mathbf{E}[N_f] = 0$ by assumption.

If not, let p denote the number of classes visited by walk q . Let S_i and S_j be two of them and assume that q has $\ell \geq 2$ visits of edges between them. Then one can check (using the inequalities of Hölder and Cauchy-Schwarz) that $\mathbf{E}\left[\prod_{e \in q, e \in S_i \times S_j} N_e\right] \leq \sigma^2 K^{\ell-2}$. By independence,⁹ we deduce

$$\mathbf{E}\left[\prod_{e \in q} N_e\right] \leq \sigma^{2(p-1)} K^{k-2(p-1)}$$

We group walks q of this type by the corresponding walk q' over the classes, where walk q over V refines walk q' over the set of classes if the class of each point in q is equal to the corresponding point in q' . Any such q' must clearly use each *superedge* between classes at

⁹And the assumption $\sigma \leq K$.

least twice or not at all. For any fixed q' there are at most β^k walks q that refine q' . Therefore for any q'

$$(4.12) \quad \mathbf{E} \left[\sum_{q \text{ refines } q'} \prod_{e \in q} N_e \right] \leq \beta^k \sigma^{2(p-1)} K^{k-2(p-1)}$$

We now use the following Lemma implicit in [15]:

LEMMA 4.6. ([15]) *The number of walks (cyclic or not) of k steps on the complete graph with r vertices that visit exactly p distinct vertices and visit each non-loop edge at least twice (in either direction) is at most*

$$r \cdot (r-1) \cdot \dots \cdot (r-p+1) \cdot \binom{k}{k-2p+2} p^{2(k-2p+2)} \frac{1}{p} \binom{2p-2}{p-1}.$$

Let $T(p)$ denote the contribution of walks that visit p distinct classes to the expected trace of N^k . Combining Lemmas 4.6 and (4.12) we see that for any $p \leq k/2 + 1$.

(4.13)

$$\begin{aligned} T(p) &\leq r^p \binom{k}{k-2p+2} p^{2(k-2p+2)} \frac{1}{p} \binom{2p-2}{p-1} \cdot \beta^k \sigma^{2p-2} K^{k-2p+2} \\ &\leq r^p 2^k p^{2(k-2p+2)} \cdot 1 \cdot 2^{2p-2} \cdot \beta^k \sigma^{2p-2} K^{k-2p+2} \\ (4.14) \quad &= (2p^2 \beta K)^k \left(\frac{4\sigma^2 r}{p^4 K^2} \right)^{p-1} r \\ &\equiv \bar{T}(p) \end{aligned}$$

Recall that $k = 10 \lceil \lg n \rceil$ hence $p \leq \frac{k}{2} + 1 \leq 5 \lceil \lg n \rceil + 1 \leq 6 \lg n$. Also recall the assumption that $\beta \cdot \frac{K^2}{\sigma^2} \leq n/(400 \lg^4 n)$, hence $\frac{\sigma^2 n}{(\lg n)^4 K^2 \beta} \geq 400$. Finally note the trivial fact $r \geq n/\beta$. Putting these facts together we conclude

$$(4.15) \quad \left(\frac{4\sigma^2 r}{p^4 K^2} \right) \geq \left(\frac{4\sigma^2 n}{(6 \lg n)^4 K^2 \beta} \right) \geq \frac{4 \cdot 800}{6^4} > 2.$$

Combining (4.14) and (4.15) we conclude

$$(4.16) \quad \begin{aligned} \mathbf{E} \left[\sum_i \lambda_i^k \right] &= \sum_{p=1}^r T(p) \leq \sum_{p=1}^{k/2+1} \bar{T}(p) \\ &\leq 2\bar{T}(k/2+1) = 2(4\beta\sigma\sqrt{r})^k \cdot 2r. \end{aligned}$$

We now finish the proof of Lemma 4.5:

$$\begin{aligned} \Pr \left(\max_i |\lambda_i| \geq 20\sigma\sqrt{\beta n} \right) &\leq \mathbf{E} \left[\sum_i \lambda_i^k \right] / (20\sigma\sqrt{\beta n})^k \quad (\text{by 4.11}) \\ &\leq \frac{(4\beta\sigma\sqrt{r})^k 2r}{(20\sigma\sqrt{\beta n})^k} \quad (\text{by 4.16}) \\ &\leq \left(\frac{1}{2} \right)^k \cdot 2n \quad (r \leq 2n/\beta \text{ and } r \leq n) \\ &\leq n^{-8} \quad (k = 10 \lceil \lg n \rceil). \end{aligned}$$

5 Proof of Theorem 2.3

5.1 Preliminaries Algorithm 3 proceeds by first identifying a candidate set of clusters \mathcal{A} and then certifying that the large clusters therein, denoted \mathcal{A}' , are in fact optimal. The following four Lemmas easily imply Theorem 2.3.

LEMMA 5.1. *W.h.p. Algorithm 3 produces intermediate set of clusters \mathcal{A} that includes all clusters of the base clustering \mathcal{B} that have size greater than or equal to $3 \lceil 70/\delta \rceil$.*

LEMMA 5.2. *W.h.p. Algorithm 3 produces intermediate set of clusters \mathcal{A} that includes no clusters except those guaranteed by Lemma 5.1.*

LEMMA 5.3. *Conditioned on the w.h.p. events of Lemmas 5.1 and 5.2 occurring Algorithm 3 outputs “failure” with probability at most $2n^{-5}$.*

LEMMA 5.4. *Let \mathcal{A}' be a clustering output by Algorithm 3 and \mathcal{C} be an optimal clustering. We have $\mathcal{A}' = \{C \in \mathcal{C} : |C| \geq 15s'\}$.*

We prove each of these Lemmas in turn. Throughout this section we assume that $p \leq n^{-\delta}/60$ for some $\delta > 0$ and $s = \lceil 70/\delta \rceil$.

For edge set E and disjoint vertex sets S and T , let $E(S, T)$ denote the number of edges with one endpoint in S and the other in T . For vertex v let $E(v, T)$ denote $E(\{v\}, T \setminus \{v\})$.

Recall that $s \geq \lceil 70/\delta \rceil$. If $p \leq n^{-5}$, then with probability $1 - n^{-3}$ the input graph is just the base clustering with no noise, in which case cluster-finding is trivial. We can therefore safely assume $\delta < 5$, and hence $s \geq 40/\delta + 5$.

A *mistake* is an edge whose label is different in the base clustering and in the input clustering.

LEMMA 5.5. *Let X be a set of edges of cardinality at least $70/\delta$. Then, with probability at least $1 - n^{-7}$, there are at most $|X|/10$ mistakes in X .*

Proof. Elementary counting: $\binom{|X|}{|X|/10} p^{|X|/10} \leq \left(\frac{e|X|p}{|X|/10}\right)^{|X|/10} \leq n^{-\delta|X|/10} \leq n^{-7}$.

LEMMA 5.6. *With probability at least $1 - n^{-5}$, every subgraph induced by a vertex set Y of size $k \geq 40/\delta + 5$ has at most $|Y|^2/40$ mistakes.*

Proof. Let $k \geq 40/\delta + 5$ be given. There are $\binom{n}{k} \leq n^k$ subsets of size k . Each subset has $\binom{k}{2} \leq k^2/2$ vertex pairs within a given subset, so the mean number of mistakes is at most $k^2 p/2$. Using Markov and Chernoff bounds, write:

$$\Pr(\exists S \text{ of size } k \text{ with at least } k \text{ mistakes}) \leq n^k \left(\frac{epk^2/2}{k^2/40} \right)^{k^2/40} \leq n^{k-\delta k^2/40}$$

We therefore want $k - \delta k^2/40 \leq -5$. By Taylor series around $k = 40/\delta$, we see that $k - \delta k^2/40 \leq -(k - 40/\delta)$, hence $k \geq 40/\delta + 5$ suffices.

LEMMA 5.7. *Let U, W be disjoint vertex sets and $s > 0$ an integer. Assume that for every subset S of U and T of W with cardinalities $|S| = |T| = s$, there are at most (resp. at least) $(.35)|S||T|$ edges between S and T . Then for every subset S' of U and T' of W with cardinalities $|S'|, |T'| \geq s$, there are also at most (resp. at least) $(.35)|S'||T'|$ edges between S' and T' .*

Proof. Here is a way to compute the number of edges between S' and T' divided by $|S'||T'|$: Pick a random subset S of S' of size s and a random subset T of T' also of size s , compute the number of edges between S and T divided by s^2 , and average over the random choices of S, T . The number of edges between S and T is bounded by assumption, hence the lemma.

For every cluster $B \in \mathcal{B}$ of size at least $3s$, fix an arbitrary seed set $K_B \subset B$ of size s .

Let *Event 1, ... Event 4* refer to:

1. For all $v \in V$ and $B \in \mathcal{B}$ with $|B| \geq 3s$, $E(v, K_B) \geq |K_B|/2$ if and only if $v \in B$.
2. For all $v \in V$ and $B \in \mathcal{B}$ with $|B| \geq 3s$, $E(v, B) \geq .9|B|$ if $v \in B$ and $E(v, B) \leq .1|B|$ if $v \notin B$.
3. All $B \in \mathcal{B}$ and disjoint sets $T, S \subseteq B$ with $|T| = |U| = s$ satisfy $E(T, U) \geq .9s^2$
4. For every pair of disjoint vertex sets S, T with cardinalities $|S| = |T| = s$, such that no cluster $B \in \mathcal{B}$ intersects both S and T ($\forall B, B \cap S = \emptyset$ or $A \cap T = \emptyset$), there are at most $(.1)|S||T|$ edges between S and T .

LEMMA 5.8. *Events 1, 2, 3, 4 all occur with probability at least $1 - O(n^{-5})$.*

Proof. Events 1 and 2 occur with that probability by Lemma 5.5 and a union bound over the $O(n^2)$ possible combinations of base cluster B and vertex v . Events 3 and 4 follow from Lemma 5.6 using $X = S \cap T$.

We henceforth assume that events 1, ... 4 occur. We refer to the three bullets in the first half of Algorithm 3 as *Condition 1 ... 3*. We refer to the three bullets in the second half of Algorithm 3 as *Property 1 ... 3*.

5.2 Proof of Lemma 5.1 In this subsection we prove that every cluster $B \in \mathcal{B}$ of size at least $3s$ is added to \mathcal{A} at some point.

We fix some cluster $B \in \mathcal{B}$ of size at least $3s$. Let K be its seed. We show that when Algorithm 3 considers $S = K$, then it adds $A = B$ to \mathcal{A} . Event 1 and the definition of A implies $A = B$. Events 3, 4 and 2 guarantee that A satisfies conditions 1, 2 and the second part of condition 3 respectively. The first part of condition 3 is satisfied by assumption on the size of B .

5.3 Proof of Lemma 5.2 In this subsection we show that every cluster A added to \mathcal{A} satisfies $|A| \geq 3s$ and $A \in \mathcal{B}$.

LEMMA 5.9. *For every cluster A that satisfies conditions 1–3 there exists $B \in \mathcal{B}$ such that $|B \setminus A| < s$ and $|A \setminus B| < s$.*

Proof. Construct a bipartition S, T of A as follows. Consider the clusters $B_i \in \mathcal{B}$ in descending order of $|B_i \cap A|$, adding $B_i \cap A$ to whichever of S, T have fewer vertices. It is well known that this greedy algorithm satisfies $||S| - |T|| \leq |B^+ \cap A|$, where B^+ is the cluster with the largest intersection with A . Without loss of generality suppose that $|T| \geq |S|$. If $|S| \geq s$, then $E(S, T) \leq .1|S||T|$ by event 4 and Lemma 5.7, contradicting condition 1, which implies $E(S, T) \geq .9|S||T|$, again using Lemma 5.7. Therefore $|S| < s$. Therefore $|T| < s + |B^+ \cap A|$, so $3s \leq |S| + |T| < 2s + |B^+ \cap A|$, so $|B^+ \cap A| > s$. Therefore the greedy procedure placed all other vertices in S , so $|A \setminus B^+| = |S| < s$.

By Condition 3 and $|A \setminus B^+| < s$ we see that $|A \cap B^+| \geq 2s$. If $|B^+ \setminus A|$ were s or more, we would detect that because then $E(A \cap B^+, B^+ \setminus A) \geq .9|A \cap B^+||B^+ \setminus A|$ by event 3, which contradicts condition 2.

Every A that passes is approximately equal to some $B \in \mathcal{B}$ by Lemma 5.9. We now show that $A = B$.

First consider some vertex $v \in B$. We use Event 2 and properties 3 to write $E(v, A) \geq E(v, A \cap B) \geq |A \cap B| - .1|B| \geq (|A| - s) - .1(|A| + s) = .9|A| - 1.1s \geq .9|A| - \frac{1.1}{3}|A| > |A|/2$, so $B \subseteq A$ by condition 3. For $v \notin B$, we similarly write $E(v, A) = E(v, A \cap B) + E(v, A \setminus B) \leq .1|B| + s \leq .1(|A| + s) + s = .1|A| + 1.1s \leq .1|A| + \frac{1.1}{3}|A| < |A|/2$. Therefore $A = B$.

5.4 Proof of Lemma 5.3 The proofs of the first property and of the second property are identical: for each v and A we apply Lemma 5.5 to the set of edges between v and A , and use the union bound. (There are at most n^2 such sets.)

To prove the third property, observe that each cluster $B_i \in \mathcal{B}$ which intersects both S and T has c_i vertices in S and c'_i vertices in T , with $c_i + c'_i \leq 3s$. Therefore in the base clustering the number of edges between S and T is $\sum_i c_i c'_i \leq \sum_i (c_i + c'_i)^2 / 4$. By convexity and using $\sum_i (c_i + c'_i) \leq |S \cup T| = 12s$ and $\max_i (c_i + c'_i) \leq 3s$, this is bounded by $36s^2 / 4 = |S||T|/4$. By Lemma 5.6 applied to $X = S \cup T$, the mistakes add another $(12s)^2 / 40 = |S||T|/10$ edges, for a total of at most $.35|S||T|$ edges.

5.5 Proof of Lemma 5.4 Let $M(S, T) = 2E(S, T) - |S||T|$, which is the profit from merging clusters S and T into a new cluster.

LEMMA 5.10. *Let \mathcal{A} be the input partial clustering that Algorithm 3 output and \mathcal{C} be an optimal clustering. For any $C \in \mathcal{C}$ satisfying $|C| \geq 18s$ there exists a unique $A \in \mathcal{A}$ such that:*

1. $|C \setminus A| < 6s$
2. $|A| \geq 3|C|$ or $C \subseteq A$

Proof. Suppose $|C| \geq 18s$. Let A be the cluster in \mathcal{A} maximizing $|A \cap C|$.

Relabel the clusters $A_i \in \mathcal{A}$ so that $|A_1 \cap C| \geq |A_i \cap C|$ for all $1 \leq i \leq |\mathcal{A}|$. Let $S_k = \bigcup_{i=1}^k A_i \cap C$ and let $S = S_k$ where k is the smallest integer such that $|S_k| \geq 6s$. (If no such k exists, let S be $\bigcup_i A_i \cap C$ plus sufficient additional vertices from $C \setminus \bigcup_i A_i \cap C$ so that $|S| = 6s$.) Let $T = C \setminus S$. If $|T|$ were at least $6s$ then Lemma 5.7 and Property 3 would contradict optimality of \mathcal{C} so $|T| < 6s$. We know $|S| + |T| \geq 18s$ so therefore $|S| > 12s$. The definition of k and the relabeling imply $k = 1$, proving the first statement of the Lemma.

For $v \in C \setminus A$ write $E(v, C \cap A) \leq E(v, A) \leq .1|A|$ using Property 2b. Note that $M(C \setminus A, C \cap A) \geq 0$ by optimality of C , so therefore $0 \leq M(C \setminus A, C \cap A) = \sum_{v \in C \setminus A} M(v, C \cap A) = \sum_v 2E(v, C \cap A) - |C \cap A| \leq |C \setminus A|(.2|A| - |C \cap A|)$. Suppose $C \not\subseteq A$, hence

$|C \setminus A| > 0$. Therefore $0 \leq .2|A| - |C \cap A|$ hence using $|C| \geq 18s$ we see $|A| \geq 5|C \cap A| \geq \frac{10}{3}|C|$, proving the second statement of the Lemma.

We are now ready to prove Lemma 5.4. We first show that $\mathcal{A}' \subseteq \{C \in \mathcal{C} : |C| \geq 45s\}$.

Let A be the largest cluster in $\mathcal{A} \setminus \mathcal{C}$. For sake of contradiction suppose $|A| > 45s$. Consider the clustering $\mathcal{C}' = \{A\} \cup \{C \setminus A : C \in \mathcal{C}\}$. We will prove that \mathcal{C}' is strictly better than \mathcal{C} , contradicting the optimality of \mathcal{C} .

Let C_u denote the cluster of \mathcal{C} containing vertex u . Let $C_1, C_2 \dots C_l$ denote the clusters of \mathcal{C} with non-empty intersection with A^* . For \mathcal{C}' and \mathcal{C} , the objective function only differs for pairs $\{u, v\}$ such that $u \in C_u \cap A, v \in C_u \setminus A$ (in which case we charge the change to C_u), or such that $u \in C_u \cap A, v \in C_v \cap A$ with $C_u \neq C_v$ (in which case we charge half of the change to C_u and half to C_v). The change in objective function can thus be written as the sum, over $1 \leq i \leq l$, of $\Delta\text{profit}(i)$, where $\Delta\text{profit}(i)$ equals

$$-M(C_i \cap A, C_i \setminus A) + \frac{1}{2}M(C_i \cap A, A \setminus C_i).$$

Let D be the smaller of $C_i \cap A$ and $A \setminus C_i$, hence $|D| \leq (1/2)|A|$. Using Property 2a we see

$$\begin{aligned} E(D, A \setminus D) &= \sum_{v \in D} E(v, A \setminus D) \\ &\geq \sum_v E(v, A) - |D| \geq |D|(.9|A| - |D|). \end{aligned}$$

Therefore

$$\begin{aligned} M(D, A \setminus D) &= 2E(D, A \setminus D) - |D| \cdot |A \setminus D| \\ &\geq |D|(1.8|A| - 2|D| - |A \setminus D|) \\ &= |D|(.8|A| - |D|) \geq .3|D| \cdot |A| > 0 \end{aligned}$$

allowing us to write

$$\begin{aligned} (5.17) \quad \Delta\text{profit}(i) &\geq -M(C_i \cap A, C_i \setminus A) + \min(|C_i \cap A|, |A \setminus C_i|) \cdot \\ &\quad \cdot (.4|A| - (1/2) \min(|C_i \cap A|, |A \setminus C_i|)). \end{aligned}$$

We show that $\Delta\text{profit}(i) > 0$ for all i by cases on $|C_i \cap A|$.

Case 1: $|C_i| < .4|A|$. We trivially see $M(C_i \cap A, C_i \setminus A) \leq |C_i \cap A|(|C_i| - |C_i \cap A|)$. Using (5.17)

$$\begin{aligned} \Delta\text{profit}(i) &\geq -|C_i \cap A|(|C_i| - |C_i \cap A|) + \\ &\quad + \frac{1}{2}|C_i \cap A|(.8|A| - |C_i \cap A|) \\ &= |C_i \cap A| \left(-(|C_i| - |C_i \cap A|) + .4|A| - \frac{1}{2}|C_i \cap A| \right) \\ &\geq |C_i \cap A|(.4|A| - |C_i|) > 0. \end{aligned}$$

Case 2: $|C_i| \geq .4|A|$. We assumed $|A| \geq 45s$ hence $|C_i| \geq .4 \cdot 45s = 18s$. Let A' be the cluster promised by Lemma 5.10. Clearly $A' \notin \mathcal{C}$ hence $|C| \geq .4|A| \geq .4|A'| > \frac{1}{3}|A'|$ so by Lemma 5.10 we have $C_i \subseteq A$. Therefore $M(C_i \cap A, C_i \setminus A) = 0$ hence using (5.17) $\Delta\text{profit}(i) > 0$.

Thus in all cases the change in profit is positive: \mathcal{C}' is strictly better than \mathcal{C} , contradicting the optimality of \mathcal{C} . This completes the proof that $\mathcal{A}' \subseteq \{C \in \mathcal{C} : |C| \geq 45s\}$.

To show $\mathcal{A}' \subseteq \{C \in \mathcal{C} : |C| \geq 45s\}$, suppose for contradiction that there is some C not in \mathcal{A} with $|C| \geq 45s$. Lemma 5.10 implies there exists a cluster $A' \in \mathcal{A}$ that intersects C such that $|A'| > |C| \geq 45s$. We know \mathcal{C} is a partition so $A' \notin \mathcal{C}$, so this contradicts the fact that we already proved $\mathcal{A}' \subseteq \{C \in \mathcal{C} : |C| \geq 45s\}$.

This completes the proof of Lemma 5.4.

5.6 Runtime The runtime is dominated by that needed to check the third property. Algorithm 3 considers $O(n^{6s})$ different values the sets T and U , so this takes time $O(n^{12s})$.

References

- [1] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. In *STOC '05: Procs. 37th ACM Symposium on Theory of Computing*, pages 684–693, 2005.
- [2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. on Optimization*, 5(1):13–51, 1995.
- [3] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. In *SODA '98: Procs. 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 594–598, 1998.
- [4] Noga Alon and Joel H. Spencer. *The Probabilistic Method*, chapter 7.8, pages 115–116. Wiley, third edition edition, 2008.
- [5] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Mach. Learn.*, 56(1-3):89–113, 2004.
- [6] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297, 1999.
- [7] Béla Bollobás. *Random Graphs*, chapter 11.1. Cambridge University Press, second edition, 2001.
- [8] R. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Procs. 28th Foundations of Computer Science*, pages 280–285, 1987.
- [9] Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *SODA '08: Procs. 19th ACM-SIAM Symposium on Discrete Algorithms*, pages 268–276, 2008.
- [10] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383, 2005.
- [11] William Cohen and Jacob Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *KDD '02: Procs. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 475–480, 2002.
- [12] Erik Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2):172–187, 2006.
- [13] Micha Elsner and Warren Schudy. Bounding and comparing methods for correlation clustering beyond ILP. In *NAACL-HLT Workshop on Integer Linear Programming for Natural Language Processing (ILP-NLP 2009)*, pages 19–27, 2009.
- [14] Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Struct. Algorithms*, 16(2):195–208, 2000.
- [15] Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.
- [16] Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(1):249–266, 2006.
- [17] Roger Horn and Charles Johnson. *Matrix Analysis*, chapter 4.3. Cambridge University Press, 1985.
- [18] M. Jerrum. Large cliques elude the metropolis process. *Random Structures & Algorithms*, 3(4):347–359, 1992.
- [19] Thorsten Joachims and John Hopcroft. Error bounds for correlation clustering. In *ICML '05: Procs. 22nd International Conference on Machine Learning*, pages 385–392, 2005.
- [20] Jeong Han Kim and Van H. Vu. Concentration of multivariate polynomials and its applications. *Combinatorica*, 20(3):417–434, 2000.
- [21] Alexandra Kolla and Madhur Tulsiani. Playing random and expanding unique games. Unpublished manuscript, 2008.
- [22] L. Kucera. Expected complexity of graph partitioning problems. *Discrete Appl. Math.*, 57(2-3):193–212, 1995.
- [23] F. McSherry. Spectral partitioning of random graphs. In *FOCS '01: Procs. 42nd IEEE Foundations of Computer Science*, page 529, 2001.
- [24] Ron Shamir and Dekel Tsur. Improved algorithms for the random cluster graph model. *Random Structures and Algorithms*, 31(4):418–449, 2007.
- [25] Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *SODA '04: Procs. of the 15th ACM-SIAM Symposium on Discrete Algorithms*, pages 526–527, 2004.
- [26] Anke van Zuylen, Rajneesh Hegde, Kamal Jain, and David P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. In *SODA '07: Procs. 18th ACM-SIAM Symposium on Discrete Algorithms*, pages 405–414, 2007.