Subsampled Power Iteration: a New Algorithm for Block Models and Planted CSP’s

Vitaly Feldman* Will Perkins† Santosh Vempala‡

Abstract

We present a new algorithm for recovering planted solutions in two well-known models, the stochastic block model and planted constraint satisfaction problems, via a common generalization in terms of random bipartite graphs. Our algorithm achieves the best-known bounds for the number of edges needed for perfect recovery and its running time is linear in the number of edges used. The time complexity is significantly better than both spectral and SDP-based approaches. The main new features of the algorithm are two-fold: (i) the critical use of power iteration with subsampling, which might be of independent interest; its analysis requires keeping track of multiple norms of an evolving solution (ii) it can be implemented statistically, i.e., with very limited access to the input distribution.

*IBM Research - Almaden
†Georgia Tech. Supported by an NSF postdoctoral fellowship.
‡Georgia Tech. Supported in part by NSF award CCF-1217793.
1 Introduction

Partitioning a graph into parts based on the density of the edges within and between the parts is a fundamental algorithmic task both in its own right as a method of clustering data into similar pieces, and as a powerful subroutine of divide-and-conquer algorithms. There are many choices for number of parts required and the measure of the quality of a partition, and different choices give rise to algorithmic problems such as Max Clique, Max Cut, Uniform Sparsest Cut, and Min Bisection.

Finding an optimal graph partition is often an NP-hard problem in the worst case, and so the focus of research has turned to two directions beyond worst-case analysis, designing approximation algorithms and analyzing the average-case complexity of graph partitioning. In this paper we focus on the second direction.

Average-case complexity is the study of probability distributions over instances of computational problems. An efficient algorithm for such a distribution is one that runs in polynomial time in expectation or one that runs in polynomial time and outputs a correct solution with high probability over the distribution of instances.

The average-case study of graph partitioning problems is particularly rich, as the underlying distributions come from natural and widely studied models of random graphs. Such study was initiated in [12, 23] in which several graph partitioning problems were found to have efficient average-case algorithms.

Planted partitioning

One way to formulate a clean algorithmic problem and to model a data clustering problem in which an underlying truth is revealed through noisy data is to plant a partition in a random graph and draw edges at random according to a distribution biased towards the planted partition. A particularly simple model of random graph partitioning with a planted solution is the stochastic block or planted bisection model: a graph on \( n \) vertices is split into an equal bipartition, and edges within a part are added independently with probability \( p \), and edges crossing the partition added with probability \( q \). Boppana [10] gave an eigenvector-based algorithm for this model, and Jerrum and Sorkin [34] gave a Metropolis approach. Another example of planted partitioning is the planted \( k \)-coloring model [8] in which the vertex set is partitioned into \( k \) equal parts and then edges crossing the partition are added independently at random while edges within the partition are forbidden. Alon and Kahale [3] gave a spectral algorithm for this problem.

Later algorithms [21, 41, 25, 14, 11, 20] improved either the running time or the density at which the algorithms succeed, and in particular, Coja-Oghlan [15] showed that the planted partition in the stochastic block model can be partially recovered when the average degree is just a constant. Based on ideas from statistical physics, Decelle et al. [22] conjectured that in fact there is a sharp threshold for efficient recovery: if \( p = a/n, q = b/n \), and \((a - b)^2 < 2(a + b) \) then any non-trivial recovery of the planted partition is impossible, while if \((a - b)^2 > 2(a + b) \) then there is an efficient algorithm (polynomial in the size of the graph) that gives a partition with significant correlation to the planting. Mossel, Sly, and Neeman proved the lower bound [42], and then Massoulie [40] and Mossel, Sly, Neeman [43] independently analyzed algorithms proving the upper bound. See also [44, 38] for more on algorithms for this model.

Planted \( k \)-CSP’s

A broad and fundamental class of algorithmic problems is the class of boolean Constraint Satisfaction Problems (CSP’s). A width-\( k \) CSP is defined by a set of \( m \) predicates denoted by \( P_1, \ldots, P_m \)
and a set of $m$ $k$-tuples of boolean variables from the set $V = \{x_1, \ldots, x_n\}$ denoted by $C_1, \ldots, C_m$. Each predicate $P_i$ is a function from $\{\pm 1\}^k$ to $\{\pm 1\}$. Identifying $+1$ with TRUE and $-1$ with FALSE, a predicate $P_i$ is satisfied by an assignment $\sigma : V \rightarrow \{\pm 1\}$ if the evaluation of the predicate $P_i$ on the values assigned by $\sigma$ to the $k$-tuple of variables $C_i = (x_{i_1}, \ldots, x_{i_k})$ is TRUE. Given such a $k$-CSP the algorithmic task is to find an assignment $\sigma$ that maximizes the number of satisfied constraints.

The average-case complexity of $k$-CSP’s is a large area of research that intersects cryptography, computational complexity, probabilistic combinatorics and statistical physics. We describe two types of distributions over $k$-CSP instances that are addressed in our work.

In the planted $k$-SAT problem each constraint is a disjunction of $k$ literals, variables or their negations, e.g. $\{\overline{x}_5, x_6, x_{10}\}$ and is referred to as $k$-clause. A random instance of this problem is produced by choosing a random and uniform assignment $\sigma$ and then selecting $k$-clauses randomly from the set of $k$-clauses satisfied by $\sigma$. In the ‘noisy’ version of the problem unsatisfied clauses are also included with some probability. The algorithmic task is to recover the planted assignment $\sigma$ specifically.

It was noted in [7] that drawing satisfied $k$-clauses uniformly at random from all those satisfied by $\sigma$ often does not result in a difficult algorithmic problem even if the number of observed clauses is relatively small. However, by changing the proportions of clauses depending on the number of satisfied literals under $\sigma$, one can create a more challenging distribution over instances. Such ‘quiet plantings’ have been further studied in [35, 1, 39, 37]. Algorithms for solving instances with various values of relative proportions for planted 3-SAT were given in [27, 36, 16]. In this work we describe a general way to define such problems using a planting distribution $Q$. This distribution is defined over $\{\pm 1\}^k$ and for a point $z$ gives the relative probability of clauses in which the value that $\sigma$ assigns to the $k$-tuple of literals in the clause is $z$ (see Section 1 for the formal definition).

A related class of problems is one in which for some fixed predicate $P$, an instance is generated by choosing a planted assignment $\sigma$ randomly and uniformly and generating a set of $m$ random and uniform $P$-constraints. That is, each constraint is of the form $P(x_{i_1}, \ldots, x_{i_k}) = P(\sigma_{i_1}, \ldots, \sigma_{i_k})$, where $(x_{i_1}, \ldots, x_{i_k})$ is a randomly and uniformly chosen $k$-tuple of variables (without repetitions). Goldreich [32] proposed a one-way function based on the apparent hardness of these problems. In his proposal the predicate is chosen randomly. The hardness of such problems for other predicates, most notably noisy $k$-XOR-SAT, has been used in cryptographic applications including public key cryptosystems [2, 5], and secure two-party computation [33]. It has also been used to derive hardness of approximation [4] (for public discussions of these problems/assumptions see [6, 46]). Problems of this type are usually referred to as Goldreich’s pseudorandom generator (PRG).

Bogdanov and Qiao [9] show that an SDP-based algorithm of Charikar and Wirth [13] can be used to find the planted assignment for any predicate that is not pairwise-independent using $m = O(n)$ constraints. The same approach can be used to recover the input for any $t$-wise independent predicate using $O(n^{(t+1)/2})$ evaluations via the folklore birthday “paradox”-based reduction to $t = 1$ (see [45] for details).

The connection of planted CSP’s to graph partitioning is that many algorithms for planted CSP’s use graph partitioning, and spectral graph partitioning in particular, as a subroutine. Examples of such algorithms for some classes of constraint distributions include Flaxman’s algorithm for planted 3-SAT [27], Krivelevich and Vilenchik’s algorithm [36] that runs in expected polynomial time, and the algorithm of Coja-Oghlan, Cooper, Frieze [16] for planted 3-SAT distributions that include the quiet plantings described above. Closely related to the problem of solving planted CSP’s is the problem of refuting the satisfiability of non-planted random CSP’s. Many of the same spectral techniques have been applied here as well [30, 31, 17, 24, 29, 19].
Our results and techniques

We propose a natural bipartite stochastic block model that generalizes the classic stochastic block model defined above. The key motivation for the study of this model is that the two types of planted \(k\)-CSP’s described above can be reduced to our block model. We then give a new algorithm for solving the random instances of the model.

The algorithm is based on applying power iteration with a sequence of matrices subsampled from the original adjacency matrix. This is in contrast to previous algorithms that compute the eigenvectors (or singular vectors) of the full adjacency matrix. The new algorithm has several advantages.

- The algorithm matches the best-known (and in some cases the best-possible) performance with respect to the edge or constraint density needed for complete recovery of the planted partition or assignment. The algorithm for planted CSP’s nearly matches computational lower bounds for SDP hierarchies \[45\] and the class of statistical algorithms \[26\].

- The algorithm is fast, running in time linear in the number of edges or constraints used, unlike other approaches that require computing eigenvalues or solving semi-definite programs.

- The algorithm is conceptually simple and easy to describe and implement. In fact it can be implemented in the statistical model, with very limited access to the input graph.

- It is based on the idea of iteration with subsampling which may have further applications in the design and analysis of algorithms.

We now define the models and state our main theorems.

Bipartite stochastic block model

**Definition 1.** For \(\delta \in [0, 2] \setminus \{1\}\), \(n_1, n_2\) even, and \(\mathcal{P}_1 = A_1 \cup B_1, \mathcal{P}_2 = A_2 \cup B_2\) bipartitions of vertex sets \(V_1, V_2\) of size \(n_1, n_2\) respectively, we define the bipartite stochastic block model \(B(n_1, n_2, \mathcal{P}_1, \mathcal{P}_2, \delta, p)\) to be the random graph in which edges between vertices in \(A_1\) and \(A_2\) and \(B_1\) and \(B_2\) are added independently with probability \(\delta p\) and edges between vertices in \(A_1\) and \(B_2\) and \(B_1\) and \(A_2\) with probability \((2 - \delta) p\).

The algorithmic task for the bipartite block model is to recover one or both partitions (completely or partially) using as few edges and as little computational time as possible. In this work we will assume that \(n_1 \leq n_2\), and we will be concerned with the algorithmic task of recovering the partition \(\mathcal{P}_1\) completely, as this will allow us to solve the planted \(k\)-CSP problems described below. We define complete recovery of \(\mathcal{P}_1\) as finding the exact partition with high probability over the randomness in the graph and in the algorithm. We define partial \((\epsilon)\)-recovery as finding any partition that agrees with \(\mathcal{P}_1\) on at least \(1/2 + \epsilon\) fraction of vertices whp.

Note that setting \(n_1 = n_2 = n\), and identifying \(A_1\) and \(B_1\) and \(A_2\) and \(B_2\) gives the usual stochastic block model (with loops allowed).

**Theorem 1.** Assume \(n_1 \leq n_2\). There is an algorithm that completely recovers the partition \(\mathcal{P}_1\) in the bipartite stochastic block model \(B(n_1, n_2, \mathcal{P}_1, \mathcal{P}_2, \delta, p)\) with probability \(1 - o(1)\) as \(n_1 \to \infty\) using \(O\left(\frac{n_1 n_2 \log n_1}{(\delta - 1)^2}\right)\) edges in expectation and running in time \(O\left(\frac{n_1 n_2 \log n_1}{(\delta - 1)^2}\right)\).

Note that for the usual stochastic block model this gives an algorithm using \(O(n \log n)\) edges and \(O(n \log n)\) time, which is optimal for complete recovery since that many edges are needed for
every vertex to appear at least once. For any \( n_1, n_2 \), at least \( \sqrt{n_1 n_2} \) edges are necessary for even non-trivial partial recovery, as below that threshold the graph consists of small components.

For very lopsided graphs, with \( n_2 \gg n_1 \log^2 n_1 \), the running time is sublinear in the size of \( V_2 \); this requires careful implementation and is essential to achieving the running time bounds for planted CSP’s described below.

**Planted \( k \)-CSP’s**

We now describe a general model for planted satisfiability problems. For an integer \( k \), let \( \mathcal{C}_k \) be the set of all ordered \( k \)-tuples of literals from \( x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n \) with no repetition of variables. For a \( k \)-tuple of literals \( C \) and an assignment \( \sigma \), \( \sigma(C) \) denotes the vector of values that \( \sigma \) assigns to the literals in \( C \). A planting distribution \( Q : \{\pm 1\}^k \to [0, 1] \) is a PDF over \( \{\pm 1\}^k \).

**Definition 2.** Given a planting distribution \( Q : \{\pm 1\}^k \to [0, 1] \), and an assignment \( \sigma \in \{\pm 1\}^n \), we define the random constraint satisfaction problem \( F_{Q, \sigma}(n, m) \) by drawing \( m \) \( k \)-clauses from \( \mathcal{C}_k \) independently according to the distribution

\[
Q_{\sigma}(C) = \frac{Q(\sigma(C'))}{\sum_{C' \in \mathcal{C}_k} Q(\sigma(C'))}
\]

where \( \sigma(C) \) is the vector of values that \( \sigma \) assigns to the \( k \)-tuple of literals comprising \( C \).

**Definition 3.** The distribution complexity \( r(Q) \) of the planting distribution \( Q \) is the smallest \( r > 0 \) so that there is some \( S \subseteq [k], |S| = r \), so that the discrete Fourier coefficient \( \hat{Q}(S) \) is non-zero.

In other words, the distribution complexity of \( Q \) is \( r \) if \( Q \) is an \( (r - 1) \)-wise independent distribution on \( \{\pm 1\}^k \) but not \( r \)-wise independent. For any \( Q \) that is not the uniform distribution
over all clauses, we have $1 \leq r(Q) \leq k$ (the uniform distribution over all clauses does not reveal any information about $\sigma$ and we can define its complexity to be $\infty$).

**Theorem 2.** For any planting distribution $Q$, there exists an algorithm that for any assignment $\sigma$, given an instance of $F_{Q,\sigma}(n, m)$ completely recovers the planted assignment $\sigma$ for $m = O(n^{r/2} \log n)$ and using $O(n^{r/2} \log n)$ time, where $r \geq 2$ is the distribution complexity of $Q$. For distribution complexity $r = 1$, the algorithm gives non-trivial partial recovery with $O(n^{1/2} \log n)$ constraints and complete recovery with $O(n \log n)$ constraints.

We also show that the same result applies to recovering the planted assignment in Goldreich’s PRG defined above.

**Theorem 3.** For any predicate $P : \{\pm 1\}^k \to \{\pm 1\}$, there exists an algorithm that for any assignment $\sigma$, given $m$ random $P$-constraints completely recovers the planted assignment $\sigma$ for $m = O(n^{r/2} \log n)$ and using $O(n^{r/2} \log n)$ time, where $r \geq 2$ is the degree of the highest-degree non-zero Fourier coefficient of $P$. For $r = 1$, the algorithm gives non-trivial partial recovery with $O(n^{1/2} \log n)$ constraints and complete recovery with $O(n \log n)$ constraints.

**Comparison with previous work:**

The algorithm of Mossel, Neeman, and Sly [43] for the case $n_1 = n_2$ also runs in near linear time, while other known algorithmic approaches for planted partitioning that succeed near the optimal edge density [41, 15, 40] perform eigenvector or singular vector computations and thus require superlinear time.

For planted satisfiability, the algorithm of Flaxman for planted 3-SAT works for distributions with complexity $r \leq 2$ using $O(n)$ constraints, while the algorithm of Coja-Oghlan, Cooper, and Frieze [16] works for all planted 3-SAT distributions that exclude unsatisfied clauses and uses $O(n^{3/2} \ln^{10} n)$ constraints.

The only previous algorithm that finds the planted assignment in Goldreich’s PRG for all predicates is the SDP-based algorithm of Bogdanov and Qiao [9] with the folklore generalization to $r$-wise independent predicates (cf. [45]). Similar to our algorithm, it uses $\tilde{O}(n^{r/2})$ constraints. This algorithm effectively solves the noisy $r$-XOR-SAT instance and therefore can be also used to solve
our general version of planted satisfiability using $\tilde{O}(n^{r/2})$ clauses (via the reduction in Section 3). Notably for both this algorithm and ours, having a completely satisfying planted assignment plays no special role: the number of constraints required depends only on the distribution complexity.

Our algorithm is arguably simpler than the approach in [9] and substantially improves the running time even for small $k$. Another advantage of our approach is that it can be implemented using restricted access to the distribution of constraints referred to as statistical queries. Roughly speaking, for the planted SAT problem this access allows an algorithm to evaluate multi-valued functions of a single clause on randomly drawn clauses or to estimate expectations of such functions, without direct access to the clauses themselves. Recently, in [26], lower bounds on the number of constraints required for solving planted $k$-CSPs were proved. It is therefore important to understand the power of such algorithms for solving planted $k$-CSPs. A statistical implementation of our algorithm gives an upper bound that nearly matches the lower bound for the problem. See [26] for the formal details of the model and statistical implementation.

In Section 2 we describe the algorithm and analyze its performance. In Section 3 we present the reduction of the planted $k$-CSP problems to the bipartite stochastic block model. The appendix contains full details of the analysis.

2 The algorithm

We now present our algorithm for the bipartite stochastic block model. We define vectors $u$ and $v$ of dimension $n_1$ and $n_2$ respectively, indexed by $V_1$ and $V_2$, with $u_i = 1$ for $i \in A_1$, $u_i = -1$ for $i \in B_1$, and similarly for $v$. To recover the partition $P_1$ it suffices to find either $u$ or $-u$.

We will find $\pm u$ by multiplying a random initial vector $x_0$ by a sequence of centered adjacency matrices and their transposes. We form these matrices as follows: let $G_p$ be the random bipartite graph drawn from the model $B(n_1,n_2,P_1,P_2,\delta,p)$, and $T$ a positive integer. Then we form bipartite graphs $G_1,\ldots,G_T$ on the same vertex sets $V_1,V_2$ by placing each edge of $G_p$ uniformly and independently at random in one of the $T$ graphs. Thus $G_1,\ldots,G_T$ are i.i.d. with distribution $G_{p/T}$. Next we form the $n_1 \times n_2$ adjacency matrices $A_1,\ldots,A_T$ with rows indexed by $V_1$ and columns by $V_2$ with a 1 in entry $(i,j)$ if vertex $i \in V_1$ is joined to vertex $j \in V_2$. Finally we center the matrices by defining $M_i = A_i - \frac{T}{p} J$ where $J$ is the $n_1 \times n_2$ all ones matrix. We will denote by $M(q)$ the distribution of these matrices, with $q = p/T$. In other words, for $i \in A_1,j \in A_2$ or $i \in B_1,j \in B_2$, we have $M(q)_{ij} = 1-q$ with probability $\delta q$ and $-q$ otherwise; for $i \in A_1,j \in B_2$ or $i \in B_1,j \in A_2$, we have $M(q)_{ij} = 1-q$ with probability $(2-\delta)q$ and $-q$ otherwise, with all entries in the matrix independent.

In the bipartite block model, the subsampled matrices are independent, leading to clean analysis and a strong bound on the number of iterations required to solve the problem. The subsampling also mitigates the influence of high-degree vertices leading to significant improvement over the spectral approach for a large subclass of planted CSP’s.

The analysis of the algorithm proceeds by tracking a potential function, $U_i = x^i \cdot u$ for a sequence of unit vectors $x^0,x^1,\ldots$ of dimension $n_1$. We must bound various norms of the $x^i$’s as well as norms of a sequence of auxiliary vectors $y^1,y^2,\ldots$ of dimension $n_2$. We use superscripts to denote the current step of the iteration and subscripts for the components of the vectors, so $x^i_j$ is the $j$th coordinate of the vector after the $i$th iteration.

The basic iterative steps are the multiplications $y = M^T x$ and $x = My$. 
Algorithm: Subsampled Power Iteration.

1. Form \(2 \cdot T = 10 \log n_1\) matrices \(M_1, \ldots, M_T\) by uniformly and independently assigning each edge of the bipartite block model to a graph \(G_1, \ldots, G_T\), then forming the matrices \(M_i = A_i - \frac{p}{T} J\), where \(A_i\) is the adjacency matrix of \(G_i\) and \(J\) is the all ones matrix.

2. Sample \(x \in \{\pm 1\}^{n_1}\) uniformly at random and let \(x^0 = \frac{x}{\sqrt{n_1}}\).

3. For \(i = 1\) to \(T\) let

\[
y^i = \frac{M_{2i-1}^T x^{i-1}}{\|M_{2i-1}^T x^{i-1}\|}; \quad x^i = \frac{M_{2i} y^i}{\|M_{2i} y^i\|}; \quad z^i = \text{sgn}(x^i).
\]

4. For each coordinate \(j\) of \(z^i\) take the majority vote for all \(i \in \{T/2, \ldots, T\}\) and call this vector \(\overline{v}\):

\[
\overline{v}_j = \text{sgn} \left( \sum_{i=T/2}^{T} z^i_j \right).
\]

5. Return the partition indicated by \(\overline{v}\).

The analysis of the resampled power iteration algorithm proceeds in four phases, during which we track the progress of two vectors \(x^i\) and \(y^i\), as measured by their inner product with \(u\) and \(v\) respectively. We define \(U_i := u \cdot x^i\) and \(V_i := v \cdot y^i\). Here we give an overview of each phase; the complete analysis is in Appendix A.

- **Phase 1.** Within \(\log n_1\) iterations, \(|U_i|\) reaches \(\log \log n_1\). We show that conditioned on the
value of $U_i$, there is at least a $1/2$ chance that $|U_{i+1}| \geq 2|U_i|$; that $U_i$ never gets too small; and that in $\log n_1$ steps, a run of $\log \log n_1$ doublings pushes the value of $U_i$ above $\log \log n_1$.

- **Phase 2.** After reaching $\log \log n_1$, $U_i$, makes steady, predictable progress, doubling at each step whp until it reaches $\Theta(\sqrt{n_1})$, at which point we say $x^i$ has strong correlation with $u$.

- **Phase 3.** Once $x^i$ is strongly correlated with $u$, we show that $z^{i+1}$ agrees with either $u$ or $-u$ on a large fraction of coordinates.

- **Phase 4.** We show that taking the majority vote of the coordinate-by-coordinate signs of $z^i$ over $O(\log n_1)$ additional iterations gives complete recovery whp.

### Number of edges used and running time

To make progress at each step, each iteration uses a matrix drawn from $M(q)$ with $q = O((\delta - 1)^{-2}(n_1n_2)^{-1/2})$. All together there are $O(\log n_1)$ iterations, and so we can take $p = O(\log n_1(\delta - 1)^{-2}(n_1n_2)^{-1/2})$.

If $n_2 = \Theta(n_1)$, then a straightforward implementation of the algorithm runs in time linear in the number of edges used: each entry of $x^i = My^i$ (resp. $y^i = M^T x^{i-1}$) can be computed as a sum over the edges in the graph associated with $M$. The rounding and majority vote are both linear in $n_1$.

However, if $n_2 \gg n_1$, then simply initializing the vector $y^i$ will take too much time. In this case, we have to implement the algorithm more carefully. We still maintain the vectors $x^0, x^1, \ldots$ as before, but instead of computing the vectors $y^i$ at each step, we create a set $S^i \subset V_2$ of all vertices with degree at least 1 in the current graph $G_i$. The size of $S^i$ is bounded by the number of edges in $G_i$, and checking membership can be done in constant time with a data structure of size $O(|S^i|)$ that requires expected time $O(|S^i|)$ to create.

Now instead of computing $y^i = M_{2i-1}^T x^{i-1}$, we create the set $S^i$. Then to compute $x^i = M_{2i}y^i$, we do the following computation:

$$x^i_j = \left( \sum_{(e,(j,k)) \in G_{2i}, k \in S^i} y^i_k \right) - q \left( \sum_{k=1}^{n_1} x^{i-1}_k \right) \cdot |\{ e = (j,k) \in G_{2i} : k \notin S^i \}|$$

$\sum_{k=1}^{n_1} x^{i-1}_k$ only needs to be computed once per iteration and runs in time $O(n_1) = O(\sqrt{n_1n_2})$. The quantity $|\{ e = (j,k) \in G_{2i} : k \notin S^i \}|$ can be computed with the membership data structure in time linear in the number of edges of $G_{2i}$. Finally, $\sum_{(e,(j,k)) \in G_{2i}, k \in S^i} y^i_k$ can be computed in time linear in the number of edges too, by looking up the edges incident to vertex $k \in V_2$ in the previous graph to determine $y^i_k$.

### 3 Reduction of planted $k$-CSP’s to the block model

Here we describe how solving the bipartite block model suffices to solve the planted $k$-CSP problems.

First consider a planted $k$-SAT problem $F_{Q, \sigma}(n, m)$ with distribution complexity $r$. Let $S \subseteq [k]$, $|S| = r$, be such that $\hat{Q}(S) = \eta \neq 0$. We will show that subsampling $r$ literals with indices in the set $S$ from each $k$-clause induces a distribution on $r$-constraints defined by $Q^S : \{-1, 1\}^r \rightarrow \mathbb{R}^+$ of the form $Q^S(C) = \delta/2^r$ for $|C|$ even, $Q^S(C) = (2-\delta)/2^r$ for $|C|$ odd, for some $\delta \in [0, 2)$, $\delta \neq 1$, where $|C|$ is the number of TRUE literals in $C$ under $\sigma$. This reduction allows us to focus on algorithms for
the specific case of a parity-based distribution on $k$-clauses with distribution complexity $k$. Recall that for a function $f : \{-1, 1\}^k \to \mathbb{R}$, its Fourier coefficients are defined for each subset $S \subseteq [k]$ as

$$\hat{f}(S) = \mathbb{E}_{x \sim \{-1, 1\}^k} [f(x)\chi_S(x)]$$

where $\chi_S$ are the Walsh basis functions of $\{-1, 1\}^k$ with respect to the uniform probability measure, i.e., $\chi_S(x) = \prod_{i \in S} x_i$.

**Lemma 1.** If the function $Q : \{-1, 1\}^k \to \mathbb{R}^+$ defines a distribution $Q_\sigma$ on $k$-clauses with distribution complexity $r$ and planted assignment $\sigma$, then for some $S \subseteq [k], |S| = r$ and $\delta \in [0, 2 \setminus \{1\}$, choosing $r$ literals with indices in $S$ from a clause drawn randomly from $Q_\sigma$ yields a random $r$-clause from $Q^\delta_\sigma$.

**Proof.** From Definition 3 we have that there exist $S$ with $|S| = r$ such that $\hat{Q}(S) \neq 0$. Note that by definition,

$$\hat{Q}(S) = \mathbb{E}_{x \sim \{-1, 1\}^k} [Q(x)\chi_S(x)] = \frac{1}{2^k} \sum_{x \in \{-1, 1\}^k} Q(x)\chi_S(x)$$

measures the difference between the probability under $Q$ that the number of true values in $\{x_i\}_{i \in S}$ is even and the probability that the number of true values in $\{x_i\}_{i \in S}$ is odd. By the definition of $Q_\sigma$, this probability being different from 0 is equivalent to the existence of $\delta \neq 1$ such that an $r$-clause generated by choosing the literals with indices in $S$ from a $k$-clause chosen randomly from $Q_\sigma$ is distributed according to $Q^\delta_\sigma$ defined as above (over $r$-clauses).

Next we describe how the parity distribution on $r$-constraint induces a bipartite block model. Let $V_1$ be the collection of all ordered $\lfloor r/2 \rfloor$-tuples of literals of the given variable set, and $V_2$ the collection of all ordered $\lceil r/2 \rceil$-tuples. We have $n_1 = |V_1| = \binom{2n}{\lfloor r/2 \rfloor}$ and $n_2 = |V_2| = \binom{2n}{\lceil r/2 \rceil}$. We partition each set into two parts based on the parity of the number of true literals in the tuples under the planted assignment $\sigma$: $A_1 \subseteq V_1, A_2 \subseteq V_2$ are the sets of $\lfloor r/2 \rfloor, \lceil r/2 \rceil$-tuples respectively with an even number of true literals, and $B_1, B_2$ are the sets with an odd number of true literals.

For each $r$-constraint $(l_1, l_2, \ldots, l_r)$, we add an edge in the block model between the tuples $(l_1, \ldots, l_{\lfloor r/2 \rfloor}) \in V_1$ and $(l_{\lceil r/2 \rceil + 1}, \ldots, l_r) \in V_2$. A constraint drawn according to $Q^\delta_\sigma$ induces a random edge between $A_1$ and $A_2$ or $B_1$ and $B_2$ with probability $\delta$ and between $A_1$ and $B_2$ or $B_1$ and $A_2$ with probability $2 - \delta$, exactly the distribution of a single edge in the bipartite block model.

Solving this bipartite block model completely for the partition $P_1 = A_1 \cup B_1$ divides the $\lfloor r/2 \rfloor$-tuples of literals into even and odd parity sets, and then solving a set of linear equations mod 2 recovers the parity of each individual literal, reconstructing either $\sigma$ or $-\sigma$.

The reduction from Goldreich’s PRG to the bipartite block model is even simpler. By definition, the value of the predicate is correlated with the parity function of some $r$ of the $k$ inputs of the predicate (see for example 9). Therefore the input can be seen as produced by the noisy $r$-XOR predicate on random and uniform $r$-tuples of variables. The $r$-tuples for which this predicate is equal to 1 give an instance of noisy $r$-XOR-SAT. A bipartite block model can now be formed on $\lfloor r/2 \rfloor$ and $\lceil r/2 \rceil$-tuples of variables (instead of literals) analogously to the construction above.

## 4 Comparison with spectral approach

As noted above, many approaches to graph partitioning problems and planted satisfiability problems use eigenvalues or singular vectors. These algorithms are essentially based on the signs of the top
Proposition 2. Let $E$ be a dimensional vector whose entries are 1 if the corresponding vertex is in $A_1$ and −1 otherwise. Define the $n_2$ dimensional vector $v$ analogously. The next propositions summarize properties of $M$.

Proposition 1. $\mathbb{E}(M) = (\delta - 1)p vw^T$.

Proposition 2. Let $M_1$ be the rank-1 approximation of $M$ drawn from $M(p)$. Then $\|M_1 - \mathbb{E}(M)\| \leq 2\|M - \mathbb{E}(M)\|$.

Proof. Using the triangle inequality and then the optimality of $M_1$, $\|M_1 - \mathbb{E}(M)\| \leq \|M - \mathbb{E}(M)\| + \|M - M_1\| \leq 2\|M - \mathbb{E}(M)\|$. \hfill \Box

Proposition 3. Let $A$ be a random $n_1 \times n_2$ matrix with independent entries in the range $[-1, 1]$ with mean zero and variance at most $\sigma^2$. Suppose $n_1 \leq n_2$. Then with probability $1 - o(1)$, $\|A\|_2 \leq C\sigma\sqrt{n_2}$.

The above lemmas suffice to show high correlation between the top singular vector and the vector $u$ when $n_2 = \Theta(n_1)$ and $p = \Omega(\log n_1 / n)$. This is because the norm of $\mathbb{E}(M)$ is $p\sqrt{n_1 n_2}$; this is higher than $O(\sqrt{n_2})$, the norm of $M - \mathbb{E}(M)$. Therefore the top singular vector of $M$ will be correlated with the top singular vector of $\mathbb{E}(M)$. The latter is a rank-1 matrix with $u$ as its right singular vector.

However, when $n_2 \gg n_1$ (e.g., $k$ odd), the norm of the zero-mean matrix $M - \mathbb{E}(M)$ is in fact much larger than the norm of $\mathbb{E}(M)$. (i.e., $p\sqrt{n_1 n_2}$ vs $\sqrt{n_2}$, the former is $O(1)$, while the latter is $\Omega((n_2/n_1)1/4)$). In other words, the top singular value of $M$ is much larger than the value obtained by the vector corresponding to the planted assignment! In spite of this, one can exploit correlations to recover the planted vector with our resampling algorithm.
References

[1] Dimitris Achlioptas, Haixia Jia, and Cristopher Moore. Hiding satisfying assignments: Two are better than one. *J. Artif. Intell. Res.(JAIR)*, 24:623–639, 2005.

[2] Michael Alekhnovich. More on average case vs approximation complexity. *Computational Complexity*, 20(4):755–786, 2011.

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

[4] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In *Proceedings of the 44th symposium on Theory of Computing*, pages 805–816. ACM, 2012.

[5] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 171–180. ACM, 2010.

[6] Boaz Barak. Truth vs proof: The unique games conjecture and Feige’s hypothesis. [http://windowsontheory.org/2012/07/31/truth-vs-proof-the-unique-games-conjecture-and-feiges-hypothesis/](http://windowsontheory.org/2012/07/31/truth-vs-proof-the-unique-games-conjecture-and-feiges-hypothesis/) July 2012.

[7] Wolfgang Barthel, Alexander K Hartmann, Michele Leone, Federico Ricci-Tersenghi, Martin Weigt, and Riccardo Zecchina. Hiding solutions in random satisfiability problems: A statistical mechanics approach. *Physical review letters*, 88(18):188701, 2002.

[8] Avrim Blum and Joel Spencer. Coloring random and semi-random $i_k, k_i$-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995.

[9] Andrej Bogdanov and Youming Qiao. On the security of goldreich’s one-way function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 392–405. Springer, 2009.

[10] Ravi B Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pages 280–285. IEEE, 1987.

[11] Julia Böttcher. Coloring sparse random k-colorable graphs in polynomial expected time. *Mathematical Foundations of Computer Science 2005*, page 156, 2005.

[12] Thang Nguyen Bui, Soma Chaudhuri, Frank Thomson Leighton, and Michael Sipser. Graph bisection algorithms with good average case behavior. *Combinatorica*, 7(2):171–191, 1987.

[13] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending grothendieck’s inequality. In *FOCS*, pages 54–60, 2004.

[14] Amin Coja-Oghlan. A spectral heuristic for bisecting random graphs. *Random Structures & Algorithms*, 29:3:351–398, 2006.

[15] Amin Coja-Oghlan. Graph partitioning via adaptive spectral techniques. *Combinatorics, Probability & Computing*, 19(2):227, 2010.

[16] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. An efficient sparse regularity concept. *SIAM Journal on Discrete Mathematics*, 23(4):2000–2034, 2010.
[17] Amin Coja-Oghlan, Andreas Goerdt, and André Lanka. Strong refutation heuristics for random k-sat. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pages 310–321. Springer, 2004.

[18] Amin Coja-Oghlan, Andreas Goerdt, André Lanka, and Frank Schädlich. Certifying unsatisfiability of random 2k-sat formulas using approximation techniques. In Fundamentals of Computation Theory, pages 15–26. Springer, 2003.

[19] Amin Coja-Oghlan, Andreas Goerdt, André Lanka, and Frank Schädlich. Techniques from combinatorial approximation algorithms yield efficient algorithms for random 2k-sat. Theoretical Computer Science, 329(1):1–45, 2004.

[20] Amin Coja-Oghlan and André Lanka. Finding planted partitions in random graphs with general degree distributions. SIAM Journal on Discrete Mathematics, 23(4):1682–1714, 2009.

[21] Anne Condon and Richard M Karp. Algorithms for graph partitioning on the planted partition model. Random Structures & Algorithms, 18(2):116–140, 2001.

[22] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. Physical Review E, 84(6):066106, 2011.

[23] ME Dyer and AM Frieze. Fast solution of some random np-hard problems. In Proceedings of the 27th Annual Symposium on Foundations of Computer Science, pages 331–336. IEEE Computer Society, 1986.

[24] Uriel Feige and Eran Ofek. Easily refutable subformulas of large random 3cnf formulas. In Automata, languages and programming, pages 519–530. Springer, 2004.

[25] Uriel Feige and Eran Ofek. Spectral techniques applied to sparse random graphs. Random Structures & Algorithms, 27(2):251–275, 2005.

[26] Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. arXiv preprint arXiv:1311.4821, 2013.

[27] Abraham Flaxman. A spectral technique for random satisfiable 3cnf formulas. In Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pages 357–363. Society for Industrial and Applied Mathematics, 2003.

[28] Michael L Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0 (1) worst case access time. Journal of the ACM (JACM), 31(3):538–544, 1984.

[29] Joel Friedman, Andreas Goerdt, and Michael Krivelevich. Recognizing more unsatisfiable random k-sat instances efficiently. SIAM Journal on Computing, 35(2):408–430, 2005.

[30] Andreas Goerdt and Michael Krivelevich. Efficient recognition of random unsatisfiable k-sat instances by spectral methods. In STACS 2001, pages 294–304. Springer, 2001.

[31] Andreas Goerdt and André Lanka. Recognizing more random unsatisfiable 3-sat instances efficiently. Electronic Notes in Discrete Mathematics, 16:21–46, 2003.

[32] Oded Goldreich. Candidate one-way functions based on expander graphs. IACR Cryptology ePrint Archive, 2000:63, 2000.
[33] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Proceedings of the 40th annual ACM symposium on Theory of computing, pages 433–442. ACM, 2008.

[34] Mark Jerrum and Gregory B Sorkin. The metropolis algorithm for graph bisection. Discrete Applied Mathematics, 82(1):155–175, 1998.

[35] Haixia Jia, Cristopher Moore, and Doug Strain. Generating hard satisfiable formulas by hiding solutions deceptively. In PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, volume 20, page 384. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[36] Michael Krivelevich and Dan Vilenchik. Solving random satisfiable 3cnf formulas in expected polynomial time. In Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pages 454–463. ACM, 2006.

[37] Florent Krzakala, Marc Mézard, and Lenka Zdeborová. Reweighted belief propagation and quiet planting for random k-sat. arXiv preprint arXiv:1203.5521, 2012.

[38] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. Proceedings of the National Academy of Sciences, 110(52):20935–20940, 2013.

[39] Florent Krzakala and Lenka Zdeborová. Hiding quiet solutions in random constraint satisfaction problems. Physical review letters, 102(23):238701, 2009.

[40] Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In STOC 2014: 46th Annual Symposium on the Theory of Computing, pages 1–10, 2014.

[41] Frank McSherry. Spectral partitioning of random graphs. In Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on, pages 529–537. IEEE, 2001.

[42] Elchanan Mossel, Joe Neeman, and Allan Sly. Stochastic block models and reconstruction. arXiv preprint arXiv:1202.1499, 2012.

[43] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. arXiv preprint arXiv:1311.4115, 2013.

[44] Raj Rao Nadakuditi and Mark EJ Newman. Graph spectra and the detectability of community structure in networks. Physical review letters, 108(18):188701, 2012.

[45] Ryan O’Donnell and David Witmer. Goldreich’s prg: Evidence for near-optimal polynomial stretch. In Conference on Computational Complexity, 2014.

[46] Luca Trevisan. More ways to prove unsatisfiability of random k-sat. http://lucatrevisan.wordpress.com/2007/08/21/more-ways-to-prove-unsatisfiability-of-random-k-sat/, August 2007.
A  Analysis of the subsampled power iteration algorithm

We abuse notation and let $A_1, B_1, A_2, B_2$ denote the sets of coordinates of the corresponding vertex sets. Recall that $u \in \{\pm 1\}^{n_1}$ is 1 on $A_1$ and $-1$ on $B_1$, and $v \in \{\pm 1\}^{n_2}$ is 1 on $A_2$, $-1$ on $B_2$. $M(q)$ is a random $n_1 \times n_2$ matrix where the entries are independent and the entry $(i, j)$ takes value $1 - q$ with probability $\delta q$, $-q$ otherwise if $i \in A_1, j \in A_2$ or $i \in B_1, j \in B_2$, and value $1 - q$ with probability $(2 - \delta)q$, $-q$ otherwise if $i \in A_1, j \in B_2$ or $i \in B_1, j \in A_2$. We assume WLOG that $\delta > 1$.

Set $q = \frac{100}{(\delta - 1)^2 \sqrt{n_1 n_2}}$ and $T = 10 \log n_2$. For convenience we denote $d = 100/(\delta - 1)^2$.

We begin with some preliminary facts on the effect of multiplying a unit vector by $M(q)$ or $M(q)^T$. We use these estimates repeatedly in what follows.

**Proposition 4.** Let $M \sim M(q)$ and let $x$ and $y$ be unit vectors of dimension $n_1$ and $n_2$ respectively. Then

1. $\mathbb{E}(u \cdot (My)) = (\delta - 1)n_1 q(v \cdot y)$ and $\text{var}(u \cdot (My)) = n_1 q(1 + o(1))$.
2. $\mathbb{E}(v \cdot (M^T x)) = (\delta - 1)n_2 q(u \cdot x)$ and $\text{var}(v \cdot (M^T x)) = n_2 q(1 + o(1))$.
3. $\mathbb{E} \|My\|^2 = n_1 q(1 + o(1)) + (\delta - 1)^2 n_1 q^2 (v \cdot y)^2 (1 + o(1))$.
4. $\mathbb{E} \|M^T x\|^2 = n_2 q(1 + o(1)) + (\delta - 1)^2 n_2 q^2 (u \cdot x)^2 (1 + o(1))$.
5. $\text{var}(\|My\|^2) = O(n_1 q \|y\|_1 \cdot \|y\|^3_\infty + q^3 \|y\|^4_1 + q \|y\|^2_\infty \cdot \|y\|^2_1 + q + q^2 \|y\|^2_1)$.
6. $\text{var}(\|M^T x\|^2) = O(n_2 n_1 q \|x\|_\infty^4)$.

**Proof.** If $j \in A_1$,

$$
\mathbb{E}(My)_j = -\sum_{i=1}^{n_2} q y_i + \sum_{i \in A_2} \delta q y_i + \sum_{i \in B_2} (2 - \delta) q y_i
$$

$$
= (\delta - 1) q (v \cdot y)
$$

and similarly for $j \in B_1$.

This gives

$$
\mathbb{E}(u \cdot (My)) = (\delta - 1)n_1 q(v \cdot y)
$$

$$
\text{var}(u \cdot (My)) = n_1 q \|y\|^2_2 + O(n_1 q^2 \|y\|^2_2)
$$

$$
= n_1 q \|y\|^2_2 (1 + O(q))
$$

Then if $j \in A_2$,

$$
\mathbb{E}(M^T x)_j = -\sum_{i=1}^{n_1} q x_i + \sum_{i \in A_1} \delta q x_i + \sum_{i \in B_1} (2 - \delta) q x_i
$$

$$
= (\delta - 1) q (u \cdot x)
$$

$$
\text{var}(M^T x)_i = \sum_{j \in A_1} \delta q (1 - \delta q) x_j^2 + \sum_{j \in B_1} (2 - \delta) q (1 - (2 - \delta) q) x_j^2
$$
and similarly for \( j \in B_2 \).

This gives

\[
\mathbb{E}(v \cdot (M^T x)) = (\delta - 1)n_2q(u \cdot x)
\]

\[
\text{var}(v \cdot (M^T x)) = n_2q|x|^2(1 + O(q))
\]

Finally we have

\[
\mathbb{E}(\|My\|_2^2) = n_1q\|y\|_2^2(1 + O(q)) + (\delta - 1)^2q^2n_1(v \cdot y)^2
\]

and

\[
\text{var}(\|My\|_2^2) = \sum_{i=1}^{n_1} \text{var}((My)_i^2)
\]

\[
\leq \sum_{i=1}^{n_1} \mathbb{E}((My)_i^4)
\]

\[
= O(n_1(\|y\|_\infty^6\|y\|_1q + q^4\|y\|_1^4 + q^2\|y\|_2^6 + q^3\|y\|_2^3\|y\|_2^3 + q^2\|y\|_\infty^2\|y\|_1^2))
\]

\[
= O(n_1q(\|y\|_\infty^3\|y\|_1 + q^3\|y\|_1^4 + q + q^2\|y\|_2^2 + q\|y\|_\infty^2\|y\|_1^2))
\]

and

\[
\mathbb{E}(\|M^Tx\|_2^2) = n_2q\|x\|_2^2(1 + O(q)) + (\delta - 1)^2q^2n_2(u \cdot x)^2
\]

and

\[
\text{var}(\|M^Tx\|_2^2) = \sum_{i=1}^{n_2} \text{var}((M^Tx)_i^2)
\]

\[
\leq \sum_{i=1}^{n_2} \mathbb{E}((M^Tx)_i^4)
\]

\[
= n_2\|x\|_\infty^4 \cdot O(n_1q + q_1^2q^2 + q_1^3q^3 + q_1^4q^4)
\]

\[
= O(n_2n_1q\|x\|_\infty^4)
\]

Next we show the normalizing factors \( \|My\|_2 \) and \( \|M^Tx\|_2 \) are concentrated at each step; the \( l_\infty \) norms of the \( x^i \)'s are bounded over all iterations, and the \( l_\infty \) and \( l_1 \) norms of the \( y^i \)'s are bounded. This proposition is critical in ensuring steady progress of our potential functions.

**Lemma 2.** With probability \( 1 - O\left(Tn_1^{-1/6}\right) \), for all \( i = 1, \ldots, T \),

1. \( \|Mi^iy^i\|_2^2 = (n_1q\|y^i\|_2^2 + (\delta - 1)^2n_1q^2(v \cdot y^i)^2)(1 + o(1)) \)
2. \( \|M_i^Tx^i\|_2^2 = (n_2q\|x^i\|_2^2 + (\delta - 1)^2n_2q^2(u \cdot x^i)^2)(1 + o(1)) \)
3. \( \|x^i\|_\infty \leq n_1^{-1/3} \)
4. \( \|y^i\|_\infty \leq n_2^{-1/4}n_1^{-1/12} \)
5. \(\|y^i\|_1 \leq 4\sqrt{n_2 n_1 q}\)

Proof. We begin by showing that

\[
\left| \left\{ j : |y^i_j| > \sqrt{\frac{2qn_1}{n_2}} \right\} \right| \leq 3n_2 n_1 q. \tag{1}
\]

We bound the number \(L\) of \((1-q)\) entries in \(M_{i-1}\). \(L\) is stochastically bounded by a \(\text{Binom}(n_2 n_1, 2q)\) random variable, and so,

\[
\Pr[L \geq 3n_2 n_1 q] \leq e^{-qn_2 n_1} = e^{-\Theta(\sqrt{n_2 n_1})}.
\]

The remaining entries have value \(-q\). If the \(j\)th row of \(M_{i-1}\) has only \(-q\) entries, then

\[
|y^i_j| \leq \frac{q\|x^{i-1}\|_1}{\sqrt{n_2 q/2}} \leq \sqrt{\frac{2qn_1}{n_2}}
\]

using (2) inductively. This proves (1).

To prove (5), partition the coordinates of \(y^i\) into two sets \(\Delta\) and \(\overline{\Delta}\), with \(\Delta\) corresponding to rows of \(M_{i-1}\) with every entry \(-q\), and \(\overline{\Delta}\) the rest. Then

\[
\|y^i\|_1 \leq \sum_{j \in \Delta} |y^i_j| + \sum_{j \in \overline{\Delta}} |y^i_j|
\leq \sqrt{\frac{2qn_1}{n_2}}|\Delta| + \sqrt{|\overline{\Delta}|} \quad \text{using part (2) inductively}
\leq \sqrt{2n_2 n_1 q} + \sqrt{3n_2 n_1 q}
\leq 4\sqrt{qn_2 n_1}
\]

We show by induction that whp the following hold for \(i = 1, \ldots T\):

1. \(\|M_i y^i\|_2^2 = \mathbb{E} (\|M_i y^i\|_2^2) (1 + O(n_1^{-1/8}))\)
2. \(\|M^T_i x^i\|_2^2 = \mathbb{E} (\|M^T_i x^i\|_2^2) (1 + O(n_1^{-1/12}))\)
3. \(\|x^i\|_{\infty} \leq n_1^{-1/3}\)
4. \(\|y^i\|_{\infty} \leq n_2^{-1/4} n_1^{-1/12}\)

Conditional on \(y^i\) and \(x^i\) respectively, we have

\[
\mathbb{E} [\|M_i y^i\|_2^2] = n_1 q + (\delta - 1)^2 n_1 q^2 (v \cdot y^i)^2 + O(n_1 q^2)
\]

\[
\mathbb{E} [\|M^T_i x^i\|_2^2] = n_2 q + (\delta - 1)^2 n_2 q^2 (u \cdot x^i)^2 + O(n_2 q^2)
\]
Using Chebyshev and part (3),
\[
\Pr \left[ \|M_i y^i\|_2^2 - \mathbb{E} \left( \|M_i y^i\|_2^2 \right) > \alpha \mathbb{E} \left( \|M_i y^i\|_2^2 \right) \right] \\
\leq \frac{\operatorname{var}(\|M_i y^i\|_2^2)}{n_2^2 q^2 \alpha^2} \\
= \alpha^{-2} \cdot O \left( \frac{\| \|y\|_1 \cdot \|y\|_\infty^3 + q^3 \|y\|_1^4 + q \|y\|_\infty^2 \cdot \|y\|_1^2 + q + q^2 \|y\|_1^2}{n_1 q} \right) \\
= \alpha^{-2} \cdot O \left( \frac{q^{1/2} n_2^{-1/4} n_1^{1/4} + q^3 n_2^2 n_1^3 + q^5 n_2^{1/6} n_1^{5/6} + q^3 n_2 n_1}{n_1 q} \right) \\
= \alpha^{-2} \cdot O \left( q^{1/2} n_2^{-1/4} n_1^{-3/4} + q^3 n_2 n_1 + q n_2^{1/2} n_1^{-1/6} + n_1^{-1} + q^2 n_2 \right) \\
= \alpha^{-2} \cdot O \left( n_1^{-1/2} + n_1^{-1} + n_1^{-2/3} + n_1^{-1} + n_1^{-1} \right) \\
= O \left( \frac{n_1^{-1/4}}{n_1^{1/2} \alpha^2} \right) \\
= O \left( n_1^{-1/4} \right) \text{ for } \alpha = n_1^{-1/8}.
\]

Similarly, using Chebyshev and part (4),
\[
\Pr \left[ \|M_i^T x^i\|_2^2 - \mathbb{E} \left( \|M_i^T x^i\|_2^2 \right) > \alpha \mathbb{E} \left( \|M_i^T x^i\|_2^2 \right) \right] \\
\leq \frac{\operatorname{var}(\|M_i^T x^i\|_2^2)}{n_2^2 q^2 \alpha^2} \\
= O \left( \frac{n_1 \|x^i\|_\infty^4}{n_2 q \alpha^2} \right) \\
= O \left( \frac{1}{n_2 n_1^{1/3} q \alpha^2} \right) \\
= O \left( \frac{n_1^{1/6}}{n_2^{1/2} \alpha^2} \right) \\
= O \left( n_1^{-1/6} \right) \text{ for } \alpha = n_1^{-1/12}.
\]

To prove (3), note that
\[
\|x^{i+1}\|_\infty = \max_{j \in [n_1]} \frac{|(M_i^T y^i)_j|}{\|M_i^T y^i\|_2}
\]

Using part (1), \(\|M_i^T y^i\|_2 \geq \sqrt{n_1 q}/2\) with probability \(1 - O(n_1^{-1/4})\). Therefore it suffices to show that for every \(j = 1, \ldots, n_1\),
\[
|(M_i^T y^i)_j| \leq \frac{n_1^{-1/3} \sqrt{n_1 q}}{2} = \sqrt{dn_1^{1/6} n_2^{-1/4}}.
\]

To this end we will show that for any \(j\),
\[
\Pr \left[ |(M_i^T y^i)_j| > \frac{\sqrt{dn_1^{1/6} n_2^{-1/4}}}{2} \right] \leq \frac{1}{n_1^2} \tag{2}
\]
Again partition the coordinates of \( y^i \), with \( \Delta \) being the set of \( j \) so that \( |y^i_j| \leq \sqrt{\frac{2n_1}{n_2}} \) and \( \overline{\Delta} \) the rest. The contribution to \( |(M_i^T y^i)_j| \) from \( \Delta \) is bounded by

\[
(n_2q + m_j) \sqrt{\frac{2qn_1}{n_2}}
\]

where \( m_j \) is the number of \( 1 - q \) entries in the \( j \)th row of \( M_i^T \). This number \( m_j \) is dominated by a Binom\((n_2, 2q)\) random variable and so with probability \( 1 - \exp(-n_2q) \), \( m_j \leq 3n_2q \). Therefore, the contribution from \( \Delta \) is bounded by

\[
(3n_2q \cdot q + m_j \cdot 1) n_2^{-1/4} n_1^{-1/12}
\]

where we have used (4) and (1), and \( m_j \) is the number of \( 1 - q \) entries in the \( j \)th row of \( M_i^T \) whose column has index in \( \overline{\Delta} \). \( m_j \) is dominated by a Binom\((3n_2n_1q, q)\) random variable, and so with probability \( 1 - O(\exp(-\Omega(3n_2n_113/12q^2))) \), \( m_j \leq 3n_2n_1q^2 \cdot n_1^{-1/12} \) in which case we have that the contribution from \( \overline{\Delta} \) is bounded by

\[
3n_2n_1q^2 n_2^{-1/4} n_1^{-1/12} + 3n_2n_1q^2 n_2^{-1/4}
= 3d^2n_2^{-1/4} n_1^{-1/12} + 3d^2n_2^{-1/4}
\leq \frac{\sqrt{dn_1}^{1/6} n_2^{-1/4}}{4}
\]

proving inequality (2). (We remark that for this part, the loose bounds we have above suffice, as it is the next part that controls parameter settings).

To prove (4), set \( \lambda = n_2^{-1/4} n_1^{-1/12} \).

\[
\|y^{i+1}\|_\infty = \max_{j \in [n_2]} |(M_i^T x^i)_j|
\]

Using part (2), \( \|M_i^T x^i\|_2 \geq \sqrt{n_2q}/2 \) with probability \( 1 - O(n_1^{-1/6}) \). Therefore it suffices to show that for every \( j = 1, \ldots n_2 \),

\[
|(M_i^T x^i)_j| \leq \frac{\lambda \sqrt{n_2q}}{2}
= \frac{\lambda \sqrt{dn_1^{1/4}}}{2n_1^{1/4}}
\]

We will show that for any \( j \),

\[
\Pr \left[ |(M_i^T x^i)_j| > \frac{\lambda \sqrt{dn_1^{1/4}}}{2n_1^{1/4}} \right] \leq \frac{1}{n_2^2}
\]
We partition the coordinates of $x^i$ according to their magnitude, in bins $B_1, \ldots B_L$, defined for $l < L$ as

$$B_l = \left\{ i : |x_i| \in \left( \frac{n_1^{-1/3}}{2^l}, \frac{n_1^{-1/3}}{2^{l-1}} \right) \right\}$$

with the interval for $B_L$ being $[0, n_1^{-1/3}/2^{L-1}]$. We set $L = \lceil \log(n_1^{1/6}) \rceil$. Let

$$t_l = |B_l| \leq 2^{2l} n_1^{2/3}$$

using the fact that $x^i$ has unit 2-norm.

We will bound the probability that bin $l$ contributes more than $\beta_l$ towards the value of $|\langle M_i^T x^i \rangle_j|$, with

$$\beta_l = \frac{\lambda \sqrt{dn_2}^{1/4}}{4n_1^{1/4} l^2}$$

If all bins fall within these bounds, then

$$|\langle M_i^T x^i \rangle_j| \leq \sum_l \beta_l \leq \frac{\lambda \sqrt{dn_2}^{1/4}}{2n_1^{1/4}}$$

and therefore $\|y^{i+1}\|_\infty \leq n_2^{-1/4} n_1^{-1/12}$.

Let $Z_l \sim \text{Bin}(t_l, q)$. The contribution of bin $l$ is bounded by the maximum of $\frac{n_1^{-1/3}}{2^{l-1}} Z_l$ and $|q \sum_r x^i_r| \leq q \sqrt{n_1} \leq \beta_l$. To bound the first term, let

$$m_l = \beta_l 2^{l-1} n_1^{1/3}$$

$$= \frac{\lambda 2^{l} \sqrt{dn_2}^{1/4} n_1^{1/12}}{8l^2}$$

$$= \frac{2^l \sqrt{d}}{8l^2}$$

and consider

$$\Pr[Z_l \geq m_l] \leq 2 \left( \frac{t_l}{m_l} \right) q^{m_l}$$

$$\leq 2 \left( \frac{et_l q}{m_l} \right)^{m_l}$$

$$\leq 2 \left( \frac{e 2^{l} n_1^{2/3} d(n_2 n_1)^{-1/2}}{\sqrt{d} 2^l / (8l^2)} \right)^{\sqrt{d} 2^l / (8l^2)}$$

$$= 2 \left( \frac{8e 2^l \sqrt{d} n_1^{1/2}}{n_2} \right)^{\sqrt{d} 2^l / (8l^2)}$$

$$\leq 2 \left( \frac{8e \sqrt{d} \log^2(n_1) n_1^{1/3}}{n_2} \right)^{\sqrt{d} 2^l / (8l^2)}$$

$$\leq 2 \left( \frac{8e \sqrt{d} \log^2(n_1) n_1^{1/3}}{n_2} \right)^{\sqrt{d}/4}$$

$$\leq \frac{(8e \sqrt{d} \log^2(n_1))^7}{n_2^{7/3}}$$

for $\sqrt{d}/4 \geq 7$. 

19
Taking a union bound over all $L$ bins, we have \(\Box\).

Next we show that the vector $y^T$ reaches high correlation with $v$ after $T/2$ steps. Recall the definitions $V_i := v \cdot y^i$ and $U_i := u \cdot x^i$.

**Proposition 5.** With probability $1 - O((\ln \ln n_1)^{-2})$, one of the following happens:

1. For all $i \in \{T/2, \ldots T\}$,
   \[ V_i \geq \frac{(\delta - 1)\sqrt{n_2 q}}{4} \]

2. For all $l \in \{T/2, \ldots T\}$,
   \[ V_i \leq -\frac{(\delta - 1)\sqrt{n_2 q}}{4} \]

First we need the following bounds on the progress of $U_i$:

**Proposition 6.** The following bounds on $U_i$ hold:

1. With probability at least $1/2$, $|U_i| \geq 1/4$ regardless of the value of $V_i$.
2. If $1/4 \leq |U_i| \leq \ln \ln n_1$, then with probability at least $1/2$, $|U_{i+1}| \geq 2|U_i|$.
3. $\Pr(|U_{i+1}| \geq 2|U_i|) \geq 1 - \frac{1}{U_i q}$ for $\ln \ln n_1 \leq |U_i| \leq \sqrt{n_1}/4$.
4. If $U_i \geq \sqrt{n_1}/4$, then $\Pr(U_{i+1} \geq \sqrt{n_1}/2) \geq 1 - O(1/\sqrt{n_1 n_2})$. Similarly, if $U_i \leq -\sqrt{n_1}/4$, then $\Pr(U_{i+1} \leq -\sqrt{n_1}/2) \geq 1 - O(1/\sqrt{n_1 n_2})$.
5. If $U_i \geq \sqrt{n_1}/2$, then $V_{i+1} \geq \frac{(\delta - 1)\sqrt{n_2 q}}{4}$ with probability $1 - O(1/\sqrt{n_1 n_2})$.

1) and 2) ensure that Phase 1 succeeds, and that $U_i$ attains value $\ln \ln n_1$ within $\ln n_1$ steps. 3) and 4) ensure steady progress in Phase 2 and that once $U_i$ attains a high value, it maintains it. 5) connects the two potential functions by showing that $V_{i+1}$ is large if $U_i$ is large.

**Proof of Proposition 6.** 1. The variance of $u \cdot (M_i y^i)$ is $\sim n_1 q$, and so a Berry-Esseen bound gives that with probability at least $1/2$, $|u \cdot M_i y^i| \geq \sqrt{n_1 q}/4$. Then using Lemma 2 we have that $||M_i y^i||_2 = \sqrt{n_1 q}(1 + o(1))$ whp, and so with probability at least $1/2$, $|U_i| = |u \cdot x^i| \geq 1/4$.

2. We prove this in two steps. The expectation of $v \cdot (M^T_i x^i)$ is $(\delta - 1)n_2 q(u \cdot x^i)$, with variance $n_2 q$. Both are $\omega(1)$, and the expectation is at least $(\delta - 1)/4$ times the variance in absolute value, and so whp, $v \cdot (M^T_i x^i) = (\delta - 1)n_2 q(u \cdot x^i)(1 + o(1))$. Using Lemma 2 again, we have that whp, $V_{i+1} = (\delta - 1)\sqrt{n_2 q}(u \cdot x^i)$.

Conditioning on this value, we have
\[
\mathbb{E}[u \cdot (M_{i+1} y^{i+1})] = (\delta - 1)^2 \sqrt{n_2 q} n_1 q(u \cdot x^i)(1 + o(1))
\]
and its variance is $n_1 q$. With probability $1/2$ we have $|u \cdot (M_{i+1} y^{i+1})| \geq (\delta - 1)^2 \sqrt{n_2 q} n_1 q(u \cdot x^i)(1 - o(1))$, and then normalizing with Lemma 2 we have $|U_{i+1}| \geq (\delta - 1)\sqrt{n_2 q} |U_i|$, which from our choice of $q$, is at least $2|U_i|$. 3. Similar to the above. Apply Chebyshev so that $v \cdot (M^T_i x^i) = (\delta - 1)n_2 q(u \cdot x^i)(1 + o(1))$ with probability $1 - o(1)$, and normalize so that $v \cdot y^{i+1} = (\delta - 1)\sqrt{n_2 q}(u \cdot x^i)(1 + o(1))$ whp. Now the
expectation of \( u \cdot (M_{i+1}y^{i+1}) \) is \((\delta - 1)^2 \sqrt{n_2 q n_1 q (u \cdot x^i)(1 + o(1))}\) with variance \( n_1 q \), and so applying Chebyshev, we have

\[
\Pr[|u \cdot (M_{i+1}y^{i+1})| < (\delta - 1)^2 \sqrt{n_2 q n_1 q |u \cdot x^i|}/2] \leq \frac{n_1 q}{(\delta - 1)^4 n_2 n_1^2 q^3 (u \cdot x^i)^2 / 4} \\
= \frac{1}{(\delta - 1)^4 n_2 n_1^2 q^3 (u \cdot x^i)^2} \\
\leq \frac{1}{25(u \cdot x^i)^2}
\]

Then normalizing, and using Prop. 4 and part (2) above, we get

\[
|U_{i+1}| \geq \frac{(\delta - 1)^2 \sqrt{n_2 q n_1 q |U_i|}}{2 \sqrt{n_1 q + (\delta - 1)^2 n_2 n_1^2 (V_{i+1})^2}} \\
\geq \frac{(\delta - 1)^2 \sqrt{n_2 q n_1 q |U_i|}}{2 \sqrt{n_1 q + (\delta - 1)^4 n_2 n_1^2 q^3 (U_i)^2}} \\
\geq 2|U_i|.
\]

4.5. Chebyshev again.

\[\square\]

**Proof of Proposition 3.** In the first phase, we show that it takes \( \ln \ln n_2 \) iterations for \( |U_i| \) to reach \( \ln \ln n_1 \) whp. Next, it takes a further \( \ln n_1 \) iterations to reach \( \sqrt{n_1}/2 \). Finally, \( |U_i| \) will remain above \( \sqrt{n_1}/2 \) whp for an additional \( 2 \ln n_1 \) iterations.

Step 1: We call a step from \( U_i \) to \( U_{i+1} \) ‘good’ if \( |U_{i+1}| \geq 2|U_i| \), or if \( |U_{i+1}| \geq 1/4 \) following a bad step. A run of \( \ln \ln n_1 \) good steps must end with \( |U_i| \geq \ln \ln n_1 \). As long as \( |U_i| < \ln \ln n_1 \), the proposition above shows that the probability of a good step is at least 1/2, so in \( \ln \ln n_1 \) steps, with probability \( 1 - o(1) \) we will either have such a run or reach \( \ln \ln n_1 \) even earlier.

Step 2: Once we have \( |U_i| \geq \ln \ln n_1 \), the value will double whp in successive steps until \( |U_i| \geq \sqrt{n_1}/4 \). This takes at most \( \ln n_1 \) steps. The total error probability, by part 3) of Proposition 6 is a geometric series that sums to \( O(1/|\ln \ln n_1|^2) \).

Step 3: Once \( |U_i| \geq \sqrt{n_1}/4 \) then for the next \( 2 \ln n_1 \) steps, \( U_{i+1}, U_{i+2}, \ldots \), we have \( |U_i| \geq \sqrt{n_1}/2 \), with total error probability \( O(T/\sqrt{n_1 n_2}) \).

Step 4: Finally we use part 5) of Proposition 6 to conclude that \( y' \) has high correlation with \( v \).

\[\square\]

We now use Proposition 5 to prove the main theorem.

**Proof of Theorem 7.** Now that we know whp \( y^{T/2}, y^{T/2+1}, \ldots \) all have large correlation with \( v \), we show that taking the majority vote for each coordinate of \( z^{T/2+1}, z^{T/2+2}, \ldots \) recovers \( \pm u \) whp.

Take the first case from Proposition 5 with \( V_i \geq \frac{(\delta - 1) \sqrt{n_2 q n_1}}{4} \). Assume \( j \in A_1 \), then we have, conditioned on the value of \( V_i \)
Pr[z^{i+1}_{j} = 1] = Pr[x_{j} > 0]
\geq 1 - \frac{\text{var}((My)^{i}_{j})}{(\mathbb{E}((My)^{i}_{j}))^2}
\geq 1 - \frac{32q}{(\delta - 1)^2q^3n_1n_2}
= 1 - \frac{32}{100^2} \geq .9

Now an application of Azuma’s inequality shows that with probability at least $1 - o(n_1^{-2})$, $\sum_{i=T/2}^{T} z^{i}_{j} > 0$. Similarly, for $j \in B_1$, we have $\sum_{i=T/2}^{T} z^{i}_{j} > 0$ with probability at least $1 - o(n_1^{-2})$, and so whp the majority vote recovers $u$ exactly. The same argument shows that if the second case of Proposition 5 holds, then $-u$ is recovered whp.