JMLR: Workshop and Conference Proceedings vol 35:1–23, 2014

Computational Limits for Matrix Completion

Moritz Hardt MHARDT@US.IBM.COM
IBM Research Almaden

Raghu Meka MEKA@MICROSOFT.COM
Microsoft Research

Prasad Raghavendra PRASAD@CS.BERKELEY.EDU
University of California, Berkeley.

Benjamin Weitz BSWEITZ@EECS.BERKELEY.EDU
University of California, Berkeley.

Abstract

Matrix Completion is the problem of recovering an unknown real-valued low-rank matrix from a subsample of its entries. Important recent results show that the problem can be solved efficiently under the assumption that the unknown matrix is incoherent and the subsample is drawn uniformly at random. Are these assumptions necessary?

It is well known that Matrix Completion in its full generality is NP-hard. However, little is known if we make additional assumptions such as incoherence and permit the algorithm to output a matrix of slightly higher rank. In this paper we prove that Matrix Completion remains computationally intractable even if the unknown matrix has rank 4 but we are allowed to output any constant rank matrix, and even if additionally we assume that the unknown matrix is incoherent and are shown 90% of the entries. This result relies on the conjectured hardness of the 4-Coloring problem. We also consider the positive semidefinite Matrix Completion problem. Here we show a similar hardness result under the standard assumption that $\mathrm{P} \neq \mathrm{NP}$.

Our results greatly narrow the gap between existing feasibility results and computational lower bounds. In particular, we believe that our results give the first complexity-theoretic justification for why distributional assumptions are needed beyond the incoherence assumption in order to obtain positive results. On the technical side, we contribute several new ideas on how to encode hard combinatorial problems in low-rank optimization problems. We hope that these techniques will be helpful in further understanding the computational limits of Matrix Completion and related problems.

Keywords: Matrix Completion, Computational Hardness, Coloring

1. Introduction

Suppose we observe a subset of the entries of an unknown low-rank matrix M, can we recover the matrix M knowing this subset alone? This problem, called Matrix Completion, is of fundamental interest in a number of fields including statistics, machine learning, signal processing and theoretical computer science. It is widely applicable to the design of recommender systems as popularized by the famous Netflix Prize. We are interested in understanding the compuational complexity of Matrix Completion.

Much of the theory of Matrix Completion revolves around a beautiful line of positive results. These results show that under certain assumptions there is a natural semidefinite relaxation that solves the problem efficiently even if the number of visible entries is asymptotically much smaller than the total number of entries Candès and Recht (2009); Candès and Tao (2010); Recht (2011). Specifically, these feasibility assumptions state that:

Low rank. M has rank k where k is typically constant or very slowly growing.

Incoherence. The row and columns spaces of M are *incoherent*. Informally, a subspace U of $\mathbb{R}^n$ is incoherent if for every standard basis vector $e_i \in \mathbb{R}^n$, the Euclidean norm of the projected vector $P_U e_i$ is much smaller than 1. Here, P_U denotes the orthogonal projection onto the space U.

Randomness. Finally, the subset of entries is drawn uniformly at random from M with a certain sufficient sampling density p.

Among these assumptions the last one is particularly taxing. In most applications, the algorithm designer cannot choose the subset of revealed entries. Instead nature determines the subsample, e.g., available user/movie ratings on Netflix. Often it is argued that without the randomness assumption, the solution to the problem may no longer be uniquely determined. But rather than insisting on uniqueness of the solution, it is natural to only require consistency with the given subset. That is, we only require the solution to agree with the observed entries. There could be multiple valid solutions. Moreover, algorithmically two additional relaxations are natural. First, we can attempt to make the problem easier by allowing some slack in terms of the rank $r > k$ of the solution. Second, we can allow an approximation error on the observed entries. That is, rather than matching the observable entries exactly we allow the algorithm to find a solution that is close in Frobenius norm.

Surprisingly, even with these relaxations the status of some deceptively simple algorithmic questions remained wide open. For example:

Question 1 *Given entries of an incoherent rank* 4 *matrix, can we find a rank* 100 *matrix that is approximately consistent with this set of entries in polynomial time?*

Even though the problem might appear to be very simple, neither upper bounds nor lower bounds are known. Matrix Completion in its full generality is of course NP-hard, but no hardness result is known for the problem we just described. In fact, for small k, the main prior hardness result we are aware of is due to Peeters Peeters (1996) who showed that given a subset of a rank 3 matrix it is NP-hard to find an exactly consistent matrix of *equal* rank.[1] However, Peeters' hardness result does not apply to the various relaxations of interest.

The lack of applicable hardness results for Matrix Completion is partially due to the nature of the problem. Low-rank decompositions over the reals do not seem to exhibit the same combinatorial rigidity common to most NP-hard optimization problems. This conundrum arises in a number of interesting machine learning problems such as Sparse PCA and Robust PCA. Indeed, only recently did Berthet and Rigollet give evidence for computational hardness of the Sparse PCA problem by reducing to the Planted Clique problem in a natural setting Berthet and Rigollet (2013). For Robust PCA, the hardness result of Hardt and Moitra Hardt and Moitra (2013) appeals to the conjectured hardness of Small Set Expansion.

1. Several results are known over finite fields, but Matrix Completion over the reals is of particular interest in applications.

Our goal is to make progress on understanding the computational complexity of Matrix Completion in the natural relaxed setting that we described above. We show that under a plausible hardness assumption, there is in fact no polynomial time algorithm that solves the task. An immediate corollary is that even if we adopt the first two feasibility assumptions, some distributional assumption on the revealed entries is necessary in order to make Matrix Completion tractable.

We also consider a natural variant of Matrix Completion where the unknown matrix is positive semidefinite and so must be the output matrix. The positive semidefinite completion problem arises naturally in the context of Support Vector Machines (SVM). The kernel matrix used in SVM learning must be positive semidefinite as it is the Gram matrix of feature vectors. But oftentimes the kernel matrix is derived from partial similarity information resulting in incomplete kernel matrices. In fact, this is a typical situation in medical and biological application domains Tsuda et al. (2003). In such cases the data analyst would like to complete the partial kernel matrix to a full kernel matrix while ensuring positive semidefiniteness. Moreover, since it is often infeasible to store a dense $n \times n$ matrix, it is desirable to also have a low-rank representation of the kernel matrix Fine and Scheinberg (2002). This is precisely the low-rank positive semidefinite completion problem. Our results establish strong hardness results for this problem under natural relexations. In this case we show that for any $k \geqslant 2$ it is NP-hard to complete a partially given rank k matrix by a rank $(2k-1)$ matrix.

1.1. Our Results

We will restrict our attention to symmetric $n \times n$ matrices throughout. As we are proving hardness results, this only makes the results stronger. We begin with a formal definition of the Matrix Completion problem. Here, we restrict our attention to the case where both input and output have bounded coefficients as is the case in most application settings.

Definition 2 *We define the* $(k, r, p, \varepsilon, c)$-COMPLETION *problem as follows:*

Input: *A matrix* $A \in (\mathbb{R} \cup \{\perp\})^{n \times n}$ *and a set* $\Omega \subseteq [n] \times [n]$ *of size* $|\Omega| \geqslant pn^2$ *such that there exists a rank* k *matrix* M *with bounded entries* $|M(i,j)| \leqslant c$ *for all* i *and* j*, such that for all* $(i,j) \in \Omega$ *we have* $A(i,j) = M(i,j)$ *and for all* $(i,j) \notin \Omega$ *we have* $A(i,j) = \perp$.

Output: *A rank* r *matrix* B *with bounded coefficients* $|B(i,j)| \leqslant c$ *for all* i *and* j*, such that* B *approximates* A *with small root-mean-squared error (RMSE):* $\sum_{(i,j)\in\Omega} |A(i,j) - B(i,j)|^2 \leqslant \varepsilon n$.

We will use (k, r, p, c)-COMPLETION *as a shorthand for* $(k, r, p, 0, c)$-COMPLETION*, i.e. exact completion. We also use* (k, r, c)-COMPLETION *as a shorthand for* $(k, r, 0, 0, c)$-COMPLETION.

We require that the output matrix be close in RMSE rather than a different error measure because the hard instances of matrix completion that we construct only have a linear number of nonzero revealed entries, so $\sum_{ij\in\Omega} |A(i,j)|^2 = O(n)$. If we were allowed more than $O(n)$ error, then we could simply output the all-zeros matrix and trivially solve the problem.

To state our first result we introduce the problem of coloring a k-colorable graph with r colors.

Definition 3 *We define the* (k, r)-COLORING *problem as follows:*

Input: *A* k*-colorable graph* G.

Output: *An r-coloring of the graph G.*

Our second theorem will appeal to a closely related variant of the problem in which the output is an independent set of size n/r rather than an r-coloring.

Definition 4 *We define the (k, r)-INDEPEDENT-SET problem as follows:*

Input: *A k colorable graph G.*

Output: *An independent set of size n/r in the graph G.*

Notice that if there exists an r-coloring of the graph then one of the color classes will be an independent set of size n/r. Thus, (k, r)-INDEPEDENT-SET reduces to (k, r)-COLORING. Despite extensive work on algorithms for k-COLORING Wigderson (1982); Berger and Rompel (1990); Blum and Karger (1997); Karger et al. (1998); Arora et al. (2006), the problem has remained notoriously hard. Given a 3-colorable graph, the best algorithms Chlamtac (2007); ichi Kawarabayashi and Thorup (2014) known can only find an independent set of size at most $n^{1-\Omega(1)}$. In particular, (k, r)-COLORING and (k, r)-INDEPEDENT-SET with $k = 3$ and $r = O(1)$ remains hopelessly out of reach of existing algorithmic techniques. From a complexity standpoint it is believed that the (k, r)-INDEPEDENT-SET problem (and hence (k, r)-COLORING) cannot be solved in polynomial time for even $k = 3$ and $r = O(1)$. This is further supported by the work of Dinur and Shinkar (2010) who show this to be the case under a variant of the Unique Games Conjecture (called 2-to-1 Label Cover) which by now underlies a number of hardness results in complexity theory.

We will show that assuming (k, r)-COLORING and (k, r)-INDEPEDENT-SET are hard for $k = 3, r = O(1)$, the Matrix Completion problem is hard in a range of natural parameters even on incoherent matrices and even if most entries are revealed. To make the theorems precise we state the assumption concretely and give a formal definition of incoherence.

Conjecture 5 *The (k, r)-COLORING problem is not in P for any $r \geqslant k \geqslant 3$ and $r = O(1)$.*

Conjecture 6 *The (k, r)-INDEPEDENT-SET problem is not in P for any $r \geqslant k \geqslant 3$ and $r = O(1)$.*

The coherence of a matrix is defined as follows.

Definition 7 *A symmetric $n \times n$ matrix M of rank k has coherence μ if there exists a singular value decomposition $M = U\Sigma V^\top$ such that for every standard basis vectors $e_i \in \mathbb{R}^n$ we have that $\|e_i^\top U\|_2 \leqslant \sqrt{k\mu/n}$ and $\|e_i^\top V\|_2 \leqslant \sqrt{k\mu/n}$.*

Note that Conjecture 5 is weaker than Conjecture 6. With the above definitions we have the following results.

Theorem 8 *Assume Conjecture 5. Then, for any constants $c \geqslant 1$, $k \geqslant 3$ and $r > k$, there is no polynomial time algorithm that solves the $(k, r, 0.9, c)$-COMPLETION problem on matrices of coherence $\mu \leqslant O(1)$. Further, for all $1/2 > \varepsilon > 0$, the same conclusion holds even if we are only required to compute a rank r matrix which approximates each entry with additive error at most ε.*

In most practical scenarios it suffices to look for a low-rank completion with small RMSE error. Our next result addresses this situation.

Theorem 9 *Assume Conjecture 6. Then, for any constants $k \geqslant 4$, $r > k$, and $0 < \varepsilon < 1/(2cr)^2$, there is no polynomial time algorithm that solves the $(k, r, 0.9, \varepsilon, c)$-*COMPLETION *problem on matrices of coherence $\mu \leqslant O(1)$.*

This result should be contrasted with positive results showing that $(k, k, O(k\mu(\log^2 n))/n)$-COMPLETION is easy so long as the entries are revealed randomly Recht (2011).

Positive Semidefinite Completions. We define the (k, r, p)-PSD-COMPLETION problem the same way we defined (k, r, p)-COMPLETION except that we drop the bound on the coefficients and additionally require that both M and B must be positive semidefinite. Our result here is incomparable to the previous one and it relies on the standard NP-hardness assumption.

Theorem 10 *Assume that* $\mathrm{P} \neq \mathrm{NP}$. *Then for every even $k \geqslant 2$ there is no polynomial time algorithm that solves the $(k, 2k - 1, 0.9)$-*PSD-COMPLETION *problem.*

This theorem strengthens a recent result by E.-Nagy et al. E.-Nagy et al. (2013) who showed that $(k, k, 0)$-PSD-COMPLETION is NP-hard for every $k \geqslant 2$.

We also prove a version of Theorem 10 for approximate completion:

Theorem 11 *Assume that* $\mathrm{P} \neq \mathrm{NP}$. *Then for every even $k \geqslant 6$ and $\varepsilon < O(k^{-5})$, there is no polynomial time algorithm that solves the $(k, 2k - 1, 0.9)$-*PSD-COMPLETION *problem even if the output matrix only approximates each entry with additive error at most ε.*

Acknowledgments

We are very grateful to Phil Long for insightful early contributions to this work. In fact, he first conjectured that (k, r)-COMPLETION should be as hard as (k, r)-COLORING as established by Theorem 8. The authors also thank the reviewers for their helpful and insightful comments, as well as the Simons Institute for the Theory of Computing at Berkeley for its hospitality.

1.2. Further Related Work

There have been several hardness results for Matrix Completion over finite fields drawing on its connection to problems in coding theory. See, for example, the discussion in Harvey et al. (2006); Tan et al. (2012). The Matrix Completion problem over the reals seems to behave rather differently and techniques do not seem to transfer from the finite field case.

PSD completions are also natural objects in discrete optimization and the study of the geometry of graphs. We refer the reader to the recent work of E.-Nagy, Laurent and Varvitsiotis E.-Nagy et al. (2013) for a more extensive discussion of related work in this area.

1.3. Proof Overview

We now give a highlevel outline of our proofs.

1.3.1. MATRIX COMPLETION

While the hardness assumption in our reduction (Conjecture 5) is similar in spirit to that of Peeters' original reduction, our proof works in a very different manner.

Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$. Now define the $n \times n$ partial matrix P_G such that $P_G(i, i) = 1$ for every $i \in [n]$, and $P_G(i, j) = 0$ if $(i, j) \in E$. The intuition behind this reduction is that, if G is k-colorable with coloring function $f : V \to [k]$, then

$$M_f = \sum_{i \in [k]} 1_{f^{-1}(i)} 1_{f^{-1}(i)}^T$$

is a rank-k completion of P_G. Peeters Peeters (1996) showed how to gadgetize a graph G so that these were the *only* rank-k completions of P_G. However, the gadgets in that work are unable to force any structure on completions of rank higher than k. To decode colorings from high rank completions, we will consider a special factorization of the completion. The row vectors of this factorization will have bounded norm, which will allow us to cover them with a constant number of small balls in $\mathbb{R}^r$. If the balls are small enough, then any two vectors that lie in the same ball cannot have zero dot product, so their corresponding vertices cannot have an edge in G. We then use the balls to color the vertices.

The above argument works when we look at exact completions (or those with entry-wise error bounds). To obtain our main result, Theorem 9, we focus on more general structure of any low-rank completion, in this case the existence of large non-zero rectangles. We will prove, under some mild assumptions on any approximate (in RMSE) completion M of P_G, that M has a large non-zero square, which corresponds to an independent set in the graph G.

1.3.2. POSITIVE SEMIDEFINITE MATRIX COMPLETION

We give two reductions for the (k, r)-PSD-COMPLETION problem: one from the PARTITION problem and one from a constraint satisfaction problem EXACT-ONE-IN-K-SAT. The first reduction has the advantage of being simple but only works for *exact* completion. Our second reduction is more involved but is much more robust and it works even when we allow for errors and gives us the theorem on approximate completions from the introduction.

Consider an instance of the (k, r)-PSD-COMPLETION problem with input $A \in (\mathbb{R} \cup \{\bot\})^{n \times n}$. Our goal is to find a PSD matrix B which agrees with A on the set of non-$\bot$ entries. Now, recall that a characterization of PSD matrices is that a $n \times n$ matrix B is PSD if and only if it can be factored as $B = UU^\top$ for some matrix U. If we let $u_1, \ldots, u_n$ be the rows of the matrix U, then we have $B_{ij} = \langle u_i, u_j \rangle$. The vectors $u_1, \ldots, u_n$ are called the *Gram vectors* of B. In the context of PSD-COMPLETION, the revealed entries of A place equality constraints on the inner products of the Gram vectors:

$$\langle u_i, u_j \rangle = A_{ij}, \quad \text{if } A_{ij} \neq \bot.$$

Moreover, these constraints completely characterize the problem and finding a rank r solution for the completion problem is equivalent to finding a set of vectors $u_1, \ldots, u_n \in \mathbb{R}^r$ satisfying the above constraints. We will adopt this perspective in our reductions and view the partial matrix as a list of such inner-product constraints.

We design constraints to simulate ± 1 variables which we can then use as *gadgets* to reduce from many different problems. For the PARTITION problem, we follow an idea proposed in E.-Nagy et al.

(2013) and associate every item in the partition with a two-dimensional basis, and constrain that the $(i+1)$th basis is a θ-rotation of the ith basis (including the first and nth bases), where θ depends on the element a_i in the PARTITION problem. This creates a cyclic dependence on the rotations of the bases that forces the total sum of the rotations to be an integer multiple of 2π. However, the important things to note are that these rotations can be in one of two direction: clockwise or counter-clockwise, and if the angles are small enough then the sum of the rotations must be zero. Thus we find a partition based on which rotations went clockwise and which went counter-clockwise. By constraining sums of basis vectors in addition to the basis vectors themselves, we can force the same rotational structure in three dimensions as in two, yielding the gap.

For the EXACT-ONE-IN-K-SAT problem, we similarly associate every variable with a basis and use the inner product constraints to force these bases to be special rotations of a reference gadget. We interpret the variable as being $+1$ or -1 depending on if the rotation is a "clockwise" or "counter-clockwise" rotation. Because the relations of EXACT-ONE-IN-K-SAT are linear, i.e. the sum of the values of the variables in each clause is exactly $(k-2)$, it is easy to force satisfying assignments. See Section 3.2 for a more thorough description and the appendix for the full details.

2. Hardness for Matrix Completion

In this section we show that the matrix completion problem is hard even with relaxed rank constraints and allowing for approximate completions. In particular, we will give a reduction to prove Theorem 9. We defer the proof of Theorem 8 to the appendix.

Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$. Now define the partial matrix $P_G \in (\mathbb{R} \cup \perp)^{n \times n}$ as follows:

$$P_G(i,j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } (i,j) \in E \\ \perp & \text{otherwise} \end{cases}.$$

As described in the introduction, if G is k-colorable with coloring function $f : V \to [k]$, then

$$M_f = \sum_{i \in [k]} 1_{f^{-1}(i)} 1_{f^{-1}(i)}^T$$

is a rank-k completion of P_G. Note that M_f has coherence $\mu = \frac{n}{k}(\min_i |f^{-1}(i)|)^{-1}$. However, we may assume that there is a perfectly balanced coloring of G, for example by copying the graph k times. Thus we can take M_f to have coherence exactly $\mu = 1$. We next prove that under some mild additional assumptions any approximate low-rank completion M of P_G yields a large independent set of G.

Our technique relies on a lemma in Linial et al. (2007) that guarantees the existence of a good factorization of low rank matrices, in the sense that the norms of the row and column vectors of the factorization are small. We state the lemma here for completeness:

Lemma 12 *Let M be a matrix with rank r. Then there exists an r-dimensional factorization $M = XY^T$ such that every row vector of X and Y has norm at most $(cr)^{1/4}$, where $c = \max_{ij} |M(i,j)|$.*

The proof of this statement in its original paper was given in a nonconstructive manner using John's theorem from convex analysis, but such a good factorization can be found in polynomial time using semidefinite programming. See the appendix for the details.

Lemma 13 *Let $G = (V, E)$ be a graph and define P_G as above with $\Omega \subseteq [n] \times [n]$ the set of revealed entries. Let M be a rank r matrix such that*

$$\sum_{(i,j)\in\Omega} (M(i,j) - P_G(i,j))^2 \leqslant \varepsilon n$$

and $|M(i,j)| \leqslant c$. Then G has an independent set T of size at least

$$|T| \geqslant \frac{(1 - 4(cr)^2\varepsilon)n}{r\sqrt{\pi}(8\sqrt{cr})^r}.$$

Moreover, there is a randomized polynomial time algorithm for finding such an independent set given M.

Proof By a simple averaging argument, there can be only $\frac{\varepsilon}{\delta^2}n$ entries of M that are different from P_G by more than δ. Thus there are at least $(1 - \varepsilon/\delta^2)n$ rows and columns such that $|M(i,j) - P_G(i,j)| \leqslant \delta$ for any row i and column j. Let M' be this submatrix of M. Certainly $\text{rank}(M') \leqslant \text{rank}(M) = r$, so Lemma 12 tells us we can find a factorization $XY^T = M$ with row vectors u_i and v_i such that $\|u_i\|, \|v_j\| \leqslant (cr)^{1/4}$ for all $i, j \in [n]$. Let $\theta(u_i, v_j)$ denote the angle between vectors u_i and v_j. Since $u_i \cdot v_i \geqslant 1 - \delta$ for all $i \in [n]$, we derive

$$\cos(\theta(u_i, v_i)) \geqslant \frac{(1-\delta)}{\sqrt{cr}} \text{ and } \|u_i\|, \|v_i\| \geqslant (1-\delta)(cr)^{-1/4}.$$

Now in order to find an independent set, we have to look for entries with $M(i,j) > \delta$ to be assured that indeed $(i,j) \notin E$. From the bound on the norms of u_i and v_j, if $\cos\theta(u_i, v_j) > \delta\frac{\sqrt{cr}}{(1-\delta)^2}$, then $M(i,j) > \delta$. In order to capture these points, we will pick the points in a random cone. Let ϕ denote the angle such that $\cos\phi = \delta\frac{\sqrt{cr}}{(1-\delta)^2}$. Our random procedure to find T is

- Normalize $\tilde{u}_i = u_i/\|u_i\|$ and $\tilde{v}_i = v_i/\|v_i\|$.
- Pick a random unit vector $x \in \mathbb{R}^r$.
- For every $i \in S$, if $\tilde{u}_i \cdot x > \cos(\phi/2)$ and $\tilde{v}_i \cdot x > \cos(\phi/2)$ then put $i \in T$.

For $i, j \in T$, since $\theta(\tilde{u}_i, x) < \phi/2$ and $\theta(\tilde{v}_i, x) < \phi/2$, by triangle inequality, $\theta(\tilde{u}_i, \tilde{v}_i) < \phi$. As noted above, since $\cos\theta(\tilde{u}_i, \tilde{v}_i) > \cos\phi$, we get $u_i \cdot v_j > \delta$, thus $M(i,j) > \delta$ and so $P_G(i,j) > 0$. We bound $|T|$ by checking the probability that i is placed in T. For each i, let w_i be the angle bisector of $\tilde{u}_i$ and $\tilde{v}_i$ and define

$$A_i = \left\{ x : \|x\| = 1, \theta(x, w_i) < \frac{1}{2}(\phi - \theta(\tilde{u}_i, \tilde{v}_i)) \right\}.$$

We will show that if $x \in A_i$ was chosen as our random vector, then $i \in T$. This implies that the probability that $i \in T$ is at least $\text{area}(A_i)/\text{area}(S^{r-1})$. For A_i to have positive area, we need δ small enough that $\phi > \theta(\tilde{u}_i, \tilde{v}_i)$. To this end, pick $\delta = 1/2cr$. It is a standard argument that the area of A_i is bounded below by the $(r-1)$-volume of a sphere with radius

$$b_i = \sin\left(\frac{\phi - \theta(\tilde{u}_i, \tilde{v}_i)}{2}\right)$$

Now noting that $\cos\phi = \delta\sqrt{cr}/(1-\delta)^2$ and $\cos\theta_i \geqslant (1-\delta)/\sqrt{cr}$ and using a Taylor Series approximation

$$\sin\frac{1}{2}\left(\cos^{-1}\left(\frac{x^2/2}{x(1-x^2/2)^2}\right) - \cos^{-1}\left(x(1-x^2/2)\right)\right) \geqslant \frac{x}{4} - O(x^3) \geqslant \frac{x}{8}$$

where $x = 1/\sqrt{cr}$ and the last inequality follows as long as $\sqrt{cr} \geqslant 1$. Now

$$\frac{\text{area}(A_i)}{\text{area}(S^{r-1})} \geqslant \frac{b_i^{r-1}}{r\sqrt{\pi}},$$

and thus $i \in T$ with probability at least $1/r\sqrt{\pi}(8\sqrt{cr})^r$. Now using linearity of expectation,

$$E[|T|] \geqslant \frac{(1-\varepsilon/\delta^2)n}{2r\sqrt{\pi}(8\sqrt{cr})^r} \geqslant \frac{(1-4(cr)^2\varepsilon)n}{r\sqrt{\pi}(8\sqrt{cr})^r}.$$

■

The above reduction produces a partial matrix P_G that has $|V| + |E|$ revealed entries, which could be much less than $0.9n^2$ if the graph G is sparse. However, we can simply pad the matrix P_G with zeros, i.e. output the $10|V| \times 10|V|$ matrix

$$P'_G = \begin{bmatrix} P_G & 0 \\ 0 & 0 \end{bmatrix}.$$

Combined with Lemma 13, the above implies Theorem 9.

3. Hardness for Positive Semidefinite Matrix Completion

To prove hardness for the (k, r, p, ε)-COMPLETION problem we appealed to a conjectured coloring hardness. In this section we show that this assumption can be weakened to the usual NP-hardness if the matrices under consideration are positive semi-definite. In particular, in this section we prove Theorem 10. We first present the hardness for the exact completion problem with $\varepsilon = 0$ with a reduction from PARTITION. We then sketch the second reduction from EXACT-ONE-IN-K-SAT that is capable of handling errors on the constraints and proves Theorem 11. The full details on the second reduction are deferred to the appendix.

3.1. Exact Completion

Our reduction is similar to Theorem 3.3 in E.-Nagy et al. (2013), but with extra constraints to retain structure in higher rank completions. We will reduce from the partition problem, i.e. given numbers $a_1, \ldots, a_n$, find a set $I \subseteq [n]$ such that $\sum_{i\in I} a_i = \sum_{i\notin I} a_i$. We will reduce this problem to $(2,3)$-PSD-COMPLETION and amplify the gap. Recall that a partial PSD matrix is equivalent to a list of inner product constraints. Given an instance $(a_1, \ldots, a_n)$, we will output a set of constraints on $3n$ different vectors. These vectors will be indexed by $I = [n] \times [3]$. Assume without loss of generality that $\sum_i a_i = 1$. Now constrain

- $u_s \cdot u_s = 1$ for all $s \in I$
- $u_{(i,1)} \cdot u_{(i,2)} = 0$ for all $i \in [n]$.

- $u_{(i,3)} \cdot u_{(i,1)} = u_{(i,3)} \cdot u_{(i,2)} = \frac{1}{\sqrt{2}}$. Equivalently, $u_{(i,3)} = \frac{1}{\sqrt{2}}(u_{(i,1)} + u_{(i,2)})$.
- $u_{(i,1)} \cdot u_{(i+1,1)} = u_{(i,2)} \cdot u_{(i+1,2)} = u_{(i,3)} \cdot u_{(i+1,3)} = \cos a_i$ for all $i \in [n]$, where addition is performed modulo n.

The intuition here is that in a rank-2 decomposition of this matrix, for every i, $\{u_{(i,1)}, u_{(i,2)}\}$ is an orthonormal basis, and the $(i+1)$st basis is an angle a_i-rotation of the ith basis. This rotation can be in one of two directions, clockwise or counter-clockwise. However since the 1st basis is an a_n-rotation of the nth basis, after rotating by every a_i we must be back where we started. Since $\sum_i a_i = 1 < 2\pi$, this means that the total rotation must be zero, so we partition the a_i based on whether the corresponding rotation was clock-wise or counterclockwise. The additional constraints on the sums of basis vectors will force this structure even in a rank-3 decomposition.

Lemma 14 *There is a set of vectors $\{u_s\}_{s\in I}$ lying in $\mathbb{R}^3$ satisfying the above constraints if and only if there is a set of such vectors lying in $\mathbb{R}^2$.*

Proof Let $\{u_s\}_{s\in I}$ be a set of vectors lying in $\mathbb{R}^3$ satisfying the constraints. For each i, there is an orthogonal transformation Q_i that maps $Q_i(u_{(i,1)}) = u_{(i+1,1)}$, $Q_i(u_{(i,2)}) = u_{(i+1,2)}$. Writing Q_i in the basis $\{u_{(i,1)}, u_{(i,2)}, u_{(i,1)} \times u_{(i,2)}\}$, and accounting for the constraints of the $\{u_s\}_{s\in I}$, we have

$$Q_i = \begin{bmatrix} \cos a_i & x & \\ -x & \cos a_i & A \\ y & z & \end{bmatrix}$$

but Q_i has orthogonal columns, which implies that $yz = 0$, so either $y = 0$ or $z = 0$. But Q_i also has columns of norm 1, so $x = \pm \sin a_i$, which implies that both y and z are zero. Thus

$$Q_i = \begin{bmatrix} \cos a_i & \pm \sin a_i & 0 \\ \mp \sin a_i & \cos a_i & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This implies that $\{u_{(i,1)}, u_{(i,2)}\}$ and $\{u_{(i+1,1)}, u_{(i+1,2)}\}$ lie in the same plane. Repeating the argument we get that $\{u_{(i,1)}, u_{(i,2)}\}$ lie in the same plane for every i. ∎

Lemma 15 *There is a partition of $(a_1, \ldots, a_n)$ if and only if there is a set of vectors satisfying the constraints lying in $\mathbb{R}^2$.*

Proof First, assume there is a partition $(I, \bar{I})$ of $[n]$ and set

$$\theta_k = \sum_{i\in I, i<k} a_i - \sum_{i\notin I, i<k} a_i$$

and by convention $\theta_1 = 0$. Note that by the definition of θ and partitions, $\theta_n = \pm a_n$. Now let e_i denote the ith standard basis vector, and set $u_{(i,1)} = e_1 \cos\theta_i + e_2 \sin\theta_i$, $u_{(i,2)} = e_1 \cos\theta_i - e_2 \sin\theta_i$, and $u_{(i,3)} = \frac{1}{\sqrt{2}}(u_{(i,1)} + u_{(i,2)})$ for every $i \in [n]$. Then $u_{(i,1)} \cdot u_{(i+1,1)} = u_{(i,2)} \cdot u_{(i+1,2)} = \cos(\theta_i - \theta_{i+1}) = \cos a_i$ for every $i < n$. Finally, since $\theta_n = \pm a_n$ and $\theta_1 = 0$, $u_{(n,1)}$ and $u_{(n,2)}$ are at an angle a_n with $u_{(1,1)}$ and $u_{(1,2)}$ respectively, so $u_{(n,1)} \cdot u_{(1,1)} = u_{(n,2)} \cdot u_{(n,1)} = \cos a_n$.

Conversely, suppose there is a set of vectors lying in $\mathbb{R}^2$ satisfying the constraints. Since the vectors are unit vectors and $u_{(i,1)} \cdot u_{(i+1,1)} = \cos a_i$, we know $u_{(i+1,1)}$ makes an angle a_i with $u_{(i,1)}$, so for every $i \in [n]$,

$$u_{(i,1)} = \cos\left(\sum_{j=1}^{i-1} s_j a_j\right) u_{(1,1)} + \sin\left(\sum_{j=1}^{i-1} s_j a_j\right) u_{(1,2)}$$

where $s \in \{+1, -1\}^n$. Finally, since $u_{(n,1)}$ is at an angle a_n with $u_{(1,1)}$, we have

$$u_{(1,1)} = \cos\left(\sum_{j=1}^{n} s_j a_j\right) u_{(1,1)} + \sin\left(\sum_{j=1}^{n} s_j a_j\right) u_{(1,2)}$$

Since $\sum_j a_j = 1 < 2\pi$, we must have $\sum_{j=1}^{n} s_j a_j = 0$. Hence the set $I = \{i : s_i = +1\}$ yields a solution to the PARTITION problem on the instance $(a_1, \ldots, a_n)$. ■

The NP-hardness of $(2,3)$-PSD-COMPLETION follows from Lemma 14 and Lemma 15. There's a simple amplification one can do to prove hardness for $(k, 2k-1)$-PSD-COMPLETION for any even k. Simply take the matrix A from the $(2,3)$-PSD-COMPLETION reduction and output the matrix

$$M = \begin{bmatrix} A & 0 & \cdots & 0 \\ 0 & A & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A \end{bmatrix}$$

where A appears $k/2$ times. Then any completion of M of rank at most $2k-1$ will restrict to a completion of A of rank at most 3. We can also pad with additional zeros to boost the number of revealed entries up to $0.9n^2$. This completes the proof of Theorem 10.

3.2. Tolerating Errors

In this section we sketch the reduction in the proof of Theorem 11. Because PARTITION is only NP-hard in a weak sense (there exists an algorithm which runs in time polynomial in the size of the weights $a_1, \ldots, a_n$), we require a different starting problem to prove hardness while tolerating errors in the constraints.

Theorem 16 *For every constant $k \geqslant 3$, constant r with $2k \leqslant r \leqslant 4k-1$, and $\varepsilon < O(r^{-5})$, there is a reduction from* EXACT-ONE-IN-K-SAT *that, given an instance Φ, outputs a partial matrix P_Φ with the following property: If there is a rank r matrix M such that $|M(i,j) - P_\Phi(i,j)| < \varepsilon$ whenever $P_\Phi(i,j) \neq \perp$, then Φ is satisfiable. Furthermore, if Φ is satisfiable, there is a rank k completion of P_Φ.*

There are two main components to the reduction in Theorem 16. The first is the *variable gadget*, a set of constraints that forces only two configurations for a set of vectors, and we can interpret these configurations as being a $+1$ or -1 assignment to the variable. The second is the *clause gadget*, a set of constraints designed to force the interpreted assignment to be satisfying.

For each variable in the instance Φ, the variable gadget is a set of constraints that creates a $2k$-dimensional orthonormal basis. There is also a special ”reference basis” that we use as a reference point because inner product constraints are invariant to rotations. For each variable, we constrain its basis to be a special rotation of the reference basis. The rotation is special in the sense that it is a set of identical rotations in k pairs of two-dimensional subspaces. Because a rotation in two dimensions has exactly two configurations, rotate clockwise or rotate counter-clockwise, there are only two possible rotations of the variable's basis. We interpret each of these rotations as setting the variable to $+1$ or -1.

For each clause in Φ, the clause gadget is a set of constraints that are intended to construct a vector whose ith coordinate is the value (either $+1$ or -1) of the ith variable appearing in Φ. Finally, we constrain that the sum of the elements of Φ is exactly $(k-2)$. This forces exactly one of the coordinates of the vector to be -1, so the variables must be set to a satisfying assignment.

It is not obvious how we are able to force such specific structure on the rotations in the variable gadget even when the ambient dimension gets as high as $4k-1$. By constraining dot products of sums of basis vectors in the variable gadget we are able to resolve this problem. The full details and proof of Theorem 16 are located in the appendix for the interested reader.

4. Conclusion and Open Problems

Our goal was to narrow the gap between existing positive results on Matrix Completion and computational lower bounds. For a hardness result to be compelling it must account for natural algorithmic relaxations. We showed that several relaxations that are natural from an algorithmic machine learning point of view do not make the problem easier. From a complexity theoretic perspective, these are the first hardness of approximation results for Matrix Completion over the reals. A consequence of our work is that the popular incoherence assumption by itself is not sufficient to make the problem tractable. An interesting question is if conversely the assumption of uniformly random entries by itself already makes the problem easy.

Question 17 *Is Matrix Completion hard when the observed entries are chosen randomly, but the observed matrix is* not *incoherent?*

Another challenging question is to determine the precise hardness threshold for (k, r)-completion.

Question 18 *For any $k \geqslant 3$, what is the largest $r \geqslant k$ such that (k, r)-*COMPLETION *is hard? Can we find a matching algorithm?*

Resolving this question will likely require progress on both lower bounds and algorithms. Deceptively simple algorithmic questions are still open such as the following.

Question 19 *We know that $(3, \sqrt{n})$-*COLORING *is easy Wigderson (1982). Is $(3, \sqrt{n})$-*COMPLETION *easy?*

References

Sanjeev Arora, Eden Chlamtac, and Moses Charikar. New approximation guarantee for chromatic number. In *Proc. 38th Annual Symposium on Theory of Computing (STOC)*, pages 215–224. ACM, 2006.

B. Berger and J. Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 5(4):459–466, 1990.

Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *Proc.* 26*th COLT*, pages 1046–1066. JMLR, 2013.

Blum and Karger. An $o(n^{3/14})$-coloring algorithm for 3-colorable graphs. *IPL: Information Processing Letters*, 61, 1997.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computional Mathematics*, 9:717–772, December 2009.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Eden Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *Proc.* 48*th Foundations of Computer Science (FOCS)*, pages 691–701. IEEE, 2007.

Irit Dinur and Igor Shinkar. On the conditional hardness of coloring a 4-colorable graph with super-constant number of colors. In *APPROX-RANDOM*, pages 138–151. Springer, 2010.

Marianna E.-Nagy, Monique Laurent, and Antonios Varvitsiotis. Complexity of the positive semidefinite matrix completion problem with a rank constraint. In *Discrete Geometry and Optimization*, volume 69 of *Fields Institute Communications*, pages 105–120. Springer International Publishing, 2013.

Shai Fine and Katya Scheinberg. Efficient svm training using low-rank kernel representations. *J. Mach. Learn. Res.*, 2:243–264, 2002.

Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. In *Proc.* 26*th COLT*, pages 354–375. JMLR, 2013.

Nicholas J. A. Harvey, David R. Karger, and Sergey Yekhanin. The complexity of matrix completion. In *Proc.* 19*th Symposium on Discrete Algorithms (SODA)*, pages 1103–1111. ACM-SIAM, 2006.

Ken ichi Kawarabayashi and Mikkel Thorup. Coloring 3-colorable graphs with $o(n^{\{}1/5\})$ colors. In *STACS*, pages 458–469, 2014.

David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *JACM: Journal of the ACM*, 45, 1998.

Nati Linial, Shahar Mendelson, Gideon Schechtman, and Adi Shraibman. Complexity measures of sign matrices. In *Proc.* 39*th ACM Symposium on the Theory of Computing (STOC)*. ACM, 2007.

René Peeters. Orthogonal representations over finite fields and the chromatic number of graphs. *Combinatorica*, 16(3):417–431, 1996.

Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

V.Y.F. Tan, L. Balzano, and S.C. Draper. Rank minimization over finite fields: Fundamental limits and coding-theoretic interpretations. *Information Theory, IEEE Transactions on*, 58(4):2018–2039, April 2012.

Koji Tsuda, Shotaro Akaho, and Kiyoshi Asai. The em algorithm for kernel matrix completion with auxiliary data. *J. Mach. Learn. Res.*, 4:67–81, 2003.

Avi Wigderson. A new approximate graph coloring algorithm. In *Proc.* 14*th Annual ACM Symposium on Theory of Computing (STOC)*, pages 325–329, 1982.

Appendix A. Finding Good Factorizations of Low Rank Matrices

In this section we will prove a constructive version of Lemma 12, restated below for convenience.

Lemma 20 *(Lemma 2.1 restated) Let M be a matrix with rank r. Then there exists an r-dimensional factorization $M = XY^T$ such that every row vector of X and Y has norm at most $(cr)^{1/4}$, where $c = \max_{ij} |M(i,j)|$.*

In Linial et al. (2007) this lemma was proven in a non-constructive manner using John's Theorem from convex analysis. However, as the authors therein note, there is a simple semi-definite program (SDP) one can write to compute a good factorization. Unfortunately, this SDP may give a factorization that is not r-dimensional, but instead up to $(m+n+1)$-dimensional. Here we give a simple self-contained algorithm to compute the decomposition $M = XY^T$.

Proof [Proof (of an algorithm to construct decomposition as in Lemma 12)] Let M be a rank r matrix, and let $c = \max_{ij} |M(i,j)|$. To construct a good r-dimensional factorization of M, we proceed in a few steps:

1. Compute a factorization $M = UV^T$ such that U and V have short row vectors using semi-definite programming. This factorization is not necessarily r-dimensional.
2. Project every row of V onto the row space of U to get V'.
3. Factor $V' = YB$, where B is an orthonormal basis for the row space of V'.
4. Output $X = UB^T$ and Y. This will be an r-dimensional factorization with short rows.

For the first step, consider the following SDP with variables $\{u_i\}_{i=1}^m$, $\{v_j\}_{j=1}^n$ and η.

$$\begin{aligned}
&\text{Minimize } \eta \\
&\text{Subject to} \\
&\qquad u_i \cdot v_j = M(i,j) \qquad && \forall i \in \{1,\dots,m\} j \in \{1,\dots,n\} \\
&\qquad u_i \cdot u_i \leqslant \eta && \forall i \in \{1,\dots,m\} \\
&\qquad v_j \cdot v_j \leqslant \eta && \forall j \in \{1,\dots,n\}
\end{aligned}$$

This SDP computes the row vectors of two matrices U and V such that $M = UV^T$ and minimizes their row norms. The lemma in Linial et al. (2007) guarantees that there exists a factorization with the norm of every row vector of U and V bounded by $(cr)^{1/4}$, and thus the factorization found by the SDP is at least this good. However, since there are $m+n+1$ variables, the vectors returned could be up to $(m+n+1)$-dimensional.

Let V' be the matrix whose rows are the rows of V projected onto the row space of U. Note that we still have $M = U(V')^T$, and projections can only decrease the lengths of the rows of V. We make the following claim.

Claim 21 *Rank of V' is equal to rank of M, i.e., r.*

Proof Let r_u and r_v denote the ranks of U and V', respectively. Clearly $r_v \geqslant r$ because V' is a factor of M.

Now pick linearly independent subsets E of r_u rows of U and F of r_v rows of V'. Then EF^T is a submatrix of M, and thus must have rank at most r. However, we will argue below that the rank of EF^T must be equal to r_v, which shows that $r_v \leqslant r$. We already know that $r_v \geqslant r$, which implies that $r_v = r$ as desired.

To see that rank of EF^T is r_v, recall that the row space of V' is contained in the row space of U. Since E and F form a basis for row-spaces of U and V', row space of F is contained in the row space of E. So for any vector x, $x^T F$ lies in the row space of F, and thus in the row space of E. This implies that whenever $x^T F$ is nonzero, $E(F^T x)$ must be nonzero as well. So the rank of EF^T is the same as the rank of F^T, i.e. r_v. ■

Now since V' has rank r, there is a factorization $V' = YB$, where the rows of B are an r-dimensional orthonormal basis for the row space V', and Y is the matrix whose rows are the coordinates for the rows of V' in the basis B. Note that since B is orthonormal, the norms of rows of Y are the same as the norms of rows of V'. Let $X = UB^T$. Note that the rows of X are given by the the projections of the rows of U onto the basis B. Since B is orthonormal, this projection cannot increase the norms of the rows.

Observe that $XY^T = UB^T(V')^T = M$. Moreover XY^T is an r-dimensional factorization for M because B is an r-dimensional basis. Finally, the norms of rows of X and Y are smaller than the norms of rows of U and V given by the semidefinite program, as we only decreased the lengths at each step described above. ■

Appendix B. Coloring Hardness of Matrix Completion

In this section we will prove Theorem 8. The basic setup is the same as in the proof of Theorem 9. Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$. Now define the partial matrix $P_G \in (\mathbb{R} \cup \perp)^{n \times n}$ as follows:

$$P_G(i,j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } (i,j) \in E \\ \perp & \text{otherwise} \end{cases}.$$

Then, as before, if G is k-colorable with coloring function $f : V \to [k]$, then

$$M_f = \sum_{i \in [k]} 1_{f^{-1}(i)} 1_{f^{-1}(i)}^T$$

is a rank-k completion of P_G. Similar to the proof Theorem 9 we can also assume that M_f has coherence $\mu = 1$. We next prove that under some mild additional assumptions any entry-wise-approximate low-rank completion M of P_G yields a coloring of G with few colors (as opposed to just finding a large independent set as done earlier).

Lemma 22 *Let G be a graph and define the partial matrix P_G as above. Let M be a rank-r matrix with bounded coefficients $|M(i,j)| \leqslant c$ for all $i, j \in [n]$ such that M approximates P_G, i.e. $|M(i,j) - P_G(i,j)| < \varepsilon$ whenever $P_G(i,j) \neq \perp$. If $\varepsilon < 1/2$ then there is an algorithm that colors G using only*

$$\chi(G) \leqslant \left(\frac{4\sqrt{cr}}{1-2\varepsilon}\right)^{2r}$$

colors.

Proof Because the entries of M are bounded by c, Lemma 12 shows how to find a factorization $XY^T = M$ such that, if u_i and v_i are the row vectors of X and Y respectively, for all $i, j \in [n]$, we have $\|u_i\|, \|v_j\| \leqslant (cr)^{1/4}$. Note that since $u_i \cdot v_i \geqslant 1 - \varepsilon$ for all i, this implies a corresponding lower bound of $\|u_i\|, \|v_j\| \geqslant (1-\varepsilon)(cr)^{-1/4}$.

Now let W be a δ-net of the hypercube of sidelength $2(cr)^{1/4}$ in $\mathbb{R}^r$ centered at the origin (we will pick δ sufficiently small later). In particular we can pick W so that $|W| \leqslant (2(cr)^{1/4}/\delta)^r$. Now define the following coloring function $f : V \to W \times W : f(i) = (w_1, w_2)$ if $u_i \in B_\delta(w_1)$ and $v_i \in B_\delta(w_2)$. If u_i or v_i lie in multiple balls of the δ-net, simply pick one arbitrarily (perhaps the closest point $w \in W$). The function f must color every vertex because the δ-net covers every vector.

We prove that f is a valid coloring if $\delta < (1-2\varepsilon)(cr)^{-1/4}/2$. If i and j are the same color, v_j lies in the same ball as v_i. Then,

$$\begin{aligned}
|\langle u_i, v_i\rangle - \langle u_i, v_j\rangle| &= |\langle u_i, v_i - v_j\rangle| \\
&\leqslant \|u_i\| \cdot \|v_i - v_j\| \\
&\leqslant (cr)^{1/4}(2\delta) \\
&< (1-2\varepsilon)
\end{aligned}$$

and since $u_i \cdot v_i \geqslant 1 - \varepsilon$, we must have $|u_i \cdot v_j| > \varepsilon$, so $(i,j) \notin E$. Now note that the size of the coloring is at most

$$\chi(G) \leqslant |W|^2 = \left(\frac{2(cr)^{1/4}}{\delta}\right)^{2r} = \left(\frac{4\sqrt{cr}}{1-2\varepsilon}\right)^{2r}$$

and this completes the proof. ∎

Finally, we can pad the output matrix with zeros just like in the previous section in order to achieve a larger number (0.9 fraction) of revealed entries. Combined with Lemma 22, the above implies Theorem 8.

Appendix C. CSP Reduction for PSD-COMPLETION

In this section, we include a CSP based hardness reduction for PSD-COMPLETION that is robust and applicable with noise.

Let ϕ be an instance of a EXACT-ONE-IN-K-SAT with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$. A clause of ϕ is satisfied if and only if exactly one variable in the clause is set to -1. Recall that a partial PSD matrix is equivalent to a list of inner product constraints. We will describe our reduction in this framework. Let x_0 be a "reference variable" unused in ϕ. For every variable $x \in \{x_0, \ldots, x_n\}$, index a set of vectors $\{u_{(x,s)}\}_{s \in I}$ by $I = [2k] \cup \left(\binom{2k}{2} \times \{\pm 1\}\right)$. Then form the *internal variable* constraints

- $u_{(x,s)} \cdot u_{(x,s)} = 1$ for all $s \in I$.
- $u_{(x,i)} \cdot u_{(x,j)} = 1$ if $i = j$, and 0 otherwise.
- $u_{(x,i,j,+1)} \cdot u_{(x,i)} = u_{(x,i,j,+1)} \cdot u_{(x,j)} = \frac{1}{\sqrt{2}}$ for $(i,j) \in \binom{2k}{2}$.
- $u_{(x,i,j,-1)} \cdot u_{(x,i)} = -u_{(x,i,j,-1)} \cdot u_{(x,j)} = \frac{1}{\sqrt{2}}$ for $(i,j) \in \binom{r_1}{2}$.

These constraints force $\{u_{(x,i)}\}_i$ to be a $2k$-dimensional orthonormal basis for any x, as well as

$$u_{(x,i,j,+1)} = \frac{1}{\sqrt{2}} \left(u_{(x,i)} + u_{(x,j)}\right),$$

and

$$u_{(x,i,j,-1)} = \frac{1}{\sqrt{2}} \left(u_{(x,i)} - u_{(x,j)}\right).$$

Let $p : [2k] \to [2k]$ be the function $p(i) = i + 1$ if i is odd, and $p(i) = i - 1$ if i is even. Now for every variable $x \in \{x_1, \ldots, x_n\}$, form the *external variable* constraints

- $u_{(x_0,i)} \cdot u_{(x,j)} = 0$ if $j \neq p(i)$.
- $u_{(x_0,i,p(i),+1)} \cdot u_{(x,i,p(i),+1)} = 0$ for odd $i \in [2k]$.
- $u_{(x_0,i,j,+1)} \cdot u_{(x_0,p(i),p(j),-1)} = 0$ for $(i,j) \in \binom{2k}{2}$ and i, j odd.

Let C_0 be a "reference clause" not referring to any clause of ϕ. Index a set of vectors $\{u_C\}_C$ by $\{C_0, \ldots, C_m\}$. Form the *internal clause* constraints

- $u_{C_0} \cdot u_{C_0} = 1$.
- $u_{C_0} \cdot u_{(x_0,2g-1)} = \frac{1}{\sqrt{k}}$ for every $g \in [k]$.

  For every clause $C \in \{C_1, \ldots, C_m\}$, with variables $\{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$ with signs $\{s_1, \ldots, s_k\}$, i.e. each $s_g \in \{+1, -1\}$,
- $u_C \cdot u_C = 1$.
- $u_C \cdot u_{(x_{i_g},2g)} = \frac{s_g}{\sqrt{k}}$ for every $g \in [k]$.

Finally, for each clause $C \in \{C_1, \ldots, C_m\}$, the *external clause* constraints are

- $u_{C_0} \cdot u_C = (1 - 2/k)$ for every $C \in \{C_1, \ldots, C_m\}$.

Then we have the following theorem:

Theorem 23 *If there is a satisfying assignment to ϕ, then there is a set of vectors lying in $\mathbb{R}^{2k}$ satisfying the above constraints exactly. Conversely, if $\{u_{(x,s)}\}_{s\in I}$ and $\{u_{C_j}\}_{j\in[m]}$ are vectors in $\mathbb{R}^r$ that satisfy the constraints up to an additive $\pm\varepsilon$, and $r \leqslant 4k-1$ and $\varepsilon < 10^{-6}k^{-5}$, then there is a satisfying assignment to ϕ.*

Proof To start, we prove completeness. Let f be a satisfying assignment to ϕ. Then we propose the following vectors:

- $u_{(x_0,i)} = e_i$, where e_i is the ith standard basis vector. For every $i, j \in \binom{2k}{2}$, set $u_{(x_0,i,j,+1)}$ and $u_{(x_0,i,j,-1)}$ to the normalized sum and difference of $u_{(x_0,i)}$ and $u_{(x_0,j)}$.
- For every $x \in \{x_1, \ldots, x_n\}$, set $u_{(x,p(i))} = f(x)e_i$ for i odd and $u_{(x,p(i))} = -f(x)e_i$ for i even. For every $i, j \in \binom{2k}{2}$, set $u_{(x,i,j,+1)}$ and $u_{(x,i,j,-1)}$ to the normalized sum and difference of $u_{(x,i)}$ and $u_{(x,j)}$.
- $u_{C_0} = \frac{1}{\sqrt{k}} \sum_g u_{(x_0,2g-1)}$.
- For every $C \in \{C_1, \ldots, C_m\}$ with variables $\{x_{i_1}, \ldots, x_{i_k}\}$ and signs $\{s_1, \ldots, s_k\}$, set $u_C = \frac{1}{\sqrt{k}} \sum_g s_g u_{(x_{i_g},2g)}$.

It is clear that all internal constraints are satisfied. The external variable constraints are also easy to verify. For any clause C,

$$\begin{aligned} u_C \cdot u_{C_0} &= \frac{1}{k} \sum_{g,g'=1}^{k} s_g (u_{(x_{i_g},2g)} \cdot u_{(x_0,2g'-1)}) \\ &= \frac{1}{k} \sum_{g,g'=1}^{k} s_g f(x_{i_g})(e_{2g-1} \cdot e_{2g'-1}) \\ &= \frac{1}{k} \sum_{g=1}^{k} s_g f(x_{i_g}). \end{aligned}$$

Since f is a satisfying assignment to ϕ, exactly one variable in the clause is -1, thus the sum is exactly $(k-2)$, and so $u_C \cdot u_{C_0} = (1 - 2/k)$.

To prove soundness, we will start by assuming that all internal constraints are satisfied exactly, and only the external constraints contain errors. We will decode a satisfying assignment to ϕ under this assumption. Then we will show how to take the initial set of vectors and adjust them slightly to get a set of vectors perfectly satisfying the internal constraints.

Lemma 24 *Fix $r < 4k$. For each $x \in \{x_0, x_1, \ldots, x_n\}$, let $\{u_{(x,s)}\}_{s\in I}$ be a set of vectors in $\mathbb{R}^r$ satisfying the internal variable constraints exactly, and assume every external variable constraint is satisfied up to a small additive $\pm\delta$ such that $\delta < 1/12k$. Then for any $x \in \{x_1, \ldots, x_n\}$ and odd $i, i' \in [2k]$,*

$$\mathit{sign}(u_{(x_0,i)} \cdot u_{(x,p(i))}) = \mathit{sign}(u_{(x_0,i')} \cdot u_{(x,p(i'))})$$

and

$$1 \geqslant |u_{(x_0,i)} \cdot u_{(x,p(i))}| \geqslant 1 - 12\delta k$$

Proof Because the internal constraints are satisfied, $T_0 = \{u_{(x_0,i)}\}_{i\in[2k]}$ and $T_x = \{u_{(x,i)}\}_{i\in[2k]}$ are orthonormal bases, so there is an orthonormal transformation $Q : \mathbb{R}^r \to \mathbb{R}^r$ such that $Q(u_{(x_0,i)}) = u_{(x,i)}$. We write Q in any basis containing T_0:

$$Q = \begin{bmatrix} Q' & A \\ B & C \end{bmatrix}$$

where Q' is a transformation from T_0 to itself. Because $|u_{(x_0,i)} \cdot u_{(x,j)}| \leqslant \delta$ for any $j \neq p(i)$, Q' is at most δ except on the 2×2 block diagonal. Now $|u_{(x_0,i,p(i),+1)} \cdot u_{(x,i,p(i),+1)}| \leqslant \delta$ implies that $|Q(i, p(i)) + Q(p(i), i)| \leqslant 3\delta$. Finally, $|u_{(x_0,i,i',+1)} \cdot u_{(x,p(i),p(i')-1)}| \leqslant \delta$ for odd i and i' implies that $|Q(i, p(i)) - Q(i', p(i'))| \leqslant 3\delta$. Because of these conditions, we can write $Q' = R + S$, where

$$R = R_1 \oplus R_2 \oplus \cdots \oplus R_k$$

and

$$R_g = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix} \text{ for every } g \in [k]$$

and $|S(i, j)| \leqslant 6\delta$ for every i, j. Then for any unit vector $x \in \text{span}(T_0)$,

$$\|Q'x\| \geqslant \|Rx\| - \|Sx\| \geqslant |a| - 6\delta k.$$

In particular, let x be a null vector of B, which exists because B is an $(r - 2k) \times 2k$ matrix, and $r < 4k$. Then

$$Q \begin{bmatrix} x \\ 0 \end{bmatrix} = \begin{bmatrix} Q'x \\ 0 \end{bmatrix}$$

and since Q is an orthogonal transformation, this implies $\|Q'x\| = 1$, and thus $|a| \geqslant 1 - 6\delta k$. Now recalling the definition of Q, for any odd i, $u_{(x_0,i)} \cdot u_{(x,p(i))} = Q(i, p(i)) = a + S(i, p(i))$, and thus for any odd i

$$1 \geqslant |u_{(x_0,i)} \cdot u_{(x,p(i))}| \geqslant 1 - 12\delta k$$

and the sign is the same for any odd i if $\delta < \frac{1}{12k}$. The upper bound follows from the fact that $u_{(x_0,i)}$ and $u_{(x,p(i))}$ are unit vectors. ■

We take the interpretation that the variable x is set to $\text{sign}(u_{(x_0,1)} \cdot u_{(x,2)})$. Now we prove that this assignment must be a satisfying assignment because of the clause constraints.

Lemma 25 *For each $x \in \{x_0, \ldots, x_n\}$, let $\{u_{(x,s)}\}_{s\in I}$ satisfy the assumptions of the previous lemma, and for each $C \in \{C_1, \ldots, C_m\}$ with variables $\{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$ with signs $\{s_1, \ldots, s_k\}$, let*

$$u_C = \frac{1}{\sqrt{k}} \sum_{g=1}^{k} s_g u_{(x_{i_g},2g)}$$

and let

$$u_{C_0} = \frac{1}{\sqrt{k}} \sum_{g=1}^{k} u_{(x_0,2g-1)}.$$

Then if the constraints $u_{C_0} \cdot u_C = (1 - 2/k)$ are satisfied up to an additive $\pm\delta$ and

$$\delta < \min\left(\frac{2}{13k^2}, \frac{2}{24k + k^2}\right)$$

then the assignment $f(x) = \text{sign}(u_{(x_0,1)} \cdot u_{(x,2)})$ is a satisfying assignment.

Proof Let $C \in \{C_1, \ldots, C_m\}$. From the definitions of u_C and u_{C_0},

$$\begin{aligned} u_{C_0} \cdot u_C &= \frac{1}{k} \sum_{g,g'=1}^{k} s_g \left(u_{(x_0, 2g'-1)} \cdot u_{(x_{i_g}, 2g)} \right) \\ &= \frac{1}{k} \sum_{g=1}^{k} s_g \left(u_{(x_0, 2g-1)} \cdot u_{(x_{i_g}, 2g)} \right) + \frac{1}{k} \sum_{g \neq g'}^{k} s_g \left(u_{(x_0, 2g'-1)} \cdot u_{(x_{i_g}, 2g)} \right) \end{aligned}$$

where x_{i_g} are the variables appearing in clause C. We argue that $|u_{C_0} \cdot u_C - (1 - 2/k)| < \delta$ implies that f is satisfying. Note that the dot products in the first sum have magnitudes between 1 and $1 - 24\delta k$, and those in the second sum have magnitudes at most δ. If $\delta < 2/(24k + k^2)$, even if $k - 2$ of the dot products have sign $+1$ and 2 have sign -1,

$$\begin{aligned} (1 - 2/k) - u_{C_0} \cdot u_C &\geqslant (1 - 2/k) - \frac{1}{k} \left[(k-2) - 2(1 - 12\delta k) + k(k-1)\delta \right] \\ &= 2/k - \delta(23 + k) \\ &> \frac{2}{24k + k^2} \\ &> \delta \end{aligned}$$

Now if $\delta < 2/13k^2$, then even if all k of the dot products have sign $+1$

$$\begin{aligned} u_{C_0} \cdot u_C - (1 - 2/k) &\geqslant \frac{1}{k} \left[k(1 - 12\delta k) - k(k-1)\delta \right] - (1 - 2/k) \\ &= 2/k - \delta(13k - 1) \\ &> \frac{2}{13k^2} \\ &> \delta. \end{aligned}$$

These two facts mean that $|u_{C_0} \cdot u_C - (1 - 2/k)| < \delta$ implies that exactly one of the dot products in the first sign can have sign -1 and the rest must have sign $+1$, i.e. that f satisfies the clause C. This is true for every clause, so f must be a satisfying assignment. ■

These lemmas prove Theorem 11 if only the external variable constraints experience errors, and u_C and u_{C_0} are constructed properly. The next sequence of lemmas show how to transform the problem from errors on all constraints into errors on only external constraints.

Lemma 26 *Let $\{u_{(x,s)}\}_{s \in I}$ be a set of vectors satisfying the internal constraints to within an additive $\pm\varepsilon$ for $\varepsilon < O(1/k)$. Then there is a set of vectors $\{\tilde{u}_{(x,s)}\}_{s \in I}$ satisfying the internal constraints exactly such that*

$$\|\tilde{u}_{(x,s)} - u_{(x,s)}\| \leqslant 3\sqrt{\varepsilon}.$$

Proof Let $u_{(x,1)} = u_1^{\perp} + u_1^{\|}$, where $u_1^{\perp}$ is the part of $u_{(x,1)}$ perpendicular to the subspace span$(\{u_{(x,i)}\}_{i \neq 1})$, and $u_1^{\|}$ is the part parallel. Note that since $|u_{(x,1)} \cdot u_{(x,i)}| \leqslant \varepsilon$ for any $i \neq 1$,

$$\|u_1^{\|}\| \leqslant \sqrt{2k-1} \frac{\varepsilon}{\sqrt{1-\varepsilon}}.$$

Let $\tilde{u}_{(x,1)} = u_1^\perp/\|u_1^\perp\|$. Then

$$\begin{aligned}
\|\tilde{u}_{(x,1)} - u_{(x,1)}\| &= \|u_1^\perp(1/\|u_1^\perp\| - 1) - u_1^\|\| \\
&\leqslant \left(1/\|u_1^\perp\| - 1\right)\|u_1^\perp\| + \|u_1^\|\| \\
&= 1 - \sqrt{\|u_{(x,1)}\|^2 - \|u_1^\|\|^2} + \|u_1^\|\| \\
&\leqslant 1 - \sqrt{1-\varepsilon} + 2\sqrt{2k-1}\frac{\varepsilon}{\sqrt{1-\varepsilon}} \\
&\leqslant \varepsilon(1 + 2\sqrt{k})
\end{aligned}$$

where the last inequality uses a series approximation. Now proceeding inductively, let $u_{(x,i)} = u_i^\perp + u_i^\|$, where the subspace considered is $\text{span}(\{\tilde{u}_{(x,j)}\}_{j<i}) \cup \text{span}(\{u_{(x,j)}\}_{j>i})$, and we obtain an identical bound on the difference of norms. Note that the $\{\tilde{u}_{(x,i)}\}_{i\in r_1}$ satisfy the internal constraints, and define the remaining vectors to be the sums forced by the remaining constraints.

To bound the dot products of sums of basis vectors, we first bound

$$\begin{aligned}
&\left\|\frac{1}{\sqrt{2}}(u_{(x,i)} + u_{(x,j)}) - u_{(x,i,j,+1)}\right\| = \\
&\qquad= \left(\frac{1}{\sqrt{2}}(u_{(x,i)} + u_{(x,j)}) - u_{(x,i,j,+1)}\right)\cdot\left(\frac{1}{\sqrt{2}}(u_{(x,i)} + u_{(x,j)}) - u_{(x,i,j,+1)}\right)^{1/2} \\
&\qquad\leqslant \left(2(1+\varepsilon) + \varepsilon - 2\sqrt{2}(1/\sqrt{2} - \varepsilon)\right)^{1/2} \\
&\qquad\leqslant \sqrt{\varepsilon}\sqrt{3 + 2\sqrt{2}}
\end{aligned}$$

and now

$$\begin{aligned}
\left\|\tilde{u}_{(x,i,j,+1)} - u_{(x,i,j,+1)}\right\| &= \left\|\frac{1}{\sqrt{2}}(\tilde{u}_{(x,i)} + \tilde{u}_{(x,j)}) - u_{(x,i,j,+1)}\right\| \\
&\leqslant \left\|\frac{1}{\sqrt{2}}(\tilde{u}_{(x,i)} + \tilde{u}_{(x,j)}) - \frac{1}{\sqrt{2}}(u_{(x,i)} - u_{(x,j)})\right\| + \left\|u_{(x,i,j,+1)} - \frac{1}{\sqrt{2}}(u_{(x,i)} + u_{(x,j)})\right\| \\
&\leqslant \varepsilon\sqrt{2}(1 + 2\sqrt{k}) + \sqrt{\varepsilon}\sqrt{3 + 2\sqrt{2}} \\
&\leqslant 3\sqrt{\varepsilon}
\end{aligned}$$

where the last inequality follows because, since $\varepsilon < O(1/k)$, $\varepsilon\sqrt{k}$ is dominated by $\sqrt{\varepsilon}$, and we simply round the coefficients to integers. Note that this also means $3\sqrt{\varepsilon} > \varepsilon(1 + 2\sqrt{k})$. ■

Lemma 27 *For any two $x, y \in \{x_0, \ldots, x_n\}$ and $s \in I$,*

$$|\tilde{u}_{(x,s)} \cdot \tilde{u}_{(y,s)} - u_{(x,s)} \cdot u_{(y,s)}| \leqslant 7\sqrt{\varepsilon}$$

Proof To compress notation, let $u_{(x,s)} = u$ and $u_{(y,s)} = v$ and likewise for the tilde versions.

$$\begin{aligned}
|\langle\tilde{u} - u, \tilde{v} - v\rangle| &= |\langle\tilde{u}, \tilde{v}\rangle + \langle u, v\rangle - \langle\tilde{u}, v\rangle - \langle u, \tilde{v}\rangle| \\
&\geqslant |\langle\tilde{u}, \tilde{v}\rangle - \langle u, v\rangle| - |\langle\tilde{u}, v\rangle - \langle u, v\rangle| - |\langle u, \tilde{v}\rangle - \langle u, v\rangle| \\
&\geqslant |\langle\tilde{u}, \tilde{v}\rangle - \langle u, v\rangle| - \|\tilde{u} - u\| - \|\tilde{v} - v\|
\end{aligned}$$

And note that

$$|\langle \tilde{u} - u, \tilde{v} - v\rangle| \leqslant \|u - \tilde{u}\|\|v - \tilde{v}\|$$

and thus

$$\begin{aligned}|\langle \tilde{u}, \tilde{v}\rangle - \langle u, v\rangle| &\leqslant \|u - \tilde{u}\|\|v - \tilde{v}\| + \|u - \tilde{u}\| + \|v - \tilde{v}\| \\ &\leqslant 9\varepsilon + 6\sqrt{\varepsilon} \\ &\leqslant 7\sqrt{\varepsilon}\end{aligned}$$

■

Lemma 28 *For each clause $C \in \{C_1, \ldots, C_m\}$ containing variables $\{x_{i_1}, \ldots, x_{i_{r_{1/2}}}\}$, let $\tilde{u}_C = \frac{1}{\sqrt{k}} \sum_g s_g \tilde{u}_{(x_{i_g}, 2g)}$, and let $\tilde{u}_{C_0} = \frac{1}{\sqrt{k}} \sum_g \tilde{u}_{(x_{i_g}, 2g-1)}$. Then for any $C \in \{C_1, \ldots, C_m\}$,*

$$|\langle \tilde{u}_{C_0}, \tilde{u}_C\rangle - \langle u_{C_0}, u_C\rangle| \leqslant 5\sqrt{k\varepsilon}$$

Proof For any clause C containing variables $\{x_{i_1}, \ldots, x_{i_k}\}$, we bound the norm,

$$\begin{aligned}\left\| u_C - \frac{1}{\sqrt{k}} \sum_{g=1}^{k} s_g u_{(x_{i_g}, 2g)} \right\| &\leqslant \left[\left(u_C - \frac{1}{\sqrt{k}} \sum_{g=1}^{k} s_g u_{(x_{i_g}, 2g)} \right) \cdot \left(u_C - \frac{1}{\sqrt{k}} \sum_{g=1}^{k} s_g u_{(x_{i_g}, 2g)} \right) \right]^{1/2} \\ &\leqslant \left(1 + \varepsilon + k \cdot \frac{1}{k}(1 + \varepsilon) + \frac{1}{k} \cdot k(k-1)\varepsilon - 2k \cdot \frac{1}{\sqrt{k}}(1/\sqrt{k} - \varepsilon) \right)^{1/2} \\ &\leqslant (\varepsilon(1 + k + 2\sqrt{k}))^{1/2} \\ &\leqslant \sqrt{2k\varepsilon}\end{aligned}$$

and

$$\begin{aligned}\left\| \tilde{u}_C - \frac{1}{\sqrt{k}} \sum_{g=1}^{k} s_g u_{(x_{i_g}, 2g)} \right\| &\leqslant \frac{1}{\sqrt{k}} \sum_{g=1}^{k} \|\tilde{u}_{(x_{i_g}, 2g)} - u_{(x_{i_g}, 2g)}\| \\ &\leqslant \varepsilon(2k + \sqrt{k})\end{aligned}$$

and thus by triangle inequality,

$$\|\tilde{u}_C - u_C\| \leqslant \sqrt{2k\varepsilon} + \varepsilon(2k + \sqrt{k}) \leqslant 2\sqrt{k\varepsilon}$$

because $k\varepsilon < \sqrt{k\varepsilon}$. The identical calculation bounds $\|\tilde{u}_{C_0} - u_{C_0}\|$. Finally, just as in the proof of Lemma 27, for any clause C, we have

$$\begin{aligned}|\langle \tilde{u}_{C_0}, \tilde{u}_C\rangle - \langle u_{C_0}, u_C\rangle &\leqslant \|\tilde{u}_{C_0} - u_{C_0}\|\|\tilde{u}_C - u_C\| + \|\tilde{u}_{C_0} - u_{C_0}\| + \|\tilde{u}_C - u_C\| \\ &\leqslant 4k\varepsilon + 4\sqrt{k\varepsilon} \\ &\leqslant 5\sqrt{k\varepsilon}\end{aligned}$$

■

If every constraint experiences error at most ε, then we can construct an alternate solution that satisfies the internal constraints exactly and every external constraint experiences error at most $\delta \leqslant 5\sqrt{k\varepsilon}$. Since we require at most

$$\delta \leqslant \min\left(\frac{1}{24k}, \frac{2}{25k^2}, \frac{2}{48k + k^2}\right)$$

error on the external constraints, we can handle error at most $\varepsilon < O(k^{-5})$. This completes the proof of the theorem. ■