Hardness of Learning Halfspaces with Noise*

Venkatesan Guruswami† Prasad Raghavendra‡

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195

Abstract

Learning an unknown halfspace (also called a perceptron) from labeled examples is one of the classic problems in machine learning. In the noise-free case, when a halfspace consistent with all the training examples exists, the problem can be solved in polynomial time using linear programming. However, under the promise that a halfspace consistent with a fraction $(1-\varepsilon)$ of the examples exists (for some small constant $\varepsilon > 0$), it was not known how to efficiently find a halfspace that is correct on even 51% of the examples. Nor was a hardness result that ruled out getting agreement on more than 99.9% of the examples known.

In this work, we close this gap in our understanding, and prove that even a tiny amount of *worst-case* noise makes the problem of learning halfspaces intractable in a strong sense. Specifically, for arbitrary $\varepsilon, \delta > 0$, we prove that given a set of examples-label pairs from the hypercube a fraction $(1-\varepsilon)$ of which can be explained by a halfspace, it is NP-hard to find a halfspace that correctly labels a fraction $(1/2+\delta)$ of the examples.

The hardness result is tight since it is trivial to get agreement on $1/2$ the examples. In learning theory parlance, we prove that *weak proper agnostic learning of halfspaces* is hard. This settles a question that was raised by Blum *et al.* in their work on learning halfspaces in the presence of *random* classification noise [10], and in some more recent works as well. Along the way, we also obtain a strong hardness result for another basic computational problem: solving a linear system over the rationals.

1 Introduction

This work deals with the complexity of two fundamental optimization problems: solving a system of linear equations over the rationals, and learning a halfspace from labeled examples. Both these problems are "easy" when a perfect solution exists. If the linear system is satisfiable, then a satisfying assignment can be found in polynomial time by Gaussian Elimination. If a halfspace consistent with all the examples exists, then one can be found using linear programming. A natural question that arises is the following: If no perfect solution exists, but say a solution satisfying 99%

*Preliminary version appeared in the *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.

†Research supported by NSF CCF-0343672, a Sloan Research Fellowship, and a Packard Foundation Fellowship.

‡Research supported in part by NSF CCF-0343672.

of the constraints exists, can we find a solution that is nearly as good (say, satisfies 90% of the constraints)?

This question has been considered for both these problems (and many others), but our focus here is the case when the instance is near-satisfiable (or only slightly noisy). That is, for arbitrarily small $\varepsilon > 0$, a solution satisfying at least a fraction $(1 - \varepsilon)$ of the constraints is promised to exist, and our goal is to find an assignment satisfying as many constraints as possible. Sometimes, the problem is relatively easy to solve on near-satisfiable instances — notable examples being the Max 2SAT and Max HornSAT problems. For both of these problems, given a $(1-\varepsilon)$-satisfiable instance, it is possible to find in polynomial time, an assignment satisfying a fraction $1 - f(\varepsilon)$ of the clauses where $f(\varepsilon) \to 0$ as $\varepsilon \to 0$ [28, 12]. Our results show that in the case of solving linear systems or learning halfspaces, we are not so lucky and finding any non-trivial assignment for $(1-\varepsilon)$-satisfiable instances is NP-hard. We describe the context and related work as well as our results for the two problems in their respective subsections below.

Before doing that, we would like to stress that for problems admitting a polynomial time algorithm for satisfiability testing, hardness results of the kind we get, with gap at the right location (namely completeness $1 - \varepsilon$ for any desired $\varepsilon > 0$), tend to be hard to get. The most celebrated example in this vein is Håstad's influential result [20] which shows that given a $(1 - \varepsilon)$-satisfiable instance of linear equations modulo a prime p, it is NP-hard to satisfy a fraction $(\frac{1}{p} + \delta)$ fraction of them (note that one can satisfy a fraction $\frac{1}{p}$ of the equations by simply picking a random assignment). Recently, Feldman [16] established a result in this vein in the domain of learning theory. He proved the following strong hardness result for weak-learning monomials: given a set of example-label pairs a $(1 - \varepsilon)$ fraction of which can be explained by a monomial, it is hard to find a monomial that correctly labels a fraction $(1/2+\delta)$ of the examples. Whether such a strong negative result holds for learning halfspaces also, or whether the problem admits a non-trivial weak learning algorithm is mentioned as a notable open question in [16], and this was also posed by Blum, Frieze, Kannan, and Vempala [10] almost 10 years ago. In this work, we establish a tight hardness result for this problem. We prove that given a set of example-label pairs a fraction $(1 - \varepsilon)$ of which can be explained by a halfspace, finding a halfspace with agreement better than $1/2$ is NP-hard. This result was also established independently (for real-valued examples) in [17].

1.1 Solving linear systems

We prove the following hardness result for solving noisy linear systems over rationals: For every $\varepsilon, \delta > 0$, given a system of linear equations over $\mathbb{Q}$ which is $(1 - \varepsilon)$-satisfiable, it is NP-hard to find an assignment that satisfies more than a fraction δ of the equations. As mentioned above, a result similar to this was shown by Håstad [20] for equations over a large finite field. But this does not seem to directly imply any result over rationals. Our proof is based on a direct reduction from the Label Cover problem. While by itself quite straightforward, this reduction is a stepping stone to our more complicated reduction for the problem of learning halfspaces.

The problem of approximating the number of satisfied equations in an unsatisfiable system of linear equations over $\mathbb{Q}$ has been studied in the literature under the label MAX-SATISFY and strong hardness of approximation results have been shown in [7, 15]. In [15], it is shown that unless NP $\subset$ BPP, for every $\varepsilon > 0$, MAX-SATISFY cannot be approximated within a ratio of $n^{1-\varepsilon}$ where n is the number of equations in the system. (On the algorithmic side, the best approximation

algorithm for the problem, due to Halldorsson [19], achieves ratio $O(n/\log n)$). The starting point of the reductions in these hardness results is a system that is ρ-satisfiable for some ρ bounded away from 1 (in the completeness case), and this only worsens when the gap is amplified.

For the related problem of linear equations over integers $\mathbb{Z}$, a strong hardness result follows easily from Håstad's work [20]. Given a system of linear equations of the form $x+y-z=r \mod p$ with the promise that there exists a solution satisfying $(1-\varepsilon)$ fraction of the equations, it is NP-hard to find a solution that satisfies $\frac{1}{p}+\varepsilon$ fraction of the equations. By mapping $x+y-z=r \mod p$ to $x+y-z-pw=r \mod \mathbb{Z}$, it gives a corresponding NP-hardness result for linear equations over integers. By choosing a large enough prime p, this reduction implies the following result: Given a set of linear equations over integers with the promise that there is a solution satisfying $(1-\varepsilon)$ fraction of equations, it is NP-hard to find one that satisfies more than a δ fraction, for all positive ε and δ.

For the complementary objective of minimizing the number of unsatisfied equations, a problem called MIN-UNSATISFY, hardness of approximation within ratio $2^{\log^{0.99} n}$ is shown in [7] (see also [5]). In particular, for arbitrarily large constants c, the reduction of Arora *et al.* [7] shows NP-hardness of distinguishing between $(1-\gamma)$-satisfiable instances and instances that are at most $(1-c\gamma)$-satisfiable, for some γ. One can get a hardness result for MAX-SATISFY like ours by applying a standard gap amplification method to such a result (using a $O(1/\gamma)$-fold product construction), provided $\gamma = \Omega(1)$. However, as presented in [7], the reduction works with $\gamma = o(1)$. It is not difficult to modify their reduction to have $\gamma = \Omega(1)$. Our reduction is somewhat different, and serves as a warm-up for the reduction for learning halfspaces, which we believe puts together an interesting combination of techniques.

1.2 Halfspace learning

Learning halfspaces (also called *perceptrons* or *linear threshold functions*) is one of the oldest problems in machine learning. Formally, a halfspace on variables $x_1, \ldots, x_n$ is a Boolean function $I[w_1x_1 + w_2x_2 + \cdots + w_nx_n \geq \theta]$ for reals $w_1, \ldots, w_n, \theta$ (here $I[E]$ is the indicator function for an event E). For definiteness, let us assume that variables x_i are Boolean, that is, we are learning functions over the hypercube $\{0,1\}^n$. In the absence of noise, one can formulate the problem of learning a halfspace as a linear program and thus solve it in polynomial time. In practice, simple incremental algorithms such as the famous Perceptron Algorithm [1, 25] or the Winnow algorithm [24] are often used.

Halfspace-based learning algorithms are popular in theory and practice, and are often applied to labeled exampled sets which are not separable by a halfspace. Therefore, an important question that arises and has been studied in several previous works is the following: what can one say about the problem of learning halfspaces in the presence of noisy data that does not obey constraints induced by an unknown halfspace?

In an important work on this subject, Blum, Frieze, Kannan, and Vempala [10] gave a PAC learning algorithm for halfspaces in the presence of *random classification noise*. Here the assumption is that the examples are generated according to a halfspace, except with a certain probability $\eta < 1/2$, the label of each example is independently flipped. The learning algorithm in [10] outputs as hypothesis a decision list of halfspaces. Later, Cohen [13] gave a different algorithm for random classification noise where the output hypothesis is also a halfspace. (Such a learning algorithm

whose output hypothesis belongs to the concept class being learned is called a *proper learner*.) These results applied to PAC learning with respect to arbitrary distributions, but assume a rather "benign" noise model that can be modeled probabilistically.

For learning in more general noise models, an elegant framework called *agnostic learning* was introduced by Kearns *et al.* [23]. Under agnostic learning, the learner is given access to labeled examples (x, y) from a fixed distribution $\mathcal{D}$ over example-label pairs $X \times Y$. However, there is no assumption that the labels are generated according to a function from a specific concept class, namely halfspaces in our case. The goal of the learner is to output a hypothesis h whose accuracy with respect to the distribution is close to that of the best halfspace — in other words the hypothesis does nearly as well in labeling the examples as the best halfspace would.

In a recent paper [21], Kalai, Klivans, Mansour and Servedio gave an efficient agnostic learning algorithm for halfspaces when the marginal $\mathcal{D}_X$ on the examples is the uniform distribution on the hypercube or sphere S^{n-1}, or any log-concave distribution on $\mathbb{R}^n$. For any desired $\varepsilon > 0$, their algorithm produces a hypothesis h with error rate $\mathbf{Pr}_{(x,y)\in\mathcal{D}}[h(x) \neq y]$ at most $\mathsf{opt} + \varepsilon$ if the best halfspace has error rate opt. Their output hypothesis itself is not a halfspace but rather a higher degree threshold function.

When the accuracy of the output hypothesis is measured by the fraction of agreements (instead of disagreements or mistakes), the problem is called *co-agnostic learning*. The combinatorial core of co-agnostic learning is the *Maximum Agreement* problem: Given a collection of example-label pairs, find the hypothesis from the concept class (a halfspace in our case) that correctly labels the maximum number of pairs. Indeed, it is well-known that an efficient α-approximation algorithm to this problem exists iff there is an efficient co-agnostic proper PAC-learning algorithm that produces a halfspace that has agreement within a factor α of the best halfspace.

The Maximum Agreement for Halfspaces problem, denoted HS-MA, was shown to be NP-hard to approximate within some constant factor for the $\{0, 1, -1\}$ domain in [5, 9] (the factor was $261/262 + \varepsilon$ in [5] and $415/418 + \varepsilon$ in [9]). The best known hardness result prior to work was due to Bshouty and Burroughs [11], who showed an inapproximability factor of $84/85 + \varepsilon$, and their result applied even for the $\{0, 1\}$ domain. For instances where a halfspace consistent with $(1 - \varepsilon)$ of the examples exists (the setting we are interested in), an inapproximability result for HS-MA was *not* known for *any* fixed factor $\alpha < 1$. For the complementary objective of minimizing disagreements, hardness of approximating within a ratio $2^{O(\log^{1-\varepsilon} n)}$ is known [7, 5]. The problem of whether an α-approximation algorithm exists for HS-MA for some $\alpha > 1/2$, i.e., whether a weak proper agnostic learning algorithm for halfspaces exists, remained open. This question was also highlighted in Feldman's recent work [16] which proved that weak agnostic learning of monomials was hard.

In this paper, we prove that no $(1/2 + \delta)$-approximation algorithm exists for HS-MA for any $\delta > 0$ unless P = NP. Specifically, for every $\varepsilon, \delta > 0$, it is NP-hard to distinguish between instances of HS-MA where a halfspace agreeing on a $(1 - \varepsilon)$ fraction of the example-label pairs exists and where no halfspace agrees on more than a $(1/2+\delta)$ fraction of the example-label pairs. Our hardness result holds for examples drawn from the hypercube. Our result indicates that for *proper* learning of halfspaces in the presence of even small amounts of noise, one needs to make assumptions about the nature of noise (such as random classification noise studied in [10]) or about the distribution of the example-label pairs (such as uniform marginal distribution on examples as in [21]).

A similar hardness result was proved independently by Feldman *et al* [17] for the case when the examples are drawn from $\mathbb{R}^n$. In contrast, our proof works when the data points are restricted to the hypercube $\{0,1\}^n$, which is the natural setting for a Boolean function. Much of the complexity of our reduction stems from ensuring that the examples belong to the hypercube.

2 Preliminaries

The first of the two problems studied in this paper is the following:

Definition 2.1. *For constants c, s, satisfying $0 \leq s \leq c \leq 1$,* LINEQ-MA(c,s) *refers to the following promise problem: Given a set of linear equations over variables $X = \{x_1, \ldots, x_n\}$, with coefficients over $\mathbb{Q}$, distinguish between the following two cases:*

- *There is an assignment of values to the variables X that satisfies at least a fraction c of the equations.*
- *Every assignment satisfies less than a fraction s of the equations.*

In the problem of learning a halfspace to represent a Boolean function, the input consists of a set of positive and negative examples all from the Boolean hypercube. These examples are embedded in the real n-dimensional space $\mathbb{R}^n$ by the natural embedding. The objective is to find a hyperplane in $\mathbb{R}^n$ that separates the positive and the negative examples.

Definition 2.2. *Given two disjoint multisets of vectors $S^+, S^- \subset \{-1,1\}^n$, a vector $a \in \mathbb{R}^n$, and a threshold θ, the* agreement *of the halfspace $a \cdot v \geq \theta$ with (S^+, S^-) is defined to be the quantity*

$$|\{v | v \in S^+, a \cdot v \geq \theta\}| + |\{v | v \in S^-, a \cdot v < \theta\}| \ .$$

where the cardinalities are computed, by counting elements with repetition. In the HS-MA *problem, the goal is to find a, θ such that the halfspace $a \cdot v \geq \theta$ maximizes this agreement.*

Notice that there is no loss of generality in assuming the embedding to be $\{-1,1\}^n$. Our hardness results translate to other embeddings as well, because the learning problem in the $\{-1,1\}^n$ embedding can be shown to be equivalent to the learning problem on most natural embeddings such as $\{0,1\}^n$. Further, our hardness result holds even if both the inequalities $\{\geq, <\}$ are replaced by strict inequalities $\{>, <\}$.

To study the hardness of approximating HS-MA, we define the following promise problem:

Definition 2.3. *For constants c, s satisfying $0 \leq s \leq c \leq 1$, define* HS-MA(c,s) *to be the following promise problem: Given multisets of positive and negative examples $S^+, S^- \subset \{-1,1\}^n$ distinguish between the following two cases:*

- *There is a halfspace $a \cdot v \geq \theta$ that has agreement at least $c|S^+ \cup S^-|$ with (S^+, S^-).*
- *Every halfspace has agreement less than $s|S^+ \cup S^-|$ with (S^+, S^-).*

The hardness results in this paper are obtained by reductions from the Label Cover promise problem defined below.

Definition 2.4. *An instance of* LABELCOVER(c, s) *represented as* $\Gamma = (U, V, E, \Sigma, \Pi)$ *consists of a bipartite graph over node sets U,V with the edges E between them, such that all nodes in U are of the same degree. Also part of the instance is a set of labels* Σ*, and a set of mappings* $\pi_e : \Sigma \to \Sigma$ *for each edge* $e \in E$*. An assignment A of labels to vertices is said to satisfy an edge* $e = (u, v)$*, if* $\pi_e(A(u)) = A(v)$*. The problem is to distinguish between the following two cases:*

- *There exists an assignment A that satisfies at least a fraction c of the edge constraints* Π*.*
- *Every assignment satisfies less than a fraction s of the constraints in* Π*.*

The reductions in this paper use the following inapproximability result for Label Cover.

Theorem 2.5. *[27, 8] There exists an absolute constant* $\gamma > 0$ *such that for all large enough integer constants R, the gap problem* LABELCOVER$(1, \frac{1}{R^\gamma})$ *is NP-hard, even when the input is restricted to label cover instances with the size of the alphabet* $|\Sigma| = R$*.*

From the PCP theorem [8], it is easy to show that there exists an absolute constant ε such that LABELCOVER$(1, 1-\varepsilon)$ is NP-hard on instances where the size of alphabet $|\Sigma|$ is restricted to a small absolute constant (say 7). With this as the starting point, one applies the Parallel Repetition theorem [27] to obtain hardness of label cover instances over larger alphabet. On applying k-wise parallel repetition, the 1 vs $1-\varepsilon$ gap is amplified to 1 vs c^k for some absolute constant c, while the alphabet size also grows exponentially in k. This yields the above inapproximability result with the required polynomial dependence between the alphabet size R and the soundness $\frac{1}{R^\gamma}$.

Throughout this paper, we use the letter E to denote a linear equation/function, with coefficients $\{0, 1, -1\}$. For a linear function E, we use $V(E)$ to denote the set of variables with non-zero coefficients in E. We shall refer to $|V(E)|$ as the *arity* of E. Further, the evaluation $E(A)$ for an assignment A of real values to the variables is the real value obtained on substituting the assignment in the equation E. Hence an assignment A satisfies the equation $E = 0$ if $E(A) = 0$. For the purposes of the proof, we make the following definitions.

Definition 2.6. *An equation tuple T consists of a set of linear equations* $E_1, \ldots, E_k$ *and a linear function E called the scaling factor.*

Definition 2.7. *A tuple* $T = (\{E_1, E_2, \ldots, E_k\}, E)$ *is said to be disjoint if the sets of variables* $V(E_i)$*,* $1 \le i \le k$*, and* $V(E)$ *are all pairwise disjoint.*

Definition 2.8. *An assignment A is said to satisfy an equation tuple* $T = (\{E_1, \ldots, E_k\}, E)$*, if the scaling factor is positive, i.e.,* $E(A) > 0$*, and for every i,* $1 \le i \le k$*,* $E_i(A) = 0$*. For* $\beta \ge 0$*, an assignment A is said to* β*-satisfy an equation* E_i *if* $|E_i(A)| \le \beta \cdot |E(A)|$*. An assignment is said to* β*-satisfy the tuple T if it* β*-satisfies all the equations* $\{E_1, \ldots, E_k\}$ *and the scaling factor* $E(A) > 0$*. Note that* 0*-satisfying an equation tuple is the same as satisfying it.*

Definition 2.9. *An assignment A is said to be C-close to* β*-satisfying an equation tuple* $T = (\{E_1, \ldots, E_k\}, E)$*, if* $E(A) > 0$ *and* $|E_i(A)| > \beta |E(A)|$ *for at most C values of i,* $1 \le i \le k$*. An assignment is said to be C-far from* β*-satisfying an equation tuple T if it is not C-close to* β*-satisfying T.*

3 Overview of the Proof

Both the hardness results are obtained through a reduction from the Label Cover problem. Let us fix a Label Cover instance Γ over the set of labels $\{1, \ldots, R\}$.

Observe that the HS-MA problem amounts to finding an assignment satisfying the maximum number of a set of homogeneous linear inequalities (see Definition 2.2). Specifically, each example $v \in S^+ \cup S^-$ yields a homogeneous linear inequality $a \cdot v \geq \theta$ or $a \cdot v < \theta$ in the variables $a \in \mathbb{R}^n$ and θ. Although θ is permitted to take any real value, let us fix $\theta = 0$ for the sake of exposition.

The HS-MA problem is closely tied to solving systems of linear equations over reals. Given a homogeneous linear equation $a \cdot v = 0$ over variables $a = (a_1, \ldots, a_n)$, it can be encoded as two linear inequalities,

$$a \cdot v + \delta \geq 0 \quad \text{and} \quad a \cdot v - \delta < 0.$$

For the moment, let us suppose that δ is a variable forced to be a very small positive real number. An assignment to a satisfies both inequalities if and only if $a \cdot v \in [-\delta, \delta]$. In other words, an assignment satisfies either **two** or **one** inequality depending on whether the equation $(a \cdot v = 0)$ is approximately satisfied or not. Roughly speaking, using the above reduction, an NP-hardness result for LINEQ-MA(c, s) should translate in to an NP-hardness result for HS-MA$(\frac{1+c}{2}, \frac{1+s}{2})$. In this light, a natural approach would be encode the label cover instance as systems of linear equations, and use this encoding to obtain NP-hardness results for both LINEQ-MA and HS-MA problems.

However, implementing the outlined reduction from Label Cover to HS-MA poses considerable difficulties. For the above reduction from homogeneous linear equations to HS-MA, we need an NP-hardness of distinguishing between perfectly satisfiable equations, and equations that do not admit even an *approximate* solution. Further, in the above reduction, we used a variable δ taking only small positive real values. In a general HS-MA problem, no such constraint can be forced on the variables. More importantly, notice that all the examples $v \in S^+ \cup S^-$ in HS-MA are required to be vectors in $\{-1, 1\}^n$. Thus every coefficient in the system of homogeneous linear equations must take values either -1 or 1.

We shall refer to homogeneous linear systems as equation tuples. Specifically, an equation tuple T consists of a system of homogeneous linear equations $\{E_1, E_2, \ldots, E_k\}$ and a "scaling factor" E. Note that a solution to homogeneous linear system can be scaled to obtain a new solution. Hence the quality of an approximate solution is to be measured in terms of relative error, rather than absolute value of the error. The scaling factor E associated with the tuple T serves this purpose (see Definition 2.8).

The proof of hardness of HS-MA proceeds in three stages as described below. In the first stage, we reduce the Label Cover instance Γ to systems of homogeneous linear equations. More precisely, starting with the instance Γ, Verifier I generates a set of equation tuples $\mathcal{T}$ such that:

- If Γ is satisfiable then there is an assignment A that satisfies all the equation tuples in $\mathcal{T}$.
- If Γ is an unsatisfiable instance of Label Cover, then every assignment A fails to satisfy most of the equation tuples even approximately. Specifically, every assignment A can β-satisfy only a tiny fraction of the tuples in $\mathcal{T}$.

The set of equation tuples $\mathcal{T}$ is generated as follows. For each vertex u of the Label Cover instance Γ, we introduce R variables $\{u_1, \ldots, u_R\}$ which indicate the label assigned to vertex u, that is, u_i is 1 if and only if u is assigned label i. The edge constraints in Γ can be translated to linear equations over the variables $\{u_i\}$. As each vertex u is assigned one of R labels, only one of the variables $u_1, \ldots, u_R$ is non-zero in the intended solution. In other words, the intended solution is "sparse". Towards enforcing sparsity, Verifier I introduces constraints $u_i = 0$ for randomly chosen variables u_i. The set $\mathcal{T}$ of equation tuples consists of all the tuples output by Verifier I over all its random choices.

Fix an equation tuple $T = (\{E_1, E_2, \ldots, E_k\}, E)$. Consider the following set of 2^{k+1} inequalities obtained by ± 1 combinations of the equations E_i.

$$\sum_{i=1}^{k} w_i E_i + E \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{k} w_i E_i - E < 0 \qquad \text{for all } w \in \{-1, 1\}^k.$$

Given an assignment A, for any $w \in \{-1, 1\}^k$, both the inequalities corresponding to w are satisfied only if $\sum_i w_i E_i(A) \in [-E(A), E(A)]$. In the completeness analysis, if all equations are satisfied, i.e., evaluate to 0 on A, then any ± 1 combination also vanishes, and in turn belongs to $[-E(A), E(A)]$. Turning to soundness analysis, if the assignment A does not satisfy many equations E_i even approximately, then by definition, many of the values $E_i(A)$ are large in comparison to the scaling factor $E(A)$. Intuitively, a random ± 1 combination of large numbers($E_i(A)$) is small($\in [-E(A), E(A)]$) with very low probability. Thus if the assignment A does not satisfy almost all equations E_i approximately, then for almost all $w \in \{-1, 1\}^k$, it satisfies only one of the two inequalities. Conversely, if the assignment A satisfies more than $\frac{1}{2}$ of the inequalities, it must satisfy almost all equations in T approximately.

Roughly speaking, the argument in the preceding paragraph already yields a reduction from equation tuples to HS-MA. However, there are two key issues, that are addressed in the next two stages of the reduction : Verifier II and Verifier III.

Firstly, the equation tuples T need not be disjoint, i.e., different equations E_i can share variables. Hence the coefficients of variables in the ± 1 combination $\sum_i w_i E_i$ could take values outside $\{-1, 1\}$. Since all the inequalities in HS-MA are required to contain only $\{-1, 1\}$ coefficients, we address this issue in the second stage (Verifier II). More precisely, Verifier II takes as input the set $\mathcal{T}$ and creates a set of equation tuples $\mathcal{T}'$. The tuples in $\mathcal{T}'$ are disjoint, they are all over the same set of variables, and each variable appears in exactly one equation of every tuple. Further, in the soundness case, almost all tuples are at least C-far from being β-satisfied. Verifier II thus plays two roles: (i) it makes the equations in each tuple have disjoint support, and (ii) in the soundness case, every assignment not just fails to β-satisfy most of the tuples, but is in fact C-far from β-satisfying most of the tuples. Both these facts are exploited by Verifier III in the third stage.

The number of inequalities produced by picking all ± 1 combinations of equations $E_1, \ldots, E_k$ is exponential in k. As $k = \Omega(n)$, this would make the running time of the reduction exponential. In the third stage (Verifier III), we will present a carefully chosen polynomial sized sample space of ± 1 combinations which is sufficient to perform the soundness analysis. More formally, given a tuple T and an assignment A, Verifier III distinguishes (by checking suitable inequalities) between the cases when an assignment A satisfies a tuple T and when it is C-far from β-satisfying T.

The equation tuples $T \in \mathcal{T}'$ generated by Verifier II are given as input to Verifier III. Since the

tuples in T are disjoint, the resulting inequalities have all the variables with coefficients $\{-1, 1\}$. Further the inequalities generated by Verifier III are designed to have a common variable (a threshold θ) on the right hand side. Hence the inequalities generated by the combined verifier (the three stages) correspond naturally to training examples in the learning problem.

To show NP-hardness for LINEQ-MA, the set of tuples $\mathcal{T}$ output by Verifier I are rather easily converted in to a set of equations. This is achieved by creating several equations for each equation tuple $T \in \mathcal{T}$, such that a large fraction of these are satisfied if and only if T is satisfied. The details of this conversion are described in Section 5.

4 Verifier I

Let $\Gamma = (U, V, E, \Sigma, \Pi)$ be an instance of Label Cover with $|\Sigma| = R$. This verifier produces a set of equation tuples, which form input to the next stage in the reduction (Verifier II). For each vertex $u \in U \cup V$, the equation tuples have variables $\{u_1, \ldots, u_R\}$ taking values in $\mathbb{Q}$. The solution that we are targeting is an encoding of the assignment to the Label Cover instance. So if a vertex u is assigned the label i by an assignment A, then we want $u_i = 1$ and $u_j = 0$ for $j \neq i, 1 \leq j \leq R$. We construct an equation tuple for every t-tuple of variables corresponding to vertices in U, for a suitable parameter t that will be chosen shortly.

Let W denote the set of variables $W = \{u_i | u \in U, 1 \leq i \leq R\}$.
For each t-tuple X of variables from W, construct the equation tuple T as follows:

- $\mathcal{P}_1$: For every pair of vertices $u, v \in U \cup V$, an equation,

$$\sum_{i=1}^{R} u_i - \sum_{j=1}^{R} v_j = 0.$$

- $\mathcal{P}_2$: For each edge $e = (u, v) \in E$, the Label Cover constraints for the edge,

$$\Big(\sum_{j \in \pi_e^{-1}(i)} u_j \Big) - v_i = 0, \text{ for all } 1 \leq i \leq R.$$

- $\mathcal{P}_3$: For each variable $w \in X$,

$$w = 0.$$

- The scaling factor is $\mathcal{P}_4$: $\sum_{i=1}^{R} u_i$ for an arbitrary fixed vertex $u \in U \cup V$.

Output the tuple $T = (\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3, \mathcal{P}_4)$.

Theorem 4.1. *For every $0 < \delta_1, \varepsilon_1, \gamma < 1$, there exists sufficiently large R, t such that if $\Gamma = (U, V, E, \Sigma, \Pi)$ is an instance of Label Cover with $|\Sigma| = R$, then with the choice of $\beta_1 = \frac{1}{R^3}$ the following holds:*

- *If Γ is satisfiable, then there is an assignment A that satisfies at least $1 - \varepsilon_1$ fraction of the output tuples.*

- *If no assignment to* Γ *satisfies a fraction* $\frac{1}{R^\gamma}$ *of the edges, then every assignment* A β_1*-satisfies less than a fraction* δ_1 *of the output tuples.*

Proof. Let us choose parameters $c_0 = \ln(1/\delta_1)$ and $t = 4c_0R^{1-\gamma}$, for $R = (\frac{4c_0}{\varepsilon_1})^{1/\gamma}$. We present the completeness and soundness arguments in turn.

Completeness: Given an assignment $\mathcal{A}$ to the Label Cover instance that satisfies all the edges, let A denote the corresponding integer solution. Clearly, the integer solution A satisfies:

- All equations in $\mathcal{P}_1$ and $\mathcal{P}_2$.
- $\left(1 - \frac{1}{R}\right)$ fraction of the equations in the set: $\{w = 0 | w \in W\}$.

Since t equations of the form $w = 0$ are present in each tuple, the assignment A satisfies at least $\left(1 - \frac{1}{R}\right)^t$ of the tuples. By the choice of parameters R and t, we have $\left(1 - \frac{1}{R}\right)^t > 1 - \frac{t}{R} = 1 - \varepsilon_1$.

Soundness: Suppose there is an assignment A that β_1-satisfies at least a fraction δ_1 of the tuples generated. Clearly A must β_1-satisfy all the equations $\mathcal{P}_1$ and $\mathcal{P}_2$, since they are common to all the tuples. Further by definition of β_1-satisfaction, the scaling factor $\mathcal{P}_4(A) > 0$. Normalize the assignment A such that the scaling factor $\mathcal{P}_4$ is equal to 1. As all the equations in $\mathcal{P}_1$ are β_1-satisfied, we get

$$1 - \beta_1 \leq \sum_{i=1}^{R} v_i \leq 1 + \beta_1, \text{ for all } v \in U \cup V. \tag{1}$$

Further, we claim that the assignment A β_1-satisfies at least a fraction $(1 - \frac{c_0}{t})$ of the equations of the form $w = 0$ for $w \in W$. Otherwise, with t of these equations belonging to every tuple, less than $(1 - \frac{c_0}{t})^t < \delta_1$ tuples will be β_1-satisfied by A. Recall that all vertices in U have same degree. Hence by an averaging argument, for at least half the edges $e = (u, v)$, at least a $(1 - \frac{2c_0}{t})$ fraction of the constraints $u_i = 0$ are β_1-satisfied. Let us call these edges *good*.

For every vertex, define the set of labels Pos as follows,

$$\text{Pos}(u) \quad = \begin{cases} \{i \in \Sigma \mid u_i \geq 8\beta_1\} & \text{if } u \in U, \\ \{i \in \Sigma \mid u_i \geq 8\beta_1(R+1)\} & \text{if } u \in V. \end{cases}$$

The set $\text{Pos}(w)$ is non-empty for each vertex $w \in U \cup V$, because otherwise $\sum_{i=1}^{R} w_i < 8\beta_1(R+1) \cdot R \leq 1 - \beta_1$, a contradiction to (1). Further if $e = (u, v)$ is a *good* edge then for at least a fraction $1 - \frac{2c_0}{t}$ of the labels $1 \leq i \leq R$, we have $u_i \leq \beta_1$. Hence $|\text{Pos}(u)| \leq (\frac{2c_0}{t})R = \frac{R^\gamma}{2}$. Further, since all the constraints $\mathcal{P}_2$ are β_1-satisfied, we know that

$$\left| \sum_{i \in \pi_e^{-1}(j)} u_i - v_j \right| \leq \beta_1.$$

Thus for every label, $j \in \text{Pos}(v)$, there is at least one label $i \in \text{Pos}(u)$ such that $\pi_e(i) = j$. For every vertex $w \in U \cup V$, assign a label chosen uniformly at random from $\text{Pos}(w)$. For any *good* edge $e = (u, v)$, the probability that the constraint π_e is satisfied is at least $\frac{1}{|\text{Pos}(u)|} \geq \frac{2}{R^\gamma}$. Since at least half of the edges are *good*, this shows that there is an assignment to the Label Cover instance that satisfies at least a fraction $1/R^\gamma$ of the edges. □

5 Linear equations over Rationals

Theorem 5.1. *For all $\varepsilon, \delta > 0$, the problem* LINEQ-MA$(1-\varepsilon, \delta)$ *is NP-hard.*

Proof. The result is obtained by a simple reduction that converts each equation tuple generated by Verifier I in to a set of equations. In fact, the notion of β-satisfiability is not necessary for the following proof. The crucial ingredient in the proof is the following well known fact about univariate polynomials:

Fact 5.2. *A real univariate polynomial of degree at most n that is not identically zero has at most n real roots.*

Given a Label Cover instance Γ, the reduction analyzed in Theorem 4.1 is applied with $\varepsilon_1 = \varepsilon$, $\delta_1 = \frac{\delta}{2}$ to obtain a set of equation tuples $\mathcal{T}$. From $\mathcal{T}$, a set of equations over $\mathbb{Q}$ is obtained as follows:

For each tuple $T = (\{E_1, \ldots, E_n\}, E) \in \mathcal{T}$, include the following set of equations :

$$E_1 + y \cdot E_2 + \ldots + y^{n-1} E_n + y^n (E - 1) \quad = \quad 0,$$

for all values of $y = 1, 2, \ldots, t$, where $t = \frac{(n+1)}{\delta_1}$.

Completeness: Observe that if Γ is satisfiable then the corresponding assignment A has a scaling factor $E(A) = 1$. Further for every equation tuple T that is satisfied by A, we have $E_i(A) = 0$ for all $1 \leq i \leq n$. Hence the assignment A satisfies at least $(1 - \varepsilon)$ fraction of the equations.

Soundness: Suppose there is an assignment A that satisfies at least a fraction $\delta = 2\delta_1$ of the equations. Hence for at least δ_1 fraction of the tuples, at least δ_1 fraction of the equations are satisfied. Let us refer to these tuples as *nice*.

For an equation tuple $T = (\{E_1, \ldots, E_n\}, E)$, consider the polynomial p of degree at most n given by

$$p(y) = E_1(A) + y \cdot E_2(A) + \ldots + y^{n-1} E_n(A) + y^n (E(A) - 1).$$

By Fact 5.2, the polynomial p has at most n real roots unless it is identically zero. Further the polynomial p is identically zero if and only if the tuple T is completely satisfied. Hence if the tuple T is not completely satisfied by A, then at most $\frac{n}{t} < \delta_1$ fraction of the equations corresponding to T can be satisfied. Thus every *nice* tuple T is completely satisfied by A. So the assignment A satisfies at least a fraction δ_1 of the tuples. □

The coefficients of variables in the above reduction could be exponential in n (their binary representation could use polynomially many bits). Using an alternate reduction, we can prove a similar hardness even if all the coefficients are integers bounded by a constant depending only on ε, δ - thus requiring only constant number of bits per coefficient. Moreover the arity of all the equations (i.e., the number of variables with a nonzero coefficient) can be restricted to a constant. The proof of the following theorem is presented in Appendix A.

Theorem 5.3. *For any constants $\varepsilon, \delta > 0$, there exist $B, b > 0$ such that* LINEQ-MA$(1-\varepsilon, \delta)$ *is NP-hard even on linear systems where each equation has arity at most b and all coefficients are integers bounded in absolute value by B.*

Using a different approach, the above result has been improved in [18] to linear systems where each equation has arity at most 3, all the coefficients are $\{+1, -1\}$ and all the constants are bounded.

6 Verifier II

The main ideas in the construction of the second verifier are described below.

To obtain halfspace examples on the hypercube $\{-1, 1\}^n$, our reduction requires that the equation tuples T are disjoint. But the equation tuples T output by Verifier I are not disjoint, i.e., there are variables that occur in more than one equation in T. This problem can be solved by using multiple copies of each variable, and using different copies for different equations. However, it is important to ensure that the different copies of the variables are consistent. To ensure this the verifier does the following: it has a very large number of copies of each variable in comparison to the number of equations. The equations of the tuple T are checked on a very small number of the copies. On all the copies that are not used for equations in T, the verifier checks pairwise equality. Any given copy of a variable is used to check an equation in T for only a very small fraction of cases. For most random choices of Verifier II, the copy of the variable is used for consistency checking. This way most of the copies are ensured to be consistent with each other.

The pairwise consistency checks made between the copies must also satisfy the disjointness property. So the verifier picks a matching at random, and performs pairwise equality checks on the matching. It can be shown that even if there are a small number of bad copies, they will get detected by the matching with high probability.

If a single equation is unsatisfied in T, at least C equations need to be unsatisfied on the output tuple. This is easily ensured by checking each equation in T on many different copies of the variables. As all the copies are consistent with each other, if one equation is unsatisfied in T a large number of equations in the output tuple will be unsatisfied.

Let us say the tuple T consists of equations $\{E_1, \ldots, E_m\}$ and a scaling factor E over variables $\{u_1, \ldots, u_n\}$. Let us denote by n_0 the maximum arity of an equation in T. We use superscripts to identify different copies of the variables. Thus $u_i^{(j)}$ refers to the variable corresponding to j^{th} copy of the variable u_i. Further for an equation/linear function E, the notation $E^{(j)}$ refers to the equation E over the j^{th} copies of variables $V(E)$. By the notation $M_i(j, k)$, we refer to the following pairwise equality check:

$$M_i(j,k): \qquad u_i^{(j)} - u_i^{(k)} = 0.$$

Let M, P be parameters which are even integers. The set of variables used by Verifier II consists of

- M copies for variables *not* in $V(E)$,
- $M + 1$ copies of variables in $V(E)$.

Given an equation tuple $T = (\{E_1, E_2, \ldots, E_r\}, E)$, Verifier II checks each of the r equations E_i on P copies of the variables. On the remaining copies, the verifier performs a set of pairwise equality checks. We now define family of pseudorandom permutations that we will use in Verifier II.

Definition 6.1. *Two distributions D_1, D_2 over a finite set Ω are said to be η-close to each other if the variation distance $\|D_1 - D_2\| = \frac{1}{2}\sum_{\omega\in\Omega} |D_1(\omega) - D_2(\omega)|$ is at most η.*

Definition 6.2. *A multiset of permutations Π of $\{1 \dots M\}$ is said to be k-wise η-dependent if for every k-tuple of distinct elements $(x_1, \dots, x_k) \in \{1 \dots M\}$, the distribution $(f(x_1), f(x_2), \dots, f(x_k))$ for $f \in \Pi$ chosen uniformly at random is η-close to the uniform distribution on k-tuples.*

Let Π denote a set of 4-wise η-dependent permutations of $\{1, \dots, M\}$. Explicit constructions of such families of permutations of size polynomial in M (specifically $\left(\frac{M}{\eta}\right)^{O(1)}$) are known, see [26, 22].

Verifier II takes as input the set of equation tuples $\mathcal{T}$ generated by Verifier I. The details of Verifier II are described below.

For each equation tuple $T = (\{E_1, E_2, \dots, E_r\}, E) \in \mathcal{T}$,
For each $s \in \{1, \dots, M+1\}$, and a permutation π in a set Π of 4-wise η-dependent permutations,

- Choose $E^{(s)}$ as the scaling factor. Re-number the remaining M copies of $V(E)$ with $\{1, \dots, M\}$ arbitrarily. Specifically, we shall index the set $\{E^{(i)} | i \neq s, 1 \leq i \leq M+1\}$ with the set $\{1, \dots, M\}$.
- Construct sets of equations $\mathcal{P}$ and $\mathcal{M}$ as follows:

$$\mathcal{P} = \{E_\ell^{(\pi(j))} | 1 \leq \ell \leq r, P(\ell-1)+1 \leq j \leq P\ell\},$$
$$\mathcal{M} = \{M_i(\pi(j), \pi(j+1)) | u_i^{(\pi(j))} \notin V(\mathcal{P}), j \text{ odd}\}.$$

- Output the tuple $(\mathcal{P} \cup \mathcal{M}, E^{(s)})$.

Theorem 6.3. *For all $1 > \varepsilon_2, \delta_2 > 0$ and positive integers C, n_0, there exists constants M, P, η such that: Given a set of equation tuples $\mathcal{T}$ of which each tuple is of arity at most n_0 and has the same scaling factor E, the following holds:*

- *If an assignment A, satisfies a fraction $1 - \varepsilon_2$ of the tuples $T \in \mathcal{T}$ then there exists an assignment A' which satisfies a fraction $1 - \varepsilon_2$ of the tuples output by the verifier.*
- *If no assignment β_1-satisfies a fraction $\frac{\delta_2}{2}$ of the tuples $T \in \mathcal{T}$, then no assignment A' is C-close to $\beta = \frac{\beta_1}{9n_0}$-satisfying a fraction δ_2 of the output tuples.*

Proof. Fix the parameters M, P, η, C_0 as follows,

$$C_0 = \lceil \max(8C, 8/\delta_2) \rceil + 1, \tag{2}$$
$$P = \left\lceil 9n_0 \left(C + \frac{1}{\delta_2} \right) \right\rceil, \tag{3}$$
$$M = \left\lceil \frac{40PrC_0}{\delta_2} \right\rceil + 1, \tag{4}$$
$$\eta = \frac{\delta_2}{64C_0}. \tag{5}$$

The completeness proof is clear, since an assignment A' consisting of several copies of A satisfies the exact same tuples that A satisfies.

Suppose an assignment A' is C-close to β-satisfying a δ_2-fraction of the output tuples. Then for at least a fraction $\frac{\delta_2}{2}$ choices of input tuple $T \in \mathcal{T}$, at least a fraction $\frac{\delta_2}{2}$ of the output tuples corresponding to T are C-close to being β-satisfied by A'. Let us call these input tuples T *good*. For a good tuple T, there are at least $\frac{\delta_2}{4}$ fraction of choices of s for which with probability more than $\frac{\delta_2}{4}$ over the choice of permutation π, the output tuple is C-close to being β-satisfied (by A'). These values of s (and the associated copy of the scaling factor $E^{(s)}$) are said to be *nice* with respect to T.

Lemma 6.4. *Let $E^{(s)}$ be a nice scaling factor of T. Then, for every equation $E_\ell \in T$, there exist at least $P - C$ values of j for which $|E_\ell^{(j)}(A')| \leq \beta |E^{(s)}(A')|$.*

Proof. Since $E^{(s)}$ is a nice scaling factor, for at least one permutation $\pi \in \Pi$, the assignment A' is C-close to β-satisfying the generated tuple. Since each equation E_ℓ is checked on P different copies, at least $P - C$ of the copies must be β-satisfied by A'. □

Lemma 6.5. *For $C_0 = \lceil \max(8C, 8/\delta_2) \rceil + 1$, the following holds: Let $E^{(s)}$ be a scaling factor that is nice with respect to some good tuple T. For every variable u_i that occurs in the equations of T (including E), all but C_0 of copies of u_i are $2\beta|E^{(s)}(A')|$ close to each other, i.e., $|A'(u_i^{(j_1)}) - A'(u_i^{(j_2)})| \leq 2\beta |E^{(s)}(A')|$ for all $j_1, j_2 \in M'$ for a set M' with $|\{1, \ldots, M\} - M'| \leq C_0$.*

Proof. As $E^{(s)}$ is a nice scaling factor with respect to T, for at least a fraction $\frac{\delta_2}{4}$ choices of $\pi \in \Pi$ the assignment A' is C-close to β-satisfying the output tuple $\mathcal{P} \cup \mathcal{M}$. In particular, this means that with probability at least $\frac{\delta_2}{4}$, at most C of the consistency checks in $\mathcal{M}$ fail to be β-satisfied.

Define a copy $u_i^{(j)}$ to be *far* from $u_i^{(j_1)}$ if $|A'(u_i^{(j)}) - A'(u_i^{(j_1)})| > \beta |E^{(s)}(A')|$. We define a copy $u_i^{(j)}$ to be *bad*, if it is far from at least $M/2$ other copies. Suppose there are more than C_0 bad copies of the variable u_i. For notational convenience, we can assume that the first C_0 copies $\{u_i^{(1)}, u_i^{(2)}, \ldots, u_i^{(C_0)}\}$ are bad. We will prove below that with high probability over the choice of $\pi \in \Pi$, at least $2C + 1$ of these bad copies will be involved in checks in $\mathcal{M}$ that are not β-satisfied. This will in turn imply that more than C of the checks in $\mathcal{M}$ are not β-satisfied.

Observe that for every variable u_i, at most Pr of its copies are used in equations in $\mathcal{P}$. Further, for a uniformly random permutation, the probability that (the index of) a fixed bad copy is the image of a fixed $j \in \{1, \ldots, M\}$ is $\frac{1}{M}$. Hence the probability that a fixed bad copy is used for an equation in $\mathcal{P}$ is at most $\frac{Pr}{M}$. Since Π is η-dependent, the probability that one of the C_0 bad copies $u_i^{(j)}, 1 \leq j \leq C_0$, is used for some equation in $\mathcal{P}$ is at most $C_0(\frac{Pr}{M} + \eta)$. So, except with this probability, *all* bad copies are assigned to consistency checks in $\mathcal{M}$.

A *bad* copy $u_i^{(j)}, 1 \leq j \leq C_0$ fails to β-satisfy a check in $\mathcal{M}$ whenever a far copy is mapped next to it by the permutation π. Formally, let ℓ be such that $j = \pi(\ell)$. The variable $u_i^{(j)}$ will fail a check in $\mathcal{M}$ if either ℓ is even and $u_i^{(\pi(\ell)-1)}$ is *far* from $u_i^{(j)}$ or if ℓ is odd and $u_i^{(\pi(\ell)+1)}$ is *far* from $u_i^{(j)}$. Let Z_j, $1 \leq j \leq C_0$ be the $0,1$ random variable indicating the event that a *far* copy is mapped next to $u_i^{(j)}$ by the permutation. We shall estimate the values of $\mathbf{E}_{\pi \in \Pi}[Z_j]$ and $\mathbf{E}_{\pi \in \Pi}[Z_{j_1} Z_{j_2}]$ for different values of j, j_1, j_2.

Let F_j denote the set of j_0, such that $u_i^{(j_0)}$ is *far* from $u_i^{(j)}$. Let ℓ be such that $j = \pi(\ell)$. Then

for each j, $1 \leq j \leq C_0$ we have

$$\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_j] = \frac{1}{2}\mathop{\mathbf{Pr}}_{\pi\in\Pi}[Z_j = 1|\ell \text{ is odd}] + \frac{1}{2}\mathop{\mathbf{Pr}}_{\pi\in\Pi}[Z_j = 1|\ell \text{ is even}] \geq \Big(\frac{1}{2} + \frac{1}{2}\Big)\Big(\frac{|F_j|}{M-1} - \eta\Big) \geq \frac{1}{3}.$$

Further for $1 \leq j_1 < j_2 \leq C_0$,

$$\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_1}Z_{j_2}] \;=\; \mathop{\mathbf{Pr}}_{\pi\in\Pi}[Z_{j_2} = 1, Z_{j_1} = 1].$$

To estimate $\mathbf{Pr}_{\pi\in\Pi}[Z_{j_2} = 1, Z_{j_1} = 1]$, we first estimate $\mathbf{Pr}[Z_{j_2} = 1, Z_{j_1} = 1]$ where the probability is over a uniformly random permutation π. Let l be such that $\pi(l) = j_1$. We shall estimate $\mathbf{Pr}[Z_{j_2} = 1, Z_{j_1} = 1]$ conditioned on l being odd, the other case follows along similar lines. There are two cases:

Case $1 : \pi(l+1) = j_2$.
This happens with probability $\frac{1}{M-1}$, and further in this case $Z_{j_1} = 1$ implies $Z_{j_2} = 1$.

Case 2*:* $\pi(l+1) \neq j_2$.
Let $\pi(l') = j_2$ for some $l' \neq l+1$. Given that $u_i^{(j_1)}$ is mapped next to one of its *far* copies, there are $M-3$ choices for $\pi(l'+1)$. Hence the probability that a *far* copy is mapped next to $u_i^{(j_2)}$ is at most $\frac{|F_{j_2}|}{M-3}$.

Hence,

$$\begin{aligned}
\mathbf{Pr}[Z_{j_2} = 1|Z_{j_1} = 1] &\leq \frac{1}{M-1} + \left(\frac{M-2}{M-1}\right)\cdot\frac{|F_{j_2}|}{M-3},\\
&\leq \frac{|F_{j_2}|+1}{M-3}.\\
\mathbf{Pr}[Z_{j_2} = 1, Z_{j_1} = 1] &\leq \frac{|F_{j_1}|}{M-1}\cdot\frac{|F_{j_2}|+1}{M-3},\\
&\leq \frac{|F_{j_1}||F_{j_2}|}{(M-1)^2} + \frac{3}{M-3}.
\end{aligned}$$

Since Π is a 4-wise η-dependent family we get,

$$\begin{aligned}
\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_1}Z_{j_2}] &= \mathop{\mathbf{Pr}}_{\pi\in\Pi}[Z_{j_2} = 1, Z_{j_1} = 1],\\
&\leq \mathbf{Pr}[Z_{j_2} = 1, Z_{j_1} = 1] + \eta,\\
&\leq \frac{|F_{j_1}||F_{j_2}|}{(M-1)^2} + \frac{3}{M-3} + \eta,\\
&\leq \left(\frac{|F_{j_1}|}{M-1} - \eta\right)\left(\frac{|F_{j_2}|}{M-1} - \eta\right) + \frac{3}{M-3} + 3\eta,\\
&\leq \mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_1}]\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_2}] + \frac{3}{M-3} + 3\eta.
\end{aligned}$$

Define the random variable X to be $\sum_{i=1}^{C_0} Z_i$. Then,

$$\mathop{\mathbf{E}}_{\pi\in\Pi}[X] = \sum_{i=1}^{C_0}\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_i] \geq \frac{C_0}{3}.$$

The variance σ^2 is given by

$$\begin{aligned}
\sigma^2 &= \mathop{\mathbf{E}}_{\pi\in\Pi}[X^2] - (\mathop{\mathbf{E}}_{\pi\in\Pi}[X])^2 \\
&= \sum_{j_1=1}^{C_0}\left(\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_1}^2] - (\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_1}])^2\right) + \sum_{j_1=1}^{C_0}\sum_{j_2\neq j_1}^{C_0}\left(\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_1}Z_{j_2}] - \mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_1}]\mathop{\mathbf{E}}_{\pi\in\Pi}[Z_{j_2}]\right) \\
&\leq C_0 + 2\binom{C_0}{2}\left(\frac{3}{M-3} + 3\eta\right).
\end{aligned}$$

Therefore $\sigma^2 < 2C_0$, for $M, \frac{1}{\eta}$ sufficiently large compared to C_0. Using Chebyshev's inequality, it follows that

$$\mathbf{Pr}[X \leq 2C] \leq \frac{2C_0}{(\frac{C_0}{3} - 2C)^2}.$$

Putting these facts together, it follows that the probability over the choice of $\pi \in \Pi$ (once a nice value of s is picked) that at most C of the consistency checks in $\mathcal{M}$ fail to be β-satisfied is at most

$$C_0\left(\frac{Pr}{M} + \eta\right) + \frac{2C_0}{(\frac{C_0}{2} - 2C)^2} < \frac{\delta_2}{4},$$

by the choice of parameters M, C_0 and η. This contradicts the niceness of the scaling factor $E^{(s)}$.

It must thus be the case that at most C_0 copies of the variable u_i are bad. Now if neither of the copies $u_i^{(j_1)}$ and $u_i^{(j_2)}$ are bad, then both $A'(u_i^{(j_1)})$ and $A'(u_i^{(j_2)})$ are within $\beta|E^{(s)}(A')|$ of the the value assigned by A' to more than half the copies of u_i. This implies that they must themselves be within $2\beta|E^{(s)}(A')|$ of each other. Thus all but C_0 copies of u_i are $2\beta|E^{(s)}(A')|$ close to each other. □

Returning to the proof of Theorem 6.3, fix T^* to be an arbitrary *good* tuple. Define s_0 to be its *nice* value for which the corresponding scaling factor $E^{(s_0)}(A')$ has the smallest absolute value. (Note that $E^{(s)}(A) > 0$ for every scaling factor $E^{(s)}$ that is nice with respect to T^*, hence $E^{(s_0)}(A') > 0$.) From Lemma 6.5, we know that all but C_0 of the copies of every variable are $2\beta|E^{(s_0)}(A')|$ close to each other. Delete all the bad copies (at most C_0) of each variable. Further, delete all the variables in $V(E^{(s_0)})$. Now define an assignment A as follows: The value of $A(u_i)$ is the average of all the copies of u_i that have survived the deletion. We claim that the assignment A β_1-satisfies all the good tuples $T' \in \mathcal{T}$.

Observe that the arity of scaling factor E is at most n_0, and at most $C_0 + 1$ copies of each variable are deleted. Since there are at least $\frac{\delta_2}{4}M$ nice scaling factors and $\frac{\delta_2}{4}M > n_0(C_0+1)$, there exists a nice scaling factor $E^{(s_1)}$ of T^* such that no variable of $V(E^{(s_1)})$ is deleted. Further by definition of s_0, $|E^{(s_1)}(A')| \geq |E^{(s_0)}(A')|$.

From Lemma 6.5, for the average assignment A and any undeleted variable $u_i^{(j)}$ occurring in an equation of T^*, we have

$$|A(u_i) - A'(u_i^{(j)})| \leq 2\beta|E^{(s_0)}(A')| \leq 2\beta|E^{(s_1)}(A')|. \tag{6}$$

In particular, $A(u_i) \geq A'(u_i^{(s_1)}) - 2\beta|E^{(s_1)}(A')|$. Using this for each of the variables in $V(E^{(s_1)})$, we get

$$|E(A)| \geq (1 - 2\beta|V(E^{(s_1)})|)|E^{(s_1)}(A')| \geq (1 - 2\beta n_0)|E^{(s_1)}(A')|.$$

Substituting back in (6), we get

$$|A(u_i) - A'(u_i^{(j)})| \leq \frac{2\beta}{(1 - 2\beta n_0)}|E(A)| \leq 4\beta|E(A)|. \tag{7}$$

Consider any good tuple $T' \in \mathcal{T}$. The same argument used for T^* shows that there exists a scaling factor $E^{(j_0)}$ that is nice with respect to T' and none of whose variables have been deleted. Using Equation (7) for variables $u_i^{(j_0)} \in V(E^{(j_0)})$, we have $A'(u_i^{(j_0)}) \leq A(u_i) + 4\beta|E(A)|$. Thus,

$$|E^{(j_0)}(A')| \leq |E(A)| + 4\beta \cdot n_0|E(A)|. \tag{8}$$

Using Lemma 6.4, and the fact $P - C > n_0C_0$, we can conclude for every equation $E_\ell \in T'$, there exists j_1 such that $|E_\ell^{(j_1)}(A')| \leq \beta|E^{(j_0)}(A')|$, and no variable of $V(E_\ell^{(j_1)})$ is deleted. Similar to (8) we get

$$|E_\ell(A)| \leq |E_\ell^{(j_1)}(A')| + 4\beta \cdot n_0|E(A)|.$$

Therefore,

$$|E_\ell(A)| \quad \leq \quad (\beta + 4\beta^2 n_0 + 4\beta n_0)|E(A)| \leq 9\beta n_0|E(A)| = \beta_1|E(A)|,$$

implying that the assignment A β_1-satisfies the tuple T'. Hence the assignment A β_1-satisfies all the good tuples. Recalling that at least a fraction $\delta_2/2$ of the tuples are good, the result of Theorem 6.3 follows. □

7 Verifier III

Given a equation tuple $T = (\{E_1, \ldots, E_n\}; E)$, Verifier III checks whether the assignment A satisfies T or is not even C-close to β-satisfying T. Towards this, we define the following notation: For a tuple of equations $\mathcal{E} = (E_1, \ldots, E_n)$, and a vector $v \in \{-1, 1\}^n$, define $\mathcal{E} \cdot v = \sum_{i=1}^n v_i E_i$.

Let V_i for an integer i, denote a 4-wise independent subset of $\{-1, 1\}^i$. Polynomial size constructions of such sets are well known, see for example [4, Chap. 15]. The verifier has an additional parameter m chosen to be some integer $m > \frac{16}{\delta_3^2}$. The details of the verifier are described below.

- Partition the set of equations $\{E_1, \ldots, E_n\}$ using n random variables that are C-wise independent and take values $\{1, \ldots, m\}$. Let us say the parts are $\mathcal{E}_i, 1 \leq i \leq m$.
- For each part $\mathcal{E}_i$, pick a random vector, $v_i \in V_{n_i}$ where $n_i = |\mathcal{E}_i|$. Compute linear functions $B_i, 1 \leq i \leq m$,
$$B_i = \mathcal{E}_i \cdot v_i.$$
Construct $B = (B_1, B_2, \ldots, B_m)$.
- Pick a vector w uniformly at random from $\{-1, 1\}^m$.
- With probability $\frac{1}{2}$, check one of the following two inequalities:
$$B \cdot w + E \geq \theta, \tag{9}$$
$$B \cdot w - E < \theta. \tag{10}$$
Accept if the check is satisfied, else Reject.

Throughout this article, we shall choose the parameter m to be a prime number. For prime values of m, polynomial size spaces of C-wise independent variables taking values $\{1, \ldots, m\}$ can be obtained using BCH codes with alphabet size m, and minimum distance $C + 1$ (See [2]).

Theorem 7.1. *For every $0 < \beta, \delta_3 < 1$ and all $m > \frac{16}{\delta_3^2}$, $C > \frac{4m}{\beta^4 \delta_3^2}$ the following holds: Given the equation tuple $T = (\{E_1, \ldots, E_n\}, E)$ and an assignment A,*

- *If the assignment A satisfies T, then with $\theta = 0$, the verifier accepts with probability 1.*
- *If the assignment A is C-far from β-satisfying the tuple T, then irrespective of the value of θ, the verifier accepts with probability less than $\frac{1}{2} + \frac{\delta_3}{2}$.*

Proof. The proof shall make use of Lemmas 7.2 and 7.3, which we shall present later in the section.

For an assignment A that satisfies the tuple T, we have $E_j(A) = 0, 1 \leq j \leq n$, and $E(A) > 0$. Hence for all the random choices of Verifier III, B is the 0 vector, and $E > 0$. Therefore, with the choice $\theta = 0$, all the checks made by the verifier succeed. (In fact, the $\geq$ conditions hold with a strict inequality.)

Suppose the assignment A is C-far from β-satisfying the tuple T. If $E(A) \leq 0$, then clearly at most one of these two inequalities checked can be satisfied, and the proof is complete. Hence, we assume $E(A) > 0$.

There are at least C values $\{E_j(A) | 1 \leq j \leq n\}$ that have absolute value greater than $\beta |E(A)|$. Let us refer to these E_j as *large*. The probability that any of the parts $\mathcal{E}_i$ contains less than $B_0 = \lceil \frac{2}{\beta^2} \rceil$ *large* values is at most $m \binom{C}{B_0} (1 - \frac{1}{m})^{C - B_0}$. From Lemma 7.2, for a part $\mathcal{E}_i$ that has at least B_0 *large* values,
$$\mathbf{Pr}[|B_i(A)| > |E(A)|] \geq \frac{1}{12}.$$
Assuming that all the parts have at least B_0 *large* values, we bound the probability that less than $\frac{m}{24}$ parts have $|B_i(A)| > |E(A)|$. Let us fix a partition $\{\mathcal{E}_i\}$ such that all parts contain at least B_0 *large* values. For a fixed partition $\{\mathcal{E}_i\}$, each of the events $|B_i(A)| > |E(A)|$ are independent by the

choice of the vectors v_i. Thus for a fixed partition $\{\mathcal{E}_i\}$, we use the Chernoff bounds and obtain

$$\mathbf{Pr}\left[\left|\{i \ : \ |B_i(A)| > |E(A)|\}\right| < \frac{m}{24} \ \middle| \ \{\mathcal{E}_i\}\right] \leq e^{-\frac{m}{96}}.$$

Note that the above inequality holds for every fixed partition with all parts containing at least B_0 large values. Hence, conditioned on the event that all parts have at least B_0 large values we have

$$\mathbf{Pr}\left[\left|\{i \ : \ |B_i(A)| > |E(A)|\}\right| < \frac{m}{24}\right] \leq e^{-\frac{m}{96}}.$$

Consider the case in which there are at least $m_0 = \frac{m}{24}$ parts with $|B_i(A)| > |E(A)|$. In this case, from Lemma 7.3 we can conclude

$$\mathbf{Pr}\big[B \cdot w - \theta \in [-E(A), E(A)]\big] \leq \frac{\binom{m_0}{m_0/2}}{2^{m_0-1}}.$$

Overall we have,

$$\mathbf{Pr}\big[B \cdot w - \theta \in [-E(A), E(A)]\big] \leq m\binom{C}{B_0}\left(1 - \frac{1}{m}\right)^{C-B_0} + e^{-\frac{m}{96}} + \frac{\binom{m_0}{m_0/2}}{2^{m_0-1}}.$$

The value of $B_0 = \frac{2}{\beta^2}$ is fixed, so for large enough values of C, m with $C > m$ the above probability is less than δ_3. In particular this holds for choices $m > 16/\delta_3^2$ and $C > 4m/\beta^4\delta_3^2$. Observe that if $B \cdot w - \theta \notin [-E(A), +E(A)]$, at most one of the two checks performed by the verifier can be satisfied. Hence the probability of acceptance of the verifier is less than $\frac{1}{2} + \frac{\delta_3}{2}$. □

Lemma 7.2. *For all $\beta > 0$, and a constant $B_0 \geq \frac{2}{\beta^2}$, if $V \subseteq \{-1, 1\}^n$ is a 4-wise independent space of vectors then for any $a \in \mathbb{R}^n$ with at least B_0 of its components greater than β in absolute value,*

$$\mathbf{Pr}[\ |a \cdot v| > 1] \geq \frac{1}{12},$$

where the probability is over random choice of $v \in V$.

Proof. The following proof is along the lines of proof of Theorem 2.2 in [3], and we include it here for the sake of completeness. Define a random variable $x = |a \cdot v|^2$ for v chosen uniformly at random from V. Then it can be shown that,

$$\begin{aligned} \mathbf{E}[x] &= \|a\|_2^2, \\ \mathbf{E}[x^2] &= 3\|a\|_2^4 - 2\|a\|_4^4 < 3\|a\|_2^4. \end{aligned}$$

Since at least B_0 components of a are larger than β, we have $\|a\|_2^2 > B_0\beta^2 \geq 2$. Therefore, if $\mathbf{Pr}[|a \cdot v| > 1] = \alpha < \frac{1}{12}$, then

$$\mathbf{E}[x|x > 1] \ \geq \ \frac{1}{\alpha}(\|a\|_2^2 - (1 - \alpha) \cdot 1) > \frac{1}{2\alpha}\|a\|_2^2.$$

Using the Cauchy-Schwartz inequality, we know

$$\mathbf{E}[x^2|x > 1] \geq (\mathbf{E}[x|x > 1])^2 > \frac{1}{4\alpha^2}\|a\|_2^4.$$

Therefore, we get

$$\mathbf{E}[x^2] \geq \mathbf{E}[x^2|x>1]\,\mathbf{Pr}[x>1] > \frac{1}{4\alpha}\|a\|_2^4 > 3|a\|_2^4,$$

which is a contradiction. □

The following Lemma is well-known in probability as the 'Littlewood-Offord Problem'[14] (see Theorem 11.1.1 in [6]).

Lemma 7.3. *For every vector $a \in \mathbb{R}^m$ with at least K of its components > 1 in absolute value and a number $\theta \in \mathbb{R}$,*

$$\mathbf{Pr}[\theta - 1 \leq a \cdot v \leq \theta + 1] \leq \frac{\binom{K}{K/2}}{2^{K-1}},$$

where the probability is over random choice of $v \in \{-1,1\}^m$.

8 Hardness of HS-MA: Putting the Verifiers Together

Theorem 8.1 (Main Result)**.** *For all $\varepsilon, \delta > 0$, the problem* HS-MA$(1-\varepsilon, \frac{1}{2}+\delta)$ *is NP-hard.*

Proof. Given a Label Cover instance Γ, we use Verifier I with parameters $\delta_1 = \frac{\delta}{4}, \varepsilon_1 = \varepsilon$ to obtain a set of equation tuples $\mathcal{T}$. Let $R = R(\varepsilon_1, \delta_1)$ denote the parameter obtained in Theorem 4.1. Note that the maximum arity of the equations in the tuples is $2R$. Using the set of equation tuples $\mathcal{T}$ as input, Verifier II with parameters $\varepsilon_2 = \varepsilon_1, \delta_2 = \frac{\delta}{2}, \beta_1 = \frac{1}{R^3}$ and arity $n_0 = 2R$ generates a set of equation tuples $\mathcal{T}'$. Apply Theorem 7.1 with $\delta_3 = \delta, \beta = \frac{1}{18R^4}$ to check one of the equation tuples $T \in \mathcal{T}'$.

Completeness: If the Label Cover instance Γ is satisfiable, Verifier I outputs a set of tuples, such that there is an assignment satisfying $1-\varepsilon_1 = 1-\varepsilon$ of the output tuples. Hence by applying Theorems 6.3,7.1, it is clear that there is an assignment A, that satisfies at least $1-\varepsilon$ of the inequalities.

Soundness: Suppose there is an assignment A, which satisfies $\frac{1}{2}+\delta$ fraction of the inequalities, then for at least $\frac{\delta}{2}$ fraction of the tuples $T \in \mathcal{T}'$, Verifier III accepts with probability at least $\frac{1}{2}+\frac{\delta}{2}$. Therefore A is C-close to β-satisfying at least $\frac{\delta}{2} = \delta_2$-fraction of the tuples $T \in \mathcal{T}'$. Using Theorem 6.3, it is clear that there exists an assignment A' which β_1- satisfies at least a fraction $\frac{\delta_2}{2} = \frac{\delta}{4} = \delta_1$ fraction of tuples $T \in \mathcal{T}$. Hence by Theorem 4.1, the Label Cover instance Γ has an assignment that satisfies at least a fraction $\frac{1}{R^\gamma}$ of its edges.

The number of random bits used by the Verifier I is given by $O(R^{1-\gamma}\log n)$. In Verifier II, a total of $\log M + O(\log M + \log\frac{1}{\eta}) = O(\log n)$ random bits are needed. Verifier III uses at most $(C-1)\log n + 2\sum \log n_i + m = O(\log n)$ random bits. Hence the entire reduction from LABELCOVER to HS-MA is a polynomial time reduction. □

By choosing the parameters of the above reduction appropriately, the following stronger hardness result can be shown

Theorem 8.2. *For all constants $\kappa > 0$, the problem* HS-MA$(1-\frac{1}{\log^\kappa n}, \frac{1}{2}+\frac{1}{\log^\kappa n})$ *is Quasi-NP-hard.*

Proof. We begin the reduction with the following quasi-NP hardness of Label Cover:

Theorem 8.3. *Unless $NP \subseteq DTIME(2^{\log^k n})$ for some constant k, the following is true: For all constants $c, m > 0$, there is no $2^{\log^m n}$ time algorithm for* LABELCOVER$(1, \frac{1}{R^\gamma})$ *with alphabet size $R = \Theta(\log^c n)$.*

Let $\varepsilon = \delta = \frac{1}{\log^\kappa n}$. Let Γ be a Label Cover instance with alphabet size $R = \Theta(\log^{2\kappa/\gamma} n)$. The three steps of the reduction are carried out as follows:

- Verifier I

  From Theorem 4.1, Verifier I on input a Label Cover instance with alphabet size $R(\varepsilon_1, \delta_1) = (\frac{4 \ln 1/\delta_1}{\varepsilon_1})^{1/\gamma}$, outputs a $(1-\varepsilon_1, \delta_1, \beta_1)$-set of equation tuples with $\beta_1 = 1/R^3$. By $(1-\varepsilon_1, \delta_1, \beta_1)$-set of equation tuples, we mean the following :

  - Completeness Case: $(1 - \varepsilon_1)$ fraction of equation tuples are satisfied.
  - Soundness Case: at most δ_1 fraction of equation tuples can be β_1-satisfied.

  Apply the Theorem 4.1 with $\varepsilon_1 = \varepsilon$ and $\delta_1 = \frac{\delta}{4}$ on the Label Cover instance Γ. Recall that the alphabet size of Γ was chosen to be $R = \Theta(\log^{2\kappa/\gamma} n) > (\frac{4 \ln 1/\delta_1}{\varepsilon_1})^{1/\gamma}$. Thus we obtain a $(1 - \varepsilon, \frac{\delta}{4}, \frac{1}{R^3})$-set of equation tuples $\mathcal{T}$.
  The number of random bits used by Verifier I is given by $O(t \log n) = O(4 \ln \frac{1}{\delta_1} R^{1-\gamma} \log n) = O(\log^{2\kappa/\gamma+1} n)$.

- Verifier II

  From Theorem 6.3, Verifier II on input a $(1 - \varepsilon_2, \frac{\delta_2}{2}, \beta_1)$-set of equation tuples with arity n_0, outputs a $(1 - \varepsilon_2, \delta_2, \beta, C)$-set of equation tuples $\mathcal{T}'$ where:

  - Completeness Case: $(1 - \varepsilon_2)$ fraction of tuples are satisfied.
  - Soundness Case: At most δ_2 fraction of tuples are C-close to being $\beta = \frac{\beta_1}{9n_0}$-satisfied.

  Feed Verifier II with the $(1 - \varepsilon, \frac{\delta}{4}, \frac{1}{R^3})$-set of equation tuples $\mathcal{T}$. Thus we are applying Theorem 6.3 with $\varepsilon_2 = \varepsilon$ and $\delta_2 = \frac{\delta}{2}$ and $\beta_1 = 1/R^3$. Further we pick the parameter $C = 100(18)^4 R^{16} \log^{4\kappa} n$. By construction of Verifier I, the maximum arity of the set of equations in $\mathcal{T}$ is $n_0 = 2R$ and the number of equations in a tuple $r = n^{O(\log \log n)}$. This implies that $\beta = \frac{\beta_1}{9n_0} = \frac{1}{18R^4}$.
  From Verifier II, we obtain a set of equation tuples $\mathcal{T}'$ with parameters $(1-\varepsilon, \frac{\delta}{2}, \beta = \frac{1}{18R^4}, C = 100(18)^4 R^{16} \log^{4\kappa} n)$.
  Now we analyze the total randomness used by Verifier II. Towards this, we determine the internal parameters C_0, P, η, M. Since R was chosen to be $R = \Theta(\log^{2\kappa/\gamma} n)$, we have $C = 100(18)^4 R^{16} \log^{4\kappa} n) = O(\log^{36\kappa/\gamma} n)$. The other parameters of Verifier II are determined by Equations 2, 3, 4 and 5. Firstly, we get $C_0 = \max(8C, 8/\delta_2) = O(\log^{36\kappa/\gamma} n)$. The values of P and $1/\eta$ are given by $P = \lceil 9(2R)(C + \frac{1}{\delta_2}) \rceil = O(\log^{38\kappa/\gamma} n)$, $\frac{1}{\eta} = 64C_0/\delta_2 = O(\log^{36\kappa/\gamma+\kappa} n)$. Finally the value of $M = \lceil \frac{40PrC_0}{\delta_2} \rceil + 1 = O(n^{O(\log \log n)} \log^{\kappa+74\kappa/\gamma} n)$, since $r = n^{O(\log \log n)}$. Hence the total randomness used by Verifier II, $\log M + O(\log M + \log \frac{1}{\eta})$ is poly logarithmic in n.

- Verifier III

  Fix $\delta_3 = \delta$ and run Verifier III on one random equation tuple in the set $\mathcal{T}'$. Recall that the set $\mathcal{T}'$ had parameters $(1-\varepsilon, \frac{\delta}{2}, \beta = \frac{1}{18R^4}, C = 100(18)^4 R^{16} \log^{4\kappa} n)$. Observe that the choice of $C = 100(18)^4 R^{16} \log^{4\kappa} n > \frac{64}{\delta^4 \beta^4}$ is sufficiently large to apply Theorem 7.1. From Theorem 7.1, we get the following:

  - Completeness case: there is an assignment that succeeds with probability $1-\varepsilon$.
  - Soundness case: Suppose there is an assignment A on which Verifier III accepts with probability $\frac{1}{2} + \delta$. Then at least $\frac{\delta}{2}$ fraction of the tuples $T \in \mathcal{T}'$ are accepted with probability at least $\frac{1}{2} + \frac{\delta}{2}$. Therefore A is C-close to β-satisfying at least $\frac{\delta}{2}$-fraction of the tuples $T \in \mathcal{T}'$.

  The parameters of Verifier III are given by $C = \log^{36\kappa/\gamma} n$ and $m = O(\log^{2\kappa} n)$. Thus the total randomness used by Verifier III given by $(C-1)\log n + 2\sum \log n_i + m$ is poly logarithmic in n.

As the number of random bits by each of the verifiers is poly logarithmic in n, the entire reduction runs in $DTIME(2^{\log^k n})$ for some k. This concludes the proof of Theorem 8.2. □

Acknowledgments

We thank the anonymous referees for their careful and thorough reading of the paper, and for several suggestions which helped us improve the quality of presentation.

References

[1] Shmuel Agmon. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6(3):382–392, 1954.

[2] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4):567–583, 1986.

[3] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal Computer and System Sciences*, 58(1):137–147, 1999.

[4] Noga Alon and Joel Spencer. *The Probabilistic Method*. John Wiley and Sons, Inc., 1992.

[5] Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 109:237–260, 1998.

[6] Ian Anderson. *Combinatorics of Finite Sets*. Dover Publications Inc., Mineola, NY, 2002.

[7] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[8] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[9] Shai Ben-David, Nadav Eiron, and Philip M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003.

[10] Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1-2):35–52, 1998.

[11] Nader Bshouty and Lynn Burroughs. Maximizing agreements and coagnostic learning. *Theoretical Computer Science*, 350(1):24–39, January 2006.

[12] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 62–68, 2007.

[13] Edith Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *Proceedings of the 38th IEEE Symposium on the Foundations of Computer Science*, pages 514–523, 1997.

[14] Paul Erdos. On a lemma of littlewood and offord. *Bulletin of American Mathematical Society*, 51(12):898–902, 1945.

[15] Uriel Feige and Daniel Reichman. On the hardness of approximating Max-Satisfy. *Information Processing Letters*, 97(1):31–35, 2006.

[16] Vitaly Feldman. Optimal hardness results for maximizing agreements with monomials. In *Proceedings of the 21st Annual IEEE Conference on Computational Complexity*, pages 226–236, 2006.

[17] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 563–574, 2006.

[18] Venkatesan Guruswami and Prasad Raghavendra. A 3-query PCP over integers. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 198–206, 2007.

[19] Magnus Halldorsson. Approximations of weighted indpendent set and hereditary subset problems. *Journal of Graph Algorithms and Applications*, 4(1):1–16, 2000.

[20] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[21] Adam Kalai, Adam Klivans, Yishay Mansour, and Rocco Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 11–20, 2005.

[22] Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. In *Proceedings of the 9th Workshop on Randomization and Computation (RANDOM)*, pages 354–365, 2005.

[23] Michael Kearns, Robert Schapire, and Linda Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.

[24] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1987.

[25] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Learning Theory*. The MIT Press, 1969.

[26] Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999.

[27] Ran Raz. A parallel repetition theorem. *SIAM Journal of Computing*, 27(3):763–803, 1998.

[28] Uri Zwick. Finding almost satisfying assignments. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, pages 551–560, May 1998.

A Linear Systems over Rationals

We now prove that solving linear systems remains hard for sparse systems with bounded coefficients, specifically when the coefficients as well as the number of non-zero coefficients per equation are both bounded by a constant. If a system has coefficients bounded in absolute value by B and each equation involves at most b variables, we say that the system is B-bounded with arity b.

For the sake of convenience, we restate Theorem 5.3 here.

Theorem A.1. *For all constants $\varepsilon, \delta > 0$, there exist $B, b > 0$ such that* LINEQ-MA$(1-\varepsilon, \delta)$ *is NP-hard even on B-bounded systems of arity b.*

Using a different approach, a stronger hardness result has been shown in [18]. Specifically, the above hardness result is extended to linear systems with arity 3, the coefficients of variables in $\{0, 1, -1\}$ and all constants bounded by a function of ε, δ.

We first prove the following Gap-Amplification lemma, that is useful in the course of the proof.

Lemma A.2. *If for some $0 < s < c < 1$, and some constants $T, l > 0$,* LINEQ-MA(c, s) *is NP-hard on T-bounded systems of arity ℓ, then for any positive integer k and constant $\varepsilon > 0$,* LINEQ-MA$(c^k, s^k + \varepsilon)$ *is NP-hard on $T(k/\varepsilon)^k$-bounded systems of arity ℓk.*

Proof: Let $\mathcal{I} = (E, X)$ be an instance of LINEQ-MA with $E = \{E_1, \ldots, E_r\}$ set of equations over variables $X = \{x_1, \ldots, x_n\}$. Each equation is of the form $E_i = 0$.

Define an instance $\mathcal{I}^k = (E^k, X)$ as follows

1. The set of variables is the same, X.
2. For any k-tuple of equations $(E_1, \ldots, E_k)$ introduce the following block of equations,

$$E_1 + y \cdot E_2 + y^2 \cdot E_3 + \ldots + y^{k-1} E_k \quad = \quad 0,$$

for all values of $y = 1, 2, \ldots, t$, where $t = \frac{k-1}{\varepsilon}$.

Clearly, if the original system is T-bounded, then the new system is $T(k/\varepsilon)^k$-bounded. Further the number of nonzero coefficients in each of the equations produced is bounded by ℓk.

Completeness: There is an assignment that satisfies c fraction of the equations E, therefore the same assignment satisfies at least c^k fraction of the new constraints.

Soundness: Suppose there is an assignment A that satisfies more than $s^k + \varepsilon$ fraction of the equations E^k. We claim that A satisfies at least s fraction of the original equations E.

Suppose not, let us say the assignment satisfies a fraction s_1 of the equations in E for some $s_1 < s$. Then s_1^k fraction of the k-tuples have all their equations satisfied. Thus for s_1^k fraction of k-tuples, the block of t equations introduced are all satisfied. For any other k-tuple with not all equations satisfied, at most $k-1$ of the equations in its block can be satisfied. Therefore at most $s_1^k + \frac{k-1}{t}$ fraction of the constraints are satisfied. This is a contradiction since $s_1^k + \varepsilon < s^k + \varepsilon$. $\square$

Proof of Theorem 5.3: We employ a reduction from the LABELCOVER problem. Let (U, V, E, Σ, Π) be an instance of Label Cover with $|\Sigma| = R$. The LINEQ-MA instance that we construct has variables $u_1, \ldots, u_R$ for each vertex $u \in U \cup V$. The solution that we are targeting to obtain is an encoding of the assignment to the label cover instance. So if a vertex u is assigned the label i by an assignment A, then we want

$$\begin{aligned} u_i &= 1, \\ u_j &= 0 \text{ for } j \neq i, 1 \leq j \leq R. \end{aligned}$$

Towards this, we introduce a set of linear combinations of the following equations for each edge $e = (u, v)$.

- $f_0 : \sum_{i=1}^{R} u_i = 1$,
- $f_1 : \sum_{i=1}^{R} v_i = 1$,
- $g_i : \sum_{j \in \pi_e^{-1}(i)} u_j = v_i$ for all $1 \leq i \leq R$.

The set of constraints corresponding to an edge $e = (u, v)$ is given by

$$\mathcal{P}_{e,i} : \quad f_0 + y f_1 + y^2 g_1 + \ldots + y^{R+2} g_R + y^{R+3} u_i = 0 \qquad \text{for all } 1 \leq y \leq t = 10(R+1).$$

Completeness: Given an assignment A to the Label Cover instance that satisfies all the edges, the corresponding integer solution satisfies:

- All equations $f_1, f_2, g_1, \ldots g_R$ for each edge e.
- For a fraction $(1 - \frac{1}{R})$ of the variables u_i, $u_i = 0$. Thus for a fraction $(1 - 1/R)$ of the labels $i \in \Sigma$, all equations in $\mathcal{P}_{e,i}$ are satisfied.

Hence in total, at least $(1 - \frac{1}{R})$ fraction of the equations are satisfied.

Soundness: Let $m = 16R^{1-\gamma}$. Suppose there is an assignment that satisfies $1 - \frac{1}{m}$ fraction of the equations, or equivalently which violates at most $\frac{1}{m}$ of the constraints. For at least half the edges e, at most $\frac{2}{m}$ of the equations corresponding to e are violated. Let us call these edges as *good* edges.

Let $e = (u, v)$ be a *good* edge. We claim that for e all the equations $f_0, f_1, g_1, \ldots, g_R$ are satisfied. If one of the equations $f_0, f_1, g_1, \ldots, g_R$ is not satisfied, then in any $\mathcal{P}_{e,i}$ at most $R + 4$ of the t equations are satisfied. Therefore at most a fraction $\frac{R+4}{t} < 0.5$ of the equations corresponding to e are satisfied. This is a contradiction, since e is a *good* edge. Further, at least $1 - \frac{8}{m}$ fraction of equations of the form $u_i = 0$ are satisfied, because otherwise the total fraction of equations satisfied is less than $(1 - \frac{8}{m}) + \frac{8}{m}\frac{R+4}{t} < 1 - \frac{2}{m}$.

For every vertex u, let $\mathrm{Pos}(u)$ denote the set of labels i such that $u_i > 0$. Formally,

$$\mathrm{Pos}(u) \quad = \quad \{i \in \Sigma \mid u_i > 0\}.$$

For every vertex u with $\mathrm{Pos}(u)$ non-empty, assign a label chosen uniformly at random from $\mathrm{Pos}(u)$. Assign arbitrary labels to the remaining vertices. Observe that if $e = (u, v)$ is a *good* edge, then $\mathrm{Pos}(u)$ and $\mathrm{Pos}(v)$ are both non-empty, because the constraints $\sum_i u_i = 1$ and $\sum_j v_j = 1$ are satisfied. Since at most $\frac{8}{m}$ fraction of the constraints $u_i = 0$ are violated, $|\mathrm{Pos}(u)| \leq R \cdot \frac{8}{m} = \frac{R^\gamma}{2}$. Furthermore, for any choice of the label l_v from $\mathrm{Pos}(v)$, there is some label in $\mathrm{Pos}(u)$ that maps to l_v, because the constraint $\sum_{j \in \pi_e^{-1}(i)} u_j = v_i$ is satisfied for edge e. Therefore the probability that the random assignment satisfies the constraint π_e is satisfied is at least $\frac{1}{|\mathrm{Pos}(u)|} \geq \frac{2}{R^\gamma}$. Since at least half the edges are good, this implies that there is an assignment that satisfies at least a fraction $\frac{1}{2} \cdot \frac{2}{R^\gamma} = \frac{1}{R^\gamma}$ of the edges.

Therefore we have shown that $\textsf{LINEQ-MA}(1 - \frac{1}{R}, 1 - \frac{1}{16R^{1-\gamma}})$ is NP-hard on $(10(R+1))^{R+3}$-bounded systems with arity $10R(R+1)$, for all large enough R. Now we use the gap amplification Lemma A.2 with $k = O(R^{1-\gamma})$ to obtain a gap of $1 - \varepsilon, \delta$ for any small ε, δ on B-bounded systems with arity b where B, b are constants depending on ε, δ.