Electronic Colloquium on Computational Complexity, Report No. 61 (2006)

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 [7], and in some more recent works as well. Along the way, we also obtain a strong hardness for another basic computational problem: solving a linear system over the rationals.

*Research supported in part by NSF Award CCF-0343672 and a Sloan Research Fellowship.
†Research supported in part by NSF CCF-0343672

ISSN 1433-8092

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 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 thus the following: If no perfect solution exists, but say a solution satisfying 99% 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 easier to solve on near-satisfiable instances — notable examples being the Max 2SAT and Max HornSAT problems. For both of these it is possible to find, in polynomial time, an assignment satisfying a fraction $1 - f(\varepsilon)$ of the clauses $f(\varepsilon) \to 0$ as $\varepsilon \to 0$ given a $(1 - \varepsilon)$-satisfiable instance [20]. 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 [13] 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 [10] 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 an open question in [10], and this was also posed by Blum, Frieze, Kannan, and Vempala [7] 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.

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 [13] 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 [4, 9]. In [9], 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 [12], 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 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 [4] (see also [3]). In particular, for arbitrarily large constants c, the reduction of Arora *et al* [4] 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 $\gamma = \Omega(1)$. As presented in [4], however, their 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 $w_1, \ldots, w_2, \theta$ (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, 18] or the Winnow algorithm [17] 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 [7] 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 [7] outputs as hypothesis a decision list of halfspaces. Later, Cohen [8] 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 *at al* [16], 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 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 [14], 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. 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 [3, 6] (the factor was $261/262 + \varepsilon$ in [3] and $415/418 + \varepsilon$ in [6]). The best known hardness result prior to work was due to Bshouty and Burroughs, who showed an inapproximability factor of $84/85 + \varepsilon$, and their result applied also 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 [4, 3]. 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.

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 [7]) or about the distribution of the example-label pairs (such as uniform marginal distribution on examples as in [14]).

A similar hardness result was proved independently by Feldman *et al* [11] 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 $0 \le s \le c \le 1$, 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 Q, distinguish between the following two cases:*

- *There is an assignment of values to the variables X, that satisfies more than a c fraction of the equations.*
- *No assignment satisfies more than s fraction 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 some 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 $0 \leq s < c \leq 1$, 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 at most $s|S^+ \cup S^-|$ with (S^+, S^-).*

The hardness results in this paper are obtained by reductions from the Label Cover problem, which can be defined as follows:

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. *[19, 5] There exists an absolute constant $\gamma > 0$ such that for all large enough R, the gap problem $LABELCOVER(1, \frac{1}{R^\gamma})$ is NP-hard, where $R = |\Sigma|$ is the size of the alphabet.*

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. 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 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 $V(E_i) 1 \leq i \leq k$ and $V(E)$ are all pairwise disjoint. An equation tuple is said to be of constant arity, if the arity of each of its equations and the scaling factor are bounded by a constant.*

Definition 2.8. *An assignment A is said to satisfy an equation tuple T, if for every E_i $1 \leq i \leq k$, $E_i(A) = 0$ and the scaling factor $E(A) > 0$. An assignment A is said to β-satisfy an equation tuple T if for each $1 \leq i \leq k$,$|E_i(A)| < \beta \cdot |E(A)|$ and moreover $E(A) > 0$.*

Definition 2.9. *An assignment A is said to be C-far from β-satisfying an equation tuple T, if for some C distinct equations $E_{a_1}, \ldots, E_{a_C}$ in the tuple T, we have $|E_{a_i}(A)| \geq \beta \cdot |E(A)|$.*

3 Overview of the Proof

Both the hardness results use a reduction from the Label Cover problem. The proof of hardness of HS-MA proceeds in three stages as described below.

In the first stage the label cover problem is reduced to a set of equation tuples $\mathcal{T}$ using Verifier I such that for a NO instance of label cover, any assignment A can β-satisfy a very tiny fraction of tuples in $\mathcal{T}$. However the tuples $T \in \mathcal{T}$ are not disjoint.

In the second stage, 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 ε-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 ε-satisfy most of the tuples, but is in fact C-far from ε-satisfying most of the tuples. Both these facts are exploited by Verifier III in the third stage.

Verifier III checks if an assignment A is C-close to satisfying a tuple T by checking inequalities. The inequalities are based on a random linear combination of the equations of the tuple with ± 1 coefficients (the random choice is made from a small sample space of vectors with ± 1 coefficients). In the completeness analysis, if all equations are satisfied, i.e., evaluate to 0 on A, then any ± 1 combination also vanishes. In the soundness analysis, most tuples have at least C equations with non-trivial absolute value, and this implies that their linear combination is unlikely to be small (a careful choice of the sample space of linear combinations is crucial to conclude this).

Each of the inequalities checked by Verifier III has all the variables with coefficients $\{-1, 1\}$, and has a common variable (a threshold θ) on the right hand side. Hence the checks made by the combined verifier correspond naturally to training examples in the learning problem.

For the hardness of 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.

4 Verifier I

Let (U, V, E, Σ, Π) be an instance of Label Cover with $|\Sigma| = R$. This verifier produces a set of equation tuples, which are tested using Verifier II. The equation tuples have variables $u_1, \ldots, u_R$ for each vertex $u \in U \cup V$. 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 $j \neq i, 0 \leq j \leq R$. 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.

For each t-tuple X of variables corresponding to vertices in U, construct the equation tuple T as follows.

- $\mathcal{P}_1$: For every pair of vertices $u, v \in U \cup V$, an equation
$$\sum_{i=1}^{R} u_i - \sum_{j=1}^{R} v_i = 0$$
- $\mathcal{P}_2$:For each edge $e = (u, v) \in E$ the label cover constraint for the edge
$$\sum_{j \in \pi_e^{-1}(i)} u_j - v_i = 0 \text{ for all } 1 \le i \le R$$
- $\mathcal{P}_3$:For each variable $v \in X$,
$$v = 0$$
- The scaling factor is $\mathcal{P}_4$: $\sum_{i=1}^{R} u_i$ for some 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 $\delta_1, \varepsilon_1 > 0$ there exists a sufficiently large $R = R(\varepsilon_1, \delta_1)$ such that if $\Gamma = (U, V, E, \Sigma, \Pi)$ is an instance of label cover with $|\Sigma| = R$ then with the choice of $\beta' = \frac{1}{R^3}$ 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'-satisfies less than a fraction δ_1 of the output tuples.*

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

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

- All equations in $\mathcal{P}_1$ and $\mathcal{P}_2$.
- $(1 - \frac{1}{R})$ fraction of the equations in $\mathcal{P}_3$ for each edge e.

Since t equations of the form $\mathcal{P}_3$ are present in each tuple, the assignment A satisfies at least $(1 - \frac{1}{R})^t > 1 - \varepsilon_1$ of the tuples for large enough R.

Soundness: Suppose there is an assignment A that β'-satisfies at least a fraction δ_1 of the tuples generated. Clearly A must β'-satisfy all the equations $\mathcal{P}_1$ and $\mathcal{P}_2$, since they are common to all the tuples. Further by definition of β'-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'-satisfied, we get

$$1 - \beta' < \sum_{i=1}^{R} v_i < 1 + \beta', \text{ for all } v \in U \cup V \tag{1}$$

Further, we claim that the assignment A β'-satisfies at least a fraction $(1-\frac{c_0}{t})$ of the equations in $\mathcal{P}_3$. Otherwise, with t of these equations belonging to every tuple, less than $(1-\frac{c_0}{t})^t < \delta_1$ tuples will be β'-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 $(1-\frac{2c_0}{t})$ of the constraints $u_i = 0$ are β'-satisfied. Let us call these edges *good* Edges.

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

$$\begin{aligned}\mathrm{Pos}(u) &= \{i \in \Sigma \mid u_i > 8\beta'\} \text{ if } u \in U \\ \mathrm{Pos}(v) &= \{j \in \Sigma \mid v_j > 8\beta'(R+1)\} \text{ if } v \in V\end{aligned}$$

The set $\mathrm{Pos}(w)$ is non-empty for each vertex $w \in U \cup V$, because otherwise $\sum_{i=1}^{R} w_i \leq 8\beta'(R+1)\cdot R \leq 1-\beta'$, a contradiction to (1). Further if $e=(u,v)$ is a *good* edge then for at least $1-\frac{2c_0}{t}$ of the labels $1 \leq i \leq R$, we have $u_i < \beta'$. Hence $|\mathrm{Pos}(u)| \leq (\frac{2c_0}{t})R = \frac{R^\gamma}{2}$. Further, since all the constraints $\mathcal{P}_2$ are β'-satisfied, we know that

$$\left|\sum_{i\in\pi_e^{-1}(j)} u_i - v_j\right| < \beta'$$

Thus for every label, $j \in \mathrm{Pos}(v)$, there is at least one label $i \in \mathrm{Pos}(u)$ such that $\pi_e(i) = j$. For every vertex $w \in U \cup V$, assign a label chosen uniformly at random from $\mathrm{Pos}(w)$. For any *good* edge $e=(u,v)$, the probability that the constraint π_e is satisfied is at least $\frac{1}{|\mathrm{Pos}(u)|} \geq \frac{2}{R^\gamma}$. 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: Given a label cover instance Π with alphabet size R, the reduction outlined in Theorem 4.1 is applied 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:

$$E_1 + y\cdot E_2 + y^2\cdot C_3 + \ldots + y^{n-1}E_n + y^n(E-1) = 0$$

for all values of $y = 1, 2, \ldots, t$, where $t = \frac{n+1}{R^\gamma}$.

Completeness: Observe that if Π 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, $E_i(A) = 0, 1 \leq i \leq n$. Hence A satisfies at least $1-\frac{1}{R}$ fraction of the equations.

Soundness: Suppose there is an assignment that satisfies A that satisfies more than $\frac{2}{R^\gamma}$ fraction of the equations. Hence for at least $\frac{1}{R^\gamma}$ fraction of the tuples, at least $\frac{1}{R^\gamma}$ fraction of the equations are satisfied. Let us refer to these tuples as *nice.* If a *nice* tuple T, is not satisfied by A, then at most $\frac{n+1}{t} < \frac{1}{R^\gamma}$ fraction of the equations corresponding to T can be satisfied. Hence every *nice* tuple T is satisfied by A. So the assignment A satisfies at least $\frac{1}{R^\gamma}$ of the tuples, which is a contradiction to Theorem 4.1.

For a sufficiently large R, we have $\frac{1}{R} < \varepsilon$ and $\frac{2}{R^\gamma} < \delta$, and hence the result follows.

□

The coefficients of variables in the above reduction could be exponential in n (their binary representation could use polynomially many bits). In Appendix A, we discuss an alternate reduction which yields the same hardness with coefficients bounded by a constant depending only on ε, δ, and moreover the arity of all the equations is also bounded by a constant.

6 Verifier II

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

The equation tuple T that needs to be tested, may not be disjoint. i.e there could be a variable that occurs 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. 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, and 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 a 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.

The verifier makes use of sets of k-wise η-dependent permutations. A set of k-wise η-dependent permutations is defined as follows:

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 atmost η*

Definition 6.2. *A family of permutations Π (can have repetitions) 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.*

Polynomial size constructions of such permutations have been presented in [15].

Let us say the tuple T consists of equations $E_1, \dots, E_m$ and a scaling factor E over variables $u_1, \dots, u_n$. Let us denote by n_0 the maximum arity of 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 even constants whose values will be chosen later. The set of variables used by Verifier II consists of

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

Let Π denote a set of C_1-wise almost independent (η-dependent) permutations of $\{1, \dots, M\}$ for some constants C_1, η, which we will choose later.

- Pick an equation tuple $T \in \mathcal{T}$ uniformly at random.
- Pick a number k uniformly at random from $\{1, \ldots, M+1\}$. Choose E^k as the scaling factor. Re-number the remaining M copies of $V(E)$, with $\{1, \ldots, M\}$.
- Choose a permutation π uniformly at random from the set Π of C_1-wise η-dependent permutations. Construct sets of equations $\mathcal{P}$ and $\mathcal{M}$ as follows

  For each $E_i \in \mathcal{T}$

  $$\mathcal{P} = \{E_i^{\pi(j)} | 1 \leq i \leq m, (P-1)i + 1 \leq j \leq Pi\}$$

  $$\mathcal{M} = \{M_i(\pi(j), \pi(j+1)) | u_i^{\pi(j)} \notin V(\mathcal{P}), j : odd\}$$
- Output the tuple $(\mathcal{P} \cup \mathcal{M}, E^k)$.

Theorem 6.3. *For all $\varepsilon_2, \delta_2 > 0$ and a positive integer C, there exists constant parameter choices M, P, C_1, η for Verifier II such that: Given a set of equation tuples $\mathcal{T}$ of which each tuple is of constant arity(n_0) and has the same scaling factor E, the following is true*

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

Proof: 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 δ_2-fraction of the output tuples. Then for at least $\frac{\delta_2}{2}$ choices of input tuple $T \in \mathcal{T}$, at least $\frac{\delta_2}{2}$ fraction of the output tuples are C-close to being β-satisfied. Let us call these input tuples T to be *good*. For a good tuple T, there are at least $\frac{\delta_2}{4}$ fraction of choices of k for which with probability more than $\frac{\delta_2}{4}$, the output tuple is C-close to being β-satisfied. These values of k are said to be *nice* with respect to T.

Lemma 6.4. *Let E^k be a scaling factor that is nice with respect to some good tuple T. For every variable u_i, all but a constant number C_0 of copies of u_i are $2\beta|E^k(A')|$ close to each other.*

$$|A'(u_i^{j_1}) - A'(u_i^{j_2})| < 2\beta|E^k(A')|$$

Proof: As E^k is a *nice* scaling factor for T, for at least $\frac{\delta_2}{4}$ choices of $\pi \in \Pi$ the assignment A' is less than C-far from β-satisfying the 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 $\mathcal{M}$ are β-violated.

Define a copy of u_i^j to be bad, if it is $\beta|E^k(A')|$ far from more than half the other copies. i.e $|A'(u_i^j) - A'(u_i^{j_1})| > \beta|E^k(A')|$ for half the values of j_1. Suppose there are more than C_0 bad copies of the variable u_i. Without loss of generality we can assume that the first C_0 copies $\{u_i^1, u_i^2, \ldots, u_i^{C_0}\}$ are bad. The probability that the permutation π maps some two of these copies to consecutive locations is at most $\binom{C_0}{2}\frac{C_0}{M}$.

Further the probability that one of these C_0 copies, is used for an equation in $\mathcal{P}$ is at most $C_0 \cdot \frac{Pm}{M}$. Now consider the case in which each of these copies is checked for consistency with some other copy of the variable.

A *bad* copy $u_i^j, 1 \le j \le C_0$ creates a β-violation in $\mathcal{M}$, whenever a distant copy is mapped next to it. Therefore with probability at least $\frac{1}{2}$, a bad copy produces a β-violation. Even if the *bad* copies share many of the distant neighbors, since $M >> C_0$, the probability that a *bad* copy produces a violation is at least $\frac{2}{3}$. Since Π is a set of $C_1 > 2C_0$-wise almost independent (η-dependent) permutations, the probability that there are less than C violations in $\mathcal{M}$ is at most $\binom{C_0}{C}(\frac{2}{3})^{(C_0-C)} + 2\eta$

Therefore,

$$\mathbf{Pr}[\text{Verifier II accepts}] \le \binom{C_0}{2}\frac{C_0}{M} + C_0 \cdot \frac{Pm}{M} + \binom{C_0}{C}(\frac{2}{3})^{(C_0-C)} + 2\eta < \frac{\delta_2}{4}$$

for $M > \max\left(\frac{40PmC_0}{\delta_2}, \frac{40C_0^3}{\delta_2}\right)$ and sufficiently large constant $C_0, \frac{1}{\eta}$. □

Lemma 6.5. *Given a nice scaling factor E^k of T, and an equation $E_i \in T$, there exists at least $P - C_0$ values of j for which $|E_i^j(A')| < \beta|E^k(A')|$*

Proof: Since E^k is a *nice* scaling factor, at least for one permutation $\pi \in \Pi$, the tuple generated is less than C-far from being β satisfied. Since each equation E_i is checked on P different copies, at least $P - C$ of the copies must β-satisfy E_i. □

Let T, be a *good* tuple. Define k_0 to be its *nice* value for which the corresponding scaling factor $E^k(A')$ has the smallest absolute value. From Lemma 6.4, we know that all but a constant C_0 of the copies of every variable are $2\beta|E^{k_0}(A')|$ close to each other. Delete all the bad copies(at most C_0) of each variable. Further, delete all the variables in $V(E^{k_0})$. 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'-satisfies all the *good* tuples $T' \in \mathcal{T}$.

Observe that the arity of E^k is at most n_0, and at most $C_0 + 1$ copies of each variable are deleted. Since $\frac{\delta_2}{4}M > n_0(C_0 + 1)$, there exists a *nice* scaling factor E^{k_1} of T such that no variable of $V(E^{k_1})$ is deleted. Further by definition of k_0, $|E^{k_1}(A')| \ge |E^{k_0}(A')|$.

¿From Lemma 6.4, we know that for the average assignment A, and any variable u_i,

$$|A(u_i) - u_i^j| < 2\beta|E^{k_0}(A')| \le 2\beta|E^{k_1}(A')| \tag{2}$$

Using the above equation for the variables in $V(E^{k_1})$, we get

$$(1 - 2\beta n_0)|E^{k_1}(A')| < |E(A)|$$

Substituting back in 2, we get

$$|A(u_i) - u_i^j| < \frac{2\beta}{(1 - 2\beta n_0)}|E(A)| < 4\beta|E(A)| \tag{3}$$

Consider any *good* tuple $T' \in \mathcal{T}$. By the same argument as T, it can be shown that there exists a nice scaling factor E^{j_0}, none of whose variables have been deleted. Hence we have

$$|E^{j_0}(A')| < |E(A)| + 4\beta \cdot n_0|E(A)|$$

Using Lemma 6.5, and the fact $P - C > n_0C_0$, we can conclude for every equation $E_i \in T'$, there exists j_1 such that $|E_i^{j_1}(A')| < \beta|E^{j_0}(A')|$, and no variable of $V(E_i^{j_1})$ is deleted. Using equation 3 with variables in $V(E_i^{j_0})$, we get

$$|E_i(A)| < |E_i^{j_1}(A')| + 4\beta \cdot n_0|E(A)|$$

Therefore,

$$|E_i(A)| < (\beta + 4\beta^2 n_0 + 4\beta n_0)|E(A)| < 9\beta n_0|E(A)| = \beta'|E(A)|$$

Thus the assignment A β'-satisfies the tuple T'. Hence the assignment A β'-satisfies all the *good* tuples, and since at least a fraction $\delta_2/2$ of the tuples are good, the result follows. □

7 Verifier III

Given a equation tuple $T = (\{E_1, \ldots, E_n\}; E)$, Verifier III checks if the assignment A is 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 [2, Chap. 15]. 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 partitions are $\mathcal{E}_i, 1 \le i \le m$.
- For each partition $\mathcal{E}_i$, pick a random vector, $v_i \in V_{n_i}$ where $n_i = |\mathcal{E}_i|$. Compute linear functions $B_i, 1 \le i \le m$
$$B_i = \mathcal{E}_i \cdot v_i$$
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 if either of the following inequalities is satisfied by A:
$$B \cdot w + E \ \ge \ \theta \tag{4}$$
$$B \cdot w - E \ < \ \theta \tag{5}$$
Accept if the check is satisfied, else Reject.

Polynomial size spaces for C-wise independent variables taking values $\{1, \ldots, m\}$, can be obtained using BCH codes with alphabet size m, and minimum distance $C + 1$.

Theorem 7.1. *For every $\beta, \delta_3 > 0$ there exists constants $C = C(\beta, \delta_3), m$ such that 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: For an assignment A that satisfies the tuple T, we have $E_j(A) = 0, 1 \le j \le n$, and $E(A) > 0$. Hence for all the random choices, $B = 0$, and $E > 0$. Therefore, with the choice $\theta = 0$, all the checks made by the verifier succeed.

Suppose the assignment A is C-far from β-satisfying the tuple T. If $E(A) \le 0$, then clearly at most one of these two inequalities 4 can satisfied, and the proof is complete. Hence, we assume $E(A) > 0$.

This implies that at least C of the values $\{E_j(A) | 1 \le j \le n\}$ have absolute value greater than $\beta|E(A)|$. Let us refer to these E_j as *large*. The probability that one of the partitions $\mathcal{E}_i$ contains less than $C_0 = \frac{2}{\beta^2}$ *large* values is at most $m\binom{C}{C_0}(1 - \frac{1}{m})^{C - C_0}$. From Lemma 7.2, for a partition $\mathcal{E}_i$ that has at least C_0 *large* values,
$$Pr[|B_i(A)| > |E(A)|] \ge \frac{1}{12}$$

Assuming, that all the partitions have at least C_0 *large* values, we bound the probability that less than $\frac{m}{24}$ partitions have $|B_i(A)| > |E(A)|$. Towards this, we use the Chernoff bounds, to obtain

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

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

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

Overall we have,

$$\mathbf{Pr}(B \cdot w \in [\theta - E(A), \theta + E(A)]) \leq m\binom{C}{C_0}(1 - \frac{1}{m})^{C - C_0} + e^{-\frac{m}{96}} + \frac{\binom{m_0}{m_0/2}}{2^{m_0 - 1}}$$

The value of $C_0 = \frac{1}{2\beta^2}$ is fixed, so for large enough values of C, m with $C > m$ the above probability is less than $\delta_3/2$. Observe that if $B \cdot w \notin [\theta - E(A), \theta + E(A)]$, 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 $\frac{1}{2} + \delta_3/2$. $\square$

Lemma 7.2. *For all $\beta > 0$, and a constant $C_0 \geq \frac{2}{\beta^2}$, 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 C_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: 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 C_0 components of a are larger than β, we have $\|a\|_2^2 > C_0\beta^2 \geq 2$. 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. $\square$

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 $\theta \in \mathbb{R}$,*

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

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

Proof: Without loss of generality, we can assume that $a_i \geq 1$ for $1 \leq i \leq K$. For a vector $v \in \{-1, 1\}^m$, we write $v = v_{|K} \circ v_{|m-K}$ where $v_{|K} \in \{-1, 1\}^K, v_{|m-K} \in \{-1, 1\}^{m-K}$ and $\circ$ denotes the concatenation of the two vectors. Denote by **-1**, and **1** the K dimensional vectors consisting of all -1s and all 1s respectively. Consider a path $\mathcal{P}$ on the hypercube, starting at $u_0 = \mathbf{-1} \circ v_{m-K}$ and reaching $u_K = \mathbf{1} \circ v_{m-K}$ by changing one variable from -1 to 1 at each step. If u_i, u_{i+1} are the i^{th} and $i+1^{th}$ nodes on the path $\mathcal{P}$, then we know

$$a \cdot u_{i+1} - a \cdot u_i = a_i > 1$$

Therefore, at most two points on the path $\mathcal{P}$ can belong to an interval $[\theta - 1, \theta + 1]$. In total there are $K!$ paths $\mathcal{P}$ from u_0 to u_K. Further any vector v' of the form $v' = v'_{|K} \circ v_{|m-K}$ is present on at least $\frac{K}{2}!\frac{K}{2}!$ different paths. Hence, we can conclude

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

□

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. Using the set of equation tuples $\mathcal{T}$ as input, Verifier II with parameters $\varepsilon_2 = \varepsilon_1, \delta_2 = \frac{\delta}{2}, \beta' = \frac{1}{R^3}$, 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 atleast $\frac{1}{2} + \frac{\delta}{2}$. 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'- 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 $R^\gamma \log n + \log M + C_1 \log M + \log \frac{1}{\eta} = O(\log n)$ 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, and using almost independent sets of random variables, the following stronger hardness result can be shown

Theorem 8.2. *For all $c > 0$, there exists a constant $\gamma > 0$, such that the problem* HS-MA$(1 - \frac{1}{2^{(\log n)^\gamma}}, \frac{1}{2} + \frac{1}{(\log n)^c})$ *is Quasi-NP-hard.*

References

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

[2] N. Alon and J. Spencer. *The Probabilistic Method.* John Wiley and Sons, Inc., 1992.

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

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

[5] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[6] S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. In *Proceedings of the 13th COLT*, pages 266–274, 1992.

[7] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *Proceedings of the 37th IEEE Symposium on the Foundations of Computer Science*, 1996.

[8] E. 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.

[9] U. Feige and D. Reichman. On the hardness of approximating Max-Satisfy. *Electronic Colloquium on Computational Complexity (ECCC), TR04-119*, 2004.

[10] V. Feldman. Optimal hardness results for maximizing agreements with monomials. *Electronic Colloquium on Computational Complexity, TR06-032*, 2006. To appear in 21st Annual IEEE Computational Complexity Conference (CCC), 2006.

[11] V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. New results for learning noisy parities and halfspaces. *ECCC Technical Report TR06-059*, 2006.

[12] M. Halldorsson. Approximations of weighted indpendent set and hereditary subset problems. *J. Gaph Algorithms Appl.*, 4(1), 2000.

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

[14] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 11–20, 2005.

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

[16] M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.

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

[18] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Learning Theory.* The MIT Press, 1969.

[19] R. Raz. A parallel repetition theorem. In *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing (STOC '95)*, pages 447–456, New York, May 1995. ACM.

[20] U. Zwick. Finding almost satisfying assignments. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC)*, 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.

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

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 $T, l > 0$, $\mathsf{LINEQ\text{-}MA}(c, s)$ is NP-hard on T-bounded systems of arity ℓ, then for any positive integer k and constant $\varepsilon > 0$, $\mathsf{LINEQ\text{-}MA}(s^k + \varepsilon, c^k)$ is NP-hard on $T(k/\varepsilon)^k$-bounded systems of arity ℓk.*

Proof: Let $\mathcal{I} = (E, X)$ be an instance of $\mathsf{LINEQ\text{-}MA}$ 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. Also, clearly the number of nonzero coefficients in each of the new equations 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 they satisfy a fraction s_1 of the equations for some $s_1 < s$. Then s_1^k of the k-tuples have all their equations satisfied. So for s_1^k tuples, the block of t equations are satisfied. For any 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 $s_1^k + \varepsilon < s^k + \varepsilon$. □

Proof of Theorem A.1: 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

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

Towards this, the following equations are introduced:
For each edge $e = (u, v)$ we introduce a block of linear combinations of the following equations.

- $f_0 : \sum_{i=1}^{R} u_i = 1$
- $f_1 : \sum_{i=1}^{R} v_i = 1$
- $g_i : \sum_{j \in \pi_e^{-1}(i)} u_j = v_i$ 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 $f_1, f_2, g_i, 1 \leq i \leq R$ for each edge e.
- A fraction $(1 - \frac{1}{R})$ of the equations $\{u_i = 0 \mid i \in \Sigma\}$ for each edge e.

So in total atleast $(1 - \frac{1}{R})$ 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 $\mathcal{P}_{e,i}$ are violated. Let us call these edges as *good* edges.

Let $e = (u, v)$ be a *good* edge. Observe 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 at most $R+4$ of the t equations are satisfied. Therefore at most a fraction $\frac{R+4}{t} < 0.5$ of the equations in $\mathcal{P}_{e,i}$ are satisfied. This is a contradiction, since e is a *good* edge. Further, at least $1 - \frac{8}{m}$ 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 $\text{Pos}(u)$ denote the set of labels i such that $u_i > 0$,

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

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 $\frac{8}{m}$ of the constraints $u_i = 0$ are violated, $|\mathrm{Pos}(u)| \leq R \cdot \frac{8}{m} = 2R^\gamma$. 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 $\mathsf{LINEQ\text{-}MA}(1 - \frac{1}{R}, 1 - \frac{1}{16R^{1-\gamma}})$ is NP-hard on $(10(R+1))^{R+3}$-bounded systems with arity $10R(R+1)$, for all large 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.

ECCC ISSN 1433-8092
http://eccc.hpi-web.de/