On Agnostic Learning of Parities, Monomials and Halfspaces

Vitaly Feldman*
Harvard University.
vitaly@eecs.harvard.edu

Parikshit Gopalan
Georgia Tech.
parik@cc.gatech.edu

Subhash Khot
Georgia Tech.
khot@cc.gatech.edu

Ashok Kumar Ponnuswami
Georgia Tech.
pashok@cc.gatech.edu

Abstract

We study the learnability of several fundamental concept classes in the agnostic learning framework of Haussler [Hau92] and Kearns *et al.* [KSS94].

We show that under the uniform distribution, agnostically learning parities reduces to learning parities with random classification noise, commonly referred to as the noisy parity problem. Together with the parity learning algorithm of Blum *et al.* [BKW03], this gives the first nontrivial algorithm for agnostic learning of parities. We use similar techniques to reduce learning of two other fundamental concept classes under the uniform distribution to learning of noisy parities. Namely, we show that learning of DNF expressions reduces to learning noisy parities of just logarithmic number of variables and learning of k-juntas reduces to learning noisy parities of k variables.

We give essentially optimal hardness results for agnostic learning of monomials over $\{0,1\}^n$ and halfspaces over $\mathbb{Q}^n$. We show that for any constant ϵ finding a monomial (halfspace) that agrees with an unknown function on $1/2+\epsilon$ fraction of examples is NP-hard even when there exists a monomial (halfspace) that agrees with the unknown function on $1-\epsilon$ fraction of examples. This resolves an open question due to Blum and significantly improves on a number of previous hardness results for these problems. We extend these result to $\epsilon = 2^{-\log^{1-\lambda} n}$ for any constant $\lambda > 0$ under stronger complexity assumptions.

[0]Preliminary versions of the results in this paper appeared in [Fel06b] and [FGKP06].

*Supported by grants from the National Science Foundation NSF-CCR-0310882, NSF-CCF-0432037, and NSF-CCF-0427129.

1 Introduction

Parities, monomials, and halfspaces are among the most fundamental concept classes in learning theory. Each of these concept classes is long-known to be learnable when examples given to the learning algorithm are guaranteed to be consistent with a function from the concept class [Val84, BEHW87, Lit88]. Real data is rarely completely consistent with a simple concept and therefore this strong assumption is a significant limitation of learning algorithms in Valiant's PAC learning model [Val84]. A general way to address this limitation was suggested by Haussler [Hau92] and Kearns *et al.* [KSS94] who introduced the *agnostic* learning model. In this model, informally, nothing is known about the process that generated the examples and the learning algorithm is required to do nearly as well as is possible using hypotheses from a given class. This corresponds to a common empirical approach when few or no assumptions are made on the data and a fixed space of hypotheses is searched to find the "best" approximation of the unknown function.

This model can also be thought of as a model of *adversarial classification noise* by viewing the data as coming from $f^* \in \mathcal{C}$ but with labels corrupted on an η^* fraction of examples (f^* is the function in $\mathcal{C}$ that has the minimum error η^*). Note however, that unlike in most other models of noise the learning algorithm is not required to recover the corrupted labels but only to classify correctly "almost" (in the PAC sense) $1-\eta^*$ fraction of examples.

Designing algorithms that learn in this model is notoriously hard and very few positive results are known [KSS94, LBW95, GKS01, KKMS05]. In this work we give the first non-trivial positive result for learning of parities and strong hardness results for learning monomials and halfspaces in this model. Our results apply to the standard agnostic learning model in which the learning algorithm outputs a hypothesis from the same class as the class against which its performance is measured. By analogy to learning in the PAC model this restrictions is often referred to as *proper* agnostic learning.

1.1 Learning Parities Under the Uniform Distribution

A parity function is the XOR of some set of variables $T \subseteq [n]$, where $[n]$ denotes the set $\{1, 2, \ldots, n\}$. In the absence of noise, one can identify the set T by running Gaussian elimination on the given examples. The presence of noise in the labels, however, leads to a number of challenging and important problems. We address learning of parities in the presence of two types of noise: *random classification noise* (each label is flipped with some fixed probability η randomly and independently) and adversarial classification noise (that is the agnostic learning). When learning with respect to the uniform distribution these problems are equivalent to decoding of random linear binary codes (from random and adversarial errors, respectively) both of which are long-standing open problems in coding theory [BMvT78, McE78, BFKL93]. Below we summarize the known results about these problems.

- **Adversarial Noise:** Without any restrictions on the distribution of examples the problem of (proper) agnostic learning parities is known to be NP-hard. This follows easily from a celebrated result of Håstad [Has01]. We are unaware of non-trivial algorithms for this problem under any fixed distribution, prior to our work. The problem of learning parities with adversarial noise under the uniform distribution is equivalent to finding a significant Fourier coefficient of a Boolean function and related to the problem of decoding Hadamard codes. If the learner can ask *membership queries* (or queries that allow the learner to get the value of function f at any point), a celebrated result of Goldreich and Levin gives a polynomial time algorithm for this problem [GL89]. Later algorithms were given by Kushilevitz and Mansour [KM91], Levin [Lev93], Bshouty *et al.* [BJT04], and Feldman [Fel06a].

- **Random Noise:** The problem of learning parities in the presence of random noise, or the noisy parity problem is a notorious open problem in computational learning theory. Blum, Kalai and Wasserman give

an algorithm for learning parity functions on n variables in the presence of random noise in time $2^{O(\frac{n}{\log n})}$ for any constant η [BKW03]. Their algorithm works for any distribution of examples. We will also consider a natural restriction of this problem in which the set T is of size at most k. The only known algorithm for this problem is the brute-force search in which one takes $O(\frac{1}{1-2\eta}k \log n)$ samples and then finds the parity on k variables that best fits the data through exhaustive search in time $O(n^k)$.

In this work, we focus on learning parities under the uniform distribution. We reduce a number of fundamental open problems on learning under the uniform distribution to learning noisy parities, establishing the central role of noisy parities in this model of learning.

Learning Parities with Adversarial Noise

We show that under the uniform distribution, learning parities with adversarial noise reduces to learning parities with random noise. In particular, our reduction and the result of Blum *et al.* imply the first non-trivial algorithm for learning parities with adversarial noise under the uniform distribution.

Theorem 1 *For any constant $\eta < 1/2$, parities are learnable under the uniform distribution with adversarial noise of rate η in time $O(2^{\frac{n}{\log n}})$.*

Equivalently, this gives the first non-trivial algorithm for agnostically learning parities. The restriction on the noise rate in the algorithm of Blum *et al.* translates into a restriction on the optimal agreement rate of the unknown function with a parity (namely it has to be a constant greater than $1/2$). Hence in this case the adversarial noise formulation is cleaner.

Our main technical contribution is to show that an algorithm for learning noisy parities gives an algorithm that finds significant Fourier coefficients (i.e. correlated parities) of a function from random samples. Thus an algorithm for learning noisy parities gives an analogue of the Goldreich-Levin/Kushilevitz-Mansour algorithm for the uniform distribution, but without membership queries. This result is proved using Fourier analysis.

Learning DNF formulae

Learning of DNF expressions from random examples is a famous open problem originating from Valiant's seminal paper on PAC learning [Val84]. In this problem we are given access to examples of a Boolean function f on points randomly chosen with respect to distribution $\mathcal{D}$, and $\epsilon > 0$. The goal is to find a hypothesis that ϵ-approximates f with respect to $\mathcal{D}$ in time polynomial in n, $s = \texttt{DNF-size}(f)$ and $1/\epsilon$, where $\texttt{DNF-size}(f)$ is the number of terms in the DNF formula for f with the minimum number of terms. The best known algorithm for learning DNF in this model was given by Klivans and Servedio [KS01] and runs in time $2^{\tilde{O}(n^{1/3})}$.

For learning DNF under the uniform distribution a simple quasipolynomial algorithm was given by Verbeurgt [Ver90]. His algorithm essentially collects all the terms of size $\log(s/\epsilon) + O(1)$ that are consistent with the target function, i.e. do not accept negative points and runs in time $O(n^{\log(s/\epsilon)})$. We are unaware of an algorithm improving on this approach. Jackson [Jac97] proved that DNFs are learnable under the uniform distribution if the learning algorithm is allowed to ask membership queries. This breakthrough and influential result gives essentially the only known approach to learning of unrestricted DNFs in polynomial time.

We show that learning of DNF expressions reduces to learning parities of $O(\log(s/\epsilon))$ variables with noise rate $\eta = 1/2 - \tilde{O}(\epsilon/s)$ under the uniform distribution.

Theorem 2 *Let $\mathcal{A}$ be an algorithm that learns parities of k variables over $\{0,1\}^n$ for every noise rate $\eta < 1/2$ in time $T(n, k, \frac{1}{1-2\eta})$ using at most $S(n, k, \frac{1}{1-2\eta})$ examples. Then there exists an algorithm that learns DNF expressions of size s in time $\tilde{O}(\frac{s^4}{\epsilon^2} \cdot T(n, \log B, B) \cdot S(n, \log B, B)^2)$, where $B = \tilde{O}(s/\epsilon)$.*

Learning k-juntas

A Boolean function is a k-junta if it depends only on k variables out of n. Learning of k-juntas was proposed by Blum and Langley [BL97, Blu94], as a clean formulation of the problem of efficient learning in the presence of irrelevant features. Moreover, for $k = O(\log n)$, a k-junta is a special case of a polynomial-size decision tree or a DNF expression. Thus, learning juntas is a first step toward learning polynomial-size decision trees and DNFs under the uniform distribution. A brute force approach to this problem would be to take $O(k \log n)$ samples, and then run through all n^k subsets of possible relevant variables. The first non-trivial algorithm was given only recently by Mossel *et al.* [MOS03], and runs in time roughly $O(n^{0.7k})$. Their algorithm relies on new analysis of the Fourier transform of juntas. However, even the question of whether one can learn k-juntas in polynomial time for $k = \omega(1)$ still remains open (*cf.* [Blu03a]).

We give a stronger and simpler reduction from learning noisy parities of size k to the problem of learning k-juntas.

Theorem 3 *Let $\mathcal{A}$ be an algorithm that learns parities of k variables on $\{0,1\}^n$ for every noise rate $\eta < 1/2$ in time $T(n, k, \frac{1}{1-2\eta})$. Then there exists an algorithm that learns k-juntas in time $O(2^{2k}k \cdot T(n, k, 2^{k-1}))$.*

This reduction also applies to learning k-juntas with random noise. A noisy parity of k variables is a special case of a k-junta. Thus we can reduce the noisy junta problem to a special case, at the cost of an increase in the noise level. By suitable modifications, the reduction from DNF can also be made resilient to random noise.

Even though at this stage our reductions for DNFs and juntas do not yield new algorithms they establish connections between well-studied open problems. Our reductions allow one to focus on functions with known and simple structure viz parities, in exchange for having to deal with random noise. They show that a non-trivial algorithm for learning parities of $O(\log n)$ variables will help make progress on a number of important questions regarding learning under the uniform distribution.

1.2 Hardness of Proper Agnostic Learning of Monomials and Halfspaces

Monomials are conjunctions of possibly negated variables and halfspaces are linear threshold functions over the input variables. These are perhaps the most fundamental and well-studied concept classes and are known to be learnable in a variety of settings. In this work we address proper agnostic learning of these concept classes. Uniform convergence results in Haussler's work [Hau92] (see also [KSS94]) imply that learnability these classes in the agnostic model is equivalent to the ability to come up with a function in $\mathcal{C}$ that has the optimal agreement rate with the given set of examples. For both monomials and halfspaces it is known that finding a hypothesis with the best agreement rate is NP-hard [JP78, AL88, HvHS95, KL93, KSS94]. However, for most practical purposes a hypothesis with agreement rate close to the optimum would be sufficient. This reduces the agnostic learning of a function class to a natural combinatorial approximation problem or, more precisely, to the following two problems: approximately minimizing the disagreement rate and approximately maximizing the agreement rate (sometimes referred to as *co-agnostic* learning). In this work we give essentially optimal hardness results for approximately maximizing the agreement rate with monomials and halfspaces.

1.2.1 Monomials

Monomials are long-known to be learnable in the PAC model and its various relatives [Val84]. They are also known to be learnable attribute-efficiently [Lit88, Hau88] and in the presence of random classification noise [Kea98]. With the exception of Littlestone's Winnow algorithm that produces halfspaces as its hypotheses these learning algorithms are proper. This situation contrasts the complexity of proper learning in the agnostic learning model. Angluin and Laird proved that finding a *monotone* (that is without negations) monomial with the maximum agreement rate (this problem is denoted MMon-MA) is NP-hard [AL88]. This was extended to

general monomials by Kearns and Li [KL93] (the problem is denoted Mon-MA). Ben-David *et al.* gave the first inapproximability result for this problem, proving that the maximum agreement rate is NP-hard to approximate within a factor of $\frac{770}{767} - \epsilon$ for any constant $\epsilon > 0$ [BDEL03]. This result was more recently improved by Bshouty and Burroughs to the inapproximability factor of $\frac{59}{58} - \epsilon$ [BB02].

The problem of approximately minimizing disagreement with a monomial (denoted Mon-MD) was first considered by Kearns *et al.* who give an approximation preserving reduction from the SET-COVER problem to Mon-MD [KSS94] (similar result was also obtained by Höffgen *et al.* [HvHS95]). This reduction together with the hardness of approximation results for SET-COVER due to Lund and Yannakakis [LY94] (see also [RS97]) implies that Mon-MD is NP-hard to approximate within a factor of $c \log n$ for some constant c.

On the positive side, the only non-trivial approximation algorithm is due to Bshouty and Burroughs and achieves $2 - \frac{\log n}{n}$-approximation for the agreement rate [BB02]. Note that factor 2 can always be achieved by either constant 0 or constant 1 function.

In this work, we give the following inapproximability results for Mon-MA.

Theorem 4 *For every constant $\epsilon > 0$,* Mon-MA *is* NP*-hard to approximate within a factor of* $2 - \epsilon$*.*

Then, under a slightly stronger assumption, we show that the second order term is small.

Theorem 5 *For any constant $\lambda > 0$, there is no polynomial-time algorithm that approximates* Mon-MA *within a factor of* $2 - 2^{-\log^{1-\lambda} n}$*, unless* $\mathrm{NP} \subseteq \mathrm{RTIME}(2^{(\log n)^{O(1)}})$.

Theorem 5 also implies strong hardness results for Mon-MD.

Corollary 1 *For any constant $\lambda > 0$, there is no polynomial time algorithm that approximates* Mon-MD *within a factor of* $2^{\log^{1-\lambda} n}$*, unless* $\mathrm{NP} \subseteq \mathrm{RTIME}(2^{(\log n)^{O(1)}})$.

In practical terms, these results imply that even very low (subconstant) amounts of adversarial noise in the examples make finding a term with agreement rate larger (even by very small amount) than $1/2$, NP-hard, in other words even *weak agnostic learning* of monomials is NP-hard. This resolves an open problem due to Blum [Blu98, Blu03b].

All of our results hold for the MMon-MA problem as well. A natural equivalent formulation of the MMon-MA problem is maximizing the number of satisfied *monotone disjunction constraints*, that is, equations of the form $t(x) = b$, where $t(x)$ is a disjunction of (unnegated) variables and $b \in \{0, 1\}$. We denote this problem by MAX-B-MSAT where B is the bound on the number of variables in each disjunction (see Definition 4 for more details). A corollary of our hardness result for MMon-MA is the following theorem

Theorem 6 *For any constant ϵ, there exists a constant B such that* MAX-B-MSAT *is* NP*-hard to approximate within* $2 - \epsilon$*.*

This result gives a form of the PCP theorem with imperfect completeness.

Finally, we show that Theorems 4 and 5 can be easily used to obtain hardness of agnostic learning results for classes richer than monomials, thereby improving on several known results and establishing hardness of agreement max/minimization for new function classes.

It is important to note that our results do not rule out agnostic learning of monomials when the disagreement rate is very low (i.e. $2^{-\log^{1-o(1)} n}$), weak agnostic learning with agreement lower than $1/2 + 2^{-\log^{1-o(1)} n}$, or non-proper agnostic learning of monomials.

Our proof technique is based on using Feige's multi-prover proof system for 3SAT-5 (3SAT with each variable occurring in exactly 5 clauses) together with set systems possessing a number of specially-designed properties. The set systems are then constructed by a simple probabilistic algorithm. As in previous approaches, our inapproximability results are eventually based on the PCP theorem. However, previous results reduced the

problem to an intermediate problem (such as MAX-CUT, MAX-E2-SAT, or SET-COVER) thereby substantially losing the generality of the constraints. We believe that key ideas of our technique might be useful in dealing with other constraint satisfaction problems involving constraints that are conjunctions or disjunctions of Boolean variables.

1.2.2 Halfspaces

The problem of learning a halfspace is one of the oldest and well-studied problems in machine learning, dating back to the work on Perceptrons in the 1950s [Agm64, Ros62, MP69]. If such a halfspace does exist, one can find it in polynomial time using efficient algorithms for Linear Programming. When the data can be separated with a significant margin simple online algorithms like Perceptron and Winnow are usually used (which also seem to be robust to noise [Gal90, Ama94]). In practice, positive examples often cannot be separated from negative using a linear threshold. Therefore much of the recent research in this area focuses on finding provably good algorithms when the data is noisy or inconsistent [BFKV97, ABSS97, Coh97, KKMS05]. Halfspaces are properly PAC learnable even in the presence of random noise: Blum *et al.* [BFKV97] show that a variant of the Perceptron algorithm can be used in this setting (see also [Coh97]).

The problem of maximizing agreements with a halfspace was first considered by Johnson and Preparata who prove that finding a halfspace that has the optimal agreement rate with the given set of examples over $\mathbb{Z}^n$ is NP-hard [JP78] (see also Hemisphere problem in [GJ79]). In the context of agnostic learning Höffgen *et al.* show that the same is true for halfspaces over $\{0,1\}^n$ [HvHS95]. A number of results are known on hardness of approximately maximizing the agreement with a halfspace (this problem is denoted HS-MA). Amaldi and Kann [AK95], Ben-David *et al.* [BDEL00], and Bshouty and Burroughs [BB02] prove that HS-MA is NP-hard to approximate within factors $\frac{262}{261}$, $\frac{418}{415}$, and $\frac{85}{84}$, respectively.

The results of Höffgen *et al.* imply that approximating the minimum disagreement rate of a halfspace within $c \log n$ is NP-hard for some constant c. Further Arora *et al.* [ABSS97] improve this factor to $2^{\log^{0.5-\delta} n}$ under stronger complexity assumption $\mathrm{NP} \not\subseteq \mathrm{DTIME}(2^{(\log n)^{O(1)}})$.

We give the optimal (up to the second order terms) hardness result for HS-MA with examples over $\mathbb{Q}^n$. Namely we show that even if there is a halfspace that correctly classifies $1-\epsilon$ fraction of the input, it is hard to find a halfspace that is correct on a $\frac{1}{2}+\epsilon$ fraction of the inputs for any $\epsilon > 0$ assuming $\mathrm{P} \neq \mathrm{NP}$. Under stronger complexity assumptions, we can take ϵ to be as small as $2^{-\sqrt{\log n}}$ where n is the size of the input.

Theorem 7 *If* $\mathrm{P} \neq \mathrm{NP}$ *then for any constant* $\epsilon > 0$ *no polynomial time algorithm can distinguish between the following cases of the halfspace problem over* $\mathbb{Q}^n$*:*

- $1-\epsilon$ *fraction of the points can be correctly classified by some halfspace.*
- *No more than* $1/2+\epsilon$ *fraction of the points can be correctly classified by any halfspace.*

Moreover if we assume that $\mathrm{NP} \not\subseteq \mathrm{DTIME}(2^{(\log n)^{O(1)}})$ *we can take* $\epsilon = 2^{-\Omega(\sqrt{\log n})}$.

As in the case of monomials this result implies that even weak agnostic learning of halfspaces is a hard problem. In an independent work Guruswami and Raghavendra showed that an analogous hardness result is true even for halfspaces over points in $\{0,1\}^n$ [GR06].

The crux of our proof is to first show a hardness result for solving systems of linear equations over the reals. Equations are easier to work with than inequalities since they admit certain *tensoring* and *boosting* operations which can be used for gap amplification. We show that given a system where there is a solution satisfying a $1-\epsilon$ fraction of the equations, it is hard to find a solution satisfying even an ϵ fraction. We then reduce this problem to the halfspace problem. The idea of repeated tensoring and boosting was used by Khot and Ponnuswami for equations over $\mathbb{Z}_2$ in order to show hardness for Max-Clique [KP06]. The main technical difference in adapting

this technique to work over $\mathbb{Q}$ is keeping track of error-margins. For the reduction to halfspaces, we need to construct systems of equations where in the 'No' case, many equations are unsatisfiable by a large margin. Indeed our tensoring and boosting operations resemble taking tensor products of codes and concatenation with Hadamard codes over finite fields.

We note that the approximability of systems of linear equations over various fields is a well-studied problem. Håstad shows that no non-trivial approximation is possible over $\mathbb{Z}_2$ [Has01]. Similar results are known for equations over $\mathbb{Z}_p$ and finite groups [Has01, HER04]. However, to our knowledge this is the first optimal hardness result for equations over $\mathbb{Q}$. On one hand, the Fourier analytic techniques that work well for finite groups and fields do not seem to apply over $\mathbb{Q}$. On the other hand, the fact that we are not restricted to equations with constantly many variables makes our task much simpler. A natural open question is whether a similar hardness result holds for equations of constant size over $\mathbb{Q}$.

1.2.3 Relation to Non-proper Agnostic Learning of Monomials and Halfspaces

A natural and commonly considered extension of the basic agnostic model allows the learner to output hypotheses in arbitrary (efficiently evaluatable) form. While it is unknown whether this strengthens the agnostic learning model several positive results are only known in this non-proper setting. Kalai *et al.* recently gave the first non-trivial algorithm for learning monomials [KKMS05] in time $2^{\tilde{O}(\sqrt{n})}$. They also gave a breakthrough result for agnostic learning of halfspaces by showing a simple algorithm that agnostically learns halfspaces with respect to the uniform distribution on the hypercube up to any constant accuracy. Their algorithms output linear thresholds of parities as hypotheses.

An efficient agnostic learning algorithm for monomials or halfspaces (not necessarily proper) would have major implications on the status of other open problems in learning theory. For example, it is known that a DNF expression can be *weakly approximated* by a monomial (that is equal with probability $1/2+\gamma$ for a non-negligible γ). Therefore, as it was observed by Kearns *et al.* [KSS94], an agnostic learning algorithm for monomials would find a function that weakly learns a DNF expression. Such learning algorithm can then be converted to a regular PAC learning algorithm using any of the the boosting algorithms [Sch90, Fre90]. In contrast, at present the best PAC learning algorithm even for DNF expressions runs in time $2^{\tilde{O}(n^{1/3})}$ [KS01]. It is also known that any AC^0 circuit can be approximated by the sign of a low-degree polynomial over the reals with respect to any distribution [BRS91, ABFR91]. Thus, an efficient algorithm for agnostic learning of halfspaces would imply a quasipolynomial algorithm for learning AC^0 circuits – a problem for which no nontrivial algorithms are known. Another evidence of the hardness of the agnostic learning of halfspaces was recently given by Feldman *et al.* [FGKP06] who show that this problem is intractable assuming the hardness of Ajtai-Dwork cryptosystem [AD97] (this result also follows easily from an independent work of Klivans and Sherstov [KS06]). Kalai *et al.* proved that agnostic learning of halfspaces with respect to the uniform distribution implies learning of parities with random classification noise – a major open problem in learning theory (see Section 3 for more details on the problem).

1.3 Organization of This Paper

In Section 2 we define the relevant learning models. Section 3 describes our result on agnostic learning of parities and its applications to learning of DNFs and juntas. In Sections 4 and 5 we prove the hardness of agnostically learning monomials and halfspaces respectively.

2 Learning Models

The learning models discussed in this work are based on Valiant's well-known PAC model [Val84]. In this model, for a concept c and distribution $\mathcal{D}$ over X, an *example oracle* $\mathrm{EX}(c, \mathcal{D})$ is an oracle that upon request returns an example $\langle x, c(x)\rangle$ where x is chosen randomly with respect to $\mathcal{D}$. For $\epsilon \geq 0$ we say that function g ϵ-approximates a function f with respect to distribution $\mathcal{D}$ if $\Pr_{\mathcal{D}}[f(x) = g(x)] \geq 1 - \epsilon$. For a concept class $\mathcal{C}$, we say that an algorithm $\mathcal{A}$ *efficiently* learns $\mathcal{C}$, if for every $\epsilon > 0$, $c \in \mathcal{C}$, and distribution $\mathcal{D}$ over X, $\mathcal{A}$ given access to $\mathrm{EX}(c, \mathcal{D})$ outputs, with probability at least $1/2$, a hypothesis h that ϵ-approximates c. The learning algorithm is efficient if it runs in time polynomial in $1/\epsilon$, and the *size* s of the learning problem where the size of the learning problem is equal to the length of an input to c plus the description length of c in the representation associated with $\mathcal{C}$. An algorithm is said to *weakly* learn $\mathcal{C}$ if it produces a hypothesis h that $(\frac{1}{2} - \frac{1}{p(s)})$-approximates (or *weakly approximates*) c for some polynomial p.

Random classification noise model introduced by Angluin and Laird formalizes the simplest type of white label noise. In this model for any $\eta \leq 1/2$ called the *noise rate* the regular example oracle $\mathrm{EX}(c, \mathcal{D})$ is replaced with the noisy oracle $\mathrm{EX}^{\eta}(c, \mathcal{D})$. On each call, $\mathrm{EX}^{\eta}(c, \mathcal{D})$, draws x according to $\mathcal{D}$, and returns $\langle x, c(x)\rangle$ with probability η and $\langle x, \neg c(x)\rangle$ with probability $1 - \eta$. When η approaches $1/2$ the label of the corrupted example approaches the result of a random coin flip, and therefore the running time of algorithms in this model is allowed to polynomially depend on $\frac{1}{1-2\eta}$.

2.1 Agnostic Learning Model

The *agnostic* PAC learning model by Haussler [Hau92] and Kearns *et al.* [KSS94] in order to relax the assumption that examples are labeled by a concept from a specific concept class. In this model no assumptions are made on the function that labels the examples, in other words, the learning algorithm has no prior beliefs about the target concept (and hence the name of the model). The goal of the agnostic learning algorithm for a concept class $\mathcal{C}$ is to produce a hypothesis $h \in \mathcal{C}$ whose error on the target concept is close to the best possible by a concept from $\mathcal{C}$.

Formally, for two Boolean functions f and h and a distribution $\mathcal{D}$ over the domain, we define $\Delta_{\mathcal{D}}(f, h) = \Pr_{\mathcal{D}}[f \neq h]$. Similarly, for a concept class $\mathcal{C}$ and a function f define $\Delta_{\mathcal{D}}(f, \mathcal{C}) = \inf_{h \in \mathcal{C}}\{\Delta_{\mathcal{D}}(f, h)\}$. Kearns *et al.* define the agnostic PAC learning model as follows [KSS94].

Definition 1 *An algorithm $\mathcal{A}$* agnostically *(PAC) learns a concept class $\mathcal{C}$ if for every $\epsilon > 0$, a Boolean function f and distribution $\mathcal{D}$ over X, $\mathcal{A}$, given access to EX$(f, \mathcal{D})$, outputs, with probability at least $1/2$, a hypothesis $h \in \mathcal{C}$ such that $\Delta_{\mathcal{D}}(f, h) \leq \Delta_{\mathcal{D}}(f, \mathcal{C}) + \epsilon$. As before, the learning algorithm is* efficient *if it runs in time polynomial in s and $1/\epsilon$.*

One can also consider a more general agnostic learning in which the examples are drawn from an arbitrary distribution over $X \times \{0, 1\}$ (and not necessarily consistent with a function). Clearly our negative results would apply in this more general setting and our positive result can be easily extended to it.

The agnostic learning model can also be thought of as a model of adversarial noise. By definition, a Boolean function f differs from some function in $c \in \mathcal{C}$ on $\Delta_{\mathcal{D}}(f, \mathcal{C})$ fraction of the domain. Therefore f can be thought of as c corrupted by noise of rate $\Delta_{\mathcal{D}}(f, \mathcal{C})$. Unlike in the random classification noise model the points on which a concept can be corrupted are unrestricted and therefore we refer to it as *adversarial classification noise*. Note that an agnostic learning algorithm will not necessarily find a hypothesis that approximates c – any other function in $\mathcal{C}$ that differs from f on at most $\Delta_{\mathcal{D}}(f, \mathcal{C}) + \epsilon$ fraction of the domain is acceptable. This way to view the agnostic learning is convenient when the performance of a learning algorithm depends on the rate of disagreement (that is the noise rate).

3 Learning Parities with Noise

In this section, we describe our reductions from learning of parities with adversarial noise to learning of parities with random noise. We will also show applications of this reduction to learning of DNF and juntas. We start by describing the main technical component of our reductions: an algorithm that using an algorithm for learning noisy parities, finds a heavy Fourier coefficient of a Boolean function if one exists. Following Jackson, we call such an algorithm a *weak parity algorithm*.

The high-level idea of the reduction is to modify the Fourier spectrum of a function f so that it is "almost" concentrated at a single point. For this, we introduce the notion of a probabilistic oracle for real-valued functions $f : \{0,1\}^n \to [-1,1]$. We then present a transformation on oracles that allows us to clear the Fourier coefficients of f belonging to a particular subspace of $\{0,1\}^n$. Using this operation we show that one can simulate an oracle which is close (in statistical distance) to a noisy parity.

3.1 Fourier Transform

Our reduction uses Fourier-analytic techniques which were first introduced to computational learning theory by Linial *et al.* [LMN93]. In this context we view Boolean functions as functions $f : \{0,1\}^n \to \{-1,1\}$. All probabilities and expectations are taken with respect to the uniform distribution unless specifically stated otherwise. For a Boolean vector $a \in \{0,1\}^n$ let $\chi_a(x) = (-1)^{a \cdot x}$, where '$\cdot$' denotes an inner product modulo 2, and let $\mathtt{weight}(a)$ denote the Hamming weight of a.

We define an inner product of two real-valued functions over $\{0,1\}^n$ to be $\langle f, g \rangle = \mathrm{E}_x[f(x)g(x)]$. The technique is based on the fact that the set of all parity functions $\{\chi_a(x)\}_{a \in \{0,1\}^n}$ forms an orthonormal basis of the linear space of real-valued functions over $\{0,1\}^n$ with the above inner product. This fact implies that any real-valued function f over $\{0,1\}^n$ can be uniquely represented as a linear combination of parities, that is $f(x) = \sum_{a \in \{0,1\}^n} \hat{f}(a)\chi_a(x)$. The coefficient $\hat{f}(a)$ is called Fourier coefficient of f on a and equals $\mathrm{E}_x[f(x)\chi_a(x)]$; a is called the *index* and $\mathtt{weight}(a)$ the *degree* of $\hat{f}(a)$. We say that a Fourier coefficient $\hat{f}(a)$ is θ-heavy if $|\hat{f}(a)| \geq \theta$. Let $L_2(f) = E_x[(f(x))^2]^{1/2}$. Parseval's identity states that

$$(L_2(f))^2 = \mathrm{E}_x[(f(x))^2] = \sum_a \hat{f}^2(a)$$

3.2 Finding Heavy Fourier Coefficients

Given the example oracle for a Boolean function f the main idea of the reduction is to transform this oracle into an oracle for a noisy parity χ_a such that $\hat{f}(a)$ is a heavy Fourier coefficient of f. First we define probabilistic oracles for real-valued functions in the range $[-1,+1]$.

Definition 2 *For any function $f : \{0,1\}^n \to [-1,1]$ a probabilistic oracle $\mathbb{O}(f)$ is the oracle that produces samples $\langle x, b \rangle$, where x is chosen randomly and uniformly from $\{0,1\}^n$ and $b \in \{-1,+1\}$ is a random variable with expectation $f(x)$.*

For a Boolean f this defines exactly $\mathrm{EX}(f,U)$. Random classification noise can also be easily described in this formalism. For $\theta \in [-1,1]$, and $f : \{0,1\}^n \to \{-1,+1\}$, define $\theta f : \{0,1\}^n \to [-1,+1]$ as $\theta f(x) = \theta \cdot f(x)$. A simple calculation shows that $\mathbb{O}(\theta f)$ is just an oracle for $f(x)$ with random noise of rate $\eta = 1/2 - \theta/2$. Our next observation is that if the Fourier spectra of f and g are close to each other, then their oracles are close in statistical distance.

Claim 1 *The statistical distance between the outputs of $\mathbb{O}(f)$ and $\mathbb{O}(g)$ is upper-bounded by $L_2(f-g)$.*

Proof: The probability that $\mathbb{O}(f)$ outputs $\langle x, 1\rangle$ is $(1 + f(x))/2$ and the probability that it outputs $\langle x, -1\rangle$ is $(1 - f(x))/2$. Therefore the statistical distance between $\mathbb{O}(f)$ and $\mathbb{O}(g)$ equals $\mathrm{E}_x\left[|f(x) - g(x)|\right]$. By Cauchy-Schwartz inequality

$$\left(\mathrm{E}_x\left[|f(x) - g(x)|\right]\right)^2 \leq \mathrm{E}_x\left[\left(f(x) - g(x)\right)^2\right]$$

and therefore the statistical distance is upper bounded by $L_2(f - g)$. □

We now describe the main transformation on a probabilistic oracle that will be used in our reductions. For a function $f : \{0,1\}^n \to [-1, 1]$ and a matrix $A \in \{0,1\}^{m\times n}$ define an A-projection of f to be

$$f_A(x) = \sum_{a\in\{0,1\}^n, Aa=1^m} \hat{f}(a)\chi_a(x),$$

where the product Aa is performed mod 2.

Lemma 1 *For the function f_A defined above:*

1. $f_A(x) = \mathrm{E}_{p\in\{0,1\}^m} f(x \oplus A^T p)\chi_{1^m}(p)$.
2. *Given access to the oracle $\mathbb{O}(f)$ one can simulate the oracle $\mathbb{O}(f_A)$.*

Proof: Note that for every $a \in \{0,1\}^n$ and $p \in \{0,1\}^m$,

$$\chi_a(A^T p) \;=\; (-1)^{a^T\cdot(A^T p)} \;=\; (-1)^{(Aa)^T\cdot p} \;=\; \chi_{Aa}(p)$$

Thus if $Aa = 1^m$ then $E_p[\chi_a(A^T p)\chi_{1^m}(p)] = E_p[\chi_{Aa\oplus 1^m}(p)] = 1$ otherwise it is 0. Now let

$$g_A(x) = \mathop{\mathrm{E}}_{p\in\{0,1\}^m}[f(x \oplus A^T p)\chi_{1^m}(p)].$$

We show that g_A is the same as the function f_A by computing its Fourier coefficients.

$$\begin{aligned}
\widehat{g_A}(a) &= \mathrm{E}_x[\mathrm{E}_p[f(x \oplus A^T p)\chi_{1^m}(p)\chi_a(x)]] \\
&= \mathrm{E}_p[\mathrm{E}_x[f(x \oplus A^T p)\chi_a(x)]\chi_{1^m}(p)] \\
&= \mathrm{E}_p[\hat{f}(a)\chi_a(A^T p)\chi_{1^m}(p)] \\
&= \hat{f}(a)\mathrm{E}_p[\chi_a(A^T p)\chi_{1^m}(p)]
\end{aligned}$$

Therefore $\widehat{g_A}(a) = \hat{f}(a)$ if $Aa = 1^m$ and $\widehat{g_A}(a) = 0$ otherwise. This is exactly the definition of $f_A(x)$.

For Part 2, we sample $\langle x, b\rangle$, choose random $p \in \{0,1\}^m$ and return $\langle x \oplus A^T p, b \cdot \chi_{1^m}(p)\rangle$. The correctness follows from Part 1 of the Lemma. □

We will use Lemma 1 to project f in a way that separates one of its significant Fourier coefficients from the rest. We will do this by choosing A to be a random $m \times n$ matrix for appropriate choice of m.

Lemma 2 *Let $f : \{0,1\}^n \to [-1, 1]$ be any function, and let $s \neq 0^n$ be any vector. Choose A randomly and uniformly from $\{0,1\}^{m\times n}$. With probability at least $2^{-(m+1)}$, the following conditions hold:*

$$\widehat{f_A}(s) = \hat{f}(s) \tag{1}$$

$$\sum_{a\in\{0,1\}^n\setminus\{s\}} \widehat{f_A}^2(a) \leq L_2^2(f)2^{-m+1} \tag{2}$$

Proof: Event (1) holds if $As = 1^m$, which happens with probability 2^{-m}.

For every $a \in \{0,1\}^n \setminus \{s, 0^n\}$ and a randomly uniformly chosen vector $v \in \{0,1\}^n$,

$$\begin{aligned} \Pr_v[v \cdot a = 1 \mid v \cdot s = 1] &= 1/2 \\ \text{Therefore,} \quad \Pr_A[Aa = 1^m \mid As = 1^m] &= 2^{-m} \end{aligned}$$

Whereas for $a = 0^n$, $\Pr_A[Aa = 1^m] = 0$. Hence

$$\begin{aligned} &\mathrm{E}_A\left[\left.\sum_{a \in \{0,1\}^n \setminus \{s\}} \widehat{f_A}^2(a) \right| As = 1^m\right] \\ &\leq \sum_{a \in \{0,1\}^n \setminus \{s\}} 2^{-m} \hat{f}^2(a) \;\leq\; 2^{-m} L_2^2(f). \end{aligned}$$

By Markov's inequality,

$$\begin{aligned} &\Pr_A\left[\left.\sum_{a \in \{0,1\}^n \setminus \{s\}} \widehat{f_A}^2(a) \geq 2^{-m+1} L_2^2(f) \right| As = 1^m\right] \\ &\leq\; 1/2. \end{aligned}$$

Thus conditioned on event (1), event (2) happens with probability at least 1/2. So both events happen with probability at least $2^{-(m+1)}$. □

Finally, we show that using this transformation, one can use an algorithm for learning noisy parities to get a weak parity algorithm.

Theorem 8 *Let $\mathcal{A}$ be an algorithm that learns parities of k variables over $\{0,1\}^n$ for every noise rate $\eta < 1/2$ in time $T(n, k, \frac{1}{1-2\eta})$ using at most $S(n, k, \frac{1}{1-2\eta})$ examples. Then there exists an algorithm* `WP-R` *that for every function $f : \{0,1\}^n \to [-1,1]$ that has a θ-heavy Fourier coefficient s of degree at most k, given access to $\mathbb{O}(f)$ finds s in time $O(T(n,k,1/\theta) \cdot S^2(n,k,1/\theta))$ with probability at least $1/2$.*

Proof: Let $S = S(n, k, \frac{1}{1-2\eta})$. The algorithm `WP-R` proceeds in two steps:

1. Let $m = \lceil 2 \log S \rceil + 3$. Let $A \in \{0,1\}^{m \times n}$ be a randomly chosen matrix and $\mathbb{O}(f_A)$ be the oracle for A-projection of f. Run the algorithm $\mathcal{A}$ on $\mathbb{O}(f_A)$.
2. If $\mathcal{A}$ stops in $T(n, k, 1/\theta)$ steps and outputs r with $\texttt{weight}(r) \leq k$, check that r is at least $\theta/2$-heavy and if so, output it.

Let s be a θ-heavy Fourier coefficient of degree at most k. Our goal is to simulate an oracle for a function that is close to a noisy version of $\chi_s(x)$.

By Lemma 2, in Step 1, with probability at least 2^{-m-1} , we create a function f_A such that $\widehat{f_A}(s) = \theta$ and

$$\sum_{a \neq s} \widehat{f_A}^2(a) \;\leq\; 2^{-m+1} L_2^2(f) \;\leq\; \frac{L_2^2(f)}{4S^2} \;\leq \frac{1}{4S^2}.$$

By Claim 1, the statistical distance between the oracle $\mathbb{O}(f_A)$ and oracle $\mathbb{O}(\widehat{f_A}(s)\chi_s(x))$ is bounded by

$$L_2(f_A - \widehat{f_A}(s)\chi_s(x)) = \left(\sum_{a \neq s} (\widehat{f_A}^2(a))\right)^{1/2} \leq \frac{1}{2S}\,,$$

hence this distance is small. Since $\mathcal{A}$ uses at most S samples, with probability at least $\frac{1}{2}$ it will not notice the difference between the two oracles. But $\mathbb{O}(\widehat{f_A}(s)\chi_s(x))$ is exactly the noisy parity χ_s with noise rate $1/2 - \widehat{f_A}/2$. If $\widehat{f_A} \geq \theta$ we will get a parity with $\eta \leq 1/2 - \theta/2 < 1/2$ and otherwise we will get a negation of χ_s with $\eta \leq 1/2 - \theta/2$. Hence we get $(1-2\eta)^{-1} \leq 1/\theta$, so the algorithm $\mathcal{A}$ will learn the parity s when executed either with the oracle $\mathbb{O}(f_A)$ or its negation. We can check that the coefficient produced by $\mathcal{A}$ is indeed heavy using Chernoff bounds, and repeat until we succeed. Using $O(2^m) = O(S^2)$ repetitions, we will get a $\theta/2$-heavy Fourier coefficient of degree k with probability at least $1/2$. We can boost this by repeating the algorithm. A-projection always clears the coefficient $\hat{f}(0^n)$ and therefore we need to check whether this coefficient is θ-heavy separately. □

Remark 1 *A function f can have at most $L_2^2(f)/\theta^2$ θ-heavy Fourier coefficients. Therefore by repeating* `WP-R` $O((L_2^2(f)/\theta^2) \cdot \log(L_2(f)/\theta)) = \tilde{O}(L_2^2(f)/\theta^2)$ *times we can, with high probability, obtain all the θ-heavy Fourier coefficients of f as it is required in some applications of this algorithm.*

3.3 Learning of Parities with Adversarial Noise

A weak parity algorithm is in its essence an algorithm for learning of parities with adversarial noise. In particular, Theorem 8 gives the following reduction from adversarial to random noise.

Theorem 9 *The problem of learning parities with adversarial noise of rate $\eta < \frac{1}{2}$ reduces to learning parities with random noise of rate η.*

Proof: Let f be a parity χ_s corrupted by noise of rate η. Then $\hat{f}(s) = \mathrm{E}[f\chi_s] \geq (1-\eta) + (-1)\eta = 1 - 2\eta$. Now apply the reduction from Theorem 8 setting $k = n$. We get an oracle for the function $\hat{f}(s)\chi_s(x)$, which is $\chi_s(x)$ with random noise of level η. □

Blum *et al.* give a sub-exponential algorithm for learning noisy parities.

Lemma 3 *[BKW03] Parity functions on $\{0,1\}^n$ can be learned in time and sample complexity $2^{O(\frac{n}{\log n})}$ in the presence of random noise of rate η for any constant $\eta < \frac{1}{2}$.*

This algorithm together with Theorem 9 gives Theorem 1.

One can also interpret Theorem 9 in terms of coding theory problems. Learning a parity function with noise is equivalent to decoding a random linear code from the same type of noise. Therefore Theorem 8 implies the following result.

Theorem 10 *Assume that there exists an algorithm* `RandCodeRandError` *that corrects a random linear $[m,n]$ code from random errors of rate η with probability at least $1/2$ (over the choice of the code, errors, and the random bits of the algorithm) in time $T(m,n)$. Then there exists an algorithm* `RandCodeAdvError` *that corrects a random linear $[M,n]$ code from up to $\eta \cdot M$ errors with probability at least $1/2$ (over the choice of the code and the random bits of the algorithm) in time $O(M \cdot T(m,n))$ for $M = 8m^2$.*

Note that for $\eta \geq 1/4$ there might be more than one codeword within the relative distance η. In this case by repetitively using `RandCodeAdvError` as in Remark 1, we can list-decode the random code.

3.4 Learning DNF Expressions

Jackson [Jac97] in his breakthrough result on learning DNF expressions with respect to the uniform distribution gives a way to use a weak parity algorithm and the boosting algorithm due to Freund [Fre90] to build a DNF learning algorithm. We can adapt Jackson's approach to our setting. We give an outline of the algorithm and omit the now-standard analysis.

We view a probability distribution $\mathcal{D}$ as a density function and define its L_∞ norm. Jackson's algorithm is based on the following Lemma (we use a refinement from [BF02]).

Lemma 4 ([BF02](Lemma 18)) *For any Boolean function f of DNF-size s and any distribution $\mathcal{D}$ over $\{0,1\}^n$ there exists a parity function χ_a such that $|\mathrm{E}_{\mathcal{D}}[f\chi_a]| \geq \frac{1}{2s+1}$ and*

$$\mathtt{weight}(a) \leq \log\left((2s+1)L_\infty(2^n\mathcal{D})\right).$$

This lemma implies that DNFs can be weakly learned by finding a parity correlated with f under distribution $\mathcal{D}(x)$ which is the same as finding a parity correlated with the function $2^n\mathcal{D}(x)f(x)$ under the uniform distribution. The range of $2^n\mathcal{D}(x)f(x)$ is not necessarily $[-1,1]$, whereas our WP−R algorithm was defined for functions with this range. So in order to apply Theorem 8, we first scale $2^n\mathcal{D}(x)f(x)$ to the range $[-1,1]$ and obtain the function $\mathcal{D}'(x)f(x)$, where $\mathcal{D}'(x) = \mathcal{D}(x)/L_\infty(2^n\mathcal{D})$ ($L_\infty(\mathcal{D})$ is known to the boosting algorithm). We then get the probabilistic oracle $\mathbb{O}(\mathcal{D}'(x)f(x))$ by flipping a ± 1 coin with expectation $\mathcal{D}'(x)f(x)$. Therefore a θ-heavy Fourier coefficient of $2^n\mathcal{D}(x)f(x)$ can be found by finding a $\theta/L_\infty(2^n\mathcal{D})$-heavy Fourier coefficient of $\mathcal{D}'(x)f(x)$ and multiplying it by $L_\infty(2^n\mathcal{D})$. We summarize this generalization in the following lemma.

Lemma 5 *Let $\mathcal{A}$ be an algorithm that learns parities of k variables over $\{0,1\}^n$ for every noise rate $\eta < 1/2$ in time $T(n,k,\frac{1}{1-2\eta})$ using at most $S(n,k,\frac{1}{1-2\eta})$ examples. Then there exists an algorithm WP−R′ that for every real-valued function ϕ that has a θ-heavy Fourier coefficient s of degree at most k, given access to random uniform examples of ϕ, finds s in time $O(T(n,k,L_\infty(\phi)/\theta)\cdot S(n,k,L_\infty(\phi)/\theta)^2)$ with probability at least $1/2$.*

The running time of WP−R′ depends on $L_\infty(2^n\mathcal{D})$ (polynomially if T is a polynomial) and therefore gives us an analogue of Jackson's algorithm for weakly learning DNFs. Hence it can be used with a boosting algorithm that produces distributions that are *polynomially-close* to the uniform distribution; that is, the distribution function is bounded by $p2^{-n}$ where p is a polynomial in learning parameters (such boosting algorithms are called *p-smooth*). In Jackson's result [Jac97], Freund's boost-by-majority algorithm [Fre90] is used to produce distribution functions bounded by $O(\epsilon^{-(2+\rho)})$ (for arbitrarily small constant ρ). More recently, Klivans and Servedio have observed [KS03] that a later algorithm by Freund [Fre92] produces distribution functions bounded by $\tilde{O}(\epsilon)$. Putting the two components together, we get the proof of Theorem 2.

3.5 Learning Juntas

For the class of k-juntas, we can get a simpler reduction with better parameters for noise. Since there are at most 2^k non-zero coefficients and each of them is at least 2^{-k+1}-heavy, for a suitable choice of m, the projection step is likely to isolate just one of them. This leaves us with an oracle $\mathbb{O}(\hat{f}(s)\chi_s)$. Since $\hat{f}(s) \geq 2^{-k+1}$, the noise parameter is bounded by $\eta < 1/2 - 2^{-k}$. Using Remark 1 we will obtain the complete Fourier spectrum of f by repeating the algorithm $O(k2^{2k})$ times. The proof of Theorem 3 follows from these observations.

3.6 Learning in the Presence of Random Noise

Our reductions from DNFs and k-juntas can be made tolerant to random noise in the original function.

This is easy to see in the case of k-juntas. An oracle for f with classification noise η' is the same as an oracle for the function $(1-2\eta')f$. By repeating the reduction used for k-juntas, we get an oracle for the function $\mathbb{O}((1-2\eta')\hat{f}_s\chi_s)$. Hence we have the following theorem:

Theorem 11 *Let $\mathcal{A}$ be an algorithm that learns parities of k variables on $\{0,1\}^n$ for every noise rate $\eta < 1/2$ in randomized time $T(n,k,\frac{1}{1-2\eta})$. Then there exists an algorithm that learns k-juntas with random noise of rate η' in time $O(k2^{2k}\cdot T(n,k,\frac{2^{k-1}}{1-2\eta'}))$.*

A noisy parity of k variables is a special case of a k-junta. Thus we have reduced the noisy junta problem to a special case viz. noisy parity, at the cost of an increase in the noise level.

Handling noise in the DNF reduction is more subtle since Freund's boosting algorithms do not necessarily work in the presence of noise, in particular Jackson's original algorithm does not handle noisy DNFs. Nevertheless, as shown by Feldman [Fel06a], the effect of noise can be offset if the weak parity algorithm can handle a "noisy" version of $2^n\mathcal{D}(x)f(x)$. More specifically, we need a generalization of the WP-R algorithm that for any real-valued function $\phi(x)$, finds a heavy Fourier coefficient of $\phi(x)$ given access to $\Phi(x)$, where $\Phi(x)$ is an independent random variable with expectation $\phi(x)$ and $L_\infty(\Phi(x)) \leq \frac{2L_\infty(\phi)}{1-2\eta}$. It is easy to see that WP-R′ can handle this case. Scaling by $L_\infty(\Phi(x))$ will give us a random variable $\Phi'(x)$ in the range $[-1,1]$ with expectation $\phi(x)/L_\infty(\Phi(x))$. By flipping a ± 1 coin with expectation $\Phi'(x)$ we will get a ± 1 random variable with expectation $\phi(x)/L_\infty(\Phi(x))$. Therefore WP-R algorithm will find a heavy Fourier coefficient of $\phi(x)$ (scaled by $L_\infty(\Phi(x)) \leq \frac{2L_\infty(\phi)}{1-2\eta}$). Altogether we obtain the following theorem for learning noisy DNFs.

Theorem 12 *Let $\mathcal{A}$ be an algorithm that learns parities of k variables on $\{0,1\}^n$ for every noise rate $\eta < 1/2$ in time $T(n,k,\frac{1}{1-2\eta})$ using at most $S(n,k,\frac{1}{1-2\eta})$ examples. Then there exists an algorithm that learns DNF expressions of size s with random noise of rate η' in time $\tilde{O}(\frac{s^4}{\epsilon^2}\cdot T(n,\log B,\frac{B}{1-2\eta'})\cdot S(n,\log B,\frac{B}{1-2\eta'})^2)$ where $B=\tilde{O}(s/\epsilon)$.*

4 Hardness of the Agnostic Learning of Monomials

In this section we prove our hardness result for agnostic learning of monomials and show some of its applications.

4.1 Preliminaries and Notation

For a vector v, we denote its i^{th} element by v_i (unless explicitly defined otherwise). In this section, we view all Boolean functions to be of the form $f:\{0,1\}^n \to \{0,1\}$. A *literal* is a variable or its negation. A *monomial* is a conjunction of literals or a constant (0 or 1). It is also commonly referred to as a *conjunction*. A monotone monomial is a monomial that includes only unnegated literals or is a constant. We denote the function class of all monomials by Mon and the class of all monotone monomials by MMon.

4.1.1 The Problem

We now proceed to define the problems of minimizing disagreements and maximizing agreements more formally. For a domain X, an *example* is a pair (x,b) where $x \in X$ and $b \in \{0,1\}$. An example is called *positive* if $b=1$, and *negative* otherwise. For a set of examples $S \subseteq X \times \{0,1\}$, we denote $S^+ = \{x \mid (x,1) \in S\}$ and similarly $S^- = \{x \mid (x,0) \in S\}$. For any function f and a set of examples S, the *agreement rate* of f with S is $\texttt{AgreeR}(f,S) = \frac{|T_f \cap S^+| + |S^- \setminus T_f|}{|S|}$, where $T_f = \{x \mid f(x)=1\}$. For a class of functions $\mathcal{C}$, let $\texttt{AgreeR}(\mathcal{C},S) = \max_{f\in\mathcal{C}}\{\texttt{AgreeR}(f,S)\}$.

Definition 3 *For a class of functions $\mathcal{C}$ and domain D, we define the* Maximum Agreement *problem $\mathcal{C}$-MA as follows: The input is a set of examples $S \subseteq D \times \{0,1\}$. The problem is to find a function $h \in \mathcal{C}$ such that $\texttt{AgreeR}(h,S) = \texttt{AgreeR}(\mathcal{C},S)$.*

For $\alpha \geq 1$, an α-approximation algorithm for $\mathcal{C}$-MA is an algorithm that returns a hypothesis h such that $\alpha \cdot \texttt{AgreeR}(h,S) \geq \texttt{AgreeR}(\mathcal{C},S)$. Similarly, an α-approximation algorithm for the *Minimum Disagreement* problem $\mathcal{C}$-MD is an algorithm that returns a hypothesis $h \in \mathcal{C}$ such that $1-\texttt{AgreeR}(h,S) \leq \alpha(1-\texttt{AgreeR}(\mathcal{C},S))$.

An extension of the original agnostic learning framework is the model in which a hypothesis may come from a richer class $\mathcal{H}$. The corresponding combinatorial problems were introduced by Bshouty and Burroughs and are denoted $\mathcal{C}/\mathcal{H}$-MA and $\mathcal{C}/\mathcal{H}$-MD [BB02]. Note that an approximation algorithm for these problems can return a value larger than $\texttt{AgreeR}(\mathcal{C}, S)$ and therefore cannot be used to approximate the value $\texttt{AgreeR}(\mathcal{C}, S)$.

Remark 2 *An α-approximation algorithm for $\mathcal{C}'$-MA(*MD*) where $\mathcal{C} \subseteq \mathcal{C}' \subseteq \mathcal{H}$ is an α-approximation algorithm for $\mathcal{C}/\mathcal{H}$-MA(*MD*).*

4.1.2 Agreement with Monomials and Set Covers

For simplicity we first consider the MMon-MA problem. The standard reduction of the general to the monotone case [KLPV87] implies that this problem is at least as hard to approximate as Mon-MA. We will later observe that our proof will hold for the unrestricted case as well. We start by giving two equivalent ways to formulate MMon-MA.

Definition 4 *The Maximum Monotone Disjunction Constraints problem* MAX-MSAT *is defined as follows: The input is a set C of* monotone disjunction constraints*, that is, equations of the form $t(x) = b$ where, $t(x)$ is a monotone disjunction and $b \in \{0, 1\}$. The output is a point $z \in \{0, 1\}^n$ that maximizes the number of satisfied equations in C. For an integer function B,* MAX-B-MSAT *is the same problem with each disjunction containing at most B variables.*

To see the equivalence of MMon-MA and MAX-MSAT, let t_i be the variable "x_i is present in the disjunction t". Then each constraint $t(z) = b$ in MMon-MA is equivalent to $\vee_{z_i=0} t_i = 1 - b$. Therefore we can interpret each point in an example as a monotone disjunction and the disjunction t as a point in $\{0, 1\}^n$.

Another equivalent way to formulate MMon-MA (and the one we will be using throughout our discussion) is the following.
Input: $\mathcal{S} = (S^+, S^-, \{S_i^+\}_{i\in[n]}, \{S_i^-\}_{i\in[n]})$ where $S_1^+, \ldots, S_n^+ \subseteq S^+$ and $S_1^-, \ldots, S_n^- \subseteq S^-$.
Output: A set of indices I that maximizes the sum of two values, $\texttt{Agr}^-(\mathcal{S}, I) = |\bigcup_{i\in I} S_i^-|$ and $\texttt{Agr}^+(\mathcal{S}, I) = |S^+| - |\bigcup_{i\in I} S_i^+|$. We denote this sum by $\texttt{Agr}(\mathcal{S}, I) = \texttt{Agr}^-(\mathcal{S}, I) + \texttt{Agr}^+(\mathcal{S}, I)$ and denote the maximum value of agreement by $\texttt{MMaxAgr}(\mathcal{S})$.

To see that this is an equivalent formulation, let $S_i^- = \{x \mid x \in S^- \text{ and } x_i = 0\}$ and $S_i^+ = \{x \mid x \in S^+ \text{ and } x_i = 0\}$. Then for any set of indices $I \subseteq [n]$, the monotone monomial $t_I = \wedge_{i\in I} x_i$ is consistent with all the examples in S^- that have a zero in at least one of the coordinates with indices in I, that is, with examples in $\bigcup_{i\in I} S_i^-$. It is also consistent with all the examples in S_+ that do not have zeros in coordinates with indices in I, that is, $S^+ \setminus \bigcup_{i\in I} S_i^+$. Therefore the number of examples with which t_I agrees is exactly $\texttt{Agr}(\mathcal{S}, I)$.

It is also possible to formulate Mon-MA in a similar fashion. We need to specify an additional bit for each variable that tells whether this variable is negated in the monomial or not (when it is present). Therefore the formulation uses the same input and the following output.
*Output(*Mon-MA*):* A set of indices I and a vector $a \in \{0, 1\}^n$ that maximizes the value

$$\texttt{Agr}(\mathcal{S}, I, a) = |\bigcup_{i\in I} Z_i^-| + |S^+| - |\bigcup_{i\in I} Z_i^+|,$$

where $Z_i^{+/-} = S_i^{+/-}$ if $a_i = 0$ and $Z_i^{+/-} = S^{+/-} \setminus S_i^{+/-}$ if $a_i = 1$. We denote the maximum value of agreement with a general monomial by $\texttt{MaxAgr}(\mathcal{S})$.

4.2 Hardness of Approximating Mon-MA and Mon-MD

It is easy to see that MMon-MA is similar to the SET-COVER problem. Indeed, our hardness of approximation result will employ some of the ideas from Feige's hardness of approximation result for SET-COVER [Fei98].

4.2.1 Feige's Multi-Prover Proof System

Feige's reduction from the SET-COVER problem is based on a multi-prover proof system for 3SAT-5. The basis of the proof system is the standard two-prover protocol for 3SAT in which the verifier chooses a random clause and a random variable in that clause. It then gets the values of all the variables in the clause from the first prover and the value of the chosen variable from the second prover. The verifier accepts if the clause is satisfied and the values of the chosen variable are consistent [ALM$^+$98]. Feige then amplifies the soundness of this proof system by repeating the test ℓ times (based on Raz' parallel repetition theorem [Raz98]). Finally, the consistency checks are distributed to k provers with each prover getting $\ell/2$ clause questions and $\ell/2$ variable questions. This is done using an asymptotically-good code with k codewords of length ℓ and Hamming weight $\ell/2$. The verifier accepts if at least two provers gave consistent answers. More formally, for integer k and ℓ such that $\ell \geq c_\ell \log k$ for some fixed constant c_ℓ, Feige defines a k-prover proof system for 3SAT-5 where:

1. Given a 3CNF-5 formula ϕ over n variables, verifier V tosses a random string r of length $\ell \log(5n)$ and generates k queries $q_1(r), \ldots q_k(r)$ of length $\ell \log(\sqrt{\frac{5}{3}}n)$.

2. Given answers $a_1, \ldots a_k$ of length 2ℓ from the provers, V computes $V_1(r, a_1), \ldots, V_k(r, a_k) \in [2^\ell]$ for fixed functions[1] $V_1, \ldots, V_k$.

3. V accepts if there exist $i \neq j$ such that $V_i(r, a_i) = V_j(r, a_j)$.

4. If $\phi \in$ 3SAT-5, then there exist a k-prover $\bar{P}$ for which $V_1(r, a_1) = V_2(r, a_2) = \cdots = V_k(r, a_k)$ with probability 1 (note that this is stronger than the acceptance predicate above).

5. If $\phi \notin$ 3SAT-5, then for any $\bar{P}$, V accepts with probability at most $k^2 2^{-c_0 \ell}$ for some fixed constant c_0.

4.2.2 Balanced Set Partitions

As in Feige's proof, the second part of our reduction is a set system with certain properties tailored to be used with the equality predicate in the Feige's proof system. Our set system consists of two main parts. The first part is sets divided into partitions in a way that sets in the same partition are highly correlated (e.g., disjoint) and sets from different partitions are uncorrelated. Covers by uncorrelated sets are balanced in the sense that they cover about the same number of points in S^+ and S^- and therefore the agreement rate is close to $1/2$. Therefore these sets force any approximating algorithm to use sets from the same partition.

The second part of our set system is a collection of uncorrelated smaller sets. These smaller sets do not substantially influence small covers but make any cover by a large number of sets balanced. Therefore unbalanced covers have to use a small number of sets and have sets in the same partition. Intuitively, this makes it possible to use an unbalanced cover to find consistent answers to verifiers questions. In this sense, the addition of smaller sets is analogous to the use of the random skew in the Håstad's long code test [Has01].

Formally, a *balanced set partition* $\mathcal{B}(m, L, M, k, \gamma)$ has the following properties:

1. There is a ground set B of m points.

2. There is a collection of L distinct partitions $p_1, \ldots, p_L$.

3. For $i \leq L$, partition p_i is a collection of k disjoint sets $B_{i,1}, \ldots, B_{i,k} \subseteq B$ whose union is B.

4. There is a collection of M sets $C_1, \ldots, C_M$.

[1] These functions choose a single variable from each answer to a clause question.

5. Let $\rho_{s,t} = 1 - (1 - \frac{1}{k^2})^s(1 - \frac{1}{k})^t$. For any $I \subseteq [M]$ and $J \subseteq [L] \times [k]$ with all elements having different first coordinate, it holds

$$\left| \frac{\left| \left(\bigcup_{i \in I} C_i\right) \bigcup \left(\bigcup_{(i,j) \in J} B_{i,j}\right) \right|}{m} - \rho_{|I|,|J|} \right| \leq \gamma \ .$$

To see why a balanced set partition could be useful in proving hardness for MMon-MA, consider an instance $\mathcal{S}$ of MMon-MA defined as follows. For $\mathcal{B}(m, L, M, k, \gamma)$ as above, let $S^+ = S^- = B$, $S^-_{j,i} = B_{j,i}$, and $S^+_{j,i} = B_{j,1}$. Now for any $j \in [L]$, and an index set $I_j = \{(j, i) \mid i \in [k]\}$, $|\mathtt{Agr}(\mathcal{S}, I_j)| \geq (2 - \frac{1}{k} - \gamma)m$. On the other hand, for any index set I that does not include two indices with the same first coordinate, we have that $|\mathtt{Agr}(\mathcal{S}, I)| \leq (1 + 2\gamma)m$. For sufficiently large k and sufficiently small γ, this creates a multiplicative gap of $2 - \epsilon$ between the two cases.

4.2.3 Creating Balanced Set Partitions

In this section, we show a straightforward randomized algorithm that produces balanced set partitions.

Theorem 13 *There exists a randomized algorithm that on input k, L, M, γ produces, with probability at least $\frac{1}{2}$, a balanced set partition $\mathcal{B}(m, L, M, k, \gamma)$ for $m = \tilde{O}(k^2\gamma^{-2} \log(M + L))$ in time $O((M + L)m)$.*

Proof: First we create the sets $B_{j,i}$. To create each partition $j \in [L]$, we roll m k-sided dice and denote the outcomes by $d_1, \ldots, d_m$. Set $B_{j,i} = \{r \mid d_r = i\}$. This clearly defines a collection of disjoint sets whose union is $[m]$. To create M sets $C_1, \ldots, C_M$, for each $i \in [M]$ and each $r \in [m]$, we include r in C_i with probability $\frac{1}{k^2}$.

Now let $I \subseteq [M]$ and $J \subseteq [L] \times [k]$ be a set of indices with different first coordinate (corresponding to sets from different partitions) and let $U = \left(\bigcup_{i \in I} C_i\right) \bigcup \left(\bigcup_{(i,j) \in J} B_{i,j}\right)$. Elements of these sets are chosen independently and therefore for each $r \in [m]$,

$$\Pr[r \in U] = 1 - (1 - \frac{1}{k^2})^{|I|}(1 - \frac{1}{k})^{|J|} = \rho_{|I|,|J|}$$

independently of other elements of $[m]$. Using Chernoff bounds, we get that for any $\delta > 0$,

$$\Pr\left[\left| \frac{|U|}{m} - \rho_{|I|,|J|} \right| > \delta\right] \leq 2e^{-2m\delta^2} \ ,$$

which is exactly the property 5 of balanced set partitions (for $\delta = \gamma$). Our next step is to ensure that property 5 holds for all possible index sets I and J. This can be done by first observing that it is enough to ensure that this condition holds for $\delta = \gamma/2$, $|I| \leq k^2 \ln \frac{1}{\delta}$ and $|J| \leq k \ln \frac{1}{\delta}$. This is true since for $|I| \geq k^2 \ln \frac{1}{\delta}$ and every t, $\rho_{|I|,t} \geq 1 - \delta$. Therefore $|U|/m - \rho_{|I|,t} \leq 1 - \rho_{|I|,t} \leq \delta < \gamma$. For the other side of the bound on the size of the union, let I' be a subset of I of size $k^2 \ln \frac{1}{\delta}$ and U' be the union of sets with indices in I' and J. It then follows that

$$\begin{aligned} \rho_{|I|,t} - \frac{|U|}{m} \leq & 1 - \frac{|U'|}{m} \leq 1 - (\rho_{k^2 \ln \frac{1}{\delta},t} - \delta) \\ \leq & 1 - (1 - \delta) + \delta = \gamma. \end{aligned}$$

The second condition, $|J| \leq k \ln \frac{1}{\delta}$, is obtained analogously.

There are at most M^s different index sets $I \subseteq [M]$ of size at most s and at most $(kL)^t$ different index sets J of size at most t. Therefore, the probability that property 5 does not hold is at most

$$((kL)^{k \ln \frac{1}{\delta}} + M^{k^2 \ln \frac{1}{\delta}}) \cdot 2e^{-2m\delta^2} .$$

For

$$m \geq 2k^2\gamma^{-2} \cdot \ln(kL + M) \cdot \ln \frac{2}{\gamma} + 2,$$

this probability is less than $1/2$. □

We can now proceed to the reduction itself.

4.2.4 Main Reduction

Below we describe our main transformation from Feige's proof system to MMon-MA. To avoid confusion we denote the number of variables in a given 3CNF-5 formula by d and use n to denote the number of sets in the produced MMon-MA instance (that corresponds to the number of variables in the original formulation).

Theorem 14 *For every $\epsilon > 0$ (not necessarily constant), there exists an algorithm Athat given a 3CNF-5 formula ϕ over d variables, produces an instance $\mathcal{S}$ of* MMon-MA *on base sets S^+ and S^- of size T such that*

1. *Aruns in time $2^{O(\ell)}$ plus the time to create a balanced set partition $\mathcal{B}(m, 2^\ell, 4^\ell, \frac{1}{4\epsilon}, \frac{\epsilon}{4})$, where $\ell = c_1 \log \frac{1}{\epsilon}$ for some constant c_1.*
2. *$|S^+| = |S^-| = T = (5d)^\ell m$, where m is the size of the ground set of the balanced set partition.*
3. *$n = \frac{4}{\epsilon}(4\sqrt{\frac{5}{3}} \cdot d)^\ell$.*
4. *If $\phi \in$3SAT-5, then $\texttt{MMaxAgr}(\mathcal{S}) \geq (2 - \epsilon)T$.*
5. *If $\phi \notin$3SAT-5, then $|\texttt{MMaxAgr}(\mathcal{S}) - T| \leq \epsilon \cdot T$.*

Proof: Let $k = \frac{1}{4\epsilon}$, $\gamma = \epsilon/4$, and V be Feige's verifier for 3SAT-5. Given ϕ, we construct an instance $\mathcal{S}$ of MMon-MA as follows. Let R denote the set of all possible random strings used by V, let Q_i denote the set of all possible queries to prover i and let $A_i = \{0,1\}^{2\ell}$ denote the set of possible answers of prover i. Let $L = 2^\ell$, $M = 2^{2\ell}$, and $\mathcal{B}(m, L, M, k, \gamma)$ be a balanced set partition. We set $S^+ = S^- = R \times B$, and for every $r \in R$ and $B' \subseteq B$, let (r, B') denote the set $\{(r, b) \mid b \in B'\}$. We now proceed to define the sets in $\mathcal{S}$. For $i \in [k]$, $q \in Q_i$ and $a \in A_i$ we set

$$S^-_{(q,a,i)} = \bigcup_{q_i(r)=q} (r, B_{V_i(r,a),i} \cup C_a) \text{ and}$$

$$S^+_{(q,a,i)} = \bigcup_{q_i(r)=q} (r, B_{V_i(r,a),1} \cup C_a) .$$

Intuitively, sets $S^-_{(q,a,i)}$ (or $S^+_{(q,a,i)}$) correspond to prover i responding a when presented with query q. We can also immediately observe that answers from different provers that are mapped to the same value (and hence cause the verifier to accept) correspond to sets in S^- that are almost disjoint and strongly overlapping sets in S^+. To formalize this intuition, we prove the following claims.

Claim 2 *If $\phi \in$3SAT-5, then $\texttt{MMaxAgr}(\mathcal{S}) \geq (2 - \epsilon)T$ for $T = m|R|$.*

Proof: Let $\bar{P}$ be the k-prover that always answers consistently and let $P_i(a)$ denote the answer of the i^{th} prover to question q. Now consider the set of indices

$$I = \{(q, P_i(q), i) \mid i \in [k],\ q \in Q_i\} .$$

For each $r \in R$, the prover $\bar{P}$ satisfies

$$\begin{aligned} V_1(r, P_1(q_1(r))) =& V_2(r, P_2(q_2(r))) = \cdots \\ =& V_k(r, P_k(q_k(r))) = c(r) . \end{aligned}$$

Therefore,

$$\bigcup_{i\in[k]} S^-_{(q_i(r),P_i(q_i(r)),i)} \subseteq \bigcup_{i\in[k]} (r, B_{c(r),i}) = (r, B) .$$

This means that sets with indices in I cover all the points in $S^- = R \times B$. On the other hand for each r,

$$\begin{aligned} \bigcup_{i\in[k]} S^+_{(q_i(r),P_i(q_i(r)),i)} =& \bigcup_{i\in[k]} (r, B_{c(r),1} \cup C_{P_i(q_i(r))}) \\ =& (r, B_{c(r),1}) \cup (r, \bigcup_{i\in[k]} C_{P_i(q_i(r))}) . \end{aligned}$$

This implies that for each r only $(r, B_{c(r),1} \cup C_{P_i(q_i(r))})$ is covered in (r, B). By property 5 of balanced set partitions, the size of this set is at most

$$\begin{aligned} (1 - (1 - \frac{1}{k})(1 - \frac{1}{k^2})^k + \gamma)m \leq& (1 - (1 - \frac{1}{k})^2 + \gamma)m \\ \leq& (\frac{2}{k} + \gamma)m < \epsilon m . \end{aligned}$$

This means that at most ϵ fraction of S^+ is covered by the sets with indices in I. Therefore,

$$\texttt{Agr}(\mathcal{S}, I) \geq (1 + 1 - \epsilon)m|R| = (2 - \epsilon)T .$$

□

For the case when $\phi \notin$ 3SAT-5, let I be any set of indices for the instance $\mathcal{S}$. Let $\mathcal{S}_r$ denote an instance of MMon-MA obtained by restricting $\mathcal{S}$ to points with the first coordinate equal to r. We denote corresponding restrictions of the base sets by S^-_r and S^+_r. It is easy to see that $\texttt{Agr}(\mathcal{S}, I) = \sum_{r\in R} \texttt{Agr}(\mathcal{S}_r, I)$. We say that r is *good* if $|\texttt{Agr}(\mathcal{S}_r, I) - m| > \frac{\epsilon}{2}m$, and let δ denote the fraction of good r's. Then it is clear that

$$\texttt{Agr}(\mathcal{S}, I) \leq \delta \cdot 2T + (1 - \delta)(1 + \epsilon/2)T \leq (1 + \epsilon/2 + 2\delta)T \text{ , and}$$

$$\texttt{Agr}(\mathcal{S}, I) \geq (1 - \delta)(1 - \epsilon/2)T \geq (1 - \epsilon/2 - \delta)T .$$

Hence

$$|\texttt{Agr}(\mathcal{S}, I) - T| \leq (\epsilon/2 + 2\delta)T. \tag{3}$$

Claim 3 *There exists a prover $\bar{P}$ that will make the verifier V accept with probability at least $\delta(k^2 \ln \frac{4}{\epsilon})^{-2}$.*

Proof: We define $\bar{P}$ with the following randomized strategy. Let q be a question to prover i. Define $A^i_q = \{a \mid (q,a,i) \in I\}$ and P_i to be the prover that presented with q answers with a random element from A^i_q. We show that properties of $\mathcal{B}$ imply that there exist i and j such that $a_i \in A^i_{q_i(r)}$, $a_j \in A^i_{q_j(r)}$, and $V_i(r,a_i) = V_j(r,a_j)$. To see this, denote $V^i_q = \{V_i(a) \mid a \in A^i_q\}$. Then

$$\begin{aligned}\texttt{Agr}^-(\mathcal{S}_r, I) &= \left| S^-_r \cap \left(\bigcup_{(q,a,i)\in I} S^-_{(q,a,i)} \right) \right| \\ &= \left| \left(\bigcup_{i\in[k],\ j\in V^i_{q_i(r)}} B_{j,i} \right) \bigcup \left(\bigcup_{i\in[k],\ a\in A^i_{q_i(r)}} C_a \right) \right| .\end{aligned}$$

Now, if for all $i \neq j$, $V^i_{q_i(r)} \cap V^j_{q_j(r)} = \emptyset$, then all elements in sets $V^1_{q_1(r)}, \ldots, V^k_{q_k(r)}$ are distinct and therefore by property 5 of balanced set partitions,

$$\left| \frac{\texttt{Agr}^-(\mathcal{S}_r, I)}{m} - 1 + (1 - \frac{1}{k^2})^s (1 - \frac{1}{k})^t \right| \leq \gamma ,$$

where $s = |\cup_{i\in[k]} A^i_{q_i(r)}|$ and $t = \sum_{i\in[k]} |V^i_{q_i(r)}|$. Similarly,

$$\begin{aligned}\texttt{Agr}^+(\mathcal{S}_r, I) &= m - \left| S^-_r \cap \left(\bigcup_{(q,a,i)\in I} S^-_{(q,a,i)} \right) \right| \\ &= \left| \left(\bigcup_{i\in[k],\ j\in V^i_{q_i(r)}} B_{j,1} \right) \bigcup \left(\bigcup_{i\in[k],\ a\in A^i_{q_i(r)}} C_a \right) \right|\end{aligned}$$

and therefore

$$\left| \frac{\texttt{Agr}^+(\mathcal{S}_r, I)}{m} - (1 - \frac{1}{k^2})^s (1 - \frac{1}{k})^t \right| \leq \gamma .$$

This implies that $|\texttt{Agr}(\mathcal{S}_r, I) - m| \leq 2\gamma m = \frac{\epsilon}{2} m$, contradicting the assumption that r is good. Hence, let i' and j' be the indices for which $V^{i'}_{q_{i'}(r)} \cap V^{j'}_{q_{j'}(r)} \neq \emptyset$. To analyze the success probability of the defined strategy, we observe that if $s \geq k^2 \ln \frac{4}{\epsilon}$, then $(1 - \frac{1}{k^2})^s < \frac{\epsilon}{4}$ and consequently

$$\left| \bigcup_{i\in[k],\ a\in A^i_{q_i(r)}} C_a \right| \geq (1 - \frac{\epsilon}{4} - \gamma) m .$$

Therefore $\texttt{Agr}^+(\mathcal{S}_r, I) \leq (\frac{\epsilon}{4} - \gamma) m$ and $\texttt{Agr}^-(\mathcal{S}_r, I) \geq (1 - \frac{\epsilon}{4} - \gamma) m$. Altogether, this would again imply that $|\texttt{Agr}(\mathcal{S}_r, I) - m| \leq (\frac{\epsilon}{4} + \gamma) m = \frac{\epsilon}{2}$, contradicting the assumption that r is good.

For all $i \in [k]$, $|A^i_{q_i(r)}| \leq s \leq k^2 \ln \frac{4}{\epsilon}$. In particular, with probability at least $(k^2 \ln \frac{4}{\epsilon})^{-2}$, $P_{i'}$ will choose $a_{i'}$ and $P_{j'}$ will choose $a_{j'}$ such that $V_{i'}(r, a_{i'}) = V_{j'}(r, a_{j'})$, causing V to accept. As this happens for all good r's, the success probability of $\bar{P}$ is at least $\delta(k^2 \ln \frac{4}{\epsilon})^{-2}$. □

Using the bound on the soundness of V, Claim 3 implies that $\delta(k^2 \ln \frac{4}{\epsilon})^{-2} \leq k^2 2^{-c_0 \ell}$, or $\delta \leq (k^3 \ln \frac{4}{\epsilon})^2 2^{-c_0 \ell}$. Thus for

$$\ell = \frac{1}{c_0} \log \left(\frac{4}{\epsilon}(k^3 \ln \frac{4}{\epsilon})^2\right) \leq c_1 \log \frac{1}{\epsilon} \tag{4}$$

we get $\delta \leq \frac{\epsilon}{4}$. We set c_1 to be at least as large as c_ℓ (constant defined in Section 4.2.1). For $\delta \leq \frac{\epsilon}{4}$ equation 3 gives $|\texttt{Agr}(\mathcal{S}, I) - T| \leq \epsilon T$. The total number of sets used in the reduction (which corresponds to the number of variables n is $k \cdot |Q| \cdot |A|$ where $|Q|$ is the number of different queries that a prover can get and $|A|$ is the total number of answers that a prover can return (both $|A|$ and $|Q|$ are equal for all the provers). Therefore, by the properties of Feige's proof system, $n = \frac{4}{\epsilon}(4\sqrt{\frac{5}{3}} \cdot d)^\ell$. □

An important property of this reduction is that all the sets that are created $S^{+/-}_{(q,a,i)}$ have size at most $\epsilon|Q||B|$, where $|Q|$ is the number of possible queries to a prover (it is the same for all the provers). Hence each set covers at most $\epsilon|Q|/|R| < \epsilon$ fraction of all the points. This implies that a monomial with a negated variable will be negative on all but fraction ϵ of all the positive examples and will be consistent with all but at most fraction ϵ of all the negative examples. In other words, a non-monotone monomial will always agree with at least $(1 - \epsilon)T$ examples and at most $(1 + \epsilon)T$ examples.

Corollary 2 *Theorem 14 holds even when the output $\mathcal{S}$ is an instance of* Mon-MA*, that is, with* $\texttt{MaxAgr}(\mathcal{S})$ *in place of* $\texttt{MMaxAgr}(\mathcal{S})$.

Remark 3 *For each $r \in R$ and $b \in B$, (r, B) belongs to at most $k \cdot M = \text{poly}(\frac{1}{\epsilon})$ sets in $\mathcal{S}$. This means that in the* MMon-MA *instance each example will have* $\text{poly}(\frac{1}{\epsilon})$ *zeros. This, in turn, implies that an equivalent instance of* MAX-MSAT *will have* $\text{poly}(\frac{1}{\epsilon})$ *variables in each disjunction.*

4.2.5 Results and Applications

We are now ready to use the reduction from Section 4.2.4 with balanced set partitions from Section 4.2.3 to prove our main theorems.

Theorem 15 (same as 4) *For every constant $\epsilon' > 0$,* MMon/Mon-MA *is* NP*-hard to approximate within a factor of $2 - \epsilon'$.*

Proof: We use Theorem 14 for $\epsilon = \epsilon'/2$. Then k, γ, and ℓ are constants and therefore $\mathcal{B}(m, 2^\ell, 4^\ell, \frac{1}{4\epsilon}, \frac{\epsilon}{4})$ can be constructed in constant randomized time. The reduction creates an instance of Mon-MA of size polynomial in d and runs in time $d^{O(\ell)} = \text{poly}(d)$. By derandomizing the construction of $\mathcal{B}$ in a trivial way, we get a deterministic polynomial-time reduction that produces a gap in Mon-MA instances of $\frac{2-\epsilon}{1+\epsilon} > 2 - \epsilon'$. □

Furthermore, Remark 3 implies that for any constant ϵ, there exists a constant B such that MAX-B-MSAT is NP-hard to approximate within $2 - \epsilon$, proving Theorem 6.

Theorem 4 can be easily extended to subconstant ϵ.

Theorem 16 (same as 5) *For any constant $\lambda > 0$, there is no polynomial-time algorithm that approximates* MMon/Mon-MA *within a factor of $2 - 2^{-\log^{1-\lambda} n}$, unless* $\text{NP} \subseteq \text{RTIME}(2^{(\log n)^{O(1)}})$.

Proof: We use Theorem 14 with $\epsilon' = 2^{-\log^r d}$ for some r to be specified later. Then $k = 4 \cdot 2^{\log^r d}$, $\gamma = 2^{-\log^r d}/4$ and $\ell = c_1 \cdot \log^r d$. Therefore $\mathcal{B}(m, 2^\ell, 4^\ell, \frac{1}{4\epsilon'}, \frac{\epsilon'}{4})$ can be constructed in polynomial in $2^{\log^r d}$ randomized time and $m = 2^{c_2 \log^r d}$. The rest of the reduction takes time $2^{O(\ell)} = 2^{O(\log^r d)}$ and creates an instance of MMon-MA over $n = d^{c_3 \log^r d} = 2^{c_3 \log^{r+1} d}$ variables. Therefore, for $r = \frac{1}{\lambda}$, $\epsilon' \leq 2^{-\log^{1-\lambda} n}$. □

It is easy to see that the gap in the agreement rate between $1-\epsilon$ and $1/2+\epsilon$ implies a gap in the disagreement rate of $\frac{1/2-\epsilon}{\epsilon} > \frac{1}{3\epsilon}$ (for small enough ϵ). That is, we get the following multiplicative gap for approximating Mon-MD.

Corollary 3 (same as 1) *For any constant $\lambda > 0$, there is no polynomial time algorithm that approximates* MMon/Mon-MD *within a factor of $2^{\log^{1-\lambda} n}$, unless* NP $\subseteq$ RTIME$(2^{(\log n)^{O(1)}})$.

A simple application of these results is hardness of approximate agreement maximization with function classes richer than monomials. More specifically, let $\mathcal{C}$ be a class that includes monotone monomials. Assume that for every $f \in \mathcal{C}$ such that f has high agreement with the sample, one can extract a monomial with "relatively" high agreement. Then we could approximate the agreement or the disagreement rate with monomials, contradicting Theorems 4 and 5. A simple and, in fact, the most general class with this property, is the class of threshold functions with low integer weights. Let $\mathrm{TH}_W(\mathcal{C})$ denote the class of all functions equal to $\frac{1}{2} + \frac{1}{2}\mathtt{sign}(\sum_{i \le k} w_i(2f_i - 1))$, where $k, w_1, \ldots, w_k$ are integer, $\sum_{i \le k} |w_i| \le W$, and $f_1, \ldots, f_k \in \mathcal{C}$ (this definition of a threshold function is simply $\mathtt{sign}(\sum_{i \le k} w_i f_i)$ when f_i and the resulting function are in the range $\{-1, +1\}$). The following lemma is a straightforward generalization of a simple lemma due to Goldmann *et al.* [GHR92] (the original version is for $\delta = 0$).

Lemma 6 *Let $\mathcal{C}$ be a class of functions and let $f \in TH_W(\mathcal{C})$. If for some function g and distribution $\mathcal{D}$, $\Pr_D[f = g] \ge 1 - \delta$, then for one of the input functions $h \in \mathcal{C}$ to the threshold function f, it holds that $|\Pr_{\mathcal{D}}[h = g] - 1/2| \ge \frac{1-\delta(W+1)}{2W}$.*

Proof: Let D' be the distribution D conditioned on $f(x) = g(x)$. By the definition of D', $\Pr_{D'}[f = g] = 1$. We can therefore apply the original lemma and get that there exists $h \in C$ such that $|\Pr_{\mathcal{D}'}[h = g] - 1/2| \ge \frac{1}{2W}$. Therefore $|\Pr_{\mathcal{D}}[h = g] - 1/2| \ge \frac{1-\delta(W+1)}{2W}$. □

Hence we obtain the following results.

Corollary 4 *For any constant $\lambda > 0$ and $t = 2^{\log^{1-\lambda} n}$, there is no polynomial-time algorithm that approximates* MMon/*TH_t*(Mon)-MD *within a factor of t, unless* NP $\subseteq$ RTIME$(2^{(\log n)^{O(1)}})$.

Corollary 5 *For every constant k and $\epsilon > 0$,* MMon/*TH_W*(Mon)-MA *is* NP*-hard to approximate within a factor of $1 + \frac{1}{W} - \epsilon$.*

Proof: The reduction in Theorem 4 proves hardness of distinguishing instances of MMon-MA with the maximum agreement rate r being $\ge 1 - \frac{\epsilon'}{2}$ and instances for which $|r - 1/2| \le \frac{\epsilon'}{2}$. If there exists an algorithm that, given sample with $r \ge 1 - \frac{\epsilon'}{2}$, can produce a function $f \in \mathrm{TH}_W(\mathsf{Mon})$ such that f agrees with at least $\frac{W}{W+1} + \epsilon'$ fraction of examples then, by Lemma 6, one of the monomials used by f has agreement rate r' that satisfies

$$|r' - \frac{1}{2}| \ge \frac{1 - \delta(W+1)}{2W} \ge \frac{1 - (\frac{1}{W+1} - \epsilon')(W+1)}{2W} = \frac{\epsilon'(W+1)}{2W} > \frac{\epsilon'}{2}\,.$$

Therefore MMon/TH_W(Mon)-MA cannot be approximated within $\frac{1-\epsilon'}{\frac{W}{W+1}+\epsilon'} \ge 1 + \frac{1}{W} - \epsilon$ for an appropriate choice of ϵ'. □

A k-term DNF can be expresses as $TH_{k+1}(\mathsf{Mon})$. Therefore Corollary 5 improves the best known inapproximability factor for (2-term DNF)-MA from $\frac{59}{58} - \epsilon$ [BB02] to $4/3 - \epsilon$ and gives the first results on hardness of agreement maximization with thresholds of any constant number of terms.

5 Hardness of the Agnostic Learning of Halfspaces

In this section we prove a hardness result for agnostic learning of halfspaces over $\mathbb{Q}^m$. As in the case of monomials, we obtain this result by proving hardness of approximately maximizing agreements with a halfspace, that is HS-MA(see Section 4.1.1 for the formal definition.

More specifically we show that the trivial factor 2 approximation algorithm for this problem is essentially the best one can do. The proof is by reduction from the gap version of 5-regular vertex cover to an intermediate problem called MaxLin-$\mathbb{Q}$, and then finally to the learning halfspaces problem. The following is the combinatorial version of the problem of learning a halfspace over $\mathbb{Q}^m$ with adversarial noise (as stated in Arora *et al.* [ABSS97]).

We begin by defining the MaxLin-$\mathbb{Q}$ problem. Informally, we are given a system of equations over rationals and we are expected to find an assignment that satisfies as many equations as possible. We will show that even if a large fraction, say 99%, of the equations can be satisfied, one can not efficiently find an assignment such that more than 1% of the equations are "almost" satisfied. That is, the difference in the left hand side and right hand side of all but 1% of the equations is "large".

Definition 5 *Given a system of linear equations with rational coefficients*

$$\{a_{i0} + \sum_{j=1}^{m} a_{ij}x_j = 0\}_{i=1,2,\ldots,N}$$

as input, the objective of the MaxLin-$\mathbb{Q}$ *problem is to find* $(x_1, x_2, \ldots, x_m) \in \mathbb{Q}^m$ *that satisfies the maximum number of equations. A system of equations is said to be a* (N, c, s, t) MaxLin-$\mathbb{Q}$ *instance if the number of equations in the system is* N *and one of the following conditions holds:*

- *At least* cN *of the equations can be satisfied by some assignment, or*
- *In any assignment,*

$$|a_{i0} + \sum_{j=1}^{m} a_{ij}x_j| < t$$

  is true for at most sN *values of* $i \in [N]$.

The goal of the MaxLin-$\mathbb{Q}$ *problem when given such an instance is to find out which of the two cases is true. If the system of equations satisfies the first condition, we say it has completeness* c*. In the other case, we say it has soundness* s *under tolerance* t*.*

An instance of MaxLin-$\mathbb{Q}$ can be specified by a matrix

$$\boldsymbol{A} = \begin{bmatrix} a_{10} & a_{11} & \ldots & a_{1m} \\ a_{20} & a_{21} & \ldots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{N0} & a_{N1} & \ldots & a_{Nm} \end{bmatrix}$$

We will refer to $\boldsymbol{A}$ itself as an instance of MaxLin-$\mathbb{Q}$. We may also use the rows of $\boldsymbol{A}$ to represent the equations in the instance. The MaxLin-$\mathbb{Q}$ problem is to find a vector $\boldsymbol{X} = (1, x_1, x_2, \ldots, x_n)$ such that $\boldsymbol{AX}$ has as many zeros as possible. In all the instances of MaxLin-$\mathbb{Q}$ that we consider, the number of variables will be less than the number of equations in the system. Also, the size of each entry of the matrix will be proportional to the number of equations. Hence, we refer to N itself as the size of the instance.

The main steps in the reduction from vertex cover to the learning halfspaces problem are as follows:

- Obtain a (N, c, s, t) instance of MaxLin-$\mathbb{Q}$ from the vertex cover instance for some fixed constants c, s and t (see Definition 5).
- Convert the above instance to a $(N', 1-\epsilon, \epsilon, t')$ gap where ϵ is a very small function of N'. To achieve this, we use two operations called *tensoring* and *boosting*.
- Convert the instance of the MaxLin-$\mathbb{Q}$ problem obtained to an instance of the halfspace problem.

5.1 A Small Hardness Factor for MaxLin-$\mathbb{Q}$

We first state the gap version of the NP-hardness result for regular vertex cover.

Lemma 7 *[PY91, ALM+98] There exist constants d and ζ such that given a 5-regular graph with n vertices, it is NP-hard to decide whether there is a vertex cover of size $\leq dn$ or every vertex cover is of size at least $(1+\zeta)dn$.*

Arora *et al.* [ABSS97] give a reduction from the above gap version of vertex cover of regular graphs to MaxLin-$\mathbb{Q}$. They show that if there is a "small" vertex cover, the reduction produces a MaxLin-$\mathbb{Q}$ instance in which a "large" fraction of the equations can be satisfied. But when there is no small vertex cover, only a small fraction of the equations can be exactly satisfied. We show that the proof can be strengthened so that if there is no small vertex cover, only a small fraction of equations can be satisfied even within a certain tolerance.

Lemma 8 *There exists a polynomial time algorithm that when given a 5-regular graph $G = (V, E)$ with n vertices as input produces a (N, c_0, s_0, t_0) MaxLin-$\mathbb{Q}$ instance $\boldsymbol{A}$ over n variables as output where $N = n^{O(1)}$, c_0 and s_0 are absolute constants satisfying $s_0 < c_0$, $t_0 = 1/3$ and:*

- *If G has a vertex cover of size dn, then at least c_0 fraction of the equations in $\boldsymbol{A}$ can be satisfied.*
- *If G has no vertex cover smaller than $(1+\zeta)dn$, then for any vector $\boldsymbol{X} = (1, x_1, x_2, \ldots, x_n)$, at least $(1 - s_0)$ fraction of the entries in $\boldsymbol{AX}$ have magnitude $\geq t_0$.*

Proof: The instance $\boldsymbol{A}$ contains one variable x_i for every vertex $v_i \in V$. Corresponding to every vertex, there is a constraint $x_i = 0$. Corresponding to every edge between v_i and $v_{i'}$, we add three constraints

$$\begin{aligned} -1 + x_i + x_{i'} &= 0 \\ -1 + x_i &= 0 \\ -1 + x_{i'} &= 0 \end{aligned}$$

In all, $\boldsymbol{A}$ has $n + 3M$ equations, where $M = |E| = 5n/2$. If there is a vertex cover V_0 of size dn, set $x_i = 1$ if $v_i \in V_0$ and $x_i = 0$ otherwise. This satisfies at least $(1-d)n + 2M$ equations.

Suppose there is no vertex cover smaller than $(1+\zeta)dn$. We will show that not too many of the $n + 3M$ equations in $\boldsymbol{A}$ can be satisfied under a tolerance of $1/3$. Under a tolerance of $1/3$, the n equations for the vertices relax to $|x_i| < 1/3$, and the equations for an edge relax to

$$\begin{aligned} |-1 + x_i + x_{i'}| &< 1/3 \\ |-1 + x_i| &< 1/3 \\ |-1 + x_{i'}| &< 1/3 \end{aligned}$$

Note that no more than two of the three inequalities for an edge can be simultaneously satisfied. We will show that given any rational assignment to the x_is, there is a $\{0, 1\}$ assignment that is just as good or better. Consider any $\boldsymbol{X} = (1, x_1, x_2, \ldots, x_n)$, where $x_i \in \mathbb{Q}$. Set $y_i = 0$ if $x_i < 1/3$ and $y_i = 1$ otherwise. It is clear that y_i satisfies the inequality for vertex v_i if x_i does. Now suppose at least one of the three inequalities for an edge (v_i, v_i') is satisfied by the x_is. Then, either $x_i > 1/3$ or $x_{i'} > 1/3$. In this case, at least one of y_i and $y_{i'}$ is set to 1. But then two of the equalities

$$\begin{aligned} y_i + y_{i'} &= 1 \\ y_i &= 1 \\ y_{i'} &= 1 \end{aligned}$$

are satisfied. Therefore, the y_i are at least as good an assignment as the x_i.

Let $\boldsymbol{Y} = (1, y_1, y_2, \ldots, y_n)$. If there is no vertex cover of size less than $(1+\zeta)dn$, $\boldsymbol{AY}$ must contain at least $(1+\zeta)dn + M$ entries that are 1. That is, $\boldsymbol{AY}$ contains at most $(1 - (1+\zeta)d)n + 2M$ zeros. The claim about the soundness follows. □

5.2 Amplifying the Gap for MaxLin-$\mathbb{Q}$

We define two operations called *tensoring* and *boosting*. Tensoring converts a $(N, 1-\epsilon, 1-\delta, t)$ MaxLin-$\mathbb{Q}$ instance to a $(N^2, 1-\epsilon^2, 1-\delta^2, t^2)$ MaxLin-$\mathbb{Q}$ instance. We use this to get the completeness close to 1. But as a side-effect, it also gets the soundness close to 1. We use boosting to overcome this problem. A (σ, ρ)-boosting converts a (N, c, s, t) MaxLin-$\mathbb{Q}$ instance to a $((\rho N)^\sigma, c^\sigma, s^\sigma, t/2)$ MaxLin-$\mathbb{Q}$ instance. We amplify the (c, s) gap for MaxLin-$\mathbb{Q}$ in four steps:

- Obtain a $(1-\epsilon, 1-K\epsilon)$ gap for very large constant K using tensoring.
- Obtain a $(1-\epsilon_0, \epsilon_0)$ gap for a very small constant $\epsilon_0 > 0$ by using a boosting operation. This gap is sufficient to prove a $2-\epsilon$ hardness factor for the halfspace problem for any constant $\epsilon > 0$.
- Improve the completeness even further to $1-o(1)$ while keeping the soundness below a constant, say $1/20$. This is done by alternately tensoring and boosting many times. At this stage, it is essential to use a more efficient variation of boosting called pseudo-boosting. A (σ, ρ)-pseudo-boosting converts a (N, c, s, t) MaxLin-$\mathbb{Q}$ instance to a $(O(\rho)^\sigma N, c^\sigma, s^{\Omega(\sigma)}, t/2)$ MaxLin-$\mathbb{Q}$ instance. Since we require $c^\sigma > s^{\Omega(\sigma)}$ for the reduction to be meaningful, we need some minimum gap between c and s. This is guaranteed by the first two steps.
- Using one more boosting operation, decrease the soundness. This gives the $(N', 1-\epsilon, \epsilon, t')$ instance where $\epsilon = 2^{-\Omega(\sqrt{\log N'})}$ as desired.

5.2.1 Large constant gap for MaxLin-$\mathbb{Q}$

We define the first operation called tensoring. This operation is similar to an operation defined by Dumer *et al.* [DMS03] on linear codes. Informally, the tensoring of a system of equations contains one equation for the "product" of every pair of equations. In this product, we replace the occurance of $x_{j_1}x_{j_2}$ with $x_{j_1 j_2}$ and x_j with x_{0j} respectively.

Definition 6 *The tensoring of the system of equations*

$$\{a_{i0} + \sum_{j=1}^{m} a_{ij}x_j = 0\}_{i=1,2,\ldots,N}$$

is the system

$$\{a_{i_1 0}a_{i_2 0} + a_{i_1 0}(\sum_{j_2=1}^{m} a_{i_2 j_2}x_{0j_2}) + a_{i_2 0}(\sum_{j_1=1}^{m} a_{i_1 j_1}x_{0j_1}) + (\sum_{j_1=1}^{m}\sum_{j_2=1}^{m} a_{i_1 j_1}a_{i_2 j_2}x_{j_1 j_2}) = 0\}_{i_1,i_2=1,\ldots,N}$$

In the matrix representation, the tensoring of

$$\begin{bmatrix} a_{10} & a_{11} & \ldots & a_{1m} \\ a_{20} & a_{21} & \ldots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{N0} & a_{N1} & \ldots & a_{Nm} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{bmatrix} = 0$$

is the system

$$\begin{bmatrix} a_{10} & a_{11} & \dots & a_{1m} \\ a_{20} & a_{21} & \dots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{N0} & a_{N1} & \dots & a_{Nm} \end{bmatrix} \begin{bmatrix} 1 & x_{01} & \dots & x_{0m} \\ x_{01} & x_{11} & \dots & x_{1m} \\ \vdots & \vdots & & \vdots \\ x_{0m} & x_{m1} & \dots & x_{mm} \end{bmatrix} \begin{bmatrix} a_{10} & a_{20} & \dots & a_{N0} \\ a_{11} & a_{21} & \dots & a_{N1} \\ \vdots & \vdots & & \vdots \\ a_{1m} & a_{2m} & \dots & a_{Nm} \end{bmatrix} = 0$$

where the the x_{ij}s in the second matrix are the variables in the new instance.

Lemma 9 *Let $\boldsymbol{A}$ be a (N, c, s, t) instance of* MaxLin-$\mathbb{Q}$*. Let $\boldsymbol{B}$ be obtained by tensoring $\boldsymbol{A}$. Then $\boldsymbol{B}$ is a $(N^2, 1-(1-c)^2, 1-(1-s)^2, t^2)$ instance*

Proof: Suppose there is a vector $\boldsymbol{X} = (1, x_1, x_2, \dots, x_m)$ such that $\boldsymbol{AX}$ has a zero in cN fraction of the entries. Define $x_{0j} = x_j$ and $x_{j_1 j_2} = x_{j_1} x_{j_2}$ for $j_1 \geq 1$. This satisfies all but $(1-c)^2 N^2$ of the equations in $\boldsymbol{B}$. It remains to show the claim about the soundness.

Suppose that for any vector $\boldsymbol{X} = (1, x_1, x_2, \dots, x_m)$, at least s fraction of the entries in $\boldsymbol{AX}$ have magnitude greater than or equal to t. Consider any assignment to the variables $(x_{j_1 j_2})$ in $\boldsymbol{B}$. Let $\boldsymbol{X}^*$ denote the matrix

$$\begin{bmatrix} 1 & x_{01} & \dots & x_{0m} \\ x_{01} & x_{11} & \dots & x_{1m} \\ \vdots & \vdots & & \vdots \\ x_{0m} & x_{m1} & \dots & x_{mm} \end{bmatrix}$$

We will show that at least $(1-s)^2 N^2$ entries in $\boldsymbol{AX}^* \boldsymbol{A}^T$ have magnitude $\geq t^2$. Let $\boldsymbol{X} = (1, x_{01}, x_{02}, \dots, x_{0m})$. The vector $\boldsymbol{AX}$ has at least $(1-s)N$ entries with magnitude $\geq t$. Let J be the set of indices of these entries. Let $\boldsymbol{V} = (\boldsymbol{AX}^*)^T$. Note that since the first column of $\boldsymbol{X}^*$ is $\boldsymbol{X}$, $\boldsymbol{V}$ has at least $(1-s)N$ entries in the first row that have magnitude $\geq t$. Let $\boldsymbol{V}_j$ denote the j^{th} column of $\boldsymbol{V}$. Note that if $j \in J$, $\boldsymbol{AV}_j$ contains at least $(1-s)N$ entries that have magnitude $\geq t^2$. Therefore, $\boldsymbol{AX}^* \boldsymbol{A}^T = \boldsymbol{V}^T \boldsymbol{A}^T = (\boldsymbol{AV})^T$ has at least $(1-s)^2 N^2$ entries with magnitude $\geq t^2$. □

We now define an operation called boosting. Roughly speaking, we pick σ equations at a time from the MaxLin-$\mathbb{Q}$ instance $\boldsymbol{A}$. We add ρ^σ linear combinations of these to the boosted instance $\boldsymbol{B}$. The intention is that even if one of the σ equations fails under some assignment, a lot of the ρ^σ corresponding equations in $\boldsymbol{B}$ must fail. This is accomplished by using a construction similar to Hadamard code.

Definition 7 *Let $\boldsymbol{A}$ be a* MaxLin-$\mathbb{Q}$ *instance with N equations. Let ρ, σ be two arbitrary numbers. We define the (ρ, σ)-boosting to be the* MaxLin-$\mathbb{Q}$ *instance $\boldsymbol{B}$ obtained as follows. For every possible choice $(\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, \dots, \boldsymbol{A}_{i_\sigma})$ of σ rows of $\boldsymbol{A}$ and a vector $(\rho_1, \rho_2, \dots, \rho_\sigma) \in [\rho]^\sigma$, add a row $\rho_1 \boldsymbol{A}_{i_1} + \rho_2 \boldsymbol{A}_{i_2} + \ldots + \rho_\sigma \boldsymbol{A}_{i_\sigma}$ to $\boldsymbol{B}$. We call the ρ^σ rows of $\boldsymbol{B}$ that correspond to a choice of $(\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, \dots, \boldsymbol{A}_{i_\sigma})$ a cluster.*

The idea behind adding ρ^σ equations to each cluster is the following. If $b_1 \geq t$, then for any b, $\rho_1 b_1 + b$ lies in the interval $(-t/2, t/2)$ for at most one value of $\rho_1 \in [\rho]$. Similarly, for any given values of $\rho_2, \dots, \rho_\sigma$ and $b_2, \dots, b_\sigma$, $\sum_{i=1}^{\sigma} \rho_i b_i$, lies in the interval $(-t/2, t/2)$ for at most one value of $\rho_1 \in [\rho]$. An analogy to Hadamard codes is that if a bit in a string is 1, then half of the positions in its Hadamard code are 1.

Lemma 10 *Let $\boldsymbol{A}$ be a (N, c, s, t)* MaxLin-$\mathbb{Q}$ *instance. Let $\boldsymbol{B}$ be a (ρ, σ)-boosting of $\boldsymbol{B}$. Then $\boldsymbol{B}$ is a $((\rho N)^\sigma, 1-\sigma(1-c), s^\sigma + \rho^{-1}, t/2)$ instance.*

Proof: There are N^σ choices for $(\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, \ldots, \boldsymbol{A}_{i_\sigma})$ and ρ^σ choices for $(\rho_1, \rho_2, \ldots, \rho_\sigma)$. This proves the claim about the size of $\boldsymbol{B}$.

Fix an assignment that satisfies c fraction of the equations in $\boldsymbol{A}$. Let W denote the set of equations in $\boldsymbol{A}$ that are satisfied by this assignment. The probability that all of the σ equations in a random choice of $(\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, \ldots, \boldsymbol{A}_{i_\sigma})$ are in W is at least $c^\sigma \geq 1 - \sigma(1-c)$. When this happens, all the equations in the cluster corresponding to the choice of $(\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, \ldots, \boldsymbol{A}_{i_\sigma})$ are satisfied by the same assignment.

Now suppose for any $\boldsymbol{X} = (1, x_1, x_2, \ldots, x_m)$, at least sN fraction of the entries in $\boldsymbol{AX}$ have magnitude $\geq t$. Fix any assignment $\boldsymbol{X}$ to the variables in $\boldsymbol{A}$. Consider σ rows $\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, \ldots, \boldsymbol{A}_{i_\sigma}$ from $\boldsymbol{A}$. Now suppose $|\boldsymbol{A}_{i_1}\boldsymbol{X}| \geq t$. Let $b \in \mathbb{Q}$. Then, for at most one value of $\rho_1 \in [\rho]$, $\rho_1 \boldsymbol{A}_{i_1}\boldsymbol{X} + b$ has magnitude less than $t/2$. Therefore, for all but a $1/\rho$ fraction of $(\rho_1, \rho_2, \ldots, \rho_\sigma) \in [\rho]^\sigma$,

$$|(\rho_1 \boldsymbol{A}_{i_1} + \rho_2 \boldsymbol{A}_{i_2} + \ldots + \rho_\sigma \boldsymbol{A}_{i_\rho})\boldsymbol{X}| \geq t/2$$

If $(\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, \ldots, \boldsymbol{A}_{i_\sigma})$ are σ random rows of $\boldsymbol{A}$, the probability that none of $\boldsymbol{A}_{i_1}\boldsymbol{X}$, $\boldsymbol{A}_{i_2}\boldsymbol{X}, \ldots, \boldsymbol{A}_{i_k}\boldsymbol{X}$ have magnitude $\geq t$ is at most s^σ. Therefore, at most $s^\sigma + (1-s^\sigma)\rho^{-1} \leq s^\sigma + \rho^{-1}$ fraction of the entries in $\boldsymbol{BX}$ have magnitude less than $t/2$. This proves the claim about the soundness. □

We now use tensoring and boosting to obtain a $1-\epsilon_0$ versus ϵ_0 gap for MaxLin-$\mathbb{Q}$.

Lemma 11 *For any constants $\epsilon_0 > 0$, $0 < s < c \leq 1$ and $t > 0$, there exists a polynomial time algorithm that when given a (N, c, s, t)* MaxLin-$\mathbb{Q}$ *instance $\boldsymbol{A}$ as input produces a $(N_1, 1-\epsilon_0, \epsilon_0, t_1)$ instance where $t_1 > 0$ is a constant.*

Proof: Let $\boldsymbol{B}$ be the instance obtained by repeatedly tensoring $\boldsymbol{A}$ l times. Then, $\boldsymbol{B}$ is a $(N^L, 1-(1-c)^L, 1-(1-s)^L, t^L)$ MaxLin-$\mathbb{Q}$ instance, where $L = 2^l$. Choose l large enough so that

$$\left(\frac{1-c}{1-s}\right)^L \ln(2/\epsilon_0) \leq \epsilon_0$$

Now we use (ρ, σ)-boosting on $\boldsymbol{B}$ where $\rho = 2/\epsilon_0$ and

$$\sigma = \frac{\ln(2/\epsilon_0)}{(1-s)^L}$$

The result is a (N_1, c_1, s_1, t_1) instance where

$$c_1 \geq 1 - \sigma(1-c)^L \geq 1 - \frac{\ln(2/\epsilon_0)}{(1-s)^L}(1-c)^L \geq 1 - \epsilon_0$$

and

$$(1-(1-s)^L)^\sigma \leq (1/e)^{\sigma(1-s)^L} = e^{-\ln(2/\epsilon_0)} = \epsilon_0/2$$

Therefore, $s_1 = (1-(1-s)^L)^\sigma + \rho^{-1} \leq \epsilon_0$ and $t_1 = t^L/2$. □

Note that combining Lemma 11 with Lemma 15 suffices to show a $2-\epsilon$ hardness factor for the halfspace problem for any constant $\epsilon > 0$. We now focus on obtaining an improved hardness result where ϵ is sub-constant.

5.2.2 Super-constant gap for MaxLin-$\mathbb{Q}$

We will now prove a $(1-\epsilon, \epsilon)$ hardness for MaxLin-$\mathbb{Q}$ for a sub-constant (as a function of the size N) value of ϵ. One hurdle to be overcome is the rapid increase in the size of the instance produced by both tensoring (from N to N^2) and boosting (from N to N^σ). To overcome this problem we now define a pseudo-random boosting, or simply *pseudo-boosting*, that acheives a similar improvement in soundness (with a similar expense in completeness) as normal boosting does, but increases the size by only a constant factor.

Definition 8 *A walk of length σ on a graph G is an ordered sequence of vertices $(v_1, v_2, \ldots, v_\sigma)$ such that there is an edge between v_i and v_{i+1} in G for all $1 \leq i < \sigma$.*

Definition 9 *Let $\boldsymbol{A}$ be a* MaxLin-$\mathbb{Q}$ *instance with N equations. Let ρ, σ be two arbitrary numbers. We define the (ρ, σ)-pseudo-boosting to be the* MaxLin-$\mathbb{Q}$ *instance $\boldsymbol{B}$ obtained as follows. Let G_N be the 5-regular Gabber-Galil graph on N vertices. Associate every vertex v of G_N to an equation $\boldsymbol{A}_v$ in $\boldsymbol{A}$. For every possible walk $(v_1, v_2, \ldots, v_\sigma)$ of length σ on G_N and a vector $(\rho_1, \rho_2, \ldots, \rho_\sigma) \in [\rho]^\sigma$, add a row $\rho_1 \boldsymbol{A}_{v_1} + \rho_2 \boldsymbol{A}_{v_2} + \ldots + \rho_\sigma \boldsymbol{A}_{v_\sigma}$ to $\boldsymbol{B}$. We call the ρ^σ rows of $\boldsymbol{B}$ that correspond to a walk on the rows of $\boldsymbol{A}$ a cluster.*

The specific kind of expander used in pseudo-boosting is not important. We would like to point out that since Gabber-Galil graphs are defined only for integers of the form $2p^2$, we might have to add some trivially satisfied equations to $\boldsymbol{A}$. This only improves the completeness of $\boldsymbol{A}$. The soundness suffers by at most $O(1/\sqrt{N})$, which is a negligible increase if the soundness of the instance $\boldsymbol{A}$ were constant. Hence, we ignore this issue from now on. Before we analyze pseudo-boosting, we mention some results about expanders that will be useful.

Lemma 12 *Let W denote a subset of the vertices of a regular graph G. If $|W| \geq (1-\epsilon)N$, then at most $\sigma\epsilon$ fraction of the walks of length σ contain a vertex from $\bar{W}$.*

Proof: Pick a walk uniformly at random from all possible walks of length σ on G. The probability that the i^{th} vertex of the walk is contained in $\bar{W}$ is at most ϵ. This is because the graph is regular and hence all vertices are equally likely to be visited as the i^{th} vertex. Applying union bound over all the σ possible locations for a vertex in the walk, the probability that at least one of the vertices in the walk is contained in $\bar{W}$ is no more than $\sigma\epsilon$. □

Lemma 13 *[LW95, Section 15] Let W be a subset of the vertices of G_N. Let $|W| \leq N/10$. There exists a constant r such that for sufficiently large N, at most r^σ fraction of all walks of length σ in G_N are contained within W.*

Lemma 14 *Let $\boldsymbol{A}$ be a $(N, c, 1/10, t)$* MaxLin-$\mathbb{Q}$ *instance. Let $\boldsymbol{B}$ be a (ρ, σ)-pseudo-boosting of $\boldsymbol{B}$. Then $\boldsymbol{B}$ is a $(5^{\sigma-1}\rho^\sigma N, 1-\sigma(1-c), r^\sigma + \rho^{-1}, t/2)$ instance, where r is the constant guaranteed by Lemma 13.*

Proof: The proof of the lemma will closely parallel that of Lemma 10. The number of walks of length σ beginning from each vertex in a graph G_N is $5^{\sigma-1}$. Corresponding to each walk, we add ρ^σ rows to $\boldsymbol{B}$. This proves the claim about the size of $\boldsymbol{B}$.

Fix an assignment that satisfies c fraction of the equations in $\boldsymbol{A}$. Let W denote the set of equations in $\boldsymbol{A}$ that are satisfied by this assignment. From Lemma 12, we know that at most $\sigma(1-c)$ fraction of walks of length σ visit a row from $\bar{W}$. If all of the σ rows visited by a walk are satisfied, then all the equations of $\boldsymbol{B}$ in the cluster corresponding to this walk are also satisfied under the same assignment.

Now suppose for any $\boldsymbol{X} = (1, x_1, x_2, \ldots, x_m)$, at least $N/10$ fraction of the entries in $\boldsymbol{AX}$ have magnitude $\geq t$. Fix any assignment $\boldsymbol{X}$ to the variables in $\boldsymbol{A}$. If $(\boldsymbol{A}_{v_1}, \boldsymbol{A}_{v_2}, \ldots, \boldsymbol{A}_{v_\sigma})$ is a random walk on G_N, then from Lemma 13, the probability that none of $\boldsymbol{A}_{v_1}\boldsymbol{X}$, $\boldsymbol{A}_{v_2}\boldsymbol{X}, \ldots, \boldsymbol{A}_{v_k}\boldsymbol{X}$ have magnitude $\geq t$ is at most r^σ for large enough N. Therefore, as in Lemma 10, at most $r^\sigma + (1-r^\sigma)\rho^{-1} \leq r^\sigma + \rho^{-1}$ fraction of the entries in $\boldsymbol{BX}$ have magnitude less than $t/2$. This proves the claim about the soundness. □

We now use tensoring with pseudo-boosting to obtain a super-constant hardness factor for MaxLin-$\mathbb{Q}$.

Theorem 17 *There exists a $2^{(\log n)^{O(1)}}$ time reduction that when given a 5-regular graph G on n vertices outputs a* MaxLin-$\mathbb{Q}$ *instance $\boldsymbol{A}_2$ of size $N_2 = 2^{(\log n)^{O(1)}}$ such that*

- *If there is a vertex cover of size dn, then there is an assignment that satisfies $1 - 2^{-\Omega(\sqrt{\log N_2})}$ fraction of the equations.*

- *If every vertex cover is of size $\geq (1+\zeta)dn$, then under any assignment, at most $2^{-\Omega(\sqrt{\log N_2})}$ fraction of the equations can be satisfied within a tolerance as large as $2^{-O(\sqrt{\log N_2})}$.*

where d and ζ are the constants mentioned in Lemma 7

We first use Lemma 8 and Lemma 11 to convert a vertex cover instance to a $(N_1, 1-\epsilon_0, \epsilon_0, t_1)$ MaxLin-$\mathbb{Q}$ instance $\boldsymbol{A}_1$. We then alternately tensor and pseudo-boost $\boldsymbol{A}_1$ so that the soundness stays below 1/20, but the completeness progressively comes closer to 1. As a final step, we pseudo-boost once more so that the completeness is $1-\epsilon$ and the soundness is ϵ for a small value ϵ as desired.

Proof: Fix

$$\epsilon_0 = \min\left\{\frac{\log r^{-1}}{4\log 40}, \frac{1}{20}\right\},$$

$\sigma_0 = \lceil (4\epsilon_0)^{-1} \rceil$ and $\rho_0 = 40$.

We first use Lemma 8 and Lemma 11 to convert the graph to a $(N_1, 1-\epsilon_0, 1/20, t_1)$ MaxLin-$\mathbb{Q}$ instance $\boldsymbol{A}_1$, where $N_1 = n^{O(1)}$. Suppose $\boldsymbol{B}_1$ is the result of tensoring and (ρ_0, σ_0)-pseudo-boosting $\boldsymbol{A}_1$ once. Then $\boldsymbol{B}_1$ is a $(O(N_1)^2, 1-\sigma_0\epsilon_0^2, r_0^\sigma + \rho_0^{-1}, t_1^2/2)$ instance (The claim about soundness follows since the soundness after tensoring is $1-(1-1/20)^2 \leq 1/10$ and we can apply Lemma 14 to bound the soundness after the pseudo-boosting). Since $\sigma_0\epsilon_0 \leq 1/2 < 1$, after one round of tensoring and pseudo-boosting, the completeness comes closer to 1. Also, the soundness stayed below $1/20$ after one round since $r_0^\sigma + \rho_0^{-1} \leq 2^{\log r/(4\epsilon_0)} + 1/40 \leq 1/20$. Now, let $\boldsymbol{A}_2$ be the result of repeatedly tensoring and (ρ_0, σ_0)-pseudo-boosting $\boldsymbol{A}_1$ l times. Let $L = 2^l$. Then $\boldsymbol{A}_2$ is a $(N_2, c_2, 1/20, t_2)$ instance where $N_2 = O(N_1)^L$, $c_2 = 1 - O(1)^L$ and $t_2 = \Omega(1)^L$.

As a final step, we now use (ρ_2, σ_2)-pseudo-boosting on $\boldsymbol{A}_2$ where $\rho_2 = \lceil 2.2^L \rceil$, $\sigma_2 = \left\lceil \frac{1+L}{\log(1/r)} \right\rceil$. This produces a (N_3, c_3, s_3, t_3) instance where $N_3 = O(\rho_2)^{\sigma_2} N_2 = 2^{O(L^2)} N_1^L$, $c_3 = 1 - O(L)O(1)^L = 1 - O(1)^L$, $s_3 = r_2^\sigma + \rho_2^{-1} \leq 2^{-L}$ and $t_3 = \Omega(1)^L$. Choose $L = \log N_1$. Then $\log N_3 = O(\log^2 N_1)$, which implies $L = \Omega(\sqrt{\log N_3})$. That is, $\boldsymbol{A}_3$ is a $(N_3, 1 - 2^{-\Omega(\sqrt{\log N_3})}, 2^{-\Omega(\sqrt{\log N_3})}, 2^{-O(\sqrt{\log N_3})})$ instance. □

5.3 From MaxLin-$\mathbb{Q}$ to the Halfspace Problem

Lemma 15 *There exists a polynomial time algorithm that when given a (N, c, s, t) instance $\boldsymbol{A}$ of* MaxLin-$\mathbb{Q}$ *produces a instance of the halfspace problem with $2N$ points such that:*

- *If there is a solution to the* MaxLin-$\mathbb{Q}$ *instance that satisfies $\geq cN$ of the equations, there is a halfspace that correctly classifies $\geq 2cN$ of the points.*
- *If $\boldsymbol{A}$ has soundness s under tolerance t, then no halfspace can correctly classify more than $(1+s)N$ of the points.*

Proof: We can rewrite each equation of $\boldsymbol{A}$ as two inequalities

$$-t' < a_{i0} + \sum_{j=1}^{n} a_{ij}x_j < t'$$

for any $t' \in \mathbb{Q}$ satisfying $0 < t' \leq t$. We select a value of t' in this range such that $t' \notin \{\pm a_{i0}\}_{i=1,2\ldots N}$. Homogenizing the above, one can rewrite $\boldsymbol{A}$ as a system of $2N$ inequalities

$$\begin{array}{l}(a_{i0} + t')x_0 + \sum_{j=1}^{n} a_{ij}x_j > 0 \\ (a_{i0} - t')x_0 + \sum_{j=1}^{n} a_{ij}x_j < 0\end{array} \tag{5}$$

where $i \in \{1, 2, \ldots, N\}$. If we could satisfy cN equations in $\boldsymbol{A}$, the new system has a solution satisfying $2cN$ inequalities by setting x_0 to 1. Suppose there is a solution satisfying $(1+s)N$ of the inequalities in (5). Note that if we set $x_0 \leq 0$, then since $t' > 0$, we will satisfy at most half of the inequalities, hence we can assume $x_0 > 0$. So we can scale the values of x_is so that x_0 is 1. Then, $(x_1, x_2, \ldots x_n)$ is a solution to $\boldsymbol{A}$ that satisfies s fraction of the equalities within tolerance $t' \leq t$.

The condition $t' \notin \{\pm a_{i0}\}_{i=1,2\ldots N}$ ensures that the coefficient of x_0 in all the $2N$ inequalities of (5) is non-zero. Now divide each inequality by the coefficient of x_0 and flip the direction of the inequality if we divided by a negative number. This way, we can convert the system (5) to an equivalent system of $2N$ inequalities, where each inequality is of the form

$$x_0 + \sum_{j=1}^{n} h_{ij} x_j > 0 \quad \text{or} \quad x_0 + \sum_{j=1}^{n} h_{ij} x_j < 0 \tag{6}$$

where $i \in \{1, 2, \ldots, 2N\}$. We now define the halfspace instance. The halfspace instance produced is over $\mathbb{R}^n$. For an inequality of the first form in (6), add the point $(h_{i1}, h_{i2}, \ldots, h_{in})$ to S^+. For an inequality of the second form add the point $(h_{i1}, h_{i2}, \ldots, h_{in})$ to S^-.

Suppose that there is an assignment $(x_0, x_1, \ldots, x_n)$ satisfying $2cN$ inequalities in (5). Then the hyperplane $x_0 + \sum_{j=1}^{n} x_j h_j = 0$ correctly classifies cN points in (S^+, S^-).

Now suppose there is a hyperplane $x_0 + \sum_{j=1}^{n} x_j h_j = 0$ that correctly classifies $(1+s)N$ points in (S^+, S^-) for some $s > 0$. Clearly, $x_0 > 0$. Scale the x_i so that $x_0 = 1$. Now, $(x_1, x_2, \ldots, x_n)$ is an assignment satisfying $(1+s)N$ inequalities in (6), and equivalently in (5). This completes the proof of the lemma. □

We can now prove Theorem 7.

Proof: We give a reduction from the vertex cover problem on 5-regular graphs mentioned in Lemma 7. The reduction will have running time $2^{(\log n)^{O(1)}}$ for n vertex graphs.

Let G be the input graph with n vertices. We use the reduction mentioned in Theorem 17 to produce a (N_2, c_2, s_2, t_2) MaxLin-$\mathbb{Q}$ instance $\boldsymbol{A}$, where $c_2 = 1 - \epsilon$, $s_2 = \epsilon$, $\epsilon = 2^{-\Omega(\sqrt{\log N_2})}$ and $t = 2^{-O(\sqrt{\log N_2})} > 0$. We transform $\boldsymbol{A}_2$ to a halfspace instance (S^+, S^-) as described in Lemma 15. Note that $N' = |S^+| + |S^-| = 4N_2$. Now, if there is a vertex cover of size $\leq dn$ in G, there is a halfspace that correctly classifies c_2 fraction of the points. On the other hand, if there is no vertex cover of size smaller than $(1+\zeta)dn$ in G, there is no halfspace that correctly classifies $\geq 1/2(1+s_2)$ fraction of the points.

Therefore the gap obtained is $c_2/(1/2(1+s_2)) = 2(1-\epsilon)/(1+\epsilon) = 2(1-O(\epsilon)) = 2 - 2^{-\Omega(\sqrt{\log N'})}$. □

6 Conclusions

We have shown connections between several well-studied open problems on learning under the uniform distribution. Our reductions imply that in a sense, the class of noisy parities is the hardest concept class for this model of learning. A natural question is whether one can reduce learning noisy parities of $O(\log n)$ variables to learning DNF (or juntas). On the positive side, a non-trivial algorithm for learning parities of $O(\log n)$ variables will help make progress on a number of important questions regarding learning under the uniform distribution. Indeed it is plausible that there exists a better algorithm than exhaustive search for this variant of the problem, as in the case of (unrestricted) noisy parity [BKW03].

It would be interesting to see whether the construction of balanced set partitions in Theorem 13 can be derandomized. We remark that derandomizing this construction would, in particular, produce a bipartite expander graph with an almost optimal expansion factor.

For halfspaces, a natural question is whether one can extend our hardness result for learning halfspaces to more general concept classes. One possible generalization would be to allow the sign of a low-degree polynomial

as hypothesis. Kalai *et al.* [KKMS05] use this hypothesis class to design algorithms for agnostic learning of halfspaces under some natural distributions. Similarly, for the problem of learning parities with adversarial noise, one could allow the algorithm to produce a low degree polynomial over $\mathbb{Z}_2$ as hypothesis. To the best of our knowledge, there are no negative results known for these problems.

Acknowledgments

The first author is grateful to Leslie Valiant for his advice and encouragement of this research. We would like to thank Avrim Blum and Salil Vadhan for valuable comments and suggestions. We would also like to thank Shaili Jain and the anonymous referees of CCC '06 and FOCS '06 for numerous helpful remarks, one of which simplified the proof of Theorem 8.

References

[ABFR91] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The expressive power of voting polynomials. In *Proc. 23^{rd} ACM Symp. on Theory of Computation*, pages 402–409, 1991.

[ABSS97] 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 and System Sciences*, 54(2):317–331, 1997.

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. 29th ACM Symposium on the Theory of Computing*, pages 284–293, 1997.

[Agm64] S. Agmon. The relaxation method for linear inequalities. *Canadian J. of Mathematics*, 6(3):382–392, 1964.

[AK95] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147(1&2):181–210, 1995.

[AL88] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988.

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[Ama94] E. Amaldi. From finding feasible susbsystems of linear systems to feedforward neural network design. *Ph.D Thesis, Swiss Federal Institute of Technology at Lausanne (EPFL)*, 1994.

[BB02] N. Bshouty and L. Burroughs. Maximizing agreements and coagnostic learning. In *Proceedings of ALT '02*, pages 83–97, 2002.

[BDEL00] S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. In *Proceedings of COLT*, pages 266–274, 2000.

[BDEL03] S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003.

[BEHW87] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[BF02] N. Bshouty and V. Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machince Learning Research*, 2:359–395, 2002.

[BFKL93] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Proceedings of International Cryptology Conference on Advances in Cryptology (CRYPTO)*, pages 278–291, 1993.

[BFKV97] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1/2):35–52, 1997.

[BJT04] N. Bshouty, J. Jackson, and C. Tamon. More efficient pac-learning of dnf with membership queries under the uniform distribution. *Journal of Computer and System Sciences*, 68(1):205–234, 2004.

[BKW03] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.

[BL97] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[Blu94] A. Blum. Relevant examples and relevant features: Thoughts from computational learning theory, 1994. In AAAI Fall Symposium on 'Relevance'.

[Blu98] A. Blum. Lecture notes for 15-854 machine learning theory. Available at `http://www.cs.cmu.edu/~avrim/ML98/index.html`, 1998.

[Blu03a] A. Blum. Open problem: Learning a function of r relevant variables. In *Proceeding of COLT*, pages 731–733, 2003.

[Blu03b] A. Blum. Tutorial on Machine Learning Theory given at FOCS '03, 2003. Available at `http://www.cs.cmu.edu/~avrim/Talks/FOCS03/`.

[BMvT78] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24, 1978.

[BRS91] R. Beigel, N. Reingold, and D. A. Spielman. The perceptron strikes back. In *Structure in Complexity Theory Conference*, pages 286–291, 1991.

[Coh97] E. Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *Proc.* 38^{th} *IEEE Symp. on Foundations of Computer Science*, pages 514–523, 1997.

[DMS03] I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003.

[Fei98] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[Fel06a] V. Feldman. On Attribute Efficient and Non-adaptive Learning of Parities and DNF Expressions, 2006. Manuscript. To appear in Journal of Machine Learning Research (preliminary version appeared in proceedings of COLT '05).

[Fel06b] V. Feldman. Optimal hardness results for maximizing agreements with monomials. In *Proceedings of Conference on Computational Complexity (CCC)*, pages 226–236, 2006.

[FGKP06] V. Feldman, P. Gopalan, S. Khot, and A. Ponuswami. New Results for Learning Noisy Parities and Halfspaces. In *Proceedings of FOCS*, pages 563–574, 2006.

[Fre90] Y. Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 202–216, 1990.

[Fre92] Y. Freund. An improved boosting algorithm and its implications on learning complexity. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 391–398, 1992.

[Gal90] S. Galant. Perceptron based learning algorithms. *IEEE Trans. on Neural Networks*, 1(2), 1990.

[GHR92] M. Goldmann, J. Hastad, and A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992.

[GJ79] M. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman, San Francisco, 1979.

[GKS01] S. A. Goldman, S. Kwek, and S. D. Scott. Agnostic learning of geometric patterns. *Journal of Computer and System Sciences*, 62(1):123–151, 2001.

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of STOC*, pages 25–32, 1989.

[GR06] V. Guruswami and P. Raghavendra. Hardness of Learning Halfspaces with Noise. In *Proceedings of FOCS*, pages 543–552, 2006.

[Has01] J. Hastad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[Hau88] D. Haussler. Space efficient learning algorithms. Technical Report UCSC-CRL-88-2, University of California at Santa Cruz, 1988.

[Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[HER04] J. Holmerin, L. Engebretsen, and A. Russell. Inapproximability results for equations over finite groups. *Theoretical Computer Science*, 312(1):17-45, 2004.

[HvHS95] K. Hoffgen, K. van Horn, and H. U. Simon. Robust trainability of single neurons. *Journal of Computer and System Sciences*, 50(1):114–125, 1995.

[Jac97] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.

[JP78] D. S. Johnson and F. P. Preparata. The densest hemisphere problem. *Theoretical Computer Science*, 6:93–107, 1978.

[Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[KKMS05] A. Tauman Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of FOCS*, pages 11–20, 2005.

[KL93] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

[KLPV87] M. Kearns, M. Li, L. Pitt, and L. Valiant. On the learnability of Boolean formulae. In *Proceedings of STOC*, pages 285–295, 1987.

[KM91] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. In *Proceedings of STOC*, pages 455–464, 1991.

[KP06] S. Khot and A. K. Ponnuswami. Better inapproximability results for maxclique, chromatic number and min-3lin-deletion. In *Proceedings of ICALP*, 2006. To appear.

[KS01] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. In *Proceedings of STOC*, pages 258–265, 2001.

[KS03] A. Klivans and R. Servedio. Boosting and hard-core set construction. *Machine Learning*, 51(3):217–238, 2003.

[KS06] A. Klivans and A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *Proceedings of FOCS*, pages 553–562, 2006.

[KSS94] M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[LBW95] W. S. Lee, P. L. Bartlett, and R. C. Williamson. On efficient agnostic learning of linear combinations of basis functions. In *Proceedings of COLT*, pages 369–376, 1995.

[Lev93] L. Levin. Randomness and non-determinism. *Journal of Symbolic Logic*, 58(3):1102–1103, 1993.

[Lit88] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[LW95] M. Luby and A. Wigderson. Pairwise independence and derandomization. Technical Report 95-035, International Computer Science Institute, 1995.

[LY94] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42-44, 1978.

[MOS03] E. Mossel, R. O'Donnell, and R. Servedio. Learning juntas. In *Proceedings of STOC*, pages 206–212, 2003.

[MP69] M. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge, MA, 1969.

[PY91] C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[Ros62] F. Rosenblatt. *Principles of neurodynamics*. Spartan Books, New York, 1962.

[RS97] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of STOC*, pages 475–484, 1997.

[Sch90] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Ver90] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 314–326, 1990.