Electronic Colloquium on Computational Complexity, Report No. 32 (2006)

Optimal Hardness Results for Maximizing Agreements with Monomials

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

December 5, 2005

Abstract

We consider the problem of finding a monomial (or a term) that maximizes the agreement rate with a given set of examples over the Boolean hypercube. The problem originates in learning and is referred to as *agnostic learning* of monomials. Finding a monomial with the highest agreement rate was proved to be NP-hard by Kearns and Li [14]. Ben-David *et al.* gave the first inapproximability result for this problem, proving that the maximum agreement rate is NP-hard to approximate within $\frac{770}{767} - \epsilon$, for any constant $\epsilon > 0$ [5]. The strongest known hardness of approximation result is due to Bshouty and Burroughs, who proved an inapproximability factor of $\frac{59}{58} - \epsilon$ [7]. We show that the agreement rate is NP-hard to approximate within $2 - \epsilon$ for any constant $\epsilon > 0$. This is optimal up to the second order terms. We extend this result to $\epsilon = 2^{-c\sqrt{\log n}}$ for some constant $c > 0$ under the assumption that $\mathsf{NP} \not\subseteq \mathsf{RTIME}(n^{\log(n)})$, thus also obtaining an inapproximability factor of $2^{c\sqrt{\log n}}$ for the symmetric problem of minimizing disagreements. This improves on the $\log n$ hardness of approximation factor due to Kearns *et al.* [16] and Hoffgen *et al.* [12].

*Supported by grants from the National Science Foundation NSF-CCF-9877049, NSF-CCF-0432037, and NSF-CCF-0427129.

ISSN 1433-8092

1 Introduction

We study the computational complexity of approximation problems arising in *agnostic learning* of monomials. The agnostic framework [11] is a useful variant of Valiant's PAC learning model in which, informally, nothing is known about the target function and a learning algorithm is required to do nearly as well as is possible using hypotheses from a given class. Haussler's work [11] implies that learnability in this model is, in a sense, equivalent to the ability to come up with a member of the hypothesis class that has high agreement rate with the given examples.

For a number of concept classes it is known that finding a hypothesis with the best agreement rate is NP-hard [3, 12, 14]. However, for most practical purposes a hypothesis with agreement rate close to the maximum would be sufficient. This reduces agnostic learning of a function class to a natural combinatorial approximation problem or, more precisely, to the following two problems: approximating the maximum agreement rate and the minimum disagreement rate. We address the approximation complexity of these problems for the class of monomials (also referred to as terms). The class of monomials is one of the simplest and most well-studied function classes easily learnable in a variety of settings. The learnability of monomials in the agnostic framework was first addressed by Angluin and Laird who proved that finding a monotone monomial with the maximum agreement rate (this problem is denoted MMon-MA) is NP-hard [3]. This was extended to general monomials by Kearns and Li [14] (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 $\epsilon > 0$ [5]. This result was more recently improved by Bshouty and Burroughs to the inapproximability factor of $\frac{59}{58} - \epsilon$ [7].

The problem of approximating the minimum 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 (similar result was also obtained by Hoffgen *et al.* [12]). This reduction together with the hardness of approximation results for SET-COVER due to Lund and Yannakakis [17] (see also [19]) 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 [7]. 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 1 *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 2 *There exists a constant $c > 0$ such that there is no polynomial-time algorithm that approximates Mon-MA within a factor of $2 - 2^{-c\sqrt{\log n}}$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\log(n)}) \cup \mathsf{RP}$.*

Theorem 2 also implies strong hardness results for Mon-MD.

Corollary 3 *There exists a constant $c > 0$, such that there is no polynomial time algorithm that approximates* Mon-*MD within a factor of* $2^{c\sqrt{\log n}}$, *unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\log(n)}) \cup \mathsf{RP}$.

In practical terms, these results imply that even very low (subconstant) amounts of "noise" in the examples make finding a term with agreement rate larger (even by very small amount) than 1/2, NP-hard.

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 clause constraints*, that is, equations of the form $t(x) = b$, where $t(x)$ is a disjunction of (unnegated) variables and $b \in \{0, 1\}$ (see Definition 6 for more details). In the proof of Theorem 1, each of the clause constrains will only have a constant number of variables and therefore our hardness result is equivalent to the PCP theorem (with imperfect completeness).

Finally, we show that Theorems 1 and 2 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.

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.

1.1 Related Work

Besides the results for monomials mentioned earlier, hardness of agnostic learning results are known for a number of other classes. Optimal hardness results are known for the class of parities. Håstad proved that approximating agreements with parities within a factor of $2 - \epsilon$ is NP-hard for any constant ϵ. Amaldi and Kann [2], Ben-David *et al.* [5], and Bshouty and Burroughs [7] prove hardness of approximating agreements with halfspaces (factors $\frac{262}{261}$, $\frac{418}{415}$, and $\frac{85}{84}$, respectively). Similar inapproximability results are also known for 2-term DNF, decision lists and balls [5, 7].

Arora *et al.* give strong inapproximability results for minimizing disagreements with halfspaces (factor $2^{\log^{0.5-\epsilon} n}$) and with parities[1] (factor $2^{\log^{1-\epsilon} n}$) under the assumption that $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{poly \log n})$. Bshouty and Burroughs prove inapproximability of minimizing disagreements with k-term multivariate polynomials (factor $\ln n$) and a number of other classes [6].

For an extension of the agnostic framework where a learner can output a hypothesis from a richer class of functions (see also Section 2.1) the first non-trivial algorithm for learning monomials was recently given by Kalai *et al.* [13]. Their algorithm learns monomials agnostically in time $2^{\tilde{O}(\sqrt{n})}$. They also gave a breakthrough result for agnostic learning of halfspaces by showing a simple algorithm that for any constant $\epsilon > 0$ agnostically learns

[1] This problem is more commonly known as "finding the nearest codeword".

halfspaces with respect to the uniform distribution up to ϵ accuracy (both their algorithm output thresholds of parities as hypotheses).

We also note that minimum disagreement cannot be approximated for classes that are known to be not *properly* learnable (i.e. when a hypothesis has to use the same representation as the class being learned). In particular, the minimum disagreement with various classes of DNF formulae, intersections of halfspaces, decision trees, and juntas cannot be approximated [18, 1].

2 Preliminaries and Notation

For a vector v, we denote its ith element by v_i (unless explicitly defined otherwise). For a positive integer m we denote $[m] = \{1, 2, \ldots, m\}$. We say that a function f (of any arity) is $\tilde{O}(g)$, where g is the function of the same parameters, if there exist two constants c and d such that $f \leq c \cdot g \cdot \log^d(g)$.

The domain of all discussed Boolean functions is the Boolean hypercube $\{0,1\}^n$. The ith literal is a function over $\{0,1\}^n$ equal to the i-th coordinate of a point and denoted x_i, or its negation, denoted $\bar{x}_i$. A *monomial* is a conjunction of literals and/or constants (0 and 1). It is also commonly referred to as a *term*. A monotone monomial is a monomial that includes only positive literals and constants. We denote the function class of all monomials by Mon and the class of all monotone monomials by MMon. A DNF formula is a disjunction of terms and a k-term DNF formula is a disjunction of k terms. A *halfspace* or a *threshold* function is a function equal to $\sum_{i \in [n]} w_i x_i \geq \theta$ (as a Boolean expression), where $w_1, \ldots, w_k, \theta$ are real numbers.

2.1 The Problem

For the purposes of this discussion we do not need the definitions of PAC learning and the agnostic learning framework. The interested reader is referred to the work of Haussler [11], Kearns *et al.* [16], and Valiant [21]. Instead, we directly define the combinatorial problems underlying learnability in this model.

For a domain D, an *example* is a pair (x, b) where $x \in D$ and $b \in \{0,1\}$. An example is called *positive* if $b = 1$, and *negative* otherwise. For a set of examples $S \subseteq D \times \{0,1\}$, 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 4 *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 [7]. 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 even just the value $\texttt{AgreeR}(\mathcal{C}, S)$.

Remark 5 *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).*

2.2 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 [15] 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 6 *The Maximum Monotone Clause Constraints problem MAX-MSAT is defined as follows: The input is a set C of* monotone clauses constraints*, that is, equations of the form $t(x) = b$ where, $t(x)$ is a clause without negated variables 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 clause 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 clause 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 clause and the clause t as a point in $\{0, 1\}^n$.

Another way 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 $\texttt{MaxAgr}(\mathcal{S})$.

3 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 [8].

3.1 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 [4]. Feige then amplifies the soundness of this proof system by repeating the test ℓ times (based on Raz' parallel repetition theorem [20]). 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[2] $V_1, \ldots, V_k$.

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.

3.2 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

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

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 [10].

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 $C_1, \ldots, C_M$.
5. For any $I \subseteq [M]$ and $J \subseteq [L] \times [k]$ with all elements having different first coordinate, it holds

$$\left| \frac{\left| \left(\bigcup_{i \in I} C_i \right) \cup \left(\bigcup_{(i,j) \in J} B_{i,j} \right) \right|}{m} - 1 + (1 - \frac{1}{k^2})^{|I|} (1 - \frac{1}{k})^{|J|} \right| \leq \gamma \ .$$

To see why a balanced set partition could be useful in proving hardness MMon-MA, consider an instance $\mathcal{S}$ of MMon-MA defined as follows. For $\mathcal{B}(m, L, k, \gamma)$ 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.

3.3 Creating Balanced Set Partitions

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

Theorem 7 *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 $r \in [m]$,

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

independently of other elements of $[m]$. Let $p_{s,t} = 1 - (1 - \frac{1}{k^2})^s(1 - \frac{1}{k})^t$, then using Chernoff bounds, we get that for any $\delta > 0$,

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

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, $p_{|I|,t} \geq 1 - \delta$. Therefore $|U|/m - p_{|I|,t} \leq 1 - p_{|I|,t} \leq \delta < \gamma$. 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

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

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.

3.4 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 8 *For every $\epsilon > 0$ (not necessarily constant), there exists an algorithm $\mathcal{A}$ that 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. *$\mathcal{A}$ runs 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 9 *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 ith prover to question a. 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

$$V_1(r, P_1(q_1(r))) = V_2(r, P_2(q_2(r))) = \cdots V_k(r, P_k(q_k(r))) = c(r).$$

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,

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

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

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

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 $\mathtt{Agr}(\mathcal{S}, I) = \sum_{r \in R} \mathtt{Agr}(\mathcal{S}_r, I)$. We say that r is *good* if $|\mathtt{Agr}(\mathcal{S}_r, I) - 1| > \frac{\epsilon}{2} m$, and let δ denote the fraction of good r's. Then it is clear that

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

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

Hence

$$|\mathtt{Agr}(\mathcal{S}, I) - T| \leq (\epsilon/2 + 2\delta)T. \tag{1}$$

Claim 10 *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 $A_q^i = \{a \mid (q, a, i) \in I\}$ and P_i to be the prover that presented with q answers with a random element from A_q^i. 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 $V_q^i = \{V_i(a) \mid a \in A_q^i\}$. Then

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

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,

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

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

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

and therefore

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

This implies that $|\mathtt{Agr}(\mathcal{S}_r, I) - m| \leq 2\gamma m = \frac{\epsilon}{2} m$, 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 10 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

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

we get $\delta \leq \frac{\epsilon}{4}$. We set c_1 to be at least as large as c_ℓ (constant defined in Section 3.1). For $\delta \leq \frac{\epsilon}{4}$ equation 1 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 11 *Theorem 8 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 12 *For each $r \in R$ and $b \in B$, (r, B) belongs to at most $k \cdot M = poly(\frac{1}{\epsilon})$ sets in $\mathcal{S}$. This means that in the* MMon-*MA instance each example will have $poly(\frac{1}{\epsilon})$ zeros. This, in turn, implies that an equivalent instance of MAX-MSAT will have $poly(\frac{1}{\epsilon})$ variables in each clause.*

3.5 Results and Applications

We are now ready to use the reduction from Section 3.4 with balanced set partitions from Section 3.3 to prove our main theorems.

Theorem 13 (same as 1) *For every constant $\epsilon' > 0$,* MMon/Mon-*MA is* NP-*hard to approximate within a factor of $2-\epsilon'$.*

Proof: We use Theorem 8 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 $d^{O(\ell)} = poly(d)$. 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 12 implies that for any constant ϵ, there exists a constant B such that MAX-B-MSAT is NP-hard to approximate within $2 - \epsilon$. This formulation implies the PCP theorem with imperfect completeness (as in the case of parities, if all the monotone clause constraints are satisfiable then the solution is easy to find).

Theorem 1 can be easily extended to subconstant ϵ.

Theorem 14 (same as 2) *There exists a constant $c > 0$, such that there is no polynomial-time algorithm that approximates* MMon/Mon-*MA within a factor of* $2 - 2^{-c\sqrt{\log n}}$*, unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\log(n)}) \cup \mathsf{RP}$.

Proof: We use Theorem 8 for $\epsilon' = d^{-1}$. Then $k = 4d$, $\gamma = d^{-1}/4$ and $\ell = c_1 \cdot \log d$. Therefore $\mathcal{B}(m, 2^\ell, 4^\ell, \frac{1}{4\epsilon'}, \frac{\epsilon'}{4})$ can be constructed in polynomial in d randomized time and $m = d^{c_2}$. The rest of the reduction takes time $d^{O(\ell)} = d^{O(\log d)}$ and creates an instance of MMon-MA over $n = d^{c_3 \log d}$ variables. Therefore, in terms of n, $\epsilon' = 2^{-c\sqrt{\log n}}$ for some constant c. □

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 15 (same as 3) *There exists a constant $c > 0$, such that there is no polynomial time algorithm that approximates* MMon/Mon-*MD within a factor of* $2^{c\sqrt{\log n}}$*, unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\log(n)}) \cup \mathsf{RP}$.

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 1 and 2. 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.* [9] (the original version is for $\delta = 0$).

Lemma 16 *Let $\mathcal{C}$ be a class of functions and let $f \in TH_W(\mathcal{C})$. If for some function g and distribution $\mathcal{D}$, $\mathbf{Pr}_D[f = g] \ge 1 - \delta$, then for one of the input functions $h \in \mathcal{C}$ to the threshold function f, it holds that $|\mathbf{Pr}_{\mathcal{D}}[h = g] - 1/2| \ge \frac{1-\delta(W+1)}{2W}$.*

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

Hence we obtain the following results.

Corollary 17 *Assume that* $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{\log(n)}) \cup \mathsf{RP}$*. There exists a constant* c *such that for* $t = 2^{c\sqrt{\log n}}$*, there is no polynomial-time algorithm that approximates* $\mathsf{MMon}/TH_t(\mathsf{Mon})$*-MD within a factor of* t*.*

Corollary 18 *For every constant* k *and* $\epsilon > 0$*,* $\mathsf{MMon}/TH_W(\mathsf{Mon})$*-MA is* NP*-hard to approximate within a factor of* $1 + \frac{1}{W} - \epsilon$*.*

Proof: The reduction in Theorem 1 proves hardness of distinguishing instances of MMon-MA with the maximum agreement rate r being $\geq 1 - \frac{\epsilon'}{2}$ and instances for which $|r - 1/2| \leq \frac{\epsilon'}{2}$. If there exists an algorithm that, given sample with $r \geq 1 - \frac{\epsilon'}{2}$, 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 16, one of the monomials used by f has agreement rate r' that satisfies

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

Therefore $\mathsf{MMon}/\mathrm{TH}_W(\mathsf{Mon})$-MA cannot be approximated within $\frac{1-\epsilon'}{\frac{W}{W+1}+\epsilon'} \geq 1 + \frac{1}{W} - \epsilon$ for an appropriate choice of ϵ'. $\Box$

A k-term DNF can be expresses as $TH_{k+1}(\mathsf{Mon})$. Therefore Corollary 18 improves the best known inapproximability factor for (2-term DNF)-MA from $\frac{59}{58} - \epsilon$ [7] to $4/3 - \epsilon$ and gives the first results on hardness of agreement maximization with thresholds of any constant number of terms.

4 Discussion and Further Work

While this work resolves approximation complexity of the maximum agreement problem for monomials, several questions remain open for other simple function classes. Most notably, the best inapproximability factor known for halfspaces is $\frac{85}{84}$, while no approximation algorithms achieving better than $(2 - \log n/n)$-approximation are known [7].

It would also be interesting to see whether the construction of balanced set partitions can be derandomized (removing RP from the $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{\log(n)}) \cup \mathsf{RP}$ assumption). We remark that derandomizing this construction would, in particular, produce a bipartite expander graph with almost optimal expansion factor.

5 Acknowledgments

I am grateful to Leslie Valiant for his advice and encouragement of this research. I would also like to thank Salil Vadhan for useful discussions and Shaili Jain for careful proofreading and valuable comments on the earlier versions of this paper.

References

[1] M. Alekhnovich, M. Braverman, V. Feldman, A. Klivans, and T. Pitassi. Learnability and automizability. In *Proceeding of FOCS '04*, pages 621–630, 2004. Extended version available at `www.eecs.harvard.edu/~vitaly/papers/properDNF.pdf`.

[2] 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.

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

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

[5] S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *J. Comput. Syst. Sci.*, 66(3):496–514, 2003.

[6] N. Bshouty and L. Burroughs. Bounds for the minimum disagreement problem with applications to learning theory. In *Proceedings of COLT '02*, pages 271–286, 2002.

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

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

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

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

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

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

[13] A. Tauman Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *FOCS '05*, pages 11–20, 2005.

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

[15] M. Kearns, M. Li, L. Pitt, and L. Valiant. On the learnability of boolean formulae. In *Proceedings of the Nineteenth Annual Symposium on Theory of Computing*, pages 285–295, 1987.

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

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

[18] L. Pitt and L. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.

[19] 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 '97*, pages 475–484, 1997.

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

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

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/