



# Maximizing Agreements with One-Sided Error with Applications to Heuristic Learning

NADER H. BSHOUTY\*  
*Department of Computer Science, Technion, Haifa 32000, Israel*

bshouty@cs.technion.ac.il

LYNN BURROUGHS  
*Department of Computer Science, University of Calgary, Calgary, Alberta T2N 1N4, Canada*

lynnb@cpsc.ucalgary.ca

**Editor:** Philip M. Long

**Abstract.** We study heuristic learnability of classes of Boolean formulas, a model proposed by Pitt and Valiant. In this type of example-based learning of a concept class  $C$  by a hypothesis class  $H$ , the learner seeks a hypothesis  $h \in H$  that agrees with all of the negative (resp. positive) examples, and a maximum number of positive (resp. negative) examples. This learning is equivalent to the problem of maximizing agreement with a training sample, with the constraint that the misclassifications be limited to examples with positive (resp. negative) labels. Several recent papers have studied the more general problem of maximizing agreements without this one-sided error constraint. We show that for many classes (though not all), the maximum agreement problem with one-sided error is more difficult than the general maximum agreement problem. We then provide lower bounds on the approximability of these one-sided error problems, for many concept classes, including Halfspaces, Decision Lists, XOR,  $k$ -term DNF, and neural nets.

**Keywords:** maximizing agreements, heuristic learning, example-based learning, agnostic learning, Boolean formulas, approximation

## 1. Introduction

Many papers have studied the problem of maximizing agreements, especially in connection to agnostic and co-agnostic learning (see for example Angluin & Laird, 1987; Kearns & Li, 1993; Höffgen, Simon, & Van Horn, 1995; Bartlett & Ben-David, 1999; Ben-David, Eiron, & Long, 2003; Kuhlmann, 2000; Bshouty & Burroughs, 2002b). In the co-agnostic learning model for concept classes  $C$  and  $H$ , a learning algorithm  $\mathcal{A}(\epsilon, \delta)$  requests random examples drawn according to some distribution  $\mathcal{D}$  over  $\{0, 1\}^n \times \{0, 1\}$  in order to determine a hypothesis  $h \in H$  that performs at least as well at fitting  $\mathcal{D}$  as the best  $f \in C$  does. That is, with probability at least  $1 - \delta$ ,  $h$  will satisfy  $\Pr[h(x) = y] \geq \Pr[f(x) = y] - \epsilon$  for a random example  $(x, y)$  chosen according to  $\mathcal{D}$ . The learning algorithm  $\mathcal{A}(\epsilon, \delta)$  must run in time polynomial in  $n$ ,  $\delta^{-1}$  and  $\epsilon^{-1}$ .

\*This research was supported by the fund for promotion of research at the Technion. Research no. 120-025.

It is implicit in the papers of Pitt and Valiant (1988) and Ben-David, Eiron, and Long (2003) that co-agnostic learning is equivalent to solving the following problem of maximizing agreements.

### $C/H$ -MA

**Input:** Multiset  $S$  of examples from  $\{0, 1\}^n \times \{0, 1\}$ .

**Output:** Hypothesis  $h \in H$  such that

$$|\{(x, y) \in S \mid h(x) = y\}| \geq \max_{f \in C} |\{(x, y) \in S \mid f(x) = y\}|.$$

When  $C \equiv H$ , we just write  $C$ -MA. For many classes (indeed, for all the classes we examine in this paper), finding a formula with the highest agreement rate in the class is an NP-hard task (Kearns & Li, 1993; Höffgen, Simon, & Van Horn, 1995; Håstad, 1997; Ben-David, Eiron, & Long, 2003; Bshouty & Burroughs, 2002b). It may be tractable, however, to find formulas with agreement rates within some fixed multiplicative factor  $\alpha$  of the optimal rate. For  $1 > \alpha > 0$ , a polynomial-time algorithm is said to be an  $\alpha$ -approximation algorithm for  $C/H$ -MA if it solves the following.

### $\alpha$ -Approximation of $C/H$ -MA

**Input:** Multiset  $S$  of examples from  $\{0, 1\}^n \times \{0, 1\}$ .

**Output:** Hypothesis  $h \in H$  such that

$$|\{(x, y) \in S \mid h(x) = y\}| \geq \alpha \max_{f \in C} |\{(x, y) \in S \mid f(x) = y\}|.$$

It is interesting to explore the values  $\alpha$  for which  $\alpha$ -approximation of  $C/H$ -MA is hard, and the values for which it is tractable. Since the constants 0 and 1 are in the classes  $C$  we consider, there is always a hypothesis that agrees with half the examples, and thus  $C/H$ -MA has a trivial  $\frac{1}{2}$ -approximation algorithm. Several researchers (Amaldi & Kann, 1995; Bartlett & Ben-David, 1999; Ben-David, Eiron, & Long, 2003; Kuhlmann, 2000; Bshouty & Burroughs, 2002b) have found constants  $\alpha$  (dependent on the classes  $C$  and  $H$  under study) such that the  $\alpha$ -approximation of  $C/H$ -MA is NP-hard. All of the classes  $C$  that we examine in this paper have some constant  $\alpha$  lower bound for approximating  $C$ -MA (Håstad, 1997; Ben-David, Eiron, & Long, 2003; Bshouty & Burroughs, 2002b).

For some applications, errors among the positive examples may incur a different cost than errors among the negative examples. It may be desirable to have a learning algorithm produce hypotheses that limit their classification errors to one specified side (either the positive or the negative examples). This motivated Pitt and Valiant (1988) to define two variants of the co-agnostic learning model, called  $\alpha$ -heuristic NFP (No False Positives) learning and  $\alpha$ -heuristic NFN (No False Negatives) learning. In these models the learner makes a query by asking for either a positive or negative example, which are drawn according to distributions  $\mathcal{D}^+$  and  $\mathcal{D}^-$ , respectively. The requirements for the hypotheses are given in the next definition.

Definition 1.  $\alpha$ -Heuristic NFN and NFP Learning Models

1. A learning algorithm  $\mathcal{A}$   $\alpha$ -heuristically NFN learns a class  $C$  by a class  $H$  if for any distributions  $\mathcal{D}^+$  on  $\{0, 1\}^n \times \{1\}$  and  $\mathcal{D}^-$  on  $\{0, 1\}^n \times \{0\}$ , and any  $\epsilon, \delta > 0$ , the algorithm  $\mathcal{A}(\epsilon, \delta)$  makes label-specific queries to receive positive examples  $(x, 1)$  drawn according to  $\mathcal{D}^+$  and negative examples  $(x, 0)$  drawn according to  $\mathcal{D}^-$ , and with probability at least  $1 - \delta$  outputs  $h \in H$  such that

$$\Pr_{\mathcal{D}^+}[h(x) = 0] < \epsilon \quad \text{and} \quad \Pr_{\mathcal{D}^-}[h(x) = 0] \geq \alpha \max_f \Pr_{\mathcal{D}^-}[f(x) = 0] - \epsilon,$$

such that  $f \in C$  satisfies  $\Pr_{\mathcal{D}^+}[f(x) = 0] < \epsilon$ .

2. If the constraint on  $h \in H$  above is changed to

$$\Pr_{\mathcal{D}^-}[h(x) = 1] < \epsilon \quad \text{and} \quad \Pr_{\mathcal{D}^+}[h(x) = 1] \geq \alpha \max_f \Pr_{\mathcal{D}^+}[f(x) = 1] - \epsilon,$$

such that  $f \in C$  satisfies  $\Pr_{\mathcal{D}^-}[f(x) = 1] < \epsilon$ , then we say that  $\mathcal{A}$   $\alpha$ -heuristically NFP learns  $C$  by  $H$ .

It can be shown that  $\alpha$ -heuristic NFN and NFP learning are equivalent to finding  $\alpha$ -approximation algorithms for two variants of the maximum agreement problems, which we call *maximum negative agreement* (MNA) and *maximum positive agreement* (MPA) respectively. We give the definitions of  $C/H$ -MNA and  $C/H$ -MPA next.

 $C/H$ -MNA

**Input:** Multisets  $\mathcal{P} \subseteq \{0, 1\}^n \times \{1\}$  and  $\mathcal{N} \subseteq \{0, 1\}^n \times \{0\}$ , of positive and negative examples respectively.

**Output:** Hypothesis  $h \in H$  s.t.  $h(y) = 1$  for **all**  $(y, 1) \in \mathcal{P}$ , and

$$|\{(u, 0) \in \mathcal{N} \mid h(u) = 0\}| \geq \max_{g \in C \cap \mathcal{P}} |\{(u, 0) \in \mathcal{N} \mid g(u) = 0\}|,$$

where  $C \cap \mathcal{P}$  contains  $g \in C$  that are consistent on  $\mathcal{P}$ . If no such  $h \in H$  exists, the output can be anything.

 $C/H$ -MPA

**Input:** Same as for  $C/H$ -MNA.

**Output:** Hypothesis  $h \in H$  s.t.  $h(y) = 0$  for **all**  $(y, 0) \in \mathcal{N}$ , and

$$|\{(u, 1) \in \mathcal{P} \mid h(u) = 1\}| \geq \max_{g \in C \cap \mathcal{N}} |\{(u, 1) \in \mathcal{P} \mid g(u) = 1\}|,$$

where  $C \cap \mathcal{N}$  contains  $g \in C$  that are consistent on  $\mathcal{N}$ . If no such  $h \in H$  exists, the output can be anything.

When  $C \equiv H$ , we will just write  $C$ -MNA and  $C$ -MPA.

This paper studies the approximability (resp. non-approximability) of  $C/H$ -MNA and  $C/H$ -MPA for a variety of classes  $C$  and  $H$ . That is, we are interested in determining for which values of  $\alpha$  the following are tractable (resp. hard).

#### $\alpha$ -Approximation of $C/H$ -MNA

**Input:** Same as for  $C/H$ -MNA.

**Output:** Hypothesis  $h \in H$  s.t.  $h(y) = 1$  for **all**  $(y, 1) \in \mathcal{P}$ , and

$$|\{(u, 0) \in \mathcal{N} \mid h(u) = 0\}| \geq \alpha \max_{g \in C \cap \mathcal{P}} |\{(u, 0) \in \mathcal{N} \mid g(u) = 0\}|,$$

where  $C \cap \mathcal{P}$  contains  $g \in C$  that are consistent on  $\mathcal{P}$ . If no such  $h \in H$  exists, the output can be anything.

#### $\alpha$ -Approximation of $C/H$ -MPA

**Input:** Same as for  $C/H$ -MPA.

**Output:** Hypothesis  $h \in H$  s.t.  $h(y) = 0$  for **all**  $(y, 0) \in \mathcal{N}$ , and

$$|\{(u, 1) \in \mathcal{P} \mid h(u) = 1\}| \geq \alpha \max_{g \in C \cap \mathcal{N}} |\{(u, 1) \in \mathcal{P} \mid g(u) = 1\}|,$$

where  $C \cap \mathcal{N}$  contains  $g \in C$  that are consistent on  $\mathcal{N}$ . If no such  $h \in H$  exists, the output can be anything.

If the  $\alpha$ -approximation of  $C/H$ -MNA (resp.  $C/H$ -MPA) is solvable in polynomial time, we say that  $C/H$ -MNA (resp.  $C/H$ -MPA) is approximable within  $\alpha$ . The constants 0 and 1 are in all classes  $C$  we study, so a hypothesis that agrees with all positive (or negative) examples always exists.

#### 1.1. *Concept classes*

We consider the following concept classes over the variable set  $X = \{x_1, \dots, x_n\}$ . Each class contains the constants 0 and 1. With the exception of Ball, all classes are defined over the Boolean domain.

**Monomial** is the set of conjunctions of literals over  $X$ .

**Clause** is the set of disjunctions of literals over  $X$ .

**Halfspace** is the set of functions of the form  $[a_1x_1 + \dots + a_nx_n \geq b]$  where  $a_1, \dots, a_n, b \in \mathbb{R}$ , and  $[E] = 1$  if  $E$  is true,  $[E] = 0$  otherwise.

**Ball** is the set of functions of form  $[(a_1 - x_1)^2 + \dots + (a_n - x_n)^2 \leq \theta]$ , where  $a_1, \dots, a_n, \theta \in \mathbb{R}$ , and  $x_1, \dots, x_n$  take values from  $\{0, 1, -1\}$ .

**Decision List** is the set of functions of the form  $D(x_1, \dots, x_n) = (\ell_1, c_1), \dots, (\ell_m, c_m)$ , where  $\ell_m$  is the constant 1,  $\ell_1, \dots, \ell_{m-1}$  are literals, and  $c_1, \dots, c_m \in \{0, 1\}$ . Then  $D(\mathbf{x}) = c_k$  if  $\ell_1(\mathbf{x}) = \dots = \ell_{k-1}(\mathbf{x}) = 0$  and  $\ell_k(\mathbf{x}) = 1$ .

**$k$ -term DNF** is the set of disjunctions of  $k$  terms (monomials), i.e., functions of the form  $M_1 \vee \dots \vee M_k$  where each  $M_i$  is a Monomial.

**$k$ -clause CNF** is the set of conjunctions of  $k$  clauses.

**$k$ -CNF** is the set of conjunctions of clauses, where each clause contains at most  $k$  literals.

**$k$ -DNF** is the set of disjunctions of monomials, each containing at most  $k$  literals.

**$k$ -term MP** ( $k$ -term multivariate polynomials) is the set of XORs of  $k$  terms (monomials).

**XOR** is the set of linear equations mod 2, i.e., functions of the form  $\sum_{i=1}^n a_i x_i \bmod 2$ , where each  $a_i \in \{0, 1\}$ .

$\cap^k \mathcal{C}$  is the intersection of  $k$  concepts from class  $\mathcal{C}$ , i.e., functions of the form  $f_1 \wedge \dots \wedge f_k$  where each  $f_i \in \mathcal{C}$ .

### 1.2. Previous results

Valiant (1984) showed that  $k$ -CNF-MNA and  $k$ -DNF-MPA can be solved in polynomial time. Since a Monomial is a 1-CNF, and a clause is a 1-DNF, the polynomial-time solvability of Monomial-MNA and Clause-MPA are implied by Valiant's result. Thus Monomial-MNA and Clause-MPA are easier than their MA counterparts, which are NP-hard (Kearns & Li, 1993), and not  $\alpha$ -approximable within some constant  $\alpha$  (Ben-David, Eiron, & Long, 2003; Bshouty & Burroughs, 2002b).

Höfgen, Simon, and Van Horn (1995) proved that it is NP-hard to  $r$ -approximate Halfspace-MPA for any constant  $r > 0$ . Amaldi and Kann (1995) improved this by showing that Halfspace-MPA and Halfspace-MNA cannot be approximated within  $n^{\gamma-1}$  for any  $\gamma > 0$  unless ZPP = NP. Pitt and Valiant (1988) showed that  $n$ -term DNF/Monomial-MPA is not  $c$ -approximable for any constant  $c$ . Subsequent improvements to the non-approximability of MAX INDEPENDENT SET (Håstad, 1996) improves their result as well, and proves that  $n$ -term DNF/Monomial-MPA cannot be approximated within  $n^{\gamma-1}$  or  $|\mathcal{P}|^{\gamma-1}$  unless ZPP = NP. So for these classes, MPA appears harder than MA, which has a  $\frac{1}{2}$ -approximation algorithm. The result for  $n$ -term DNF/Monomial-MPA gives lower bounds of  $n^{\gamma-1}$  and  $|\mathcal{N}|^{\gamma-1}$  for  $n$ -clause CNF/Clause-MNA, by a kind of duality (see Lemma 3).

Blum and Rivest (1988) showed that  $\cap^k$ Halfspace-MNA is as hard as coloring an  $n$ -vertex  $k$ -colorable graph with  $O(k \log n)$  colors. It has not yet been shown whether this coloring problem is NP-hard, or tractable.

### 1.3. Our results

We extend the result of Pitt and Valiant (1988) for Monomial-MPA to  $k$ -term-DNF-MPA, and show that it is not approximable within  $(n/k)^{\gamma-1}$  or  $(|\mathcal{P}|/k)^{\gamma-1}$  for any  $\gamma > 0$  unless ZPP = NP. By the Duality Lemma 3,  $k$ -clause-CNF-MNA is not approximable within

$(n/k)^{\gamma-1}$  or  $(|\mathcal{N}|/k)^{\gamma-1}$ . Although Monomial-MNA is tractable (Valiant, 1984), we show that  $k$ -term-DNF-MNA is not, even for  $k \geq 2$ . Also, for any constant  $\gamma > 0$ ,  $k$ -term-DNF-MNA is not approximable within  $16/17 + \gamma$  when  $k = 2$ , and not approximable within  $21/22 + \gamma$  for larger  $k$ , unless  $P = NP$ .

We extend the result of Amaldi and Kann (1995), and show that  $C/H$ -MPA, for  $C, H \in \{\text{Halfspace, Decision List}\}$  cannot be approximated within  $n^{\gamma-1}$  or  $|\mathcal{P}|^{\gamma-1}$  for any  $\gamma > 0$ , and the MNA versions cannot be approximated within  $n^{\gamma-1}$  or  $|\mathcal{N}|^{\gamma-1}$  unless  $ZPP = NP$ . Under the same complexity assumption, we show that MNA for the intersection of  $k$  Halfspaces is not approximable within  $(n/k)^{\gamma-1}$  or  $(|\mathcal{N}|/k)^{\gamma-1}$ , thus improving the result of Blum and Rivest (1988).

We then give new hardness results for some other classes. We show that unless  $ZPP = NP$ , for any  $\gamma > 0$ , Ball-MNA cannot be approximated within  $n^{\gamma-1}$  or  $|\mathcal{N}|^{\gamma-1}$ , Ball-MPA cannot be approximated within  $n^{\gamma-1}$  or  $|\mathcal{P}|^{\gamma-1}$ , and MNA for the intersection of  $k$  Balls cannot be approximated within  $(n/k)^{\gamma-1}$  or  $(|\mathcal{N}|/k)^{\gamma-1}$ .

For the class of XOR functions, we give a  $\frac{1}{2}$ -approximation algorithm for both XOR-MPA and XOR-MNA. Then we show that there exists a  $c$  such that XOR-MNA cannot be approximated within  $1/2 + 2^{-(\log n)^c}$  and XOR-MPA cannot be approximated within  $2/3 + 2^{-(\log n)^c}$  unless  $NP \subseteq RTIME(n^{O(\log \log n)})$ . We also show that for 2-term-MP (XOR of two monomials), 2-term-MP-MNA cannot be approximated within  $16/17 + \gamma$  for any constant  $\gamma > 0$  unless  $P = NP$ .

Negative results for these problems are summarized in Table 1.

The paper is organized as follows. In Section 2 we give some preliminary results for MNA and MPA, for general concept classes, and give the results from the literature on which we base our work. In Section 3 we give negative results for several specific concept classes. In Section 4 we give positive and negative results for the XOR class.

## 2. Preliminaries

### 2.1. General results for MNA and MPA

In this section we give some general results for MNA and MPA.

Let  $X_m = \{0, 1\}^m$  and  $X = \bigcup_m X_m$ . Let  $C_t$  be a concept class over  $X_t$  and let  $C = \bigcup_t C_t$ . Let  $G$  be an ordered table of functions  $g_{m,i} : X_m \rightarrow \{0, 1\}$  for all  $m \geq 0$  and  $1 \leq i \leq t(m)$ , where  $t : \mathbb{N} \rightarrow \mathbb{N}$ . Let  $G_m = (g_{m,1}, \dots, g_{m,t(m)})$ . Define the concept class

$$C_{t(m)}(G_m) = \{f(g_{m,1}, \dots, g_{m,t(m)}) \mid f \in C_{t(m)}\},$$

and

$$C(G) = \bigcup_{m \geq 0} C_{t(m)}(G_m).$$

We will provide an example of a class  $C(G)$  after we prove the following.

Table 1. Negative results for MNA and MPA.

Negative Results			
Problem	Lower bounds	Condition	Where
Halfspace-MNA	$n^{\gamma-1},  \mathcal{N} ^{\gamma-1}$	ZPP $\neq$ NP	(ak)
Decision List-MNA	$n^{\gamma-1},  \mathcal{N} ^{\gamma-1}$	ZPP $\neq$ NP	Thm. 15
Halfspace-MPA	$n^{\gamma-1},  \mathcal{P} ^{\gamma-1}$	ZPP $\neq$ NP	(ak)
Decision List-MPA	$n^{\gamma-1},  \mathcal{P} ^{\gamma-1}$	ZPP $\neq$ NP	Thm. 15
$\cap^k$ Halfspace-MNA	$(\frac{n}{k})^{\gamma-1}, (\frac{ \mathcal{N} }{k})^{\gamma-1}$	ZPP $\neq$ NP	Thm. 19
Ball-MNA	$n^{\gamma-1},  \mathcal{N} ^{\gamma-1}$	ZPP $\neq$ NP	Thm. 16
Ball-MPA	$n^{\gamma-1},  \mathcal{P} ^{\gamma-1}$	ZPP $\neq$ NP	Thm. 17
$\cap^k$ Ball-MNA	$(\frac{n}{k})^{\gamma-1}, (\frac{ \mathcal{N} }{k})^{\gamma-1}$	ZPP $\neq$ NP	Cor. 20
Clause-MNA	$n^{\gamma-1},  \mathcal{N} ^{\gamma-1}$	ZPP $\neq$ NP	(pvh)
Monomial-MPA	$n^{\gamma-1},  \mathcal{P} ^{\gamma-1}$	ZPP $\neq$ NP	(pvh)
2-term-DNF-MNA	$16/17 + \gamma$	P $\neq$ NP	Thm. 21
$k$ -term-DNF-MNA	$21/22 + \gamma$	P $\neq$ NP	Thm. 22
$k$ -clause-CNF-MNA	$(\frac{n}{k})^{\gamma-1}, (\frac{ \mathcal{N} }{k})^{\gamma-1}$	ZPP $\neq$ NP	Thm. 25
$k$ -term DNF-MPA	$(\frac{n}{k})^{\gamma-1}, (\frac{ \mathcal{P} }{k})^{\gamma-1}$	ZPP $\neq$ NP	Thm. 25
XOR-MNA	$1/2 + 2^{-(\log n)^c}$	(*)	Thm. 29
XOR-MPA	$2/3 + 2^{-(\log n)^c}$	(*)	Thm. 30
2-term-MP-MNA	$16/17 + \gamma$	P $\neq$ NP	Thm. 26

(\*): NP  $\not\subseteq$  RTIME( $n^{O(\log \log n)}$ ).

(ak): (Amaldi & Kann, 1995),

(pvh): (Pitt and Valiant, 1988; Håstad, 1996).

**Lemma 2 (Composition Lemma).** *If  $C/H$ -MNA has an  $\alpha(|\mathcal{N}|)$ - (resp.  $\beta(n)$ -) approximation algorithm that runs in time  $T(n)$  then  $C(G)/H(G)$ -MNA has an  $\alpha(|\mathcal{N}|)$ - (resp.  $\beta(t(n))$ -) approximation algorithm that runs in time  $T(t(n))$ . If  $C/H$ -MPA has an  $\alpha(|\mathcal{P}|)$ - (resp.  $\beta(n)$ -) approximation algorithm that runs in time  $T(n)$  then  $C(G)/H(G)$ -MPA has an  $\alpha(|\mathcal{P}|)$ - (resp.  $\beta(t(n))$ -) approximation algorithm that runs in time  $T(t(n))$ .*

**Proof:** Let  $t = t(n)$  and  $g_i = g_{n,i}$ . We give the proof for MNA. The proof for MPA is similar. Let  $\mathcal{A}(n, \mathcal{P} \cup \mathcal{N})$  be an  $\alpha(|\mathcal{N}|)$ -approximation algorithm for  $C/H$ -MNA that runs in time  $T(n)$ . For  $h \in H_t$ , let  $h_G(x) = h(g_1(x), \dots, g_t(x))$ . For  $x \in \{0, 1\}^n$  let  $x_G = (g_1(x), \dots, g_t(x))$ . Define the following algorithm for  $C(G)/H(G)$ -MNA.

Algorithm  $\mathcal{B}$

Input :  $\mathcal{P} \cup \mathcal{N} \subseteq \{0, 1\}^n \times \{0, 1\}$

Create  $\mathcal{P}_G = \{(y_G, 1) \mid (y, 1) \in \mathcal{P}\} \subseteq \{0, 1\}^t \times \{1\}$

Create  $\mathcal{N}_G = \{(y_G, 0) \mid (y, 0) \in \mathcal{N}\} \subseteq \{0, 1\}^t \times \{0\}$

*Run* $A(t, \mathcal{P}_G \cup \mathcal{N}_G)$ *togeth*  $\in H$   
*Create* $h_G = h(g_1, \dots, g_t)$   
*Return* $h_G$

For each example  $(y, 1) \in \mathcal{P}$ , we have  $h_G(y) = h(g_1(y), \dots, g_t(y))$ , but since  $(g_1(y), \dots, g_t(y), 1)$  is an example in  $\mathcal{P}_G$ , and  $\mathcal{A}$  is an algorithm for  $C/H$ -MNA, we have  $h_G(y) = 1$ . By a similar argument,  $h_G$  agrees with  $k$  examples from  $\mathcal{N}$  if and only if  $h$  agrees with  $k$  examples from  $\mathcal{N}_G$ . Since  $\mathcal{A}$  is an  $\alpha(|\mathcal{N}_G|)$ -approximation algorithm with running time  $T(n)$ , and  $|\mathcal{N}_G| = |\mathcal{N}|$ ,  $\mathcal{B}$  is an  $\alpha(|\mathcal{N}|)$ -approximation algorithm, with running time  $T(t)$ . If  $\mathcal{A}$ 's approximation ratio depends on the dimension  $n$ , then the approximation ratio for  $\mathcal{B}$  depends on  $t(n)$ , the dimension for  $\mathcal{P}_G \cup \mathcal{N}_G$ .  $\square$

As an example, consider the class  $C$  of monotone monomials (monomials that have no negated literals). Let  $G$  be a table of functions  $g_{n,i}$  with  $1 \leq i \leq 2n$  defined by

$$g_{n,i}(x_1, \dots, x_n) = \begin{cases} x_i & \text{if } 1 \leq i \leq n \\ \bar{x}_{i-n} & \text{otherwise.} \end{cases}$$

Let  $f : X_{2n} \rightarrow \{0, 1\}$  be a monotone monomial, and define  $f_G : X_n \rightarrow \{0, 1\}$  by  $f_G(x_1, \dots, x_n) = f(g_{n,1}(x_1, \dots, x_n), \dots, g_{n,2n}(x_1, \dots, x_n)) = f(x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n)$ . Then  $f_G$  is a monomial,  $C^{t(n)}(G)$  for  $t(n) = 2n$  is the class of monomials, and the Composition Lemma states that any  $\alpha$ -approximation algorithm for Monotone-Monomial-MNA gives an  $\alpha$ -approximation algorithm for Monomial-MNA.

Let  $C$  be a concept class over  $\{0, 1\}^n$ . We define the dual class  $C^d = \{f(\mathbf{x}) \mid \bar{f}(\bar{\mathbf{x}}) \in C\}$ , where  $\mathbf{x} = (x_1, \dots, x_n)$  and  $\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_n)$ . Our unconventional use of the term ‘‘dual’’ is borrowed from Pitt and Valiant (1988). Then  $C/H$ -MNA and  $C^d/H^d$ -MPA are related in the expected way:

**Lemma 3 (Duality).**  *$C/H$ -MNA is  $\alpha(|\mathcal{N}|)$ -approximable if and only if  $C^d/H^d$ -MPA is  $\alpha(|\mathcal{P}|)$ -approximable.*

We will also make use of the following trivial result.

**Lemma 4.** *If  $C/H$ -MNA is not  $\alpha$ -approximable then for any  $C' \supseteq C$ ,  $C'/H$ -MNA is not  $\alpha$ -approximable. In particular, if  $C \subset H$  and  $C/H$ -MNA is not  $\alpha$ -approximable then  $H$ -MNA is not  $\alpha$ -approximable.*

For classes that are PAC-learnable we prove the following bound.

**Theorem 5.** *If  $C$  is PAC-learnable from  $H$  then there is a randomized  $\alpha$ -approximation algorithm for  $C/H$ -MNA for*

$$\alpha = \frac{c \log |\mathcal{N}|}{|\mathcal{N}| \log \log |\mathcal{N}|}$$

for any constant  $c$ .



**Proof:** Let  $T = \mathcal{P} \cup \mathcal{N}$  be a training set. Let  $f \in C$  agree with all the examples in  $\mathcal{P}$  and a maximum number of examples in  $\mathcal{N}$ . Let  $\mathcal{N}' \subseteq \mathcal{N}$  be the negative examples that  $f$  agrees with. We now have two cases.

*Case 1.*  $|\mathcal{N}'| \leq 1/\alpha$ . In this case we can use the PAC learning algorithm to find a hypothesis  $h \in H$  consistent with  $\mathcal{P}$  and one example from  $\mathcal{N}$ . That is, we run the algorithm on  $\mathcal{P} \cup \{(y, 0)\}$  for all possible  $(y, 0) \in \mathcal{N}$  with a uniform distribution and error  $\epsilon = 1/(2|\mathcal{P}| + 2)$ . The approximation ratio will be  $1/|\mathcal{N}'| \geq \alpha$ .

*Case 2.*  $|\mathcal{N}'| \geq 1/\alpha$ . In this case we choose  $\alpha|\mathcal{N}'|$  random examples  $\mathcal{N}''$  from  $\mathcal{N}'$  and using the PAC learning algorithm, we find a consistent hypothesis for  $\mathcal{P} \cup \mathcal{N}''$ . If we do find a consistent hypothesis then the ratio is  $|\mathcal{N}''|/|\mathcal{N}'| > |\mathcal{N}''|/|\mathcal{N}'| = \alpha$ . Now the probability of success is at least

$$\begin{aligned} \Pr[\mathcal{N}'' \subseteq \mathcal{N}'] &= (\Pr[x \in \mathcal{N}'])^{|\mathcal{N}''|} \\ &\geq \left(\frac{1}{\alpha|\mathcal{N}'|}\right)^{\alpha|\mathcal{N}'|} \\ &\geq \frac{1}{|\mathcal{N}'|^{O(c)}}. \end{aligned}$$

Therefore the expected time for this case is polynomial.

The algorithm tries Case 2 and if it fails after  $\text{poly}(\mathcal{N})$  trials, it applies Case 1. □

It is easy to see using the techniques of Bshouty and Burroughs (2002a) that when the class is PAC learnable and the VC-dimension of the class is constant then there is a polynomial time algorithm that  $\alpha$ -approximates  $C$ -MNA for any constant  $\alpha$ .

The next result shows that if  $C/H$ -MNA has a  $1 - \beta$ -approximation algorithm then  $C/(\cap^k H)$ -MNA has a  $1 - \beta^k$ -approximation algorithm. Notice that this implies that if  $C/H$ -MNA has an  $\alpha$ -approximation algorithm then for any constant  $\lambda$  there is a  $\hat{H}$  such that  $C/\hat{H}$ -MNA has a  $\lambda$ -approximation algorithm.

**Theorem 6.** *If  $C/H$ -MNA has a  $1 - \beta$ -approximation algorithm then  $C/(\cap^k H)$ -MNA has a  $1 - \beta^k$ -approximation algorithm.*

**Proof:** Let  $\mathcal{A}$  and  $\mathcal{B}$  be  $1 - \beta_1$  and  $1 - \beta_2$ -approximation algorithms for  $C/H_1$ -MNA and  $C/H_2$ -MNA, respectively. We will give a new algorithm that  $1 - \beta_1\beta_2$ -approximates  $C/(H_1 \wedge H_2)$ -MNA. This will prove the result.

Let  $T = \mathcal{P} \cup \mathcal{N}$  be a training set. We run  $\mathcal{A}$  on  $\mathcal{P} \cup \mathcal{N}$  and get  $h_1 \in H_1$ . Then we run  $\mathcal{B}$  on  $\mathcal{P} \cup \mathcal{N}'$  where  $\mathcal{N}' = \{(x, 0) \in \mathcal{N} | h_1(x) \neq f(x)\}$  and get  $h_2 \in H_2$ . Then we output  $h = h_1 \wedge h_2$ .

Let  $m$  be the maximum possible points in  $\mathcal{N}$  that any  $f \in C$  can agree with, while being consistent with  $\mathcal{P}$ . The first hypothesis  $h_1$  agrees with  $\gamma_1 \geq (1 - \beta_1)m$  points from  $\mathcal{N}$ . The second hypothesis  $h_2$  agrees with  $\gamma_2 \geq (1 - \beta_2)(m - \gamma_1)$  more points from  $\mathcal{N}$ . Since  $h_1$

and  $h_2$  are consistent on  $\mathcal{P}$ , the function  $h = h_1 \wedge h_2$  is also consistent on  $\mathcal{P}$  and it agrees with  $\gamma_1 + \gamma_2$  points from  $\mathcal{N}$ . This gives the ratio

$$\begin{aligned} \frac{\gamma_1 + \gamma_2}{m} &\geq \frac{\gamma_1 + (1 - \beta_2)(m - \gamma_1)}{m} \\ &= \frac{(1 - \beta_2)m + \beta_2\gamma_1}{m} \\ &\geq \frac{(1 - \beta_2)m + \beta_2(1 - \beta_1)m}{m} \\ &= 1 - \beta_1\beta_2. \end{aligned}$$

From the theorem, we get this corollary. □

**Corollary 7.** *If  $C/H$ -MNA is  $1 - \beta$ -approximable then  $C/(\cap^k H)$ -MNA has a randomized, polynomial-time algorithm when  $k = 1 + \log |\mathcal{N}| / \log(1/\beta)$ .*

**Proof:** When  $k = 1 + \log |\mathcal{N}| / \log(1/\beta)$  then  $1 - \beta^k > 1 - 1/|\mathcal{N}|$  and the output hypothesis is optimal. □

On the other hand we have the following.

**Theorem 8.** *If  $C/(\cap^k H)$ -MNA has an  $\alpha$ -approximation algorithm then  $C/H$ -MNA has an  $\alpha/k$ -approximation algorithm.*

*If  $C/H$ -MNA has an  $\alpha$ -approximation algorithm then  $(\cap^k C)/H$ -MNA has an  $\alpha/k$ -approximation algorithm.*

**Proof:** Suppose  $C/(\cap^k H)$ -MNA has an  $\alpha$ -approximation algorithm  $\mathcal{A}$ . Let  $T = \mathcal{P} \cup \mathcal{N}$  be a training set. We run  $\mathcal{A}$  and get some hypothesis  $h = h_1 \wedge \dots \wedge h_k$ . Then we choose  $i_0$  that maximizes  $|\{x \in \mathcal{N} | h_{i_0}(x) = 0\}|$ . Since  $\mathcal{A}$  is an  $\alpha$ -approximation algorithm we have  $h(x) = 1$  for all  $(x, 1) \in \mathcal{P}$  and

$$|\{(x, 0) \in \mathcal{N} | h(x) = 0\}| \geq \alpha \max_{f \in \mathcal{C}} |\{(x, 0) \in \mathcal{N} | f(x) = 0\}|.$$

Since

$$|\{(x, 0) \in \mathcal{N} | h(x) = 0\}| \leq k |\{(x, 0) \in \mathcal{N} | h_{i_0}(x) = 0\}|$$

we have

$$|\{(x, 0) \in \mathcal{N} | h_{i_0}(x) = 0\}| \geq \frac{\alpha}{k} \max_{f \in \mathcal{C}} |\{(x, 0) \in \mathcal{N} | f(x) = 0\}|.$$

The result follows.

The second claim of the theorem can be proved in a similar way.  $\square$

As a corollary we have the following.

**Corollary 9.** *If  $C$ -MNA is not  $\alpha$ -approximable then  $(\cap^k C)$ -MNA is not  $k\alpha$ -approximable.*

**Proof:** If  $C$ -MNA is not  $\alpha$ -approximable then by Theorem 8,  $C/(\cap^k C)$ -MNA is not  $k\alpha$ -approximable. Then by Lemma 4,  $(\cap^k C)$ -MNA is not  $k\alpha$ -approximable.  $\square$

Now we give a negative result with the assumption that DNF is not PAC-learnable. This result is implicit in many computational learning theory papers in different forms and settings (see Blum et al., 1994, for example). Learning DNF is still one of the outstanding open problems in computational learning theory. For people that believe that DNF is not learnable these negative results are convincing enough that heuristic learning even for the simplest classes is hard.

**Theorem 10.** *If DNF is not PAC-learnable then for any constants  $c$  and  $\gamma < 1$ , no  $n^{-c}|\mathcal{P}|^{\gamma-1}$ -approximation algorithm for Monomial/ $H$ -MPA exists for any class  $H$ .*

**Proof:** Let  $\alpha = n^{-c}|\mathcal{P}|^{\gamma-1}$ . Suppose for some class  $H$ , Monomial/ $H$ -MPA has an  $\alpha$ -approximation algorithm  $\mathcal{A}$ . We use  $\mathcal{A}$  to PAC-learn DNF as follows. Let  $f = T_1 \vee T_2 \vee \dots \vee T_m$  be the target DNF. We take a sample  $T = \mathcal{N} \cup \mathcal{P}$ , whose size will be specified later. Since  $f = 0$  implies  $T_i = 0$  and

$$1 = \Pr_{x \in \mathcal{P}} [f = T_1 \vee \dots \vee T_m] \leq \sum_{i=1}^m \Pr_{x \in \mathcal{P}} [f = T_i]$$

there is  $T_{i_0}$  that agrees on all the points of  $\mathcal{N}$  and agrees on at least  $|\mathcal{P}|/m$  of the points of  $\mathcal{P}$ . We run  $\mathcal{A}$  and get a function  $T'_1 \in H$  that is consistent with  $\mathcal{N}$  and agrees with  $\alpha/m$  of the points in  $\mathcal{P}$ . We remove the points in  $\mathcal{P}$  that  $T'_1$  agrees with. Let  $\mathcal{P}_1$  be the remaining points of  $\mathcal{P}$ . Then  $|\mathcal{P}_1| \leq (1 - \alpha/m)|\mathcal{P}|$ . We run the algorithm again for  $\mathcal{N} \cup \mathcal{P}_1$  and again get  $T'_2 \in H$  that agrees with all the points in  $\mathcal{N}$  and  $\alpha/m$  of the points in  $\mathcal{P}_1$ . We continue as before. That is, we remove the points in  $\mathcal{P}_1$  that agree with  $T'_2$  and get  $\mathcal{P}_2$ . Then we run again the algorithm on  $\mathcal{N} \cup \mathcal{P}_2$ . Notice that  $|\mathcal{P}_i| \leq (1 - \alpha/m)|\mathcal{P}_{i-1}|$  and therefore after

$$k = \frac{m}{\alpha} \log |\mathcal{P}| = mn^c |\mathcal{P}|^{1-\gamma} \log |\mathcal{P}|$$

iterations we find a consistent hypothesis  $T'_1 \vee \dots \vee T'_k$  for  $T = \mathcal{N} \cup \mathcal{P}$ . By the Occam Theorem (Blumer et al., 1987) it is enough to start from a sample  $T$  of size

$$|T| = \left( \frac{mn^c}{\epsilon} \right)^{\frac{1}{1-\gamma}}.$$

which is polynomial for constants  $c$  and  $\gamma < 1$ .  $\square$

**Corollary 11.** *For  $C \in \{\text{Halfspace, Decision List, } k\text{-term DNF, } k\text{-term MP}\}$  if DNF is not learnable then for any constant  $c$  and  $\gamma < 1$  there is no  $n^{-c}|\mathcal{P}|^{\gamma-1}$ -approximation algorithm for  $C/H$ -MPA for any class  $H$ .*

*For  $C \in \{\text{Clause, } k\text{-clause CNF, Decision List}\}$  if DNF is not learnable then for any constant  $c$  and  $\gamma < 1$  there is no  $n^{-c}|\mathcal{P}|^{\gamma-1}$ -approximation algorithm for  $C/H$ -MNA for any class  $H$ .*

**Proof:** Follows from Lemmas 3 and 4 because  $\text{Monomial} \subset C$ . □

On the other hand, in the next sections we will give a  $n^{1-k}$ -approximation algorithm for  $k$ -clause CNF-MPA and  $k$ -term DNF-MNA and a  $1/2$ -approximation algorithm for XOR-MPA and XOR-MNA.

## 2.2. Proving non-approximability results

In Sections 3 and 4 we give results of the form “Maximization problem  $\Pi$  is not approximable within  $\alpha$  unless complexity class  $C$  has  $O(T)$ -time algorithms.” We prove such a non-approximability result by reducing a  $C$ -hard problem  $\Delta$  to  $\Pi$  such that it remains  $C$ -hard to distinguish instances  $x$  of  $\Pi$  with  $\text{opt}(x) > \xi$  from instances  $x$  with  $\text{opt}(x) < \beta$ , where  $\xi$  and  $\beta$  satisfy  $\alpha\xi > \beta$ .  $T$  measures the time taken for the reduction plus the polynomial time of the  $\alpha$ -approximation algorithm.

Once a result is established for  $\Pi$ , results for other optimization problems  $\Gamma$  can be achieved by demonstrating a polynomial-time, *gap-preserving* reduction from  $\Pi$  to  $\Gamma$ . A reduction  $f$  between maximization problems is gap-preserving if there exists factors  $\xi > \beta$  and  $\gamma < \lambda$  such that for any instance  $x$  of  $\Pi$  mapped to  $f(x)$  of  $\Gamma$ , we have

$$\begin{aligned} \text{if } \text{opt}(x) > \xi &\Rightarrow \text{opt}(f(x)) > \gamma \\ \text{if } \text{opt}(x) < \beta &\Rightarrow \text{opt}(f(x)) < \lambda. \end{aligned}$$

Note that such a reduction proves that if it is hard to distinguish whether an instance  $x$  of  $\Pi$  has  $\text{opt}(x) > \xi$  or  $\text{opt}(x) < \beta$ , then  $\Gamma$  is hard to approximate within  $\lambda/\gamma$ . It is not enough, however, to prove that if  $\Gamma$  has an  $\lambda/\gamma$ -approximation algorithm, then  $\Pi$  has a  $\beta/\xi$ -approximation algorithm. The non-approximability result is based on the ability to solve a decision problem, while an approximability result must give a solution to an optimization problem. For a reduction from  $\Pi$  to  $\Gamma$  to be part of an approximation algorithm for  $\Pi$ , there must be two total mappings, one from instances of  $\Pi$  to instances of  $\Gamma$ , the other from solutions of  $\Gamma$  to solutions of  $\Pi$ . Together, these mappings must achieve the stated approximation ratio for  $\Pi$ . Papadimitriou and Yannakakis (1991) defined L-reductions for this purpose. For our non-approximability results, however, the weaker reductions suffice.

Our results in the next two sections build on non-approximability results proved for the problems MAX INDEPENDENT SET, MAX CUT and MAX- $k$ -CUT. We state the definitions of these problems for reference. Then, in Theorem 12, we list the results on which we base our work.

MAX INDEPENDENT SET

**Input:** Graph  $G = (V, E)$  on  $n = |V|$  vertices.

**Output:** Subset  $I \subseteq V$  of maximum cardinality such that for all  $(u, v) \in E$ , either  $u \notin I$  or  $v \notin I$ .

MAX CUT

**Input:** Multigraph  $G = (V, E)$  on  $n = |V|$  vertices.

**Output:** Subset  $S \subseteq V$  that maximizes the number of edges  $(u, v)$  with exactly one endpoint in  $S$ .

MAX- $k$ -CUT

**Input:** Multigraph  $G = (V, E)$  on  $n = |V|$  vertices.

**Output:** Partition  $V_1, \dots, V_k$  of  $V$  that maximizes the number of edges  $(u, v)$  such that  $u$  and  $v$  are in different sets.

**Theorem 12.**

1. (Håstad, 1996) (MAX INDEPENDENT SET) *For any constant  $\gamma > 0$  there exist functions  $c(n)$  and  $s(n)$  with  $s(n)/c(n) = n^{\gamma-1}$ , such that if a polynomial-time algorithm can distinguish whether a graph's largest independent set has size at least  $c(n)$  or at most  $s(n)$ , then  $ZPP = NP$ .*
2. (Håstad, 1997) (MAX CUT) *There exists a method for generating graphs with  $20m_0 + 22m_1$  edges where  $m_1 \leq m_0$ , such that for some small constants  $\gamma, \xi > 0$ , a maximum cut in this graph has size at least  $(16 - 2\gamma)m_0 + (18 - 2\gamma)m_1$ , or at most  $(15 + \xi)m_0 + (17 + \xi)m_1$  edges, and it is NP-hard to distinguish the two cases.*
3. (Bshouty & Burroughs, 2002a) (XOR-MA) *There exists a method for generating a labeled example set  $I$  such that for some small constants  $c_1, c_2$ , either there exists an XOR function that agrees with at least  $(1 - \frac{1}{(\log n)^{c_1}})|I|$  examples, or no XOR function can agree with more than  $(\frac{1}{2} + \frac{1}{O(2^{(2 \log n)^{c_2}})})|I|$  examples. The two cases cannot be distinguished (and thus XOR-MA cannot be approximated within  $\frac{1}{2} + \frac{1}{2^{(2 \log n)^{c_2}}}$ ) in polynomial time unless  $NP \subseteq RTIME(n^{O(\log \log n)})$ . There is always an XOR function that agrees with  $\frac{1}{2}|I|$  examples.*
4. (Håstad, 1997) MAX CUT cannot be approximated within  $\frac{16}{17} + \gamma$  for any constant  $\gamma > 0$  unless  $P = NP$ .
5. (Kann et al., 1996) *There exist values  $c$  and  $s$  with  $s/c = 1 - \frac{1}{21k-25}$  such that it is NP-hard to distinguish instances of MAX- $k$ -CUT with optimal solutions of size at least  $c$  from those with size at most  $s$ . Thus MAX- $k$ -CUT cannot be approximated within  $1 - \frac{1}{21k-25} + \gamma$  for any constant  $\gamma > 0$  unless  $P = NP$ .*

Item 5 is not explicit in Kann et al. (1996), but it follows from their reduction and item 2 above. Note that it includes the item 4 result when  $k = 2$ .

We also use a result for the MAX  $k$ -COLORABLE INDUCED SUBGRAPH problem.

MAX  $k$ -COLORABLE INDUCED SUBGRAPH

**Input:** Graph  $G = (V, E)$  on  $n = |V|$  vertices.

**Output:** Subset  $V' \subseteq V$  of maximum cardinality such that the subgraph of  $G$  induced by  $V'$  is  $k$ -colorable.

Panconesi and Ranjan (1993) show that when  $k$  is part of the input, MAX  $k$ -COLORABLE INDUCED SUBGRAPH is as hard as MAX INDEPENDENT SET. We show that for every fixed  $k$  MAX  $k$ -COLORABLE INDUCED SUBGRAPH is similarly hard.

**Theorem 13.** *For all  $\gamma > 0$  and all integer constants  $k > 0$  there exist functions  $c'(n)$  and  $s'(n)$  with  $s'(n)/c'(n) = (n/k)^{\gamma-1}$  such that unless  $ZPP = NP$ , no polynomial-time algorithm can distinguish whether an instance of MAX  $k$ -COLORABLE INDUCED SUBGRAPH has a  $k$ -colorable subgraph on at least  $c'(n)$  vertices, or if the largest  $k$ -colorable subgraph has at most  $s'(n)$  vertices.*

**Proof:** We give a reduction from MAX INDEPENDENT SET to MAX  $k$ -COLORABLE INDUCED SUBGRAPH (which looks for  $k$  disjoint independent sets). The reduction is easier to see in the complement graphs, where independent sets become cliques. Given a complement graph  $\bar{G}$ , we simply make  $k$  disjoint copies of it to get a graph  $\bar{G}^k$ . Clearly if  $\bar{G}$  has a clique of size  $\lambda$  then  $\bar{G}^k$  contains  $k$  disjoint cliques on a total of  $k\lambda$  vertices. The converse is also true since any clique in  $\bar{G}^k$  must contain vertices from only a single copy of  $\bar{G}$ .

If  $\bar{G}$  contains  $n$  vertices, then  $\bar{G}^k$  contains  $M = nk$  vertices. Thus distinguishing  $M$ -vertex graphs that have  $k$ -colorable subgraphs on at least  $c'(M) = c(n)k$  vertices from those whose largest  $k$ -colorable subgraphs have at most  $s'(M) = s(n)k$  vertices, is just as hard as distinguishing graphs with independent sets of size at least  $c(n)$  from those graphs with independent sets of size at most  $s(n)$ . The result then follows from Theorem 12, part 1, since  $s'(M)/c'(M) = n^{\gamma-1} = (M/k)^{\gamma-1}$ .  $\square$

The reduction above also holds for some non-constant values of  $k$ . Specifically, if  $k(n)$  is an integral function with  $1 \leq k(n) \leq M^{1-c}$  for some constant  $c > 0$ , then the reduction above remains polynomial-time, and we get the following.

**Corollary 14.** *For all  $\gamma > 0$  and  $1 \leq k(n) < n^{1-c}$  for some  $c > 0$ , there exist functions  $c(n)$  and  $s(n)$  with  $s(n)/c(n) = (n/k(n))^{\gamma-1}$  such that unless  $ZPP = NP$ , no polynomial-time algorithm can distinguish whether an instance of MAX  $k(n)$ -COLORABLE INDUCED SUBGRAPH has a  $k(n)$ -colorable subgraph on at least  $c(n)$  vertices, or if the largest  $k(n)$ -colorable subgraph has at most  $s(n)$  vertices.*

### 3. Negative results

In this section we give lower bounds on the approximability of MPA and MNA for several familiar concept classes. We use the following notation throughout our proofs.

**Notation 1.** Let  $p^{uv} \in \{0, 1\}^n$  have 0s in positions  $u$  and  $v$ , and 1s everywhere else. Similarly let  $p^u$  have a 0 in position  $u$  and 1s everywhere else. Let  $z^{uv} \in \{0, 1\}^n$  have 1s in positions  $u$  and  $v$ , and 0s everywhere else. Let  $z^u$  have a 1 in position  $u$  and 0s everywhere else.

### 3.1. Decision lists, Halfspaces and balls

Amaldi and Kann (1995) showed that Halfspace-MPA and Halfspace-MNA are as hard to approximate as MAX INDEPENDENT SET. The next theorem expands this result to learning Halfspaces from Decision Lists and vice versa.

**Theorem 15.** *For all  $\gamma > 0$ , Decision List-MPA, Decision List/Halfspace-MPA and Halfspace/Decision List-MPA cannot be approximated within  $n^{\gamma-1}$  or  $|\mathcal{P}|^{\gamma-1}$  unless ZPP = NP. For all  $\gamma > 0$  Decision List-MNA, Decision List/Halfspace-MNA and Halfspace/Decision List-MNA cannot be approximated within  $n^{\gamma-1}$  or  $|\mathcal{N}|^{\gamma-1}$ , unless ZPP = NP.*

**Proof:** We use the fact (see Höffgen, Simon, & Van Horn, 1995, for example) that Decision List  $\subseteq$  Halfspace. Then we give a reduction from MAX INDEPENDENT SET to  $C/H$ -MPA, where  $C, H \in \{\text{Decision List, Halfspace}\}$ .

Let  $G = (V, E)$  be an instance of MAX INDEPENDENT SET with  $|V| = n$ . Create

$$\mathcal{N} = \{(\mathbf{0}, 0)\} \cup \{(z^{uv}, 0) \mid (u, v) \in E\} \quad \text{and} \quad \mathcal{P} = \{(z^u, 1) \mid u \in V\}.$$

Let  $S = \{v_{j_1}, \dots, v_{j_k}\} \subseteq V$  be a maximum independent set in  $G$ , and  $C = \{v_{i_1}, \dots, v_{i_\ell}\} = V \setminus S$  be the corresponding vertex cover. Define decision list  $D$  as

$$D(x_1, \dots, x_n) = (x_{i_1}, 0), \dots, (x_{i_\ell}, 0), (x_{j_1}, 1), \dots, (x_{j_k}, 1), (1, 0).$$

Note that  $D$  agrees with  $(\mathbf{0}, 0)$ . Since  $C$  is a vertex cover,  $D$  agrees with all examples  $(z^{uv}, 0)$  from  $\mathcal{N}$ . Furthermore,  $D$  agrees with all examples  $(z^u, 1)$  in  $\mathcal{P}$  where  $u$  is in the independent set  $S$ . Thus  $D$  agrees with  $|S|$  positive examples.

Now let  $H(x_1, \dots, x_n) = [a_1x_1 + \dots + a_nx_n \geq b]$  be a halfspace that is consistent with  $\mathcal{N}$  and agrees with a maximum number of examples from  $\mathcal{P}$ . Let  $I = \{u \mid H \text{ agrees with } (z^u, 1)\}$ . Then  $I$  is an independent set in  $G$ , which we prove by contradiction: Suppose  $u, v \in I$  and  $(u, v) \in E$ . Since  $(z^u, 1)$  and  $(z^v, 1)$  agree with  $H$ , we have  $a_u \geq b$  and  $a_v \geq b$ , which gives  $a_u + a_v \geq 2b$ . Since example  $(\mathbf{0}, 0)$  agrees with  $H$ , we have  $b > 0$  which implies  $a_u + a_v \geq 2b > b$ . But then  $H$  disagrees with negative example  $(z^{uv}, 0)$  – a contradiction. So  $I$  is an independent set in  $G$ , and thus  $H$  agrees with  $|I| \leq |S|$  positive examples.

Thus a Decision List or Halfspace which agrees with all negative examples, is optimal iff it agrees with  $|S|$  positive examples, where  $S$  is the maximum independent set in  $G$ . The results for the MPA problems listed above then follow from Theorem 12, part 1.

Note that halfspace  $H$  we describe above would satisfy the proof even if the inequality in the halfspace function were replaced with a strict inequality. The dual of that class is again Halfspace. Decision List is its own dual class, so the results listed above for MNA follow by the Duality Lemma 3.  $\square$

Balls are formulas of the form  $B(x_1, \dots, x_n) = [(w_1 - x_1)^2 + \dots + (w_n - x_n)^2 \leq \theta]$ , where  $x_1, \dots, x_n$  take values from  $\{0, 1, -1\}$ . On the Boolean domain, Ball and Halfspace are equivalent classes. That is, the ball above, when restricted to the  $\{0, 1\}^n$  domain, is equivalent to the halfspace  $H(x_1, \dots, x_n) = [a_1x_1 + \dots + a_nx_n \geq b]$  for  $w_i = (a_i + 1)/2$  and  $\theta = -b + \sum_i w_i^2$ . Thus the reduction of Theorem 15 gives the following result for Ball over  $\{0, 1, -1\}^n$ .

**Theorem 16.** *For all  $\gamma > 0$ , Ball-MNA cannot be approximated within  $n^{\gamma-1}$  or within  $|\mathcal{N}|^{\gamma-1}$  unless ZPP = NP.*

**Theorem 17.** *For all  $\gamma > 0$ , Ball-MPA cannot be approximated within  $n^{\gamma-1}$  or within  $|\mathcal{P}|^{\gamma-1}$  unless ZPP = NP.*

**Corollary 18.** *For all  $\gamma > 0$ , Halfspace/Ball-MPA and Decision List/Ball-MPA cannot be approximated within  $n^{\gamma-1}$  or within  $|\mathcal{P}|^{\gamma-1}$ , and Halfspace/Ball-MNA and Decision List/Ball-MNA cannot be approximated within  $n^{\gamma-1}$  or within  $|\mathcal{N}|^{\gamma-1}$ , unless ZPP = NP.*

### 3.2. Neural nets: Conjunction of $k$ halfspaces

Combining the result of Amaldi and Kann (1995) and Theorem 8, we have that  $\cap^k$ Halfspace-MNA cannot be approximated within  $kn^{\gamma-1}$  unless ZPP = NP. We improve that slightly in the next Theorem.

**Theorem 19.** *For all  $\gamma > 0$ ,  $\cap^k$ Halfspace-MNA cannot be approximated within  $(n/k)^{\gamma-1}$  or within  $(|\mathcal{N}|/k)^{\gamma-1}$  unless ZPP = NP.*

**Proof:** Let  $G = (V, E)$  be an instance of MAX  $k$ -COLORABLE INDUCED SUBGRAPH. The instance of  $\cap^k$ Halfspace-MNA will be

$$\mathcal{P} = \{(\mathbf{0}, 1)\} \cup \{(z^{uv}, 1) \mid (u, v) \in E\} \quad \text{and} \quad \mathcal{N} = \{(z^u, 0) \mid u \in V\}.$$

Let  $V' \subseteq V$  induce a  $k$ -colorable subgraph with  $k$ -coloring  $\chi : V' \rightarrow \{1, \dots, k\}$ . Let

$$f(x_1, \dots, x_n) = \bigwedge_{i=1}^k [a_1^{(i)}x_1 + \dots + a_n^{(i)}x_n \geq A_i]$$

where  $A_1 = \dots = A_k = -1$  and  $a_u^{(i)} = -2$  if  $u \in V'$  and  $\chi(u) = i$ ,  $a_u^{(i)} = 3$  otherwise. Clearly  $f$  agrees with example  $(\mathbf{0}, 1)$ . Now consider  $(z^{uv}, 1)$ . Since  $u$  and  $v$  cannot be the same color,



we do not have  $a_u^{(i)} = a_v^{(i)} = -2$ , thus  $a_u^{(i)} + a_v^{(i)} \geq 1 \geq A_i$  for all  $i$ , which implies  $f$  agrees with  $(z^{uv}, 1)$ . For the  $(z^u, 0)$  examples, note that if  $u \notin V'$  we have  $a_u^{(i)} = 3 \not\leq A_i$  for all  $i$ , and thus  $(z^u, 0)$  does not agree with  $f$ . For  $u \in V'$  there exists an  $i$  such that  $\chi(u) = i$  and since  $a_u^{(i)} = -2 < A_i$ , the example  $(z^u, 0)$  does agree with  $f$ . So  $f$  agrees with  $|V'|$  examples.

Now suppose  $g(x_1, \dots, x_n) = \bigwedge_{i=1}^k [b_1^{(i)}x_1 + \dots + b_n^{(i)}x_n \geq B_i]$  agrees with all examples in  $\mathcal{P}$  and a maximum number of examples in  $\mathcal{N}$ . Let  $\tilde{V} = \{u \mid g(z^u) = 0\}$  and  $\chi : \tilde{V} \rightarrow \{1, \dots, k\}$  be defined by  $\chi(u) = i$  if and only if  $i$  is the smallest index such that  $b_u^{(i)} < B_i$ . Now suppose  $u, v \in \tilde{V}$  and  $(u, v) \in E$  and  $\chi(u) = \chi(v) = i$ . This implies that  $b_u^{(i)} < B_i$  and  $b_v^{(i)} < B_i$ . Since  $g$  agrees with  $(\mathbf{0}, 1)$  we must also have  $B_i < 0$ . But then  $b_u^{(i)} + b_v^{(i)} < 2B_i < B_i$ , which contradicts  $g$ 's agreement with example  $(z^{uv}, 1)$ . Thus  $\chi$  is a valid coloring of  $V'$ , and graph  $G$  has a  $k$ -colorable subgraph of size equal to the negative agreement.

So  $G$  has a  $k$ -colorable subgraph of size  $t$  if and only if there is a function in  $\cap^k \text{Halfspace}$  consistent with  $\mathcal{P}$  that agrees with  $t$  examples in  $\mathcal{N}$ . The result follows from Theorem 14.  $\square$

**Corollary 20.** *For all  $\gamma > 0$ ,  $\cap^k \text{Ball-MNA}$  cannot be approximated within  $(n/k)^{\gamma-1}$  or within  $(|\mathcal{N}|/k)^{\gamma-1}$  unless  $ZPP = NP$ .*

### 3.3. DNF and CNF

For  $k$ -term DNF and  $k$ -clause CNF we have the following.

**Theorem 21.** *For all  $\gamma > 0$ , 2-term-DNF-MNA and 2-clause CNF-MPA cannot be approximated within  $\frac{16}{17} + \gamma$  unless  $P = NP$ .*

**Proof:** We consider 2-term-DNF-MNA. Let  $G = (V, E)$  be an instance of MAX CUT. Create

$$\mathcal{P} = \{(p^u, 1) \mid u \in V\} \quad \text{and} \quad \mathcal{N} = \{(p^{uv}, 0) \mid (u, v) \in E\}.$$

Let  $S \subseteq V$  cut  $k$  edges in  $G$ . Let  $f_S(x_1, \dots, x_n) = m_1 \vee m_2$  where

$$m_1 = \prod_{u \in S} x_u, \quad m_2 = \prod_{u \notin S} x_u.$$

Then since each  $x_u$  appears in just one monomial,  $f_S$  agrees with all the  $(p^u, 1)$  examples. Furthermore,  $f_S$  agrees with precisely those examples  $(p^{uv}, 0)$  for which  $x_u$  and  $x_v$  appear in different monomials. So  $f_S$  agrees with  $k$  examples in  $\mathcal{N}$ .

Now, let  $f(x_1, \dots, x_n) = M_1 \vee M_2$  be a 2-term DNF that agrees with all examples in  $\mathcal{P}$ , and a maximum number of examples in  $\mathcal{N}$ . Let  $S_f = \{u \mid M_1(p^u) = 1\}$ . Suppose  $(p^{uv}, 0)$  agrees with  $f$ . Since  $f$  agrees with  $(p^u, 1)$  and  $(p^v, 1)$ , we have  $M_i(p^u) = 1$  and  $M_j(p^v) = 1$  for  $i, j \in \{1, 2\}$ . Since  $M_i(p^{uv}) = 0 = M_j(p^{uv})$ , and  $p^{uv}$  differs from  $p^u$

only in position  $v$ , and similarly,  $p^{uv}$  and  $p^v$  differ only in position  $u$ , this implies that  $M_i$  contains  $x_v$  while  $M_j$  contains  $x_u$ . If  $M_i$  also contains  $x_u$  (likewise,  $M_j$  also contains  $x_v$ ), then  $M_i(p^u) = 0$  ( $M_j(p_v) = 0$ ), a contradiction. So  $i \neq j$  and  $(u, v)$  is cut by  $S_f$ . Thus if  $f$  agrees with  $k$  examples from  $\mathcal{N}$ , there is a cut of size  $k$  in  $G$ .

So a 2-term DNF agrees with  $k$  negative examples if and only if  $G$  has a cut of size  $k$ . The result follows from the hardness of MAX CUT.  $\square$

The proof above extends to general  $k$  using the MAX- $k$ -CUT result of Kann et al. (1996). This would prove that  $k$ -term DNF-MNA and  $k$ -clause CNF-MPA cannot be approximated within  $1 - \frac{1}{2^{1k-25}} + \gamma$  unless  $P = NP$ . The lower bound approaches 1 as  $k$  increases. This seems intuitively correct, since the more terms a DNF has, the more flexible it is in adapting to the character of the sample. However, we note that in the proof of Kann et al. (1996),  $k$  always remains polynomially smaller than the dimension  $n$ . For such values of  $k$ , we provide the following lower bound that is independent of  $k$ .

**Theorem 22.** *For all constants  $\gamma, \delta > 0$  and every even  $k < n^{1-\delta}$ ,  $k$ -term-DNF-MNA and  $k$ -clause CNF-MPA cannot be approximated within  $\frac{21}{22} + \gamma$  unless  $P = NP$ .*

**Proof:** We give a reduction from MAX CUT to  $k$ -term-DNF-MNA. Let  $G = (V, E)$  be an instance of MAX CUT as described in Theorem 12, part 2. We will create examples  $(p, b)$  where  $p$  will have length  $|V|^{\frac{k}{2}}$ . We will view  $p$  as being the concatenation of  $\frac{k}{2}$  blocks, each of length  $|V|$ . Let  $\mathbf{1}^{(j)} \in \{0, 1\}^{\frac{k}{2}|V|}$  have 1s in all block  $j$  positions, and 0s elsewhere (i.e.,  $\mathbf{1}^{(1)} = 1^{|V|}0^{|V|} \dots 0^{|V|}$ ,  $\mathbf{1}^{(2)} = 0^{|V|}1^{|V|}0^{|V|} \dots 0^{|V|}$ , etc.). Let  $p_u^{(j)}$  have 1s in all block  $j$  positions except for a 0 in position  $u$ , and have 0s in all other blocks. Similarly, let  $p_{uv}^{(j)}$  have 1s in all block  $j$  positions except for 0s in positions  $u$  and  $v$ . All blocks other than  $j$  contain 0s. Finally, let  $\mathbf{0} = 0^{\frac{k}{2}|V|}$  be the zero vector. Create the following example set:

$$\begin{aligned} \mathcal{P} &= \{(\mathbf{1}^{(j)}, 1) \mid j = 1, \dots, k/2\} \cup \{(p_u^{(j)}, 1) \mid j = 1, \dots, k/2, u \in V\}, \\ \mathcal{N} &= \{(p_{uv}^{(j)}, 0) \mid j = 1, \dots, k/2, (u, v) \in E\} \cup \{(\mathbf{0}, 0)\}. \end{aligned}$$

Example  $(\mathbf{0}, 0)$  appears  $\lambda = \frac{k}{2}(5m_0 + 5m_1)$  times. All other examples appear once.

Let  $S$  be a cut of  $\beta$  edges in  $G$ , and define

$$f(x_{1,1}, \dots, x_{|V|, \frac{k}{2}}) = \bigvee_{j=1}^{k/2} \left( \prod_{u \in S} x_{u,j} \vee \prod_{u \notin S} x_{u,j} \right).$$

Then  $f$  agrees with all the positive examples and the  $(\mathbf{0}, 0)$  examples. It agrees with  $(p_{uv}^{(j)}, 0)$  iff  $(u, v) \in E$  is cut by  $S$ . Thus  $f$  agrees with all the positive examples and  $\lambda + \frac{k}{2}\beta$  negative examples.

Now, let  $g = M_1 \vee \dots \vee M_k$  agree with all the positive examples and an optimal number of negative examples. If  $g$  does not agree with  $(\mathbf{0}, 0)$ , then it agrees with at most  $\frac{k}{2}|E|$  negative examples.

Assume now that  $g$  agrees with  $(\mathbf{0}, 0)$ , which implies that each  $M_i$  contains at least one positive literal. If some  $M_i$  contains positive literals from different blocks, then, since the 1s in each example vector are confined to a single block,  $M_i$  is 0 on all the example vectors, and is thus redundant. So assume w.l.o.g. that the positive literals in each  $M_i$  are from the same block. We will say that  $M_i$  represents block  $j$  if  $M_i$ 's positive literals are from block  $j$ . Note that  $M_i(p^{(j)}) = 0$  if  $M_i$  does not represent block  $j$ . In order to agree with all  $(p_u^{(j)}, 1)$ , each block  $j$  must be represented by at least one  $M_i$ . If just one  $M_i$  represents block  $j$ , then  $M_i$  contains at least one positive literal  $x_{u,j}$ , and we have  $g(p_u^{(j)}) = M_i(p_u^{(j)}) = 0$ , a contradiction. Therefore, each block is represented by at least two  $M_i$ s. Since there are  $k/2$  blocks, and  $k$   $M_i$ s, each block is represented by exactly two  $M_i$ s.

Now let's concentrate on block  $j$ . Suppose it's represented by  $M_i$  and  $M_\ell$ . A positive literal  $x_{u,j}$  cannot be in both monomials or we have  $M_i(p_u^{(j)}) = M_\ell(p_u^{(j)}) = g(p_u^{(j)}) = 0$ . So each positive, block  $j$  literal is in at most one of  $M_i, M_\ell$ . If  $x_{u,j}$  appears in neither monomial, it can be added to  $M_i$  without affecting the positive agreement (we still have  $m_\ell(p_u^{(j)}) = 1$ ), but possibly increasing negative agreement. So assume w.l.o.g. that each positive literal  $x_{u,j}$  appears in exactly one of  $M_i, M_\ell$ . Then  $g(p_{uv}^{(j)}) = 0$  if and only if  $x_{u,j}$  and  $x_{v,j}$  appear in different monomials. The number of agreements with the  $(p_{uv}^{(j)}, 0)$  examples thus gives the size of a cut in  $G$  by setting  $S = \{u \mid x_{u,j} \in M_i\}$ . Thus  $g$  can agree with at most  $\lambda + \frac{k}{2}\beta$  examples. Note that since  $\beta \geq 15m_0 + 17m_1$  and  $\lambda = \frac{k}{2}(5m_0 + 5m_1)$ , this is at least  $\frac{k}{2}(20m_0 + 22m_1)$ , which is the maximum number of negative examples that a  $k$ -term DNF can agree with if it disagrees with  $(\mathbf{0}, 0)$ .

So  $G$  has a cut of size  $\beta$  if and only if there exists a  $k$ -term DNF that agrees with all the positive examples, and  $\lambda + \frac{k}{2}\beta$  negative examples. By Theorem 12, part 2, this implies that it is NP-hard to  $\alpha$ -approximate  $k$ -term DNF-MNA if

$$\frac{k}{2}((20 + \xi)m_0 + (22 + \xi)m_1) < \alpha \frac{k}{2}((21 - 2\gamma)m_0 + (23 - 2\gamma)m_1),$$

i.e., if  $\alpha < \frac{21}{22} + \gamma'$ . The vector dimension is  $n = \frac{k}{2}|V|$ , and for this to be a polynomial reduction, we require  $k < n^{1-\delta}$  for some  $\delta > 0$ .  $\square$

We now show the following positive result.

**Theorem 23.** *There is an  $n^{1-k}$ -approximation algorithm for  $k$ -term DNF-MNA and  $k$ -clause CNF-MPA.*

The proof will use the following Lemma. Define MonomialUG to be the set  $\{T \vee g \mid T \in \text{Monomial}, g \in G\}$ .

We first prove

**Lemma 24.** *Let  $G$  be a set of polynomial number of functions. Then (MonomialUG)-MNA is in P.*

**Proof:** Let  $\mathcal{P} \cup \mathcal{N}$  be an instance for MonomialUG-MNA. Suppose  $T \vee g$  is the optimal function. Let  $\mathcal{P}_g = \{(x, 1) \in \mathcal{P} \mid g(x) = 1\}$  and  $T_{\max}$  is the largest possible term that is

consistent with  $\mathcal{P} \setminus \mathcal{P}_g$ . Notice first that since  $T$  is consistent with  $\mathcal{P} \setminus \mathcal{P}_g$  we must have  $T_{\max} \Rightarrow T$ . If for some  $(y, 0) \in \mathcal{N}$ ,  $(T \vee g)(y) = 0$  then  $(T_{\max} \vee g)(y) = 0$  and therefore  $T_{\max} \vee g$  is also optimal. Therefore, the algorithm can just exhaustively search  $T_{\max} \wedge g$  as follows.

1. For all  $g \in G$
2. Define  $\mathcal{P}_g = \{(x, 1) \in \mathcal{P} \mid g(x) = 1\}$ .
3. Find the maximal possible term  $T_{\max}$  consistent with  $\mathcal{P} \setminus \mathcal{P}_g$ .
4.  $W \leftarrow \{g \vee T_{\max}\}$
5. Find an  $h \in W$  that minimizes the error.

This algorithm runs in polynomial time when  $|G|$  is polynomial. □

Now we are ready to prove the theorem.

**Proof of Theorem 23:** Let  $k$ -Clause be the set of all clauses that contain at most  $k$  literals. Notice that every  $k$ -term DNF,  $f = T_1 \vee T_2 \vee \dots \vee T_k$  can be written as

$$f = \bigwedge_{l_2 \in T_2, \dots, l_k \in T_k} (T_1 \vee l_2 \vee l_3 \vee \dots \vee l_k)$$

where  $l_i$  is a literal in  $T_i$  for  $i = 2, 3, \dots, k$ . Therefore,

$$k\text{-term DNF} \subseteq \bigcap^{n^{k-1}} (\text{Monomial} \cup (k-1)\text{-Clause}).$$

By Lemma 24 (Monomial  $\cup$   $(k-1)$ -Clause)-MNA is in  $P$  (so it has an  $\alpha$ -approximation algorithm for  $\alpha = 1$ ). By Theorem 8 we have  $(k\text{-term DNF}/\text{Monomial} \cup (k-1)\text{-Clause})\text{-MNA}$  has an  $n^{1-k}$ -approximation algorithm. Since  $(\text{Monomial} \cup (k-1)\text{-Clause}) \subset k\text{-term DNF}$ , the same algorithm is an  $n^{1-k}$ -approximation algorithm for  $k\text{-term-DNF-MNA}$ .

The result for  $k$ -clause CNF-MPA follows from Lemma 3. □

**Theorem 25.** *For all  $\gamma > 0$ ,  $k$ -clause CNF-MNA and  $k$ -term DNF-MPA cannot be approximated within  $(n/k)^{\gamma-1}$  unless  $ZPP = NP$ .*

**Proof:** This is similar to the proofs for  $\cap^k$ Ball and  $\cap^k$ Halfspace. We use the sample  $\mathcal{P} = \{(z^{uv}, 1) \mid (u, v) \in E\}$ ,  $\mathcal{N} = \{(z^u, 0) \mid u \in V\}$ . If the graph has  $k$  independent sets  $S_1, \dots, S_k$ , then the  $k$ -term CNF

$$\left( \bigvee_{u \notin S_1} x_u \right) \wedge \dots \wedge \left( \bigvee_{u \notin S_k} x_u \right)$$

agrees with all  $(z_{uv}, 1) \in \mathcal{P}$  because each clause contains either literal  $x_u$  or  $x_v$ . If  $u \in S_i$  for some  $i$ , then the  $i^{\text{th}}$  clause has only positive literals which do not include  $x_u$ , and is zero

on  $p^u$ . If  $u \notin S_i$  for all  $i$ , then all clauses contain  $x_u$  and thus the function is 1 on  $x_u$ . So the CNF agrees with all examples in  $\mathcal{P}$  and  $|V_1| + \dots + |V_k|$  examples in  $\mathcal{N}$ .

Now let  $f = c_1 \wedge \dots \wedge c_k$  agree with all examples in  $\mathcal{P}$  and a maximum number of examples in  $\mathcal{N}$ . Let  $V_i$  contain all vertices  $u$  such that  $c_i(z^u) = 0$ . Suppose  $u, v \in V_i$  where  $(u, v) \in E$ . Then since  $c_i(z^u) = 0$ , the literals in  $c_i$  are among  $\{\bar{x}_u, x_\ell : \ell \neq u\}$ . Since we also have  $c_i(z^v) = 0$ , the literals in  $c_i$  are among  $\{x_\ell : \ell \notin \{u, v\}\}$ . But then  $c_i(z^{uv}) = 0$ , a contradiction. So  $V_1, \dots, V_k$  are  $k$  disjoint independent sets whose total cardinality equals the total negative agreement of  $f$ .

So there is a  $k$ -clause CNF that agrees with all positive and  $t$  negative examples if and only if  $G$  has a  $k$ -colorable subgraph on  $t$  vertices. The result follows from Theorem 14.  $\square$

### 3.4. 2-term MP

**Theorem 26.** *For any  $\gamma > 0$ , 2-term-MP-MNA cannot be approximated within  $\frac{16}{17} + \gamma$  unless  $P = NP$ .*

**Proof:** Let  $G = (V, E)$  be an instance of MAX CUT. The instance of 2-term-MP-MNA will be

$$\mathcal{P} = \{(p^u, 1) \mid u \in V\} \quad \text{and} \quad \mathcal{N} = \{(p^{uv}, 0) \mid (u, v) \in E\}.$$

Let  $S \subseteq V$  cut  $k$  edges in  $G$  and let  $f_S(x_1, \dots, x_n) = \prod_{u \in S} x_u \oplus \prod_{u \notin S} x_u$ . Then  $f_S$  agrees with all  $(p^u, 1)$  since  $f_S$  contains one monotone monomial with  $x_u$  and one without. Furthermore, for each  $(u, v) \in E$  cut by  $S$ , one monomial of  $f_S$  contains  $x_u$  while the other contains  $x_v$ , so  $f_S$  agrees with  $(p^{uv}, 0)$ . Thus  $f_S$  agrees with  $k$  examples in  $\mathcal{N}$ .

Now, let  $f(x_1, \dots, x_n) = M_0 \oplus M_1$  agree with all examples in  $\mathcal{P}$  and a maximum number of examples in  $\mathcal{N}$ . No literal can appear in both monomials. If  $x_u$  is in both, then  $M_0(p^u) = M_1(p^u) = 0$ . If  $\bar{x}_u$  is in both, then  $M_0(p^w) = M_1(p^w) = 0$  for each  $w \neq u$ . Both cases contradict  $f$ 's agreement with  $\mathcal{P}$ . Suppose now that  $x_u$  appears in neither monomial. Since  $f$  agrees with  $(p^u, 1)$ , w.l.o.g. let  $M_0(p^u) = 0$  and  $M_1(p^u) = 1$ . Let  $M'_0 = M_0 x_u$  (i.e., add literal  $x_u$  to  $M_0$ ) and set  $f' = M'_0 \oplus M_1$ . This change does not affect the agreement with  $\mathcal{P}$ . Suppose it harms the negative agreement, i.e., for some edge  $(a, b)$ , we have  $f(p^{ab}) = 0$  but  $f'(p^{ab}) = 1$ . Since  $M'_0 \Rightarrow M_0$ , this implies  $M_0(p^{ab}) = 1$  and  $M'_0(p^{ab}) = 0$ . So  $p^{ab}$  has a 0 in position  $u$  and is thus  $p^{uv}$  for some  $v$ . But since neither  $M_0$  nor  $M_1$  contained  $x_u$ , we have  $M_1(p^v) = M_1(p^{uv}) = 1 = M_0(p^v) = M_0(p^{uv})$ , which contradicts  $f$ 's agreement with  $(p^v, 1)$ . So each  $x_u$  appears in exactly one monomial.

Let  $S_f = \{u \mid M_1(p^u) = 1\}$ . Suppose  $u, v \in S_f$ . Then  $M_1(p^u) = M_1(p^v) = 1$  and  $M_0(p^u) = M_0(p^v) = 0$ . This implies  $M_1$  contains neither  $x_u$  nor  $x_v$ , which in turn implies that  $M_0$  contains both  $x_u$  and  $x_v$ . Then  $M_1(p^{uv}) = 1$  and  $M_0(p^{uv}) = 0$  and  $(p^{uv}, 0)$  does not agree with  $f$ . By symmetry, this is also true if  $u, v \notin S$ . Now suppose  $u \in S$  and  $v \notin S$ . Then  $M_1(p^u) = 1$ ,  $M_1(p^v) = 0$ ,  $M_0(p^u) = 0$  and  $M_0(p^v) = 1$ , which implies  $x_u$  is in  $M_0$  and  $x_v$  is in  $M_1$ . Thus  $M_1(p^{uv}) = M_0(p^{uv}) = 0$  and  $(p^{uv}, 0)$  agrees with  $f$ . So the number of edges cut by  $S_f$  is exactly the number of examples in  $\mathcal{N}$  that agree with  $f$ .

So  $G$  has a cut of size  $k$  if and only if there exists a 2-term-MP that agrees with all examples in  $\mathcal{P}$  and  $k$  examples in  $\mathcal{N}$ . Therefore 2-term-MP-MNA is at least as hard as MAX CUT. The result follows from Theorem 12, part 2.  $\square$

#### 4. XOR

In this section we give upper and lower bounds for XOR-MNA, and XOR-MPA.

**Theorem 27.** *XOR-MNA and XOR-MPA are  $\frac{1}{2}$ -approximable.*

**Proof:** We give the proof for XOR-MNA. XOR-MPA is similar.

For an instance  $\mathcal{P} \cup \mathcal{N}$  of XOR-MNA, a valid hypothesis is either the constant 1 function, or an XOR formula  $h(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i \pmod{2}$ , which is fully defined by its coefficients  $a_1, \dots, a_n \in \{0, 1\}$ . Each example  $(\mathbf{y}, b) \in \mathcal{P} \cup \mathcal{N}$  for  $\mathbf{y} = (y_1, \dots, y_n)$  and  $y_1, \dots, y_n, b \in \{0, 1\}$ , puts a linear constraint  $c_{\mathbf{y}}(a_1, \dots, a_n) : \sum_{i=1}^n a_i y_i = b \pmod{2}$  on these coefficients. The following algorithm finds a hypothesis that satisfies all the constraints given by  $\mathcal{P}$ , and at least half the optimal number of constraints given by  $\mathcal{N}$ .

Input:  $\mathcal{P} \cup \mathcal{N}$  given as sets  $L_{\mathcal{P}}, L_{\mathcal{N}}$  of linear constraints

- 1 If  $L_{\mathcal{P}}$  is inconsistent,
- 2 Return the constant function  $h(x_1, \dots, x_n) = 1$ .
- 3 Else
- 4 Let  $r$  be the rank of  $L_{\mathcal{P}}$
- 5 Find  $S_{\mathcal{P}} = \{a_j = s_j(t_1, \dots, t_{n-r}) \mid j = 1, \dots, n\}$ , a general solution for  $L_{\mathcal{P}}$  in the Boolean parameters  $t_1, \dots, t_{n-r}$
- 6 Build  $L'_{\mathcal{N}} = \{e \in L_{\mathcal{N}} \mid L_{\mathcal{P}} \cup \{e\} \text{ is feasible.}\}$
- 7 Replace each equation  $e(a_1, \dots, a_n) \in L'_{\mathcal{N}}$  with an equation  $e'(t_1, \dots, t_{n-r})$  by substituting  $s_j(t_1, \dots, t_{n-r})$  in place of  $a_j$  for  $j = 1, \dots, n$ .  
(Treat  $L'_{\mathcal{N}}$  as a multiset. Do not delete duplicates).
- 8 For  $i \in \{1, \dots, (n-r)\}$  do
- 9 Let  $E_i \subseteq L'_{\mathcal{N}}$  be the multiset of equations from  $L'_{\mathcal{N}}$  whose satisfiability depends only on parameter  $t_i$ .
- 10 Choose  $b \in \{0, 1\}$  such that at least half the equations in  $E_i$  are satisfied by setting  $t_i = b$
- 11 Replace  $t_i$  with  $b$  in all the equations in  $L'_{\mathcal{N}} \cup S_{\mathcal{P}}$
- 12 Return  $h(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i$ , for  $a_1, \dots, a_n$  given by  $S_{\mathcal{P}}$ .

The algorithm first checks that there are no contradictions among the constraints given by the examples in  $\mathcal{P}$ . If  $L_{\mathcal{P}}$  is inconsistent, then no equation  $\sum_{i=1}^n a_i x_i \pmod{2}$  can agree with all examples in  $\mathcal{P}$ . In this case, the only solution is the constant 1 function. This is the optimal solution.

Since the constraints are linear, Gaussian elimination can be used to check the consistency of  $L_{\mathcal{P}}$ , as well as to find its reduced row echelon form, from which a parametrized solution  $S_{\mathcal{P}}$  can be extracted.

If an equation  $e \in L_{\mathcal{N}}$  arising from an example  $(\mathbf{y}, 0) \in \mathcal{N}$  is contradicted by the constraints in  $L_{\mathcal{P}}$  (that is,  $L_{\mathcal{P}} \cup \{e\}$  is infeasible), then no hypothesis that satisfies all the examples in  $\mathcal{P}$  can also agree with  $(\mathbf{y}, 0)$ . We remove all such  $e$  at line 6. The number of remaining equations from  $L_{\mathcal{N}}$  gives an upper bound on the number of negative examples from  $\mathcal{N}$  that a hypothesis can agree with, while also agreeing with all examples in  $\mathcal{P}$ .

At line 7, each equation in  $L'_{\mathcal{N}}$  is subjected to the constraints of  $L_{\mathcal{P}}$  by substituting each variable  $a_j$  with its parametrized equivalent  $s_j(t_1, \dots, t_{n-r})$  from  $S_{\mathcal{P}}$ . It should be clear that any Boolean assignment to the parameters  $t_1, \dots, t_{n-r}$  will give (via  $S_{\mathcal{P}}$ ) a Boolean assignment to  $a_1, \dots, a_n$  that will satisfy all the equations in  $\mathcal{P}$ . At lines 8–11, the algorithm sets a parameter  $t_i$  to 0 or 1, depending on which value will satisfy the larger number of equations in  $E_i$  (equations from  $L'_{\mathcal{N}}$  that contain the single parameter  $t_i$ . Since we replace parameters with their assigned value on each iteration, every equation in  $L'_{\mathcal{N}}$  will be placed in  $E_i$  on some iteration  $i$ ). At each iteration  $i$ , at least as many equations are satisfied by the choice of  $t_i$ , as are left unsatisfied by it. Therefore, by line 12, at least half of the equations in  $L'_{\mathcal{N}}$  (that is, at least half the optimal number) are satisfied.

The hypothesis  $h(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i \bmod 2$  returned at line 12 agrees with an example  $(\mathbf{y}, b) \in \mathcal{P} \cup \mathcal{N}$  for  $\mathbf{y} = (y_1, \dots, y_n)$  if and only if the coefficients  $a_1, \dots, a_n$  satisfy the constraint  $c(a_1, \dots, a_n) : \sum_{i=1}^n a_i y_i = b$ . So  $h$  agrees with all  $(\mathbf{y}, 1) \in \mathcal{P}$  and at least half the optimal number of  $(\mathbf{y}, 0) \in \mathcal{N}$ . Therefore, this is a 1/2-approximation algorithm for XOR-MNA.  $\square$

We now may apply Corollary 7 to get the following.

**Theorem 28.** *XOR/(\(\cap^k\) XOR)-MNA and XOR/(\(\cup^k\) XOR)-MPA for  $k = 1 + \log |\mathcal{N}|$  and  $k' = 1 + \log |\mathcal{P}|$  are in P.*

**Theorem 29.** *For some small constant  $c$ , XOR-MNA cannot be approximated within  $\frac{1}{2} + \frac{1}{2^{(\log n)^c}}$  unless  $NP \subseteq RTIME(n^{O(\log \log n)})$ .*

**Proof:** We reduce XOR-MA to XOR-MNA. Let  $I \subseteq \{0, 1\}^n \times \{0, 1\}$  be an instance of XOR-MA. We create an instance  $\mathcal{P} \cup \mathcal{N} \subseteq \{0, 1\}^{n+1} \times \{0, 1\}$  of XOR-MNA as follows. For each example  $(a, b)$  in  $I$ , put an example  $(ab, 0)$  in  $\mathcal{N}$  (that is, append label  $b$  to the vector  $a$  and label the resulting vector 0). Then set  $\mathcal{P} = \{(0^n 1, 1)\}$ . Note that any XOR function  $\ell(x_1, \dots, x_{n+1})$  consistent with  $\mathcal{P}$  must be  $\ell(x_1, \dots, x_{n+1}) = f(x_1, \dots, x_n) \oplus x_{n+1}$  for some other XOR function  $f$ . Now, for each example  $(ab, 0)$  that agrees with  $\ell$ , we have  $f(x_1, \dots, x_n) = b$ , and for each example  $(ab, 0)$  that does not agree with  $\ell$ , we have  $f(x_1, \dots, x_n) \neq b$ . Thus an  $\alpha$ -approximation algorithm for XOR-MNA gives an  $\alpha$ -approximation algorithm for XOR-MA (find  $\ell$  and return  $f$ ). The result then follows from the non-approximability of XOR-MA, Theorem 12, item 3.  $\square$

**Theorem 30.** *For some small constant  $c$ , XOR-MPA cannot be approximated within  $\frac{2}{3} + \frac{1}{2^{(\log n)^c}}$  unless  $NP \subseteq RTIME(n^{O(\log \log n)})$ .*

**Proof:** We give a reduction from XOR-MA to XOR-MPA. Let  $I \subseteq \{0, 1\}^n \times \{0, 1\}$  be an instance of XOR-MA as described in Theorem 12, item 3. We create an instance  $\mathcal{P} \cup \mathcal{N} \subseteq \{0, 1\}^{n+1} \times \{0, 1\}$  of XOR-MPA as follows. For each  $(a, b) \in I$ , we put an example  $(a\bar{b}, 1)$  in  $\mathcal{P}$ . We add to  $\mathcal{P}$ ,  $\frac{1}{2}|I|$  copies of example  $(0^n 1, 1)$ . We set  $\mathcal{N} = \{(0^{n+1}, 0)\}$ . Now, let  $f(x_1, \dots, x_{n+1})$  be an XOR function that agrees with  $(0^{n+1}, 0)$  and a maximum number of the positive examples. Without loss of generality,  $f$  is not a constant function. If  $f$  disagrees with example  $(0^n 1, 1)$ , then it agrees with at most  $|I|$  examples. On the other hand, if  $f$  agrees with  $(0^n 1, 1)$ , then by Theorem 12 it agrees with at least  $|I|$  examples. So assume w.l.o.g.  $f$  agrees with  $(0^n 1, 1)$ . This implies that  $f = g(x_1, \dots, x_n) \oplus x_{n+1}$  for some XOR function  $g$ . Then for each example  $(a\bar{b}, 1)$  that agrees with  $f$ , we have  $g(a) \oplus \bar{b} = 1$ , which implies  $g(a) = b$ . For each example  $(a\bar{b}, 1)$  that does not agree with  $f$ , we have  $g(a) \neq b$ . So  $f$  agrees with  $k + \frac{1}{2}|I|$  examples in  $\mathcal{P}$  if and only if  $g$  agrees with  $k$  examples from  $I$ . If XOR-MPA has a polynomial-time  $\beta$ -approximation algorithm where  $\beta(1 - \frac{1}{(\log n)^c} + \frac{1}{2}) > \frac{1}{2} + \frac{1}{2^{(\log n)^c}} + \frac{1}{2}$ , then by Theorem 12  $\text{NP} \subseteq \text{RTIME}(n^{O(\log \log n)})$ . The result follows.  $\square$

In the reduction above, we set  $\mathcal{N} = \{(0^{n+1}, 0)\}$  to ensure that optimal  $f$  is not the constant 1 function (recall that we define the XOR class to contain the constant 1). If we remove the constant 1 from our definition of XOR, the proof above works with  $\mathcal{N} = \emptyset$ . The resulting instance (with positive examples only) is an instance of the MAX WEIGHT problem discussed by Itoh (2000), for which we seek a codeword whose Hamming weight (number of 1s) is as large as possible. Itoh showed that MAX WEIGHT cannot be approximated within  $\frac{9}{10}$ , unless  $\text{P} = \text{NP}$ . Our result improves this to  $\frac{2}{3} + \frac{1}{2^{(\log n)^c}}$  under the assumption that  $\text{NP} \not\subseteq \text{RTIME}(n^{O(\log \log n)})$ .

## Acknowledgments

We would like to thank the anonymous referees for their helpful comments.

## References

- Amaldi, E., & Kann, V. (1995). The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147:1/2, 181–210.
- Angluin, D., & Laird, P. D. (1987). Learning from noisy examples. *Machine Learning*, 2:4, 343–370.
- Bartlett, P. L., & Ben-David, S. (1999). Hardness results for neural network approximation problems. In *Proceedings of the 4th European Conference on Computational Learning Theory* (pp. 50–62).
- Ben-David, S., Eiron, N., & Long, P. M. (2003). On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66:3, 496–514.
- Blum, A., Furst, M., Jackson, J., Kearns, M., Mansour, Y., & Rudich, S. (1994). Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing* (pp. 253–262).
- Blum, A. L., & Rivest, R. L. (1988). Training a 3-node neural network is NP-complete. In *Proceedings of the 1988 Workshop on Computational Learning Theory* (pp. 9–18).
- Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1987). Occam's razor. *Information Processing Letters*, 24:6, 377–380.



- Bshouty, N. H., & Burroughs, L. (2002a). Bounds for the minimum disagreement problem with applications to learning theory. In *Proceedings of the 15th Annual Conference on Computational Learning Theory* (pp. 271–286).
- Bshouty, N. H., & Burroughs, L. (2002b). Maximizing agreements and coagnostic learning. In *Proceedings of the 13th International Conference on Algorithmic Learning Theory*.
- Håstad, J. (1996). Clique is hard to approximate within  $n^{1-\epsilon}$ . In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science* (pp. 627–636).
- Håstad, J. (1997). Some optimal inapproximability results. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* (pp. 1–10).
- Höfgen, K.-U., Simon, H.-U., & Van Horn, K. S. (1995). Robust trainability of single neurons, *JCSS*, 50:1, 114–125.
- Itoh, T. (2000). Approximating the maximum weight of linear codes is APX-complete. *On Fundamentals of Electronics, Communications and Computer Sciences*, E83-A:4, 606–613.
- Kann, V., Khanna, S., Lagergren, J., & Panconesi, A. (1996). On the hardness of approximating max-k-cut and its dual. In *Proceedings of the Fourth Israeli Symposium on Theory of Computing and Systems* (pp. 61–67).
- Kearns, M., & Li, M. (1993). Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22:4, 807–837.
- Kuhlmann, C. (2000). Hardness results for general two-layer neural networks. In *Proceedings of the 13th Annual Conference on Computational Learning Theory* (pp. 275–285).
- Panconesi, A., & Ranjan, D. (1993). Quantifiers and approximation. *Theoretical Computer Science*, 107:1, 145–163.
- Papadimitriou, C., & Yannakakis, M. (1991). Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43, 425–440.
- Pitt, L., & Valiant, L. G. (1988). Computational limitations on learning from examples. *JACM*, 35:4, 965–984.
- Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27:11, 1134–1142.

Received November 8, 2002

Revised May 4, 2004

Accepted October 4, 2004