On Polynomial Time Constructions of Minimum Height Decision Tree

Nader H. Bshouty
Department of Computer Science, Technion, Haifa, Israel
bshouty@cs.technion.ac.il

Waseem Makhoul
Department of Computer Science, Technion, Haifa, Israel
waseemmakhoul@gmail.com

Abstract

A decision tree T in $B_m := \{0,1\}^m$ is a binary tree where each of its internal nodes is labeled with an integer in $[m] = \{1, 2, \ldots, m\}$, each leaf is labeled with an assignment $a \in B_m$ and each internal node has two outgoing edges that are labeled with 0 and 1, respectively. Let $A \subset \{0,1\}^m$. We say that T is a decision tree for A if (1) For every $a \in A$ there is one leaf of T that is labeled with a. (2) For every path from the root to a leaf with internal nodes labeled with $i_1, i_2, \ldots, i_k \in [m]$, a leaf labeled with $a \in A$ and edges labeled with $\xi_{i_1}, \ldots, \xi_{i_k} \in \{0,1\}$, a is the only element in A that satisfies $a_{i_j} = \xi_{i_j}$ for all $j = 1, \ldots, k$.

Our goal is to write a polynomial time (in $n := |A|$ and m) algorithm that for an input $A \subseteq B_m$ outputs a decision tree for A of minimum depth. This problem has many applications that include, to name a few, computer vision, group testing, exact learning from membership queries and game theory.

Arkin et al. and Moshkov [4, 15] gave a polynomial time $(\ln|A|)$- approximation algorithm (for the depth). The result of Dinur and Steurer [7] for set cover implies that this problem cannot be approximated with ratio $(1 - o(1)) \cdot \ln|A|$, unless P=NP. Moshkov studied in [15, 13, 14] the combinatorial measure of extended teaching dimension of A, $\mathrm{ETD}(A)$. He showed that $\mathrm{ETD}(A)$ is a lower bound for the depth of the decision tree for A and then gave an *exponential time* $\mathrm{ETD}(A)/\log(\mathrm{ETD}(A))$-approximation algorithm and a polynomial time $2(\ln 2)\mathrm{ETD}(A)$-approximation algorithm.

In this paper we further study the $\mathrm{ETD}(A)$ measure and a new combinatorial measure, $\mathrm{DEN}(A)$, that we call the density of the set A. We show that $\mathrm{DEN}(A) \leq \mathrm{ETD}(A) + 1$. We then give two results. The first result is that the lower bound $\mathrm{ETD}(A)$ of Moshkov for the depth of the decision tree for A is greater than the bounds that are obtained by the classical technique used in the literature. The second result is a polynomial time $(\ln 2)\mathrm{DEN}(A)$-approximation (and therefore $(\ln 2)\mathrm{ETD}(A)$-approximation) algorithm for the depth of the decision tree of A.

We then apply the above results to learning the class of disjunctions of predicates from membership queries [5]. We show that the ETD of this class is bounded from above by the degree d of its Hasse diagram. We then show that Moshkov algorithm can be run in polynomial time and is $(d/\log d)$-approximation algorithm. This gives optimal algorithms when the degree is constant. For example, learning axis parallel rays over constant dimension space.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization

Keywords and phrases Decision Tree, Minimal Depth, Approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.34

Related Version A full version of the paper is available at [6], https://arxiv.org/abs/1802.00233.

29th International Symposium on Algorithms and Computation (ISAAC 2018).
Editors: Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao; Article No. 34; pp. 34:1–34:12

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Consider the following problem: Given an n-element set $A \subseteq B_m := \{0,1\}^m$ from some class of sets $\mathcal{A}$ and a hidden element $a \in A$. Given an oracle that answers queries of the type: "What is the value of a_i?". Find a polynomial time algorithm that with an input A, asks minimum number of queries to the oracle and finds the hidden element a. This is equivalent to constructing a minimum height decision tree for A. A decision tree is a binary tree where each internal node is labeled with an index from $[m]$ and each leaf is labeled with an assignment $a \in B_m$. Each internal node has two outgoing edges one that is labeled with 0 and the other is labeled with 1. A node that is labeled with i corresponds to the query "Is $a_i = 0$?". An edge that is labeled with ξ corresponds to the answer ξ. This decision tree is an algorithm in an obvious way and its height is the worst case complexity of the number of queries. A decision tree T is said to be a *decision tree for* A if the algorithm that corresponds to T predicts correctly the hidden assignment $a \in A$. Our goal is to construct a small height decision tree for $A \subseteq B_m$ in time polynomial in m and $n := |A|$. We will denote by $\mathrm{OPT}(A)$ the minimum height decision tree for A.

This problem is related to the following problem in exact learning [1]: Given a class C of boolean functions $f : X \to \{0,1\}$. Construct in $poly(|C|, |X|)$ time an optimal adaptive algorithm that learns C from membership queries. This learning problem is equivalent to constructing a minimum height decision tree for the set $A = \{a^{(i)} | a^{(i)}_j = f_i(x_j)\}$ where f_i is the ith function in C and x_j is the jth instance in X. In computer vision the problem is related to minimizing the number of "probes" (queries) needed to determine which one of a finite set of geometric figures is present in an image [4]. In game theory the problem is related to the minimum number of turns required in order to win a guessing game.

1.1 Previous and New Results

In [4], Arkin et al. showed that (AMMRS-algorithm) if at every node the decision tree chooses i that partitions the current set (the set of assignments that are consistent to the answers of the queries so far) as evenly as possible, then the height of the tree is within a factor of $\log |A|$ from optimal. I.e., $\log |A|$-approximation algorithm. Moshkov [15] analysis shows that this algorithm is $(\ln |A|)$-approximation algorithm. This algorithm runs in polynomial time in m and $|A|$.

Hyafil and Rivest, [11], show that the problem of constructing a minimum depth decision tree is NP-Hard. They actually consider the average depth but their technique can be adopted to the minimum depth. The reduction of Laber and Nogueira, [12] to set cover with the inapproximability result of Dinur and Steurer [7] for set cover implies that it cannot be approximated to a factor of $(1-o(1)) \cdot \ln |A|$ unless P=NP. Therefore, no better approximation ratio can be obtained if no constraint is added to the set A.

Moshkov, [13], studied the extended teaching dimension combinatorial measure, $\mathrm{ETD}(A)$, of a set $A \subseteq B_m$. It is the maximum over all the possible assignments $b \in B_m$ of the minimum number of indices $I \subset [m]$ in which b agrees with at most one $a \in A$. Moshkov showed two results. The first is that $\mathrm{ETD}(A)$ is a lower bound for $\mathrm{OPT}(A)$. The second is an exponential time algorithm that asks $(2\mathrm{ETD}(A)/\log \mathrm{ETD}(A)) \log n$ queries. This gives a $(\ln 2)(\ln |A|)/\log \mathrm{ETD}(A)$ -approximation (exponential time) algorithm (since $\mathrm{OPT}(A) \geq \mathrm{ETD}(A)$) and at the same time $2\mathrm{ETD}(A)/\log \mathrm{ETD}(A)$-approximation algorithm (since $\mathrm{OPT}(A) \geq \log |A|$). Since many interesting classes have small ETD dimension, the latter result gives small approximation ratio but unfortunately Moshkov algorithm runs in exponential time. In [14], Moshkov gave a polynomial time $2(\ln 2)\mathrm{ETD}(C)$-approximation algorithm.

In this paper we further study the ETD measure. We show that the above AMMRS-algorithm, [4], is polynomial time $(\ln 2)\mathrm{ETD}(C)$-approximation algorithm. This improves the $2(\ln 2)\mathrm{ETD}(C)$-approximation algorithm of Moshkov.

Another reason for studying the ETD of classes is the following: If you find the ETD of the set A then you either get a lower bound that is better than the information theoretic lower bound $\log|A|$ or you get an approximation algorithm with a better ratio than $\ln|A|$. This is because if $\mathrm{ETD}(A) < \log|A|$ then the AMMRS-algorithm has a ratio $(\ln 2)\mathrm{ETD}(A)$ that is better than the $\ln|A|$ ratio and if $\mathrm{ETD}(A) > \log|A|$ then Moshkov lower bound, $\mathrm{ETD}(A)$, for $\mathrm{OPT}(A)$ is better than the information theoretic lower bound $\log|A|$.

To get the above results, we define a new combinatorial measure called the *density* $\mathrm{DEN}(A)$ of the set A. If $Q = \mathrm{DEN}(A)$ then there is a subset $B \subseteq A$ such that an adversary can give answers to the queries that eliminate at most $1/Q$ fraction of the number of elements in B. This forces the learner to ask at least Q queries. We then show that $\mathrm{ETD}(A) \geq \mathrm{DEN}(A) - 1$. On the other hand, we show that if $Q = \mathrm{DEN}(A)$ then a query in the AMMRS-algorithm eliminates at least $(1 - 1/Q)$ fraction of the assignments in A. This gives a polynomial time $(\ln 2)\mathrm{DEN}(A)$-approximation algorithm which is also a $(\ln 2)(\mathrm{ETD}(A) + 1)$-approximation algorithm.

In order to compare both algorithms we show that $(\mathrm{ETD}(A) - 1)/\ln|A| \leq \mathrm{DEN}(A) \leq \mathrm{ETD}(A) + 1$ and for random uniform A (and therefore for almost all A), with high probability $\mathrm{DEN}(A) = \Theta(\mathrm{ETD}(A)/\ln|A|)$. Since $|A| > \mathrm{ETD}(A)$, this shows that AMMRS-algorithm may get a better approximation ratio than Moshkov algorithm.

The inapproximability results follows from the reduction of Laber and Nogueira, [12] to set cover with the inapproximability result of Dinur and Steurer [7] and the fact that $\mathrm{DEN}(A) \leq \mathrm{ETD}(A) + 1 \leq \mathrm{OPT}(A) + 1$.

We then apply the above results to learning the class of disjunctions of predicates from a set of predicates $\mathcal{F}$ from membership queries [5]. We show that the ETD of this class is bounded from above by the degree d of its Hasse diagram. We then show that Moshkov algorithm, for this class, runs in *polynomial time* and is $(d/\log d)$-approximation algorithm. Since $|\mathcal{F}| \geq d$ (and in many applications, $|\mathcal{F}| \gg d$), this improves the $|\mathcal{F}|$-approximation algorithm SPEX in [5] when the size of Hasse diagram is polynomial. This also gives optimal algorithms when the degree d is constant. For example, learning axis parallel rays over constant dimension space.

2 Definitions and Preliminary Results

In this section we give some definitions and preliminary results

2.1 Notation

Let $B_m = \{0,1\}^m$. Let $A = \{a^{(1)}, \ldots, a^{(n)}\} \subseteq B_m$ be an n-element set. We will write $|A|$ for the number of elements in A. For $h \in B_m$ we define $A + h = \{a + h | a \in A\}$ where $+$ (in the square brackets) is the bitwise exclusive or of elements in B_m.

For integer q let $[q] = \{1, 2, \ldots, q\}$. Throughout the paper, $\log x = \log_2 x$.

2.2 Optimal Algorithm

We denote by $\mathrm{OPT}(A)$ the minimum depth of a decision tree for A. Our goal is to build a decision tree for A with small depth. Obviously

$$\log n \leq \mathrm{OPT}(A) \leq n - 1 \tag{1}$$

where $n := |A|$. The following result is easy to prove (see the full paper [6])

▶ **Lemma 1.** *We have* $\mathrm{OPT}(A) = \mathrm{OPT}(A + h)$.

2.3 Extended Teaching Dimension

In this section we define the extended teaching dimension.

Let $h \in B_m$ be any element. We say that a set $S \subseteq [m]$ is a *specifying set for h with respect to A* if $|\{a \in A \mid (\forall i \in S) h_i = a_i\}| \leq 1$. That is, there is at most one element in A that is *consistent with h* on the entries of S. Denote by $\mathrm{ETD}(A, h)$ the minimum size of a specifying set for h with respect to A. The *extended teaching dimension* of A is

$$\mathrm{ETD}(A) = \max_{h \in B_m} \mathrm{ETD}(A, h). \tag{2}$$

We will write $\mathrm{ETD}z(A)$ for $\mathrm{ETD}(A, 0)$. It is easy to see that

$$\mathrm{ETD}(A, h) = \mathrm{ETD}z(A + h) \text{ and } \mathrm{ETD}(A) = \mathrm{ETD}(A + h). \tag{3}$$

We say that a set $S \subseteq [m]$ is a *strong specifying set for h with respect to A* if either $h \in A$ and $|\{a \in A \mid (\forall i \in S) h_i = a_i\}| = 1$, or $|\{a \in A \mid (\forall i \in S) h_i = a_i\}| = 0$. That is, if $h \in A$ then there is exactly one element in A that is *consistent with h* on the entries of S. Otherwise, no element in A is consistent with h on S. Denote $\mathrm{SETD}(A, h)$ the minimum size of a strong specifying set for h with respect to A. The *strong extended teaching dimension* of A is

$$\mathrm{SETD}(A) = \max_{h \in B_m} \mathrm{SETD}(A, h). \tag{4}$$

We will write $\mathrm{SETD}z(A)$ for $\mathrm{SETD}(A, 0)$. It is easy to see that

$$\mathrm{SETD}(A, h) = \mathrm{SETD}z(A + h) \text{ and } \mathrm{SETD}(A) = \mathrm{SETD}(A + h). \tag{5}$$

Obviously, $\mathrm{ETD}(A, h) \leq \min(m, n - 1)$ and $\mathrm{ETD}(A, h) \leq \mathrm{SETD}(A, h) \leq \min(m, n)$

We now show

▶ **Lemma 2.** *We have* $\mathrm{ETD}(A, h) \leq \mathrm{SETD}(A, h) \leq \mathrm{ETD}(A, h) + 1$ *and therefore* $\mathrm{ETD}(A) \leq \mathrm{SETD}(A) \leq \mathrm{ETD}(A) + 1$.

Proof. The fact $\mathrm{ETD}(A, h) \leq \mathrm{SETD}(A, h)$ follows from the definitions. Let $S \subseteq [m]$ be a specifying set for h with respect to A. Then for $T := \{a \in A \mid (\forall i \in S) h_i = a_i\}$ we have $t := |T| \leq 1$. If $t = 0$ or $h \in A$ then S is a strong specifying set for h with respect to A. If $t = 1$ and $h \notin A$ then for the element $a \in T$ there is $j \in [m]$ such that $a_j \neq h_j$ and then $S \cup \{j\}$ is a strong specifying set for h with respect to A. This proves that $\mathrm{SETD}(A, h) \leq \mathrm{ETD}(A, h) + 1$.

The other claims follows immediately. ◀

Obviously, for any $B \subseteq A$

$$\mathrm{ETD}(B) \leq \mathrm{ETD}(A), \quad \mathrm{SETD}(B) \leq \mathrm{SETD}(A). \tag{6}$$

2.4 Hitting Set

A *hitting set for A* is a set $S \subseteq [m]$ such that for every non-zero element $a \in A$ there is $j \in S$ such that $a_j = 1$. That is, S *hits* every element in A except the zero element (if it exists). The size of the minimum size hitting set for A is denoted by $\mathrm{HS}(A)$.

We now show

▶ **Lemma 3.** *We have* $\mathrm{HS}(A) = \mathrm{SETD}z(A)$. *In particular,* $\mathrm{SETD}(A, h) = \mathrm{HS}(A + h)$ *and* $\mathrm{SETD}(A) = \max_{h \in B_m} \mathrm{HS}(A + h)$.

Proof. If $0 \in A$ then SETD$z(A)$ is the minimum size of a set S such that $\{a \in A \mid (\forall i \in S)a_i = 0\} = \{0\}$ and if $0 \notin A$ then it is the minimum size of a set S such that $\{a \in A \mid (\forall i \in S)a_i = 0\} = \emptyset$. Therefore the set S hits all the nonzero elements in A.

The other results follow from (5) and the definition of SETD. ◀

2.5 Density of a Set

In this section we define our new measure DEN of a set.

Let $A = \{a^{(1)}, \ldots, a^{(n)}\} \subseteq B_m$. We define $\text{MAJ}(A) \in B_m$ such that $\text{MAJ}(A)_i = 1$ if the number of ones in $(a_i^{(1)}, \cdots, a_i^{(n)})$ is greater or equal the number of zeros and $\text{MAJ}(A)_i = 0$ otherwise. We denote by $\text{MAX}(A)$ the maximum number of ones in $(a_i^{(1)}, \cdots, a_i^{(n)})$ over all $i = 1, \ldots, m$. Let

$$\text{MAMI}(A) = \min_{h \in B_m} \text{MAX}(A + h) = \text{MAX}(A + \text{MAJ}(A)). \tag{7}$$

For $j \in [m]$ and $\xi \in \{0, 1\}$ let $A_{j,\xi} = \{a \in A \mid a_j = \xi\}$. Then

$$\text{MAMI}(A) = \max_j \min(|A_{j,0}|, |A_{j,1}|). \tag{8}$$

We define the *density* of a set $A \subseteq B_m$ by

$$\text{DEN}(A) = \max_{B \subseteq A} \frac{|B| - 1}{\text{MAMI}(B)}. \tag{9}$$

Notice that since every $j \in [m]$ can hit at most $\text{MAX}(A)$ elements in A we have

$$\text{HS}(A) \geq \frac{|A| - 1}{\text{MAX}(A)}. \tag{10}$$

3 Bounds for OPT

In this section we give upper and lower bounds for OPT.

3.1 Lower Bound

Moshkov results in [13, 10] and the information theoretic bound in (1) give the following lower bound. We give the proof in the full paper [6] for completeness.

▶ **Lemma 4.** *[13, 10] Let $A \subseteq B_m$ be any set. Then* $\text{OPT}(A) \geq \max(\text{ETD}(A), \log |A|)$.

Many lower bounds in the literature for $\text{OPT}(A)$ are based on finding a subset $B \subseteq A$ such that for each query there is an answer that eliminates at most small fraction E of B. Then $(|B| - 1)/E$ is a lower bound for $\text{OPT}(A)$. The best possible bound that one can get using this technique is exactly $\text{DEN}(A)$ (Lemma 5), the density defined in Section 2.5. Lemma 6 shows that the lower bound $\text{ETD}(A)$ for $\text{OPT}(A)$ exceeds any such bound.

In the full paper [6] we prove

▶ **Lemma 5.** *We have* $\text{OPT}(A) \geq \text{DEN}(A)$.

▶ **Lemma 6.** *We have* $\text{ETD}(A) \geq \text{DEN}(A) - 1$.

Proof. By (7) and (9) there is $B \subseteq A$ such that

$$\mathrm{DEN}(A) = \frac{|B|-1}{\mathrm{MAMI}(B)} = \frac{|B|-1}{\mathrm{MAX}(B+h)} \tag{11}$$

where $h = \mathrm{MAJ}(B)$. Then

$$\begin{aligned}\mathrm{ETD}(A) &\overset{(6)}{\geq} \mathrm{ETD}(B) \overset{(2)}{\geq} \mathrm{ETD}(B,h) \overset{L2}{\geq} \mathrm{SETD}(B,h) - 1 \overset{L3}{=} \mathrm{HS}(B+h) - 1 \\ &\overset{(10)}{\geq} \frac{|B|-1}{\mathrm{MAX}(B+h)} - 1 \overset{(11)}{=} \mathrm{DEN}(A) - 1.\end{aligned}$$

◀

In the full paper [6] we also prove

▶ **Lemma 7.** *We have* $\mathrm{ETD}(A) \leq \ln|A| \cdot \mathrm{DEN}(A) + 1$.

It is also easy to see (by standard analysis using Chernoff Bound) that for a random uniform A, with positive probability, $\mathrm{DEN}(A) = O(1)$ and $\mathrm{ETD}(A) = \Theta(\log|A|)$. See the proof sketch in the full paper [6]. So the bound in Lemma 7 is asymptotically best possible.

3.2 Upper Bounds

Moshkov [13, 10] proved the following upper bound. We gave the proof in the full paper [6] for completeness.

▶ **Lemma 8.** *[13, 10] Let* $A \subseteq \{0,1\}^m$ *of size* n. *Then*

$$\mathrm{OPT}(A) \leq \mathrm{ETD}(A) + \frac{\mathrm{ETD}(A)}{\log \mathrm{ETD}(A)} \log n \leq \frac{2 \cdot \mathrm{ETD}(A)}{\log \mathrm{ETD}(A)} \log n.$$

In [13, 10], Moshkov gave an example of a n-set $A_E \subseteq \{0,1\}^m$ with $\mathrm{ETD}(A_E) = E$ and $\mathrm{OPT}(A_E) = \Omega((E/\log E)\log n)$. So the upper bound in the above lemma is the best possible.

4 Polynomial Time Approximation Algorithm

Given a a set $A \subseteq B_m$. Can one construct an algorithm that finds a hidden $a \in A$ with $\mathrm{OPT}(A)$ queries? Obviously, with unlimited computational power this can be done so the question is: How close to $\mathrm{OPT}(A)$ can one get when polynomial time $poly(m,n)$ is allowed for the construction?

An exponential time algorithm follows from the following

$$\mathrm{OPT}(A) = \min_{i \in [m]} \max(\mathrm{OPT}(A_{i,0}), \mathrm{OPT}(A_{i,1}))$$

where $A_{i,\xi} = \{a \in A \mid a_i = \xi\}$. This algorithm runs in time at least $m! \geq (m/e)^m$. See also [8, 3].

Can one give a better exponential time algorithm? In what follows (Theorem 9) we use Moshkov [13, 10] result (Lemma 8) to give a better exponential time approximation algorithm. In in the full paper [6] we give another simple proof of the Moshkov [13, 10] result that in practice uses less number of specifying sets. When the extended teaching dimension is constant, the algorithm is $O(1)$-approximation algorithm and runs in polynomial time.

▶ **Theorem 9.** *Let $\mathcal{A}$ be a class of sets $A \subseteq B_m$ of size n. If there is an algorithm that for any $h \in B_m$ and any $A \in \mathcal{A}$ gives a specifying set for h with respect to A of size at most E in time T then there is an algorithm that for any $A \in \mathcal{A}$ constructs a decision tree for A of depth at most*

$$E + \frac{E}{\log E}\log n \le E + \frac{E}{\log E}\mathrm{OPT}(A)$$

queries and runs in time $O(T \log n + nm)$.

Proof. Follows immediately from Moshkov algorithm [13, 10]. See the full paper [6]. ◀

The following result immediately follows from Theorem 9.

▶ **Theorem 10.** *Let $A \subseteq B_m$ be a n-set. There is an algorithm that finds the hidden column in time*

$$\binom{m}{\mathrm{ETD}(A)} \cdot \mathrm{ETD}(A) \cdot n \log n$$

and asks at most

$$\frac{2 \cdot \mathrm{ETD}(A) \cdot \log n}{\log \mathrm{ETD}(A)} \le \frac{2 \cdot \min(\mathrm{ETD}(A), \log n)}{\log \mathrm{ETD}(A)}\mathrm{OPT}(A)$$

queries.

In particular, if $\mathrm{ETD}(A)$ is constant then the algorithm is $O(1)$-approximation algorithm that runs in polynomial time.

Proof. To find a specifying set for h with respect to A we exhaustively check each $\mathrm{ETD}(A)$ row of A. Each check takes time n. Since the algorithm asks at most $\mathrm{ETD}(A) \cdot \log n$ queries, the time complexity is as stated in the Theorem. ◀

Can one do it in $poly(m, n)$ time? Hyafil and Rivest, [11], show that the problem of finding OPT is NP-Complete. The reduction of Laber and Nogueira, [12], of set cover to this problem with the inapproximability result of Dinur and Steurer [7] for set cover implies that it cannot be approximated to $(1 - o(1)) \cdot \ln n$ unless P=NP.

In [4], Arkin et al. showed that (the AMMRS-algorithm) if at the ith query the algorithm chooses an index j that partitions the current node set (the elements in A that are consistent with the answers until this node) A as evenly as possible, that is, that maximizes $\min(|\{a \in A | a_j = 0\}|, |\{a \in A | a_j = 1\}|)$, then the query complexity is within a factor of $\lceil \log n \rceil$ from optimal. The AMMRS-algorithm, [4], runs in time $poly(m, n)$. Moshkov [4, 15] analysis shows that this algorithm is $\ln n$-approximation algorithm and therefore is optimal. In this section we will give a simple proof.

In [13, 10], Moshkov gave a simple $\mathrm{ETD}(A)$-approximation algorithm (Algorithm MEMB-HALVING-1 in [10]). He then gave another algorithm that achieves the query complexity in Lemma 8 (Algorithm MEMB-HALVING-2 in [10]). This is within a factor of

$$\frac{2 \cdot \min(\mathrm{ETD}(A), \log n)}{\log \mathrm{ETD}(A)}$$

from optimal. This is better than the ratio $\ln n$, but, unfortunately, both algorithms require finding a minimum size specifying set and the problem of finding a minimum size specifying set for h is NP-Hard, [16, 2, 9]. Moshkov gave in [14] a polynomial time $2(\ln 2)$-approximation algorithm.

Can one achieve a better approximation ratio? In the following we give a surprising result. We show that the AMMRS-algorithm asks $\mathrm{DEN}(A)\ln|A|$ queries. Therefore, it is a $(\ln 2)\mathrm{DEN}(A)$-approximation algorithm and therefore it is a $(\ln 2)\mathrm{ETD}(A)$-approximation algorithm. This also prove that it is a $\ln|A|$-approximation algorithm. We also show that no algorithm with query complexity $(1-\epsilon)\mathrm{DEN}(A)\ln|A|$ is possible unless P=NP.

▶ **Theorem 11.** *The AMMRS-algorithm runs in time $O(mn)$ and finds the hidden element $a \in A$ with at most*

$$\begin{aligned}\mathrm{DEN}(A)\cdot\ln(n) &\leq \min((\ln 2)\mathrm{DEN}(A), \ln n)\cdot \mathrm{OPT}(A)\\ &\leq \min((\ln 2)(\mathrm{ETD}(A)+1), \ln n)\cdot \mathrm{OPT}(A)\end{aligned}$$

queries.

Proof. Let B be any subset of A. Then,

$$\mathrm{DEN}(B) \overset{(9)}{\geq} \frac{|B|-1}{\mathrm{MAMI}(B)}$$

and therefore

$$\mathrm{MAMI}(B) \geq \frac{|B|-1}{\mathrm{DEN}(B)} \geq \frac{|B|-1}{\mathrm{DEN}(A)}.$$

Since the AMMRS-algorithm chooses at each node in the decision tree the index j that maximizes $\min(|B_{j,0}|, |B_{j,1}|)$ where $B_{j,\xi} = \{a \in B | a_j = \xi\}$ and B is the set of elements in A that are consistent with the answers until this node, we have

$$\begin{aligned}\max(|B_{j,0}|, |B_{j,1}|) - 1 &= |B| - 1 - \min(|B_{j,0}|, |B_{j,1}|)\\ &\overset{(8)}{=} |B| - 1 - \mathrm{MAMI}(B) \leq (|B|-1)\left(1 - \frac{1}{\mathrm{DEN}(A)}\right).\end{aligned}$$

Therefore, for a node v of depth h in the decision tree, the set $B(v)$ of elements in A that are consistent with the answers until this node contains at most

$$(|A|-1)\left(1-\frac{1}{\mathrm{DEN}(A)}\right)^h + 1$$

elements. Therefore the depth of the tree is at most $\mathrm{DEN}(A)\ln|A|$. ◀

We now show that the query complexity of this algorithm is optimal unless P=NP.

▶ **Theorem 12.** *Let ϵ be any constant. There is no polynomial time algorithm that finds the hidden element with less than $(1-\epsilon)\mathrm{DEN}(A)\cdot\ln|A|$ unless P=NP.*

Proof. Suppose such an algorithm exists. Then $(1-\epsilon)\mathrm{DEN}(A)\ln|A| \overset{L5}{\leq} (1-\epsilon)\ln|A|\mathrm{OPT}(A)$. That is, the algorithm is also $(1-\epsilon)\ln|A|$-approximation algorithm. Laber and Nogueira, [12] gave a polynomial time algorithm reduction of minimum depth decision tree to set cover and Dinur and Steurer [7] show that there is no polynomial time $(1-o(1))\cdot\ln|A|$ for set cover unless P=NP. Therefore, such an algorithm implies P=NP. ◀

5 Applications to Disjunction of Predicates

In this section we apply the above results to learning the class of disjunctions of predicates from a set of predicates $\mathcal{F}$ from membership queries [5].

Let $C = \{f_1, \ldots, f_n\}$ be a set of boolean functions $f_i : X \to \{0,1\}$ where $X = \{x_1, \ldots, x_m\}$. Let $A_C = \{(f_i(x_1), \ldots, f_i(x_m)) \mid i = 1, \ldots, n\}$. We will write $\mathrm{OPT}(A_C)$, $\mathrm{ETD}(A_C)$, etc. as $\mathrm{OPT}(C), \mathrm{ETD}(C)$, etc.

Let $\mathcal{F}$ be a set of boolean functions (predicates) over a domain X. We consider the class of functions $\mathcal{F}_\vee := \{\vee_{f\in S} f \mid S \subseteq \mathcal{F}\}$.

5.1 An Equivalence Relation Over $\mathcal{F}_\vee$

In this section, we present an equivalence relation over $\mathcal{F}_\vee$ and define the representatives of the equivalence classes. This enables us in later sections to focus on the representative elements from $\mathcal{F}_\vee$. Let $\mathcal{F}$ be a set of boolean functions over the domain X. The equivalence relation $=$ over $\mathcal{F}_\vee$ is defined as follows: two disjunctions $F_1, F_2 \in \mathcal{F}_\vee$ are equivalent ($F_1 = F_2$) if F_1 is logically equal to F_2. In other words, they represent the same function (from X to $\{0,1\}$). We write $F_1 \equiv F_2$ to denote that F_1 and F_2 are identical; that is, they have the same representation. For example, consider $f_1, f_2 : \{0,1\} \to \{0,1\}$ where $f_1(x) = 1$ and $f_2(x) = x$. Then, $f_1 \vee f_2 = f_1$ but $f_1 \vee f_2 \not\equiv f_1$.

We denote by $\mathcal{F}_\vee^*$ the set of equivalence classes of $=$ and write each equivalence class as $[F]$, where $F \in \mathcal{F}_\vee$. Notice that if $[F_1] = [F_2]$, then $[F_1 \vee F_2] = [F_1] = [F_2]$. Therefore, for every $[F]$, we can choose the *representative element* to be $G_F := \vee_{F'\in S} F'$ where $S \subseteq \mathcal{F}$ is the maximum size set that satisfies $\vee S := \vee_{f\in S} f = F$. We denote by $G(\mathcal{F}_\vee)$ the set of all representative elements. Accordingly, $G(\mathcal{F}_\vee) = \{G_F \mid F \in \mathcal{F}_\vee\}$. As an example, consider the set $\mathcal{F}$ consisting of four functions $f_{11}, f_{12}, f_{21}, f_{22} : \{1,2\}^2 \to \{0,1\}$ where $f_{ij}(x_1, x_2) = [x_i \geq j]$ where $[x_i \geq j] = 1$ if $x_i \geq j$ and 0 otherwise. There are $2^4 = 16$ elements in $\mathrm{Ray}_2^2 := \mathcal{F}_\vee$ and five representative functions in $G(\mathcal{F}_\vee)$: $G(\mathcal{F}_\vee) = \{f_{11} \vee f_{12} \vee f_{21} \vee f_{22},\ f_{12} \vee f_{22},\ f_{12},\ f_{22}, 0\}$ (where 0 is the zero function).

5.2 A Partial Order Over $\mathcal{F}_\vee$ and Hasse Diagram

In this section, we define a partial order over $\mathcal{F}_\vee$ and present related definitions. The partial order, denoted by $\Rightarrow$, is defined as follows: $F_1 \Rightarrow F_2$ if F_1 logically implies F_2. Consider the Hasse diagram $H(\mathcal{F}_\vee)$ of $G(\mathcal{F}_\vee)$ for this partial order. The maximum (top) element in the diagram is $G_{\max} := \vee_{f\in\mathcal{F}} f$. The minimum (bottom) element is $G_{\min} := \vee_{f\in\emptyset} f$, i.e., the zero function.

In a Hasse diagram, G_1 is a *descendant* (resp., *ascendent*) of G_2 if there is a (nonempty) downward path from G_2 to G_1 (resp., from G_1 to G_2), i.e., $G_1 \Rightarrow G_2$ (resp., $G_2 \Rightarrow G_1$) and $G_1 \neq G_2$. G_1 is an *immediate descendant* of G_2 in $H(\mathcal{F}_\vee)$ if $G_1 \Rightarrow G_2$, $G_1 \neq G_2$ and there is no $G \in G(\mathcal{F}_\vee)$ such that $G \neq G_1$, $G \neq G_2$ and $G_1 \Rightarrow G \Rightarrow G_2$. G_1 is an *immediate ascendant* of G_2 if G_2 is an immediate descendant of G_1.

We denote by $\mathrm{De}(G)$ and $\mathrm{As}(G)$ the sets of all the immediate descendants and immediate ascendants of G, respectively. The *neighbours set* of G is $\mathrm{Ne}(G) = \mathrm{De}(G) \cup \mathrm{As}(G)$. We further denote by $\mathrm{DE}(G)$ and $\mathrm{AS}(G)$ the sets of all G's descendants and ascendants, respectively.

▶ **Definition 13.** The *degree* of G is $\deg(G) = |\mathrm{Ne}(G)|$ and the degree $\deg(\mathcal{F}_\vee)$ of $\mathcal{F}_\vee$ is $\max_{G\in G(\mathcal{F}_\vee)} \deg(G)$.

For G_1 and G_2, we define their *lowest common ascendent* (resp., greatest common descendant) $G = \mathrm{lca}(G_1, G_2)$ (resp., $G = \mathrm{gcd}(G_1, G_2)$) to be the minimum (resp., maximum) element in $\mathrm{AS}(G_1) \cap \mathrm{AS}(G_2)$ (resp., $\mathrm{DE}(G_1) \cap \mathrm{DE}(G_2)$).

The following result is from [5]

▶ **Lemma 14.** *Let $G_1, G_2 \in G(\mathcal{F}_\vee)$. Then,* $\mathrm{lca}(G_1, G_2) = G_1 \vee G_2$.

In particular, if G_1, G_2 are two distinct immediate descendants of G, then $G_1 \vee G_2 = G$.

5.3 Witnesses

In this subsection we define the term *witness*. Let G_1 and G_2 be elements in $G(\mathcal{F}_\vee)$. An element $a \in X$ is a *witness* for G_1 and G_2 if $G_1(a) \neq G_2(a)$.

For a class of boolean functions C over a domain X and a function $G \in C$ we say that a set of elements $W \subseteq X$ is a *witness set* for G in C if for every $G' \in C$ and $G' \neq G$ there is a witness in W for G and G'.

5.4 The Extended Teaching Dimension of $\mathcal{F}_\vee$

In this section we prove

▶ **Lemma 15.** *For every $h : X \to \{0,1\}$ if $h \not\Rightarrow G_{\max}$ then* $\mathrm{ETD}(\mathcal{F}_\vee, h) = 1$. *Otherwise, there is $G \in G(\mathcal{F}_\vee)$ such that*

$$\mathrm{ETD}(\mathcal{F}_\vee, h) \leq |\mathrm{De}(G)| + \mathrm{HS}(\mathrm{As}(G) \wedge \bar{G}) \leq |\mathrm{Ne}(G)| = \deg(G)$$

where $As(G) \wedge \bar{G} = \{s \wedge \bar{G} \mid s \in \mathrm{As}(G)\}$. In particular,

$$\mathrm{ETD}(\mathcal{F}_\vee) \leq \max_{G \in G(\mathcal{F}_\vee)} \left(|\mathrm{De}(G)| + \mathrm{HS}(\mathrm{As}(G) \wedge \bar{G})\right) \leq \deg(\mathcal{F}_\vee).$$

Proof. Let $h : X \to \{0,1\}$ be any function. If $h \not\Rightarrow G_{\max}$ then there is an assignment a that satisfies $h(a) = 1$ and $G_{\max}(a) = 0$. Since for all $G \in G(\mathcal{F}_\vee)$, $G \Rightarrow G_{\max}$ we have $G(a) = 0$. Therefore, the set $\{a\}$ is a specifying set for h with respect to $\mathcal{F}_\vee$ and $\mathrm{ETD}(\mathcal{F}_\vee, h) = 1$.

Let $h \Rightarrow G_{\max}$. Consider any $G \in G(\mathcal{F}_\vee)$ such that $h \Rightarrow G$ and for every immediate descendant G' of G we have $h \not\Rightarrow G'$. Now for every immediate descendent G' of G find an assignment a such that $G'(a) = 0$ and $h(a) = 1$. Then a is a witness for h and G'. Therefore, a is also a witness for h and every descendant of G'. Let A be the set of all such assignments, i.e., for every descendant of G one witness. Then $|A| \leq |\mathrm{De}(G)|$ and A is a witness set for h and all the descendants of G. We note here that if $h = 0$ then $G = G_{\min}$ which has no immediate descendants and then $A = \emptyset$.

Consider a hitting set B for $As(G) \wedge \bar{G}$ of size $\mathrm{HS}(As(G) \wedge \bar{G})$. Now for every immediate ascendant G'' of G find an assignment $b \in B$ such that $G''(b) \wedge \bar{G}(b) = 1$. Then $G''(b) = 1$ and $G(b) = 0$. Since $G(b) = 0$ we have $h(b) = 0$ and then b is a witness for h and G''. Therefore, b is also a witness for h and every ascendant of G''. Thus B is a witness set for h in all the ascendants of G.

Let G_0 be any element in $G(\mathcal{F}_\vee)$ (that is not a descendant or an ascendant). Consider $G_1 = \mathrm{lca}(G, G_0)$. By Lemma 14, we have $G_1 = G \vee G_0$. Since G_1 is an ascendent of G there is a witness $a \in B$ such that $G_1(a) = 1$ and $G(a) = 0$. Then $G_0(a) = 1$, $h(a) = 0$ and a is a witness of h and G_0. Therefore $A \cup B$ is a specifying set for h with respect to $G(\mathcal{F}_\vee)$. Since for every $F \in \mathcal{F}_\vee$ we have $F = G_F \in G(\mathcal{F}_\vee)$, $A \cup B$ is also a specifying set for h with respect to $\mathcal{F}_\vee$.

Since

$$\mathrm{ETD}(\mathcal{F}_\vee, h) \leq |A| + |B| \leq |\mathrm{De}(G)| + \mathrm{HS}(\mathrm{As}(G) \wedge \bar{G})$$

the result follows. ◀

In in the full paper [6] we show that

$$\mathrm{ETD}(\mathcal{F}_\vee) = \max_{G \in G(\mathcal{F}_\vee)} \left(|\mathrm{De}(G)| + \mathrm{HS}(\mathrm{As}(G) \wedge \bar{G}) \right).$$

We could have replaced $|\mathrm{De}(G)|$ by $\mathrm{HS}(\overline{\mathrm{De}(G)} \wedge G)$, but in the full paper [6] we show that they are both equal.

The following result follows immediately from the proof of Lemma 15

▶ **Lemma 16.** *For any $h : X \to \{0,1\}$, a specifying set for h with respect to $\mathcal{F}_\vee$ of size $\deg(\mathcal{F}_\vee)$ can be found in time $O(nm)$.*

By Theorem 9 we have

▶ **Theorem 17.** *There is an algorithm that learns $\mathcal{F}_\vee$ in time $O(nm)$ and asks at most*

$$\deg(\mathcal{F}_\vee) + \frac{\deg(\mathcal{F}_\vee)}{\log \deg(\mathcal{F}_\vee)} \log n \leq \left(\frac{\deg(\mathcal{F}_\vee)}{\log \deg(\mathcal{F}_\vee)} + 1 \right) \mathrm{OPT}(\mathcal{F}_\vee)$$

membership queries.

5.5 Learning Other Classes

If a specifying set of small size cannot be found in polynomial time then from Theorem 10, 11 and Lemma 15, we have

▶ **Theorem 18.** *For a class C we have*

1. *There is an algorithm that learns C in time*

$$\binom{m}{\deg(C)} \cdot \mathrm{ETD}(C) \cdot n \log n$$

and asks at most

$$\frac{2 \cdot \mathrm{ETD}(C) \cdot \log n}{\log \mathrm{ETD}(C))} \leq \frac{2 \cdot \min(\mathrm{ETD}(C)), \log n)}{\log \mathrm{ETD}(C))} \mathrm{OPT}(C)$$

membership queries.
In particular, when $\mathrm{ETD}(C)$ is constant the algorithm runs in polynomial time and its query complexity is (asymptotically) optimal.

2. *There is an algorithm that learns C in time $O(nm)$ and asks at most*

$$\begin{aligned} \mathrm{DEN}(C) \cdot \ln(n) &\leq \min((\ln 2)\mathrm{DEN}(C), \ln n) \cdot \mathrm{OPT}(C) \\ &\leq \min((\ln 2)(\mathrm{ETD}(C) + 1), \ln n) \cdot \mathrm{OPT}(C) \end{aligned}$$

membership queries.

References

1. Dana Angluin. Queries and Concept Learning. *Machine Learning*, 2(4):319–342, 1988. doi:10.1023/A:1022821128753.
2. Martin Anthony, Graham R. Brightwell, David A. Cohen, and John Shawe-Taylor. On Exact Specification by Examples. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992.*, pages 311–318, 1992. doi:10.1145/130385.130420.
3. Esther M. Arkin, Michael T. Goodrich, Joseph S. B. Mitchell, David M. Mount, Christine D. Piatko, and Steven Skiena. Point Probe Decision Trees for Geometric Concept Classes. In *Algorithms and Data Structures, Third Workshop, WADS '93, Montréal, Canada, August 11-13, 1993, Proceedings*, pages 95–106, 1993. doi:10.1007/3-540-57155-8_239.
4. Esther M. Arkin, Henk Meijer, Joseph S. B. Mitchell, David Rappaport, and Steven Skiena. Decision trees for geometric models. *Int. J. Comput. Geometry Appl.*, 8(3):343–364, 1998. doi:10.1142/S0218195998000175.
5. Nader H. Bshouty, Dana Drachsler-Cohen, Martin T. Vechev, and Eran Yahav. Learning Disjunctions of Predicates. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 346–369, 2017. URL: http://proceedings.mlr.press/v65/bshouty17a.html.
6. Nader H. Bshouty and Waseem Makhoul. On Polynomial time Constructions of Minimum Height Decision Tree. *CoRR*, abs/1802.00233, 2018. arXiv:1802.00233.
7. Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633, 2014. doi:10.1145/2591796.2591884.
8. M. R. Garey. Optimal Binary Identification Procedures. *SIAM Journal on Applied Mathematics*, 23(2):173–186, 1971. URL: http://epubs.siam.org/doi/abs/10.1137/0123019.
9. Sally A. Goldman and Michael J. Kearns. On the Complexity of Teaching. *J. Comput. Syst. Sci.*, 50(1):20–31, 1995. doi:10.1006/jcss.1995.1003.
10. Tibor Hegedüs. Generalized Teaching Dimensions and the Query Complexity of Learning. In *Proceedings of the Eigth Annual Conference on Computational Learning Theory, COLT 1995, Santa Cruz, California, USA, July 5-8, 1995*, pages 108–117, 1995. doi:10.1145/225298.225311.
11. Laurent Hyafil and Ronald L. Rivest. Constructing Optimal Binary Decision Trees is NP-Complete. *Inf. Process. Lett.*, 5(1):15–17, 1976. doi:10.1016/0020-0190(76)90095-8.
12. Eduardo Sany Laber and Loana Tito Nogueira. On the hardness of the minimum height decision tree problem. *Discrete Applied Mathematics*, 144(1-2):209–212, 2004. doi:10.1016/j.dam.2004.06.002.
13. Mikhail Ju. Moshkov. On conditional tests. *Problemy Kibernetiki. and Sov. Phys. Dokl.*, 27(7):528–530, 1982.
14. Mikhail Ju. Moshkov. On conditional tests. *Problems of Cybernetics, Nauka, Moscow*, 40:131–170, 1982.
15. Mikhail Ju. Moshkov. Greedy Algorithm of Decision Tree Construction for Real Data Tables. *Transactions on Rough Sets I, Lecture Notes in Computer Science 3100, Springer-Verlag, Heidelberg.*, pages 161–168, 2004. doi:10.1007/978-3-540-27794-1_7.
16. Ayumi Shinohara. Teachability in Computational Learning. *New Generation Comput.*, 8(4):337–347, 1991. doi:10.1007/BF03037091.