Journal of Machine Learning Research 3 (2002) 49-57 Submitted 3/02; Published 7/02

Learning Monotone DNF from a Teacher that Almost Does Not Answer Membership Queries

Nader H. Bshouty BSHOUTY@CS.TECHNION.AC.IL
Department of Computer Science,
Technion – Israel Institute of Technology,
Haifa 32000, Israel.

Nadav Eiron NADAV@US.IBM.COM
IBM Almaden Research Center, 52BC/B2
650 Harry Road
San Jose, CA, 95120, USA.

Editor: Dana Ron

Abstract

We present results concerning the learning of Monotone DNF (MDNF) from Incomplete Membership Queries and Equivalence Queries. Our main result is a new algorithm that allows efficient learning of MDNF using Equivalence Queries and Incomplete Membership Queries with probability of $p = 1 - 1/\text{poly}(n, t)$ of failing. Our algorithm is expected to make

$$O\left(\left(\frac{tn}{1-p}\right)^2\right)$$

queries, when learning a MDNF formula with t terms over n variables. Note that this is polynomial for any failure probability $p = 1 - 1/\text{poly}(n, t)$. The algorithm's running time is also polynomial in t, n, and $1/(1-p)$. In a sense this is the best possible, as learning with $p = 1 - 1/\omega(\text{poly}(n, t))$ would imply learning MDNF, and thus also DNF, from equivalence queries alone.[1]

1. Introduction

Angluin's exact learning model (see Angluin 1988), using equivalence queries and membership queries, has attracted much attention in COLT literature. One of its biggest disadvantages, however, is its requirement of the membership oracle to be able to supply, on demand, the value of the target function at any point of its domain. To relax this requirement, the model of Incomplete Membership Queries was suggested by Angluin and Slonim (1994). In this model, some of the points cannot be queried: an MQ query for such a point returns the result "I don't know" instead of the value of the target function. Points for which MQs are not answered are chosen randomly and independently of each other. Initially, this model was known to allow learning of functions not learnable by EQs alone, only if many of the MQs were answered (see, for example, Angluin and Slonim 1994, Goldman and Mathias 1992, Chen 1994).

1. An early version of this paper appeared as Bshouty and Eiron (2001).

In this work we show that even when the MQ oracle has a polynomially small (in n, the number of variables, and t the number of terms in the target) probability of supplying an answer, Monotone DNF can still be efficiently learned. As Monotone DNF is not efficiently learnable from equivalence queries alone, this shows that even adding a very "weak" membership oracle to the equivalence oracle gives additional power to the learning algorithm. The idea behind our algorithm is that, at any stage, it is possible to find polynomially many points on which MQs may be asked, such that any one of them being answered would allow us to, again, generate polynomially many new points on which MQs may be asked. Our result improves upon the previous results that required a constant fraction of the queries to be answered for successful learning (see Angluin and Slonim 1994). Note that this is the best possible, since an efficient learning algorithm that uses an incomplete membership oracle with sub-polynomial probability of answering queries should expect to never receive any answers to MQs, as it will make only polynomially many queries. Thus, efficient exact learning with an Incomplete Membership Oracle with probability less than polynomial of answering queries is equivalent to learning from EQs alone, and it is well known that learning MDNF from EQs alone implies the learning of DNF from EQs alone (see Kearns et al. 1987 for a discussion of the hardness of learning DNF formulae). Bshouty and Owshanko (2001) have shown DFA (and therefore also the classes considered in Chen 1994) to be learnable in the same model we consider.

The rest of this paper is organized as follows: Section 2 formally defines the framework within which our results are presented, including the learning models used and the type of problems addressed. Section 3 briefly introduces our results. Section 4 includes a detailed presentation of our main positive result, showing that Monotone DNF can be efficiently learned using Equivalence Queries and Incomplete Membership Queries of polynomially small probability of success. We conclude with discussion of our results and some open problems in Section 5.

2. Preliminaries

2.1 Exact Learning

The exact learning model was introduced by Angluin (1988), with many variants of it defined over the years (for instance, by Littlestone 1989, Littlestone and Warmuth 1994, Auer 1993, Auer and Cesa-Bianchi 1998). We first define some of the basic terms used in exact learning. We concern ourselves with learning boolean concepts. A learning algorithm is trying to identify a *target function* $f : \mathcal{X} \mapsto \{0,1\}$, which is known to have been chosen from a *concept class* $\mathcal{C}$, of boolean functions over an instance space, or domain $\mathcal{X}$. The learning process is successful if the algorithm outputs a hypothesis function h that is equivalent to f. The learning algorithm obtains (partial) information about the target function f through the use of oracles. One type of oracle used is the Equivalence Oracle EQ. This oracle receives as input some hypothesis $h : \mathcal{X} \mapsto \{0,1\}$, and either answers 'YES', if $h \equiv f$, or else, answers with some counterexample $x \in \mathcal{X}$ for which $f(x) \neq h(x)$. Note that there is no requirement of the equivalence oracle to present specific counterexamples. Indeed, a concept class $\mathcal{C}$ is said to be exactly learnable from equivalence queries, if there exists an algorithm such that for all $f \in \mathcal{C}$ and *any* equivalence oracle, the algorithm, using the equivalence oracle, will output, in polynomial time, a hypothesis h that is equivalent to f.

Another often-used oracle in exact learning is the membership oracle MQ. The membership oracle is given as input a point $x \in \mathcal{X}$, and returns as a result the value of the target function at the given point $f(x)$. A concept class $\mathcal{C}$ is said to be exactly learnable from equivalence and membership queries, if there exists an algorithm such that for all $f \in \mathcal{C}$ and any equivalence oracle, the algorithm, using the equivalence oracle and a membership oracle, will output, in polynomial time, a hypothesis h that is equivalent to f.

2.2 Incomplete Membership Queries

In this paper, we consider a variant of the exact learning model in which the learning algorithm has access to an *Incomplete Membership Oracle*, which was originally introduced by Angluin and Slonim (1994). Formally, the learner is given access to an oracle MQ^p, called the Incomplete Membership Oracle of Probability p (where clear from the context, we will refer to this oracle as the "Incomplete Membership Oracle", and drop the superscript p from the notation), in addition to the equivalence oracle EQ. MQ^p is a probabilistic function mapping the instance space $\mathcal{X}$ to the set $\{0, 1, \text{I don't know}\}$. When given as input a point $x \in \mathcal{X}$, MQ^p first checks to see if this point was ever given to it before during the execution of the learning algorithm. If so, it answers with the same answer it gave on the previous time. If not, it tosses a biased coin with probability p of landing on "heads". If it lands "heads", MQ^p answers "I don't know". Otherwise, it answers with $f(x)$.

2.3 Monotone DNF

The concept class which we concern ourselves with is the class of Monotone DNF formulas. A Monotone DNF formula f over n variables $x_1, \ldots, x_n$ is the disjunction of t terms, denoted $T_1, \ldots, T_t$. Each term is a monotone monomial in the variables over which the formula is defined.

We may regard an assignment x to the n variables as a binary vector. We further consider the lattice defined over these vectors by the component-wise "and" and "or" operators. This lattice implies the ordering such that $x \leq y$ iff $x_i \leq y_i$ for all $i \in \{1, \ldots, n\}$. Once this partial order is defined, we may uniquely define any monotone term by the minimal assignment that satisfies it. We refer to such an assignment as the *minterm* of a monotone term, and denote the monotone term whose minterm is x by $T(x)$. We denote by $\mathrm{wt}(x)$ the Hamming weight of the assignment x. Given a pair of assignments x and y, we define the distance between them, $\mathrm{dist}(x, y) \stackrel{\triangle}{=} \mathrm{wt}(x + y)$ where $x + y$ is the sum of x and y as vectors over $\mathrm{GF}(2)$. We say y is a descendant of x if $y < x$.

2.4 Previous Results

Angluin and Slonim (1994) presented, along with the introduction of the incomplete membership oracle, an algorithm that could learn Monotone DNF using an EQ Oracle and an Incomplete MQ Oracle of constant probability. Bshouty and Eiron (2001) presented a more precise analysis of the same algorithm that proves the following:

Theorem 1 *(Bshouty and Eiron 2001) For any $p < 1 - 1/2^{\log^{\alpha} n}$ for any constant $\alpha < 1$, the time complexity of Angluin-Slonim's algorithm is*

$$tn^{O\left(\frac{\log \frac{1}{1-p}}{\log\log n}\right)}.$$

In particular, the algorithm runs in polynomial time for any

$$p = 1 - \frac{1}{poly(\log n)}.$$

3. Statement of Main Result

We greatly improve upon previous known results on the learnability of Monotone DNF with incomplete membership queries, by proving the following theorem:

Theorem 2 *There exists an algorithm that, with probability at least $1-\delta$, learns Monotone DNF from EQs plus incomplete MQs of probability p in time (and number of queries)*

$$O\left(\left(\frac{tn}{1-p}\log\frac{1}{\delta}\right)^3\right)$$

Section 4 presents such a learning algorithm and its analysis. Note that, by using $p = 1 - 1/\text{poly}(n, |f|)$, Theorem 2 implies:

Corollary 3 *It is possible to efficiently exact-learn Monotone DNF formulas from EQs and Incomplete MQs, even if the Incomplete MQ Oracle has probability $p = 1 - 1/poly(n, |f|)$ of not providing an answer (i.e., MQs are answered with polynomially small probability).*

4. A New Algorithm for Learning Monotone DNF with Incomplete Membership Queries

In this section we present an algorithm (see Figure 1) that exact-learns the class of Monotone DNF, using an Equivalence Oracle, and an Incomplete Membership Oracle. The algorithm we present runs in time polynomial in the size of the target function (that is n – the number of variables, and t – the number of terms), $\log 1/\delta$, where δ is a confidence parameter, and $1/(1-p)$ where p is the probability that the Incomplete Membership Oracle does not answer a query.

First, we give an informal explanation of the algorithm: The algorithm works by maintaining three sets of points in $\mathcal{X}$. The set N contains points that are known to be negative. The set B contains points that are known to be positive. The set P contains points that are generated by taking the conjunction of a pair of points from B (which implies $|P| = O(|B|^2)$). Points in P for which the classification is known are moved to either B or N, depending on the classification. The algorithm also removes "redundant" points from B and N (a point x is redundant if it is covered by another point in the same set that implies x's classification).

The algorithm initializes N and P to be empty, and B to contain some positive point. It then progresses by presenting the function whose minterms are $B \cup P$ to the EQ oracle. If

Algorithm LearnFromIncomplete:

1. Initialization: $N \leftarrow \emptyset$, $P \leftarrow \emptyset$.
2. $x \leftarrow \mathrm{EQ}(0)$.
   If $x =$"YES" – output $f(x) \equiv 0$ and terminate.
3. Otherwise, let $B \leftarrow \{x\}$.
4. As long as no EQ was answered "YES", do:
   (a) Remove from P all points x for which there exist $x' \in N$ such that $x' \geq x$ or for which there exists $x' \in B$ such that $x' \leq x$.
   (b) Let $H \leftarrow \bigvee_{x \in P \cup B} T(x)$
   Ask $x \leftarrow \mathrm{EQ}(H)$.
   If $x =$ YES, output H and halt.
   If $H(x) = 1$, let $N \leftarrow N \cup \{x\}$, and goto 4a.
   (c) For all $x' \in B$:
   let $z \leftarrow x \wedge x'$,
   let $b \leftarrow \mathrm{MQ}^p(z)$.
   If $b = 1$, insert z to B, remove x' from B.
   If $b = 0$, let $N \leftarrow N \cup \{z\}$.
   If $b =$"I don't know", let $P \leftarrow P \cup \{z\}$.
   (d) If none of the MQs in Step 4c were answered positively, insert x into B.

Figure 1: LearnFromIncomplete – An algorithm for Learning Monotone DNF

a negative counterexample is returned, it must imply that some point in P actually belongs in N. In that case, the algorithm will remove that point from P, and try the EQ again. If the result of the EQ is a positive counterexample, the algorithm will ask Incomplete MQs on all conjunctions of the new point with a point from B. The points are then stored in one of the three sets depending on the result of the queries. A formal analysis of the algorithm follows.

In the sequel, we assume that the target function f has t terms whose minterms are $M_1, \ldots, M_t$ (i.e., $f = \bigvee_{i=1}^{t} T(M_i)$). Furthermore, we denote by B_i the following subset of B:

$$B_i \stackrel{\triangle}{=} \{x \in B \ : \ x \geq M_i\}\,.$$

Lemma 4 *In each complete iteration of Step 4 of LearnFromIncomplete at least one of the following happens: Either a point x is inserted into B such that for all $i \in \{1, \ldots, t\}$, if $x \geq M_i$ then $B_i = \emptyset$, or, at least one MQ is asked on a point y such that for some i for which $y \geq M_i$, y has Hamming weight strictly less than that of any point in B_i. Furthermore, in the second case, this y was never queried on before.*

Proof The algorithm is such that the only way to finish an iteration of Step 4 is if an equivalence query returns a positive counterexample. If the EQ returns a positive counterexample x that satisfies that for all i, if $M_i \leq x$ then no other point exists in B_i then we are done with the proof. We therefore assume that we receive a positive counterexample x such that there is some M_i such that $M_i \leq x$, and B_i is not empty. Let $x' \in B_i$ be a point with minimal Hamming weight in B_i. Consider the point $x \wedge x'$. Since x and x' both satisfy $T(M_i)$, we have that $f(x \wedge x') = 1$. Furthermore, since $T(x')$ was part of the hypothesis we used as input for the EQ oracle, we know that x is not above x' in the lattice order. Hence $x' \wedge x < x'$.

All that remains to be seen is that the point $x' \wedge x$ was never queried before. Indeed, had it been queried before, the result would have been either positive (in which case either $x' \wedge x$ itself, or some point below it would have been inserted into B), or it would have been answered with “I don't know”, in which case it would have been inserted into P, and never be removed from it, as its true label is 1. In both cases, $T(x \wedge x')$ would have been one of the terms in the hypothesis passed to the EQ oracle, in contradiction with the fact that x is a positive counterexample. Hence, $x' \wedge x$ was never passed as a query to the Incomplete Membership Oracle before. ∎

Lemma 5 *With probability at least $1-\delta$, Algorithm LearnFromIncomplete terminates after at most*

$$O\left(\frac{tn}{1-p} \log \frac{1}{\delta}\right)$$

iterations.

Proof Consider the two options that we have for completing an iteration of Step 4, according to Lemma 4: If the point we receive is covered only by minterms M_is for which their B_is were empty, then for at least one of these minterms, from now on, there will always be a point in B that belongs to it (i.e., for at least one B_i that was empty, it will never be empty anymore). Therefore, throughout the run of the algorithm, there may be at most t iterations of Step 4 in which all minterms below the counterexample returned from the EQ oracle do not have other points above them in B (i.e., the first case of Lemma 4 happens).

Consider the other option by which an iteration of Step 4 may be completed, according to Lemma 4: Let i be as in the statement of Lemma 4. Let x' be a point with minimal Hamming weight in B_i. If the MQ on $x' \wedge x$ will be answered by the Incomplete Membership Oracle (which will happen with probability $1-p$, independent of other points in the run of the algorithm), the minimal Hamming weight for points in B_i will be strictly reduced. Since Hamming weight is non-negative, and since no point can have Hamming weight greater than n, the minimal Hamming weight for a point in B_i may be lowered at most n times. Thus, for each $i \in \{1, \ldots, t\}$, at most n "successful" iterations, in which the query on $x' \wedge x$ is answered, may occur during the running of the algorithm. Since each iteration is "successful" for at least one term, with probability $1-p$, we have, by the Chernoff inequality, that after

$$O\left(\frac{tn}{1-p}\log\frac{1}{\delta}\right)$$

iterations, with probability $1-\delta$ we had nt successful iterations, meaning all minimal points in all the sets B_i are the minterms M_i themselves, which means the target was found, and the algorithm will terminate. ■

Lemma 6 *With probability at least $1-\delta$, each iteration of Step 4 of Algorithm LearnFromIncomplete takes time at most:*

$$O\left(\left(\frac{tn}{1-p}\log\frac{1}{\delta}\right)^2\right).$$

Proof In each iteration of Step 4, $|B|$ grows by at most 1. Hence, by Lemma 5, with probability at least $1-\delta$, $|B|$ will never be more than

$$O\left(\frac{tn}{1-p}\log\frac{1}{\delta}\right).$$

In each iteration, MQs are asked for every element in B (which takes $O(|B|)$ time). For each MQ asked, an element may be inserted into P. Thus the maximal size of P is $O(|B|^2)$ (since there are $|B|$ iterations, with possibly $|B|$ elements inserted in each one). Now consider the loop of Step 4a – Step 4b. In each iteration (over these two steps) we receive a negative counterexample to an equivalence query (or else we continue to Step 4c). A negative counterexample x' must have some $x \in P$ such that $x' \geq x$ (or else x' wouldn't have been classified as positive by the hypothesis). Since such an x is removed from P in Step 4a, in each iteration over Step 4a – Step 4b at least one element is removed from P. Thus, there are at most $O(|B|^2)$ EQs made in that stage, before P is emptied, and a positive counterexample must be returned. All in all, the running time of a single iteration is $O(|B|^2)$. This completes the proof of this lemma. ■

Lemma 7 *Throughout the running of Algorithm LearnFromIncomplete, with probability at least* $1-\delta$, *at most*

$$O\left(\left(\frac{tn}{1-p}\log\frac{1}{\delta}\right)^2\right)$$

queries (both EQs and MQs) are made.

Proof Clearly, in each iteration, at most $|B|$ MQs are made. Since $|B|$ is at most the total number of iterations, it follows from Lemma 5 that, w.h.p., at most

$$O\left(\left(\frac{tn}{1-p}\log\frac{1}{\delta}\right)^2\right)$$

membership queries are asked during the running of the algorithm. For each EQ asked, either a point is added to B (if a positive counterexample is returned), or a point is added to N (if a negative counterexample is returned). However, when a point is added to N, it causes at least one point to be removed from P in Step 4a, as explained in the proof of Lemma 6. Thus, the total number of equivalence queries cannot be more than $|B|+|B|^2$ (as there are at most $|B|$ EQs answered positively, and at most $|P| \leq |B|^2$ answered negatively). Using Lemma 5, this gives the required bound on the number of equivalence queries. ■

The proof of Theorem 2 now becomes an immediate application of Lemmas 5, 6, and 7.

5. Conclusion

In this work we have shown that, even when limited to answering a very small fraction of the queries (in a sense, the smallest fraction that can still be used by an efficient algorithm at all), the Membership Query Oracle adds significant power to the learner. We demonstrated how such a weak Incomplete MQ oracle can still be used to allow exact learning of Monotone DNF. Two natural questions arise:

1. Which algorithms may be modified to work with such an Incomplete Membership Oracle? Specifically, can other interesting classes, such as Decision Trees be efficiently learned with such an Incomplete Membership Oracle?

2. Are there any concept classes for which it may be proven that learning with an EQ oracle and an Incomplete MQ oracle that gives answers with polynomially small probability is not possible, but for which efficient learning algorithms for learning with an EQ oracle and a standard MQ oracle exist? This may be restated in the following way: Can we prove that an Incomplete MQ oracle of polynomial probability is strictly weaker than a standard MQ oracle?

Acknowledgments

This research was supported by the fund for the promotion of research at the Technion. Part of this research was done while the first author was at the University of Calgary, Calgary, Alberta, Canada. Part of this research was done while the second author was a graduate student at the Department of Computer Science, Technion – Israel Institute of Technology, Haifa, Israel.

References

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, April 1988.

Dana Angluin and Donna K. Slonim. Randomly fallible teachers: Learning monotone DNF with an incomplete membership oracle. *Machine Learning*, 14:7–26, 1994.

Peter Auer. On-line learning of rectangles in noisy environments. In *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory*, pages 253–261. ACM Press, New York, NY, 1993.

Peter Auer and Nicolò Cesa-Bianchi. On-line learning with malicious noise and the closure algorithm. *Annals of Mathematics and Artificial Intelligence*, 23(1/2):83–99, 1998. Special issue for AII '94 and ALT '94.

Nader H. Bshouty and Nadav Eiron. Learning monotone DNF from a teacher that almost does not answer membership queries. In *14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 2001, Proceedings*, volume 2111 of *Lecture Notes in Artificial Intelligence*, pages 546–557. Springer, Berlin, 2001.

Nader H. Bshouty and Avi Owshanko. Learning regular sets with an incomplete membership oracle. In *14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 2001, Proceedings*, volume 2111 of *Lecture Notes in Artificial Intelligence*, pages 574–588. Springer, Berlin, 2001.

Zhixiang Chen. A note on learning DNF formulas using equivalence and incomplete membership queries. In *Algorithmic Learning Theory, 4th International Workshop on Analogical and Inductive Inference, AII '94, 5th International Workshop on Algorithmic Learning Theory, ALT '94, Reinhardsbrunn Castle, Germany, October 1994, Proceedings*, volume 872 of *Lecture Notes in Artificial Intelligence*, pages 272–281. Springer-Verlag, 1994.

Sally A. Goldman and H. David Mathias. Learning k-term DNF formulas with an incomplete membership oracle. In *Proc. 5th Annual ACM Workshop on Comput. Learning Theory*, pages 77–84. ACM Press, New York, NY, 1992.

Michael Kearns, Ming Li, Leonard Pitt, and Leslie Valiant. On the learnability of Boolean formulae. In *Proc. 19th Annu. ACM Sympos. Theory Comput.*, pages 285–294. ACM Press, New York, NY, 1987.

Nick Littlestone. *Mistake Bounds and Logarithmic Linear-threshold Learning Algorithms*. PhD thesis, Technical Report UCSC-CRL-89-11, Univ. of California Santa Cruz, 1989.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inform. Comput.*, 108(2):212–261, 1994.