

Can Complexity Theory Benefit from Learning Theory?

Tibor Hegedűs

Department of Computer Science

Comenius University, 84215 Bratislava, Slovakia

hegedus@mff.uniba.cs

We show that the results achieved within the framework of Computational Learning Theory are relevant enough to have non-trivial applications in other areas of Computer Science, namely in Complexity Theory. Using known results on efficient query-learnability of some Boolean concept classes, we prove several (co-NP-completeness) results on the complexity of certain decision problems concerning representability of general Boolean functions in special forms.

1. Introduction

The seminal paper of Valiant [11] initiated a considerable amount of research in Computational Learning Theory. The goal of the field is to "give a rigorous, computationally detailed and plausible account of how learning can be done" [2]. This characterization indicates a close relationship to Complexity Theory. In fact, the considered models and criteria of successful learning are all in the spirit of traditional Complexity Theory. In addition to this methodological contribution, it turned out to be the case [10] that complexity-theoretic results can be applied directly to show non-learnability results in the standard PAC model. More precisely, if the CONSISTENCY problem for a class of Boolean functions (deciding whether there is a function in the class consistent with given data) is NP-hard, then that class is not PAC-learnable in a representation-dependent sense (if $RP \neq NP$), i.e., Learning Theory also benefits from some particular results of Complexity Theory.

However, in this paper we show that the cooperation between the two fields is not necessarily destined to be a one-way traffic. To do this, we use a recent result of [5] which says that if a class of Boolean functions is learnable using equivalence and membership queries (again in a representation-dependent sense), then the MEMBERSHIP problem for that class (deciding whether a function given, say, in DNF, belongs to that class), is in co-NP. MEMBERSHIP is essentially the problem of deciding whether a general Boolean function is representable in a special form. Because a number of classes have

been shown to be query-learnable, we obtain several non-trivial inclusions in co-NP for decision problems which are intuitively only in Σ_2^P . In fact, in most cases we prove that the considered problems are co-NP-complete. This is the case of linearly separable functions (functions computable by a single neuron - the basic component of neural networks), threshold functions of order at most k (for any fixed $k \geq 0$), n -dimensional balls, k -DNF and k -CNF functions (for any fixed $k \geq 1$), read-once and read-twice DNF formulas, and k -term DNF (k -clause CNF) formulas for any fixed $k \geq 1$ (using the last result, some progress on the complexity of the MINIMAL DNF EXPRESSION problem [4] is also achieved). As far as the corresponding search problems (find a special representation of a function given in DNF if one exists) are concerned, using query-learnability they can be solved in time $2^{O(nm)}$ (n is the number of variables and m the number of terms in the input).

All these results suggest that Computational Learning Theory is a mature field and its results are relevant enough to have non-trivial applications in other areas of Computer Science.

2. Definitions and models

We denote (for $n \geq 1$) $X_n = \{0, 1\}^n$, $F_n = \{f \mid f : X_n \rightarrow \{0, 1\}\}$. For a function (concept) $f \in F_n$, $\text{POS}(f) = \{x \mid x \in X_n \text{ and } f(x) = 1\}$. A *concept class* is a sequence $C = \{C_n\}_{n \geq 1}$, $C_n \subseteq F_n$ for all $n \geq 1$.

Informally, *learning* a concept is equivalent to identifying it from a given class of possibilities (concept class). The formal models of concept learning are further specified by determining what is a learning algorithm and what are the criteria of successful identification.

A concept class C is *PAC-learnable* if there is a polynomial algorithm A and a polynomial p such that for every $n \geq 1$, every target concept $f \in C_n$, every probability distribution D_n on X_n , and every ϵ, δ ($0 < \epsilon, \delta < 1$) the following holds: if A is given a sample of $p(n, \frac{1}{\epsilon}, \frac{1}{\delta})$ pairs $(x, f(x))$ selected from X_n according to D_n , then it outputs a representation of a *hypothesis* $h \in C_n$ such that with probability at least $1 - \delta$, the *error* of h (the probability that a randomly selected point $x \in X_n$ is classified differently by the target function and the final hypothesis) is at most ϵ (this is the original model introduced by Valiant [11]; see [2, 3] for more details).

If exact, on-line learning is considered, we will say that C is *query-learnable* if for every $n \geq 1$ each concept $f \in C_n$ is exactly identifiable by a polynomial time (in n) algorithm using *equivalence queries* from C_n and *membership queries*. An equivalence query is a question of the type "Does the concept $f \in C_n$ equal to the target concept?" (if not, we are given a counterexample); a membership query is a question like "Does the value of the target concept on $x \in X_n$ equal to 1?" (see [2] for details).

3. Results

The basic idea of [10] in using complexity-theoretic results within the framework of Learning Theory is based on a correspondence between PAC-learnability of a concept class C and the complexity of the decision problem CONSISTENCY for C .

CONSISTENCY for C

Instance: $POS, NEG \subseteq \{0, 1\}^n$ (for some $n \geq 1$).

Question: Is there a function $f \in C_n$ such that $f(x) = 1$ for all $x \in POS$ and $f(x) = 0$ for all $x \in NEG$?

Pitt and Valiant [10] showed that if a class C is PAC-learnable then CONSISTENCY for C is in RP (see [4] for details on RP and other notions of Complexity Theory used in the paper). It follows that to obtain a non-learnability result for a class C (assuming $RP \neq NP$), it suffices to show that CONSISTENCY for C is NP-hard (results of this type are given in [10]; see also [5]). That is, one can make use of pure complexity-theoretic results within the framework of Learning Theory. Unfortunately, the only currently known way to prove that a class C is PAC-learnable is using the inclusion of CONSISTENCY for C in P (or at least in RP), hence one cannot expect that Learning Theory could contribute to Complexity Theory in proving new inclusion results in RP. That is, the approach of [10] establishes a quite one-sided cooperation between the two fields.

However, it is not the case that this cooperation is destined to be a one-way traffic. To see this, we use a recent result of [5] which relates another decision problem to learning tasks.

MEMBERSHIP for C

Instance: A Boolean function f given in DNF.

Question: Is $f \in C$?

The basic argument, due to [5], is the following.

Theorem 3.1. *If a class C is query-learnable, then MEMBERSHIP for C is in co-NP.*

Once again, NP-hardness of MEMBERSHIP for a class C establishes that C is not query-learnable (if $NP \neq \text{co-NP}$). Results along this line are given in [5] for unions (intersections) of k halfspaces (for any fixed $k \geq 2$), corresponding to simple 2-layer neural network architectures (see [5] for details).

Conversely, for classes which are known to be query-learnable, we have that the corresponding MEMBERSHIP problems belong to co-NP. Usually these inclusion results

are not obvious; for "reasonable" classes MEMBERSHIP is - intuitively - in Σ_2^P , but its inclusion in NP or co-NP is much less trivial. Of course, we do not claim that our inclusion proofs based on query-learnability are "nice" or "instructive", but once you know that some result holds true, you can try to find a more instructive proof.

To sum up, now it suffices to list the known positive results on query-learnability to obtain interesting contributions to Complexity Theory.

First, consider some geometrically defined classes of functions corresponding to natural learning systems like neural networks. A Boolean function $f \in F_n$ is *linearly separable* if there exist real weights w_1, \dots, w_n and a real threshold t such that $\text{POS}(f) = \{x \mid x \in X_n \text{ and } \sum_{i=1}^n w_i x_i \geq t\}$. *Threshold functions of order at most k* (for any fixed $k \geq 0$) mean a generalization of linear separability: weights are assigned to all monomials $x_{i_1} x_{i_2} \dots x_{i_l}$ of the input attributes, $0 \leq l \leq k$; n -dimensional balls are defined analogously. All the described concept classes are query-learnable (even without membership queries) [7, 8, 9], hence the following results can be obtained.

Theorem 3.2. *It is co-NP-complete to decide whether a Boolean function given in DNF is*

- (1) *linearly separable;*
- (2) *of threshold order at most k ;*
- (3) *an n -dimensional ball.*

Proof: The inclusions in co-NP follow from the query-learnability of the listed classes; the co-NP-hardness results are proved by reduction from the DNF-TAUTOLGY problem (given a Boolean function f in DNF, does f equal to the constant *one* function?). See [6] for details, where direct proofs of inclusions in co-NP are given using a result from combinatorial geometry (Helly's theorem) and a linear programming approach (parts (1) and (2) correct some mistakes in the literature, where the considered decision problems were claimed to be NP-complete). ■

A further class which is query-learnable without membership queries is k -DNF (k -CNF) for any fixed $k \geq 1$ [11]. These are Boolean functions representable as DNF (CNF) expressions with at most k literals in each term (clause).

Theorem 3.3. *For any fixed $k \geq 1$, it is co-NP-complete to decide whether a Boolean function given in DNF (CNF) belongs to k -DNF (k -CNF).*

Proof: We only have to prove co-NP-hardness. Let k be fixed, $k \geq 1$, and consider the following reduction from DNF-TAUTOLGY: given $f \in F_n$ in DNF, construct a DNF for the function $g(x, y_1, \dots, y_{k+1}) = f(x) \vee y_1 \dots y_{k+1}$. One can show that $f \equiv 1$ if and only if g is in k -DNF; the result for k -CNF follows. ■

Some concept classes are known to be not query-learnable without membership queries (if $RP \neq NP$), but they become learnable if membership queries are allowed. However, we can make use of these learnability results in exactly the same way as above.

The class *k-term DNF* (*k-clause CNF*) - for any fixed $k \geq 1$ - is the class of Boolean functions representable by DNF (CNF) expressions with at most k terms (clauses). It is known that for any fixed $k \geq 1$ *k-term DNF* (*k-clause CNF*) is query-learnable [1], hence we easily obtain the following.

Theorem 3.4. *For any fixed $k \geq 1$, it is co-NP-complete to decide whether*

- (1) *a Boolean function (given in DNF) has a DNF with at most k terms;*
- (2) *a Boolean function (given in CNF) has a CNF with at most k clauses.*

Proof: To prove co-NP-hardness for (1), use the following reduction from DNF-TAUTOLOGY: for $f \in F_n$ in DNF, construct a DNF representation for the function $g(x, y_1, \dots, y_{k+1}) = f(x) \vee y_1 \vee \dots \vee y_{k+1}$; clearly, $f \equiv 1$ if and only if g can be expressed as a DNF with at most k terms; part (2) follows from part (1). ■

As a by-product, some progress on the complexity of the famous MINIMAL DNF EXPRESSION [4] problem is achieved.

MINIMAL DNF EXPRESSION

Instance: A Boolean function f in DNF and an integer $K \geq 1$.

Question: Has f a DNF representation with at most K terms?

Theorem 3.5. *The MINIMAL DNF EXPRESSION problem is NP-hard, co-NP-hard, and belongs to Σ_2^P .*

Proof: The NP-hardness result is given in [4], the co-NP-hardness follows from our previous arguments, and the inclusion in Σ_2^P is straightforward. ■

Further examples of query-learnable classes are *read-once* and *read-twice* DNF formulas [2]. These are Boolean functions representable by DNF formulas where each variable appears at most once (twice). For this class we can prove the following.

Theorem 3.6. *It is co-NP-complete to decide whether a Boolean function (given in DNF) is a read-once (read-twice) DNF formula.*

Proof: It suffices to use the following reduction from DNF-TAUTOLOGY: given a function $f \in F_n$ in DNF, construct a DNF representation for the function $g(x, y_1, y_2) =$

$f(x) \vee y_1 y_2 \vee \bar{y}_1 \bar{y}_2$. One can show that $f \equiv 1$ if and only if g can be expressed as a read-once DNF. The case of read-twice DNF formulas is handled analogously. ■

Finally, consider the search version of the above problems: given a Boolean function $f \in F_n$ in DNF (with m terms), construct a special representation of f if one exists (say, find the weights and the threshold of a neuron computing the given function if one exists). A simple simulation of a query learning algorithm gives the following result.

Theorem 3.7. *If a class C is query-learnable, then the corresponding search problem for C can be solved in time $2^{O(nm)}$.*

References

- [1] D. Angluin, "Learning k -term DNF Formulas Using Queries and Counterexamples", Technical Report, Yale University, YALE/DCS/RR-559, 1987.
- [2] D. Angluin, "Computational Learning Theory: Survey and Selected Bibliography", in: *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*, 1992, pp. 351–369.
- [3] M. Anthony and N. Biggs, *Computational Learning Theory*, Cambridge University Press, Cambridge, 1992.
- [4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
- [5] T. Hegedűs, "Computational Limitations on PAC and On-Line Learning over the Boolean Domain: a Comparison", submitted for publication.
- [6] T. Hegedűs and N. Megiddo, "On the Geometric Separability of Boolean Functions", submitted for publication.
- [7] W. Maass and Gy. Turán, "On the Complexity of Learning from Counterexamples", in: *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Angeles, 1989, pp. 262–267.
- [8] W. Maass and Gy. Turán, "Algorithms and Lower Bounds for On-Line Learning of Geometrical Concepts", Report 316, IIG-Report Series, Graz University of Technology, 1991.
- [9] W. Maass and Gy. Turán, "How Fast can a Threshold Gate Learn?" Report 321, IIG-Report Series, Graz University of Technology, 1991.
- [10] L. Pitt and L. Valiant, "Computational Limitations on Learning from Examples", *Journal of the ACM* 35 (1988) 965–984.
- [11] L. Valiant, "A Theory of the Learnable", *Communications of the ACM* 27 (1984) 1134–1142.