

Hardness of Proper Learning (1988; Pitt, Valiant)

Vitaly Feldman, IBM Research - Almaden, researcher.ibm.com/view.php?person=us-vitaly

entry editor: Rocco A. Servedio

INDEX TERMS: Proper learning, PAC learning, NP-hardness of learning, DNF, function representation, representation-based hardness.

1 PROBLEM DEFINITION

The work of Pitt and Valiant [18] deals with learning Boolean functions in the Probably Approximately Correct (PAC) learning model introduced by Valiant [19]. A learning algorithm in Valiant's original model is given random examples of a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ from a representation class \mathcal{F} and produces a hypothesis $h \in \mathcal{F}$ that closely approximates f . Here a *representation class* is a set of functions and a language for describing the functions in the set. The authors give examples of natural representation classes that are NP-hard to learn in this model whereas they can be learned if the learning algorithm is allowed to produce hypotheses from a richer representation class \mathcal{H} . Such an algorithm is said to learn \mathcal{F} by \mathcal{H} ; learning \mathcal{F} by \mathcal{F} is called *proper learning*.

The results of Pitt and Valiant were the first to demonstrate that the choice of representation of hypotheses can have a dramatic impact on the computational complexity of a learning problem. Their specific reductions from NP-hard problems are the basis of several other follow-up works on the hardness of proper learning [1, 3, 7].

1.1 Notation

Learning in the PAC model is based on the assumption that the unknown function (or *concept*) belongs to a certain class of concepts \mathcal{C} . In order to discuss algorithms that learn and output functions one needs to define how these functions are represented. Informally, a representation for a concept class \mathcal{C} is a way to describe concepts from \mathcal{C} that defines a procedure to evaluate a concept in \mathcal{C} on any input. For example, one can represent a conjunction of input variables by listing the variables in the conjunction. More formally, a representation class can be defined as follows.

Definition 1. A representation class \mathcal{F} is a pair (L, \mathcal{R}) where

- L is a language over some fixed finite alphabet (e.g. $\{0, 1\}$);
- \mathcal{R} is an algorithm that for $\sigma \in L$, on input $(\sigma, 1^n)$ returns a Boolean circuit over $\{0, 1\}^n$.

In the context of efficient learning, only efficient representations are considered, or, representations for which \mathcal{R} is a polynomial-time algorithm. The concept class represented by \mathcal{F} is set of functions over $\{0, 1\}^n$ defined by the circuits in $\{\mathcal{R}(\sigma, 1^n) \mid \sigma \in L\}$. For a Boolean function f , “ $f \in \mathcal{F}$ ” means that f belongs to the concept class represented by \mathcal{F} and that there is a $\sigma \in L$ whose associated Boolean circuit computes f . For most of the representations discussed in the context of learning it is straightforward to construct a language L and the corresponding translating function \mathcal{R} , and therefore they are not specified explicitly.

Associated with each representation is the complexity of describing a Boolean function using this representation. More formally, for a Boolean function $f \in \mathcal{C}$, $\mathcal{F}\text{-size}(f)$ is the length of the shortest way to represent f using \mathcal{F} , or $\min\{|\sigma| \mid \sigma \in L, \mathcal{R}(\sigma, 1^n) \equiv f\}$.

We consider Valiant's PAC model of learning [19] as generalized by Pitt and Valiant [18]. In this model, for a function f and a distribution \mathcal{D} over X , an *example oracle* $EX(f, \mathcal{D})$ is an oracle that, when invoked, returns an example $\langle x, f(x) \rangle$, where x is chosen randomly with respect to \mathcal{D} , independently of any previous examples. For $\epsilon \geq 0$ we say that function g ϵ -approximates a function f with respect to distribution \mathcal{D} if $\Pr_{\mathcal{D}}[f(x) \neq g(x)] \leq \epsilon$.

Definition 2. *A representation class \mathcal{F} is PAC learnable by representation class \mathcal{H} if there exist an algorithm that for every $\epsilon > 0$, $\delta > 0$, n , $f \in \mathcal{F}$, and distribution \mathcal{D} over X , given ϵ , δ , and access to $EX(f, \mathcal{D})$, runs in time polynomial in $n, s = \mathcal{F}\text{-size}(c), 1/\epsilon$ and $1/\delta$, and outputs, with probability at least $1 - \delta$, a hypothesis $h \in \mathcal{H}$ that ϵ -approximates f .*

A DNF expression is defined as an OR of ANDs of literals, where a *literal* is a possibly negated input variable. We refer to the ANDs of a DNF formula as its *terms*. Let $\text{DNF}(k)$ denote the representation class of k -term DNF expressions. Similarly a CNF expression is an OR of ANDs of literals. Let $k\text{-CNF}$ denote the representation class of CNF expressions with each AND having at most k literals.

For a real-valued vector $c \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$, a *linear threshold function* (also called a *halfspace*) $T_{c,\theta}(x)$ is the function that equals 1 if and only if $\sum_{i \leq n} c_i x_i \geq \theta$. The representation class of Boolean threshold functions consists of all linear threshold functions with $c \in \{0, 1\}^n$ and θ an integer.

2 KEY RESULTS

Theorem 3 ([18]). *For every $k \geq 2$, the representation class of $\text{DNF}(k)$ is not properly learnable unless $\text{RP} = \text{NP}$.*

More specifically, Pitt and Valiant show that learning $\text{DNF}(k)$ by $\text{DNF}(\ell)$ is at least as hard as coloring a k -colorable graph using ℓ colors. For the case $k = 2$ they obtain the result by reducing from Set Splitting (see [9] for details on the problems). Theorem 3 is in sharp contrast with the fact that $\text{DNF}(k)$ is learnable by $k\text{-CNF}$ [19].

Theorem 4 ([18]). *The representation class of Boolean threshold functions is not properly learnable unless $\text{RP} = \text{NP}$.*

This result is obtained via a reduction from the NP-complete Zero-One Integer Programming problem (see [9](p.245) for details on the problem). The result is contrasted by the fact that general linear thresholds are properly learnable [4].

These results show that using a specific representation of hypotheses forces the learning algorithm to solve a combinatorial problem that can be NP-hard. In most machine learning applications it is not important which representation of hypotheses is used as long as the value of the unknown function is predicted correctly. Therefore learning in the PAC model is now defined without any restrictions on the output hypothesis (other than it being efficiently evaluatable). Hardness results in this setting are usually based on cryptographic assumptions (*cf.* [15]).

Hardness results for proper learning based on assumption $\text{NP} \neq \text{RP}$ are now known for several other representation classes and for other variants and extensions of the PAC learning model. Blum and Rivest show that for any $k \geq 3$, unions of k halfspaces are not properly learnable [3]. Hancock *et al.* prove that decision trees (*cf.* [16] for the definition of this representation) are not learnable by decision trees of somewhat larger size [11]. This result was strengthened by Alekhovich *et al.* who also prove that intersections of two halfspaces are not learnable by intersections of k halfspaces

for any constant k , general DNF expressions are not learnable by unions of halfspaces (and in particular are not properly learnable), and k -juntas are not properly learnable [1]. Further, DNF expressions remain NP-hard to learn properly even if *membership queries*, or the ability to query the unknown function at any point, are allowed [7]. Khot and Saket show that the problem of learning intersections of two halfspaces remains NP-hard even if a hypothesis with any constant error smaller than $1/2$ is required [17]. No efficient algorithms or hardness results are known for any of the above learning problems if no restriction is placed on the representation of hypotheses.

The choice of representation is important even in powerful learning models. Feldman proved that n^c -term DNF are not properly learnable for any constant c even when the distribution of examples is assumed to be uniform and membership queries are available [7]. This contrasts with Jackson’s celebrated algorithm for learning DNF in this setting [13], which is not proper.

In the *agnostic learning* model of Haussler [12] and Kearns *et al.* [14] even the representation classes of conjunctions, decision lists, halfspaces, and parity functions are NP-hard to learn properly (*cf.* [2, 6, 8, 10] and references therein). Here again the status of these problems in the representation-independent setting is largely unknown.

3 APPLICATIONS

A large number of practical algorithms use representations for which hardness results are known (most notably decision trees, halfspaces, and neural networks). Hardness of learning \mathcal{F} by \mathcal{H} implies that an algorithm that uses \mathcal{H} to represent its hypotheses will not be able to learn \mathcal{F} in the PAC sense. Therefore such hardness results elucidate the limitations of algorithms used in practice. In particular, the reduction from an NP-hard problem used to prove the hardness of learning \mathcal{F} by \mathcal{H} can be used to generate hard instances of the learning problem.

4 OPEN PROBLEMS

A number of problems related to proper learning in the PAC model and its extensions are open. Almost all hardness of proper learning results are for learning with respect to unrestricted distributions. For most of the problems mentioned in Section 2 it is unknown whether the result is true if the distribution is restricted to belong to some natural class of distributions (e.g. product distributions). It is unknown whether decision trees are learnable properly in the PAC model or in the PAC model with membership queries. This question is open even in the PAC model restricted to the uniform distribution only. Note that decision trees are learnable (non-properly) if membership queries are available [5] and are learnable properly in time $O(n^{\log s})$, where s is the number of leaves in the decision tree [1].

An even more interesting direction of research would be to obtain hardness results for learning by richer representations classes, such as AC^0 circuits, classes of neural networks and, ultimately, unrestricted circuits.

5 EXPERIMENTAL RESULTS

None is reported.

6 DATA SETS

None is reported.

7 URL to CODE

None is reported.

8 CROSS REFERENCES

None is reported. (Fill in later.)

9 RECOMMENDED READING

- [1] M. Alekhnovich, M. Braverman, V. Feldman, A. Klivans, and T. Pitassi. The complexity of properly learning simple classes. *Journal of Computer and System Sciences*, 74(1):16–34, 2008.
- [2] S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003.
- [3] A. L. Blum and R. L. Rivest. Training a 3-node neural network is NP-complete. *Neural Networks*, 5(1):117–127, 1992.
- [4] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.
- [5] N. Bshouty. Exact learning via the monotone theory. *Information and Computation*, 123(1):146–153, 1995.
- [6] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami. On agnostic learning of parities, monomials and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009.
- [7] Vitaly Feldman. Hardness of approximate two-level logic minimization and pac learning with membership queries. *Journal of Computer System Sciences*, 75(1):13–26, 2009.
- [8] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM J. Comput.*, 41(6):1558–1590, 2012.
- [9] M. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman, San Francisco, 1979.
- [10] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM J. Comput.*, 39(2):742–765, 2009.
- [11] T. Hancock, T. Jiang, M. Li, and J. Tromp. Lower bounds on learning decision lists and trees. In *12th Annual Symposium on Theoretical Aspects of Computer Science*, pages 527–538, 1995.
- [12] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.
- [13] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.
- [14] M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.
- [15] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

- [16] M. Kearns and U. Vazirani. *An introduction to computational learning theory*. MIT Press, Cambridge, MA, 1994.
- [17] Subhash Khot and Rishi Saket. On the hardness of learning intersections of two halfspaces. *J. Comput. Syst. Sci.*, 77(1):129–141, 2011.
- [18] L. Pitt and L. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.
- [19] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.