Critique of Theory of Learnable by Leslie Valiant

In Computer Science or in any other field of study, it is many a times important to know the feasibility of a solution to a problem. Many problems which might appear complex can be toned down to simpler ones by somehow devising a means for the solution of the problem to be less complex.

A typical case of this is the *Search Problem.* When you have to search an infinite space for the required result, it is almost impossible to solve the problem. But by utilizing some techniques, the result can be easily acquired. Most of the times, there would be several ways to accomplish a given task but to find out which would be the best solution, you would need a common yardstick of measurement. This problem of the feasibility study and the rating of the simplicity/complexity of the solution comprises the ***Theory of Computation***. This field of Computer Science deals with

1. Time Complexity
2. Space Complexity
3. And a few others…

Likewise, in the field of Artificial Intelligence, it is imperative to measure another form of complexity i.e., the ability of an algorithm to learn. Hence, Leslie Valiant felt the need and proposed the ***Theory of Learnability*** which sets some parameters to judge the ability of the algorithm to learn.

He sets 3 conditions for a concept to be learnable by the algorithm:

1. The algorithms/learning machines must be able to wholly learn the classes of concepts. Each of the classes must be characterizable.
2. The classes of concepts (the learning) must be non-trivial and appropriate for general-purpose knowledge.
3. The deduction/inference must happen within feasible (polynomial) number of steps.

The author in this paper has proved his proposition using Boolean functions. Hence all the variables considered are Boolean irrespective of whether the variable is independent or not. The function that is output is also Boolean.

$$X^N = p_1, p_2, \dots\dots p_n \rightarrow Input\ Variables \in \{0, 1, *\ (undetermined)\}^N$$

$$f: \{0, 1\}^N \rightarrow \{0, 1\}$$

$$f, F, h \in H\ are\ all\ Hypothesis/functions\ belonging\ to\ the\ set\ of\ all\ possible\ Hypothesis$$

$$D \rightarrow True\ Vector\ i.e., X^N \notin \{*\}$$

There can be many solutions for a given positive example and this set of solutions are referred to as *Distributions* in the paper. Since we deal with only Boolean functions, it is expected that a solution vector X^N from the distribution of input set where X^N is a solution set and a total vector adds to 1 i.e.,

$$\sum_i P_D(X_i) = 1\ \forall\ X_i\ such\ that\ f(X_i) = 1$$

Defining these parameters, an algorithm can be said to be probably approximately learnable if it satisfies the below combinatorial bound:

Let's define for two positives integers, h and S (with $h > 1$), a function $L(h,S)$ which is the minimal integer such that $L(h,S)$ is the number of independent Bernoulli trials with a probability h^{-1} of success and $P(number\ of\ success\ < S) < h^{-1}$.

It characterizes the minimum learning sample set size needed to have with a probability $1 - h^{-1}$, an accuracy of $S/(dataset\ test\ size)$.

$$For\ all\ integers\ S \geq 1\ and\ all\ real\ h > 1,$$

$$L(h,S) \leq 2h(S + \log_e h)$$

This combinatorial bound is further proven by using CNF, DNF and μ-expressions.

References:

1. http://www.montefiore.ulg.ac.be/~geurts/Cours/AML/Discussions/8-learning-theory/slides.pdf
2. https://blog.acolyer.org/2018/01/31/a-theory-of-the-learnable/
3. https://stats.stackexchange.com/questions/69554/pac-learning-theory-and-lower-bound-on-the-amount-of-input-samples
4. https://stats.stackexchange.com/questions/142906/what-does-pac-learning-theory-mean?rq=1
5. https://stats.stackexchange.com/questions/353935/pac-learning-definition-and-the-properties-of-the-problem?rq=1