4

Hierarchical Learning of Boolean Functions

Patrick Suppes and Shuzo Takahashi

1 Introduction

The learning of Boolean functions has been the focus of a number of papers in computer science in the last few years, many of which have been stimulated by the work of Valiant [3, 4]. We also mention especially the recent long article by Blumer *et al.* [1] on learnability for various classes of concepts including especially Boolean classes. The thrust of most of this theoretical work has essentially been to show under what conditions concepts can be learned in polynomial time or space. It is of course important to show that learning is feasible for as wide a class of concepts as possible. However, it is also important to focus on detailed results of a very finitistic kind, which are characteristic of learning theory as applied to human behavior, and are of intrinsic interest also in machine learning. In this paper we consider conditions under which a hierarchy of Boolean functions can increase in significant ways the rate of learning of a given Boolean function of n variables. In this context we also compare serial and parallel hierarchical learning, with, as might be expected, the most dramatic effects being produced by parallel hierarchical learning. Because in general the results are better for parallel than for serial learning the reader might wonder why we include a systematic study of serial learning. The reason is that even under serial learning conditions hierarchical learning can often lead to dramatic improvement in the rate of learning.

There is some literature on hierarchical learning, but the results are not closely related to ours. We refer the reader to Rivest and Sloan [2], and the references given there.

There are three distinct psychological processes involved in our theoretical approach to hierarchical learning. They are (i) stimulus sampling, often given in the form of a pattern; (ii) concept sampling; and (iii) function formation—meaning forming some function of the concepts sampled. In some contexts it may be useful to refer to function formation as hypothesis sampling. In any case, process (i) brings in information from the world outside, but (ii) and (iii) are essentially internal.

P. I. Bystrov and V. N. Sadovsky (eds.), Philosophical Logic and Logical Philosophy, 51–61.

With probabilistic assumptions about each process, a variety of measures of learning can be defined and many of them have been used in the literature. Because we are interested in comparing a number of different assumptions about each process in terms of effects on the rate of learning, we have chosen as the comparative measure we compute the expected number of trials to learn the concept in question, either with no residual response error in the model or with some specified lower bound on the probability of an error or the expected number of trials. By and large we do not analyze possible processes of function formation, in spite of their interest for a general theory of learning. We comment on this point in more detail later.

1.1 Relation to neural networks

It is important to emphasize that hierarchies, as defined and used in this article, are not neural networks, except in the most trivial sense. The reason is simple. In the hierarchies we consider there are no hidden units. Each level of a hierarchy can provide reinforcement or feedback on learning any Boolean function occurring at that level. This assumption is made for all the different cases we consider.

On the other hand, wide variations in sampling assumptions are made. The most important dichotomy is between (i) the assumption of sampling of input to a Boolean function at one level of a hierarchy being independent of such sampling at any other level, and (ii) the assumption that the sampling distribution is given at the lowest level of the hierarchy, the level of stimulus patterns, and this distribution then determines the sampling at all higher levels.

We underscore the point that it would be a mistake to suppose that sampling assumption (ii) is almost always more realistic than assumption (i). In the teaching of a complex hierarchical subject like mathematics or a foreign language, students in general sample artificial exercises at each level of learning, essentially independent of their distribution of occurrence as components of real-world problems to be encountered later.

The hierarchies we consider are not neural nets, but it is obvious that a neural net could be added at each node by so formulating the learning of each Boolean function at a node of the hierarchy. This we have not done, for reasons that will become clear later.

2 Some Preliminaries

A Boolean function of n variables is a function from $\{0,1\}^n$ to $\{0,1\}$. We use variables F, G, H, F_1, F_2, etc. for Boolean functions. A pattern is an element in $\{0,1\}^n$. Variables p, q, p_1, p_2, etc. are used for patterns. When there are n variables, then there are 2^n patterns, and Boolean functions can also be defined by subsets of the 2^n patterns. For this reason we shall sometimes refer to Boolean functions as concepts defined by subsets of the patterns. The usual convention is that the

Boolean function has the value 1 for the patterns that are instances of the concept, and 0 otherwise.

The kind of learning process we study here is what is called in the literature supervised learning from examples only. This means in classical learning terms that the learner is only presented examples and is informed after each trial what is the correct response. We can assume that on each trial the learner guesses whether a given pattern is an instance of the concept or not, that is, whether the Boolean function has the value 1 or not for this pattern. Note that under the assumption of feedback we are making here it is not important whether the response is correct or incorrect. What is important is the assumption that the learner internalizes the correct answer.

We can now note a familiar result, but one that is important for comparison. If the learner is asked to learn an arbitrary Boolean function purely from presentation of examples, then in general there are for n variables at least 2^n trials required. To achieve this lower bound of 2^n, we must assume that the learner is perfect in the sense of remembering and correctly registering the feedback on each trial. Then there is no way that the learning can be speeded up without the introduction of information that goes beyond the pure presentation of examples. This means that for a function of 1000 variables, which could easily arise in a vision system, the learning of patterns would require 2^{1000} trials, obviously an unfeasibly large number. Valiant and others have emphasized learning restricted classes of functions that will not require such an exponentially large number of trials. The analysis that we give of hierarchical learning is similarly motivated but moves in a different direction.

We introduce some further notation useful in studying hierarchies. The variables x, y, z, x_1, x_2, etc. take values in $\{0, 1\}$, and f, g, h, f_1, f_2, etc. are function symbols with certain arities. $A(f)$ denotes the arity, or number of arguments, of f. Terms are recursively defined from variables and function symbols. For each term t, we define a set Q_t which contains all function symbols in t. This can be recursively defined (we omit the precise definition). For example, for $t = f(x, y)$, $Q_t = \{f\}$, and for $t = f(g(x_1, x_2), h(x_1, x_2, x_3))$, $Q_t = \{f, g, h\}$. Also $|Q_t|$ is the cardinality of Q_t. For each term t, the number of distinct variables in t is denoted by V_t. For example, for $t = f(x, y)$, $V_t = 2$, and for $t = f(g(x_1, x_2), h(x_1, x_2, x_3))$, $V_t = 3$.

Throughout we use, with or without arguments or subscripts, N for nonhierarchical learning, S for serial hierarchical learning and P for parallel hierarchical learning.

3 Serial Hierarchical Learning

In this section we compare nonhierarchical and serial hierarchical learning of Boolean functions. Intuitively the comparison of nonhierarchical and serial hierarchical learning in the framework we have defined is straightforward, given that in the present analysis we are concerned with perfect learning. Later we shall con-

sider learning to a particular statistical criterion, of the sort discussed by Valiant. The nonhierarchical learning of an arbitrary Boolean function of n variables is just learning the correct response to each of the 2^n patterns. The hierarchical learning depends upon learning each function in a corresponding way. In other words, each individual function, at any level in a hierarchy, is a special case of nonhierarchical learning. More explicitly for each term t, N_t and S_t are defined by

$$N_t = 2^{V_t}$$

and

$$S_t = \sum_{f \in Q_t} 2^{A(f)}.$$

These two quantities are used to measure the rates for nonhierarchical and serial hierarchical learning, as just mentioned above. For example, for $t = (f(x, y), N_t = 2^2 = 4$ and $S_t = 2^2 = 4$, for $t = f(g(x_1, x_2), h(x_1, x_2, x_3)), N_t = 2^3 = 8$ and $S_t = 2^2+2^2+2^3 = 16$, and for $t = f(g(x_1, x_2, x_3), g(x_2, x_3, x_4)), N_t = 2^4 = 16$ and $S_t = 2^2 + 2^3 = 12$.

Note that the quantities N_t and S_t measure exactly the rates of learning under the assumptions that *sampling is without replacement and that at each level of a hierarchy only relevant concepts are sampled.* There is no need to say sampling is uniform. The only requirement is that on each trial a pattern is sampled that has not been sampled before. When studying perfect learning under the nonreplacement assumption, what actual distribution is used for sampling does not affect the number of trials for the results we are now computing.

The most extreme sort of example favoring serial hierarchical learning over nonhierarchical learning is given by:

$$t = h(g(f(x_1, x_2), f(x_3, x_4)), g(f(x_5, x_6), f(x_7, x_8))). \tag{1}$$

For this t,

$$N_t = 2^8 = 256,$$

while

$$S_t = 3 \times 2^2 = 12.$$

More generally, for any term t with n variables, which has a similar structure. $N_t = 2^n$ while $S_t = 4 \log_2 n$.

The most extreme example favoring nonhierarchical learning is given by:

$$t = h(g_1(f_1(x_1, x_2), f_2(x_1, x_2)), g_2(f_3(x_1, x_2), f_4(x_1, x_2))). \tag{2}$$

For this t,

$$N_t = 2^2 = 4,$$

while

$$S_t = 7 \times 2^2 = 28.$$

A less extreme yet interesting example, also of interest for parallel processing, is given by:

$$t = h(g_1(f_1(x_1, x_2), f_2(x_3, x_4)), g_2(f_3(x_5, x_6), f_4(x_7, x_8))). \quad (3)$$

For this t,

$$N_t = 2^8 = 256,$$

while

$$S_t = 7 \times 2^2 = 28.$$

Figure 1 shows the hierarchy of t as defined by (3). And more generally, for any term t with n variables, which has a similar structure, i.e., n is a power of 2, $N_t = 2^n$ while $S_t = 4(n-1)$.

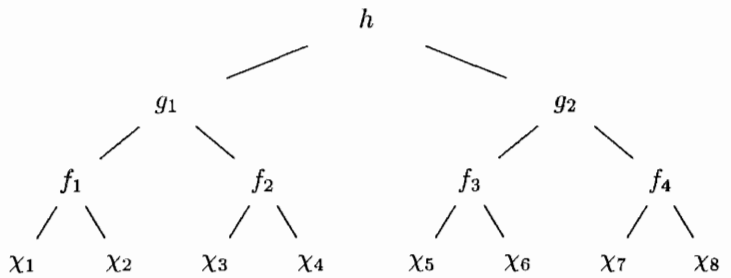

Figure 1: Hierarchy defined by Equation (3).

3.1 Sufficient conditions for $S \leq N$

We now examine sufficient conditions for $S \leq N$ (henceforth we omit the subscript t from the notation), that is, for

$$\sum_{f \in Q} 2^{A(f)} \leq 2^V. \quad (4)$$

First of all, it is easy to show that

Theorem 3.1 *If* $V \leq 3$, *then* $N \leq S$.

Thus our main interest is in the case $V \geq 4$. It is difficult to analyze (4) without a further simplification[1]. We start with the following simple (rather crude) estimation of S as the basis of our analysis of $S \leq N$:

$$\sum_{f \in Q} 2^{A(f)} \leq |Q| 2^M \quad (5)$$

[1]We might also consider some structural assumptions, such as the depth of terms is 1. However, it seems that such cases are already as difficult as the general case, without further simplifications as exemplified below.

where $M = \max\{A(f) : f \in Q\}$. Then, a sufficient condition for $S \leq N\ (= 2^V)$ is

$$|Q|2^M \leq 2^V. \tag{6}$$

Now two cases may be considered: $M = 2$ and M/V is constant, i.e., $M = hV$ for some constant h.

These two cases are further analyzed in the following two subsections.

3.1.1 Analysis of $S \leq N$ in the case $M = 2$

Putting $M = 2$ and $|Q|/V = k$ where $0 < k$, (6) becomes

$$4kV \leq 2^V. \tag{7}$$

It is easy to show that a necessary and sufficient condition for (7) (for $V \geq 4$) is $k \leq 1$. Thus we obtain

Theorem 3.2 *For a binary term with at least 4 variables, if the number of the function symbols in the term does not exceed the number of the variables in the term, then $S \leq N$.*

It is also interesting to consider the ratio $r = S/N$ in the case $S \leq N$. For this, replacing k by k/r in the above argument, we obtain

Theorem 3.3 *For a binary term with at least 4 variables, if $|Q|/V \leq r$, then $S/N \leq r$.*

3.1.2 Analysis of $S \leq N$ in the case $M = hV$

Putting $M/V = h$ and $|Q|/V = k$ where $0 < h < 1$ and $0 < k$, (6) becomes

$$kV \leq 2^{(1-h)V}. \tag{8}$$

Let us consider the function

$$\phi(V) = 2^{(1-h)V} - kV. \tag{9}$$

The function ϕ has its minimum at

$$m(h,k) = \frac{1}{(1-h)\log 2} \log \frac{k}{(1-h)\log 2}.$$

Hence, the inequality (8) holds for all V if and only if

$$\phi(m(h,k)) \geq 0,$$

which is equivalent to

$$k \leq (e \log 2)(1-h). \tag{10}$$

Thus we obtain

Theorem 3.4 *A sufficient condition for $S \leq N$ is that (10) holds. (Notice that if (10) holds for h and k, then it holds for any $h' \leq h$ and $k' \leq k$).*

We are also interested in the ratio $r = S/N$ in the case $S \leq N$. This can be easily obtained by replacing k by k/r in the above derivation, that is,

Theorem 3.5 *If the following inequality holds*

$$\frac{k}{r} \leq (e \log 2)(1 - h), \tag{11}$$

then $S/N \leq r$.

Note that $e \log 2$ is about (yet bigger than) 1.88. As an instance of (11), if $M/V \leq 0.60$ and $|Q|/V \leq 0.56$, then $S/N \leq 0.75$.

In spite of starting with the rather crude estimation (5), the numbers we obtained for h, k, and r look reasonable. Now let us consider what kind of improvements for h, k, and r can be made. First of all, notice that Theorems 3.4 and 3.5 are valid for all V, and do not reflect the following fact:

If V gets larger, then (8) holds for larger h and k.

To use this fact, let us consider the following condition:

$$C(h, k, V_0) \text{ if and only if for all } V \geq V_0, kV \leq 2^{(1-h)V}. \tag{12}$$

Note that if we want to consider the ratio $r = S/N$, then simply replace k by k/r. Simple properties of this condition are:

- For all h and k, there exists V_0 such that $C(h, k, V_0)$.
- If $C(h, k, V_0), h' \leq h$, and $k' \leq k$, then $C(h', k', V_0)$.
- If we expect V_0 to be small, then both h and k must also be sufficiently small.
- If we expect h and k to be large, then V_0 must also be sufficiently large.
- Even though h is large, if k is sufficiently small, then V_0 can be reasonably small.
- Even though k is large, if h is sufficiently small, then V_0 can be reasonably small.

Next we pose the problem: Given V_0, find a simple relation between h and k such that $C(h, k, V_0)$.

Theorems 3.4 and 3.5 were obtained by considering $\phi(m(h, k)) \geq 0$, but this does not give a good result in the case $m(h, k) \leq V_0$, which is equivalent to

$$k \leq ((1 - h) \log 2) 2^{(1-h)V_0}. \tag{13}$$

In this case, a sufficient condition for $C(h, k, V_0)$ is $\phi(V_0) \geq 0$, which is equivalent to

$$k \leq \tfrac{1}{V_0} 2^{(1-h)V_0}. \tag{14}$$

Combining (13) and (14), we obtain

Theorem 3.6 *If*

$$k \leq \min\{(1-h)\log 2, \frac{1}{V_0}\} 2^{(1-h)V_0},$$

then for $V \geq V_0$

$$S \leq N.$$

Substituting k/r for k, we obtain

Theorem 3.7 *If*

$$k/r \leq \min\{(1-h)\log 2, \frac{1}{V_0}\} 2^{(1-h)V_0},$$

then for $V \geq V_0$

$$\frac{S}{N} \leq r.$$

3.2 Uniform Sampling with replacement of stimulus patterns

Let E_N be the expected number of trials to sample all stimulus patterns under non-hierarchical learning when the sampling distribution is uniform with replacement, and let E_S be the corresponding expected number for serial learning with the sampling distribution uniform with replacement for each function at any level of the hierarchy. Thus, for example, if we consider the hierarchy of Figure 1, the expected number of trials to sample all patterns for each of the 7 binary functions is 25/3, so $E_S = 58.3$. These numbers follow at once from the following familiar results.

Let E_N be the expected number of trials under the uniform distribution to completely sample with replacement all N patterns. First, the sampling may be represented by the Markov chain whose transition matrix is given by:

$$\begin{array}{c} \\ 0 \\ 1 \\ j \\ N-1 \\ N \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & j & j+1 & N-1 & N \end{array} \\ \begin{pmatrix} & 1 & & & & & \\ & \frac{1}{N} & \frac{N-1}{N} & & & & \\ & & & \frac{j}{N} & \frac{N-j}{N} & & \\ & & & & & \frac{N-1}{N} & \frac{1}{N} \\ & & & & & & 1 \end{pmatrix} \end{array}$$

The probability distribution of the number of trials in the state $j < N$ is given by:

$$P_{N,j}(n) = \left(\frac{j}{N}\right)^{n-1} \left(1 - \frac{j}{N}\right).$$

Hence,

$$E_{N,j} = \sum_{n=1}^{\infty} nP_{N,j}(n) = \sum_{n=1}^{\infty} n \left(\frac{j}{N}\right)^{n-1} \left(1 - \frac{j}{N}\right) = \frac{1}{1 - \frac{j}{N}} = \frac{N}{N-j}.$$

So, the expected number of trials for completely sampling all N patterns is:

$$E_N = \left(\sum_{j=1}^{N-1} E_j\right) + 1 = \left(\sum_{j=1}^{N-1} \frac{N}{N-j}\right) + 1 = N \sum_{k=1}^{N} \frac{1}{k}.$$

For $N = 4$,

$$E_N = 4(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}) = \frac{25}{3},$$

as stated above. (Note that learning for any given function in the hierarchy is itself nonhierarchical, just depending on the number of patterns to be sampled.) In contrast to $E_S = 58.3$ for Figure 1, the nonhierarchical learning of 2^8 patterns under uniform sampling is

$$E_N(8) = 256 \sum_{k=1}^{256} \frac{1}{k},$$

which is a much larger number. Explicitly,

$$E_N(8) \approx 1567.3$$

More generally, for the same hierarchical structure of binary functions as that of Figure 1, but for arbitrary n,

$$E_S(n) = \frac{25}{3}(n-1),$$

for uniform sampling without replacement, and for the nonhierarchical case

$$E_N(n) = 2^n \sum_{k=1}^{2^n} \frac{1}{k} \approx 2^n(n \log 2 + 0.577),$$

since it may be shown from well-known results that for any $n > 0$

$$n \log 2 + 0.577 \leq \sum_{k=1}^{2^n} \frac{1}{k}.$$

4 Parallel Hierarchical Learning

We turn now to a brief treatment of parallel hierarchical learning. The general assumption is that at each level of the hierarchy all functions at that level are being learned simultaneously, i.e., on the same trials.

To provide some immediate intuitive comparisons, consider the hierarchy of Figure 1 with 8 variables. As discussed earlier, $N_t = 256$ and $S_t = 28$, but $P_t = 12$, i.e., just 4 trials for each level of the hierarchy, of which there are 3. For that same structure generalized to n variables, as previously noted $N_t = 2^n$ and $S_t = 4(n-1)$. We state the result for P_t as a theorem.

Theorem 4.1 *With learner control of pattern selection at each level for the hierarchical pattern of Figure 1 generalized to n variables, $P_t = 4\log_2 n$.*

PROOF. The proof is quite simple. The four selected patterns for parallel processing at the initial stage and recursively then for each next level have the simple form of presenting the same values to every binary function at a given level. For example,

x_1x_2	x_3x_4	.....	$x_{n-1}x_n$
11	11	.....	11
10	10	.....	10
01	01	.....	01
00	00	.....	00

Finally, it is easy to see that for this special hierarchy the number of levels is $\log_2 n$.

The reduction from $N_t = 2^n$ to $P_t = 4\log_2 n$ is dramatic, but of the sort required for massive parallel networks like the human visual system to be feasible. Of course, the strict condition of learner control of pattern selection at each level of a hierarchy will never be satisfied in nonexperimental natural environments where the distribution of stimulus patterns is not under learner control. Moreover, this natural distribution also fixes the sampling at every level of the hierarchy.

Even under desirable sampling distributions such as the uniform distribution, whether independent sampling occurs at each level or only once at the bottom level can, as would be expected, make a big difference in the rate of learning. Referring once again to the type of hierarchy shown in Figure 1 generalized to n variables, we make the following specific additional assumptions to illustrate the impact of this variation in sampling patterns.

(i) The 2^n stimulus patterns are divided into $2^{\frac{n}{2}}$ classes and each class contains exactly $2^{\frac{n}{2}}$ patterns.

(ii) For each class the functions $f_i, i = 1, \ldots, 2^{n-1}$ at the first level above the bottom are the same Boolean function for all patterns in a class.

(iii) But all 16 binary Boolean functions are used by the various classes, with a uniform distribution across classes.

For serial hierarchical learning with independent uniform sampling for each binary function at every level the expected number of trials for complete learning is given above, namely, $4(n-1)$, and for parallel processing under the same sampling assumption, $4\log_2 n$.

In contrast, when sampling occurs only at the lowest level of the hierarchy—a realistic assumption in many environments—, but is uniformly distributed at that level, and the hierarchy satisfies conditions (i)-(iii), then the expected number of

trials for complete learning is bounded from below by

$$2^{\frac{n}{2}} \sum_{k=1}^{2^{\frac{n}{2}}} \frac{1}{k},$$

since a necessary condition of learning is that at least one stimulus pattern must be sampled from each class.

The nature of this example shows that even with parallel hierarchical processing, if the sampling pattern is determined only by sampling at the lowest level of the hierarchy, then in relatively simple cases learning will take exponentially many trials. The strongly contrasting results for independent sampling at each level of the hierarchy suggest why the teaching of any complex subject is broken up into independent stages of learning.

Stanford University, USA

References

[1] A. Blumer, A. Ehrenfeucht, D. Haussler and M. K. Warmuth. Learnability and the Vapnik–Chervonenkis dimension. *Journal of the Association for Computing Machinery*, **36**, 929–965, 1989.

[2] R. L. Rivest and R. Sloan. Learning complicated concepts reliably and usefully. [Extended Abstract]. *Proceedings of the 7th National Conference on Artificial Intelligence*, **2**, 635–640, 1988.

[3] L. G. Valiant. A theory of the learnable. *Journal of the Association for Computing Machinery*, **27**, 1134–1142, 1984.

[4] L. G. Valiant. Learning disjunctions of conjunctions. *Proceedings of the 9th International Conference on Artificial Intelligence*, **1**, 560–566, 1988.