Thoughts on Hypothesis Boosting.

Machine Learning class project, Dec. 1988

Michael Kearns

In this paper we present initial and modest progress on the *Hypothesis Boosting Problem*. Informally, this problem asks whether an efficient learning algorithm (in the distribution-free model of [V84]) that outputs an hypothesis whose performance is only slightly better than random guessing implies the existence of an efficient algorithm that outputs an hypothesis of arbitrary accuracy. The resolution of this question is of theoretical interest and possibly of practical importance. From the theoretical standpoint, we are interested more generally in the question of whether there is a discrete hierarchy of achievable accuracy in the model of [V84]; from a practical standpoint, the collapse of such a proposed hierarchy may yield an efficient algorithm for converting relatively poor hypotheses into very good hypotheses.

Our goals are to offer some simple observations and results that seem related to the above questions, with the eventual goal of resolving the Hypothesis Boosting Problem. We begin with the definitions of strong and weak learnability.

Let C and H be parameterized classes of representations of Boolean functions; that is, $C = \cup_{k \geq 1} C_k$ and $H = \cup_{k \geq 1} H_k$ where C_k and H_k are representations of Boolean functions on $\{0,1\}^k$. We assume that C and H are *polynomially evaluable*: there is a polynomial-time algorithm that on input a representation c and a vector x computes the value of $c(x)$. We refer to C and H as the *target class* and the *hypothesis class* respectively. We assume that the representations in C and H are written under some standard encoding, and will denote the length in bits of a representation c by $|c|$.

For a given *target representation* $c \in C$ we assume there are fixed but arbitrary *target distributions* D^+ and D^- over the positive and negative examples of c respectively. Learning algorithms have access to oracles POS and NEG that sample these distributions in unit time.

We say that C *is strongly learnable by* H if there is an algorithm A with access to POS and NEG, taking inputs $0 < \epsilon, \delta < 1$, with the property that for any target representation $c \in C_k$, for any target distributions D^+ and D^-, and for any ϵ and δ, algorithm A runs in time polynomial

in $\frac{1}{\epsilon}, \frac{1}{\delta}, |c|$ and k and outputs a representation $h \in H$ that with probability at least $1 - \delta$ satisfies $D^+(c - h) \leq \epsilon$ and $D^-(h) \leq \epsilon$. Here we have identified c and h with the set of points on which they evaluate to 1.

We say that C *is weakly learnable by* H if there is a polynomial p and an algorithm A with access to POS and NEG, taking input $0 < \delta < 1$, with the property that for any target representation $c \in C_k$, for any target distributions D^+ and D^-, and for any δ, algorithm A runs in time polynomial in $\frac{1}{\delta}, |c|$ and k and outputs a representation $h \in H$ that with probability at least $1 - \delta$ satisfies $D^+(c - h) \leq \frac{1}{2} - \frac{1}{p(|c|,k)}$ and $D^-(h) \leq \frac{1}{2} - \frac{1}{p(|c|,k)}$.

We will call A a (*strong, weak*) *learning algorithm* for C. Further, we say that C is (*strongly, weakly*) *learnable* if it is learnable by H for some polynomially evaluable H.

We can now formulate the Hypothesis Boosting Problem more formally as follows: is it the case that any C that is weakly learnable is in fact strongly learnable? Note that we pose the question in a *representation-independent* setting in that the hypothesis class used by the strong learning algorithm need not be the same as that used by the weak learning algorithm. We will also be interested in the question of whether the weak learnability of C by H always implies the strong learnability of C by H.

Our first observation stems from the differing motivations for the strong and weak learning models. The motivation for the strong learning model seems clear: the desire for the efficient induction of highly accurate hypotheses from examples. Given that these examples are stochastically generated, perfect accuracy is not possible, so strong learning is the most we can ask of an algorithm. What is the motivation for the weak learning model? Here we are asking for an hypothesis whose accuracy is slightly better than random guessing — in particular, we are comparing the hypothesis of the algorithm to the hypothesis that flips a fair coin to decide its answer to each unseen example. This comparison is natural only if H can incorporate the power of a fair coin. As a specific example of a case where this comparison may be unfair, note that if C is the class of two-term DNF formulae and H the class of monomials, it is not clear that there is an algorithm for C using H that even approaches the performance of random guessing, much less weak learning. However, for the purposes of proving theorems and analyzing the behavior of algorithms, allowing H to always include arbitrary probabilistic polynomial time algorithms is somewhat unwieldy.

In particular, H no longer defines a class of concepts, so the notion of the Vapnik-Chervonenkis dimension is no longer defined, and Occam's Razor-like arguments no longer apply since we are unable to bound the size of hypotheses. We are thus led to ask the following question: is there a *deterministic* class H such that for any polynomially evaluable C, there is an efficient algorithm for C using H that matches the performance of random guessing? Our first theorem says that if H is the class of poly-random functions in the sense of [GGM86] (restricted to the first output bit so as to obtain a well-defined concept class), then the following algorithm RANDOM works for any C: for target concept $c \in C_k$, choose $h \in H_k$ uniformly and at random, and output h.

Theorem 1. If there exists a one-way function, then for any polynomially evaluable C, algorithm RANDOM matches the performance of random guessing within an additive factor that vanishes faster than any inverse polynomial.

Proof Idea: We first prove a lemma stating that if representations in H_k are at most polynomially larger than representations in C_k, then any level of accuracy that can be achieved by a polynomial time algorithm for C using H under the guarantee that the distributions are generated in polynomial time can in fact be achieved in polynomial time regardless of the distribution. This implies that any distribution that is "hard" for learning C by H can be assumed to be generated in polynomial time without loss of generality. Now let H be the class of poly-random functions defined above. If RANDOM fails to match the performance of random guessing, we obtain a contradiction as follows: since RANDOM fails, there is some concept with distributions that are generated in polynomial time on which RANDOM fails to match random guessing with some non-negligible probability. These distributions can be used as a polynomial-time statistical test for functions as follows: query the unknown function on points drawn from the hard distributions and test the accuracy of the unknown function. If it is noticeably different from that of random guessing, vote "poly-random"; otherwise, vote "truly random". ♠

Thus, we have a class of small deterministic circuits parameterized by seeds of length k that can always be used to efficiently find an hypothesis as good as guessing. An interesting open question is to find simpler classes with the same property, or to prove that certain classes do not have this property. For example, for which classes C can we always efficiently find a 5-term DNF that is as good as guessing? Note that such questions may have both computational and information-

theoretic aspects: for some H, hypotheses as good as guessing may simply not exist for some C; in other cases, they may exist but be intractable to find. Lest this seem to be somewhat far afield of the Hypothesis Boosting Problem, note that the weak learning model essentially asks for small correlations in the underlying distributions; from this viewpoint, the issue of how succinctly this correlation can be expressed is likely to be a central one.

We next note that the question of the equivalence of weak and strong learning is inherently a computational one, in the sense that the number of examples required to achieve weak learning regardless of computation time is essentially the same as the number of examples required for strong learning. This follows from the results in [EHKV87], where is is shown that the number of examples needed for strong learning is $\Omega(\frac{1}{\epsilon}\ln\frac{1}{\delta} + \frac{d}{\epsilon})$, where d is the Vapnik-Chervonenkis of the target class C. This result in shown in [EHKV87] even for *fixed* ϵ; thus, setting $\epsilon = \frac{1}{2}$ verifies the above claim. Using arguments similar to those in [HKLW88] we can equate weak learning with the problem of finding an hypothesis consistent with slightly more than half of an input sample, and thus state a slightly stronger result:

Theorem 2. Any class C that is weakly learnable from a polynomial number of examples and exponential time is strongly learnable from a polynomial number of examples and exponential time.

Let R denote a restriction on the target distributions; for example, R might be the restriction that the distributions can be generated in polynomial time, or that no point in the domain has zero probability. Then learning under restriction R is defined in the obvious way.

Theorem 3. There exist classes C and restrictions R such that C is weakly learnable under R, but C is not strongly learnable under R (under assorted cryptographic assumptions).

Proof Idea: Consider the quadratic residue concept class (each concept is defined by n, the product of two primes; the positive examples are the quadratic residues, the negative examples are the quadratic non-residues), augmented with a special point z_1 that is a positive example of all concepts, and a special point z_2 that is a negative example of all concepts. The restricted distribution class always gives z_1 one percent of D^+ and z_2 one percent of D^-; the rest of the distributions are uniform over the residues and non-residues. Then C is clearly weakly learnable under R by simply answering in the obvious way on z_1 and z_2 and flipping a fair coin to answer on other points. However, to achieve accuracy even 51 percent under restriction R implies that

the algorithm has a 1 percent probability of answering correctly on a randomly chosen residue or non-residue, and this contradicts the Quadratic Residue Assumption. ♠

Theorem 3 brings us to a vague but interesting issue: it says that any proposed proof that weak learning implies strong learning must make use of the fact that C is weakly learnable under *every* distribution, not just the target distribution. Unfortunately, we cannot say more about how this fact may be used in the proof — in particular, it places no obvious restrictions on the behavior of the learning algorithm. For instance, suppose we intend to prove that weak learning implies strong learning by direct simulation: using the weak learning algorithm as an oracle, we give an efficient algorithm for strong learning. As pointed out by [K88], Theorem 3 does *not* directly imply that such a simulation must alter the underlying distribution when calling the weak learning oracle — perhaps, for example, the guarantee that C is weakly learnable is strong enough to always imply the existence of an efficient algorithm for finding a (completely) consistent hypothesis, and no calls to the oracle are even necessary! This observation becomes more intriguing when it is applied to particular proposed simulations, such as the following MAJORITY algorithm: run the weak learning algorithm "enough" times, each time on the same underlying target distributions. When classifying an unseen example, classify it according to a majority vote of the weak learning hypotheses. The fact that this algorithm must fail to boost the hypotheses obtained under restriction R of Theorem 3 in no way precludes the possibility that such a simulation always works in the case that C is weakly learnable, since the C of Theorem 3 is not weakly learnable under *arbitrary* distributions. The MAJORITY algorithm, however, is simple and plausible enough that it warrants being discredited by other means.

Consider the following algorithm A for weakly learning monotone monomials: Over k variables, the initial hypothesis of A is $x_1 \cdots x_k$. A is then the same as Valiant's algorithm, except that after each time a variable is crossed out of the current hypothesis h (due to its negation appearing in a positive example), the hypothesis is tested against random examples to see if it meets the conditions of weak learnability. Note that we know the hypothesis never makes a mistake on negative examples, so we test only against positive examples; if h classifies noticeably more than half of these correctly, halt and output h. We now show that when A is the weak learning algorithm used as the oracle in the MAJORITY algorithm above, MAJORITY fails to strongly learn.

Let the number of variables be $k = 5$ and let the target monomial be $c = x_1$. We define the following distribution D^+:

$$D^+(10111) = .51$$
$$D^+(11011) = .24$$
$$D^+(11101) = .24$$
$$D^+(11110) = .01$$

The exact values given above are not as important as the following observation: on a "typical" run of A, the vector 10111 is received as the first positive example, and the resulting hypothesis is $h = x_1x_3x_4x_5$. In subsequent testing, it is then determined that h has accuracy .51 on D^+, thus meeting the requirements of weak learning. Note that such a typical run occurs with probability .51; thus, most runs have failed to cross out the other irrelevant variables. In particular, at least 99 percent of the runs fail to cross out x_5. Thus a majority vote of the hypotheses from many runs classifies 11110 as negative with high probability, so the error of the hypothesis of the MAJORITY algorithm on the above distribution with the given A is at least .01. Arbitrary accuracy is *not* achieved.

We now introduce yet a third notion of learnability called *skewed learning* that we feel is relevant to the Hypothesis Boosting Problem. Informally, we will say that C is *skew learnable* by H if the hypothesis output has zero error on the distribution D^+ (D^-) and error at most $1 - \frac{1}{p(|c|,k)}$ on the distribution D^- (D^+), for some polynomial p. Several relaxations of this definition are possible without affecting what is said below; for example, we can ask for accuracy only $1 - \epsilon$ on the distribution for which high accuracy is achieved, rather than accuracy 1. However, we will keep this basic definition for simplicity.

Intuitively, skew learning asks that the hypothesis explain all of the positive examples and some nonnegligible fraction of the negative examples. That this implies not only weak learnability but strong learnability is the content of the next theorem.

Theorem 4. If C is skew learnable, then C is strongly learnable.

Proof Idea: We sketch the proof for the case where the error on D^+ is zero; there is a dual proof for the case where the error on D^- is zero. The basic idea is that of "filtering": run the skew learning algorithm A many times, each time training A on the failures of the previous hypotheses.

More formally, run A once to obtain skew hypothesis h_1. Then run A again, but when a negative example is requested, only give those negative examples that are classified as positive by h_1 (i.e., h_1 acts as a filter for the second run of A). The positive examples are always passed through to the current run unfiltered. In general, on the kth run of A, the negative examples that are filtered through are those that are classified as positive by *all* of $h_1, \ldots, h_{k-1}$. Thus each run of A covers a "piece" of D^- of size $\frac{1}{p(|c|,k)}$ that was previously uncovered. If we run A approximately $l \approx p(|c|, k)$ times, the hypothesis $h_1 \wedge \ldots \wedge h_l$ should have high accuracy on D^-. Note that there is no degradation of performance on D^+ since all of the h_i have zero error on D^+. ♠

Note that in the relaxed definition of skew learning in which the accuracy on D^+ is $1-\epsilon$ instead of 1, the proof of Theorem 4 still goes through, except that each iteration of algorithm A may introduce as much as ϵ error on D^+. Thus Theorem 4 may be regarded as providing a method of trading accuracy on D^+ for accuracy on D^-; we can also prove simple but useful generalizations of this technique.

As a brief aside, we point out that Theorem 4 can be applied to obtain an alternative algorithm for learning k-term DNF by kCNF. We sketch the idea for 2-term DNF. Suppose that the target concept is $T_1 + T_2$. Then the following is a skew learning algorithm : take a large sample of both positive and negative examples and output the disjunction $l_1 + l_2$ that is satisfied by all of the positive examples and the fewest of the negatives. It is not hard to show that this is a skew learning algorithm where the error on D^- is at most $1 - \frac{1}{n^2}$, where here n is the number of variables. Applying Theorem 4 to this skew learning algorithm gives an algorithm outputting a kCNF. The resulting algorithm is less efficient than Valiant's algorithm both in terms of sample complexity and computational complexity. However, if we are interested in less than arbitrarily small error, we can stop the simulation in the proof of Theorem 4 after as many iterations as are necessary to reach the desired accuracy. In this case the hypothesis obtained may be shorter and more quickly evaluated than that given by Valiant's algorithm. It would be interesting to find other applications of Theorem 4.

Note that one intuitive way of achieving weak learning is the following: find some small but nonnegligible (with respect to the target distributions) region S which the right classification is known with certainty or very high accuracy. If we are lucky and a new point falls in S, we

answer correctly; if we are unlucky we flip a fair coin to decide the classification. The accuracy of this method is approximately $\frac{1}{2} + p$, where p denotes the probability of S. In the case that a weak learning algorithm works by this method, we can use Theorem 4 to obtain a strong learning algorithm by simply always answering positive whenever the new point falls outside of S, rather than flipping a coin to decide. It is for this reason that the equivalence of strong and weak learning does not seem implausible: intuitively, if a weak learning algorithm does not work by the above region-finding method, it has instead found a small (slightly larger than $\frac{1}{2}$) over a large area of the distribution. However, as noted earlier in this paper, such a correlation may be quite difficult to express deterministically, especially with a restricted hypothesis space. For this reason we conjecture that at the least it is possible to prove results of the form "If C is weakly learnable by H, then C is strongly learnable" for *arbitrary* C but *particular* H.

We conclude by proposing three algorithms for converting weak learning algorithms into strong learning algorithms, both based on the idea of filtering. One of our next research projects will be to try to discredit these algorithms or (hopefully) prove one of them correct.

Algorithm A_1 runs the weak learning algorithm many times, each time training the algorithm on the failures of previous hypotheses; that is, an example is filtered through only if *all* previous hypotheses answered incorrectly on the example. After doing this enough times, the hypothesis of A_1 is a majority vote.

Algorithm A_2 is the same as A_1 except an example is filtered through if a majority of the previous hypotheses failed on it. The hypothesis of A_2 is again a majority vote.

Algorithm A_3 is the same as A_1 except that a single previous hypothesis is chosen randomly as the filter — that is, a previous h_i is chosen at random, and the current run is trained on the failures of h_i only. The hypothesis of A_3 is a majority vote.

Note that even the question of whether hypotheses of accuracy $\frac{1}{2} + \alpha_1$ can be boosted to hypotheses of accuracy $\frac{1}{2} + \alpha_2$ for any interesting $\alpha_2 > \alpha_1$ remains open.

Bibliography.

[EHKV87] A. Ehrenfeucht, D. Haussler, M. Kearns, L. Valiant, "A general lower bound on the number of examples needed for learning", to appear in *Information and Computation*.

[GGM86] O. Goldreich, S. Goldwasser, S. Micali, "How to construct random functions", *JACM*, 33(4), 1986.

[HKLW88] D. Haussler, M. Kearns, N. Littlestone, M. Warmuth, "Equivalence of models for polynomial learnability", submitted to *Information and Computation*.

[K88] J. Killian, personal communication, 1988.

[V84] L. Valiant, "A theory of the learnable", *Comm. ACM*, 27(11), 1984.