AN INVESTIGATION OF EXHAUSTIVE LEARNING

by

David H. Wolpert[*], Alan Lapedes[**]

* - Theoretical Division and Center for Non-linear Studies, Los Alamos National Laboratory, Los Alamos, NM, 87545. Currently at The Santa Fe Institute, 1660 Old Pecos Trail, Suite A, Santa Fe, NM, 87501. (dhw@santafe.edu)

** - Theoretical Division, Los Alamos National Laboratory, and The Santa Fe Institute.

SFI TR-92-04-20, LA-UR-92-2408

Abstract: An extended version of the Bayesian formalism is reviewed. We use this formalism to investigate the "exhaustive learning" scenario, first introduced by Schwartz et al. This scenario is perhaps the simplest possible supervised learning scenario. It is identical to the noise-free "Gibbs learning" scenario studied recently by Haussler et al., and can also be viewed as the zero-temperature limit of the "statistical mechanics" work of Tishby et al. We prove that the crucial "self-averaging" assumption invoked in the conventional analysis of exhaustive learning does not hold in the simplest non-trivial implementation of exhaustive learning. Therefore the central result of that analysis, that generalization accuracy necessarily rises as training set size is increased, is not generic. More importantly, we show that if one (reasonably) changes the definition of "generalization accuracy", to reflect only the error on inputs <u>outside</u> of the training set, then this central result does not hold even when the self-averaging assumption *is* valid, and even in the limit of an infinite input space. This implies that the central result is a reflection of the following simple phenomenon: if you add an input/output pair to the training set, the number of distinct input values on which you know exactly how you should guess has either increased or stayed the same, and therefore your generalization accuracy will either increase or stay the same. In addition to using the extended Bayesian formalism to analyze the central result of the conventional analysis of exhaustive learning, we also use it to extend the results of exhaustive learning, to issues not considered in previous analyses of the subject.

INTRODUCTION

This paper concerns the problem of inductive inference, sometimes also known as (supervised) machine learning. For current purposes this problem can be formulated as follows: We have an *input space* X and an *output space* Y. There is an unknown function from X to Y which will be referred to as the *target* function. One is given a set of m samples of the target function (the *training set*), perhaps made with observational noise. One is then given a value from the input space as a *question*. The problem is to use the training set to guess what output space value on the target function corresponds to the given question. Such a guessed function from questions to outputs is known as a *hypothesis* function. An algorithm which produces a hypothesis function as a guess for a target function, basing the guess only on the training set of m $(X \times Y)$ vectors read off of that target function, is called a *generalizer*.

This paper is an investigation of the "exhaustive learning" scenario recently introduced by Schwartz et al. [Schwartz et al. 1990, Van der Broeck and Kawai 1991, Hertz et al. 1991]. This scenario is perhaps the simplest possible supervised learning scenario. It is identical to the noise-free "Gibbs learning" scenario studied recently in [Haussler et al. 1991], and can also be viewed as the zero-temperature limit of the "statistical mechanics" work of Tishby et al. [Tishby et al. 1989]. It consists of examining the noise-free performance of generalizers of the following type: exclude all hypothesis functions not consistent with the training set, and guess randomly amongst the rest. The central result of the conventional analysis of exhaustive learning [Schwartz et al. 1990] is that in this scenario, when "self-averaging" holds, $P(E \mid f, m) \sim (1 - E)^m$, where E is a value of the generalization error, f is a target function, and m is a training set size. The implication of this result is that as m increases, generalization improves.

The formalism used in this paper, which was introduced in [Wolpert 1992], is an extension of conventional Bayesian analysis. This formalism doesn't restrict itself to a certain kind of generalizer (as do the various versions of the "statistical mechanics" machine learning formalism, for example [Tishby et al. 1989, Seung et al. 1991]), nor does it restrict itself to finding worst-case bounds, where one assumes one knows very little about the generalizer (as does PAC, for example [Blumer et al. 1987, Blumer et al. 1989, Valiant 1984]). It does not need to assume that one only knows the size of the training set, and not the actual data in the training set (like both PAC and the statistical mechanics formalisms). It does not assume one's "priors" are correct, like conventional Bayesian analysis (see [Buntine and Weigend 1991 and references therein]). The extended Bayesian formalism is completely general, and the previously mentioned formalisms are special cases of it.

Using the extended Bayesian formalism we prove that the crucial "self-averaging" assumption invoked in the conventional analysis of exhaustive learning does not hold in the simplest non-trivial implementation of exhaustive learning. Therefore the central result of that analysis (hereafter called "the central result"), that generalization accuracy necessarily rises exponentially as training set size m is increased, is not generic. (This might be part of the explanation for the results reported [Cohn and Tesauro 1991].)

More importantly, this paper proves that a reasonable change in the definition of what one calls the "generalization accuracy", to reflect only the error on inputs <u>outside</u> of the training set, negates the central result even when self-averaging *does* hold. Perhaps surprisingly, this is true even in the limit of infinite input spaces. In other words, the central result is a reflection of the following phenomenon: If you add an input/output pair to the training set, the number of distinct input values on which you know exactly how you should guess has either increased or stayed the same, and therefore your generalization accuracy will either increase or stay the same. This phenomenon would, taken naively, appear to only result in a linear rise in generalization accuracy with training set size in the exhaustive learning scenario, where generalization accuracy reflects on-training set as well as off-training set errors. However, as this paper shows, it actually results in exhaustive learning's *exponential* rise ($P(E \mid f\ m) \sim (1 - E)^m$).[1]

This result serves as a caution against reading too much meaning into exponential gains in generalization accuracy when that accuracy reflects on-training set as well as off-training set errors. Note in particular that PAC's error measure reflects on-training set as well as off-training errors. Note also that the foundational theorem of PAC, $\delta \le (1 - \varepsilon)^m \times |H'|$ ([Blumer et al. 1987]), has a form almost identical to that of the central result of exhaustive learning. See the end of section III.

To understand all of this intuitively, it helps to consider the case where when the generalizer "guesses randomly among ... the hypothesis functions ... consistent with the training set", "randomly" means according to a uniform distribution over allowed hypothesis functions h. Since such a uniform-distribution generalizer will guess any one of the h consistent with the training set with equal probability, its behavior is equivalent to guessing randomly off of the training set. In other words, it is the following rule: for a question equivalent to one of the inputs in the training set, answer with the corresponding output. For any other question, flip a uniformly weighted coin.

To see why such a uniform-distribution generalizer obeys $P(E \mid f, m) \sim (1 - E)^m$, note that exhaustive learning uses an i.i.d. (independent identically distributed) error function, thereby allowing test set questions to range over the training set as well as off of it. In addition, $P(E \mid f, m)$ has a fixed (though unknown) f, and averages over h's. Because of all this, for a binary output space, and for a uniform-distribution generalizer, the average error E - the average number of errors for randomly chosen test questions - is trivially given by (n - m') / 2n, where n is the size of the input space, and m' is the number of distinct pairs in the training set (this result assumes that the sampling distribution over the input space is uniform). Now exhaustive learning also uses an i.i.d. likelihood function, thereby allowing duplicates in the training set. This means that m' is usually less than m. Therefore when expressed in terms of m, the straight line giving the average error develops a "tail"; it gets stretched, pulled towards $m = \infty$. It look like 1 / m, or e^{-m}, in fact. And as Schwartz et al. demonstrated, such average errors are precisely what one would expect if the probability of error E goes like $(1 - E)^m$.

Presumably this uniform distribution generalizer isn't the one Schwartz et al. had in mind. However there is nothing whatsoever in the definition of exhaustive learning to exclude this distribution. In fact, all of the general mathematical results which have been reported on exhaustive learning hold for this generalizer. In addition, the mathematics associated with the uniform distribution serves as the "backbone" of the mathematics for other distributions. Changing the distribution changes the details of the analysis (e.g., constants changes, precise forms of monotonic functions change, etc.), and generally makes the analysis more complicated. But it doesn't affect the underlying behavior. (E.g., the central result of exhaustive learning still holds, assuming self-averaging.)

In short, exhaustive learning is, at its core, identical to random guessing off of the training set. Accordingly, one should view the theorems of exhaustive learning as "meaningful" only to the degree that one is willing to ascribe such meaning even for random off-training set guessing.

In addition to elaborating the foregoing, this paper also does the following:

1) It rederives the mathematics of exhaustive learning in a context more general than that of neural nets and in a manner more rigorous than has been done previously.

2) Doing this allows one to discern that the conditional probability of interest is in exhaustive learning is P(E | f, m). Together with other factors clarified by the analysis in (1), this demonstrates that exhaustive learning is very closely related to PAC (e.g., PAC also examines P(E | f, m)). It also demonstrates that exhaustive learning is not a variation of conventional Bayesian analysis, as some have surmised. (This paper briefly discusses the relative strengths and weaknesses of exhaustive learning in comparison with conventional Bayesian analysis.)

3) Let h be an input-output function guessed by the exhaustive learning generalizer. The probability distribution over h's induced by running the generalizer before it has seen any training examples is T(h), i.e., T(h) = P(h | empty training set). T(h) is the distribution defining "randomly" in the definition of the exhaustive generalizer given above. This paper shows that T(h) is not the prior over h as some have implied (e.g., [Haussler et al. 1991]). In other words, P(h | empty training set) = T(h) ≠ P(h). Rather, as this paper formally proves, P(h) is the distribution, evaluated *before* seeing any training examples, of what h the generalizer will guess *after* learning has taken place. This helps clarify what the term "prior probability" means in supervised learning.

4) This paper also clarifies the following: Exhaustive learning is sometimes couched as though its input space were uncountably infinite (e.g., $\mathbf{R}^K$). However because in exhaustive learning the num-

ber of distinct hypothesis functions possibly output by the generalizer is countable, the input space is effectively discrete. The mathematics can not “access” the full freedom inherent in an uncountably infinite input space. This discreteness of X can often aid one’s intuition, for example when considering if there are any X-space duplicates in the training set L.

5) This paper proves that if the T(h) mentioned in (3) is independent of h, then as is claimed in the conventional analysis of exhaustive learning [Schwartz et al. 1990], “self-averaging” obtains in the limit where the input space is infinitely bigger than the training set. Accordingly, for such a T(h), the central result of exhaustive learning applies. This result holds despite the fact that for some schemes in which h is parameterized in terms of a vector **w** (e.g., perceptrons), self-averaging does not hold when T(**w**) is uniform.

6) This paper shows how to calculate things exactly, without recourse to limits and any consequent “self-averaging”, for the T(h) constant case mentioned in (5).

7) This paper shows that self-averaging, and therefore the central result of exhaustive learning, does **not** hold for the simplest non-constant T(h), no matter what limits one operates in.

8) Exhaustive learning calculates P(error value = E | target function f, training set size m). This paper shows how to calculate P(error value = E | target function f, training set L), where L has m elements. The result differs from P(error value = E | target function f, training set size m), even if P(error value = E | target function f, training set L) is independent of what m input-output pairs make up L. However in the limit of n >> m, the two probability distributions agree.

9) In practice, one either may or may not want to use an error function which runs over the training set. For example, if testing is done through the same process that generated the training set, ultimately one’s “error” gets measured with such a testing set. On the other hand, if testing is *not* done through the same process as training, such a measure of error can be inappropriate.

More importantly, if we use an error function which runs over the training set, then the upper bound on the generalization error will shrink with m. To get a “normalized” measure of generalization accuracy, one might wish to correct for this effect. Such a correction results in use of an off-training set error function. Another reason to use such an error function is to concentrate on the natural dividing line in the mathematics; when there is no noise (as in exhaustive learning), the mathematics for questions from within the training set is trivial, whereas the mathematics for questions from outside of it is not. Indeed, it is only such off-training set questions which can distinguish amongst a set of candidate generalizers when noise-free learning is assumed.

This paper proves that with an off-training set error function the improvement in generalization with increasing training set size characteristic of exhaustive learning goes away. Indeed, for the limit of large errors, when the output space is binary the generalization gets *worse* exponentially with m. Perhaps surprising, this is true regardless of the size of the input space. This is in accord with the discussion above of the scenario where the exhaustive learning generalizer is equivalent to a coin flip for questions outside of the training set.

10) Regardless of what error function one uses, usually in formalisms like exhaustive learning one is ultimately interested in examining "learning curves" which show how generalization will change, on average, as one adds new elements to one's *already extant* training set. In general, this quantity of interest need *not* be given by the quantity calculated in exhaustive learning, $P(E \mid f, m)$, which simply averages over all training sets without regard to how they could arise from smaller ones. This paper finds a sufficient condition for when this quantity of interest is given by $P(E \mid f, m)$.

11) Exploiting the symmetry of the extended Bayesian formalism of this paper under $f \leftrightarrow h$, this paper derives formulas for $P(E \mid \text{hypothesis function } h, m)$ directly from the exhaustive learning formulas for $P(E \mid \text{target function } f, m)$.

Section I of this paper introduces the extended Bayesian formalism used in the rest of the paper. It is illustrated by giving a formal definition of the exhaustive learning scenario. Section II of this paper then reproduces and extends the conventional analysis of exhaustive learning for the case which it addresses, where the testing set can run over the training set. Section III is an analysis of exhaustive learning when an off-training-set error function is used. This section proves that the characteristic $(1 - E)^m$ behavior of exhaustive learning disappears with such an error function, even in the limit of an infinite input space. Finally, using the linearity of probability theory, section IV formally justifies interpreting exhaustive learning in terms of "learning curves". Section IV also exploits some symmetries in the extended Bayesian formalism to extend exhaustive learning's results to other scenarios. In particular, it uses those symmetries to extend the results of exhaustive learning from the distribution P(error = E | target function f, training set size m) to the distribution P(error = E | guessed hypothesis function h, training set size m). This extension of exhaustive learning is the only result in this paper which draws substantially on the full power of the extended Bayesian formalism

I. THE EXHAUSTIVE LEARNING SCENARIO

All supervised machine learning issues can be cast in terms of the extended Bayesian formalism presented in [Wolpert 1992a]. This section presents a simplified version of that formalism, illustrating it by giving a rigorous formulation of the exhaustive learning scenario. The reader interested in a more detailed exposition of the extended Bayesian formalism is directed to [Wolpert 1992a].

Although as conventionally formulated it appears that exhaustive learning can have an infinite input space, in fact its input space, here denoted by X, is finite and discrete, consisting of n elements.[2] The output space Y is also finite and discrete, and consists of r elements. Let F and H both be the set of all functions from X to Y. The elements of F, generically denoted by f, are to be viewed as target functions, and the elements of H, generically denoted by h, are to be viewed as hypothesis functions. (In this paper, such functions from X to Y will often be viewed as sets, of n input-output pairs.) In addition to target functions and hypothesis functions, we also have training sets of input/output pairs, $\{x_i \in X, y_i \in Y\}$. A generic training set will be indicated by L, and consists of m input/output pairs. The set of all input values x_i in the pairs making up L will be indicated by L_X. The corresponding output values will be indicated by L_Y.

For generality, training sets are assumed ordered; symmetry under permutation of the pairs making up a training set can be imposed afterwards, if desired. Whether one uses ordered or unordered training sets makes a difference whenever (as in exhaustive learning) one sums over training sets. For example, the unordered set {1, 2, 2} corresponds to 3 ordered sets ({1, 2, 2}, {2, 1, 2}, and {2, 2, 1}). However the unordered set {1, 2, 3} corresponds to 6 ordered sets. This illustrates that one can not transform from sums over one kind of set to sums over the other kind by multiplying by a constant, or some such. (Although never explicitly stated, in [Schwartz et al. 1990] sums over training sets are apparently sums over ordered training sets.)

The majority of supervised machine learning scenarios implicitly concern an event space consisting of the set of all triples $\{f \in F, h \in H, L\}$; all probabilities involved in such scenarios derive from the full joint probability P(f, h, L). (In particular, note that even if we wish to restrict the set of allowed hypothesis functions to a “concept class”, we are still implicitly using the full event space $\{f \in F, h \in H, L)$; use of a “concept class” simply means that we have set to 0 all probabilities P(f, h, L) for which h is not in that concept class.) In particular, exhaustive learning uses this event space, which means that *any* formal probability-based analysis of exhaustive learning must, implicitly or otherwise, be couched in terms of probabilities over this event space.

In denoting probabilities in exhaustive learning it will sometimes be notationally convenient to substitute the quadruple $\{f, h, L_X, L_Y)$ for the equivalent triple $\{f, h, L = \{L_X, L_Y\}\}$; the reader is warned that we will interchange one notation for the other without any warning.[3]

It is instructive to note that conventional “Bayesian” analysis is not the proper way to evaluate exhaustive learning, because the event space implicitly used in such analysis is a proper subset of the full {f, h, L} space. Conventional “Bayesians” do not distinguish hypothesis functions from target functions, and implicitly use an event space {f, L}.[4] In fact, what a Bayesian refers to as a “hypothesis” is what is referred to in this paper (and in the PAC literature) as a target function; there is nothing in conventional Bayesian analysis which is analogous to what is here being called hypothesis functions.[5]

Any generalizer one might use in supervised machine learning is a rule which assigns conditional probabilities to hypothesis functions, given a particular training set. In other words, it is the conditional distribution $P(h \mid L)$. For example, if the generalizer is deterministic, it always outputs the same hypothesis function when taught with the same training set, and for fixed L, $P(h \mid L)$ is a delta function over H. If instead the generalizer is stochastic, then for a given L $P(h \mid L)$ has support extending over more than one $h \in H$.

In exhaustive learning, one uses the following stochastic, "noise-free" generalizer: Start with a nowhere-negative distribution $T(h \in H)$. Given L, throw away all $h \in H$ which do not go through the elements of L (i.e., throw away all h except those h such that the relation $L \subset h$ holds). Then up to an overall normalization factor, the probabilities of the generalizer picking one of the remaining h is given by T(h). (The probability of the generalizer picking one of the "thrown out" h is 0.) In other words, the exhaustive learning generalizer assumes no noise was used in creating L and then guesses randomly, according to the distribution T(h).

More formally, in exhaustive learning

$P(h \mid L) \propto T(h) \times \delta(L \subset h)$,

where the delta function equals 1 if the set relation is true, 0 otherwise. The proportionality constant is set by normalization and depends on L. It is written as $k(L_X, L_Y) \equiv [\Sigma_h T(h) \times \delta(L \subset h)]^{-1}$, and is implicitly assumed to be non-zero. Below it will be useful to introduce the count function $S(g_1, g_2, \{x_i\}) \equiv \Sigma_i \{\delta(g_1(x_i), g_2(x_i)\}$ which measures the number of times function g_1 agrees with function g_2 over the input values in $\{x_i\}$. Using this function, with m' being the number of distinct elements in L ($m' \leq m$), for exhaustive learning $P(h \mid L) = k(L_X, L_Y) \times T(h) \times \delta(S(L, h, L_X), m')$ with $[k(L_X, L_Y)]^{-1} = \Sigma_h T(h) \times \delta(S(L, h, L_X), m'))$.

There are no a priori restrictions on T(h) in exhaustive learning. In particular, one is free to choose T(h) to equal zero for many h's, so that the generalizer has limited expressive power. Of course, one should never set T(h) = 0 for h = f, since then the generalizer would be undefined if L equalled the set $\{x \in X, f(x)\}$. (Because of this, there are some practical difficulties associated with using a T(h) which is not nowhere-zero; see the discussion below on the advantages of exhaustive learning as compared with conventional Bayesian analysis.)

The exhaustive learning generalizer is sometimes referred to as a "Gibbs learning algorithm" [Haussler et al. 1991]. It should be noted that although T(h) intuitively seems like "the prior beliefs of the learner about which target [function] it will be learning" ([Haussler et al. 1991]), this is in fact wrong. In general T(h) does **not** equal P(h), the prior probability of h. This is because $P(h) = \Sigma_L P(h \mid L) \times P(L) = T(h) \times \Sigma_L [k(L_X, L_Y) \times \delta(L \subset h) \times P(L)]$ and in general the sum depends on h.[6] The term "prior" is also misleading here because as used in conventional Bayesian analysis it refers to P(f); conventional Bayesian analysis does not concern itself with h at all.

Note that in all supervised machine learning, exhaustive learning included, it is implicitly assumed that the generalizer only has access to the training set, and can not *directly* observe anything else about the target function. (The generalizer might incorporate *assumptions* about the target function, but those assumptions need not be true; such assumptions are reflected in P(h) rather than P(f).) In other words, if one surreptitiously changes the target function, but leaves the training set unchanged, then the guessing behavior of the generalizer is unchanged. This means that $P(h \mid f, L)$ must be independent of f, which in turn means that $P(h \mid f, L) = P(h \mid L)$ (which in turn means that $P(f \mid h, L) = P(f \mid L)$ and $P(h, f \mid L) = P(h \mid L) \times P(f \mid L)$; see [Wolpert 1992a].) Like the assumption concerning $P(L_X \mid f)$ (see below), it turns out that this assumption concerning $P(h \mid f, L)$ is used in almost all machine learning research, though it is never explicitly acknowledged.

In addition to fixing the generalizer, exhaustive learning also fixes the *sampling assumption* (sometimes called the "likelihood function"), $P(L \mid f)$. This is the rule for how training sets are made from target functions, and in general reflects three things: how m, the size of training set, is chosen, how the elements of L_X (the points at which the target function is to be sampled) are selected, and finally, what kind of noise accompanies the sampling of the target function at the elements of L_X.

In exhaustive learning, these three characteristics of the sampling assumption are set as follows: First, it is assumed that training sets are constructed without any noise, which is formalized by setting $P(L \mid f) = 0$ unless L agrees with f, i.e., unless $L \subset f$. (Note that this requirement is formally equivalent to setting to 0 all $P(f, h, L)$ for which $L \not\subset f$.) In addition, the elements of L_X are selected in an i.i.d. (independent identically distributed) manner:

$$\frac{P(L \mid f)}{P(L' \mid f)} = \frac{\Pi_{i=1}^{m} \quad \pi(L_X(i))}{\Pi_{i=1}^{m} \quad \pi(L'_X(i))}$$ if both L and L' are consistent with f and if they have the same cardinality, m. In this formula, $L_X(i)$ is the input component of the i'th element of L, and similarly for $L'_X(i)$. $\pi(x \in X)$ is a (usually unknown) sampling distribution over X.[7] Finally, in exhaustive learning only conditional probabilities with m fixed are calculated, so the aspect of $P(L \mid f)$ concerning training set size is irrelevant.

Note that in exhaustive learning repeats are allowed in L, i.e., the same input value can occur more than once in L_X. There are two immediate ramifications of this. First, this means that conditions like "$L \subset f$" are being interpreted a bit loosely in this paper; formally, they should be taken to mean that L is consistent with f, rather than that the set of input-output pairs $\{L_X(i), L_Y(i); 1 \leq i \leq m\} \subset$ the set of input-output pairs $\{x, f(x); x \in X\}$. Second, note that due to our noise-free sampling assumption, there can be no conflicts in L: anytime an input value is repeated, the corresponding output values must agree. This means that any such repeat in L conveys no extra information concerning f to the generalizer. This points up an odd aspect of exhaustive learning: As already mentioned, in exhaustive learning probabilities conditioned on m will be calculated. However it is not m but rather m', the number of distinct pairs in L, which is directly related to "the amount of information" in L.

We have now defined "the exhaustive learning scenario". To analyze this scenario, in addition to probabilities over $\{f \in F, h \in H, L\}$, one must relate f and h to a real-world "cost". This is usually done through an error function, which is a function over the event space, Er(f, h, L). The conventional analysis of exhaustive learning implicitly uses an error function independent of L but dependent on $\pi(x)$: $Er(f, h) = \Sigma_{x \in X} \pi(x) [1 - \delta(f(x), h(x))]$. As is discussed later, although in many circumstances this error function entails "distribution-free" results (i.e., results independent of $\pi(x)$), this error function also has a number of undesirable features. (This paper will not restrict itself to this error function.)

The conventional analysis of exhaustive learning is concerned with evaluating the average error when the target function is held fixed. In other words, it is concerned with the quantity $\langle P(\text{error} = E \mid \text{target function} = f, \text{training set} = L)\rangle$, the expectation value being over all possible training sets L of size m chosen from the (fixed but unknown) target function f. That the target function is assumed fixed can be seen by noting that exhaustive learning is concerned with the "learning curves" of how average error changes as more samples of the target function are added to L. Note that because X and Y are both finite, the set of all possible E values is also finite; the expectation value must equal 0 for all but a finite number of E values.

It's worth pointing out that the foregoing definition of exhaustive learning is almost identical to that of PAC. In particular, PAC uses the same sampling assumption and error function as exhaustive learning. It also has the same distribution of interest, $\langle P(E \mid f, L)\rangle$. Where PAC differs from exhaustive learning is in its generalizer (although it should be noted that both formalisms, in their "vanilla" versions, use noise-free generalizers). For PAC's generalizer, one can not deduce $\langle P(E \mid f, L)\rangle$) directly, as one can in exhaustive learning (See below.) Instead one must settle for bounds on it. (E.g., upper bounds over all generalizers in a particular class of the quantity $\delta \equiv \Sigma^{1}_{E=\varepsilon} P(E \mid f, m)$.) PAC then goes on to concern itself with issues of polynomial vs. exponential relationships governing those bounds, something which exhaustive learning does not do. Even so, the results of PAC and exhaustive learning betray the close relationship of their foundations. In particular, whereas the central result of exhaustive learning is $P(E \mid f, m) \sim (1 - E)^m$, the starting point of PAC is the very similar $\delta \leq (1 - \varepsilon)^m \times |H'|$.

It will be useful to rewrite $\langle P(E \mid f\ L)\rangle$ as follows:

(1) $\langle P(E \mid f, L)\rangle = P(\text{error} = E \mid \text{target function} = f, \text{training set has size } m)$

Proof: Writing it out, $<P(E \mid f, L)> \equiv \Sigma_{L \subset f;m} \{P(E \mid f, L) \times P(L \mid f, m)$ where ";m" means of size m. (Note that the "$\subset$ f" conditions are actually superfluous; since there is no noise, they are automatically enforced by the probabilities in the summands. Nonetheless, they will be kept for clarity.). Now $P(L \mid f, m) = P(L, f, m) / P(f, m) = P(L, f) / P(f, m)$. Therefore $<P(E \mid f, L)> = \Sigma_{L \subset f;m} \{P(E, f, L) / P(f, m)\} = P(E, f, m) / P(f, m) = P(\text{error} = E \mid \text{target function} = f, \text{training set has size } m)$. QED.

It's worth making a brief aside on the implications for the relative "merit" of exhaustive learning which derive from its concentrating on $P(E \mid f, m)$. The first thing to note is that the axioms of probability theory tell us that if one wishes to *use* it in a real-world situation, one's conditional probability should fix only that which one knows. In practice, one almost always knows L in addition to m, whereas one never knows f. This means that exhaustive learning is not concentrating on the "correct" distribution.

Things are a bit more subtle than this however. First note that because f is fixed (though unknown) in $P(E \mid f, m)$, no probabilities concerning f arise in the calculations of exhaustive learning. P(f) will not arise in any of the calculations in this paper. Moreover, so long as T(h) is nowhere-zero, the exhaustive learning generalizer is well-defined for any L, and it can be used with impunity in the real world, without any regard for f or the L which are sampled from f.[8] On the other hand, conventional Bayesian analysis concentrates on $P(E \mid L)$, the "correct" distribution. Since f is not fixed in this distribution, Bayesian analysis is led to concern itself directly with P(f), and must make assumptions for P(f) - assumptions exhaustive learning need not make. This illustrates one of the strengths of exhaustive learning: in contrast to exhaustive learning, both the conventional "Bayesian" results recommending certain generalizers and the PAC results recommending certain other generalizers rely on (perhaps incorrect) assumptions concerning f.

Although conventionally couched in terms of neural nets, volumes in weight space, etc., in point of fact the underlying concepts of exhaustive learning have nothing to do with neural nets.

The exposition given above defines exhaustive learning exactly, and in a completely general manner. The precise scenario considered by the conventional analysis of exhaustive learning is simply a special case, where both h and f are explicitly parameterized neural nets. Of course, everything discussed in this paper applies just as well to such a special case.

It is important to note that use of the extended Bayesian formalism does more than simply extend the analysis to the case where the functions of interest are not explicitly parameterized. As is illustrated in the following sections, it is only by couching exhaustive learning in terms of the extended Bayesian formalism that one can address all the issues discussed in this paper. In particular, one could not address all those issues using PAC, conventional Bayesian analysis, or the "statistical mechanical formalism"; those are restricted cases of our formalism, and are therefore restricted in their range of applicability.

Another advantage of using the extended Bayesian formalism is that it reduces supervised learning analysis to a two-stage process: a thinking stage, and a calculating stage. In the first stage, one fixes the error function, and perhaps some conditional probabilities (e.g. the generalizer). To complete this stage, one then chooses some other conditional probability distribution(s) of interest, as the object(s) one wishes to calculate. All "thought" has been used in this first stage of the process. In the second stage, all one does is calculate the "conditional probability distributions of interest". No subtleties concerning interpretation issues arise in this second stage.

In this paper, we will explicitly carry out this two-stage process. The first stage was carried out in this section. The rest of this paper is a completion of the second stage.

II. CALCULATION OF GENERALIZATION ERROR WHEN THE TESTING SET CAN INCLUDE THE TRAINING SET

We now have the formal tools needed to perform a rigorous analysis of exhaustive learning. This section shows how to use those tools to replicate the conventional analysis of exhaustive learning [Schwartz et al. 1990] without assuming that one is using a neural net generalizer. This section also begins the exposition of how to use those tools to go beyond the conventional analysis.[9]

The following discussion will make reference to the version of exhaustive learning presented in [Hertz et al. 1991]. For those readers interested in translating between [Hertz et al. 1991] and the exposition presented here, note that what in [Hertz et al. 1991] are written as functions "f" are actually hypothesis functions, written here as "h". Loosely speaking, what is written there as "$R_0(f)$" is written here as T(h), and in general what are treated as "volumes in weight space" in [Hertz et al. 1991] are here generalized to probabilities of hypothesis functions. Furthermore, in [Hertz et al. 1991] "p" is the size of the training set, which is here written as "m". The fixed (though unknown to the experimenter) target function, referred to as $\bar{f}$ in [Hertz et al. 1991], is here written as f. Finally, rather than the "generalization accuracy g(.)" used in [Hertz et al. 1991], as was mentioned in the previous section we here use the "exhaustive learning error function" Er(f, h), which is linearly related to g(.).

Section I completed the first stage of machine learning analysis. As was mentioned above, from here on out there is no need to think. One just turns the calculational crank; evaluate P(E | f, m), given the Er(f, h), P(h | L) and P(L | f) of the previous section. Doing this we will derive the results of the conventional analysis of exhaustive learning.

Expanding in terms of the triples making up our event space, P(error = E | target function = f, training set has size m) $\equiv$ P(E | f, m) = $\{\Sigma_h \Sigma_{L \subset f;m} P(f, h, L)\, \delta(E, Er(f, h))\} / \{P(f, m)\}$ (where this delta is the Kronecker delta function, and as above ";m" means "of size m".). This can be rewritten as

(2) $$P(E \mid f, m) = \Sigma_h P(h \mid f, m)\, \delta(E, Er(f, h)).$$

The analogous formula in Schwartz et al.'s analysis is "$\rho_p(g) = \Sigma_f P_p(f)\, \delta(g - g(f))$" from equation

(6.67) in [Hertz et al. 1991]. (N.b. in this paper we use straight probabilities, implicitly viewing errors as taking on only one of a finite set of possible values, whereas Schwartz et al. use probability densities, i.e., implicitly view errors as taking on continuous values, and therefore require Dirac delta functions.) So in particular, the term "$P_p(f)$" from [Hertz et al. 1991] means $P(h \mid f, m)$ in the notation of this paper.

To proceed further we need to evaluate $P(h \mid f, m)$. Doing this will take us back to equations (6.64) through (6.66) in [Hertz et al. 1991]. First, note the following lemma, which holds because training sets are ordered with repeats allowed:

(3) $\Sigma_{L_X;m} \Pi_{i=1}^{m} \; [g_i(L_X(i))] \; = \; \Pi_{i=1}^{m} \; [\Sigma_{x \in X} \, g_i(x)]$, for any function g(.).

Proof: We can prove the lemma using induction over m. To base the induction, note that the formula obviously holds when m = 1: $\Sigma_{L_X;1} \; \Pi_{i=1}^{1} \; [g_i(L_X(i))] \; = \; \Sigma_{L_X(1) \in X} \; [g_1(L_X(1))] \; = \; \Sigma_{x \in X} \, [g_1(x)]$. Now we assume that the formula holds for m = c, and check to see that it holds for m = c + 1. To do this, write $\Sigma_{L_X;c+1} \; \Pi_{i=1}^{c+1} [g_i(L_X(i))] \; = \; \Sigma_{x \in X} \; \Sigma_{L_X;c} \; [g_{c+1}(x) \times \Pi_{i=1}^{c} \; [g_i(L_X(i))]] = \Sigma_{x \in X} \{g_{c+1}(x) \times \Sigma_{L_X;c} \Pi_{i=1}^{c} \; [g_i(L_X(i))]\}$, which equals $\Sigma_{x \in X} \{g_{c+1}(x) \times \Pi_{i=1}^{c} \; [\Sigma_{x' \in X} \; g_i(x')]\}$, by the inductive hypothesis. This last expression can be rewritten as $[\Sigma_{x \in X} \; g_{c+1}(x)] \times \Pi_{i=1}^{c} \; [\Sigma_{x' \in X} \; g_i(x')] \; = \; \Pi_{i=1}^{c+1} \quad [\Sigma_{x \in X} \; g_i(x)]$, which completes the proof. QED.

As was mentioned previously, it will sometimes be convenient to substitute the space of quadruples $\{f, h, L_X, L_Y)$ for the equivalent space of triples $\{f, h, L = \{L_X, L_Y\}\}$. Along with exploiting (3), it is necessary to make this kind of substitution to evaluate $P(h \mid f, m)$:

(4) $$P(h \mid f, m) = \Sigma_{L \subset f;m} P(h \mid L) \times \Pi_{i=1}^{m} [\pi(L_X(i))].$$

Proof: $P(h \mid f, m) = \{\Sigma_{L \subset f;m} P(f, h, L)\} / \{\Sigma_{h, L' \subset f;m} P(f, h, L')\}$

$= \{\Sigma_{L \subset f;m} P(h \mid L) \times P(f, L)\} / \{\Sigma_{L' \subset f;m} P(f, L')\}$

(since $P(h \mid f, L) = P(h \mid L)$)

$= \{\Sigma_{L \subset f;m} P(h \mid L) \times P(L \mid f)\} / \{\Sigma_{L' \subset f;m} P(L' \mid f)\}$

$= \{\Sigma_{L \subset f;m} P(h \mid L) \times \Pi_{i=1}^{m} [\pi(L_X(i))]\} / \{\Sigma_{L' \subset f;m} \Pi_{i=1}^{m} \pi(L'_X(i))]\}$

(due to the exhaustive learning sampling assumption).

Now rewrite $\Sigma_{L \subset f;m} \Pi_{i=1}^{m} [\pi(L_X(i))] = \Sigma_{L_X;m} \Pi_{i=1}^{m} [\pi(L_X(i))] = [\Sigma_{x \in X} \pi(x)]^m = [1]^m = 1$.

(In writing down the second equality, use has been made of the fact that $\Sigma_{L_X;m} \Pi_{i=1}^{m} [g(L_X(i))]$ $= [\Sigma_{x \in X} g(x)]^m$ for any function g(.), a fact which is a special case of the more general formula (3).) QED.

This equality shows how $P(h \mid f, m)$, which in [Hertz et al. 1991] is called "$P_p(f)$", can be an expectation value over training sets, as it is presented in equation (6.66) of [Hertz et al. 1991]: simply re-write $\Sigma_{L \subset f;m} P(h \mid L) \times \Pi_{i=1}^{m} [\pi(L_X(i))]$ as $< P(h \mid L) >$ (up to an overall proportionality constant), the expectation value being over all possible training sets of size m chosen from the target function.

To complete the evaluation of $P(h \mid f, m)$, we must plug the exhaustive learning $P(h \mid L)$ into equation (4). After doing this we will plug the resultant expression for $P(h \mid f, m)$ into equation (2).

(5) $P(h \mid f, m) = T(h) \times \Sigma_{L_X;m} \{k(L_X, f(L_X)) \times \Pi_{i=1}^{m} [\delta(h(L_X(i)), f(L_X(i))) \times \pi(L_X(i))]\}$.

Proof: $P(h \mid f, m) = \Sigma_{L \subset f;m} \{P(h \mid L) \times \Pi_{i=1}^{m} [\pi(L_X(i))]\}$

$= \Sigma_{L_X;m} \{P(h \mid L_X, L_Y = f(L_X)) \times \Pi_{i=1}^{m} [\pi(L_X(i))]\}$

$= \Sigma_{L_X;m} \{k(L_X, f(L_X)) \times T(h) \times \delta(S(f, h, L_X), m') \times \Pi_{i=1}^{m} [\pi(L_X(i))]\}$

(due to our assumption concerning $P(h \mid L)$; recall that m' is the number of distinct elements in L, and $[k(L_X, f(L_X))]$ is the normalization constant for $P(h \mid L)$.)

$= \Sigma_{L_X;m} \{k(L_X, f(L_X)) \times T(h) \times \Pi_{i=1}^{m} [\delta(h(L_X(i)), f(L_X(i)))] \times \Pi_{i=1}^{m} [\pi(L_X(i))]\}$

$= \Sigma_{L_X;m} \{k(L_X, f(L_X)) \times T(h) \times \Pi_{i=1}^{m} [\delta(h(L_X(i)), f(L_X(i))) \times \pi(L_X(i))]\}$. QED

Equation (5) exhibits “the contraction ... (emphasizing) regions of configuration space with intrinsically high generalization ability” (page 379 of [Schwartz et al. 1990]). As m grows, (5) shows that h must agree with f for an increasing number of inputs if it is to have non-zero probability.

It is after deriving equation (5) that the conventional analysis of exhaustive learning makes an approximation; it assumes that the $k(L_X, f(L_X))$ term in (5) can be absorbed into an overall proportionality constant, with negligible effect on $P(E \mid f\ m)$. In other words, it assumes that $k(L_X, f(L_X))$ is independent of L_X for all L_X which contribute substantially to $P(h \mid f\ m)$, for all h which contribute substantially to $P(E \mid f\ m)$ (see equation (2)). Schwartz et al. call this approximation “self-averaging” and claim that one would expect it to hold whenever n >> m. Self-averaging is similar to the “annealed approximation” [Seung et al. 1991] which, loosely speaking, consists of replacing k(., .) with the L_X-average of k(., .).

Given self-averaging and (3), $P(h \mid f, m) \propto T(h) \times [\ \Sigma_{x \in X} \{\delta(h(x), f(x)) \times \pi(x)]\}\ ]^m =$

$T(h) \times [1 - Er(f, h)]^m$. Substituting this result into equation (2), collecting all constants into an overall normalization constant, and defining $\rho_0(E) \equiv \Sigma_h \{T(h) \times \delta(E, Er(f, h))\}$ (in loose agreement with the terminology used in [Hertz et al. 1991]), one derives the central result of the conventional analysis of exhaustive learning: $P(E \mid f, m) \propto [1 - E]^m \times \rho_0(E)$ (equation (6.68) in [Hertz et al. 1991]). In other words, $P(E \mid f, m) = \{[1 - E]^m \times \rho_0(E)\} / \{\Sigma_{E'}[1 - E']^m \times \rho_0(E')\}$. (Note that for every value of $E' \in [0, 1]$ the summand of the expression $\{\Sigma_{E'} [1 - E']^m \times \rho_0(E')\}$ decreases as m is raised, and therefore the entire sum is a decreasing function of m.)

Note that $\rho_0(E)$ will equal 0 for all but a finite number of E values. This is why the normalization constant giving $P(E \mid f, m)$ is a sum rather than an integral. Note also that in general the set of allowed E values is not uniformly distributed across [0, 1]. For example, there will always be an upper bound on E which is less than 1 so long as no $\pi(x)$ value equals 0. (This point is returned to in section III.) As another example, when $r = 2$ and $\pi(x) = 1 / n$ for all x, there are n h's which go into the sum for $\rho_0(E = (n - 1) / n)$ (i.e., there are n h's having this error value). However there are $C^n_{n/2}$ h's which go into the sum for $\rho_0(E = 1 / 2)$. Unless T(h) is highly skewed, this phenomenon will induce a large hump in $\rho_0(E)$ favoring E values near 1 / 2.

One can calculate a number of interesting quantities from this "central result". For example, one could calculate the average error with f and m fixed, $\Sigma_E (E \times P(E \mid f, m))$. (Generically, this goes like 1/m or like $e^{-\alpha m}$ - see [Hertz et al. 1991].) Or one could calculate the probability of error E when f is unspecified (which calculation allows one to find the average error when f is unspecified): $P(E \mid m) = \Sigma_{f \in F} (P(E \mid f, m) \times P(f \mid m)) = \Sigma_{f \in F,L;m} (P(E \mid f, m) \times P(f \mid L) \times P(L)) / \Sigma_{L;m} P(L))$.

Equation (6.68) of [Hertz et al. 1991], the "central result" of the conventional analysis of exhaustive learning, is taken to imply that as m increases, lower E's become more likely. (Though note that if m increases too much, then it's no longer true that n >> m, and one shouldn't expect the central result to apply.) In other words, it is taken as an argument that "inductive inference works”, if one uses the exhaustive learning generalizer (or as it’s put in [Schwartz et al. 1990], there is a “monotonic increase of the average generalization ability with increasing m”).

One can proceed further by using the extended Bayesian formalism. In particular, using this formalism one can directly test whether or not self-averaging holds (in the limit of n >> m or otherwise) for various T(h) and $\pi(x)$. For example, one can prove the following:

(6) Let ε and σ be fixed, where σ is an upper bound on the ratio $[\max_x \pi(x)] / [\min_x \pi(x)]$, and $\varepsilon < 1$ is an upper bound on E. Then one special case where self-averaging does hold in the limit of n >> m is when T(h) is independent of h.

The proof of (6) is in appendix B. Note that as n increases, the space X changes, and therefore $\pi(x)$ becomes a distribution over a different space. The σ requirement in (6) limits how much $\pi(x)$ can change in this process, as n gets large. The requirement on ε serves a similar function, for E.

It is important to realize that in (6) (and everywhere else in this paper) the term “constant T(h)” need not mean $T(h_1) = T(h_2)$ for all h_1, h_2. Rather it suffices to have $T(h_1) = T(h_2)$ for all h_1, h_2 such that both h_1 and h_2 are consistent with at least one L chosen from f. In other words, T(h) can have non-uniformities and (6) still holds. Note also that the “constant T(h)” case referred to in (6) is *not* the same as the constant T(**w**) case, where functions h are parameterized as perceptrons with weight vector **w**, or some such. (This paper is not concerned with quaint but useless parameterization schemes.)

This constant T(h) case can be viewed as the “MDL”, or “MaxEnt”, most parsimonious exhaustive learning generalizer. (See [Wolpert 1992a] for an argument that the mere possibility of a “MaxEnt” P(f) serves as a proof that there exists no first-principles proof of inductive inference.) However it isn’t necessary to use the central result of the conventional analysis of exhaustive learning in this T(h)-independent-of-h case; by using the extended Bayesian formalism, we can solve for $P(h \mid f, m)$ (and therefore for $P(E \mid f, m)$) for this case *exactly*, regardless of the relative sizes of n and m. Interestingly, the result explicitly depends on r, the size of the output space.

It is simplest to state the result when $\pi(x)$ is uniform. The proof is in appendix C:

(7) When T(h) is independent of h and $\pi(x)$ is uniform, for those E values which can occur

$$P(E \mid f, m) \propto \{\Sigma_{i=1}^{\min(m,\, n(1 - E))} [r^{(i-n)} \times \varphi(m, i) \times C_i^{n(1-E)}]\} \times C_{n(1-E)}^{n} \times (r - 1)^{nE},$$

where $\varphi(., .)$ is a combinatoric function defined in appendix C.

In addition to allowing exact calculations when n is not >> m, the extended Bayesian formalism also allows the straight-forward evaluation of $P(h \mid f, m)$ (and therefore of $P(E \mid f, m)$) in a number of situations where $k(L_X, f(L_X))$ is not independent of L_X, so that self-averaging does not apply. In appendix D it is shown that what is perhaps the simplest such case explicitly results in a different result than the self-averaging-based "central result" of the conventional analysis of exhaustive learning. Therefore that central result is not generic. In other words, contrary to the claims of [Schwartz et al. 1990], self-averaging does **not** hold whenever n >> m, and in particular it does not hold for what is perhaps the simplest non-trivial case. This means that even if one accepts that the behavior of $P(E \mid f, m)$ is a direct reflection of "whether or not inductive inference works", the conventional analysis of exhaustive learning does **not** prove that inductive inference works with the exhaustive learning generalizer.

One peculiar aspect of exhaustive learning is that the probability it's interested in is $P(E \mid f, m)$ rather than $P(E \mid f, L)$. What's odd about this is that what goes on the right hand side of a conditional probability should be the set of *everything* that's fixed for the problem. This set is a superset of the set of all that's known for the problem. So by evaluating $P(E \mid f, m)$ rather than $P(E \mid f, L)$, the conventional analysis of exhaustive learning is assuming that we know the size of the training set but not the actual elements making it up. Another drawback to concentrating on $P(E \mid f, m)$ rather than $P(E \mid f, L)$ is that it is always possible to calculate $P(E \mid f, m)$ from $P(E \mid f, L)$ (just use (1)), but the reverse does not hold.

By using the extended Bayesian formalism however, we can evaluate $P(E \mid f, L)$. For convenience, assume that T(h) is uniform across h, and that $\pi(x)$ is also uniform and equals 1 / n, n being the cardinality of X. (As was mentioned previously, in such a case (6.68) of [Hertz et al. 1991] holds exactly when n >> m.) Under these conditions, assuming that the error E can occur,

$$(8)\ P(E \mid f, L) = C_{z}^{(n\text{-}m')} \times (r - 1)^{(n\text{-}m'\text{-}z)} \ / \ r^{(n\text{-}m')},$$

where $z \equiv [n - m' - nE]$, and is the number of agreements between f and any hypothesis function having error E, for questions outside of the training set.

Proof: $P(E \mid f, L) = \Sigma_{h} P(h \mid f, L)\ \delta(Er(f, h), E)$

$= \Sigma_{h} P(h \mid L)\ \delta(Er(f, h), E)$

$= \{\Sigma_{h} T(h)\ \delta(Er(f, h), E)\ \delta(S(L, h, L_X), m')\} \ / \ \{\Sigma_{h} T(h)\ \delta(S(L, h, L_X), m')\}$

$= \{\Sigma_{h \supset L} T(h)\ \delta(Er(f, h), E)\} \ / \ \{\Sigma_{h \supset L} T(h)\}.$

Under the condition $h \supset L$, the uniformity of $\pi(x)$ allows us to rewrite the error function Er(f, h) as $[n - m' - S(f, h, X - L_X)] / n$. We can then use the uniformity of T(h) to cancel it out from the numerator and the denominator: $P(E \mid f, L) = \{\Sigma_{h \supset L} \delta(S(f, h, X - L_X), n(1 - E) - m')\} \ / \ \{\Sigma_{h \supset L} 1\}$. Note that the value of P(E | f, L) depends only on m', the number of distinct elements in the training set, and not on the particular training set chosen. In other words, for this simple scenario P(E | f, L) is independent of L, depending only on the number of distinct elements in L. So this expression we are calculating gives the probability of error E, no matter what X values are chosen to sample f. As such, it is extremely similar to P(E | f, m). However the expression derived here is simply {the number of ways to pick a Y-valued function of a variable (that variable taking (n - m') possible values) such that the function equals some pre-fixed values on some n - m' - nE of the values of the variable and nowhere else} divided by {the number of ways to pick a Y-valued function of a variable, that variable taking on (n - m') possible values}. Carrying through the combinatorics gives the result in (8). QED.

Note that z must be a positive integer for equation (8) to be meaningful. This is a direct reflection of the fact that only certain E values can occur.

Equation (8) tells us that the form of $P(E \mid f, L)$ can be vastly different from the form of $P(E \mid f, m)$, even when (as here) $P(E \mid f, L)$, like $P(E \mid f, m)$, does not depend on the actual pairs making up L. Part of this difference between the two conditional probabilities is a reflection of the fact that $P(E \mid f, m)$ is a function of m, whereas $P(E \mid f, L)$ is independent of m and depends only on m'.

This distinction between $P(E \mid f, L)$ and $P(E \mid f, m)$ notwithstanding, we would expect the two expressions to exhibit the same kind of m dependence in the limit of $n >> m$ where we can take $m = m'$ (see the proof of (6)). This is indeed the case:

(9) When $n >> m$ and both $T(h)$ and $\pi(x)$ are constant, $P(E \mid f, L) \propto (1 - E)^m \times \rho_0(E)$.

Proof: First rewrite (8) as $P(E \mid f, L) = C^{n-m'}_{n-m'-nE} \times (r - 1)^{nE} / r^{n-m'}$. This means that the ratio $P(E \mid h, L') / P(E \mid h, L)$, where L is a training set of m' distinct elements and L' is a training set of m' + 1 distinct elements, equals $[r \times C^{(n-m'-1)}_{z'}] / C^{(n-m')}_{z}$, z' being the z value for the training set L', which has m' + 1 distinct elements; $z' = z - 1$. (Note that it is being assumed that for both L and L' it's possible to have error E, i.e., it's being assumed that both $E < (n - m') / n$ and $E < (n - m' - 1) / n$. If this assumption isn't made, then z and/or z' might be negative.) The ratio can be rewritten as $r \times [1 - nE / (n - m')]$. If we now assume that $n >> m \geq m'$, we can approximate this ratio as $r(1 - E)$. So in this approximation, $P(E \mid f, L) = r^{m'} \times [1 - E]^{m'} \times \omega(E)$, for some function $\omega(.)$. Now for $n >> m$, the probability that L has a repeat in it is vanishingly small. Therefore we can replace m' with m; $P(E \mid f, L) \propto [1 - E]^m \times \omega(E)$. Now using (1) and the fact that $\pi(x)$ is being assumed constant, for $m = m' = 1$, $P(E \mid f, m)$ must equal $P(E \mid f, L)$ for any L of size m. Therefore

$\omega(E) = \rho_0(E)$. QED.

The interesting implication is that the increase in generalization accuracy with m embodied in equation (6.68) of [Hertz et al. 1991] arises from the same causes behind the increase in generalization accuracy in (8): To get an error value E one must have f and h disagree for a certain number of the off-training-set X values (they will always agree on all of L_X). The probability of that error value E, loosely speaking, is determined by how many distinct h there are which disagree with f for that “certain number of the off-training-set X values”. Increasing the number of distinct elements in the training set decreases the number of X values off of the training set. This in turn changes the counting; this change in the combinatorics is the sole cause of the increase in generalization accuracy accompanying an increase in the size of the training set. Nothing more mysterious is going on. The increase in (8) - and by implication the increase in (6.68) as well - occurs simply from counting how many hypothesis functions exist which agree with a (fixed but unknown) target function a certain number of times for questions off of the training set. This point is emphasized by the analysis in the following section.

III. CALCULATION OF GENERALIZATION ERROR WHEN THE TESTING SET IS DISTINCT FROM THE TRAINING SET

In addition to its insistence on looking at probabilities conditioned on m rather than on L, there are a number of other peculiar aspects of exhaustive learning. In particular, the error function it uses, although it is conventional in much machine learning research, has a number of major disadvantages. These arise from the fact that the exhaustive learning error function is based on looking at *all* X values for errors, including those in the training set. In other words, the error function gives a generalizer credit simply for reproducing a training set. Use of this error function forces one to calculate quantities which are completely oblivious of the natural distinction between error on the training set and error off of it. Especially when, as in exhaustive learning, a generalizer is explicitly constructed so that it will never have any errors on the training set, it is odd to fail to distinguish such (never-occurring) on-training set errors from (occurring quite often) errors off of the training set. After all, when comparing noise-free generalizers, it is *only* the off-training set behavior which can distinguish the generalizers.

There is another, more concrete problem with the exhaustive learning error function, which turns out to be related to the on/off training set dichotomy. This problem is the fact that because the exhaustive learning generalizer always perfectly reproduces the training set, as m' is increased the upper bound on the error function shrinks. The immediate question is how much of (6.68) simply reflects this fact that exhaustive learning uses an error function whose upper bound shrinks with m.

To address this issue, we must replace the old error function Er(f, h) with the normalized error function, $Er(h, f, L) \equiv \Sigma_{x \in L_X} \pi(x)\,[1 - \delta(f(x), h(x))] \;/\; \Sigma_{x \notin L_X} \pi(x)$. Given the noise-free nature of P(h | L) and P(L | f), this error function will always have the same lower and upper bound, regardless of m'. Moreover, it directly measures off-training set error; since there is no noise in exhaustive learning, the normalized error function can be rewritten as $\Sigma_{x \notin L_X} \pi(x)\,[1 - \delta(f(x), h(x))]$ $/\; \Sigma_{x \notin L_X} \pi(x)$.

For the same reasons mentioned in the discussion preceding (8), we will calculate P(E | f, L) rather than P(E | f, m). For simplicity, again make the assumptions used to derive (9). We get the following:

(10) When both T(h) and π(x) are constant and one uses the off-training set error function:

i) For low E, $P(E \mid f, L) \propto [(r - 1)^{-E} \times (1 - E)]^{m'} \times \omega(E)$, for some function ω(.), and

ii) For high E, $P(E \mid f, L) \propto [(r - 1)^{-E} \times E]^{m'} \times \omega'(E)$, for some function ω'(.).

Proof: First continue along with the proof of (8). We get $P(E \mid f, L) = \{\Sigma_{h \supset L}\ \delta(Er(h, f, L), E)\}$ / $\{\Sigma_{h \supset L}\ 1\}$, which holds independent of the error function used. Whereas in the proof of (8) we next replaced the error function with the expression $[n - m' - S(f, h, X - L_X)] / n$, our new error function must instead be replaced by $[n - m' - S(f, h, X - L_X)] / [n - m]$. Plugging this in, exactly as in the proof of (8), gives $P(E \mid f, L) = \{\Sigma_{h \supset L}\ \delta(S(f, h, X - L_X), (1 - E)(n - m'))\}\ /\ \{\Sigma_{h \supset L}\ 1\}$. Carrying through the combinatorics gives the same result as in (8), $P(E \mid f, L) = C_z^{(n-m')} \times (r - 1)^{(n-m'-z)}$ / $r^{(n-m')}$, except that here $z \equiv (1 - E)(n - m')$ rather than $(1 - E)n - m'$. Now we must follow along with the proof of (9). First rewrite our formula for $P(E \mid f, L)$ as $C^{n-m'}_{(1-E)(n-m')} \times (r - 1)^{(n-m')E}$ / $r^{n-m'}$. Now we want to calculate the ratio $P(E \mid f, L') / P(E \mid f, L)$, where L is a training set of m' distinct elements and L' is a training set of m' + 1 distinct elements. Our answer is $r \times (r - 1)^{-E} \times [(n - m')]^{-1} \times \{[(n - m' - En + Em')! / (n - m' - 1 - En + Em' + E)!] \times [(nE - m'E)! / (nE - m'E + E)!]\}$. Now notice that since $E \in [0, 1]$, it is impossible for all of these factorials to be defined. This reflects the fact that since E is always normalized properly, as one changes m' and therefore the number of elements in $X - L_X$, one changes the possible values of E. To get around this problem note that it is the shape of the curve connecting the allowed values of $P(E \mid f, L)$ (as a function of E) which interests us. To get this curve we must analytically extend the factorials. The most natural thing to extend them to is gamma functions, but the precise extension won’t matter so long as we restrict our attention to the low E and the high E regimes. For low E, $(n - m' - 1 - En + Em' + E) \cong (n - m' - En + Em') + 1$. This means our first ratio of factorials $\cong (n - m' - En + Em') = (n - m') \times (1 - E)$. Similarly, our second ratio of factorials $\cong 1$ for low E. Therefore for low E, $P(E \mid f, L) = r^m \times (r - 1)^{-Em'} \times (1 - E)^{m'} \times \omega(E)$ for some function $\omega(.)$. This holds regardless of the relative sizes of n and m. (As usual, for n >> m, we could replace m' with m if we wanted to.) For large E, instead of setting E to 0 everywhere it appears by itself in a factorial we can set it to 1. The result is $P(E \mid f, L) = r^m \times (r - 1)^{-Em'} \times (E)^{m'} \times \omega'(E)$ for some function $\omega'(.)$. QED.

In particular, for $r = 2$ and E large, $P(E \mid f, L) \propto (E)^m \times \omega'(E)$. This result should be contrasted with the $(1 - E)^m$ behavior of equation (8) and equation (6.68) of [Hertz et al. 1991]; if one goes to the off-training set error function, then in certain regimes the $(1 - E)^m$ generalization behavior of exhaustive learning not only disappears, it is actually reversed. Phrased differently, $(1 - E)^m$ generalization behavior is not as significant as it appears. For a random (i.e., exhaustive learning) generalizer, this behavior is expected whenever the error measure counts as successful "generalization" the correct reproduction of the input-output pairs in the training set. Loosely speaking, the gain in "generalization accuracy" with increased m implied by the relation $P(E \mid f, m) \propto (1 - E)^m$ simply reflects the fact that as m increases, there are more points on which you're assured of guessing the output correctly (since you're assured of reproducing the training set correctly). In other words, the apparently exponential increase of equation (6.68) of [Hertz et al. 1991] is a reflection of what is really a linear phenomenon.

This might be a somewhat surprising result. After all, in the limit of infinite n, one might expect that the probability that a randomly chosen X value $\in L_X$ is essentially 0. Accordingly, so this line of reasoning goes, it should not matter if one averages over all X values (as in the conventional exhaustive learning error function) or if one averages only over X those outside of L_X (as in the off-training set error function). Equation (10) is a formal proof that this line of reasoning does not hold. To help understand this result intuitively, first note that it does not matter if $X = \mathbf{R}^k$ (in which case the probability that a randomly chosen X value $\in L_X$ equals 0 exactly). This is because X is always effectively countable (see appendix A). Second, recall that if P(f) were uniform over all f, then it would make no difference how one guesses for off-training set error (see [Wolpert 1992a]). Given this, it would be quite surprising if somehow $(1 - E)^m$ behavior *were* preserved when one shifts to an off-training set error function. Finally, it's worth noting that for the case considered in (10), the average error, $\Sigma_E\ E \times P(E \mid f\ m)$, $= 1 / r$, independent of m.[10] In certain respects, this is more "intuitively reasonable" than the generic e^{-m} or 1 / m behavior associated with the error function of the conventional analysis of exhaustive learning (see page 152 of [Hertz et al. 1991]).

It's important to note that even though (10) was derived under very specific assumptions for T(h) and $\pi(x)$, the general result - $(1 - E)^m$ behavior goes away for an off-training set error function - holds generically. Broadly applicable results are not in hand, but some salient arguments can be made which should illustrate the point. In particular, note from the proof of (10) that for uniform $\pi(x)$, $P(E \mid f\, L) = \{\Sigma_{h \supset L} [T(h) \times \delta(Er(f, h), E)]\} \;/\; \{\Sigma_{h \supset L} T(h)\}$. Allowing T(h) to be non-uniform means the results of (10) might not obtain. However allowing non-uniform T(h) also means that we can have essentially arbitrary $P(E \mid f\, L)$. For example, if T(h) is infinitesimal for all h except for one special h, h', then by appropriate choice of h' we can make $P(E \mid f\, L)$ essentially a delta function about any of the allowed values of E.

Equation (10) points up two dangers inherent in many machine learning formalisms. First, if one uses an error function which roams over the training set as well as off of it (i.e., if one confuses generalization with learning), one can have the illusion of impressive generalization when in fact nothing useful has been achieved. Second, especially when using those kinds of error functions, any "exponential" generalization behavior might have little (if any) significance.

It is especially important to keep these dangers in mind when one considers the PAC machine-learning formalism, which relies heavily on exponential vs. non-exponential behavior as way of distinguishing "good" generalizers from "bad" generalizers. In this regard, it is worth noting that PAC and exhaustive learning use the same $P(L \mid f)$, the same error function, and have the same distribution of interest, $P(E \mid f\, m)$. (See [Wolpert 1992c].) It is a direct result of these shared traits that PAC has the result $\delta \le (1 - \varepsilon)^m \times |H'|$ (where δ, loosely speaking, is a bound on $P(E \ge \varepsilon \mid f\, m)$) and exhaustive learning has the functionally identical m-dependence $P(E \mid f\, m) \propto (1 - E)^m \times \rho_0(E)$.

IV. DISCUSSION

The idea behind calculating P(E | f, training set has size m) is to fix the target function, many times choose a random training set of size m, and measure the resultant distribution in E. Do this for several m. Equation (6.68) of [Hertz et al. 1991] says how this distribution varies with m. In practice though, often we're not going to many times take m samples of the target function, collect statistics, and then many times do the same thing for *completely novel* training sets of size m + 1. Rather we're going to take m samples of the target function, and then add another sample point (to the training set already in hand) to increase the size of the training set to m + 1. In practice, we usually want to know something about how the probability distribution over E will change when we add this next sample point, "on average". As it turns out, the answer to this question is given precisely by P(E | f, m). More formally,

11) Assume that for all training sets L,

P(training set = L | f, the training set has size m) =

P(first m pairs in the training set = L | f, the training set has size m + 1).

Then the average value of {the change in the probability of error E which accompanies the addition of an input -output pair to a training set of size m} is given by

P(E | f, m+1) - P(E | f, m).

Proof: Let ω indicate a generic training set, and let "$\omega=\{L\}^+$" mean that we have a training set, size m + 1, whose first m pairs are given by L (nothing else is known about the training set). The quantity we want to calculate is $\Sigma_{\theta;m}$ P(L | f, m) $\times$ [P(E | f, $\omega=\{L\}^+$) - P(E | f, L)]. Expanding P(E | f, $\omega=\{L\}^+$), we get $\Sigma_{\theta'=\{\theta\}^+}$ P(E, f, L') / $\Sigma_{\theta'=\{\theta\}^+}$ P(f, L') = [$\Sigma_{\theta'=\{\theta\}^+}$ P(E | f, L') $\times$ P(L', f)] / [$\Sigma_{\theta'=\{\theta\}^+}$ P(L', f)]. We can rewrite this as [$\Sigma_{\theta'=\{\theta\}^+}$ P(E | f, L') $\times$ P(L', f, the size of L' = m+1)] / [$\Sigma_{\theta'=\{\theta\}^+}$ P(L', f, the size of L' = m+1)]. Dividing top and bottom by P(f, the size of L' = m+1), we get $[\Sigma_{\theta'=\{\theta\}^+} P(E \mid f, L') \times P(L' \mid f, \text{the size of } L' = m+1)]$ / $[\Sigma_{\theta'=\{\theta\}^+} P(L' \mid f, \text{the size of } L' = m+1)]$. The denominator can be rewritten as $P(\omega=\{L\}^+ \mid f, \text{the size of } \omega = m+1)$. Under our assumption concerning probabilities of training sets, this = P(training set = L | f, the training set has size m) = $P(L \mid f, m)$. Therefore the quantity we want to calculate is $\Sigma_{\theta;m} [\Sigma_{\theta'=\{\theta\}^+} P(E \mid f, L') \times P(L' \mid f, \text{the size of } L' = m+1)] - \Sigma_{\theta;m} P(E \mid f, L) \times P(L \mid f, m)$. Because training sets are ordered, we can rewrite the double sum as $\Sigma_{\theta';m+1}$, giving us $\Sigma_{\theta';m+1} P(E \mid f, L') \times P(L' \mid f, m+1) - \Sigma_{\theta;m} P(E \mid f, L) \times P(L \mid f, m)$. Using equation (1), we see that this just equals $P(E \mid f, m+1) - P(E \mid f, m)$. QED.

It seems that for almost all reasonable schemes for creating training sets, the assumption in (11) holds, and therefore it suffices to calculate $P(E \mid f, m)$. For example, assume that $P(L_Y \mid f, L_X, m) = \Pi^m_{i=1} t[f(L_X(i)), L_Y(i)]$ for some function t(., .) obeying $\Sigma_y t(z, y) = 1 \; \forall z \in Y$ (as when we have Gaussian noise for example), and assume further that $P(L_X \mid f, m) = \Pi^m_{i=1} \pi(L_X(i))$ for some function $\pi(.)$ obeying $\Sigma_x \pi(x) = 1$ (as in i.i.d. sampling). Now it is always true that $P(L \mid f, m) = P(L_Y \mid f, L_X, m) \times P(L_X \mid f, m)$. Therefore, given the assumptions for $P(L_Y \mid f, L_X, m)$ and $P(L_X \mid f, m)$, the probability P(first m pairs in the training set = L | f, the training set has size m + 1) just equals $\Sigma_{x,y}$ P(training set = {L, (x, y)} | f, m+1) = $\Sigma_{x,y} [\{\Pi^m_{i=1} t[f(L_X(i)), L_Y(i)] \times \pi(L_X(i))\} \times t(f(x), y) \times \pi(x)] = \Pi^m_{i=1} t[f(L_X(i)), L_Y(i)] \times \pi(L_X(i))$. This last term equals $P(L \mid f, m)$ however, corroborating the assumption made in (11).

As was mentioned in connection with equation (8), in general all of one’s knowledge about the machine learning problem at hand should go on the right hand side of the conditional probability of interest. Therefore, in a certain sense $P(E \mid h, L)$ is what we’re really interested in; in any single machine-learning experiment, it is usually the case that our provided information is the training set we used along with the hypothesis function we fit to that training set. In this regard it is interesting to note that the expression on the right-hand side of equation (8) is identical to the expression worked out in [Wolpert 1992a] for $P(E \mid h, L)$ under the assumption of a "maximum-entropy universe".[11]

This shouldn't be too surprising because for many scenarios there is a formal symmetry under interchange of f and h (see [Wolpert 1992a]). In particular, because the error function is symmetric under such an interchange, this is true in exhaustive learning if $T(h) = P(h) = P(f)|_{f=h}$. Accordingly, under these conditions equation (6.68) holds with hypothesis and target functions reversed, i.e., with the appropriate self-averaging approximation,

$$\text{(12)} \qquad P(E \mid h, m) = \frac{[1 - E]^m \times \rho_0(E)}{\Sigma_{E'} [1 - E']^m \times \rho_0(E')} .$$

(where $\rho_0(E)$ is now defined as $\Sigma_f \{k \times P(f) \times \delta(E, Er(f, h))\}$ rather than as $\Sigma_h \{k \times T(h) \times \delta(E, Er(f, h))\}$).

Proof: It is always true that both $P(f \mid h, L) = P(f \mid L)$ as well as $P(h \mid f, L) = P(h \mid L)$. In addition, the error function in exhaustive learning is symmetric under $f \leftrightarrow h$. Therefore, since $P(h \mid L)$ and $P(L \mid f)$ are the only remaining quantities which are specified in exhaustive learning, we only have to prove that the both the pair $\{P(f \mid L)$ and $P(h \mid L)\}$ and the pair $\{P(L \mid f)$ and $P(L \mid h)\}$ are interchangeable in exhaustive learning. Once we have done this, there will be no formal distinction in the math between f and h. Now in [Wolpert 1992b] it is proven that whenever $P(L_X \mid f)$ is independent of f and there is no noise in the sampling (both of which conditions hold in exhaustive learning), $P(f \mid L) \propto P(f) \times \delta(L \subset f)$, where the proportionality constant depends on L as usual. This establishes the calculational symmetry under $f \leftrightarrow h$ of the pair $\{P(f \mid L)$ and $P(h \mid L)\}$. It is also proven in [Wolpert 1992b] that if $P(h \mid L) \propto P(h) \times \delta(L \subset h)$ (as in exhaustive learning, if $T(h) = P(h)$), then $P(L_X \mid h) = P(L) \times k(L)$, where k(.) is the usual normalizing constant $[\Sigma_h T(h)\ \delta(L \subset h)]^{-1} = [\Sigma_h P(h)\ \delta(L \subset h)]^{-1}$. Now $P(L) = \Sigma_f P(L \mid f) \times P(f)$, and it is proven in [Wolpert 1992b] that $P(L \mid f) = P(L_Y \mid L_X, f) \times P(L_X \mid f)$. Given our sampling assumption, this means that $P(L) =$

Σ_f $\{\kappa \times [\Pi_{i=1}^{m}\ \pi(L_X(i))] \times \delta(L \subset f) \times P(f)\}$ for some constant κ, which in turn equals $\{\kappa \times [\Pi_{i=1}^{m}\ \pi(L_X(i))]\} \times \Sigma_f \{P(f) \times \delta(L \subset f)\}$. But by assumption $P(h) = P(f)|_{f=h}$, which means that Σ_h $P(h)$ $\delta(L \subset h) = \Sigma_f \{P(f) \times \delta(L \subset f)\}$. Therefore $P(L_X \mid h) = \{\kappa \times [\Pi_{i=1}^{m}\ \pi(L_X(i))]\}$, and $P(L \mid h)$ is the same function of h as $P(L \mid f)$ is of f. QED.

In equation (12), instead of one target function and many possible hypothesis functions, we have a single hypothesis function and many possible target functions. The distinction from the conventional exhaustive learning scenario can be understood as follows:

In exhaustive learning, we conduct many learning experiments, with target functions chosen according to the probability distribution P(f) and training sets L chosen from the target function according to $P(L \mid f)$. Look at all those cases where the training set has size m and the target function is some pre-fixed function f. Then equations (2) and (5) give the probability of error value E.

In the scenario considered in (12), we also conduct many learning experiments, with target functions chosen according to the probability distribution P(f) and training sets L chosen from the target function according to $P(L \mid f)$. Again we look at all those cases where the training set has size m. However rather than restricting our attention to those cases with a particular pre-fixed target function f, we restrict our attention to those cases where the hypothesis function chosen by the generalizer is some particular pre-fixed hypothesis function h. Then (12) gives the probability of error E. Intuitively, (12) can be viewed as telling us something about the efficacy of sticking with the same hypothesis function h as the training set size changes.

FOOTNOTES

1. Generically, exponential $P(E \mid f\ m)$ results in either e^{-m} or 1/m behavior for the average error,

Σ_E E $\times$ P(E | f m). (See page 152 of [Hertz et al. 1991].) To parallel the PAC formalism, rather than concentrate on the average error, this paper concentrates on P(E | f m) and the issue of whether or not P(E | f m) is exponential.

2. Usually exhaustive learning is described as allowing uncountably infinite input spaces; X = $\mathbf{R}^k$. This is misleading. In exhaustive learning, the input space is always effectively discrete, and can be made explicitly discrete without any loss of generality. See appendix A after reading the rest of this section's definition of exhaustive learning.

3. As a general rule, the use of such quadruples is helpful whenever only the size of the training set is specified in the conditional probability being calculated rather than the training set *in toto*. See the following section of this paper.

4. Strictly speaking, conventional Bayesians actually use event spaces of the form {f, L, other stuff}, where "other stuff" consists of things like prior knowledge, model class, noise and regularizing parameters, etc. The important point is that "other stuff" never includes hypothesis functions.

5. In addition to making Bayesian analysis unsuited for investigating exhaustive learning, this also means that such analysis can not address situations where the prior one assumes in one's generalizer is different from the actual one describing the physical universe. See [Wolpert 1992a].

6. It turns out that if T(h) = $P(f)|_{f=h}$, the (unknown to the researcher) prior probability of target function f = h, then the sum is independent of h and T(h) *is* proportional to P(h) (which in turn implies that P(h) $\propto$ $P(f)|_{f=h}$). Of course, in the real world, there is no way of knowing that T(h) = $P(f)|_{f=h}$. In general however, when T(h) $\neq$ $P(f)|_{f=h}$, the sum varies with h. See [Wolpert 1992b] for a discussion of this issue and its implication for what priors "really mean".

7. Note that in exhaustive learning, $P(L_X \mid f)$ is independent of f. This state of affairs is quite common in supervised machine learning, though rarely explicitly stated. The general properties accruing from such sampling assumptions are discussed in [Wolpert 1992b].

8. The need for T(h) to be nowhere-zero arises not in the calculations of exhaustive learning, but rather when one tries to apply the exhaustive learning generalizer in the real world. If rather than being infinitesimal T(h') equalled zero exactly for some particular hypothesis h', use of the exhaustive learning $P(h \mid L)$ would implicitly set strong - and in the real world unverifiable - restrictions on f. This is because if T(h') = 0, then the exhaustive learning $P(h \mid L)$ is undefined for L = h'. Since we have no noise in the sampling assumption, the only way we could be assured of avoiding such an L in the real world is if the corresponding f can not occur (assuming $\pi(x)$ is non-zero for all x), i.e., if $P(f)|_{f=h'}$ equals 0 exactly. In other words, if we set T(h') to zero, either we must somehow know that it is physically impossible for the target function f = h' to occur, or we must be willing to have our $P(h \mid L)$ undefined for certain L. To avoid this dilemma, one should ensure that T(h) never equals zero exactly. Unfortunately this can be quite difficult if, as is conventional, functions h are viewed as neural nets, so that rather than writing down T(h) directly one writes down a sort of T(**w**), where **w** is a weight vector parameterizing h.

9. After this paper was written we received a technical report by Seung et al. [1992] which also extends the analysis in [Schwartz et al. 1990]. In particular, they restricted themselves to the case where input-output functions are explicitly parameterized (as neural nets). Then they analyzed certain aspects of an approximation (the “annealing approximation”) intimately related to the self-averaging assumption made in [Schwartz et al. 1990].

10. From the proof of equation (10), we see that the average number of disagreements between f

and h off of the training set equals $\Sigma_{z=0}^{n-m'}$ $[C_z^{n-m'}\ z\ (r - 1)^{n-m'-z}\ /\ r^{n-m'}]$. Now $(x + y)^{n-m'}$ = $\Sigma_{z=0}^{n-m'}$ $[C_z^{n-m'}\ x^z\ y^{n-m'-z}]$. Therefore $x \times \partial_x[(x + y)^{n-m'}] = \Sigma_{z=0}^{n-m'}$ $[C_z^{n-m'}\ z\ x^z\ y^{n-m'-z}]$. If we plug in $(r - 1)^{-1}$ for x and 1 for y, we see that the average number of disagreements equals (n - m') / r. Therefore the average error equals 1/r.

11. There is one small difference. In the expression in [Wolpert 1992a] for $P(E \mid h, L)$, $z = (n - m) \times (1 - E)$ rather than $n(1 - E) - m$. This difference follows from the fact that the error function used in [Wolpert 1992a] differs by a factor of n / (n - m) from the one used in equation (8); [Wolpert 1992a] uses the off-training set error function introduced in this paper in section III.

APPENDIX A

Why in exhaustive learning X is always effectively countable.

Sometimes exhaustive learning is presented as though it could have an uncountably infinite input space, $X = \mathbf{R}^K$ for some K. In truth however, in exhaustive learning the input space is effectively discrete, and can be made explicitly discrete without any loss of generality. This follows from the fact that in exhaustive learning, even though the number of weight vectors **w** might be uncountable, the number of possible functions parameterized by those vectors - i.e., the number of possible h - *is* countable (c.f. equation 6.67 in [Hertz et al. 1991] or equation 2.10 of [Schwartz et al. 1990].)

To prove that X is effectively discrete, assume for simplicity that K = 1. Assume further, also for simplicity, that for any finite interval S in X, and any h for which $P(h \mid f) \neq 0$, the number of changes in the output value of h(x) as x moves across S is finite. (Recall that Y is discrete. Note also that $P(h \mid f) = \Sigma_L P(h \mid L\ f)\ P(L \mid f) = \Sigma_L P(h \mid L)\ P(L \mid f)$.) This property will certainly hold for a typical neural net h, and it will hold for most of the other functions usually used in supervised learning as well. Accordingly, although it is relatively straight-forward to generalize to the case where the number of changes can be infinite, for the purposes of this paper there is no need to add such a complication.

Consider the x interval [0, 1). Define ch(h) as the set of all $x \in [0, 1)$ for which $\exists\ \delta > 0$ such that $\forall\ \varepsilon$ where $0 < \varepsilon < \delta$, $h(x - \varepsilon) \neq h(x + \varepsilon)$. ch(h) is the set of all x values at which h(x) changes, in the interval [0, 1). By hypothesis, $ch(h_i)$ is finite $\forall\ h_i$ with which we're concerned (i.e., for all h_i such that $P(h_i \mid f) \neq 0$). Therefore $\{\cup_i ch(h_i)\}$, a countable union of countable sets, is itself countable. Call this union $Ch\{[0, 1)\}$. Since it is a countable union of countable sets, $\cup_j Ch\{[j, j+1)\}$, where j runs over all integers, is a countable set.

Now by construction, the values of all of the h_i with which we're concerned are constant over each of the (countably many) regions demarcated by an adjacent pair of points from the set $\cup_j Ch\{[j, j+1)\}$. In other words, for any one of the countable number of regions in **R** lying in-between adjacent demarcating points, [a, b), it suffices to specify the values of the h_i at only a single point from within [a, b). (See figure 6.14 in [Hertz et al. 1991].)

Furthermore, by hypothesis the Gibbs generalizer always reproduces the training set. Since there are no a priori restrictions on $\pi(x)$, this means that there must be an h which reproduces any training set L containing input-output pairs i and $j \neq i$ such that both $L_X(i)$ and $L_X(j) \neq L_X(i)$ are contained in [a, b). Since training sets are formed by noise-free sampling of f, the only way this can be assured is if f also doesn't change across [a, b). Generalizing, we have shown that neither any of the h_i nor f can change over any of the (countably many) regions demarcated by an adjacent pair of points from the set $\cup_j Ch\{[j, j+1)\}$. In other words, for all function of interest, it suffices to specify the value of such a function at only a single point from within each demarcated region.

Accordingly, it suffices to treat X as discrete, by associating each separate value $x \in X$ with a separate one of the demarcated regions in **R**. So for example, the expression "$\pi(i \in X)$" implicitly means the integrated probability $\int dx\ \pi(x \in \mathbf{R})$, where the integral extends over the region associated with the index i. Similarly, to say that one X value does not "equal" another implicitly means that we are talking about two separate demarcated regions. So in particular, saying that an X value lies "outside of the training set" implicitly means that the region it's associated with has zero intersection with any of the regions containing the elements of the training set. It does *not* only mean that the exact, infinite precision X value differs from all of the exact, infinite precision X values in the training set.

More precisely, let X be our original input space, and let B be a set of n values, delineating the regions in $\cup_j\ Ch\{[j, j+1)\}$. The idea is to show that we can replace the input space X = **R** with the input space X = B. To this end, define B(x) as the region (i.e., B value) demarcated by those adjoining points in $\cup_j\ Ch\{[j, j+1)\}$ which contains the point $x \in \mathbf{R}$.

Let h be any hypothesis function such that $P(h \mid L)$ is not zero. By the argument given above, h is fully specified by giving its Y values for every point b in B. Accordingly, T(h) is fully specified if its argument h is only specified to that accuracy. Similarly, $\delta(L \subset h)$ is fully specified if h is only specified to that accuracy and if every value $L_X(i)$ is replaced by $B(L_X(i))$. Accordingly, as far as $P(h \mid L)$ is concerned, one can replace **R** with B.

Similarly, in this paper we will only deal with error functions like $\int dx\ \pi(x)[1 - \delta(f(x), h(x)] = \Sigma_{b \in B} \int_{(B^{-1}(b))} dx\ \ \pi(x)[1 - \delta(f(x), h(x)]\ =\ \Sigma_{b \in B}\ \{[1 - \delta(f[B^{-1}(b)], h[B^{-1}(b)])]\ \times\ \int_{(B^{-1}(b))} dx\ \pi(x)\}$. (The subscript on the integrals delineates a region of integration.) So here too, we can replace **R** with B, assuming we also replace $\pi(x)$ with $\pi(B(x)) \equiv \int_{x \in B^{-1}(b)} dx\ \pi(x)$.

Finally, the other quantity specified in exhaustive learning is $P(L \mid f) = P(L_Y \mid L_X\ f) \times P(L_X \mid f)$, where $P(L_Y \mid L_X\ f) = \delta(L \subset f)$. Accordingly, since f is constant within the regions indexed by $b \in B$, $P(L_Y \mid L_X\ f)$ is fully specified if we only specify f by its value in each region indexed by $b \in B$ and if we also replace the $L_X(i)$ with $B(L_X(i))$. In other words, to give the value of $P(L_Y \mid L_X\ f)$, we only need to know f and L_X to the accuracy of B. (This is exactly the same justification as the one we used to replace **R** with B in $\delta(L \subset h)$.)

On the other hand, $P(L_X \mid f) = \Pi^m_{i=1} \pi(L_X(i))$, and this quantity does *not* allow replacing X = **R** with X = B. To know the value of $P(L_X \mid f)$, we need to know more about L_X than is given by $B(L_X)$.

This is not a problem however. To see this, first note that we can use Bayes' theorem to write $P(f \mid L) = \{P(L_Y \mid L_X f) \times P(L_X \mid f) \times P(f)\} \ / \ \Sigma_f \{P(L_Y \mid L_X f) \times P(L_X \mid f) \times P(f)\}$. Since $P(L_X \mid f)$ is independent of f (see footnote 7), this equals $\{P(L_Y \mid L_X f) \times P(f)\} \ / \ \Sigma_f \{P(L_Y \mid L_X f) \times P(f)\}$ - the $P(L_X \mid f)$ term has dropped out. So as far as $P(f \mid L)$ is concerned we can replace **R** with B. Note also that any quantity of the form $\int dL_X \{P(L_X \mid f) \times \text{func}[B(L_X(1)), ..., B(L_X(m))]\}$ equals $\Sigma_{b_1,...,b_m} \int_{B^{-1}(b_1)} dx_1 \ ... \int_{B^{-1}(b_m)} dx_m \{\Pi^m_{i=1} \pi(x_i) \times \text{func}[B(x_1), ..., B(x_m)]\}$. This in turn equals $\Sigma_{b_1,...,b_m} \{\Pi^m_{i=1} \pi(b_i) \times \text{func}[b_1, ..., b_m]\}$. This expression is of the form $\Sigma_{L_X} \{P(L_X \mid f) \times \text{func}[L_X(1), ..., L_X(m)]\}$, if X = **R** is replaced with X = B. So as far as quantities like $\int dL_X \{P(L_X \mid f) \times \text{func}[B(L_X(1)), ..., B(L_X(m))]\}$ are concerned, we can replace **R** with B.

Now in this paper, we will not be concerned with any distribution P(E | stuff) which depends on L_X in any way other than via $P(f \mid L)$ or via $\int dL_X \{P(L_X \mid f) \times \text{func}[B(L_X(1)), ..., B(L_X(m))]\}$. Accordingly, although $P(L_X \mid f)$ could in theory prevent us from replacing X = **R** with X = B, it does not do this for any of the calculations of interest. Since none of the other quantities specified by exhaustive learning (the error function and $P(h \mid L)$) can prevent the replacement of **R** with B under any circumstances, we are allowed to make the substitution.

In practice, "countable" $\cup_j$ *Ch*{[j, j+1)} will almost always be finite, and we can set the cardinality of X to $n < \infty$. The countably infinite case is recovered in the limit of $n \to \infty$.

APPENDIX B

Proof of self-averaging when both T(h) is uniform and $\pi(x)$ is bounded, in the limit of n >> m.

The factor $k(L_X, f(L_X))$ is not independent of L_X, even when T(h) is uniform. However for such a T(h), $k(L_X, f(L_X))$ only depends on m', the number of distinct elements in L_X. This implies that our task in proving self-averaging is to show, in effect, that we can ignore all terms in the sum giving P(h | f m) which have $m' \neq m$. We will be able to do this because the number of terms in the sum which have m' = m so exceeds the number of terms which have $m' \neq m$.

Start by combining the equation in the middle of the proof of (5) with equation (2): P(E | f m) $= \Sigma_h \, \delta(E, Er(f, h)) \times T(h) \times \Sigma_{L_X;m} \{k(L_X, f(L_X)) \times \delta(S(f, h, L_X), m') \times \Pi_{i=1}^{m} \; [\pi(L_X(i))]\}$. As an aside, note that if E = 1, the $\delta(E, Er(f, h))$ factor means that $Er(f, h)$ must equal 1, which in turn means that f and h must disagree everywhere (since $\pi(x)$ is nowhere zero - see (6)). This means that $\delta(S(f, h, L_X), m')$ equals 0 for all h in the sum. Consequently, P(E = 1 | f m) = 0, in agreement with the $\propto (1 - E)^m$ result of self-averaging.

First we must derive a lower bound on N, the number of X on which f and h agree. To do this we again use the fact that the $\delta(E, Er(f, h))$ term in the sum over h means we can set $Er(f, h)$ to E. Now $E < \varepsilon$ and ε is a number which is independent of n and < 1. Therefore, in the limit of n >> m (i.e., with m fixed, in the limit of $n \to \infty$), $nE << n \cong n - m$; $nE << n - m$. This means that $n \times \Sigma_x \pi(x) \times [1 - \delta(f(x), h(x))] << n - m$; $\Sigma_x \pi(x) \, \delta(f(x), h(x)) >> m / n$. Now $\Sigma_x \pi(x) \, \delta(f(x), h(x)) \leq \max_x \pi(x) \times \Sigma_x \, \delta(f(x), h(x))$. Therefore $\Sigma_x \, \delta(f(x), h(x)) >> m / [n \times \max_x \pi(x)]$. However by hypothesis $[\max_x \pi(x)] / [\min_x \pi(x)] \leq \sigma$, which means that $\max_x \pi(x) \leq \sigma / (\sigma + n - 1)$. Plugging this in, $\Sigma_x \, \delta(f(x), h(x)) >> m / [n\sigma / (\sigma + n - 1)]$. In our limit of $n \to \infty$, the right-hand side become m / σ; N >> m / σ. Since σ is just a pre-set finite constant, in fact we have N >> m.

Now the inner sum over L_X in the formula given above for P(E | f m) can be rewritten as

$\Sigma_{L_X;m'<m}$ {k(., .) × Π × δ(S(., ., .), m')} + $\Sigma_{L_X;m'=m}$ {k(., .) × Π × δ(S(., ., .), m')}, with obvious notation. Using the bound on N we will prove that the first of these sums is insignificant compared to the second for all *h* of interest (i.e., for all *h* for which Er(f, *h*) = E), and can therefore be ignored. This will allow us to set m' = m, which we will use to prove self-averaging. (See appendix B in [Wolpert 1992b] for an intuitive version of this argument.)

It is fairly straight-forward to prove m' = m using the uniformity of T(h). To keep the argument as general as possible however, we will prove m' = m without making any assumptions for T(h). (Only after this will we exploit the uniformity of T(h).)

The first step in the proof is to get rid of the Π factors; bound the ratio of our two sums by

$$\frac{\Sigma_{L_X;m'<m} \{k(.,.) \times \Pi \times \delta(S(.,.,.), m')\}}{\Sigma_{L_X;m'=m} \{k(.,.) \times \Pi \times \delta(S(.,.,.), m')\}} \leq \sigma^m \times \frac{\Sigma_{L_X;m'<m} \{k(.,.) \times \delta(S(.,.,.), m')\}}{\Sigma_{L_X;m'=m} \{k(.,.) \times \delta(S(.,.,.), m')\}}.$$

Since σ is some pre-set constant, if the ratio of sums on the right-hand side goes to 0 as $n \to \infty$, we can take m' = m, as desired. The fact that we are currently allowing non-uniform T(h) affects things via the k(., .) factors occurring in the summands.

Let X' be the set of N X values on which f and *h* agree. Due to the δ functions in the summands, every non-zero term in both of our two sums over L_X equals the normalization constant k for P(*h* | L = {L_X, f(L_X)}), for an L_X all of whose elements come from X'. Due to the lower bound on N, we know that N >> the size of any of these L_X's.

To prove that the ratio of sums on the right-hand side goes to 0, we will construct a set of one-to-one mappings between (non-zero) terms in the denominator sum and (non-zero) terms in the numerator sum. (I.e., between k of an ordered sets of m elements chosen from X', all of which are distinct, and k of an ordered sets of m elements from X', with m' < m of the elements distinct.) Each such mapping will range over all of the terms in the numerator sum, but only a subset of the terms in the denominator sum. Indicate these mappings by Γ_i(a denominator term) = (a numerator term), where i indexes the mappings. Indicate the domain of Γ_i (which is a set of terms occurring in the denominator sum) by D(i), and the range (the set of all terms in the numerator sum) by R. We will construct the mappings so that D(i) has zero intersection with D(j) for $i \neq j$.

It will turn out that for any i and any term $d \in D(i)$, $d \geq \Gamma_i(d)$. (In other words, the k(., .) term associated with the L_X d is larger than the k(., .) term associated with the L_X $\Gamma_i(d)$.) This means that $\Sigma_{d\in D(i)}\, d \;\geq\; \Sigma_{c\in R}\, c$. Since R is the set of all the terms in the numerator sum, we can rewrite this inequality as $\Sigma_{d\in D(i)}\, d \;\geq\; \Sigma_{L_X;m'<m}\, \{k(., .) \times \delta(S(., ., .), m')$. With κ the number of mappings (i.e., the number of Γ_i), $\Sigma_i \Sigma_{d\in D(i)}\, d \;\geq\; \kappa \times \Sigma_{L_X;m'<m}\, \{k(., .) \times \delta(S(., ., .), m')$. Since $D(i) \cap D(j{\neq}i) = \varnothing$, the double-sum on the left is less than (or equal to) our denominator sum, $\Sigma_{L_X;m'=m}\, \{k(., .) \times \delta(S(., ., .), m')$. Therefore the denominator sum is more than κ times as big as the numerator sum. We will show that $\kappa \rightarrow \infty$ as $n \rightarrow \infty$, which will complete the proof that we can take m' = m.

There are five properties we need of the Γ_i: 1) They are one-to-one; 2) Their range is the set of all terms in the numerator sum; 3) Their domains have zero intersection; 4) The number of them goes to infinity as n does; 5) for all $d \in D(i)$, $d > \Gamma_i(d)$. In what follows, we assume without loss of generality that X is the set of the first n integers, {1, 2, ..., n}, and X' the first N integers, {1, 2, ..., N} (i.e., whenever referring to “an X value”, we really mean the ordinal index of that X value, where the X values have been arranged so that the values on which f and h agree come first).

Rather than directly construct the Γ_i, we will construct their inverses, Γ_i^{-1}. We start with Γ_0^{-1}. Let Λ_X be a particular L_X chosen with m' < m. It consists of an ordered set of m X values (at least two of which are identical), each chosen from X'. In other words, it consists of an m-dimensional vector, each component of which can equal any one of the integers from 1 to N. A total of m - m' of the m components of Λ_X are identical to (at least) one of the remaining m' components. We map Λ_X to Λ'_X, an L_X with m' = m (i.e., a vector all of whose components are distinct), via Γ_0^{-1}, as follows:

1) Let Ξ_0 be the set of m' distinct values found in the components of Λ_X. The components of the vector Λ_X, are all contained in the set Ξ_0.

2) Let S be an m-dimensional vector, m' of whose components are identical to those from Λ_X. (The remaining m - m' components of S equal zero - a value not found in X.) These m' components are chosen to be the first m' components of Λ_X all of which have distinct X values. In other words, the first component of S is identical to the first component of Λ_X; the second component of S = 0 if $\Lambda_X(2) = \Lambda_X(1)$, otherwise it equals $\Lambda_X(2)$, etc.

Note that every value in Ξ_0 is found as a component of S and visa-versa, and that all these values are found in X'. However sInce N >> m, there are many values in X' not found in Ξ_0.

3) Let T be the vector Λ_X - S.

Note that every non-zero component of T has a value already found in one of S's non-zero components. In other words, if $T_i \neq 0$, $T_i = S_j$ for some $j < i$. Moreover, if $T_i \neq 0$, $T_i = \Lambda_X(i)$, and $\Lambda_X(i) = \Lambda_X(j)$ for some $j < i$.

4) Let t_1 be the first non-zero component of T, t_2 the next non-zero component of T, etc. To create Λ'_X, start with Λ_X, and replace the component t_1 with t'_1, the lowest X value contained in $\{X' - \Xi_0\}$. Then replace t_2 with t'_2, the lowest X value contained in $\{X' - \Xi_1\}$, where $\Xi_1 \equiv \{\Xi_0 \cup t'_1\}$. Repeat this procedure until all non-zero components of T but the very last one have been replaced.

We have now fixed all but one of the components of Λ'_X. Note that the values of those components of Λ'_X fixed so far are, by construction, distinct from one another. Let the un-fixed component be component number J.

5) We will fix that last component of Λ'_X to a value chosen so that the sum of all the components of Λ'_X, $\Sigma^m_{i=1} \Lambda'_X(i)$, is a number which codes for the pattern of duplications of component values in Λ_X. Given this pattern and the values of the other components of Λ'_X, we can uniquely

recover Λ_X from Λ'_X.

More precisely, let F(m, m') be the number of possible duplication patterns of the components of an m-dimensional vector, m' of whose components are distinct. (A “duplication pattern” is a set of rules like “the value in component 3 is identical with the values in components 7 and 8”.) The number of such possible patterns for Λ_X is given by $F(m) \equiv \Sigma^{m-1}_{m'=1} \; F(m, i)$. Fix a one-to-one mapping Q from the integers {1, ..., F(m)} to the set of possible duplication patterns. Define $q \equiv Q^{-1}$(duplication pattern of Λ_X), the number coding for the duplication pattern of Λ_X.

6) Consider the smallest integer q' such that setting $\Lambda'_X(J) = q'$ means that $\Sigma^m_{i=1} \Lambda'_X(i)$, evaluated mod (F(m)), equals q. (Note that this q' is bounded by F(m).) If $q' \in \Xi_{m-m'}$, keep adding F(m) to it until we get a number not contained in $\Xi_{m-m'}$. (Note that we will have to add F(m) at most m - 1 times, since that’s the maximum number of elements in $\Xi_{m-m'}$.) Set $\Lambda'_X(J)$ to this value of q'. (Since N >> m and $m \times F(m)$ is a bounded function of m, this value of $q' \in X'$.)

7) This completes the specification of Γ_0^{-1}. Γ_1^{-1} is identical, except that it starts trying to set $\Lambda'_X(J)$ at the sum [the smallest integer q' such that setting $\Lambda'_X(J) = q'$ means that $\Sigma^m_{i=1} \Lambda'_X(i)$, evaluated mod (F(m)), equals q] + [$m \times F(m)$]. Similarly for all the other Γ_i^{-1}. We have as many Γ_i^{-1} as there are contiguous blocks of size [$m \times F(m)$] which we can fit into N, the number of possible values of $\Lambda'_X(J)$.

i) First we must prove that Γ_i as defined this way is one-to-one. To do this, note that Γ_i^{-1} as defined above is single-valued. To show that the inverse mapping, Γ_i, is single-valued, it suffices to show how, using only Λ'_X, one can find the single Λ_X which, when mapped through Γ_i^{-1}, gives that Λ'_X. This is done as follows: Find the sum of the components of Λ'_X. Evaluate this mod(F(m)). This gives the duplication pattern in Λ_X, i.e., it tells us which of the components of Λ_X are duplicates of which other components of Λ_X. All the non-duplicated components of Λ_X are identical to the corresponding components of Λ'_X. To find the values of the duplicated components, if, as an example, {a, b, c, ...} is a set of component indices which (are known to) have identical component values, with (a) the smallest, then set the values of Λ_X's components with indices a, b, c, ... all to the value of component (a) of Λ'_X.

ii) We must show that the range of Γ is all L_X with m' < m. This follows immediately from the fact that Γ^{-1} is defined for all such L_X, having as image an L_X with m' = m.

iii) We must show that the domains of the Γ_i (i.e., ranges of Γ_i^{-1}) are distinct for different i. To do this, consider the hypothesis that there is overlap: for some pair of values i and j ≠ i, there exists a vector Λ_{Xi} and a vector Λ_{Xj} such that $\Gamma_i^{-1}(\Lambda_{X_i}) = \Gamma_j^{-1}(\Lambda_{X_j})$. By the procedure outlined in (i) and the properties of the mod operator, $\Gamma_i^{-1}(\Lambda_{X_i}) = \Gamma_j^{-1}(\Lambda_{X_j})$ implies that $\Lambda_{X_i} = \Lambda_{X_j}$. However by the definition of Γ^{-1}, $\Gamma_i^{-1}(\Lambda_X)$ can not equal $\Gamma_j^{-1}(\Lambda_X)$ for $i \neq j$. QED.

iv) We must show that the number of Γ_i goes to infinity as i does. This follows directly from the statement at the end of (7).

v) We must show that "for all $d \in D(i)$, $d > \Gamma_i(d)$", i.e., the normalization constant k(d, f(d)) for any $d \in D(i)$ is greater than $k(\Gamma_i(d), f(\Gamma_i(d)))$. This follows immediately from the definition of k(., .): Since the set of distinct X values in d properly contains the set of distinct X values in $\Gamma_i(d)$, the set of h agreeing with f for the X values in d is a proper subset of the set of h agreeing with f for the X values in $\Gamma_i(d)$. Substituting $k(L_X, f(L_X)) = [\Sigma_h T(h) \times \delta(S(f(L_X), h, L_X), m'))]^{-1}$, we see that $k(d, f(d)) \geq k(\Gamma_i(d), \Gamma_i(f(d)))$, as required.

In this way, using only n >> m and the bounds on $\pi(x)$, we have proven that one can replace $\Sigma_{L_X; m' \leq m} \{k(., .) \times \Pi \times \delta(S(., ., .), m')\}$ with $\Sigma_{L_X; m'=m} \{k(., .) \times \Pi \times \delta(S(., ., .), m')\}$, thereby incurring arbitrarily small error in $P(E \mid f\ m)$. Now note that with T(h) constant, we can immediately write $k(L_X, f(L_X)) \propto r^{m-n}$ for all L_X for which m' = m. This value of k(., .) is independent of L_X, and therefore it can be replaced with a constant in $\Sigma_{L_X; m'=m} \{k(., .) \times \Pi \times \delta(S(., ., .), m')\}$, giving

$\Sigma_{L_X;m'=m}$ {$\Pi \times \delta$(S(., ., .), m')} (up to an overall multiplicative constant).

Note, however, that self-averaging requires that we can replace k(., .) with an overall multiplicative constant in the sum $\Sigma_{L_X;m'\leq m}$ {k(., .) $\times\ \Pi \times \delta$(S(., ., .), m')}. So we're not quite done; we must now prove that $\Sigma_{L_X;m'=m}$ {$\Pi \times \delta$(S(., ., .), m')} can be approximated arbitrarily well by $\Sigma_{L_X;m'\leq m}$ {$\Pi \times \delta$(S(., ., .), m')} (in the large n limit, for those h of interest). To do this we proceed as before: examine [$\Sigma_{L_X;m'<m}$ {$\Pi \times \delta$(S(., ., .), m')}] / [$\Sigma_{L_X;m'=m}$ {$\Pi \times \delta$(S(., ., .), m')}]. This ratio is bounded by $\sigma^m \times$ [$\Sigma_{L_X;m'<m}$ {δ(S(., ., .), m')} / $\Sigma_{L_X;m'=m}$ {δ(S(., ., .), m')}].

The sum in the denominator is the number of ways one can choose m distinct ordered X values so that f and h agree on those values. Since f and h agree with each other N times and L_X is an ordered set (see the introduction), this is just N! / m!. The numerator sum can be written as $\Sigma_{m'<m}\ \varphi(m, m') \times C^N_{m'}$]. ($C^N_{m'}$ is the number of ways of choosing m' distinct L_X values from out of N. φ(m, m') is the number of ordered ways of picking m objects from m' distinct bins, making sure that each bin is picked at least once. This quantity is also discussed in appendix C.) Therefore the ratio of sums is $\Sigma_{m'<m}$ {[φ(m, m') / m'!] $\times$ [(N - m)! / (N - m')!]}. This is strictly less than $(N - m - 1)^{-1} \times \Sigma_{m'<m}$ {[φ(m, m') / m'!]. Since the sum is independent of n, as $n \to \infty$, N >> m, and the sum gets swamped; our ratio of sums goes to 0. Therefore we can replace the sum $\Sigma_{L_X;m'=m}$ {$\Pi \times \delta$(S(., ., .), m')} with the sum $\Sigma_{L_X;m'\leq m}$ {$\Pi \times \delta$(S(., ., .), m')}, and therefore we have self-averaging. QED.

APPENDIX C

Calculating P(E | f, m) exactly, without invoking self-averaging.

Note that $\pi(x)$ is (often) under our direct control; for simplicity, assume it is a constant (and therefore equals 1 / n). Plugging this and the assumption that T(h) is constant into equation (4) gives

$$P(h \mid f, m) \propto (1/n)^m \times \Sigma_{L_X;m} \{\delta(S(f, h, L_X), m') \,/\, \Sigma_{h}\, \delta(S(f, h, L_X), m')\}$$

$$\propto \Sigma_{L_X;m} \{[\delta(h(.) \text{ agrees with } f(.) \text{ for all X values in } L_X] \,/$$

$$[\text{number of hypothesis functions which agree with } f(.) \text{ for all X values in } L_X]\}.$$

Now the denominator inside in the sum depends only on m'; it equals $r^{(n-m')}$. Therefore it makes sense to break up the sum into a sum of sums, each of the inner sums having m' fixed:

$$\text{(C.1)}\quad P(h \mid f, m) \propto \Sigma_{m'=1}^{m} \{r^{(m'-n)} \times \Sigma_{L_X;m,m'} [\delta(h(.) \text{ agrees with } f(.) \text{ for all X values in } L_X]\},$$

where by $\Sigma_{L_X;m,m'}$ is meant the sum over all m-element L_X where the m $L_X(i)$ take on only m' distinct values.

We must now evaluate $\Sigma_{L_X;m,m'}$ {δ(h(.) agrees with f(.) for all X values in L_X}. To do this, define $z \equiv n(1 - Er(f, h)) = S(f, h, X)$. Without loss of generality, re-order the X values so that the z agreements between h(.) and f(.) occur for the first z values of X. Our task is to count the number of ways to pick an ordered set of m X values, which take on m' < m distinct values all together, so that all of those X values they take on are contained in the first z values of X.

First note that if m' > z, this “number of ways” equals 0. Therefore the upper limit in the first sum in (C.1) can be replaced by min(m, z), and we can always assume that $m' \leq z$ for purposes of counting the “number of ways”.

The “number of ways” can be written as the product of two numbers. The first number gives the number of ways to choose m' distinct elements from amongst z possibilities, i.e., $C^{z}_{m'}$. The second number is the number of ways of assigning one of m' labels to each of m distinct elements, so that for each of the labels there is at least one element with that label. Write this second number as φ(m, m'). We have just proven that

$$P(h \mid f, m) \propto \Sigma_{m'=1}^{\min(m,\, n[1-Er(f,h)])} [r^{(m'-n)} \times C^{z}_{m'} \times \varphi(m, m')].$$

If we plug this into equation (1), and collect all normalization constants, then by definition of z, for any E such that n(1 - E) is an integer between 0 and n,

$$\text{(C.2)} \quad P(E \mid f, m) \propto \{\Sigma_{i=1}^{\min(m,\, n(1-E))} [r^{(i-n)} \times \varphi(m, i) \times C^{n(1-E)}_{i}]\} \times \{\Sigma_{h} \delta(E, Er(f, h))\}$$

$$= \{\Sigma_{i=1}^{\min(m,\, n(1-E))} [r^{(i-n)} \times \varphi(m, i) \times C^{n(1-E)}_{i}]\} \times C^{n}_{n(1-E)} \times (r-1)^{nE}.$$

(P(E | f, m) for all other E is zero, since those E can not occur.)

APPENDIX D

An example of when self-averaging does not hold.

Consider the situation where T(h) factors: $T(h) = \Pi_{x \in X} V(h(x), x)$, for some everywhere non-negative function V(., .). This is perhaps the simplest way one can have a non-constant distribution over a set of functions. Intuitively, this form for T(h) arises from viewing the functions h as a set of n pairs, $\{x_i, h(x_i)\}$; the value of $T(\{x_i, h(x_i)\})$ is simply the product of the n values $V(h(x_i), x_i)$ for some function V.

Write $d(x) \equiv \Sigma_{y \in Y} V(y, x)$. d(x) is finite and non-zero for all x ($\{d(x) = 0\} \Rightarrow \{V(y, x) = 0 \ \forall \ y \in Y\} \Rightarrow \{T(h) = 0 \ \forall \ h\}$). Now define $V'(y, x) \equiv V(y, x) / d(x)$. $T(h) = \{\Pi_{x \in X} d(x)\} \times \{\Pi_{x \in X} V'(h(x), x)\} \equiv D \times T'(h)$:

(D.1) For n >> m and $T(h) = \Pi_{x \in X} V(h(x), x)$,

$$P(h \mid f, m) = T'(h) \times [\Sigma_{x \in X} \{\delta(h(x), f(x)) \times \pi(x) / V'(f(x), x)\}]^m.$$

Proof: Note that we can cancel a D in the numerator of (5) with a D in the denominator, k^{-1}, i.e., (5) still holds if we replace T(h) everywhere with T'(h). Making this substitution, instead of $[k(L_X, f(L_X))]^{-1}$ in the denominator in (5) we now have $\Sigma_{\{h(x)\};x \notin L_X}\{[\Pi_{x \notin L_X}[V'(h(x), x)] \times \Pi_{x \in L_X}[V'(f(x), x)]\}$, where the sum is understood to extend over all possible sets of n - m' values $\{h(x)\}$, and where x is extending over the n - m' X values not contained in L_X (there are $r^{n-m'}$ terms in the sum). By pulling the $\Pi_{x \in L_X}$ term out of the sum we can rewrite this as the product of two terms: $\{\Sigma_{\{h(x)\};x \notin L_X} [\Pi_{x \notin L_X}[V'(h(x), x)]\} \times \{\Pi_{x \in L_X}[V'(f(x), x)]\}$.

Now note that $\Sigma_{\{h(x)\};x \notin L_X}$ is a sum over events all of which are an ordered set of numbers (here there are n - m' such numbers), all those numbers coming from a definite range (which here is Y), with repeats allowed in that set of numbers. Therefore this sum is formally equivalent to the $\Sigma_{L_X;m}$ found in (3) (where the value “m” in (3) is n - m' here.) Continuing with the formal parallel between $\{\Sigma_{\{h(x)\};x\notin L_X} [\Pi_{x\notin L_X} [V'(h(x), x)]\}$ and what is written as $\Sigma_{L_X;m} \Pi_{i=1}^{m} [g_i(L_X(i)]$ in (3), we can use (3) to write $\Sigma_{\{h(x)\};x\notin L_X} [\Pi_{x\notin L_X}[V'(h(x), x)] = \Pi_{x\notin L_X}[\Sigma_{y\in Y} V'(y, x)]$. Since $\Sigma_{y\in Y}[V'(y, x)] = 1$, this sum-of-a-product term just equals 1. This means that we have reduced the $[k(L_X, f(L_X))]^{-1}$ term in the denominator of (5) to $\Pi_{x\in L_X} [V'(f(x), x)]$.

Now assume that n >> m and $\pi(x)$ is uniform, so that just as in the appendix B we can here take m' = m in the sum over L_X giving P(h | f, m). Doing this allows us to replace $\Pi_{x\in L_X}[V'(f(x), x)]$ with $\Pi_{i=1}^{m} [V'(f(L_X(i)), L_X(i))]$. This in turn allows us to write the equality $P(h \mid f, m) = T'(h) \times [\, \Sigma_{x\in X} \{\delta(h(x), f(x)) \times \pi(x) / V'(f(x), x)\} \,]^m$. QED.

The formula for P(E | f, m) resulting from (D.1) will not, in general, be easily expressible as in equation (6.68) of [Hertz et al. 1991]. In particular, unless V(., .) is flat, you do not get the central result of the conventional analysis of exhaustive learning. The problem is the V' term in the denominator inside the sum. To try to follow along with the derivation of (6.68), instead of $\rho_0(E) \equiv \Sigma_h \{T(h) \times \delta(E, Er(f, h))\}$, you get $\rho_0(E, m) \equiv \Sigma_h \{T'(h) \times \delta(E, Er(f, h)) \times [\eta(h)]^m\}$, where $\eta(h) \equiv [\, \Sigma_{x\in X} \{\delta(h(x), f(x)) \times \pi(x) / V'(f(x), x)]\}] \,/\, [\, \Sigma_{x\in X} \{\delta(h(x), f(x)) \times \pi(x)\} \,]$ and varies with h.

Finally, it is interesting to note that we *could* say that (6.68) holds, but only if we redefine the error function so that it uses a different sampling distribution from the one used by the sampling assumption ($\{\pi(x) / V'(f(x), x)\}$ vs. $\{\pi(x)\}$). One is extremely hard-pressed however to justify such a change in the definition of the error function.

ACKNOWLEDGMENTS

This work was done under the auspices of the Department of Energy.

REFERENCES

Blumer, A., et al. (1987). Occam's razor. *Information Processing Letters*, **24**, 377-380.

Blumer, A., et al. (1989). Learnability and Vapnik-Chervonenkis dimension. *Journal of the ACM*, **36**, 929-965.

Buntine, W. , Weigend, A. (1991). Bayesian back-propagation. *Complex Systems*, **5**, 603-643.

Cohn, D., and Tesauro, G. (1991). How tight are the Vapnik-Chervonenkis bounds? University of Washington Department of Computer Science, TR 91-03-04.

Haussler, D., et al. (1991). Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. In *COLT '91*, Morgan Kauffman, San Mateo, CA.

Hertz J., et al. (1991). *Introduction to the theory of neural computation*. Addison-Wesley, NY, NY.

Schwartz et alia (1990). Exhaustive Learning. *Neural Computation*, **2**, 374-385.

Seung H., et al. (1991). Statistical mechanics of learning from examples I, II. Submitted.

Tishby et alia (1989). Consistent inference of probabilities in layered networks: Predictions and generalization. In *IJCNN International Joint Conference on Neural Networks*, Vol. II, 403-409, IEEE, New York.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM*, **27**, 1134-1142.

Van der Broeck, C., and Kawai, R. (1991). Generalization in feedforward neural and Boolean networks. To be presented at *AMSE International conference on neural networks*, San Diego, CA.

Wolpert, D. (1992a). On the connection between in-sample testing and generalization error. *Complex Systems*, **6**, 47-94.

Wolpert, D. (1992b). Filter sampling assumptions and exhaustive learning. To appear in the proceedings of the 1991 CLNL workshop.

Wolpert, D. (1992c). The limitations of PAC as a model of real-world generalization. In preparation.