A RIGOROUS INVESTIGATION OF EXHAUSTIVE LEARNING

by

David H. Wolpert*, Alan Lapedes**

* - Theoretical Division and Center for Non-linear Studies, Los Alamos National Lab. Currently at The Santa Fe Institute, 1660 Old Pecos Trail, Suite A, Santa Fe, NM, 87501. (dhw@santafe.edu)
** - Theoretical Division, MS B213, LANL, Los Alamos, NM, 87545 (asl@nets.lanl.gov)

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 work of Tishby et al. It is here proven that the crucial "self-averaging" assumption invoked in the analysis of Schwartz et al. does not hold in the simplest non-trivial implementation of exhaustive learning. Therefore their primary result, that generalization accuracy necessarily rises exponentially as training set size m is increased, is not generic. More importantly, it is proven here that a change in the definition of what one calls the "generalization accuracy", to reflect only the error on inputs outside of the training set, completely negates the result of Schwartz et al. even when self-averaging does hold. In other words, their 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, trivially, your generalization accuracy will either increase or stay the same. The important point is that although 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 it results in an exponential rise. We examine a number of other issues concerning exhaustive learning and its relation to real-world generalization as well.
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 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 Schwartz et al. is that in this scenario, $P(E \mid f, m) \sim (1 - E)^m$, 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 is an extension of conventional Bayesian analysis which was introduced in [Wolpert 1992]. This formalism doesn't restrict itself to a certain kind of generalizer (like exhaustive learning and the various versions of the "statistical mechanics" machine learning formalism do [Tishby et al. 1989, Seung et al. 1991]), nor does it restrict itself to finding worst-case bounds, where one pretends one knows very little about the generalizer (like PAC [Blumer et
al. 1987, Blumer et al. 1989, Valiant 1984), for example). Nor does it need to pretend 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 school). Nor does it assume one's "priors" are correct, like conventional
Bayesian analysis (see [Buntine and Weigend 1991 and references therein]). The extended Baye­
sian formalism is completely general. In fact, the previously mentioned formalisms are just special
cases of it.

Using extended Bayesian analysis we prove that the crucial "self-averaging" assumption in­
voked in the analysis of Schwartz et al. does not hold in the simplest non-trivial implementation of
exhaustive learning. Therefore their primary result, that generalization accuracy necessarily rises
exponentially as training set size m is increased, is not generic. (This might help clarify the results
of [Cohn and Tesauro 1991].)

More importantly, this paper proves that a change in the definition of what one calls the
"generalization accuracy", to reflect only the error on inputs outside of the training set, completely
negates the result of Schwartz et al. even when self-averaging does hold. In other words, their 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 in­
creased or stayed the same, and therefore, trivially, your generalization accuracy will either in­
crease or stay the same. The important point is that although this phenomenon would, taken naive­
ly, appear to only result in a linear rise in generalization accuracy with training set size, in the ex­
haustive learning scenario it results in an exponential rise. This serves as a caution to all who would
read 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, and that the foundational theorem of PAC,
\[\delta \leq (1 - \epsilon)^m \times \log'\] ([Blumer et al. 1987]), bears a very close similarity to the central result of ex­
haustive learning.)

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 analysis of Schwartz et al. for the case which interested them, 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. Finally, section IV discusses several miscellaneous aspects of exhaustive learning.

In addition to these issues of self-averaging and on- versus off- training set errors, this paper also examines a number of other issues concerning exhaustive learning and its relation to real-world generalization. A summary of the results of this paper can be found in the conclusion.

\section{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].

In exhaustive learning, we have a finite discrete input space $X$ consisting of $n$ elements. 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 $\Theta$, and consists of $m$ input/output pairs. The set of all input values $x_i$ in the pairs making up $\Theta$ will be indicated by $\Theta_X$. The corresponding output values will be indicated by $\Theta_Y$. For generality, training sets are assumed ordered; symmetry under permutation of the pairs making up a training set can be imposed afterwards, if desired.
The vast majority of supervised machine learning issues implicitly concern an event space consisting of the set of all triples \( \{f \in F, h \in H, \theta\} \). (Even if for some reason we wish to restrict the set of allowed hypothesis functions to a "concept class", we can still use the full event space \( \{f \in F, h \in H, \theta\} \); simply set to 0 all probabilities \( P(f, h, \theta) \) 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 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 \( \{f, h, \theta_X, \theta_Y\} \) for the equivalent triple \( \{f, h, \theta = \{\theta_X, \theta_Y\}\} \); the reader is warned that we will interchange one notation for the other without any warning. (As a general rule, the use of such quadruples is helpful whenever rather than the training set \( \textit{set in toto} \) only the size of the training set is specified in the conditional probability being calculated. See the following section of this paper.)

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 \( \{f, h, \theta\} \) space. Conventional "Bayesians" don't distinguish hypothesis functions from target functions, and implicitly use an event space \( \{f, \theta\} \).\(^1\) In fact, what a Bayesian refers to as a "hypothesis" is what is referred to in this paper as a target function; there is nothing in conventional Bayesian analysis which is analogous to what is here being called hypothesis functions. (In addition to making Bayesian analysis unsuited for investigating exhaustive learning, this also means that Bayesians can not analyze situations where the prior one assumes is different from the actual one describing the physical universe. See [Wolpert 1992a].)

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 \( P(h \mid \theta) \). For example, if the generalizer is deterministic, it always outputs the same hypothesis function when taught with the same training set, and for fixed \( \theta \), \( P(h \mid \theta) \) is a delta function over \( H \). If instead the generalizer is stochastic, then for a given \( \theta \) \( P(h \mid \theta) \) 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 \( \theta \), throw away all \( h \in H \) which do not go through the elements of \( \theta \) (i.e., throw away all \( h \) except those \( h \) such that the relation \( \theta \subset h \) 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 \( \theta \) and then guesses randomly, according to the distribution \( T(h) \).

More formally, in exhaustive learning

\[
P(h | \theta) \propto T(h) \times \delta(\theta \subset h),
\]

where the delta function equals 1 if the set relation is true, 0 otherwise. The proportionality constant is set by normalization and depends on \( \theta \). It is written as \([k(\theta_X, \theta_Y)] = \Sigma_h T(h) \times \delta(\theta \subset h)^{-1}\), and is implicitly assumed to be non-zero. Below it will be useful to introduce the count function \( S(g_1, g_2, \{x_i\}) = \Sigma_i \{\delta(g_1(x_i), g_2(x_i))\} \) 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 \( \theta \) (\( m' \leq m \)), for exhaustive learning \( P(h | \theta) = k(\theta_X, \theta_Y) \times T(h) \times \delta(S(\theta, h, \theta_X), m') \)

with \([k(\theta_X, \theta_Y)]^{-1} = \Sigma_h T(h) \times \delta(S(\theta, h, \theta_X), m')\).

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 \( P(h) = \Sigma_\theta P(h | \theta) \times P(\theta) = T(h) \times \Sigma_\theta \{k(\theta_X, \theta_Y) \times \delta(\theta \subset h) \times P(\theta)\} \) and in general the sum depends on \( h \).

In addition to fixing the generalizer, exhaustive learning also fixes the sampling assumption, (sometimes called the "likelihood function"), \( P(\theta | f) \). 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 $\theta_X$ (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 $\theta_X$.

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 $P(\theta \mid f) = 0$ unless $\theta$ agrees with $f$, i.e., unless $\theta \subset f$. (Note that this requirement is formally equivalent to setting to 0 all $P(f, h, \Theta)$ for which $\theta \not\subset f$.) In addition, the elements of $\theta_X$ are selected in an i.i.d. (independent identically distributed) manner:

$$
\frac{P(\theta \mid f)}{P(\theta' \mid f)} = \frac{\prod_{i=1}^{m} \pi(\theta_X(i))}{\prod_{i=1}^{m} \pi(\theta'_X(i))}
$$

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

Note that in exhaustive learning repeats are allowed in $\theta$, i.e., the same input value can occur more than once in $\theta_X$. There are two immediate ramifications of this. First, this means that conditions like “$q \subset f$” are being interpreted a bit loosely in this paper; formally, they should be taken to mean that $\theta$ is consistent with $f$, rather than that the set of input-output pairs $(\theta_X(i), \theta_Y(i); 1 \leq i \leq m)$ is 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 $\theta$: anytime an input value is repeated, the corresponding output values must agree. This means that any such repeat in $\theta$ 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 $\theta$, which is directly related to “the amount of information” in $\theta$. 
In addition to probabilities over \( \{f \in F, h \in H, \theta\} \), one must relate \( f \) and \( h \) to a real-world "cost". This is conventionally done through an error function, which is a function over the event space, \( \mathcal{E} \). Schwartz et al. implicitly use an error function independent of \( \theta \) but dependent on \( \pi(x) \):

\[
\mathcal{E}(f, h) = \sum_{x \in \mathcal{X}} \pi(x) [1 - \delta(f(x), h(x))].
\]

This is the same error function used in PAC, the "statistical mechanical school", etc. As is discussed later, although in many circumstances it entails "distribution-free" results (i.e., results independent of \( \pi(x) \)), this error function also has a number of undesirable features.

There is one final fact which we shall need to perform our calculations. In all supervised machine learning, exhaustive learning included, it is implicitly assumed that the generalizer only has access to the training set, and cannot 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 \( P(h \mid f, \theta) \) must be independent of \( f \), which in turn means that \( P(h \mid f, \theta) = P(h \mid \theta) \) (which in turn means that \( P(f \mid h, \theta) = P(f \mid \theta) \) and \( P(h, f \mid \theta) = P(h \mid \theta) \times P(f \mid \theta) \); see [Wolpert 1992a].) Like the assumption concerning \( P(\theta_X \mid f) \) (see footnote [3]), it turns out that this assumption concerning \( P(h \mid f, \theta) \) is used in almost all machine learning research, though it is never explicitly acknowledged.

Although Schwartz et al. couch exhaustive learning in terms of neural nets, volumes in weight space, etc., none of that machinery is necessary, and is in fact obfuscatory. The exposition given above defines exhaustive learning exactly, and in a more general manner than the formalism used by Schwartz et al. Moreover, as is illustrated in the following sections, it is only by couching exhaustive learning this way in terms of the extended Bayesian formalism that one can address all the issues discussed in this paper (see the conclusion). In particular, in addition to the formalism used by Schwartz et al., neither PAC, conventional Bayesian analysis, nor the "statistical mechanical school" are able to address all of these issues. This is a simple illustration of the fact that those formalisms are all special cases of the extended Bayesian formalism.
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 analysis in [Schwartz et al. 1990] without assuming (as they do) that one is using a neural net generalizer. This section also begins the exposition of how to use those tools to go beyond the analysis in [Schwartz et al. 1990].

In going through this section the reader should note one of the advantages of using the extended Bayesian formalism: Using that formalism, machine learning is reduced to a two-stage process. 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".

In this paper, we will explicitly carry out this two-stage process. Most of the first stage was carried out in the previous section. The rest of it will be completed by equation (1) below. After that, the rest of this paper is a completion of the second stage.

The following discussion will make reference to the version of exhaustive learning presented in [Hertz et al. 1991]. (The original paper by Schwartz et al. has some minor errors and in some ways is less clear than the version 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". Similarly, 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 \( \tilde{f} \) 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 men-
tioned in the previous section we here use the “exhaustive learning error function” \( E_r(f, h) \), which is linearly related to \( g(.) \).

The paper by Schwartz et al. is concerned with evaluating \( \langle P(\text{error} = E \mid \text{target function} = f, \text{training set} = \theta) \rangle \), the expectation value being over all possible training sets \( \theta \) of size \( m \) chosen from the (fixed but unknown) target function \( f \). 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.

We have now completely the first stage of machine learning. As was mentioned above, from here on out there is no need to think; one just turns the calculational crank. Doing this we will derive the results of Schwartz et al. We start with the following:

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

Proof: Writing it out, \( \langle P(E \mid f, \theta) \rangle = \sum_{\theta \subset f, m} \{ P(E \mid f, \theta) \times P(\theta \mid f, m) \} \) where \( \subset m \) 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 \( P(\theta \mid f, m) = P(\theta, f, m) / P(f, m) = P(\theta, f) / P(f, m) \). Therefore \( \langle P(E \mid f, \theta) \rangle = \sum_{\theta \subset f, m} \{ P(E, f, \theta) / P(f, m) \} = P(E, f, m) / P(f, m) = P(\text{error} = E \mid \text{target function} = f, \text{training set has size } m) \). QED.

Expanding in terms of the triples making up our event space, \( P(\text{error} = E \mid \text{target function} = f, \text{training set has size } m) = P(E \mid f, m) = \{ \sum_{h} \sum_{\theta \subset f, m} P(f, h, \theta) \delta(E, E_r(f, h)) \} / \{ P(f, m) \} \) (where this delta is the Kronecker delta function, and the meaning of \( \theta \subset f, m \) is given in the proof of (1)). This can be rewritten as

(2) \( P(E \mid f, m) = \sum_{h} P(h \mid f, m) \delta(E, E_r(f, h)) \).
The analogous formula in Schwartz et al.'s analysis is \( p_p(g) = \sum_f P_p(f) \delta(g - g(f)) \) from equation (6.67) in [Hertz et al. 1991]. (N.b. Schwartz et al. use probability densities, i.e., implicitly viewing errors as taking on continuous values, whereas in this paper we use straight probabilities, implicitly viewing errors as taking on only one of a finite set of possible values.) 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:

\[
\sum_{\mathbf{x} \in \mathcal{X}} \prod_{i=1}^{m} [g_i(\theta_{\mathbf{x}}(i))] = \prod_{i=1}^{m} \sum_{\mathbf{x} \in \mathcal{X}} g_i(x), \text{ for any function } g(.).
\]

Proof: We can prove the lemma using induction over \( m \). To base the induction, note that the formula obviously holds when \( m = 1 \):

\[
\sum_{\mathbf{x} \in \mathcal{X}} \prod_{i=1}^{1} [g_i(\theta_{\mathbf{x}}(i))] = \sum_{\mathbf{x} \in \mathcal{X}} [g_1(\theta_{\mathbf{x}}(1))] = \sum_{\mathbf{x} \in \mathcal{X}} g_1(x).
\]

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

\[
\sum_{\mathbf{x} \in \mathcal{X}} \prod_{i=1}^{c+1} [g_i(\theta_{\mathbf{x}}(i))] = \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\theta_{\mathbf{x}} : \mathcal{X}} \prod_{i=1}^{c} [g_i(\theta_{\mathbf{x}}(i))] [g_{c+1}(x)],
\]

which equals

\[
\sum_{\mathbf{x} \in \mathcal{X}} [g_{c+1}(x) \times \prod_{i=1}^{c} [g_i(\theta_{\mathbf{x}}(i))]],
\]

by the inductive hypothesis. This last expression can be rewritten as

\[
[\sum_{\mathbf{x} \in \mathcal{X}} g_{c+1}(x)] \times \prod_{i=1}^{c} [\sum_{x' \in \mathcal{X}} g_i(x')], \text{ which equals } \prod_{i=1}^{c+1} [\sum_{x' \in \mathcal{X}} g_i(x')],
\]

which completes the proof. QED.

As was mentioned previously, it will sometimes be convenient to substitute the space of qua-
druples \{f, h, \theta_X, \theta_Y\} for the equivalent space of triples \{f, h, \theta = \{\theta_X, \theta_Y\}\}. Along with exploiting (3), it is necessary to make this kind of substitution to evaluate \(P(h \mid f, m)\):

\[
P(h \mid f, m) = \sum_{\theta \in C} P(h \mid \theta) \times \prod_{i=1}^{m} [\pi(\theta_X(i))].
\]  

Proof: 

\[
P(h \mid f, m) = \{\sum_{\theta \in C} P(f, h, \theta)\} / \{\sum_{\theta' \in C} P(f, h, \theta')\}
\]

\[
= \{\sum_{\theta \in C} P(h \mid \theta) \times P(f, \theta)\} / \{\sum_{\theta' \in C} P(f, \theta')\}
\]

(since \(P(h \mid f, \theta) = P(h \mid \theta)\))

\[
= \{\sum_{\theta \in C} P(h \mid \theta) \times P(\theta \mid f)\} / \{\sum_{\theta' \in C} P(\theta' \mid f)\}
\]

\[
= \{\sum_{\theta \in C} P(h \mid \theta) \times \prod_{i=1}^{m} [\pi(\theta_X(i))]\} / \{\sum_{\theta' \in C} \prod_{i=1}^{m} \pi[\theta_X(i)]\}
\]

(due to the exhaustive learning sampling assumption).

Now rewrite \(\sum_{\theta \in C} \prod_{i=1}^{m} [\pi(\theta_X(i))] = \sum_{\theta_X \in X} \prod_{i=1}^{m} [\pi(\theta_X(i))] = [\sum_{x \in X} \pi(x)]^m = [1]^m = 1\).

(In writing down the second equality, use has been made of the fact that \(\sum_{\theta_X \in X} \prod_{i=1}^{m} [g(\theta_X(i))] = [\sum_{x \in X} g(x)]^m\) 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 \(\sum_{\theta \in C} P(h \mid \theta) \times \prod_{i=1}^{m} [\pi(\theta_X(i))]\) as \(< P(h \mid \theta) >\) (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 \(P(h \mid \theta)\) into equation (4). After doing this we will plug the resultant expression for \(P(h \mid f, m)\) into equation (2).
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): \[ P(E \mid m) = \sum_{f \in F} (P(E \mid f, m) \times P(f \mid m)) = \sum_{f \in F, \theta; m} (P(E \mid f, m) \times P(f \mid \theta) \times P(\theta)) / \sum_{\theta; m} P(\theta). \]

In [Hertz et al. 1991], this "central result" is presented as equation (6.68). It is the main result of the analysis of Schwartz et al. and is taken to imply that as $m$ increases, lower $E$'s become more likely. In other words, it is taken as an argument that "inductive inference works", if one uses the exhaustive learning generalizer. (Though note that if $m$ increases too much, then it's no longer true that $n \gg m$, and Schwartz et al. shouldn't expect their result to apply.) In fact, since for generic $\rho_0(E)$ one would expect $\{\sum E [1 - E']^m \times \rho_0(E')\}^{-1}$ to be a low-order polynomial in $m$ (since $\int_0^1 dE E^m = (m + 1)^{-1}$), when self-averaging holds $P(E \mid f, m)$ should depend on $m$ exponentially.

This is essentially as far as one can get using the framework of Schwartz et al. One can proceed further however, by using the extended Bayesian formalism. In particular, using that formalism one can directly test whether or not self-averaging holds (in the limit of $n \gg m$ or otherwise) for various $T(h)$. For example, using that formalism one can prove the following:

(6) One special case where self-averaging does hold in the limit of $n \gg m$ is where $T(h)$ is independent of $h$.

Proof: First recall that $P(h \mid f, m) = \sum_{\theta \in F; m} (P(h \mid \theta) \times \Pi_{i=1}^m [\pi(\theta_{X(i)})])$. In the limit of $n \gg m$, the vast majority of terms in the sum will have $m' = m$. Now if $T(h)$ is constant, then for no $h$ will $P(h \mid \theta)$ be vastly larger when $\theta$ has a repeat in it than when it doesn't. This means that we can throw away all terms in the sum for which $m' \neq m$. Now recall that $k(\theta_{X'}, f(\theta_{X'})) = [\Sigma_h T(h) \times \delta(S(f(\theta_{X}), h, \theta_{X'}, m'))]^{-1}$. Since we can set $m' = m$ and since $T(h)$ is a constant, this equals $[T(h) \times \Sigma_h \delta(S(f(\theta_{X}), h, \theta_{X'}, m))]^{-1}$. The sum is just the number of hypothesis functions which go through
all m distinct elements of \( \theta \). This number is independent of the locations in \( \theta_X \). Therefore \( k(\theta_X, f(\theta_X)) \) is independent of \( \theta_X \), and as mentioned before this condition is sufficient for self-averaging. QED.

Note that 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.) It turns out that it isn’t necessary to use the formula of Schwartz et al. in this \( T(h) \)-independent-of-\( h \) case however; by using the extended Bayesian formalism, we can solve for \( P(h | f, m) \) (and therefore for \( P(E | f, m) \)) 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 A:

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

\[
P(E | f, m) \propto \left( \sum_{i=1}^{\min(m, n(1 - E))} [r(i - n) \times \chi(m, i) \times \frac{C_i^{n(1-E)}}{n^{n(1-E)}}] \times C_n^{n(1-E)} \times (r - 1)^n E,\]

where \( \chi(., .) \) is a combinatoric function defined in appendix A.

In addition to allowing exact calculations when \( n \) is not >> \( m \), the extended Bayesian formalism also allows the straight-forward evaluation of \( P(h | f, m) \) (and therefore of \( P(E | f, m) \)) in a number of situations where \( k(\theta_X, f(\theta_X)) \) is not independent of \( \theta_X \), so that self-averaging does not apply. In appendix B it is shown that what is perhaps the simplest such case explicitly results in a different result than the self-averaging-based result of Schwartz et al. (This fact alone establishes that the result of their paper is not generic.) In other words, contrary to their claims, 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 \( P(E | f, m) \) is a direct reflection of “whether or not inductive inference works” (a claim which is disputed below), the analysis of
Schwartz et al. does not prove that inductive inference works with their exhaustive learning generalizer.

One peculiar aspect of exhaustive learning is that the probability it's interested in is $P(E | f, m)$ rather than $P(E | f, \theta)$. 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 $P(E | f, m)$ rather than $P(E | f, \theta)$, Schwartz et al. are pretending that we know the size of the training set but not the actual elements making it up. Another drawback to concentrating on $P(E | f, m)$ rather than $P(E | f, \theta)$ is that it is always possible to calculate $P(E | f, m)$ from $P(E | f, \theta)$ (just use (1)), but the reverse does not hold.

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

$$P(E | f, \theta) = \sum_{h} P(h | f, \theta) \delta(Er(f, h), E)$$

From (1),

$$= \sum_{h} h \delta(Er(f, h), E) \delta(S(\theta, h, \theta_{X', m'}), m') / \{\sum_{h} T(h) \delta(S(\theta, h, \theta_{X}, m'))\}$$

Under the condition $h \supset \theta$, the uniformity of $\pi(x)$ allows us to rewrite the error function $Er(f, h)$ as $[n - m' - S(f, h, X - \theta_{X'})] / n$. We can then use the uniformity of $T(h)$ to cancel it out from the numerator and the denominator: $P(E | f, \theta) = \{\sum_{h \supset \theta} \delta(S(f, h, X \theta_{X'}), n(1 - E) - m')\} / \{\sum_{h \supset \theta} 1\}$. 

\[ P(E | f, \theta) = C_{z}^{(n-m')} \times (r-1)^{(n-m'-z)} / r^{(n-m')}, \]

where $z = [n - m' - nE]$ is the number of agreements between $f$ and a hypothesis function for questions outside of the training set.
Note that the value of $P(E \mid f, \theta)$ depends only on the $m'$, the number of distinct elements in the training set, and not on the particular training set chosen. In other words, for this simple scenario $P(E \mid f, \theta)$ is independent of $\theta$, depending only on the number of distinct elements in $\theta$. 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 $P(E \mid f, m)$. 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 $P(E \mid f, \theta)$ can be vastly different from the form of $P(E \mid f, m)$, even when (as here) $P(E \mid f, \theta)$, like $P(E \mid f, m)$, does not depend on the actual pairs making up $\theta$. 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 $P(E \mid f, \theta)$ is independent of $m$ and depends only on $m'$.

This distinction between $P(E \mid f, \theta)$ 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 $n \gg m$ where we can take $m = m'$ (see the proof of (6)). This is indeed the case:

(9) When $n \gg m$ and both $T(h)$ and $\pi(x)$ are constant,

$$P(E \mid f, \theta) \propto (1 - E)^m \times \rho_0(E).$$

Proof: First rewrite (8) as $P(E \mid f, \theta) = \binom{n-m'}{n-m' - nE} \times (r - 1)^{nE} / r^{n-m'}$. This means that the ratio
\[ P(E \mid h, \theta') / P(E \mid h, \theta), \] where \( \theta \) is a training set of \( m' \) distinct elements and \( \theta' \) is a training set of \( m' + 1 \) distinct elements, equals \[ r \times \frac{C_z^{(n-m'-1)}}{C_z^{(n-m')}} \] \( z' \) being the \( z \) value for the training set \( \theta' \), which has \( m' + 1 \) distinct elements; \( z' = z - 1 \). \( \text{Note that it is being assumed that for both } \theta \text{ and } \theta' \text{ it's possible to have error } E, \text{ i.e., it's being assumed that both } E < (n - m') / n \text{ and } E < (n - m' - 1) / n. \) 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, \( P(E \mid f, \theta) = r^{m'} \times [1 - E]^{m'} \times \omega(E) \), for some function \( \omega(.) \). Now for \( n >> m \), the probability that \( \theta \) has a repeat in it is vanishingly small. Therefore we can replace \( m' \) with \( m; P(E \mid f, \theta) \propto [1 - E]^m \times \omega(E) \). Now using (1) and the fact that \( \pi(x) \) is being assumed constant, for \( m = m' = 1 \), \( P(E \mid f, m) \) must equal \( P(E \mid f, \theta) \) for any \( \theta \) 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 \( \theta_X \)). 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 \( \theta \), 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. Note that these arguments are implicitly acknowledged in the physical sciences, where one judges between two theories which both explain a set of old experimental data by conducting completely new experiments, with different physical conditions from the old experiments.

There is another, more concrete problem with the exhaustive learning error function. This 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 off-training-set error function, \( Er(h, f, \theta) = \sum_{x \in X} \theta_x \pi(x) \left[ 1 - \delta(f(x), h(x)) \right] / \sum_{x \in X} \theta_x \pi(x) \). This error function will always have the same lower and upper bound, regardless of \( m' \). Moreover, it directly measures off-
training set error. (To directly measure on-training set error is trivial; there never are any on-training set errors.) Note that for the exhaustive learning generalizer, \( \sum_{x \in \emptyset_X} \pi(x) [1 - \delta(f(x), h(x))] = \sum_{x} \pi(x) [1 - \delta(f(x), h(x))] \), so although written as an off-training set error function, \( \text{Er}(f, h, \theta) \) is simply exhaustive learning's \( \text{Er}(f, h) \) rescaled.

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

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

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

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

Proof: First continue along with the proof of (8). We get

\[
P(E \mid f, \theta) = \left\{ \sum_{h \in \emptyset} \delta(\text{Er}(h, f, \theta), E) \right\} / \left\{ \sum_{h \in \emptyset} 1 \right\}
\]

which holds independent of the error function used. Whereas in the proof of (8) we next replaced the error function with the expression \([n - m' - S(f, h, X - \emptyset_X)] / [n - m] \), our new error function must instead be replaced by \([n - m' - S(f, h, X - \emptyset_X)] / [n - m] \). Plugging this in, exactly as in the proof of (8), gives

\[
P(E \mid f, \theta) = \left\{ \sum_{h \in \emptyset} \delta(S(f, h, X - \emptyset_X), (1 - E)(n - m')) \right\} / \left\{ \sum_{h \in \emptyset} 1 \right\}
\]

Carrying through the combinatorics gives the same result as in (8), \( P(E \mid f, \theta) = \binom{(n-m')}{z} \times (r - 1)^{(n-m')z} / r^{n-m'} \), except that here \( z = (1 - E)(n - m') \) rather than \( (1 - E)n - m' \). Now we must follow along with the proof of (9). First rewrite our formula for \( P(E \mid f, \theta) \) as \( \binom{n-m'}{(1-E)(n-m')} \times (r - 1)^{(n-m')E} / r^{n-m'} \). Now we want to calculate the ratio \( P(E \mid h, \theta') / P(E \mid h, \theta) \), where \( \theta \) is a training set of \( m' \) distinct elements and \( \theta' \) is a training set of \( m' + 1 \) distinct elements. Our answer is

\[
r \times (r - 1)^E \times [(n - m')]^{-1} \times \left\{ [(n - m' - En + Em')! / (n - m' - 1 - En + Em' + E)!] \times [(nE - m'E)! / (nE - m'E + E)!] \right\}.\]

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 \( X - \theta_X \), 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 \( P(E \mid f, \theta) \) (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 \( E \), \( (n - m' - 1 - En + Em' + E) \equiv (n - m' - En + Em') + 1 \). This means our first ratio of factorials \( \equiv (n - m' - En + Em') = (n - m') \times (1 - E) \). Similarly, our second ratio of factorials \( \equiv 1 \) for low \( E \). Therefore for low \( E \), \( P(E \mid f, \theta) = r^m \times (r - 1)^{-Em'} \times (1 - E)^{m'} \times \omega(E) \) for some function \( \omega(\cdot) \). This holds regardless of the relative sizes of \( n \) and \( m \). (As usual, for \( n \gg m \), 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 \( P(E \mid f, \theta) = r^m \times (r - 1)^{-Em'} \times (E)^{m'} \times \omega'(E) \) for some function \( \omega'(\cdot) \). QED.

In particular, for \( r = 2 \) and \( E \) large, \( P(E \mid f, \theta) \propto (E)^m \times \omega'(E) \). 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 \((1 - E)^m\) generalization behavior of exhaustive learning not only disappears, it is actually reversed. Phrased differently, \((1 - E)^m\) generalization behavior is not nearly 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 calculation 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 both uses the exhaustive learning error function and relies heavily on exponential vs. non-exponential behavior as way of distinguishing “good” generalizers from “bad” generalizers. (See [Wolpert 1992c].)

IV. DISCUSSION

The idea behind calculating $P(E \mid f, \text{training set has size } m)$ 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 $P(E \mid f, m)$. More formally,

11) Assume that for all training sets $\theta$,

$$P(\text{training set } = \theta \mid f, \text{the training set has size } m) =$$

$$P(\text{first } m \text{ pairs in the training set } = \theta \mid f, \text{the training set has size } m + 1).$$
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
\[
P(E \mid f, m+1) - P(E \mid f, m).
\]

Proof: Let \( \omega \) indicate a generic training set, and let \( \omega = \{ \theta \}^+ \) mean that we have a training set, size \( m + 1 \), whose first \( m \) pairs are given by \( \theta \) (nothing else is known about the training set). The quantity we want to calculate is \( \sum_{\theta; m} P(\theta \mid f, m) \times [P(E \mid f, \omega = \{ \theta \}^+) - P(E \mid f, \theta)] \). Expanding \( P(E \mid f, \omega = \{ \theta \}^+) \), we get \( \sum_{\theta' = \{ \theta \}^+} P(E \mid f, \theta') / \sum_{\theta' = \{ \theta \}^+} P(f, \theta') = [\sum_{\theta' = \{ \theta \}^+} P(E \mid f, \theta') \times P(\theta' \mid f)] / [\sum_{\theta' = \{ \theta \}^+} P(\theta' \mid f)] \). We can rewrite this as \( [\sum_{\theta' = \{ \theta \}^+} P(E \mid f, \theta') \times P(\theta' \mid f, \text{the size of } \theta' = m+1)] / [\sum_{\theta' = \{ \theta \}^+} P(\theta' \mid f, \text{the size of } \theta' = m+1)] \). Dividing top and bottom by \( P(f, \text{the size of } \theta' = m+1) \), we get \( [\sum_{\theta' = \{ \theta \}^+} P(E \mid f, \theta') \times P(\theta' \mid f, \text{the size of } \theta' = m+1)] / [\sum_{\theta' = \{ \theta \}^+} P(\theta' \mid f, \text{the size of } \theta' = m+1)] \).

The denominator can be rewritten as \( P(\omega = \{ \theta \}^+ \mid f, \text{the size of } \omega = m+1) \). Under our assumption concerning probabilities of training sets, this is \( P(\text{training set } = 0 \mid f, \text{the training set has size } m) = P(\theta \mid f, m) \). Therefore the quantity we want to calculate is \( \sum_{\theta; m} [\sum_{\theta' = \{ \theta \}^+} P(E \mid f, \theta') \times P(\theta' \mid f, \text{the size of } \theta' = m+1)] - \sum_{\theta; m} P(E \mid f, \theta) \times P(\theta \mid f, m) \). Because training sets are ordered, we can rewrite the double sum as \( \sum_{\theta; m+1} \text{ giving us } \sum_{\theta; m+1} P(E \mid f, \theta') \times P(\theta' \mid f, m+1) - \sum_{\theta; m} P(E \mid f, \theta) \times P(\theta \mid f, m) \). 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 \( P(\theta_Y \mid f, \theta_X, m) = \prod_{i=1}^{m} t[f(\theta_X(i)), \theta_Y(i)] \) for some function \( t(\cdot, \cdot) \) obeying \( \Sigma_y t(z, y) = 1 \forall z \in Y \) (as when we have Gaussian noise), and assume further that \( P(\theta_X \mid f, m) = \prod_{i=1}^{m} \pi(\theta_X(i)) \) for some function \( \pi(\cdot) \) obeying \( \Sigma_x \pi(x) = 1 \) (as in i.i.d. sampling). Now it is always true that \( P(\theta \mid f, m) = P(\theta_Y \mid f, \theta_X, m) \times P(\theta_X \mid f, m) \). Therefore, given the assumptions for \( P(\theta_Y \mid f, \theta_X, m) \) and \( P(\theta_X \mid f, m) \), the probability
P(first $m$ pairs in the training set = $\theta \mid f$, the training set has size $m + 1$) just equals $\Sigma_{x,y} P($training set $= \{\theta, (x, y)\} \mid f, m+1) = \Sigma_{x,y} [\prod_{i=1}^{m} t[f(\theta_X(i)), \theta_Y(i)] \times \pi(\theta_X(i))] \times t(f(x), y) \times \pi(x)] = \Pi_{i=1}^{m} t[f(\theta_X(i)), \theta_Y(i)] \times \pi(\theta_X(i))$. This last term equals $P(\theta \mid f, m)$ 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 $P(E \mid h, \theta)$ 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 $P(E \mid h, \theta)$ under the assumption of a "maximum-entropy universe".5

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,

(12) 

$$P(E \mid h, m) = \frac{[1 - E]^m \times p_0(E)}{\Sigma_{E'} [1 - E']^m \times p_0(E')}$$

(where $p_0(E)$ 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 $P(f \mid h, \theta) = P(f \mid \theta)$ as well as $P(h \mid f, \theta) = P(h \mid \theta)$. In addition, the error function in exhaustive learning is symmetric under $f \leftrightarrow h$. Therefore, since $P(h \mid \theta)$ and
P(θ | f) are the only remaining quantities which are specified in exhaustive learning, we only have to prove that the both the pair \{P(f | θ) and P(h | θ)\} and the pair \{P(θ | f) and P(θ | h)\} are inter-changeable 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(θ_X | f) is independent of f and there is no noise in the sampling (both of which conditions hold in exhaustive learning), P(f | θ) ∝ P(f) × δ(θ ⊂ f), where the proportionality constant depends on θ as usual. This establishes the calculational symmetry under f ↔ h of the pair \{P(f | θ) and P(h | θ)\}. It is also proven in [Wolpert 1992b] that if P(h | θ) ∝ P(h) × δ(θ ⊂ h) (as in exhaustive learning, if T(h) = P(h)), then P(θ_X | h) = P(θ) × k(θ), where k(.) is the usual normalizing constant [Σ_h T(h) δ(θ ⊂ h)]^{-1} = [Σ_h P(h) δ(θ ⊂ h)]^{-1}. Now P(θ) = Σ_f P(θ | f) × P(f), and it is proven in [Wolpert 1992b] that P(θ | f) = P(θ_X | θ_X, f) × P(θ_X | f). Given our sampling assumption, this means that P(θ) = Σ_f \{κ × [Π_{i=1}^m π(θ_X(i))] × δ(θ ⊂ f) × P(f)\} for some constant κ, which in turn equals \{κ × [Π_{i=1}^m π(θ_X(i))]\} × Σ_f \{P(f) × δ(θ ⊂ f)\}. But by assumption P(h) = P(f)_{f=h}, which means that Σ_h P(θ) δ(θ ⊂ h) = Σ_f \{P(f) × δ(θ ⊂ f)\}. Therefore P(θ_X | h) = \{κ × [Π_{i=1}^m π(θ_X(i))]\}, and P(θ | h) is the same function of h as P(θ | 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 scenario is as follows: We conduct many learning experiments, with target functions chosen according to the probability distribution P(f). Look at all those cases where the training set has size m and 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.
CONCLUSION

This paper does the following:
1) It rederivesthemathematicsofexhaustivelearninginacontextmoregeneralthanthatofneural
nets.
2) It proves that if $T(h)$ (the generalizer's initial "bias") is independent of $h$, then as Schwartz et al.
claim, "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.
3) It showshowtocalculate things exactly, without recourse to limits, for the $T(h)$ constant case
mentioned in (2).
4) It showsthatself-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.
5) Exhaustivelearning calculates $P(\text{error value} = E \mid \text{target function } f, \text{training set size } m)$. This paper shows how to calculate
$P(\text{error value} = E \mid \text{target function } f, \text{training set } \theta)$, where $\theta$ has $m$ elements. The result differs from
$P(\text{error value} = E \mid \text{target function } f, \text{training set size } m)$, even if
$P(\text{error value} = E \mid \text{target function } f, \text{training set size } m)$ is independent of what $m$ input-output pairs
make up $\theta$. However in the limit of $n \gg m$, the two probability distributions agree.
6) 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 generali-
zation 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, the mathematics for questions
from within the training set is trivial (for the noise-free scenario of exhaustive learning), 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.

This paper proves that with an off-training set error function the exponential improvement in generalization goes away. Indeed, for the limit of large errors, when the output space is binary the generalization gets worse exponentially with m.

7) Regardless of what error function one uses, usually what one is ultimately interested in examining something like \{the probability of error for fixed f and m\} is how generalization will change, on average, as I add new elements to a training set. In general, this need not be given by the quantity calculated in exhaustive learning, P(E | f, m). This paper finds a sufficient condition for when it is given P(E | f, m).

8) Exploiting the symmetry of the extended Bayesian formalism under f ↔ h, this paper derives formulas for P(E | hypothesis function h, m) directly from the exhaustive learning formulas for P(E | target function f, m).

FOOTNOTES

1. Strictly speaking, Bayesians actually use event spaces of the form \{f, θ, 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.

2. It turns out that if T(h) = P(θ)_{θ=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) ∝ P(θ)_{θ=h}). In general however, when T(h) ≠ P(θ)_{θ=h}, the sum varies with h. See [Wolpert 1992b] for a discussion of this issue and its implication for what priors “really mean”. 
3. Note that in exhaustive learning, \( P(\theta_X \mid f) \) 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].

4. Note that since it does not distinguish \( H \) from \( F \), conventional Bayesian analysis cannot, in a natural manner, investigate issues related to this error function. This, implicitly, is why no Bayesian analysis appears in [Schwartz et al. 1990].

5. There is one small difference. In the expression in [Wolpert 1992a] for \( P(E \mid h, \theta) \), \( z = (n - m) \times (1 - E) \) 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

Calculating \( P(E \mid f, m) \) 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

\[
P(h \mid f, m) \propto (1/n)^m \times \sum_{\theta_X \cdot m} \{ \delta(S(f, h, \theta_X), m') / \sum_h \delta(S(f, h, \theta_X), m') \}
\]

\[
\propto \sum_{\theta_X \cdot m} \{ \delta(h(.)) \text{ agrees with } f(.) \text{ for all } X \text{ values in } \theta_X \} / \]
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:

\[
\sum_{m'=1}^{m} \{ r^{(m'-n)} \times \sum_{\theta_X : m, m'} \{ \delta(h(.) \text{ agrees with } f(.) \text{ for all } X \text{ values in } \theta_X} \},
\]

where by \( \sum_{\theta_X : m, m'} \) is meant the sum over all \( m \)-element \( \theta_X \) where the \( m \theta_X(i) \) take on only \( m' \) distinct values.

We must now evaluate \( \sum_{\theta_X : m, m'} \{ \delta(h(.) \text{ agrees with } f(.) \text{ for all } X \text{ values in } \theta_X} \). To do this, define \( z = n(1 - \text{Er}(f, h)) = S(f, h, X) \). 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, so that all of those \( X \) values 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 (A.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 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 \( \chi(m, m') \). We have just proven that

\[
P(h \mid f, m) \propto \sum_{m'=1}^{\min(m, n[1-\text{Er}(f,h)])} \{ r^{(m'-n)} \times C^z_{m'} \times \chi(m, m') \}.
\]

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 $n$,

\[
\begin{align*}
(A.2) \quad P(E \mid f, m) & = \left\{ \sum_{i=1}^{\min(m, n(1 - E))} \left[ r^{i-n} \times \chi(m, i) \times C_i^{n(1-E)} \right] \times \{ \sum_n \delta(E, Er(f, h)) \} \right\} \\
& = \left\{ \sum_{i=1}^{\min(m, n(1 - E))} \left[ r^{i-n} \times \chi(m, i) \times C_i^{n(1-E)} \right] \times C_n^{n(1-E)} \times (r - 1)^n \right\}.
\end{align*}
\]

($P(E \mid f, m)$ for all other $E$ is zero, since those $E$ cannot occur.)

APPENDIX B

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 $V(x_i, h(x_i))$ for some function $V$.

Write $d(x) = \sum_{y \in Y} V(y, x)$. $d(x)$ 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 $V'(y, x) \equiv V(y, x) / d(x)$. $T(h) = \{\Pi_{x \in X} d(x)\} \times \{\Pi_{x \in X} V'(h(x), x)\} \equiv D \times T'(h)$:

\[
(B.1) \text{ For } n \gg m \text{ and } T(h) = \Pi_{x \in X} V(h(x), x), \quad P(h \mid f, m) = T'(h) \times \left\{ \sum_{x \in X} \left( \delta(h(x), f(x)) \times \pi(x) / V'(f(x), x) \right) \right\}^m.
\]
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 
$[k(\theta_X, f(\theta_X))]^{-1}$ in the denominator in (5) we now have 
$\sum_{(h(x))} \prod_{x \in \theta_X} [V'(h(x), x)] \times \prod_{x \in \theta_X} [V'(f(x), x))]$, where the sum is understood to extend over all possible sets of $n - m'$ values 
$\{h(x)\}$, where $x$ is extending over the $n - m'$ $X$ values not contained in $\theta_X$ (there are $r^{n-m'}$ terms in the sum). By pulling the $\prod_{x \in \theta_X}$ term out of the sum we can rewrite this as the product of two 
terms: 
$[\sum_{(h(x))} \prod_{x \in \theta_X} [V'(h(x), x)] \times \prod_{x \in \theta_X} [V'(f(x), x))]$.

Now note that $\sum_{(h(x))} \prod_{x \in \theta_X} [V'(h(x), x)]$ 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 
$\Sigma_{\theta_X : m}$ found in (3) (where the value "m" in (3) is $n - m'$ here.) Continuing with the formal parallel 
between $[\sum_{(h(x))} \prod_{x \in \theta_X} [V'(h(x), x)] \times \prod_{x \in \theta_X} [V'(f(x), x))]$ and what is written as $\Sigma_{\theta_X : m} \prod_{i=1}^{m} [g_i(\theta_X(i))$ 
in (3), we can use (3) to write $\sum_{(h(x))} \prod_{x \in \theta_X} [V'(h(x), x)] = \prod_{x \in \theta_X} \Sigma_{y \in Y} [V'(y, x)]$.

Since $\Sigma_{y \in Y} [V'(y, x)] = 1$, this sum-of-a-product term just equals 1. This means that we have re-duced the $[k(\theta_X, f(\theta_X))]^{-1}$ term in the denominator of (5) to $\prod_{x \in \theta_X} [V'(f(x), x)]$.

Now assume that $n >> m$ and that $T(h)$ doesn't have any huge peaks, so that just as in the text when $T(h)$ was assumed constant we can here take $m' = m$ in the sum over $\theta_X$ giving $P(h \mid f, m)$. Doing this allows us to replace $\prod_{x \in \theta_X} [V'(f(x), x)] \times \prod_{i=1}^{m} [V'(f(\theta_X(i)), \theta_X(i))]$. This means that 
$P(h \mid f, m) = T(h) \times [\sum_{x \in X} \{\delta(h(x), f(x)) \times \pi(x) / V'(f(x), x)) \}^m$. QED.
The formula for $P(E | f, m)$ resulting from (B.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 result of Schwartz et al. 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 $\rho_0(E) \equiv \sum_h \{ T(h) \times \delta(E, Er(f, h)) \}$, you get $\rho_0(E, m) \\
\equiv \sum_h \{ T(h) \times \delta(E, Er(f, h)) \times [\eta(h)]^m \},$ where $\eta(h) \equiv [ \sum_{x \in X} \{ \delta(h(x), f(x)) \times \pi(x) / V'(f(x), x) \} ]$ \\
/ [ $\sum_{x \in X} \{ \delta(h(x), f(x)) \times \pi(x) \} ]$ 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


Washington Department of Computer Science, TR 91-03-04.


