# ON THE CONNECTION BETWEEN IN-SAMPLE TESTING AND GENERALIZATION ERROR.

by David H. Wolpert[1,2]

1 - Theoretical Division and Center for Nonlinear Studies, MS B213, LANL, Los Alamos, NM, 87545, (dhw@tweety.lanl.gov)
2 - The Santa Fe Institute, 1660 Old Pecos Trail, Suite A, Santa Fe, NM, 87501 (current address)

Abstract: This paper proves that it is impossible to justify a correlation between reproduction of a training set and generalization error off of the training set using only *a priori* reasoning. As a result, the use in the real world of any generalizer which fits a hypothesis function to a training set (e.g., the use of back-propagation) is implicitly predicated on an assumption about the physical universe. This paper shows how this assumption can be expressed in terms of a non-Euclidean inner product between two vectors, one representing the physical universe and one representing the generalizer. In deriving this result, a novel formalism for addressing machine learning is developed. This new formalism can be viewed as an extension of the conventional "Bayesian" formalism which (amongst other things) allows one to address the case where one's assumed "priors" are not exactly correct. The most important feature of this new formalism is that it uses an extremely low-level event space, consisting of triples of {target function, hypothesis function, training set}. Partly as a result of this feature, most other formalisms that have been constructed to address machine learning (e.g., PAC, the Bayesian formalism, the "statistical mechanics" formalism) are special cases of the formalism presented in this paper. Consequently such formalisms are capable of addressing only a subset of the issues addressed in this paper. In fact, the formalism of this paper can be used to address <u>all</u> generalization issues of which I am aware: over-training, the need to restrict the number of free parameters in the hypothesis function, the problems associated with a "non-representative" training set, whether and when cross-validation works, whether and when stacked generalization works, whether and when a particular regularizer will work, etc. A summary of some of the more important results of this paper concerning these and related topics can be found in the conclusion.

# I. INTRODUCTION.

## i) This paper's context.

This paper concerns the problem of inductive inference, sometimes also known as (supervised) machine learning. For most 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. (This function is sometimes instead called the "parent" function, or the "generating" 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*.

Some examples of generalizers are back-propagated neural nets [1], Holland's classifier system [2], and some implementations of Rissanen's minimum description length principle [3,4] (which, along with all other schemes which attempt to exploit Occam's razor, is analyzed in [5]). Other important examples are memory-based reasoning schemes [6], regularization theory [7,8], and similar schemes for overt surface fitting of a hypothesis function to the training set [9-13]. Conventional classifiers which work via Bayes' theorem, information theory, clustering analysis or the like (e.g. ID3 [14], Bayesian classifiers like Schlimmer's Stagger system described in [15], the systems described in [16], etc.) also serve as examples of generalizers. However such classifiers usually can only guess a particular output value if that value occurs in the training set. This paper assumes no such restriction on the guessing.

If for any training set $\theta$ of m pairs $\{x_i, y_i\}$ the generalizer always guesses $y_i$ when presented with the question $x_i$, we say that the generalizer *reproduces* the training set.[1] (Some researchers refer to the problem of reproducing the training set as the problem of "learning", to distinguish it from the problem of generalizing for questions outside of the training set.) For the sake of simplicity, in this paper I will usually assume noiseless data. For such a situation the training set should be reproduced exactly. It is trivial to ensure such exact reproduction of the training set; simply build a look-up table. (Difficulties only arise when one insists that the look-up table be implemented in an odd way, e.g., as a feed-forward neural net.) Therefore the only questions of interest are those outside of the training set.

The problem before us is to reach rigorous and meaningful conclusions concerning inductive inference. Recently there has been a lot of research attempting to do this while using reasoning which is as close to a priori as possible ([15, 17-25]). An archetypal example of such an analysis (related to the reasoning used in [17,19,20,23, 24]) is the "coin-tossing proof" of inductive inference. It can be summarized as follows (a more detailed presentation is made later in this paper):

First consider a simple Bernoulli problem. Randomly toss a coin with a well-defined probability of heads. Do this m times. Assume heads comes up s times. What is the probability that the next toss will come up heads? To answer this question, denote the true probability of heads by E, and the data that of m tosses s were heads by D. Then the conditional probability distribution $P(D \mid E)$ equals $C^m_s \times E^s \times (1 - E)^{m-s}$. This probability is maximized for $E = s / m$. Therefore a maximum likelihood analysis would say the probability of heads is $s / m$. We could instead do a full Bayesian analysis, assuming (for example) a uniform distribution of prior probabilities of possible E values, and then calculate the expectation value of E, $\int dE \, \{E \times P(E \mid D)\}$. As opposed to the maximum likelihood analysis, this Bayesian analysis says that the best guess for the probability of heads is $(s + 1) / (m + 2)$ (Laplace's law of succession). In either analysis, if s is close to m and m grows large, then our belief is high that on the next toss we'll get a head.

Now consider the following scenario: we have two functions, f and h, both going from X to Y. We randomly sample X according to some well-defined distribution, and call it a "heads" if for the randomly sampled X value f and h are in agreement, and call it a "tails" if they disagree. Do this m times, getting heads s times. Perform the exact same analysis as in the coin-tossing problem to estimate the probability that on the next "toss" we'll get heads. Then if s is close to m and m is large, we should expect that for all future questions from X, including those that haven't been seen before, it's likely that h and f will agree. Therefore it would appear that if a hypothesis function agrees with most of the elements of a randomly chosen training set, then it is likely to agree on future samples of the target function which generated the training set.

The coin-tossing argument makes a superficially convincing case for inductive inference. However it is at least as easy to make a counter-case, against inductive inference. This is done by noting that for *any* arbitrary behavior off of a training set, I can always design a surface fitter generalizer which will create a hypothesis function with that behavior. Moreover, if I so choose I can have that hypothesis function reproduce the training set perfectly (and still obey the "arbitrary behavior" off of the training set). The implication is that the behavior of a hypothesis function across a training set is completely independent of its behavior off of the training set, in apparent contradiction of the coin-tossing argument.

**ii) Precise issues addressed by this paper.**

This paper starts by examining the question of whether there can in principle be any applicability of an a priori reasoning scheme (like the coin-tossing argument) to the real world, or whether any such apparent applicability can only be mathematical sleight of hand.[2]

More precisely, this paper first investigates the following issue:

1) How should one construct a formalism so that any "sleight of hand" is overt? As part of the answer to this question, note that <u>no theory of generalization can have any real-world applicability whatsoever unless it takes into account the probability distribution of target functions in the real world</u>. "Sleights of hand" very often involve implicit and unjustified assumptions about that distribution. To avoid such assumptions, the formalism should require that the target function distribution is explicitly delineated. Also, unlike the conventional Bayesian approach, the formalism should acknowledge that whatever the target function distribution really is, it will almost certainly

differ (at least slightly) from what we assume it to be.

The primary point of this paper is to present the formalism developed in response to issue (1). To illustrate the use of that formalism, this paper also investigates the following (and related) issues:

2) Assume we have a certain level of agreement between a training set and a hypothesis function (e.g., the level of agreement could be 100%, meaning we have exact reproduction of the training set). Can this level of agreement, by itself, with no a priori information concerning the distribution of real-world target functions, give us any information concerning the error between the hypothesis function and the target function for questions outside of the training set, as is suggested by the coin-tossing analysis?

3) Given a level of agreement between a hypothesis function and a training set, does it matter, as is suggested in [23] and [26], whether the hypothesis function is guessed randomly, without any regard to the training set, or if it is instead constructed with the training set explicitly in mind?

4) Given the (negative) answer to (2), what assumptions about the real-world distribution of target functions are implicit in techniques like back-propagation which work by trying to minimize error on the training set?

5) If we pretend that we know the "distribution of real-world target functions" alluded to in (2) and (4) beforehand, how should we generalize?

6) Is there a mathematical basis for the often-voiced view that one can "over-train" a neural net, that training a neural net to the point that it perfectly reproduces the training set often reduces generalization accuracy? Can one estimate if over-training is occurring and perhaps modify the generalizer being used to mitigate such over-training?

7) Much research has gone into designing various "information criteria" (e.g., AIC [25]), "description length criteria" (e.g., [3,4]), PAC-framework criteria (e.g., [18]) and the like to address the utility of restricting the number of degrees of freedom of the hypothesis function. All this work notwithstanding, it is trivial to construct generalization problems by hand in which adding extra

degrees of freedom actually improves generalization accuracy. This raises the question: under what circumstances will it matter if one has "too many degrees of freedom" in the hypothesis function being used to fit the training set?

8) Researchers often speak of the difficulties encountered if the provided training set isn't "representative" of the full target function. How can these difficulties be given a mathematical expression?

9) There are a number of non-parametric statistics techniques that are designed to choose which of several candidate generalizers one should use to generalize from a provided training set. An archetypal example of such a "meta-generalization" technique is cross-validation. Under what circumstances will a particular meta-generalizer result in diminished generalization error?

10) How do the answers to these questions vary if we change the provided information? For example, if we're provided with a hypothesis function and a training set, do we get different predicted generalization errors from the case where we're not given the actual training set but only the knowledge of how often it agrees with the hypothesis function?

### iii) Synopsis of this paper.

In short, with its answer to issue (1), this paper presents a framework in which all machine learning issues of which I am aware can be addressed, in a rigorous, direct, and overt fashion. Most of the previous schemes attempting to address machine learning in a rigorous manner (e.g., PAC [15, 18-21], the work in [5], the Bayesian formalism [35-37, 42], the "statistical mechanics" formalism [17, 22, 43, 44]) are special cases of the formalism presented in this paper. Such schemes only address a subset of the issues which can be addressed with the formalism presented in this paper. Moreover, such schemes can be expressed in terms of the formalism presented in this paper, whereas the reverse is not true. In particular, the "Bayesian" approach to machine learning requires one's assumptions concerning "priors" to be exactly correct. In contrast, the formalism presented in this paper concentrates on the relationship between generalization error and {the level of agreement between one's assumed priors (contained in one's generalizer) and the "true" priors of the physical universe}. *Exact* agreement (which presumably never occurs in our universe) is simply a special case.

Section II of this paper addresses issue (1), i.e., presents the formalism used in this paper to investigate machine learning. This formalism can be viewed as an extension of the conventional Bayesian formalism. (Unlike the formalism presented in this paper, the conventional Bayesian formalism fails to distinguish target from hypothesis functions, and therefore *by construction* is incapable of addressing many of the issues addressed in this paper.) Section II then goes on to address issues (2) and (3). Section III addresses issues 4 and 6 through 8. It is in this section that the rule is derived relating generalization error to the inner product between one's generalizer (i.e., one's assumptions about the universe) and the actual universe. Section IV addresses issue 5 and shows that any generalization method, to believe that it is generalizing well, must make assumptions, either implicitly or explicitly, about the physical universe. Section V addresses issue 9, and is the most subtle and sophisticated of the sections making up this paper. It is shown in this section that, formally speaking, generalization and meta-generalization are essentially identical. An immediate consequence is that any phenomenon which occurs when one generalizes has a parallel when one meta-generalizes, and visa-versa. Issue 10 is addressed throughout the paper. A cursory summary of some of the results of this paper concerning all of these (and related) issues can be found in the conclusion.

It should be noted that this paper does not present and investigate any novel machine-learning algorithm. (Although a number of such algorithms, arising from the formal equivalence between meta-generalization and generalization, are suggested in section V.) This paper is primarily concerned with laying out a formalism and thereby disentangling some theoretical machine-learning issues. Detailed attempts to exploit that formalism to aid real-world generalization are beyond the scope of this (already lengthy) paper.[3]

## II. PRELIMINARIES.

This section resolves the first, second, and third issues presented at the end of the introduction. To address these issues, and in particular to avoid any possibility of “sleight of hand”, it is necessary to take extreme care in setting up the mathematics.[4] In particular, special care is necessary in defining the event space over which probabilities are defined. Although many researchers don’t bother to define their event space, it is only after that space is defined that can we conduct a rigorous probability-based analysis.

### i) The event space.

For simplicity of the analysis, assume the two spaces X and Y are discrete, with cardinalities n and r, respectively.[5] We also have two sets, F and H, both of which consist of all the functions taking X to Y. The elements of the first set are meant to represent the possible target functions. A generic element of F (e.g., an argument in a summand) will usually be written as *f*, whereas a particular target function will usually be indicated by f. The elements of the second set H are meant to represent the possible hypothesis functions. A generic element of H will usually be written as *h*, whereas a particular hypothesis function will usually be indicated by h. Throughout this paper, I will often view functions from X to Y as sets, of n pairs $\{x_j, y_j\}$.

Our event space (aka. universe of discourse, aka. Borel field) is written as U and consists of the set of possible triples $\{f, h, \theta\}$, where $f \in F$, $h \in H$, and $\theta$ is a set of m pairs $(x_i, y(x_i))$ $(1 \leq i \leq m)$, consisting of m values chosen from X along with m corresponding Y values. For full generality, I assume that the elements of $\theta$ are ordered; symmetry under permutation of the elements of $\theta$ can always be imposed afterward, if so desired. (Whether or not $\theta$ is an ordered set is relevant when we sum over training sets as (for example) in appendix B.)

In this paper, it will rarely be necessary to specify whether the event space encompasses $\theta$’s with different numbers of elements, whether it encompasses $\theta$’s with duplicate input-output pairs ("repeats"), etc. The context should make it clear in those rare instances when such a specification is being assumed. (In general, when such a specification is not made, the reader should assume that no repeats are allowed and that training sets are all of one particular size.) Note that whenever such issues *are* relevant, they can be formalized in a number of different ways. For example, to have no repeats one could have an event space encompassing θ's with duplications and assign a probability of 0 to all such θ's, or alternatively one could simply restrict the event space to θ's which have no duplications.

In this paper I assume that $m > 0$ (i.e., θ is non-empty). For simplicity, it is also assumed that $P(f, h, \theta)$, the joint probability of the target function f, the hypothesis function h, and the training set θ, equals zero unless the set θ is contained in the set f, i.e. unless $y(x_i) = f(x_i)$ for all pairs $\{x_i, y(x_i)\}$ in θ. In other words, we're assuming no noise. As was mentioned earlier, the assumption of no noise means that the problem of how to guess in response to a question q is only non-trivial if q is outside of θ. Whenever the no-repeats assumption is in effect, this in turn implies that we want $m < n$.

In this paper any function from X to Y can serve as a target function. However there might be cases where we wish to limit target functions to a subset of all functions from X to Y. To do this we would set to zero the probability of any triple containing an f from outside of the set of allowed target functions. (This scheme for addressing so-called "concept classes" allows us to avoid having to worry about whether or not F is a proper subset of all functions from X to Y.) Similar considerations apply for any limitations one might wish to impose on the set of possible hypothesis functions.

In this event space, how h depends on θ is set by the conditional probability distribution $P(h \mid \theta) = P(h, \theta) / P(\theta) = [\, \Sigma_{\{f\}} P(f, h, \theta) \,] \; / \; \Sigma_{\{h, f\}} P(f, h, \theta)]$. Due to our assumption of no noise, this can be rewritten as $[\Sigma_{\{f \supset \theta\}} P(f, h, \theta)] \; / \; [\Sigma_{\{h, f \supset \theta\}} P(f, h, \theta)]$. If h is determined independently of θ, $P(h \mid \theta)$ is independent of θ; alternatively, if h is fixed by θ (as in a generalizer), then this instead is reflected in $P(h \mid \theta)$. With a deterministic generalizer h is fixed uniquely by θ, which means that for $\theta$ fixed, $P(h \mid \theta)$ is a delta function over H. With a stochastic generalizer, although $P(h \mid \theta)$ depends on $\theta$, for a given $\theta$ it has support extending over more than one h.

In that h is supposed to be a guess made by the researcher, either via a random process or via a generalizer using $\theta$, we must impose special requirements on distributions concerning H. In particular, other than the samples of f, $\theta$, the researcher can have no knowledge of f when guessing a hypothesis function. This is expressed by requiring that $P(h \mid f, \theta)$ be independent of f.[6] In appendix A it is shown that this requirement is equivalent to requiring that $P(h \mid f, \theta) = P(h \mid \theta)$, which in turn is equivalent to requiring that $P(f \mid h, \theta) = P(f \mid \theta)$.[7]

In contrast to distributions concerning H, the probability distribution concerning target functions, $P(f) = [\Sigma_{\{h, \theta \subset f\}} \{P(f, h, \theta)\}]$, **is completely outside of the researcher's control**. P(f) is set by the physical universe; it is the probability distribution of target functions which humans encounter when trying to apply inductive inference (see appendix E). As such, it is the ultimate arbiter of what is and is not good generalization. In this paper we make no a priori restrictions on such distributions over F.

To relate $P(f, h, \theta)$ to the real world we need a way to measure the real-world "cost" associated with a particular choice of h. This is usually done with an error function, which can be viewed as measuring the level of agreement between a hypothesis function and a target function. (Note that although such a measure is conventionally referred to as a function, strictly speaking it is a functional.) Although the error is generically written as a function of all three elements of U, $\{f, h, \theta\}$, it sometimes doesn't depend on all of those elements. For example, if we assume that questions are generated at random, with repeats allowed, according to a pre-set distribution $\pi(x \in X)$, then an appropriate error function might be $Er(f, h, \theta) = \Sigma_x \pi(x) [1 - \delta(f(x), h(x)]$, where $\delta(., .)$ is the Kronecker delta function. Given an error function, $P(f, h, \theta)$ can be used to determine the probability distribution of error values, which (by hypothesis) is what gets measured in the real world. [8]

Often when investigating machine learning one makes an explicit assumption concerning how the target function is sampled to produce a training set. Such an assumption often goes hand in hand with a particular choice of error function. For example, we could assume that the X values of the elements of $\theta$ are chosen randomly, with repeats allowed, according to a pre-set distribution $\pi(x \in X)$, i.e., we could assume that whenever both $\theta \subset f$ and $\theta' \subset f$, and both training sets have the same cardinality, $P(\theta \mid f) / P(\theta' \mid f) = \prod_i[\pi(x_i)] / \prod_i[\pi(x_i')]$ (where $\theta$ is the set of pairs $\{x_i, y_i\}$ and $\theta'$ is the set of pairs $\{x'_i, y'_i\}$). If one also assumes no noise, then $P(\theta \mid f) = 0$ unless $\theta \subset f$. In general, the distribution $P(\theta \mid f)$ is called the "sampling assumption". The sampling assumption illustrated in this paragraph is very similar to the one implicit in the "coin-tossing" argument for inductive inference (see appendix B).) In general, the sampling assumption reflects three quantities: how X values at which to sample the target functions are chosen, how training set sizes are chosen (for the case where m is allowed to vary), and what kind of noise is in effect.

As much as possible the generalization errors calculated in this paper will be conditioned on a particular training set. In this way we can avoid undue reliance on a particular sampling assumption. For completeness however, it's necessary to state the sampling assumption made in this paper, rarely used though it might be. In this paper, for simplicity of the analysis, I will assume that $\theta$ is chosen according to a uniform distribution, with no repeats and no noise. More precisely, our sampling assumption states first that $P(f, h, \theta)$ equals zero unless all the X components of $\theta$ are distinct. It then states that for a given cardinality of training sets, $P(\theta \mid f)$ is independent of training set $\theta$, so long as $\theta$ and f are mutually consistent. (If they aren't consistent then due to our assumption of no noise the probability equals zero.) This second stipulation means that $P(f \mid h, \theta)$, which equals $P(f \mid \theta)$, is proportional to $P(f) / P(\theta)$ for those $\theta$ consistent with f.[9] However this fact will never be used in this paper.

Almost always, when generalizing in the real world what we're really interested in is the error rate of h for X values not contained in the training set. (This is especially true when there's little or no noise so we know how best to guess for a question contained in the training set. Even if there's a lot of noise however, if we know the parametric form of the noise then calculating how best to guess for questions from inside the training set is straight-forward (in theory at least), and again the only questions of interest are those outside of the training set.) An appropriate error function for this scenario and for the sampling assumption used in this paper is Er(f, h, θ) = $\Sigma_{\{x \notin \theta_X\}}$ [1 – δ(f(x), h(x)] / (n - m) ($\theta_X$ is defined as the set of the X components of the elements making up θ). More elaborate error measures could be used - e.g. $\Sigma_{\{x \notin \theta_X\}}$ (f(x) - h(x))$^2$ - but for simplicity such measures will not be considered here.[10]

More sophisticated versions of the analysis of this paper might make less restrictive sampling assumptions than the ones made here, perhaps allowing for noise or involving a sampling distribution π(x), and they might also use a correspondingly modified error function (e.g. [$\Sigma_{\{x \notin \theta_X\}}$ π(x) [1 – δ(f(x), h(x)]] / [$\Sigma_{\{x \notin \theta_X\}}$ π(x)]). By and large however, such changes in the sampling assumption and/or corresponding changes in the error function have little effect on the conclusions of this paper. (In fact, unless explicitly indicated otherwise, **all** conclusions reached in this paper are independent of the sampling assumption.) Nor do most of those conclusions change much if we modify the error function to reflect an assumption of independent identically distributed (i.i.d.) testing questions (i.e., to allow questions ∈ $\theta_X$) rather than the off-training set questions used here.

**ii) Comments on U and its use in a theory of supervised machine learning.**

It is important to note that the event space presented in this paper is completely symmetric between F and H; no aspect of the *definition* of that space treats hypothesis and target functions differently. The difference between F and H comes instead from the way we view F and H. As an example, due to this way of viewing F and H we add (!) the condition that P(f, h, θ) equals 0 unless θ ⊂ f. However such a condition is not intrinsic to the event space. Moreover, except for those rare circumstances when we will make use of a sampling assumption, this no-noise condition is the <u>only</u> way in which F and H will be treated differently (in particular, the error function is symmetric between F and H). With this in mind, it is very often possible to extend the mathematical results presented in this paper simply by interchanging target functions and hypothesis functions.

The following ten points summarize the definition of U:

1) The input space, with cardinality n, is X.

2) The output space, with cardinality r, is Y.

3) F and H are both the set of all functions from X to Y.

4) Training sets $\theta$ consist of m pairs $\{x_i, y_i\}$. $\theta_X \equiv$ the set of all the $\{x_i\}$ in training set $\theta$.

5) U is the set of all triples $\{f \in F, h \in H, \theta\}$.

6) $P(h \mid \theta)$ is the generalizer, usually set by the researcher.

7) $P(\theta \mid f)$ is the sampling assumption, sometimes set by the researcher.

8) P(f) is set by the universe, and is unknown to the researcher. (Conventional Bayesian analysis involves making assumptions directly for P(f).)

9) $P(h \mid f, \theta) = P(h \mid \theta)$. $P(f \mid h, \theta) = P(f \mid \theta)$.

10) $Er(f, h, \theta)$ is the error function, usually set by the researcher.

Using the formalism presented in this paper, supervised machine learning involves two stages. First, one fixes certain aspects of the full distribution $P(f, h, \theta)$ (e.g., one fixes $P(h \mid \theta)$) and chooses a conditional probability distribution of interest (e.g., $P(Er(f, h, \theta) = E \mid h, \theta)$). Then one goes about calculating that distribution.

One advantage of this process is that by having the researcher start by writing down the conditional probability distribution in which (s)he is interested, the context and assumptions are forced to be explicit. For example, as it turns out, some previous machine learning research implicitly has target functions fixed (as can be deduced by noting that it calculates quantities like $P(Er(f, h, \theta) = E \mid f, \theta)$), some other research implicitly allows target functions to vary (i.e., calculates quantities like $P(Er(f, h, \theta) = E \mid \theta)$), some other research implicitly only fixes the training set size and not its elements (i.e., calculates quantities like $P(Er(f, h, \theta) = E \mid m)$), etc. It is suspected that in some of this previous work the researchers themselves weren't aware of the implicit conditions underlying their results. If the formalism of this paper had been used, such ignorance would not have been possible.

Another advantage of the formalism presented in this paper is that because U is so low-level, essentially *every* aspect of a supervised machine-learning issue can be case in terms of U. As a result, as was mentioned in the introduction essentially all previous supervised machine learning formalisms which are based on probability theory can be cast in terms of the formalism presented in this paper.[11] Using the formalism presented here, it is seen that the differences amongst those previous formalisms is just that they analyze different conditional probabilities over the space U, with different a priori restrictions on the full joint distribution $P(f, h, \theta)$.

Aside from section V, the rest of this paper consists of the calculation of some important conditional probability distributions over the space U which have not previously been calculated.

**iii) The independence of reproducing the training set and having low generalization error.**

When calculating conditional probabilities, the right-hand-side should have everything which is fixed, which includes in particular everything known to the researcher. The left-hand-side should be the quantity of interest. Therefore, since almost always what the researcher knows is the training data and his resultant guess, and since what the researcher is interested in is the error associated with that guess, we need to calculate $P(E \mid h, \theta)$. Formally speaking, by the "E" argument is meant the union of all events $(f, h, \theta')$ such that $Er(f, h, \theta') = E$. The "h" argument in the distribution P is the event of our particular hypothesis function h (i.e., it is the union over our event space of all events which have hypothesis function h). The final argument, $\theta$, is the union of all events which have training set $\theta$.

Restrict attention to those error values E such there exists at least one $f \in F$ containing $\theta$ for which $Er(f, h, \theta) = E$. $P(E \mid h, \theta)$ necessarily equals zero for all other E values. (Amongst others, this restriction excludes all E values which are negative and all values which exceed 1.) By definition,

$P(E \mid h, \theta) \;=\; P(E, h, \theta) \,/\, P(h, \theta)$

$$= \frac{\Sigma_{\{f \supset \theta\}} \{\delta[Er(f, h, \theta), E] \times P(f, h, \theta)\}}{\Sigma_{\{f \supset \theta\}} \{P(f, h, \theta)\}} \,,$$

(Note that the condition in the sum "$f \supset \theta$" is actually superfluous, since $P(f, h, \theta)$ equals zero if this condition isn't met. Nonetheless, for clarity, in this paper "$f \supset \theta$" will be explicitly written where $\delta(.,.)$ is the Kronecker delta function. (For clarity, in this paper the condition "$f \supset \theta$" will be explicitly written whenever it applies. Note that it is actually superfluous to do so however, since that condition is a direct reflection of the fact that $P(f, h, \theta)$ equals zero if $\theta \not\subset f$.)

Now use the fact that $P(f, h, \theta) = P(f \mid h, \theta) \times P(h \mid \theta) \times P(\theta) = P(f \mid \theta) \times P(h \mid \theta) \times P(\theta)$ to rewrite the formula for $P(E \mid h, \theta)$:

(2.1) $$P(E \mid h, \theta) = \Sigma_{\{f \supset \theta\}} \{\delta[Er(f, h, \theta), E] \times P(f \mid \theta)\}.$$

To proceed further, we need to make some assumptions about $P(f \mid \theta)$. As a first example, assume that $P(f \mid \theta)$ is constant over its support in F; all target functions consistent with $\theta$ are equally likely, given only $\theta$ and some guessed hypothesis function h.[12] We have the following theorem:

(2.2) When $P(f \mid \theta)$ is independent of f for all f consistent with $\theta$,

$$P(E \mid h, \theta) \;=\; C_z^{(n-m)} \times (r - 1)^{(n-m-z)} \,/\, r^{(n-m)}, \text{ where } z \equiv [(n - m)(1 - E)].$$

Proof: Label the m elements of $\theta$ as $(x_{n-m+1}, y_{n-m+1})$, $(x_{n-m+2}, y_{n-m+2})$, ... $(x_n, y_n)$. Label the remaining elements of X according to the same scheme, so that questions outside of $\theta$ are chosen from the set $\{x_1, ..., x_{n-m}\}$. This allows us to rewrite the sum $\Sigma_{\{f \supset \theta\}}$ as $\Sigma_{\{f_1, ..., f_{n-m}\}}$, where $f_i$ is shorthand for $f(x_i)$, and the sum is understood to extend over all $r^{(n-m)}$ possible values of its subscript. Now write the constant value P(f | θ) has over its support as p. This allows us to rewrite P(h, θ) as $p \times \Sigma_{\{f_1, ..., f_{n-m}\}}\{1\} = p \times r^{(n-m)}$. Similarly, P(E, h, θ) becomes

$$p \times \Sigma_{\{f_1, ..., f_{n-m}\}}\{\delta[Er(f, h, \theta_X), E]\}$$

$$= p \times \Sigma_{\{f_1, ..., f_{n-m}\}}\{\delta([\Sigma_{i=1}^{n-m} \delta(f_i, h(x_i))], (n - m)(1 - E))\}.$$

This is just p times the number of instances in which the set $\{f_1, ..., f_{n-m}\}$ agrees with $\{h(x_i)\}$ exactly $z \equiv [(n - m)(1 - E)]$ times.[13] Simple combinatorics tells us that this equals $p \times C_z^{(n-m)} \times (r - 1)^{(n-m-z)}$. Therefore when P(f | θ) is independent of f, $P(E \mid h, \theta) = C_z^{(n-m)} \times (r - 1)^{(n-m-z)} / r^{(n-m)}$. QED.

This case where P(f | θ) is constant over its support is the distribution with the maximum entropy. It serves as a benchmark case, corresponding to a "random" universe.[14] Equation (2.2) shows that for such a maximum entropy P(f | θ), P(E | h, θ) is explicitly independent of s, the number of agreements between the training set θ and the hypothesis function h. Since there is no a priori reason to rule out the possibility that P(f | θ) has maximum entropy, there is no *a priori* reason to rule out (2.2), and therefore there is no theoretical justification for the oft-voiced claim that, for a priori reasons alone, P(E | h, θ) depends on s. This answers question (2) from the introduction; we have an explicit proof that all arguments trying to justify the claim of inductive inference without making a priori assumptions (e.g., without assuming P(f | θ) depends on f) are wrong.

In fact, (2.2) shows that for this benchmark case of flat $P(f \mid \theta)$, not only is $P(E \mid h, \theta)$ independent of s, it is independent of h entirely. This means, for example, that one can not meaningfully say "what size neural net gives valid generalization" without making some ultimately ad hoc assumptions about $P(f \mid \theta)$. To put it another way, (2.2) shows that any attempt to determine in an a priori fashion what size neural net to use implicitly makes assumptions about $P(f \mid \theta)$, and without justifying those assumptions there is no reason to believe the resulting conclusion.

Note also that (2.1) says that $P(E \mid h, \theta)$ is independent of $P(h \mid \theta)$. In fact, (2.1) shows that $P(E \mid h, \theta)$ is independent of $P(h \mid \theta)$ even if $P(f \mid \theta)$ varies with f, since changing $P(h \mid \theta)$ has no effect on $P(f \mid \theta)$. In other words, although replacing our maximum entropy assumption with some other assumption allows the choice of h to affect $P(E \mid h, \theta)$ (see the next section), no assumption for the form of $P(f \mid \theta)$ results in a correlation between $P(E \mid h, \theta)$ and $P(h \mid \theta)$. This fact can be used to address question (3) from the introduction: Despite the suggestions of some authors ([23-24]) and despite what intuition might say, given a particular hypothesis function h, as far as the distribution $P(E \mid h, \theta)$ is concerned, it does not matter if h was fixed before the researcher has any knowledge of $\theta$, or if instead h was chosen based on $\theta$ (as in a generalizer).

## III. THE ASSUMPTIONS NEEDED FOR IN-SAMPLE TESTING TO BE RELEVANT.

Given (2.2), how is that in our physical universe so many learning techniques which expend all of their effort at reproducing the training set manage to generalize fairly well? This section addresses this question by investigating what assumptions can result in a correlation between {the level of agreement between h and $\theta$} and {the generalizing error for questions outside of $\theta$}. In doing so, this section answers issues four, six, seven, and eight from the introduction.

### i) Non-constant $P(f \mid \theta)$.

$P(f \mid \theta)$ is determined by the physical universe, and by what kinds of f's are likely in our physical universe. If, in our universe, for the kinds of problems generalizing algorithms are usually tested on, reproducing the training set results in good generalization, then it must be that the MaxEnt assumption for $P(f \mid \theta)$ is wrong. In other words, we have empirical evidence suggesting that there are non-uniformities in the distribution of inference problems (or at least in the distribution of such problems with which humanity has so far concerned itself).[16,17]

In this section I consider the case where MaxEnt is wrong and $P(f \mid \theta)$ is more elaborate than $\{P(f \mid \theta) = 0$ if $\theta \not\subset f$; $P(f \mid \theta) =$ a constant over its support in F$\}$. First rewrite (2.1) to explicitly delineate what aspect(s) of h are relevant to the generalization error:

$$(3.1) \qquad P(E \mid h, \theta) = \Sigma_{\{f \supset \theta\}} \{P(f \mid \theta) \times \delta([\Sigma_{i=1}^{n-m} \delta(f(x_i), h(x_i))], z)\},$$

where z is the number of off-training set agreements between h and $f$, $(n - m)(1 - E)$, (just like in the proof of (2.2)), and the $n - m$ elements of the set $X - \theta_X$ are labelled $x_1$ through $x_{n-m}$. By appropriate choice of the function $P(f \mid \theta)$, we can make almost any distribution $P(E \mid h, \theta)$ we choose. In particular, choose $P(f \mid \theta)$ to equal 0 unless all of the $f(x_i)$ equal $h'(x_i)$ for some pre-set hypothesis function h', in which case it equals 1. In this case, $P(E \mid h', \theta)$ equals 1 for $E = 0$, and 0 for all other E values; we get perfect generalization.[18] However for generic $h \neq h'$ we do not get perfect generalization. In other words, given that $P(f \mid \theta)$ is allowed to vary, for fixed $\theta$ different h might lead to different $P(E \mid h, \theta)$.

Although this clarifies how choice of h can affect generalization, it still leaves unresolved the two other issues raised at the end of the previous section: Even for a non-uniform $P(f \mid \theta)$, $P(E \mid h, \theta)$ is independent of both s and of $P(h \mid \theta)$, in apparent violation of common experience.

To resolve these two issues, first make the definition $S(f, h, \{x_i\}) \equiv \Sigma_{\{i\}} \delta[h(x_i), f(x_i)]$. S is a mapping taking two functions and a set of X values to an integer. That integer is the number of times the two functions agree with one another on what output corresponds to an input, over the set of provided X values. Note that $S(f, h, \{x_i\})$ is symmetric under interchange of f and h. Often for conciseness the third argument to S will be given as the X components of a set of X-Y pairs (i.e., a training set) rather than directly as a set of X values. For example, I will sometimes write $S(f, h, \theta_X)$ by which I mean $S(f, h, \{x_i\})$, where the $\{x_i\}$ are the X components of $\theta$. Using this notation, $S(f, h, X - \theta_X)$ is the number of times h(x) agrees with f(x) for x outside of $\theta$; $Er(f, h, \theta) = 1 - S(f, h, X - \theta_X) / (n - m)$. Similarly, if $\theta \subset f$, then $S(f, h, \theta_X)$ is the number of times h agrees with $\theta$. Note that in such cases we can write $S(\theta, h, \theta_X)$ instead of $S(f, h, \theta_X)$.

$S(\theta, h, \theta_X)$ never occurs in equation (3.1); if I know h and $\theta$, then I have determined $P(E \mid h, \theta)$, and counting $S(\theta, h, \theta_X)$ doesn't have any effect on $P(E \mid h, \theta)$. In other words, despite the fact that we're no longer assuming maximum entropy $P(f \mid \theta)$, (3.1) tells us that it's still true that only h's behavior outside of the training set is relevant.[19]

To see why (3.1) might not be the final word on inductive inference, note that we're interested in the "dependence" between $s \equiv S(\theta, h, \theta_X)$ and E. Now such "dependence" is a meaningless concept if s isn't allowed to vary. However s *can't* vary if both h and $\theta$ are fixed. This suggests that to analyze inductive inference, we should fail to specify either h and/or $\theta$, and evaluate (for example) $P(E \mid s, \theta)$ rather than $P(E \mid h, \theta)$.

Another reason for examining a quantity like $P(E \mid s, \theta)$ rather than $P(E \mid h, \theta)$ comes from the fact that in practice we can not evaluate (3.1), since we don't know $P(f \mid \theta)$ - it's determined by the universe. To get around this problem, we could just make a direct assumption for $P(f \mid \theta)$. That's a *huge* assumption to make however (which, interestingly enough, doesn't stop Bayesians from making it). As an alternative, it would be preferable to make one (or more) relatively weak *indirect* assumptions which can "take the place" of knowledge *directly* concerning $P(f \mid \theta)$.

As an example, we can make an "indirect" assumption about $P(f \mid \theta)$ and assume that there exists a correspondence between $P(f \mid \theta)$ and the distribution $P(h \mid \theta)$. In other words, we can assume that the probability distribution over the architecture on which we're going to implement our hypothesis function (e.g., a feedforward neural net) corresponds in a certain way to the probability distribution over target functions in the real world.[20]

As an example of how such an indirect assumption can mitigate (3.1) and its implication that generalization error is independent of $S(\theta, h, \theta_X)$, assume that over our event space only those target functions can occur which are constant (i.e., independent of X). Now assume that we only examine constant hypothesis functions, i.e. assume that the space of hypothesis functions we're examining corresponds to the space of target functions. Then the agreement (or lack thereof) between a hypothesis function and a target function over the elements of the training set fixes exactly the agreement between the two functions over input values outside of the training set. It's still true that only h's behavior outside of the training set is relevant, as stipulated by (3.1). But now the correspondence between the distribution over F and the distribution over H couples the error outside of the training set with the error inside the training set. The next subsection is a formal investigation of such coupling behavior in the context of evaluating $P(E \mid s, \theta)$.[21]

**ii) How agreement between a hypothesis and a training set can affect generalization.**

We have some probability distribution across our event space, and therefore in particular a distribution $P(h \mid \theta)$. Choose a stochastic $P(h \mid \theta)$, i.e., a $P(h \mid \theta)$ whose support extends over more than one h for a given $\theta$ (so that the hypothesis guessed by the generalizer isn't uniquely fixed by the training set). In fact, in practice the $P(h \mid \theta)$ used in the following analysis is often completely independent of $\theta$. For example, the support of $P(h \mid \theta)$ might be the set of feed-forward neural nets smaller than a given size. As another example, the support of $P(h \mid \theta)$ might be the set of functions formed by taking linear combinations of the elements of some set of basis functions.

Now consider the following scheme: Let s'' be some integer and randomly pick a hypothesis function h from H (according to the probability distribution over H), subject to the condition that $S(\theta, h, \theta_X) = s''$. This procedure defines a distribution $P(h \mid \theta, s'')$ which can be used as a generalizer. Now pick a new integer $s' > s''$. s' defines a new generalizer $P(h \mid \theta, s')$.[22]

We wish to choose between a hypothesis function output by the generalizer P(h | θ, s') and a hypothesis function output by the generalizer P(h | θ, s''). Our strategy is to use the fact that s'' < s' to choose the hypothesis function output by the generalizer P(h | θ, s'). The following argument shows that this strategy results in desirable P(E | s, θ) if there is a correspondence between P(f | θ) and P(h | θ).

By expressing it in terms of f and then marginalizing over F, one proves that P(E, s, θ) = $\Sigma_{\{h\}}\{P(E, h, \theta) \times \delta[S(\theta, h, \theta_X), s]\}$. Similarly, P(s, θ) = $\Sigma_{\{h\}} P(h, \theta) \times \delta[S(\theta, h, \theta_X), s]$. Therefore we can write

$$(3.2) \qquad P(E \mid s, \theta) = \frac{\Sigma_{\{h\}}\{P(E \mid h, \theta) \times P(h, \theta) \times \delta[S(\theta, h, \theta_X), s]\}}{\Sigma_{\{h\}}\{P(h, \theta) \times \delta[S(\theta, h, \theta_X), s]\}} \quad .$$

Note that one can replace both of the P($h$, θ) terms in (3.2) (one in the numerator, one in the denominator) by P($h$ | θ).

(3.2) explicitly relates P(E | s, θ) to P(E | h, θ), the quantity analyzed in the previous section. If P(E | h, θ) is independent of h, as in the maximum entropy case described by equation (2.2) where P(f | θ) is independent of f, then we can take P(E | h, θ) out of the sum in the numerator in (3.2), and P(E | s, θ) is independent of s. In other words, for the maximum-entropy case, it doesn't matter what level of agreement there is between our hypothesis function and the training set; inductive inference still doesn't hold, even though we're examining P(E | s, θ) rather than P(E | h, θ).

Similarly, if P(h | θ) is independent of h, then again P(E | s, θ) is independent of s. This is suggested by the interchange symmetry between F and H. A formal proof is as follows:

Proof: First, note that if P(h | θ) is independent of h (3.2) becomes

$$P(E \mid s, \theta) = \frac{\Sigma_{\{h\}}\{P(E \mid h, \theta) \times \delta[S(\theta, h, \theta_X), s]\}\}}{\Sigma_{\{h\}}\{\delta[S(\theta, h, \theta_X), s]\}\}} .$$

In a manner similar to that in section II, we can split up the sum over h into two sums, one over the values of $h$(x) where x is within the training set, and one over the values of $h$(x) where x is outside of the training set: $\Sigma_{\{h\}} = \Sigma_{\{h_1, h_2, ..., h_{n-m}\}} \Sigma_{\{h_{n-m+1}, h_{n-m+2}, ..., h_n\}}$. This allows us to re-write our denominator as $\Sigma_{\{h_1,h_2,...,h_{n-m}\}}\Sigma_{\{h_{n-m+1},h_{n-m+2},...,h_n\}}\{\delta(S(\theta, h, \theta_X), s)\} = r^{n-m} \times \Sigma_{\{h_{n-m+1},h_{n-m+2},...,h_n\}} \{\delta(S(\theta, \boldsymbol{h'}, \theta_X), s)\}$, where $\boldsymbol{h'}$ is defined as the m-tuple $\{h_{n-m+1}, h_{n-m+2}, ..., h_n\}$. Evaluating, we get $r^{n-m} \times C^m_s \times (r - 1)^{m-s}$. Since P(E | $h$, θ) is only dependent on that part of $h$ outside of θ, we can rewrite our numerator in a similar fashion, getting $\Sigma_{\{h_1,h_2,...,h_{n-m}\}}\{P(E \mid \boldsymbol{h}, \theta)\} \times \Sigma_{\{h_{n-m+1},h_{n-m+2},...,h_n\}} \{\delta(S(\theta, \boldsymbol{h'}, \theta_X), s)\}$, where $\boldsymbol{h}$ is defined as the (n-m)-tuple $\{h_1, h_2, ..., h_{n-m}\}$ and $\boldsymbol{h'}$ is as before. Therefore when P($h$, θ) is independent of $h$, $P(E \mid s, \theta) = r^{m-n} \times \Sigma_{\{h_1,h_2,...,h_{n-m}\}}\{P(E \mid \boldsymbol{h}, \theta)\}$, and is independent of s. QED.

In particular, if h is chosen at random according to a uniform distribution, the generalization error is independent of s.

If P(E | $h$, θ) is **not** independent of $h$ (cf. equation (3.1)), and if P(h | θ) is **not** uniform, then P(E | s, θ) **can** depend on s. In other words, under these circumstances there can be coupling between on-training set error and off-training set error (as measured by P(E | s, θ)) and inductive inference can occur. The precise form of the inductive inference, i.e., how P(E | s, θ) depends on s, is determined by the form of P(f | θ) (which determines P(E | $h$, θ)) and the form of P(h | θ). The following is an example of this:

Example: Assume $P(f \mid \theta) = 1$ if f is some particular function f*, 0 otherwise. (Note that $S(\theta, f^*, \theta_X) = m$, i.e., f* must agree with θ, so we can rewrite $S(\theta, h, \theta_X)$ as $S(f^*, h, \theta_X)$ for any function h.) Similarly, assume $P(h \mid \theta) = 1/2$ if h is f*, 1/2 if h is some particular function h' for which $S(f^*, h', \theta_X) = m - 1$ ($S(f^*, h', X - \theta_X)$ being unspecified except that it doesn't equal n - m), and 0 for all other h. For this scenario, the denominator in (3.2) equals 1/2 if s = m, and 1/2 if s = m - 1. (If s equals neither m nor m - 1, $P(E \mid s, \theta)$ is undefined, since the event (s, θ) can never occur.) To evaluate the numerator in (3.2), note that $P(E \mid h, \theta) = 1$ if $S(f^*, h, X - \theta_X) = (n - m)(1 - E)$, 0 otherwise. In other words, $P(E \mid h, \theta) = \delta(E, 1 - [S(f^*, h, X - \theta_X) / (n - m)])$. For s = m, this means that $P(E \mid s, \theta) = 1$ for E = 0, 0 otherwise. For s = m - 1, $P(E \mid s, \theta) = 1$ for $E = 1 - S(f^*, h', X - \theta_X) / (n - m)$ (which is greater than 0), 0 otherwise.

A single number giving "the generalization error" can be defined in a number of ways (e.g. as $\text{argmax}_E\, P(E \mid s, \theta)$, as $\Sigma_{\{E\}}\{E \times P(E \mid s, \theta)\}$, etc.). Whatever the precise definition used, for the scenario recounted in the example above, if $S(f^*, h', X - \theta_X) > 0$ then the generalization error either improves or stays constant as s increases. We will refer to this behavior by saying that "reproduction correlates with generalization", for all s, for this scenario, for any (reasonable) definition of generalization error. When reproduction correlates with generalization, we should pick a hypothesis function h found by randomly sampling H subject to the constraint that h agrees with all of θ, rather than a hypothesis function h found by randomly sampling H subject to the constraint that h does *not* agree with all of θ.

**iii) When reproduction correlates with generalization.**

Reproduction does not correlate with generalization for all pairs $\{P(f \mid \theta), P(h \mid \theta)\}$. An example is the following:

Example: As in the previous example, assume that $P(f \mid \theta) = 1$ if f is some particular function f*, 0 otherwise. Similarly, assume $P(h \mid \theta) = 1/2$ if h is some particular function h* for which $S(\theta, h^*, \theta_X) = m$, 1/2 if h is some particular function h' for which $S(\theta, h', \theta_X) = m - 1$, and 0 for all other h. (Note that h* need not equal f*.) As before, the denominator in (3.2) equals 1/2 if s = m, and 1/2 if s = m - 1. Also as before, $P(E \mid h, \theta) = \delta(E, 1 - [S(f^*, h, X - \theta_X) / (n - m)])$. For s = m, this means that $P(E \mid s, \theta) = 1$ for $E = 1 - S(f^*, h^*, X - \theta_X) / (n - m)$, 0 otherwise. Similarly, or s = m - 1, $P(E \mid s, \theta) = 1$ for $E = 1 - S(f^*, h', X - \theta_X) / (n - m)$, 0 otherwise. If $S(f^*, h^*, X - \theta_X) < S(f^*, h', X - \theta_X)$, then generalization error <u>increases</u> as s goes up.

This behavior is not so uncommon as might be hoped. For example, as reported in [38], for the parity target function the error of backpropagation *increases* as the cardinality of the training set goes up.

It has often been observed that one can "over-train" an architecture (e.g., a feedforward neural net), and thereby degrade the generalization. What is meant by this is that if s is forced to be too close to m, then for certain scenarios the generalization error is empirically observed to start to increase. "Over-training" means that reproduction correlates with generalization for small s but not for large s. Usually over-training is considered a side-effect of noise; one "learns the noise" if one trains too much. However it can occur even in the absence of noise, as the following example illustrates:

Example: Consider the situation where in addition to the f*, h*, and h' of the previous example, we also have h'', and h''', where $S(\theta, h'', \theta_X) = m - 2$, and $S(\theta, h''', \theta_X) = m - 3$. As above, have the probabilities of all of the elements of H under consideration (of which there are now four) equal to one another, and the probability of any other element of H equal to zero. Then if $S(f^*, h^*, X - \theta_X)$

< $S(f^*, h', X - \theta_X)$ while $S(f^*, h', X - \theta_X) > S(f^*, h'', X - \theta_X) > S(f^*, h''', X - \theta_X)$, reproduction correlates with generalization in going from s = m - 3 to s = m -2 to s = m - 1, but in going from s = m - 1 to s = m the generalization error increases.

It is still an open problem to classify all pairs $\{P(f \mid \theta), P(h \mid \theta)\}$ for which reproduction correlates with generalization over given ranges in s. In particular, it is not known precisely how $P(h \mid \theta)$ must relate to $P(f \mid \theta)$ if reproduction is to correlate with generalization for all s. Nor is it known precisely how they must be related to result in "over-training". Nor is it currently known whether or not reproduction correlates with generalization *on average*, where the average is over all possible pairs $\{P(f \mid \theta), P(h \mid \theta)\}$ and over all s. (Note however that (3.1) suggests that on average reproduction does not correlate with generalization just as often as it does.)

Nonetheless, some interesting observations can be made by rewriting (3.2). In particular, $P(E \mid s, \theta)$ can be rewritten as the inner product $\Sigma_{\{h\}} P(E \mid h, \theta) \times P(h \mid s, \theta)$. More enlighteningly, we can rewrite it as an inner product with a non-Euclidean metric, where the two vectors are simply $P(h \mid \theta)$ and $P(f \mid \theta)$:

$$(3.3) \qquad P(E \mid s, \theta) = \Sigma_{\{f, h\}} P(h \mid \theta) \times P(f \mid \theta) \times M_{E,s,\theta}(h, f),$$

where $M_{E,s,\theta}(h, f) \equiv \kappa_{s,\theta} \times \delta[S(f, h, \theta_X), s] \times \delta[S(f, h, X - \theta_X), z]$, $\kappa_{s,\theta}$ being a normalization constant set by the condition $\Sigma_{\{E\}} P(E \mid s, \theta) = 1$. (The "$\theta \subset f$" condition is being enforced by the $P(f \mid \theta)$ term. There is no equivalent restriction on the $P(h \mid \theta)$ term.) Note that $M_{E,s,\theta}(h, f)$ is a symmetric matrix; this is what allows us to interpret (3.3) as an inner product of two vectors with $M_{s,E}(h, f)$ being the metric. Note that if noise is allowed, we must replace the $\delta[S(f, h, \theta_X), s]$ term with $\delta[S(\theta, h, \theta_X), s]$ and the matrix is no longer symmetric. (On the other hand, even if noise is allowed a quantity like $P(E \mid \theta)$ *is* an inner product with a symmetric matrix.) Also note that equation (3.3) holds for any error function, even one which allows questions to run over the training set. (Different error functions simply correspond to different ME,s,q(., .).)

Whether or not reproduction correlates with generalization is determined by the precise definition of generalization error used and by the two vectors $P(h \mid \theta)$ and $P(f \mid \theta)$. The metric $M_{E,s,\theta}(h, f)$ is parameterized by s and E; this allows us to view the inner product of our two vectors as a function of E, this function being parameterized by s. To determine whether or not reproduction correlates with generalization, (3.3) instructs us to see how the inner product of our two vectors, viewed as a function of E, changes as s is increased. Implicitly, whenever one uses a technique like back-propagation, one is making an assumption about how this inner product function between $P(h \mid \theta)$ and $P(f \mid \theta)$ changes with s. Rather than making an assumption about (3.1) directly (by assuming something about $P(f \mid \theta)$), one is making an assumption about (3.3) (by assuming something about the correspondence between $P(h \mid \theta)$ and $P(f \mid \theta)$). Phrased another way, whenever one uses a technique like back-propagation over a particular set of allowed neural nets, one is implicitly assuming we live in a universe whose $P(f \mid \theta)$ "corresponds" to the set of allowed feed-forward neural nets, $P(h \mid \theta)$.

**iv) Ramifications of (3.3).**

The arguments of the preceding subsections can be used even if one isn't solely interested in $S(f, h, \theta_X)$. For example, one might want to use a regularizer in conjunction with $S(f, h, \theta_X)$ to guess a hypothesis function (see [7,8], and references therein). In such a situation, instead of investigating how generalization error depends on $S(f, h, \theta_X)$, one wants to investigate how it depends on $S(f, h, \theta_X)$ together with R(h), where R(h) is some regularizer cost function (e.g., the integrated curvature of h). The formalism presented above can be modified to describe whether or not minimizing both $S(f, h, \theta_X)$ and R(h) together leads to better generalization on average, whether "over-regularizing" (akin to over-training) occurs, etc. All such issues reduce to an investigation of a non-Euclidean inner product between $P(f \mid \theta)$ and $P(h \mid \theta)$ (where now the matrix M(h, f) is not necessarily symmetric in h and f, since the regularizer is only a function of h). This is because using a regularizer simply sub-divides the partition of equivalence classes of hypothesis functions h with identical $S(f, h, \theta_X)$. The precise form of the metric defining the inner product is determined by the final partition used, i.e., it is determined by the regularizer used.

There are a number of interesting issues raised by (3.3), even if one doesn't extend it to encompass regularization. Besides questions of genericness (e.g., how generic is over-training), (3.3) could perhaps allow one to predict the full function $P(E \mid s, \theta)$ from samples of $P(E \mid s, \theta)$. For example, one might be able to examine the behavior of $P(E \mid s, \theta)$ for several low s cases, and from this information deduce how likely over-training will be for high s cases. If the conclusion is that over-training is likely, then perhaps one could change $P(h \mid \theta)$ to obviate the over-training.

In addition to raising such new issues, (3.3) is also a way to address many old issues. For example, a commonly held belief is that, loosely speaking, if one has too many parameters with which to vary the hypothesis function then one might end up with large generalization error ([25, 4]). Such a phenomenon can be translated into the terms of this paper's formalism in a number of different ways. For example, one might wish to view this phenomenon as similar to over-training, which was dealt with above. Alternatively, having too many parameters to vary might be interpreted as meaning that the substrate distribution $P(h \mid \theta)$ is too broad (viewed as a function of h); too many hypothesis functions are allowed. With this interpretation, whether or not too many parameters to fit is undesirable, "on average", is determined by whether or not a broad $P(h \mid \theta)$ means that reproduction does not correlate with generalization, "on average".

Taking this viewpoint, even without performing the relevant calculations in detail one can use (3.3) to make an heuristic argument that it's beneficial to hold down the number of free parameters. The argument starts by noting that $P(E \mid s, \theta)$ is a continuous functional of $P(h \mid \theta)$. We also know that at the one extreme, if $P(h \mid \theta)$ is uniform then reproduction has zero correlation with generalization. Moreover when $P(h \mid \theta)$ is not uniform, reproduction can correlate with generalization. Therefore, in general, as $P(h \mid \theta)$ is varied from uniformity to being sharply peaked over H, there can be more and more of a correlation between reproduction of a training set and generalization.

Since such peakedness in $P(h \mid \theta)$ corresponds to limiting the number of free parameters in the hypothesis function, it is reasonable to expect that holding down the number of free parameters, if it is done in such a way as to "align" $P(h \mid \theta)$ with $P(f \mid \theta)$, facilitates correlation between reproduction of a training set and generalization.

Another issue that can be addressed by this kind of heuristic argument is the often-observed necessity of having the training set be "representative" of the target function in order to get good generalization from that training set. To address this issue, first one must define what it means for a training set not to be "representative" of the target function from which it's sampled. One possible definition is that a training set $\theta$ is not "representative" of its actual target function if there are many target functions all of which, with high probability, might have served as the parent of $\theta$. In other words, $\theta$ is not representative of its target function if $P(f \mid \theta)$ is broad and not sharply peaked, so $\theta$ doesn't help much in picking the correct target function. As an extreme case, we know that if $P(f \mid \theta)$ is flat then reproduction of $\theta$ has 0 correlation with generalization. We can now invoke the continuous dependence of $P(E \mid s, \theta)$ on $P(f \mid \theta)$ to argue that even if $P(f \mid \theta)$ isn't flat, if it is close to flat, i.e., if it is broad and not sharply peaked, then the correlation between reproduction of $\theta$ and generalization from $\theta$ is weak. This argument suggests, in accord with empirical observation, that $\theta$ must be representative of its target function in order for one to generalize from $\theta$ with low generalization error.

Both this argument concerning how representative $\theta$ is and the argument concerning the number of free parameters are only meant to be illustrative. One could carry out the analyses in much more detail, getting fully rigorous results. Such a detailed investigation of these issues is beyond the scope of this paper however. It should also be noted that there are other ways of addressing these issues. (For example, holding down the number of free parameters is an implementation of Occam's razor. Accordingly, [5] describes, in a context slightly different from that presented in this paper, when and how such a strategy will result in diminished generalization error.)

## IV. THE BEST POSSIBLE HYPOTHESIS FUNCTION.

If, as in conventional Bayesian analysis, we know (or pretend we know) the U-space distribution $P(f \mid \theta)$, there's no need to explicitly consider reproduction of the training set, meta-generalization, or any such issue. In such a scenario, we already know everything that can possibly be relevant. This kind of scenario raises its own questions however. For example, since we know $P(f \mid \theta)$, then given that choice of h can make a difference (see the discussion below (3.1)), what h should we use for a given $\theta$? (Note that this is issue five from the introduction.)

There are several ways to answer this question. For example, in analogy to maximum likelihood techniques, we could stipulate that one should pick the h which minimizes the mode of a probability distribution over errors (e.g., minimize the mode over E of $P(E \mid h, \theta)$). Another possibility is to minimize the expectation value of the error with respect to such a distribution. A third possible approach, which is investigated in this section, is to find the h which maximizes the probability of zero error.

For illustrative purposes, we will not try to find the h which is optimal for the distribution $P(E \mid h, \theta)$. Instead we will try to find the $P(h \mid \theta)$ which is optimal for the distribution $P(E \mid \theta)$. More precisely, we will find the $P(h \mid \theta)$ which maximizes $P(E = 0 \mid \theta)$. The answer is

(4.1) To maximize $P(E = 0 \mid \theta)$, one should guess a hypothesis function h such that the output values of h for points outside of $\theta$ are the same as those for the target function $\text{argmax}_{f \in F}$ $P(f \mid \theta)$. The resultant value of $P(E = 0 \mid \theta)$ is $\max_{\{f \in F\}} \{P(f \mid \theta)\}$.

Proof: First recall that $P(f, h, \theta) = P(\theta) \times P(h \mid \theta) \times P(f \mid \theta)$. $P(h \mid \theta)$ is determined by our generalizer. This allows us to write

$$P(E \mid \theta) = P(E, \theta) / P(\theta)$$

$$= \frac{\sum_{\{h, f \supset \theta\}} \{\delta[Er(f, h, \theta_X), E] \times P(f, h, \theta)\}}{P(\theta)}$$

$$= \sum_{\{h, f \supset \theta\}} \{\delta[Er(f, h, \theta_X), E] \times P(h \mid \theta) \times P(f \mid \theta)\} .$$

(Compare to equation (3.3).) In a manner similar to the proof of (2.2) we can rewrite this as

$$\sum_{\{f_1, \ldots, f_{n-m};\, h_1, \ldots, h_{n-m}\}} \{P'(f_1, \ldots, f_{n-m}) \times P'(h_1 \ldots, h_{n-m}) \times \delta([\sum_{i=1}^{n-m} \delta(f_i, h_i)], z)\},$$

where $P'(\text{event}) \equiv P(\text{event} \mid \theta)$. $P'(f_1, \ldots, f_{n-m})$ is determined by the universe. Our job is to determine the optimal $P'(h_1 \ldots, h_{n-m})$.

Now make the following notational conventions: A generic (n-m)-tuple $\{f_1, \ldots, f_{n-m}\}$ is indicated by $\boldsymbol{f}$, whereas a particular (n-m)-tuple is indicated by $\mathbf{f}$. Both tuples live in the space $\mathbf{F}$. The i'th component of such a tuple, $\mathbf{f}_i$, equals $f_i$, where f is the full n-tuple $\{f_1, \ldots, f_n\}$. (The difference between f and $\mathbf{f}$ is that f is an n-dimensional vector $\in$ F, whereas $\mathbf{f}$ is only (n-m)-dimensional and lives in $\mathbf{F}$.) Similarly, the generic (n-m)-tuple $\{h_1, \ldots, h_{n-m}\}$ is indicated by $\boldsymbol{h} \in \mathbf{H}$. Note that since f must contain θ, $P'(\mathbf{f}) = P'(f)$. However h need not contain θ, so giving h's values for inputs outside of the training set doesn't specify h completely, and $P'(\mathbf{h})$ need not equal $P'(h)$: $P'(\mathbf{h}) = P'(h_1, \ldots, h_{n-m}) = \sum_{\{h_{n-m+1}, \ldots, h_n\}} P'(h_1, \ldots, h_{n-m}, h_{n-m+1}, \ldots, h_n)$. Also make the definition $T(\mathbf{f}, \mathbf{h}) \equiv$

$\delta([\Sigma_{i=1}^{n-m} \delta(f_i, h_i)], n - m))$. (Note that (n - m) is the value of z when E = 0.) Finally, define U(**h**) $\equiv [\Sigma_f \{P'(f) \times T(f, \mathbf{h})\}]$. Note that $U(\mathbf{h}) \geq 0 \ \forall \ \mathbf{h}$.

Our task is to find the probability distribution P'(**h**) which maximizes $\Sigma_{\boldsymbol{h}} \{P'(\boldsymbol{h}) \times U(\boldsymbol{h})\}$, i.e., find the probability distribution which maximizes the expectation value of U(**h**), $\langle U(\mathbf{h}) \rangle_{\{\mathbf{h}\}}$. This maximizing distribution is simply $P'(\mathbf{h}) = \delta[\mathbf{h}, \text{argmax}(U(\mathbf{h}))]$. In other words, for a given training set θ, the optimal generalizer guesses an **h** which maximizes $\Sigma_{\{f\}} \ [\ P'(f) \times \delta\{[\Sigma_{i=1}^{n-m} \delta(f_i, \mathbf{h}_i)], (n - m)\}\ ]$. Now note that the outermost Kronecker delta in this expression is 1 iff $f$ agrees with **h** on all (n - m) points outside of the training set. Therefore $\Sigma_{\{f\}} \ [\ P'(f) \times \delta\{[\Sigma_{i=1}^{n-m} \delta(f_i, \mathbf{h}_i)], (n - m)\}\ ]$ just equals P'(**f**) evaluated for **f** = **h**. The rest of the proof follows from the fact that our task is to maximize $\langle U(\mathbf{h}) \rangle_{\{\mathbf{h}\}}$. QED.

Note that in generalizer $\text{argmax}_{f \in F} P(f \mid \theta)$, the hypothesis function one should guess to maximize $P(E = 0 \mid \theta)$, is <u>not</u> the same as the "Bayes-optimal" function consisting of the input-output pairs $\{x, \text{argmax}_{y \in Y} \Sigma_f [P(f \mid \theta) \times \delta(f(x), y)]\}$.

If the set H is a proper subset of F, then (4.1) doesn't apply, since we can't guess the mode of $P(f \mid \theta)$. For this case we must instead solve the full problem of finding the (allowed) **h** which maximizes U(**h**) (see the proof of (4.1) for the definition of U(.) and **h**).

Theorem (4.1) tells us that under the assumption that $P(\mathbf{h} \mid \theta)$ is the same distribution over **H** as $P(\mathbf{f} \mid \theta)$ is over **F**, then we should guess the mode of $P(\mathbf{h} \mid \theta)$. This advice is exactly the same as that given by Occam's razor when a uniform simplicity measure is used. (Most simplicity measures - e.g., the number of weights in a neural net, or the coding length of a theory - appear to be approximations of simplicity measures; see [5]). Moreover, section III told us that this assumption of a correspondence between $P(f \mid \theta)$ and $P(h \mid \theta)$ is what allows us to assume that reproduction of the training set correlates with generalization. In other words, the assumption that Occam's razor works is related to the assumption that reproduction of the training set correlates with generalization. This is exactly the result proven (more formally) in [5] using a completely different formalism.

Theorem (4.1) answers the question of how best to generalize from a given training set, at least for this definition of "best". We can phrase (5.1) differently: any generalizer, no matter what the context, to believe that it is generalizing as well as possible, is of necessity making an assumption concerning the mode of P(f). Sometimes this assumption will be explicit. For example, in regularization theory one might guess h from $\theta$ by fitting the elements of $\theta$ with the surface of minimal integrated curvature going through the elements of $\theta$. Often however, and especially in neural net learning schemes, the assumption about P(f) is never made explicitly.

## V. META-GENERALIZATION.

For almost any real-world generalization problem, there are many generalizers which could be used to guess a hypothesis function. Therefore, either implicitly or otherwise, every time one generalizes one uses a scheme for choosing which generalizer to apply. This means that it is desirable to have an algorithm for explicitly choosing which amongst a set of possible generalizers to use with a particular provided training set $\theta$. This is the idea behind techniques like cross-validation [26-30, 34], bootstrap [30], and the like. It is also the starting point for more sophisticated techniques where one generalizes amongst generalizers [31,32].

There are several ways to modify the formalism of the previous sections to allow one to address the issue of whether or not a particular scheme for choosing amongst generalizers will lead to small generalization error. For simplicity, it is easiest to illustrate the ideas when the scheme for choosing amongst generalizers is cross-validation. The rest of this section consists of such an illustration. (As such, it is an answer to issue nine presented in the introduction.) First a relatively simple way to address the efficacy of cross-validation is presented, and then a more sophisticated scheme is outlined. In this second approach, a new "meta" space is created in which minimizing cross-validation error plays an identical role to the role played, in the analysis of U, by minimizing error at reproduction of the training set.

**i) Cross-validation.**

Recall that a (deterministic) generalizer is any algorithm which takes in a training set θ of m pairs ($x_i \in X$, $y_i \in Y$) along with a question $q \in X$ and produces a guess $\in Y$.[23] As is pointed out in [11], any such algorithm g is equivalent to a countably infinite set of functions g{i}, where g{i} is a mapping from $\{(X \times Y)^i \times X\}$ to Y. g{1} maps a training set consisting of 1 pair ($x_1 \in X$, $y_1 \in Y$) along with a question q to an output (guess) in Y; the resulting output is written as $g\{1\}(x_1, y_1; q)$. Similarly, g{2} takes a training set consisting of 2 pairs $\{(x_1 \in X, y_1 \in Y), (x_2 \in X, y_2 \in Y)\}$ along with a question q to an output $g\{2\}(x_1, y_1, x_2, y_2; q)$ in Y, etc. Any set of g{i} (defined for all i > 0) uniquely fixes a deterministic generalizer and visa-versa.[24]

Sometimes I won't explicitly indicate the cardinality of the training set and simply write g(θ; q) when what I really mean is g{cardinality of θ}(θ; q). Similarly, I will sometimes take liberties with the definition of a generalizer and view it as a mapping from training sets to input-output functions rather than from training sets together with inputs to outputs. (Such a meaning is assumed whenever I write g(θ) as opposed to g(θ; q).) As always, the context should make meanings clear. I will say that a particular generalizer g is "trained" with a particular training set θ when I have in mind the function g(θ; q) (or g(θ), as the case might be).

Assume we have a set of p generalizers, indicated by $g_1, ..., g_p$ (so the i'th function defining the j'th generalizer is indicated by $g_j\{i\}$). Let θ be a training set consisting of m elements. Use the notation that $\theta_{-i}$ is the training set θ minus the i'th pair, $\{x_i, y_i\}$. Then the cross-validation error associated with the j'th generalizer is defined as $\Sigma_{i=1}^{m} \; g_j\{m-1\}(\theta_{-i}; x_i) - y_i]^2$. It measures the average error of generalizer j at guessing one of the input-output pairs in θ when taught with the rest of θ. The idea of cross-validation is simple; generalize from θ with the generalizer $g_j$ having the lowest cross-validation error over θ.[25,26] (See [32,26-30, 34].) Rather than directly assuming a best generalizer, cross-validation assumes something about how best to choose a generalizer.

**ii) Addressing cross-validation via a new event space.**

It's straight-forward to modify the formalism of the previous several sections to directly address the issue of whether and when cross-validation will result in low generalization error. Again have an event space consisting of triples. Two of the triples are identical to those in the previous sections: target functions $f \in F: X \rightarrow Y$, and training sets θ consisting of m pairs of elements from $X \times Y$. For the third element of the triple however, we no longer have hypothesis functions from X to Y. Rather we have "hypothesis" generalizers $g \in G$ taking training sets to functions from X to Y. (G is the space of all possible generalizers for input space X and output space Y.[27])

Denote the new event space by V. V and U (the event space of the previous several sections) are very similar to one another. In particular, all the probability axioms from the previous sections are assumed to hold for V, with the set G replacing the set H. (For example, it's assumed that $P(g \mid f, \theta)$ is independent of f.) We also make the same sampling assumption as when analyzing U. Moreover, the error function for V is directly related to the error function for U: $Er_V(f, g, \theta) = \Sigma_{\{x \notin \theta\}} [1 - \delta(g(\theta; x), h(x)] / (n - m)$.[28]

In this new event space however, conditional probabilities have different meanings from those they held in U. In particular, $P(g \mid \theta)$ is interpreted as the probability of picking generalizer g given training set θ. This probability is set by the researcher (just as $P(h \mid \theta)$ was before). It reflects our use of cross-validation (or whatever scheme one is interested in) to choose a generalizer based on the provided training set.[29]

All of the analysis of the previous sections goes through essentially unchanged for this new event space. In particular, the maximum entropy distribution of target functions results in a formula for $P(E \mid g, \theta)$ which is independent of g. Therefore, just as previously we were forced to assume that the distribution of target functions isn't uniform to allow choice of h to be of any consequence, so must we now make this assumption to allow choice of g to be of any consequence. In particular, without such a non-uniform distribution, use of cross-validation will not help lower generalization error. In other words, contrary to the claims of some ([17]), cross-validation and related schemes can not be proven to work from first principles; it is an assumption that cross-validation results in good generalization, equivalent to the assumption that reproducing the training set results in good generalization.

In section III we supplemented the investigation of $P(E \mid h, \theta)$ with an investigation of $P(E \mid s, \theta)$. We can do a similar thing with our new event space, and investigate $P(E \mid s, \theta)$ rather than $P(E \mid g, \theta)$, where in this new context s is given by the functional S' rather than the functional S, and measures the sum, over i, of whether or not $g(\theta_{-i}; x_i)$ agrees with $y_i$ (i ranging over the m elements making up $\theta$). The conclusion is similar to that reached in section III concerning whether or not reproduction correlates with generalization: whether or not "cross-validation correlates with generalization" is determined by the behavior of the inner product $P(E \mid s, \theta) = \Sigma_{\{f,g\}} P(g \mid \theta) \times P(f \mid \theta) \times M'_{E,s,\theta}(g, f)$, where $M'_{E,s,\theta}(g, f) \equiv \kappa'_{s,\theta} \times \delta[S'(f, g, \theta), s] \times \delta[S(f, g(\theta), X - \theta_X), z]$, $\kappa'_{s,\theta}$ being a normalization constant set by the condition $\Sigma_{\{E\}} P(E \mid s, \theta) = 1$.

This formula relating how one chooses a generalizer to the generalization error allows us to tackle issues previously unaddressable. For example, it's straight-forward to show that for any training set, there exists a generalizer with zero cross-validation error making any guess desired for questions outside of that training set (see [34]). Therefore the only way that a technique like cross-validation could be helpful is if before using it one restricts the set of generalizers over which the cross-validation is run. However once that set of generalizers is restricted, one has clearly made some sort of implicit assumption about what generalizers should use in our particular physical universe. The explicit form of that assumption is given by the formula at the end of the preceding paragraph.