Complex Systems **6** (1992) 47–94

On the Connection between In-sample Testing and Generalization Error

David H. Wolpert*
Theoretical Division and Center for Nonlinear Studies,
MS B213, Los Alamos National Laboratory, Los Alamos, NM, 87545, USA

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 that 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, to (among other things) allow one to address the case in which 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, and 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 *all* generalization issues of which the author is 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, and so forth. A summary of some of the more important results of this paper concerning these and related topics can be found in the conclusion.

*Current address: The Santa Fe Institute, 1660 Old Pecos Trail, Suite A, Santa Fe, NM, 87501. Electronic mail address: dhw@sfi.santafe.edu

1. Introduction

1.1 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 that will be referred to as the *target* function. (This function is sometimes called the "parent" or "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 that 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 that 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 that 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], or the systems described in [16]) also serve as examples of generalizers. However, such classifiers usually can guess a particular output value only if that value occurs in the training set. This paper assumes no such restriction on the guessing.

If for any training set θ 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 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 arise only 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 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 that attempts to do this while using reasoning that is as close to a priori as possible [15, 17–25]. An archetypal example of such an analysis (related

[1] It is implicitly assumed in such a statement that there are no two pairs in θ with the same input values x_i but different output values y_i. See [11] for more details.

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 {of m tosses s were heads} by D. Then the conditional probability distribution $P(D \mid E)$ equals $C_s^m \times E^s \times (1 - E)^{m-s}$. 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, $\int dE\{E \times P(E \mid D)\}$. In contrast 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 will get heads.

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 "heads" if, for the randomly sampled X value, f and h are in agreement; we call it "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 will 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 have not been seen before, it is 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 that 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 that 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 to the coin-tossing argument.

1.2 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 *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.* "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 be delineated explicitly. 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. Can one prove inductive inference from first principles? 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 the training set, as suggested by the coin-tossing analysis?

3. Given a level of agreement between a hypothesis function and a training set, does it matter, as suggested in [23] and [26], whether the hypothesis function is guessed randomly, without any regard to the training set, or 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 that work by trying to minimize error on the training set?

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

[2]One example of such a sleight of hand is confusing the prior distribution of input-output functions in the real world with the prior distribution of feed-forward neural nets (as in [17] and any attempts to apply studies like [18] to the real world). Another sleight of hand is limiting the space of allowed target functions in some "reasonable way." Yet another is allowing questions to run over the training set. (After all, the only non-trivial issue in the noise-free case—the only issue of interest—is how to guess for questions outside the training set; allowing questions to run over the training set is, at best, obfuscatory.) See appendix D for a discussion of such sleights of hand in the PAC formalism.

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 following 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 is not "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, do the predicted generalization errors when we are provided with a hypothesis function and a training set differ from when we are not given the actual training set but only the knowledge of how often it agrees with the hypothesis function?

1.3 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], and 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 that 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 2 of this paper addresses issue (1), that is, it 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 functions from hypothesis functions, and therefore *by construction* is incapable of addressing many of the issues addressed in this paper.) Section 2 then addresses issues (2) and (3). Section 3 addresses issues (4), (6), (7), and (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 4 addresses issue (5) and shows that any generalization method, if it generalizes well, must make assumptions, either implicitly or explicitly, about the physical universe. Section 5 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 that occurs when one generalizes has a parallel when one meta-generalizes, and vice 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 5. This paper is primarily concerned with laying out a machine-learning formalism and thereby disentangling some theoretical machine-learning issues. Detailed attempts to exploit the formalism to aid real-world generalization are beyond the scope of this (already lengthy) paper.[3]

2. 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

[3]It should be noted that this limitation of scope is shared by essentially all other theoretical machine-learning research in the literature that does not directly make assumptions about the real universe (e.g., PAC, the statistical mechanical formalism, and the analysis of various linear models). Essentially none of this previous research even *suggests* novel machine-learning algorithms (as is done in this paper in section 5), let alone empirically investigates the real-world utility of such an algorithm. In point of fact, as a general rule any of this previous machine-learning research that at first glance appears to have real-world ramifications will, on closer inspection, be found to have none (e.g., [18]; see [41]).

up the mathematics.[4] In particular, special care is necessary in defining the event space over which probabilities are defined. Although many researchers do not define their event space, it is only after that space is defined that we can conduct a rigorous probability-based analysis.

2.1 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 that take 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 (or universe of discourse, or 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 θ 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 θ are ordered; symmetry under permutation of the elements of θ can always be imposed afterward, if so desired. (Whether or not θ 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 θ's with different numbers of elements, whether it encompasses θ's with duplicate input-output pairs ("repeats"), and so forth. 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 that includes θ's with repeats and assign a probability of 0 to all such θ's; alternatively one could simply restrict the event space to θ's that have no repeats.

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

[4]Such care shows that the "coin-tossing" argument as it stands is flawed, for example. See appendix B after reading through this section.

[5]This assumption has the immediate consequence that, for the most part, no asymptotic arguments either for or against a particular scheme can be made (infinity is not defined for discrete spaces). This restriction is hardly a shortcoming since, even for continuous input and output spaces, asymptotic behavior is *never* what we are directly interested in (since training sets are always finite), and therefore arguments concerning such behavior can be extremely misleading. (This is especially true in the many instances in which those arguments are made without any concern for bounding the error that comes from applying their results to the finite case.)

f, the hypothesis function h, and the training set θ—equals zero unless the set θ is contained in the set f, that is, unless $y(x_i) = f(x_i)$ for all pairs $\{x_i, y(x_i)\}$ in θ. In other words, I assume no noise. As 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 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 in which 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 the set of allowed target functions. (This scheme for addressing so-called "concept classes" allows us to avoid having to worry about whether 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 $P(h \mid \theta) = P(h, \theta)/P(\theta) = [\sum_{\{\mathrm{f}\}} P(\mathrm{f}, h, \theta)]/\sum_{\{\mathrm{h,f}\}} P(\mathrm{f}, \mathrm{h}, \theta)]$. Due to our assumption of no noise, this equation can be rewritten as $[\sum_{\{\mathrm{f} \supset \theta\}} P(\mathrm{f}, h, \theta)]/[\sum_{\{\mathrm{f} \supset \theta, \mathrm{h}\}} P(\mathrm{f}, \mathrm{h}, \theta)]$. 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 θ fixed, $P(h \mid \theta)$ is a delta function over H. For a stochastic generalizer, $P(h \mid \theta)$ depends on θ but, for a given θ, it has support extending over more than one h.

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

[6]It should be noted that this requirement is more of a tautology than an "assumption," given that we are doing supervised machine learning. (Indeed, this requirement is implicitly made in every theoretical treatment of machine learning of which I am aware.) Consider changing the target function *while keeping the training data the same.* Since the generalizer sees only the training data (by definition), such a change in the target function provides no change in the information at our disposal that tells us how the generalizer is likely to guess. To put it another way, any algorithm run on a computer—and a generalizer in particular—*must* have its output depend solely on its input. And in supervised machine learning, that output is the hypothesis function, and that input is the training data, θ.

[7]In our physical universe the probability of a given target function can vary with the training set (in fact it has to since $P(f, h, \theta)$ is zero unless f and θ are mutually consistent.) On the other hand, the hypothesis function h chosen by us either is also determined (at least in part) by θ, or is random. In either case, specifying the hypothesis function we guess in addition to specifying θ does not help the universe determine what target function generated θ. This is an intuitive justification of the result that $P(f \mid h, \theta) = P(f \mid \theta)$.

In contrast to distributions concerning H, the probability distribution concerning target functions, $P(f) = [\sum_{\{h,\theta \subset f\}}\{P(f, h, \theta)\}]$, *is completely outside the researcher's control.* $P(f)$ is set by the physical universe; it is the probability distribution of target functions that humans encounter when trying to apply inductive inference (see appendix E). As such, it is the ultimate arbiter of what is good generalization. In this paper no a priori restrictions are made 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 does not 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 $\text{Er}(f, h, \theta) = \sum_x \pi(x)[1 - \delta(f(x), h(x))]$, where δ 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 θ are chosen randomly, with repeats allowed, according to a pre-set distribution $\pi(x \in X)$, that is, we could assume that whenever both $\theta \subset f$ and $\theta' \subset f$, and both training sets have the same cardinality, then $P(\theta \mid f)/P(\theta' \mid f) = \prod_i[\pi(x_i)]/\prod_i[\pi(x'_i)]$ (where θ is the set of pairs $\{x_i, y_i\}$ and θ' 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

[8] Although in the real world we usually have direct access to both the hypothesis function (after all, we constructed it) and the training set, we do not necessarily have such access to target functions. The only connection between the real world and the mathematical construct of target functions occurs through the (probability distribution concerning the) error function. However this error function need not concern the level of agreement between the function h and some hypothesized pre-fixed target function f *that we believe was sampled to produce* θ. In other words, although we usually think of the training set as being a sample of a target function, such a view is *not* intrinsic to U. Nothing in the mathematics that defines U says that we assume there exists a fixed, unknown target function f that is sampled to create θ. Nonetheless, it is often the case that the researcher has in mind precisely this scenario in which θ is a sampling of some pre-fixed target function. (This is implicitly the case, for example, whenever we talk of "no noise" and therefore set $P(f, h, \theta) = 0$ for $\theta \not\subset f$.) Accordingly, the discussion in the rest of this paper will be presented in terms of such a scenario, and error functions and the like will be chosen accordingly.

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 is necessary to state the sampling assumption made in this paper, rarely used though it might be. For simplicity of the analysis, I will assume that θ 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 θ are distinct. It then states that, for a given cardinality of training sets, $P(\theta \mid f)$ is independent of the training set θ, so long as θ and f are mutually consistent. (If they are not 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 θ consistent with f.[9] However, this fact will never be used in this paper.

As previously mentioned, almost always, when generalizing in the real world, what we are really interested in is the error rate of h for X values not contained in the training set. (Even if there is a lot of noise, if we know the parametric form of the noise then calculating how best to guess for questions from inside the training set is straightforward, at least in theory, and again the only questions of interest are those outside the training set.) An appropriate error function for this scenario and for the sampling assumption used in this paper is $\mathrm{Er}(f, h, \theta) = \sum_{\{x \notin \theta_X\}}[1 - \delta(f(x), h(x))]/(n - m)$ (θ_X is defined as the set of the X components of the elements making up θ). More elaborate error measures could be used—for example, $\sum_{\{x \notin \theta_X\}}(f(x) - h(x))^2$—but for simplicity such measures will not be considered here.[10]

[9] $P(\theta \mid f) = P(f, \theta)/P(f) = P(f, \theta)/[\sum_{\{w \subseteq f\}} P(f, w)]$, where $\{w \subseteq f\}$ is the set of all training sets w consistent with f. $P(\theta \mid f) = P(\theta' \mid f) \forall \theta$ and θ' that are consistent with f then implies that if $\theta \subset f$, $P(\theta \mid f) = \{1/[\sum_{\{\omega \subseteq f\}} 1]\} \equiv k$. ($k$ can be calculated by summing over all allowed training set cardinalities the number of training sets of that cardinality that can be chosen from f. For example, if all training set cardinalities are allowed in U, but training sets are not allowed to contain repeats, then $k^{-1} = \sum_{i=1}^{n} [n!/i!]$.) Note that k is independent of both f and θ. As a result, $P(f \mid \theta) = P(\theta \mid f) \times P(f)/P(\theta) \propto P(f)/P(\theta)$, which proves the supposition.

[10] Note that in addition to avoiding reliance on a sampling assumption (by calculating errors conditioned on a particular training set), it is also possible to avoid making an assumption—as is implicit in any error function—concerning how questions are chosen. To do this we must calculate "errors" conditioned on a particular question, that is, we replace an investigation of the probability of errors of the form $\sum_{\{x \notin \theta_X\}}[1 - \delta(f(x), h(x))]/(n - m)$ with an investigation of the probability of errors of the form $[1 - \delta(f(q), h(q))]$, where q is a provided question. Formally, this can be done in several ways. One way is to expand our event space to be quadruples (f, h, θ, q). Another way is to keep the original event space involving triples (f, h, θ) and implement the following procedure: After being given θ, reduce X to the set of $m+1$ elements $\{\theta_X \cup q\}$ and then use the original error measure $\sum_{\{x \notin \theta_X\}}[1 - \delta(f(x), h(x))]/(n - m)$. Investigations of any sort based on being provided a single question q will not be pursued in this paper.

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 $\pi(x)$; and they might also use a correspondingly modified error function (e.g., $[\sum_{\{x \notin \theta_X\}} \pi(x)[1 - \delta(f(x), h(x))]]/[\sum_{\{x \notin \theta_X\}} \pi(x)]$). 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 $\in \theta_X$) rather than the off-training set questions used here.

2.2 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 $P(f, h, \theta)$ equals 0 unless $\theta \subset f$. 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 *only* 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 θ consist of m pairs $\{x_i, y_i\}$. $\theta_X \equiv$ the set of all the $\{x_i\}$ in training set θ. Similarly for θ_Y.
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. $\mathrm{Er}(f, h, \theta)$ 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., $P(\mathrm{Er}(f, h, \theta) = E \mid h, \theta)$). Then one goes about calculating that distribution.

One advantage of this two-stage process is that since the researcher starts by writing down the conditional probability distribution that is of interest, 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 $P(\mathrm{Er}(f, h, \theta) = E \mid f, \theta)$), some research implicitly allows target functions to vary (i.e., calculates quantities like $P(\mathrm{Er}(f, h, \theta) = E \mid \theta)$), and other research implicitly fixes only the training set size and not its elements (i.e., calculates quantities like $P(\mathrm{Er}(f, h, \theta) = E \mid m)$). It is suspected that in some of this previous work the researchers themselves were not aware of these implicit conditions underlying their results. If the formalism of this paper had been used, such lack of rigor 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 cast in terms of U. As a result, as was mentioned in the introduction, essentially all previous supervised machine-learning formalisms that 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 among 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 5, the rest of this paper consists of the calculation of some important conditional probability distributions over the space U that have not previously been calculated in any formalism.

2.3 The independence of reproducing the training set and having low generalization error

When calculating conditional probabilities, the right-hand side (i.e., the conditions) should have everything that is fixed, which includes in particular everything known to the researcher. The left-hand side should be the quantity of interest. Therefore, since what the researcher knows is almost always restricted to the training data and the resultant guess by the generalizer, 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 $(\mathrm{f}, \mathrm{h}, \theta')$ such that $\mathrm{Er}(\mathrm{f}, \mathrm{h}, \theta') = E$. 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

[11] Note that conventional Bayesian analysis, which does not distinguish H from F, cannot serve as such an over-arching framework.

that have hypothesis function h). The final argument, θ, refers to the union of all events that have training set θ.

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

$$\begin{aligned} P(E \mid h, \theta) &= P(E, h, \theta)/P(h, \theta) \\ &= \frac{\sum_{\{\mathrm{f} \supset \theta\}} \{\delta[\mathrm{Er}(\mathrm{f}, h, \theta), E] \times P(\mathrm{f}, h, \theta)\}}{\sum_{\{\mathrm{f} \supset \theta\}} \{P(\mathrm{f}, h, \theta)\}}, \end{aligned}$$

where δ is the Kronecker delta function. (For clarity, in this paper the condition "$\supset \theta$" 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 $P(E \mid h, \theta)$:

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

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 θ are equally likely, given only θ and some guessed hypothesis function h.[12] We have the following theorem:

Theorem 1. *When $P(f \mid \theta)$ is independent of f,*

$$P(E \mid h, \theta) = C_z^{(n-m)} \times (r-1)^{(n-m-z)}/r^{(n-m)}, \tag{2.2}$$

where $z \equiv [(n-m)(1-E)]$.

Proof. Label the m elements of θ as $(x_{n-m+1}, y_{n-m+1}), (x_{n-m+2}, y_{n-m+2}), \ldots, (x_n, y_n)$. Label the remaining elements of X according to the same scheme, so that questions outside θ are chosen from the set $\{x_1, \ldots, x_{n-m}\}$. This allows us to rewrite the sum $\sum_{\{\mathrm{f} \supset \theta\}}$ as $\sum_{\{\mathrm{f}_1, \ldots, \mathrm{f}_{n-m}\}}$, where f_i is shorthand for $\mathrm{f}(x_i)$, and the sum is understood to extend over all $r^{(n-m)}$ possible values of its subscript. Now write the constant value that $P(f \mid \theta)$ has over its support as p. With (2.1), this allows us to rewrite $P(E \mid h, \theta)$ as

$$\begin{aligned} & p \times \sum_{\{\mathrm{f}_1, \ldots, \mathrm{f}_{n-m}\}} \{\delta[\mathrm{Er}(\mathrm{f}, h, \theta_X), E]\} = \\ & \quad p \times \sum_{\{\mathrm{f}_1, \ldots, \mathrm{f}_{n-m}\}} \left\{ \delta \left(\left[\sum_{i=1}^{n-m} \delta(\mathrm{f}_i, h(x_i)) \right], (n-m)(1-E) \right) \right\}. \end{aligned}$$

[12] Note that this assumption of uniform probability over the space of possible target functions is not the same as assuming a uniform probability over the space of possible error values.

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

This case in which $P(f \mid \theta)$ 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 $P(f \mid \theta)$, $P(E \mid h, \theta)$ 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 $P(f \mid \theta)$ 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, $P(E \mid h, \theta)$ 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 $P(f \mid \theta)$ 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 *cannot* 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 θ, or if instead h was chosen based on θ (as in a generalizer).

[13] Note that z is always an integer. This reflects the fact that only certain E values are allowed; in particular, no E value is allowed for which $(n-m)E$ is not an integer.

[14] One could define "random" differently from how it is defined here. For example, considered as a function of f, with θ fixed, $P(f \mid \theta)$ is a vector in $\mathbf{R}^{(r^{(n-m)})}$. The only constraint on this vector is that its components are all ≥ 0 and that the sum of its components equals 1. This means that, a priori, $P(f \mid \theta)$ can live anywhere in a certain simplex T that is a subset of $\mathbf{R}^{(r^{(n-m)})}$. We could now define a "random" probability distribution over T (rather than over F) and estimate $\langle P(E \mid h, \theta) \rangle_{\{T\}}$. For simplicity, in this paper no such alternative definition of "random" will be used.

3. The assumptions needed for in-sample testing to be relevant

Given (2.2), how is it that in our physical universe so many learning techniques that 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 θ} and {the generalizing error for questions outside θ}. In doing so, this section answers issues (4), (6), (7), and (8) from the introduction.

3.1 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 maximum entropy 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).[15,16]

In this section I consider the case in which 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 is relevant to the generalization error:

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

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 labeled 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.[17] 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 θ different h might lead to different $P(E \mid h, \theta)$.

[15]It is important to realize that we still have no a priori basis for believing $P(f \mid \theta)$ depends on f. To *prove* that $P(f \mid \theta)$ is of a certain form (as opposed to collecting empirical evidence to that effect) would require knowledge of physics and psychology far in excess of what we have today.

[16]Apparently humanity has learned to recognize the local maxima in this non-uniform $P(f \mid \theta)$—humans refer to the f at those maxima as being "regular" or "parsimonious." See [5].

[17]Of course, this is a very unlikely $P(f \mid \theta)$. In the real world, $P(f \mid \theta)$ is non-zero for all f consistent with θ.

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 $P(h \mid \theta)$, in apparent violation of common experience.

To resolve these two issues, first make the definition $S(f, h, \{x_i\}) \equiv \sum_{\{i\}} \delta[h(x_i), f(x_i)]$. S is a mapping that takes 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 θ. Using this notation, $S(f, h, X - \theta_X)$ is the number of times $h(x)$ agrees with $f(x)$ for x outside θ; $\mathrm{Er}(f, h, \theta) = 1 - S(f, h, X - \theta_X)/(n - m)$. Similarly, if $\theta \subset f$, then $S(f, h, \theta_X)$ is the number of times h agrees with θ. 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 we know h and θ, then we have determined $P(E \mid h, \theta)$, and counting $S(\theta, h, \theta_X)$ does not have any effect on $P(E \mid h, \theta)$. In other words, despite the fact that we are no longer assuming maximum entropy $P(f \mid \theta)$, (3.1) tells us that it is still true that only h's behavior outside the training set is relevant.[18]

To see why (3.1) might not be the final word on inductive inference, note that we are interested in the "dependence" between $s \equiv S(\theta, h, \theta_X)$ and E. Now such "dependence" is a meaningless concept if s is not allowed to vary. However, s *cannot* vary if both h and θ are fixed. This suggests that, to analyze inductive inference, we should fail to specify h and/or θ, 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 cannot evaluate (3.1) since we do not know $P(f \mid \theta)$ (it is determined by the universe). To get around this problem, we could just make a direct assumption for $P(f \mid \theta)$. That is a *huge* assumption to make, however (which, interestingly enough, does not stop Bayesians from making it). As an alternative, it would be preferable to make one (or more) relatively weak *indirect* assumptions that can "take the place" of knowledge *directly* concerning $P(f \mid \theta)$.

As an example, we can make an "indirect" assumption about $P(f \mid \theta)$ by assuming 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 are going to implement our hypothesis

[18]To understand this intuitively, imagine that we have a generalizer G that tries to reproduce the training set (e.g., let G be back-propagation run on feed-forward neural nets), and use G to construct an input-output surface h that reproduces a provided training set θ. Then the generalization error is unchanged if we replace h with some new function h' that is identical to h for questions outside θ_X but disagrees with h for questions inside θ_X.

function (e.g., a feed-forward neural net) corresponds in a certain way to the probability distribution over target functions in the real world.[19]

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 that are constant (i.e., independent of X). Now assume that we examine only constant hypothesis functions, that is, assume that the space of hypothesis functions we are 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 the training set. It is still true that only h's behavior outside 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 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 $P(E \mid s, \theta)$.[20]

3.2 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 $P(h \mid \theta)$, that is, a $P(h \mid \theta)$ whose support extends over more than one h for a given θ (so that the hypothesis guessed by the generalizer is not 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 θ. 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 $P(h \mid \theta, s')$.[21]

We wish to choose between a hypothesis function output by the generalizer $P(h \mid \theta, s')$ and a hypothesis function output by the generalizer

[19] See [5] for a discussion of how the existence (or lack thereof) of this kind of correspondence affects whether application of Occam's razor results in improved generalization.

[20] This next subsection will *not* show that the knowledge that the hypothesis was chosen according to a particular $P(h \mid \theta)$, by itself, can affect generalization error. After all, such a result would contradict the discussion at the end of section 2.2. Rather it is this knowledge together with the knowledge that $P(h \mid \theta)$ corresponds to $P(f \mid \theta)$ that (the next subsection shows) can affect the probable generalization error.

[21] It is essentially a semantic distinction whether (a) $P(h \mid \theta)$ itself is viewed as a generalizer, with s being an observed level of agreement between the hypothesis output by the generalizer and θ, or (b) whether $P(h \mid \theta)$ is viewed as being only a common substrate over which other generalizers are defined by their s values. What is important is how $P(h \mid \theta)$ is used in the mathematics.

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

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

$$P(E \mid s, \theta) = \frac{\sum_{\{\mathrm{h}\}} \{P(E \mid \mathrm{h}, \theta) \times P(\mathrm{h}, \theta) \times \delta[S(\theta, \mathrm{h}, \theta_X), s]\}}{\sum_{\{\mathrm{h}\}} \{P(\mathrm{h}, \theta) \times \delta[S(\theta, \mathrm{h}, \theta_X), s]\}}. \tag{3.2}$$

Note that one can replace both of the $P(\mathrm{h}, \theta)$ terms in (3.2) (one in the numerator, one in the denominator) with $P(\mathrm{h} \mid \theta)$ terms.

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

Similarly, if $P(h \mid \theta)$ is independent of h, then again $P(E \mid s, \theta)$ is independent of s. This is suggested by the interchange symmetry between F and H (see section 2.2). A formal proof follows.

Proof. First note that, if $P(h \mid \theta)$ is independent of h, (3.2) becomes

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

In a manner similar to that in section 2, we can split the sum over h into two sums, one over the values of $\mathrm{h}(x)$ in which x is within the training set, and one over the values of $\mathrm{h}(x)$ in which x is outside the training set: $\sum_{\{\mathrm{h}\}} = \sum_{\{\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_{n-m}\}} \sum_{\{\mathrm{h}_{n-m+1}, \mathrm{h}_{n-m+2}, \ldots, \mathrm{h}_n\}}$. This allows us to rewrite our denominator as $\sum_{\{\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_{n-m}\}} \sum_{\{\mathrm{h}_{n-m+1}, \mathrm{h}_{n-m+2}, \ldots, \mathrm{h}_n\}} \{\delta(S(\theta, \mathrm{h}, \theta_X), s)\} = r^{n-m} \times \sum_{\{\mathrm{h}_{n-m+1}, \mathrm{h}_{n-m+2}, \ldots, \mathrm{h}_n\}} \{\delta(S(\theta, \mathbf{h}', \theta_X), s)\}$, where $\mathbf{h}'$ is defined as the m-tuple $\{\mathrm{h}_{n-m+1}, \mathrm{h}_{n-m+2}, \ldots, \mathrm{h}_n\}$. Evaluating, we get $r^{n-m} \times C_s^m \times (r-1)^{m-s}$. Since $P(E \mid \mathrm{h}, \theta)$ is only dependent on that part of h outside θ, we can rewrite our numerator in a similar fashion, getting $\sum_{\{\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_{n-m}\}} \{P(E \mid \mathbf{h}, \theta)\} \times \sum_{\{\mathrm{h}_{n-m+1}, \mathrm{h}_{n-m+2}, \ldots, \mathrm{h}_n\}} \{\delta(S(\theta, \mathbf{h}', \theta_X), s)\}$, where $\mathbf{h}$ is defined as the $(n-m)$-tuple $\{\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_{n-m}\}$ and $\mathbf{h}'$ is as before. Therefore when $P(\mathrm{h}, \theta)$ is independent of h, $P(E \mid s, \theta) = r^{m-n} \times \sum_{\{\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_{n-m}\}} \{P(E \mid \mathbf{h}, \theta)\}$, and is independent of s. ∎

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

If $P(E \mid \mathrm{h}, \theta)$ is *not* independent of h (see equation (3.1)), and if $P(h \mid \theta)$ is *not* uniform, then $P(E \mid s, \theta)$ *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 $P(E \mid s, \theta)$) and inductive inference can occur. The precise form of the inductive inference—how $P(E \mid s, \theta)$ depends on s—is determined by the form of $P(f \mid \theta)$ (which determines $P(E \mid \mathrm{h}, \theta)$) and the form of $P(h \mid \theta)$. 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 does not equal $n - m$), and 0 for all other h. For this scenario, the denominator in (3.2) equals $1/2$ if $s = m$ or 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 $\mathrm{argmax}_E P(E \mid s, \theta)$, or as $\sum_{\{E\}}\{E \times P(E \mid s, \theta)\}$). 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.

3.3 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 $s = m$ or 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, for $s = m - 1$,

$P(E \mid s, \theta) = 1$ 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 *increases* 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 back-propagation *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 feed-forward 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 in which, 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 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 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 correlate with generalization.)

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 $\sum_{\{\mathrm{h}\}} P(E \mid \mathrm{h}, \theta) \times P(\mathrm{h} \mid s, \theta)$. 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 $P(f \mid \theta)$:

$$P(E \mid s, \theta) = \sum_{\{\mathrm{f},\mathrm{h}\}} P(\mathrm{h} \mid \theta) \times P(\mathrm{f} \mid \theta) \times M_{E,s,\theta}(\mathrm{h}, \mathrm{f}), \tag{3.3}$$

where $M_{E,s,\theta}(h, f) \equiv \kappa_{s,\theta} \times \delta[S(f, h, \theta_X), s] \times \delta[S(f, h, X - \theta_X), (n - m)(1 - E)]$, $\kappa_{s,\theta}$ being a normalization constant set by the condition $\sum_{\{E\}} P(E \mid s, \theta) = 1$.

(The "$\theta \subset f$" condition is being enforced by the $P(\mathrm{f} \mid \theta)$ term. There is no equivalent restriction on the $P(\mathrm{h} \mid \theta)$ 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 $M_{s,E,\theta}(h, f)$ being the metric. Note that if noise is allowed, we must replace the $\delta[S(\mathrm{f}, \mathrm{h}, \theta_X), s]$ term with $\delta[S(\theta, \mathrm{h}, \theta_X), s]$ 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.) Note also that (3.3) holds for any error function (different Er simply lead to different $M_{s,E,\theta}$). Finally, note that (3.3) tells us that the average $\sum_E(E \times P(E \mid s, \theta))$ is an inner product between $P(f \mid \theta)$ and $P(h \mid \theta)$.

Whether 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 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, $P(h \mid \theta)$.

3.4 Ramifications of equation (3.3)

The arguments of the preceding subsections can be used even if one is not 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 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, and so forth. 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 a function only 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, that is, it is determined by the regularizer used.

There are a number of interesting issues raised by (3.3), even if one does not 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) also addresses 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 too many parameters to fit is undesirable, "on average," is determined by whether 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 is 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 is sampled. One possible definition is that a training set θ 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 θ. In other words, θ is not representative of its target function if $P(f \mid \theta)$ is broad and not sharply

peaked, so θ does not 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 θ 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 $P(f \mid \theta)$ is not flat, then if it is close to flat (i.e., if it is broad and not sharply peaked), the correlation between reproduction of θ and generalization from θ is weak. This argument suggests, in accord with empirical observation, that θ must be representative of its target function for one to generalize from θ with low generalization error.

Both this argument concerning how representative θ 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.)

4. 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 is no need to consider explicitly 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 the choice of h can make a difference (see the discussion below equation (3.1)), what h should we use for a given θ? This is issue (5) 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 that 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 that maximizes the probability of zero error.

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

Theorem 2. *To maximize* $P(E = 0 \mid \theta)$, *one should guess a hypothesis function* h *such that the output values of* h *for points outside* θ *are the same as those for the target function* $\text{argmax}_{\text{f} \in F} P(\text{f} \mid \theta)$. *The resultant value of* $P(E = 0 \mid \theta)$ *is* $\max_{\{\text{f} \in F\}} \{P(\text{f} \mid \theta)\}$.

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

$$\begin{aligned} P(E \mid \theta) &= P(E, \theta)/P(\theta) \\ &= \frac{\sum_{\{\mathrm{h},\mathrm{f} \supset \theta\}} \{\delta[\mathrm{Er}(\mathrm{f}, \mathrm{h}, \theta), E] \times P(\mathrm{f}, \mathrm{h}, \theta)\}}{P(\theta)} \\ &= \sum_{\{\mathrm{h},\mathrm{f} \supset \theta\}} \{\delta[\mathrm{Er}(\mathrm{f}, \mathrm{h}, \theta_X), E] \times P(\mathrm{h} \mid \theta) \times P(\mathrm{f} \mid \theta)\} \end{aligned}$$

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

$$\sum_{\{\mathrm{f}_1, \ldots, \mathrm{f}_{n-m}; \mathrm{h}_1, \ldots, \mathrm{h}_{n-m}\}} \{P'(\mathrm{f}_1, \ldots, \mathrm{f}_{n-m}) \times P'(\mathrm{h}_1, \ldots, \mathrm{h}_{n-m}) \times \delta\left(\left[\sum_{i=1}^{n-m} \delta(\mathrm{f}_i, \mathrm{h}_i)\right], z\right)\},$$

where $z \equiv (n-m)(1-E)$ and $P'(\text{event}) \equiv P(\text{event} \mid \theta)$. $P'(\mathrm{f}_1, \ldots, \mathrm{f}_{n-m})$ is determined by the universe. Our job is to determine the optimal $P'(\mathrm{h}_1, \ldots, \mathrm{h}_{n-m})$.

Now make the following notational conventions. A generic $(n-m)$-tuple $\{\mathrm{f}_1, \ldots, \mathrm{f}_{n-m}\}$ is indicated by $\mathbf{f}$, whereas a particular $(n-m)$-tuple is indicated by $\boldsymbol{f}$. Both tuples live in the space $\boldsymbol{F}$. The ith component of such a tuple, $\boldsymbol{f}_i$, equals f_i, where f is the full n-tuple $\{f_1, \ldots, f_n\}$. (The difference between f and $\boldsymbol{f}$ is that f is an n-dimensional vector $\in F$, whereas $\boldsymbol{f}$ is only $(n-m)$-dimensional and lives in $\boldsymbol{F}$.) Similarly, the generic $(n-m)$-tuple $\{\mathrm{h}_1, \ldots, \mathrm{h}_{n-m}\}$ is indicated by $\mathbf{h} \in \boldsymbol{H}$. Note that since f must contain θ, $P'(\boldsymbol{f}) = P'(f)$. However, h need not contain θ, so giving the values of h for all input values outside the training set does not specify h completely, and $P'(\boldsymbol{h})$ need not equal $P'(\mathrm{h})$: $P'(\boldsymbol{h}) = P'(h_1, \ldots, h_{n-m}) = \sum_{\{\mathrm{h}_{n-m+1}, \ldots, \mathrm{h}_n\}} P'(h_1, \ldots, h_{n-m}, \mathrm{h}_{n-m+1}, \ldots, \mathrm{h}_n)$. Also make the definition $T(\boldsymbol{f}, \boldsymbol{h}) \equiv \delta([\sum_{i=1}^{n-m} \delta(f_i, h_i)], n-m)$. (Note that $(n-m)$ is the value of z when $E = 0$.) Finally, define $U(\boldsymbol{h}) \equiv [\sum_{\mathbf{f}} \{P'(\mathbf{f}) \times T(\mathbf{f}, \boldsymbol{h})\}]$. Note that $U(\boldsymbol{h}) \geq 0 \, \forall \, \boldsymbol{h}$.

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

Note that this generalizer $\mathrm{argmax}_{\mathrm{f} \in F} P(\mathrm{f} \mid \theta)$ gives the hypothesis function one should guess to maximize $P(E = 0 \mid \theta)$, which in general is *not* the

same as the "Bayes-optimal" function consisting of the input-output pairs $\{x, \text{argmax}_{y \in Y} \sum_{\text{f}} [P(\text{f} \mid \theta) \times \delta(\text{f}(x), y)]\}$.

If the set H is a proper subset of F, then Theorem 2 does not apply since we cannot guess the mode of $P(f \mid \theta)$. For this case we must instead solve the full problem of finding the (allowed) $\boldsymbol{h}$ that maximizes $U(\boldsymbol{h})$ (see the proof of Theorem 2 for the definition of U and $\boldsymbol{h}$).

Theorem 2 tells us that, under the assumption that $P(\boldsymbol{h} \mid \theta)$ is the same distribution over $\boldsymbol{H}$ as $P(\boldsymbol{f} \mid \theta)$ is over $\boldsymbol{F}$, we should guess the mode of $P(\boldsymbol{h} \mid \theta)$. 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 3 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 2 answers the question of how best to generalize from a given training set, at least for this definition of "best." We can phrase Theorem 2 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 θ by fitting the elements of θ with the surface of minimal integrated curvature going through the elements of θ. Often, however, and especially in neural net learning schemes, the assumption about $P(f)$ is never made explicitly.

5. Meta-generalization

For almost any real-world generalization problem, there are many generalizers that 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 among a set of possible generalizers to use with a particular provided training set θ. 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 in which one generalizes among generalizers [31, 32].

There are several ways to modify the formalism of the previous sections to allow one to address the issue of whether a particular scheme for choosing among generalizers will lead to small generalization error. For simplicity, it is easiest to illustrate the ideas when the scheme for choosing among generalizers is cross-validation. The rest of this section consists of such an illustration. As such, it is an answer to issue (9) 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.

5.1 Cross-validation

Recall that a (deterministic) generalizer is any algorithm that 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 $\in Y$.[22] As 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 one 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 two 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, and so forth. Any set of $g\{i\}$ (defined for all $i > 0$) uniquely fixes a deterministic generalizer and vice versa.[23]

Sometimes I will not explicitly indicate the cardinality of the training set, and will simply write $g(\theta; q)$ when what I really mean is $g\{\text{cardinality of } \theta\}(\theta; q)$. 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 $g(\theta)$ as opposed to $g(\theta; q)$.) 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 $g(\theta; q)$ (or $g(\theta)$, as the case might be).

Assume we have a set of p generalizers, indicated by $g_1, \ldots, g_p$ (so the ith function defining the jth generalizer is indicated by $g_j\{i\}$). Let θ be a training set consisting of m elements. Use the notation that θ_{-i} is the training set θ minus the ith pair, $\{x_i, y_i\}$. Then the cross-validation error associated with the jth generalizer is defined as $\sum_{i=1}^{m} [g_j\{m-1\}(\theta_{-i}; x_i) - y_i]^2$. 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

[22] For a stochastic generalizer, the output is a probability distribution over Y rather than a single element of Y. For the moment, we will ignore such stochastic generalizers. However, it is worth noting that, for most practical situations, at the end of the day one must have a single guess, and therefore one has to have a means of collapsing the set of possible guesses provided by a stochastic generalizer down to a single guess. (Examples of such a collapsing process are averaging the stochastic generalizer's guesses or picking one of the possible guesses according to a pseudo-random number generator.) For such cases, the stochastic generalizer is essentially a sub-algorithm in a larger deterministic generalizer.

[23] Note that for some applications, if X is a k-dimensional space then $g\{i\}$ is undefined for $i < k + 1$, and a generalizer is defined by a set of $g\{i\}$ for $i > k$ rather than for $i > 0$. See [11] for details. Such a scenario will never be explicitly considered in this paper.

error over θ.[24,25] (See [32, 26–30, 34].) Rather than directly assuming a best generalizer, cross-validation assumes something about how best to choose a generalizer.

5.2 Addressing cross-validation via a new event space

It is straightforward 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 $\mathrm{f} \in F : X \rightarrow Y$, 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.[26])

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 is 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: $\mathrm{Er}_V(f, g, \theta) = \sum_{\{x \notin \theta\}} [1 - \delta(g(\theta; x), h(x))]/(n - m)$.[27]

[24]Note that using cross-validation with a fixed set of generalizers is itself a generalizer; cross-validation maps training sets and questions to outputs by setting the output to $g_j\{\theta; q\}$, where g_j is the generalizer that has the lowest cross-validation error for θ. Viewed this way, schemes like cross-validation differ from schemes like back-propagation in two ways. First, they choose among hypothesis functions indirectly rather than directly, by having the direct choice be among generalizers. Second, they do their choosing by means of partitioning the training set. See [32].

[25]One interesting feature of cross-validation is that the $\{g_i\}$, the set of generalizers among which one is choosing, cannot be the set of all possible generalizers. The reason is that, for any training set θ, question $q \in X$, and guess $t \in Y$, there is a generalizer with zero cross-validation error on θ that makes the guess t in response to the question q. (The parallel with the discussion in section 2 is immediate.) In other words, by itself the criterion of zero cross-validation error is under-restrictive in the sense that it cannot uniquely fix how one should generalize. Moreover, it can be proven that there is no subset of the set of all generalizers such that for any training set there is always one (and only one) generalizer from that subset that has zero cross-validation error for that training set. In other words, even in concert with other generalization criteria, one cannot use cross-validation to fix uniquely how one should generalize. See [34] for details.

[26]One might want to have some restrictions on the set of generalizers. For example, in this paper I will usually want to restrict attention to those generalizers whose guessing is invariant under re-ordering of the elements of the training set, so that their guessing is defined even for unordered training sets. It is implicitly assumed in this paper that "G" is the appropriately restricted set of generalizers if any such restrictions are desired.

[27]With all these equivalences between U and V, one is tempted to say that the probability of elements in V is given by the probability of elements in U via $P_V(\mathrm{f} \in F, \mathrm{g} \in G, \theta) \equiv P_U(\mathrm{f} \in F, \mathrm{g}(\theta), \theta)$, where $\mathrm{g}(\theta)$ is a particular hypothesis function. However, there is no a priori reason to require that this equality hold, and in many circumstances it would violate normalization (e.g., there are many generalizers with the same function $g(\theta)$ for a particular θ, so summing $P_V(f, g, \theta)$ over all g might give a number greater than 1 if $P_U(f, h, \theta)$ is normalized.)

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.[28]

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 is not 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 (e.g., [17]), cross-validation and related schemes cannot 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 3 we supplemented the investigation of $P(E \mid h, \theta)$ with an investigation of $P(E \mid s, \theta)$. With our new event space we can similarly 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 by the functional S, and measures the sum, over i, of whether $g(\theta_{-i}; x_i)$ agrees with y_i (i ranging over the m elements making up θ). The conclusion is similar to that reached in section 3 concerning whether reproduction correlates with generalization: whether "cross-validation correlates with generalization" is determined by the behavior of the inner product $P(E \mid s, \theta) = \sum_{\{\mathrm{f,g}\}} P(\mathrm{g} \mid \theta) \times P(\mathrm{f} \mid \theta) \times M'_{E,s,\theta}(\mathrm{g}, \mathrm{f})$, where $M'_{E,s,\theta}(\mathrm{g}, \mathrm{f}) \equiv \kappa'_{s,\theta} \times \delta[S'(\mathrm{f}, \mathrm{g}, \theta), s] \times \delta[S(\mathrm{f}, \mathrm{g}(\theta), X - \theta_X), z]$, $\kappa'_{s,\theta}$ being a normalization constant set by the condition $\sum_{\{E\}} P(E \mid s, \theta) = 1$.

This formula relating how one chooses a generalizer to the generalization error allows us to tackle issues previously unaddressable. For example, it is straightforward to show that, for any training set, there exists a generalizer with zero cross-validation error making any guess desired for questions outside that training set (see [34] and footnote 35). Therefore, the only way 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 to use in our particular physical universe. The explicit form of that assumption is given by the formula at the end of the preceding paragraph.

[28] Note that for cross-validation, $P(g \mid \theta)$ is deterministic; for fixed θ, $P(g \mid \theta)$ is a delta function over G.

5.3 A "meta" event space

Consider again the event space U. Equation (3.1) implicitly tells us the optimal hypothesis function for any training set, as a function of $P(f \mid \theta)$. Unfortunately, we do not know $P(f \mid \theta)$ a priori—it is determined by the physical universe. As mentioned in section 3, one way around this dilemma is to investigate $P(E \mid s, \theta)$ rather than $P(E \mid h, \theta)$, since to evaluate $P(E \mid s, \theta)$ we need only make assumptions about the inner product of $P(f \mid \theta)$ with $P(h \mid \theta)$ rather than about $P(f \mid \theta)$ directly. Another way around the dilemma is to estimate $P(f \mid \theta)$ over all $f \in F$. If one does not want to make a bald-faced assumption for $P(f \mid \theta)$, one might try to estimate $P(f \mid \theta)$ by examining many instances in the physical universe of training sets identical to θ and associated target functions f. This is hardly practical, however. An alternative is to try to extrapolate from $P(f \mid \theta' \neq \theta)$ to make our estimation for $P(f \mid \theta)$.

Unless we wish to examine many different generalizing situations, the only such training sets $\theta' \neq \theta$ at our disposal are the subsets of θ. We can view such subsets as examples that tell us how to generalize for training sets different from θ. For example, we can view the pair $\{(\theta_{-i}; x_i), y_i\}$ as telling us how to generalize when the training set is θ_{-i} and the question is x_i. Therefore our task is to extrapolate, from information concerning how to generalize with subsets of θ (this information being provided by θ), to the case of generalizing with the full training set, θ. Replacing the word "extrapolate" with the synonym "generalize," we see that this is a meta-generalization problem; we wish to generalize how to generalize.

Cross-validation is one, rather primitive way of carrying out this meta-generalization. To consider more sophisticated forms of meta-generalization, we must create a completely new "meta" event space, indicated by U'. Intuitively, in this new event space, in addition to replacing H by G, we replace target functions by "meta" target functions, that is, by "target" generalizers. Similarly, we replace training sets θ (hereafter referred to as "base" training sets), which are examples of how to map inputs to outputs, with meta-training sets, consisting of examples of how to map pairs {base training set, input} to outputs. Such meta-training sets are constructed by partitioning the original base training set. For example, using the partitioning of cross-validation, a base training set θ consisting of the m pairs $\{x_i, y_i\}$ results in a meta-training set consisting of the m pairs $\{(\theta_{-i}; x_i), y_i\}$. Whereas elements of a base training set tell us how to map inputs to outputs, elements of meta-training sets tell us how to generalize.

More formally, as in the preceding sections, we start with an input space X and an output space Y. An arbitrary element of X is now called a "base question," and X is called the "base input space." From X and Y we construct a new "meta" input space X', which consists of all possible ordered sets $\{x_1, y_1, x_2, y_2, \ldots, x_m, y_m; q\}$ for all $m > 0$ (and $m <$ some sufficiently large upper cut-off), where all x_i and q are elements of X, and

all the y_i are elements of Y. In other words, X' consists of all possible {base training set, base question} pairs.

The event space U' is constructed from X' and Y in *exactly* the same manner that U is constructed from X and Y. U' consists of triples. The first element of such a triple is a mapping from the space X' to the space Y. This should be viewed as a "meta" target function, that is, as a target generalizer. Such a target generalizer is indicated by an element $d \in D$. The second triple in U' is also a mapping from X' to Y. It should be viewed as a meta-hypothesis function, that is, a hypothesis generalizer. Such a hypothesis generalizer is indicated by an element $g \in G$. The third element of a triple in U' is a meta-training set, that is, a set of m' pairs $\{x'_i \in X', y_i \in Y\}$. Such a meta-training set is indicated by an element $\omega \in \Omega$. We are interested in probability distributions over U' (see appendix C).

For brevity, I will write $P(d, g, \omega)$ rather than $P_{U'}(d, g, \omega)$, letting the argument list tell us we are interested in probabilities over U'. In an analogous fashion to when we were investigating U, with the event space U' the hypothesis generalizer is under the researcher's control whereas the target generalizer is not. More precisely, it is up to the researcher to set the meta-generalizer $P(g \mid \omega)$. In exact analogy with the similar situation in the analysis of U, this leads us to conclude that $P(g \mid d, \omega) = P(g \mid \omega)$ and $P(d \mid g, \omega) = P(d \mid \omega)$.

5.4 Calculating errors via the meta event space

The meta error function $\mathrm{Er}_{U'}$ is determined by what cost function interests us. Since the error function is what we "measure," it (and only it!) gives physical meaning to the probabilities over our event space. For our current purposes, we are not interested in using an error function $\mathrm{Er}_{U'}(d, g, \omega) \propto \sum_{\{x' \notin \omega_{X'}\}} [1 - \delta(d(x'), g(x'))]$, the direct analogue of the base error function $\mathrm{Er}_U(f, h, \theta_X)$. Rather, since what ultimately interests us is the same kind of generalization error involved in base generalizing, $\mathrm{Er}_{U'}$ is given in terms of the base error function. This is done by the following formula: $\mathrm{Er}_{U'}(d, g, \omega) \equiv \mathrm{Er}_U(d(\theta(\omega)), g(\theta(\omega)), \theta(\omega))$, where $d(\theta(\omega))$ is being viewed as an element of F while $g(\theta(\omega))$ is being viewed as an element of H, and $\theta(\omega)$ is defined as the base training set that generated ω (see appendix C). Expanding, we can write $\mathrm{Er}_{U'}(d, g, \omega) = \sum_{\{x \notin [\theta(\omega)]_X\}} [1 - \delta(d(\theta(\omega); x), g(\theta(\omega); x))]/(n - m)$. From now on, unless noted otherwise, the subscript will be dropped from error functions; the arguments will determine if I mean Er_U or $\mathrm{Er}_{U'}$.

As in section 2, we start by evaluating $P(E \mid g, \omega)$. To do this first we write

$$\begin{aligned} P(E \mid g, \omega) &= \frac{\sum_{\{\mathrm{d} \in D\}} P(\mathrm{d}, g, \omega) \times \delta(S(\mathrm{d}(\theta(\omega)), g(\theta(\omega)), X - [\theta(\omega)]_X), z)}{\sum_{\{\mathrm{d} \in D\}} P(d, g, \omega)} \\ &= \sum_{\{\mathrm{d} \in D\}} P(d \mid \omega) \times \delta[S(\mathrm{d}(\theta(\omega)), g(\theta(\omega)), X - [\theta(\omega)]_X), z], \end{aligned} \tag{5.1}$$

where S is the same function S as in section 3; z is defined in terms of E, n, and m just as in section 2; and use has been made of the fact that $P(d \mid g, \omega) = P(d \mid \omega)$.

In appendix C, (5.1) is used to reduce $P(E \mid g, \omega)$ to a probability over the original event space U. The ensuing analysis shows that a probability distribution $P(d, g, \omega)$ uniform over D results in $P(E \mid g, \omega) = C_z^{(n-m)} \times (r-1)^{(n-m-z)}/r^{(n-m)}$, the same value calculated in section 2 for $P(E \mid h, \theta)$ under the assumption of a uniform distribution over F in the event space U. This is precisely the result we would have expected. Without some non-uniformity in our probability distribution over U', choice of generalizer g is irrelevant.

5.5 Generalizing how to generalize

We can now evaluate the U'-space analogue of the U-space probability $P(E \mid s, \theta)$. We are interested in $P(E \mid s, \omega)$, where here $s = S(\omega, g, \omega_{X'})$ rather than $S(\theta, h, \theta_X)$ (as in the analysis of U). The idea, in direct analogy to the analysis of section 3, is to analyze $P(E \mid s, \omega)$ as a function of E and s. In a manner analogous to when we analyzed U, if we wish we can view this analysis as assuming a "substrate" meta-generalizer $P(g \mid \omega)$, from which we are picking a generalizer with a given value of s. But such an interpretation is not demanded by the math.

Proceeding as in section 3, we write $P(E, s, \omega) = \sum_{\{\mathrm{d,g}\}} P(\mathrm{d}, \mathrm{g}, \omega) \times \delta(S(\omega, \mathrm{g}, \omega_{X'}), s) \times \delta(\mathrm{Er}(\mathrm{d}, \mathrm{g}, \omega), E) = \sum_{\{\mathrm{g}\}} P(E, \mathrm{g}, \omega) \times \delta(S(\omega, \mathrm{g}, \omega_{X'}), s)$. Similarly, $P(s, \omega) = \sum_{\{\mathrm{g}\}} P(\mathrm{g}, \omega) \times \delta(S(\omega, \mathrm{g}, \omega_{X'}), s)$. This allows us to write down the analogue of (3.2):

$$P(E \mid s, \omega) = \frac{\sum_{\{\mathrm{g}\}} P(E \mid \mathrm{g}, \omega) \times P(\mathrm{g}, \omega) \times \delta[S(\omega, \mathrm{g}, \omega_{X'}), s]}{\sum_{\{\mathrm{g}\}} P(\mathrm{g}, \omega) \times \delta[S(\omega, \mathrm{g}, \omega_{X'}), s]} \tag{5.2}$$

As before, if $P(E \mid g, \omega)$ is independent of g, then $P(E \mid s, \omega)$ is independent of s. Carrying on as when we analyzed U, we can now plug (5.1) into (5.2), getting

$$P(E \mid s, \omega) = \sum_{\{\mathrm{d,g}\}} P(\mathrm{d} \mid \omega) \times P(\mathrm{g} \mid \omega) \times M_{E,s,\omega}(\mathrm{d}, \mathrm{g}), \tag{5.3}$$

where $M_{E,s,\omega}(d, g) \equiv \kappa'_{s,\omega} \times \delta(S(\omega, g, \omega_{X'}), s) \times \delta[(S(d(\theta(\omega)), g(\theta(\omega))), X - [\theta(\omega)]_X), z]$, $\kappa'_{s,\omega}$ being a normalization constant set by the condition $\sum_{\{E\}} P(E \mid s, \omega) = 1$. If we insist that only those target generalizers d that reproduce ω have non-zero probability, then we can replace $S(\omega, g, \omega_{X'})$ by $S(d, g, \omega_{X'})$ in the formula for $M_{E,s,\omega}(d, g)$. Once this is done, $M_{E,s,\omega}(d, g)$ becomes a symmetric matrix indexed by d and g.

Under this assumption that d reproduces ω, $P(E \mid s, \omega)$ becomes a non-Euclidean inner product between $P(d \mid \omega)$ and $P(g \mid \omega)$, in direct analogy to the formula for $P(E \mid s, \theta)$ derived in section 3. The conclusion is

similar: the "correspondence" between a particular distribution of hypothesis generalizers (i.e., a particular $P(g \mid \omega)$) and the universe's conditional probability for generalizers, $P(d \mid \omega)$, determines whether the level of agreement between {the generalizing of a generalizer g guessed according to $P(g \mid \omega)$} and {the generalization instances in the meta-training set ω} results in low generalization error.

Note that correlation of reproduction of the meta-training set with generalization supersedes correlation of reproduction of the base training set with generalization. In other words, if the meta-generalizer works, it should be used regardless of how well a particular base generalizer works. For example, even if one has several generalizers for all of whom reproduction correlates with generalization, if in addition cross-validation works, then cross-validation should be used to weed out all but one of the base generalizers.

5.6 Discussion

One could argue about whether and how (5.3) can have ramifications for our physical universe. In particular, one can argue about whether the probability of a target generalizer is a meaningful concept; the physical universe is chocker-block full of target input-output functions, but target generalizers? One way to address this issue is the following: instead of viewing the problem of inductive inference as your being given a target input-output function that is sampled, view the problem as your being given a target generalizer that is sampled. After all, recall that the probability of target input-output functions really indicates some sort of "degree of belief" that we are likely to encounter a sampling of such a function. (The frequentist interpretation of probability, in which we are said to view the probability of target input-output functions as some sort of objective, universe-wide frequency count of such functions, has long been discredited. See in particular [35–37].) Similarly, the probability of target generalizers indicates our degree of belief that we are likely to encounter a sampling of such a generalizer (as opposed to some objective, universe-wide frequency count of such generalizers). Less metaphysically, one can simply note that, for the purposes of this paper, the physical meaning of probabilities is set by their use in error functions: $P(\mathrm{d} \mid \omega)$ is simply whatever distribution gives the experimentally observed[29] function $P(E \mid s, \omega)$ for our hypothesis generalizer substrate $P(\mathrm{g} \mid \omega)$.

Independent of the issue of how to interpret probabilities of target generalizers, the parallel between (3.3) and (5.3) immediately suggests many interesting and novel features of inductive inference. For example, base generalizing phenomena like over-training, the usefulness of reducing the number of free parameters, the need to have a "representative" training set, the usefulness of regularizers, and so forth (see the end of section 3) all have meta-generalizing versions. For example, over-training corresponds to

[29] If one insists on using a frequency-count interpretation of probability, then "experimentally observed" can be taken to mean frequency counts.

reducing cross-validation error so much that generalization suffers. As another example, reducing the number of free parameters might aid the performance of hypothesis generalizers, just as it can aid the performance of hypothesis functions. Similarly, a meta-training set w can be not "representative" if $P(d \mid \omega)$ is broad and flat, in which case using a meta-generalizer to determine how to generalize might not result in low generalization error. In addition, applying regularizer cost functions to hypothesis generalizers might be beneficial, just as applying regularizer cost functions to hypothesis functions can be beneficial.

In addition to resulting in predictions concerning the phenomenology of meta-generalization, the parallel between (3.3) and (5.3) has many other implications. For example, it means that almost any method that is usually viewed as meta-generalizing (e.g., cross-validation [26–30], bootstrap [30], fan generalizers, and time-series analysis [38, 12]) can also be applied directly to (base) generalizing. For example, in techniques akin to the bootstrap method, one does not simply sum, over all partitions of the base training set, the errors in guessing one side of the partition when trained with the other (as one does in cross-validation). Rather, one might look at the probability distribution of such errors. Such a distribution can be estimated by constructing a large ensemble of partitions and collating a histogram of {number of partitions that had a certain error in guessing one side of the partition when trained with the rest of it} vs. {that error value}. The parallel between (3.3) and (5.3) immediately suggests the idea of doing a similar thing with base generalization: rather than simply sum the errors of a hypothesis function at reproducing a training set (the analogy of cross-validation), construct a histogram of {number of questions leading to a given error between the hypothesis function and the target function} vs. {those error values}. In this way one could, for example, estimate whether a difference in the average error at reproducing a training set between two neural nets is statistically significant.

In a similar fashion, the parallel between (3.3) and (5.3) suggests using techniques usually used in base generalization to meta-generalize. For example, one might apply gradient descent in meta-generalization, thereby arriving at a generalizer with low cross-validation error. Alternatively, one might wish to apply Rissanen's minimum description length principle to hypothesis generalizers rather than hypothesis functions; pick the generalizer that when combined with the training set results in the smallest coding length, and use that generalizer to generalize from the training set.

As another possibility, note that using cross-validation to meta-generalize is akin to base generalizing by the following procedure: choose from a set of candidate hypothesis functions the hypothesis function that best reproduces the training set. However, in base generalization one often creates the hypothesis function from the base training set directly with surface-fitters [9–13] rather than by searching over a set of possible hypothesis functions. The parallel between (3.3) and (5.3) suggests meta-generalizing in a similar

fashion, with "meta" surface-fitters rather than with cross-validation. For example, the analogy suggests that, just as one can base generalize by taking as one's hypothesis function the linear combination of basis hypothesis functions with the best fit to the base training set, so might one meta-generalize by taking as one's hypothesis generalizer the linear combination of basis hypothesis generalizers with the best fit to the meta training set. Similarly, just as one might generalize over a set of residuals between a hypothesis function and the base training set, one might try to meta-generalize over a set of residuals between a hypothesis generalizer and the meta-training set. In point of fact, schemes of these types have been investigated before, and often work extremely well in practice [31–33, 40].

Another version of such "meta-surface-fitting" follows from the observation that when base generalizing one often pre-processes the input space, for example to reduce the dimensionality of the input space values fed to the generalizer. In other words, one often maps $X \rightarrow Z$ via a mapping T and then does the generalizing from $Z \rightarrow Y$, where Z has lower dimension than X and yet still (hopefully) captures the salient characteristics of X. Again, the suggestion of (5.3) is to do the same thing when meta-generalizing: reduce X' to some smaller space Z' via some mapping T', and then map Z' to Y. Returning for a moment to the case of base generalizing, if Z is a Cartesian product of subspaces all of which are copies of Y, then T is just a Cartesian product of functions from X to Y. Similarly, if Z' is a Cartesian product of subspaces all of which are copies of Y, then T' is a Cartesian product of functions from X' to Y. In other words, if Z' is a Cartesian product of subspaces all of which are copies of Y, then T' is a Cartesian product of generalizers. Now note that, with such a space Z', the mapping from Z' to Y can be done via a base generalizer (just like the mapping from Z to Y). Therefore such pre-processing via T' constitutes a very interesting means of meta-generalizing; under such schemes one *combines* generalizers (which make up T'), by piping their guesses *through another generalizer* (the mapping from Z' to Y). As an example, with such a scheme one can use surface fitters to combine decision tree generalizers, neural nets, and surface-fitters.

These kinds of "meta-surface-fitting" are examples of the technique of stacked generalization [32]. One important aspect of the procedure of stacked generalization is that the entire procedure can itself be stacked, that is, fed into yet another generalizer. In terms of the parallel between (3.3) and (5.3), this simply means that since one can jump to a meta-realm (i.e., go from (3.3) to (5.3)) and maintain essentially the same formal structure for calculating the value of the error function, so can one jump to a meta-meta-realm. In meta-generalization one generalizes among generalizers. In meta-meta-generalization, one generalizes among meta-generalizers. An example of such meta-meta-generalization is to decide between using cross-validation and some other scheme for choosing among generalizers by partitioning the meta-training set and then calculating whether cross-validation or the other scheme results in lower generalization error.

6. Conclusion

This paper addresses the question of how and why in-sample testing can correlate with generalization error off of the testing set. In addressing this question, a formalism is developed that can be viewed as an extension of the conventional Bayesian formalism. This formalism can be used to address *all* 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, and so forth.

The most important feature of the formalism presented in this paper is that it uses an extremely low-level event space, consisting of triples of {target function, hypothesis function, training set}. In much previous theoretical research peculiar definitions of machine-learning issues have been used to allow the researcher to (try to) shoehorn a pet formalism into the field of machine learning. Using the extremely low-level event space employed in this paper ensures that no such "sleight of hand" occurs; all machine-learning issues are addressed directly and overtly. For example, use of this event space ensures that one's assumptions about the probability distribution of target functions in the physical universe are explicit.

Most (if not all) other formalisms that have been constructed to address machine learning (e.g., PAC) are special cases of the formalism presented in this paper, and are capable of addressing only a subset of the issues addressed 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 conventional Bayesian formalism uses only two-thirds of the full event space exploited in this paper; it has probabilities involving target functions and training sets, but hypothesis functions are ignored. (Despite use of the word "hypothesis," what a Bayesian would call "P(hypothesis such-and-such)" is equivalent not to what is called "P(hypothesis function such-and-such)" in this paper, but rather to what is called "P(target function such-and-such).") As a result, *by construction* the Bayesian formalism is incapable of deriving results like equation (3.3).

Some of the conclusions of this paper are:

1. If one assumes a maximum-entropy universe, then it is entirely irrelevant what hypothesis function one uses and there is no correlation between reproduction of the training set and off-training set generalization error. Since such a universe cannot be ruled out on an a priori basis, it is theoretically impossible to come to any conclusions about how to generalize using *only* a priori reasoning.

2. Given (1), empirical evidence that the choice of hypothesis function *is* relevant serves as empirical evidence that the probability distribution of target functions in our universe is not uniform (i.e., has sub-maximal entropy). Peaks in this distribution presumably correspond to what humans call "parsimonious" or "regular" target functions.

3. Assuming a physical universe with less than maximal entropy, not only is choice of hypothesis function relevant, but the correspondence between the hypothesis function and the training set can correlate with the error of the hypothesis function off of the training set. (Interestingly, a necessary condition for such a correspondence is that the hypothesis function was *not* chosen at random, according to a uniform distribution, independently of the training set. This is contrary to the suggestion of some researchers that choosing the hypothesis function at random is a sufficient condition for such correspondence.) The correlation can be written as a non-Euclidean inner product between two vectors, one representing the physical universe and one representing the generalizer used to create the hypothesis function. So far as it assumes that such a strategy results in improved generalization, *any* generalizer that strives to create a hypothesis function in agreement with the training set (e.g., back-propagation run on neural nets) is implicitly making an assumption about this non-Euclidean inner product.

4. The inner product mentioned in (3) can be used to demonstrate over-training, the utility of minimizing the number of free parameters in the hypothesis functions, difficulties arising from the use of training sets that are not "representative" of their target function, the utility of regularizers, and so forth. That inner product can also be used to demonstrate some very counter-intuitive phenomena, such as situations in which the *worse* the fit between the hypothesis function and the training set, the *better* the generalization.

5. If one knows the distribution of target functions beforehand, then for a given definition of generalization error one can build an optimal generalizer. For example, to maximize the probability that the chosen hypothesis function has perfect generalization, one should guess the hypothesis function lying at the mode of the distribution of target functions. (In general that function is not the same as the "Bayes-optimal" function.)

6. A "meta-formalism" can be constructed for addressing issues like how to combine generalizers, how and when cross-validation works, and so forth. This meta-formalism is formally equivalent to the formalism alluded to in (1) through (4), and therefore all the conclusions in (1) through (4) carry over to the meta-realm. For example, just as one can have "over-training" in which one over-minimizes error at reproducing the training set and thereby increases generalization error, so might one "over-minimize" cross-validation error and thereby increase generalization error.

Future research involves:

1. Investigating in more detail some of the issues discussed in the text: how and when over-training occurs, how and when regularizers are

helpful, how and when "over-regularizing" occurs, how and when limiting the number of free parameters is helpful, how and when it helps to choose a training set that is "representative" of the target function, and so forth.

2. Using the answers to (1) to improve real-world generalizers. In particular, following up on the suggestions in the text on how to avoid overtraining, and using the analysis of the utility of limiting the number of free parameters to deduce the relationship between what simplicity measure one should use and what assumptions one makes about the physical universe.

3. Answering the "meta" versions of all the questions in (1), and using these answers to improve real-world generalization, just as in (2).

4. Extending the analysis to different error functions, in particular to error functions involving a metric over the output space; extending the analysis to continuous input and output spaces; and extending the analysis of section 4 to different measures of "best" generalization error.

5. Extending the analysis to situations in which (like in PAC and like in appendix B) one sums over training sets as well as over target functions, and/or in which one has error functions that run over the elements of the training set as well as off of it.

6. Reconciling the analysis in this paper with the analysis in [5] concerning Occam's razor and uniform simplicity measures (the optimal measures of the simplicity of a hypothesis function).

7. Investigating whether and when requiring various invariances of the generalizer (as in [11]) results in improved generalization. Investigating whether and when stacked generalization [32] results in improved generalization, whether and when fan generalizers [38] result in improved generalization, and so forth.

8. Extending the analysis to fields closely related to machine learning (e.g., time-series analysis).

Appendix A. The ramifications of requiring that $P(h \mid f, \theta)$ be independent of f

Since $P(h \mid f, \theta)$ is undefined for $\theta \not\subset f$ when there is no noise, this appendix works with the restricted requirement that $P(h \mid f, \theta) = P(h \mid f', \theta) \, \forall \, h$, θ, f, and f' such that $f \supset \theta$ and $f' \supset \theta$.

First note that this requirement follows from the requirement that $P(h \mid f, \theta) = P(h \mid \theta) \, \forall \, h$, θ, and $f \supset \theta$. Therefore, to prove the equivalence of these requirements, we must prove the converse: $\{P(h \mid f, \theta) = P(h \mid f', \theta) \, \forall \, h$, θ, f, and f' such that $f \supset \theta$ and $f' \supset \theta\} \Rightarrow \{P(h \mid f, \theta) = P(h \mid \theta) \, \forall \, h$, θ, and $f \supset \theta\}$. To prove this, define k_f as the ratio $P(h \mid f, \theta) / P(h \mid \theta)$

where $f \supset \theta$. Our task is to prove that $\{P(h \mid f, \theta) = P(h \mid f', \theta) \forall h, \theta, f$, and f' such that $f \supset \theta$ and $f' \supset \theta\}$ implies k_f is constant and equals 1. Expanding both the conditional probability in the numerator and the one in the denominator, we get

$$k_f^{-1} = \frac{\sum_{\{\mathrm{f}\}} P(h, \mathrm{f}, \theta) \times \sum_{\{\mathrm{h}\}} P(\mathrm{h}, f, \theta)}{P(h, f, \theta) \times \sum_{\{\mathrm{h,f}\}} P(\mathrm{h}, \mathrm{f}, \theta)}. \tag{A.1}$$

Now rewrite $\{P(h \mid f, \theta) = P(h \mid f', \theta) \forall h, \theta, f$, and f' such that $f \supset \theta$ and $f' \supset \theta\}$ as

$$\frac{P(h, f, \theta)}{P(h, f', \theta)} = \frac{\sum_{\{\mathrm{h}\}} P(\mathrm{h}, f, \theta)}{\sum_{\{\mathrm{h}\}} P(\mathrm{h}, f', \theta)}$$

$\forall h, \theta, f$, and f' such that $f \supset \theta$ and $f' \supset \theta$. Plugging this into (A.1), we get

$$k_f^{-1} = \sum_{(\mathrm{f})} \left\{ \frac{\sum_{\{\mathrm{h}\}} P(\mathrm{h}, \mathrm{f}, \theta)}{\sum_{\{\mathrm{h}\}} P(\mathrm{h}, f, \theta)} \right\} \times \frac{\sum_{\{\mathrm{h}\}} P(\mathrm{h}, f, \theta)}{\sum_{\{\mathrm{h,f}\}} P(\mathrm{h}, \mathrm{f}, \theta)}.$$

This just equals 1, however, independent of f, which proves the supposition

$$P(h \mid f, \theta) = P(h \mid \theta) \ \forall h, \theta, \text{ and } f \supset \theta. \tag{A.2}$$

A similar proof, without the "$\supset \theta$" conditions, holds when noise is allowed (so that $P(h \mid f, \theta)$ is defined even for $\theta \not\subset f$).

Now rewrite $P(h \mid f, \theta) = P(h \mid \theta)$ as $P(h, f, \theta)/P(f, \theta) = P(h, \theta)/P(\theta)$. This last equality can be rewritten as $P(h, f, \theta)/P(h, \theta) = P(f, \theta)/P(\theta)$, which is equivalent to $P(f \mid h, \theta) = P(f \mid \theta)$. Therefore,

$$P(f \mid h, \theta) = P(f \mid \theta) \ \forall h, \theta, \text{ and } f. \tag{A.3}$$

A number of interesting corollaries follow immediately from (A.3). For example, (A.3) means that $P(f \mid \theta) \times P(h \mid \theta) = P(f \mid h, \theta) \times P(h \mid \theta)$, that is, the joint probability $P(f, h \mid \theta)$ factors:

$$P(f, h \mid \theta) = P(f \mid \theta) \times P(h \mid \theta) \ \forall h, \theta, \text{ and } f. \tag{A.4}$$

Note that, in both (A.3) and (A.4), even when there is no noise we do not need to specify $\theta \subset f$ since, for $\theta \not\subset f$, (A.3) and (A.4) both reduce to the equality $0 = 0$.

Appendix B. A rigorous investigation of the "coin-tossing proof" of inductive inference

This appendix is a rigorous investigation of the "coin-tossing proof" of inductive inference outlined in the introduction. This "proof" makes at least two crucial assumptions, both of which are hard to justify from the point of view of real-world machine learning. The first is that when measuring generalization error one allows questions from within the training set. The second

is that the training set is unknown; we only know how many times it agrees with a hypothesis function. However, even given these two assumptions—both implicit in the coin-tossing argument—as this appendix shows we do not recover the conclusion suggested in the introduction (i.e., these assumptions do not lead to Laplace's law of succession for generalization).

For simplicity, let Y be $\{0,1\}$, and let X be the n integers $\{1,2,\ldots,n\}$. $P(f,h,\theta) = 0$ if $\theta \not\subset f$, as usual. Assume also that training sets are chosen, independently of h, according to a uniform sampling distribution over X with repeats allowed; in other words, training sets θ consist of any finite ordered set of pairs $(x_i \in X, y_i \in Y)$ (with or without repeats), and $P(f,h,\theta_1) = P(f,h,\theta_2)$ if both θ_1 and θ_2 contain the same number of elements, all of which are chosen from f.[30,31] Let the cardinality of the training set be m, as usual. Use an error function independent of θ: $\mathrm{Er}(f,h) = \sum_{\{x \in X\}}\{1 - \delta(f(x),h(x))\}/n \equiv \sum_{i=1}^{n}\{1 - \delta(f_i,h_i)\}/n$.

We are interested in $P(\mathrm{Er(f,}h) = E \mid h,\ \theta$ has m elements, and s of the elements of θ agree with $h)$. Note that θ itself is not known, that is, it is not an argument in this probability distribution. Write this distribution symbolically as $P(A \mid B,C,D)$. Bayes' theorem tells us that $P(A \mid B,C,D) \propto P(D \mid A,B,C) \times P(A \mid B,C)$, where the proportionality constant is independent of A. $P(D \mid A,B,C)$ is $P(\{s$ of the elements of θ agree with $h\} \mid \mathrm{Er(f,}h) = E,\ h,\ \theta$ has m elements). This can be written out as

$$\frac{\sum_{\{\mathrm{f;Er(f,}h)=E\}} \sum_{\{\theta \subset \mathrm{f};m\}} \{\delta(S(\theta,h,\theta_X),s) \times P(h,\mathrm{f},\theta)\}}{\sum_{\{\mathrm{f;Er(f,}h)=E\}} \sum_{\{\theta \subset \mathrm{f};m\}} \{P(h,\mathrm{f},\theta)\}},$$

where by $\{\theta \subset \mathrm{f};m\}$ is meant all training sets θ with m elements all chosen from f; $S(\theta,h,\theta_X)$ is the number of agreements between θ and h (see section

[30]Note that since repeats are being allowed, if training sets were unordered then a uniform sampling distribution over X would *not* result in $P(\mathrm{f,h},\theta_1) = P(\mathrm{f,h},\theta_2)\,\forall\,\theta_1$ and θ_2 of the same cardinality and both $\subset$ f. For example, if training sets were unordered, then if θ_1 contained three elements with three distinct X components whereas θ_2 contained three elements two of which shared the same X component, then a uniform sampling distribution over X would give $P(f,h,\theta_1)/P(f,h,\theta_2) = 3!/3 = 2$.

[31]A sampling assumption concerns $P(\theta \mid f)$. For example, the uniform sampling assumption introduced in section 2 assumes that $P(\theta \mid f)$ is independent of θ for all allowed θ. The assumption $\{P(f,h,\theta_1) = P(f,h,\theta_2)$ if both θ_1 and θ_2 contain the same number of elements, all of which are chosen from $f\}$ is more than just a sampling assumption, however. To see this, write $P(f,h,\theta) = P(f,h \mid \theta) \times P(\theta) = P(f \mid \theta) \times P(h \mid \theta) \times P(\theta)$ (due to (A.4)) $= P(\theta \mid f) \times P(f) \times P(h \mid \theta)$. We are not making a uniform sampling assumption; rather we are assuming that $P(\theta \mid f)$ is chosen to cancel out the θ dependence of the generalizer $P(h \mid \theta)$. This is clearly a somewhat peculiar assumption to make. Unfortunately, this assumption is not just a side-effect of appendix B's formalization of the coin-tossing argument. The coin-tossing argument presented in section 1 implicitly fixes h first; *after this* θ is chosen, and we want it to be chosen according to a uniform distribution over X. However, by assumption h is the hypothesis function output by the generalizer after training on θ. Therefore, knowledge of h will tell us something about what θ can be, that is, it will introduce non-uniformities in probability distributions over θ. To get the distribution over θ to be uniform despite knowledge of h, $P(\theta \mid f)$ must compensate for the non-uniformity introduced by that knowledge of h.

3.1); and by $\{\mathrm{f}; \mathrm{Er}(\mathrm{f}, h) = E\}$ is meant the set of all target functions that have error E with hypothesis function h.

We are assuming that $P(\mathrm{f}, h, \theta)$ is independent of θ, so long as θ agrees with f and has m elements. Therefore, $P(\{s$ of the elements of θ agree with $h\} \mid \mathrm{Er}(\mathrm{f}, h) = E$, h, θ has m elements) becomes

$$\frac{\sum_{\{\mathrm{f};\mathrm{Er}(\mathrm{f},h)=E\}}[P(\mathrm{f}, h, \_) \times \sum_{\{\theta \subset \mathrm{f};m\}}\{\delta(S(\theta, h, \theta_X), s)\}]}{\sum_{\{\mathrm{f};\mathrm{Er}(\mathrm{f},h)=E\}}[P(\mathrm{f}, h, \_) \times \sum_{\{\theta \subset \mathrm{f};m\}}\{1\}]},$$

where by $P(\mathrm{f}, h, \_)$ is meant $P(\mathrm{f}, h, \theta)$ for any m-element θ in agreement with f.

Since repeats are allowed and training sets are ordered, $\sum_{\{\theta \subset f;m\}}\{1\} = n^m$ for any f. Now examine a function f such that $\mathrm{Er}(f, h) = E$. To calculate $\sum_{\{\theta \subset f;m\}}\{\delta(S(\theta, h, \theta_X), s)\}$, relabel the elements of X so that h disagrees with f on the elements $1, 2, \ldots, nE$, and h agrees with f on the remaining elements $nE+1, \ldots, n$. To evaluate the sum we must calculate the number of ordered sets of m X values, s of whose elements are in the set $\{nE+1, \ldots, n\}$. This tells us that $\sum_{\{\theta \subset f;m\}}\{\delta(S((\theta, h, \theta_X), s))\} = C_s^m \times [n - (nE + 1) + 1]^s \times [nE]^{(m-s)} = C_s^m \times [n]^m \times [1 - E]^s \times [E]^{(m-s)}$.

These values for $\sum_{\{\theta \subset f;m\}}\{1\}$ and $\sum_{\{\theta \subset f;m\}}\{\delta(S((\theta, h, \theta_X), s))\}$ are both independent of f. Therefore, $P(\{s$ of the elements of θ agree with $h\} \mid \mathrm{Er}(\mathrm{f}, h) = E$, h, θ has m elements$) = C_s^m \times [1-E]^s \times [E]^{(m-s)}$. This is exactly the value of $P(E \mid D)$ calculated in the introduction for the coin-tossing problem ("D" there meaning the provided data). Also as in the coin-tossing problem, here our result is independent of any assumptions concerning the probability distribution of target functions.

When discussing the coin-tossing problem in the introduction, we noted that for this $P(E \mid D)$, *if we assume a uniform prior* $P(E)$, then $\langle E \rangle|_D$, the expectation value of E subject to the constraint of the data D, equals $(s+1)/(m+2)$. The analogous assumption here to a uniform prior $P(E)$ is to assume that $P(A \mid B, C) = P(\mathrm{Er}(\mathrm{f}, h) = E \mid h, \theta$ has m elements$)$ is uniform (i.e., has the same value for all values E). However, $P(\mathrm{Er}(\mathrm{f}, h) = E \mid h$, θ has m elements) equals

$$\frac{\sum_{\{\mathrm{f};\mathrm{Er}(\mathrm{f},h)=E\}} \sum_{\{\theta \subset \mathrm{f};m\}}\{P(\mathrm{f}, h, \theta)\}}{\sum_{\{\mathrm{f}\}} \sum_{\{\theta \subset \mathrm{f};m\}}\{P(\mathrm{f}, h, \theta)\}}.$$

Since the $\sum_{\{\mathrm{f}\}}$ in the numerator has an extra condition on f not present in the $\sum_{\{\mathrm{f}\}}$ in the denominator, although we *can* just pull $P(\mathrm{f}, h, \theta)$ out of the $\sum_{\{\theta\}}$ like we did previously, we do *not* get a result independent of $P(\mathrm{f}, h, \theta)$. Therefore we have to make an assumption about $P(\mathrm{f}, h, \theta)$. The most reasonable (i.e., least informative) a priori assumption is that $P(\mathrm{f}, h, \theta)$ is independent of f (the exact same assumption we made in section 2.1). Under this assumption, we can rewrite our quotient of sums as

$$\frac{\sum_{\{\mathrm{f}\}}[\{\delta[S(\mathrm{f}, h, X), n - nE]\} \times \sum_{\{\theta \subset \mathrm{f};m\}}\{1\}]}{\sum_{\{\mathrm{f}\}} \sum_{\{\theta \subset \mathrm{f};m\}}\{1\}}.$$

As before, $\sum_{\{\theta \subset \mathrm{f}; m\}} \{1\} = n^m$, independent of f. Therefore we only need to calculate $\sum_{\{f\}} \{\delta[S(\mathrm{f}, h, X), n - nE]$ to see if $P(\mathrm{Er}(\mathrm{f}, h) = E \mid h, \theta$ has m elements) is indeed uniform. To evaluate this sum, we must count how many target functions f agree $n - nE$ times with a given hypothesis function h. Simple combinatorics gives us $C^n_{nE} \times (r-1)^{(nE)}$. This expression is not uniform over E. Therefore we have a constructive proof that the coin-tossing proof has dubious validity; there exists a (repeat-allowing) sampling assumption that, together with the benchmark distribution $P(\mathrm{f}, h, \theta)$, does not result in Laplace's law of succession for generalization.

As a final comment on this problem, note that calculating $P(E \mid h, m, s)$ is a somewhat odd thing to do. To perform the calculation we must sum over different training sets. However, the hypothesis function h is being held fixed. It is as though we are assuming the generalizer makes the same guess h regardless of θ. (This oddity obtains whether or not we use a sampling assumption that implies $P(\theta \mid f)$ is independent of f; see footnote 31.) Perhaps a more natural distribution to calculate is $P(E \mid m, s)$. However, this distribution can be very nasty indeed. Consider the case in which, with probability 1, the target function is the constant function $Y = 1$, and the generalizer is the following rule: for questions contained in the training set, guess in agreement with the training set s times (disagreeing $m - s$ times), and for all other questions, guess a 0. For this scenario the hypothesis function guessed by the generalizer disagrees with the target function for every question outside the training set, for every training set. Although $\mathrm{Er}(f, h)$ for this problem will decrease as s increases (in loose accord with the implication of the naive version of the "coin-tossing problem" presented in the text), for $q \notin \theta_X$—the questions of interest—the error is always *maximal*.

Appendix C. Miscellaneous characteristics of meta-generalization

This appendix fleshes out the discussion in section 5 of meta-generalization in the event space U'.

First, note that for the purposes of this paper we wish to restrict our attention to those meta-training sets that are produced via a cross-validation-type partitioning of a base training set θ. Second, note that θ is assumed to have been made with the same sampling assumption as in the previous sections. We can formally express these facts via the following restrictions on the probability distribution over U'.

Write $\omega \equiv \{x'_i, y_i\}$, where $1 \leq i \leq m'$ for some counting number m'. Also write x'_i as a set product of a question component and a training set component: $x'_i \equiv \{x'_i\}_q \times \{x'_i\}_{\mathrm{ts}}$. $\{x'_i\}_q$ is a single base input-space value, and $\{x'_i\}_{\mathrm{ts}}$ is a base training set. We require that $P(d, g, \omega)$ be zero unless (1) the unordered set of $X \times Y$ pairs given by $\theta(\omega) \equiv \{x'_i\}_{\mathrm{ts}} \times (\{x'_i\}_q, y_i)$ is independent of i, and (2) the elements of ω are those formed by cross-validation-type partitioning of $\theta(\omega)$. These two restrictions formally express the fact that we restrict our attention to those meta-training sets that are produced via a cross-validation-type partitioning from a base training set θ.

To formally express the fact that we assume the same sampling assumption as in the previous sections, we can also require that $P(d, g, \omega)$ is zero unless (3) $\theta(\omega)$ could have been made using that sampling assumption over X for base training sets discussed in the previous sections. (Though note that, formally speaking, an ordering must be imposed on $\theta(\omega)$ before it can serve as a base training set.)

For an appropriate sampling assumption, requirement (3) means that the elements of $\theta(\omega)$ have no duplications in their X components. It is interesting to note that as a result, if so desired, X' can be defined as a single, $(n+2)$-dimensional space. The idea is that the first $n+1$ components of an element of the space X' represent a single argument list for a generalizer (that is, they represent a base question together with a base training set), whereas the last component gives ordering information. More precisely, the first component of an element of X' can only take on one of the n values of X; it represents a base question. The next n components represent the base training set. Each of these next n components take on any one of $r+1$ values. The first r of these values are the r values of Y, and the last one is a special value, "blank." The number of non-blank values in these n components of $x' \in X'$ is m, the cardinality of a base training set. These m non-blank Y values of x' give the Y values of the elements of the base training set. The components of x' in which they occur give the corresponding X values of the elements of the base training set. For example, with $X = \{1, 2, 3, 4, 5\}$ and $Y = \{0, 1\}$, $\{3; \text{blank}, 1, \text{blank}, 1, 0\}$ represents the element of X' with question (3) and (unordered) base training set $\{2, 1; 4, 1; 5, 0\}$. The last of the $n+2$ components is an integer coding for the ordering of the m $\{x_i, y_i\}$ pairs given in components 2 through $n+1$.

In some circumstances one might wish to impose a particular sampling assumption over the allowed region of X' (and thereby over the allowed ω). As another possibility, in some circumstances one might wish to disallow meta-training sets ω for which $\theta(\omega)$ has too low a cardinality (see footnote 23). For current purposes, we have no need to impose any assumptions of these types.

In the text it is claimed that, under the assumption of a uniform distribution over target generalizers, $P(E \mid g, \omega)$ has the same value that $P(E \mid h, \theta)$ has under the assumption of a uniform distribution over target functions. The proof follows.

Proof. Given a particular meta-training set ω, index the target generalizers $d \in D$ as d_{ij}. The first index, i, fixes the value of the function $d(\theta(\omega); x \notin [\theta(\omega)]_X)$, taking (some) base input values to output values. The second index, j, runs over all generalizers with identical first index. Together, i and j uniquely fix the generalizer. So, for example, $d_{ij}(\theta(\omega); x) = d_{ik}(\theta(\omega); x) \, \forall i, j, k, x \notin [\theta(\omega)]_X$. However, if $x \in [\theta(\omega)]_X, d_{ij}(\theta(\omega); x)$ does not necessarily equal $d_{ik}(\theta(\omega); x)$ for $k \neq j$. Similarly, nothing is implied about the relation between d_{ij} and d_{ik} when they are trained on a training set other than $\theta(\omega)$. In fact, if $k \neq j$, then it *must be* that the

generalizers d_{ij} and d_{ik} differ either when trained on $\theta(\omega)$ and asked some question $x \in [\theta(\omega)]_X$, or when trained on a training set other than $\theta(\omega)$ and asked some question $x \in X$.

When we just wish to refer to the function $d(\theta(\omega); x \notin [\theta(\omega)]_X)$, without specifying what generalizer created this function (i.e., without specifying j), we will write $d_{i;\omega}$. We can now rewrite (5.1) as

$$\begin{aligned} P(E \mid g, \omega) &= \sum_{i,j} P(d_{ij} \mid \omega) \times \delta(S(d_{i;\omega}, g(\theta(\omega)), X - [\theta(\omega)]_X), z) \\ &= \sum_{i} [\delta(S(d_{i;\omega}, g(\theta(\omega)), X - [\theta(\omega)]_X), z) \times \sum_{j} P(d_{ij} \mid \omega)] \\ &= \sum_{i} [\delta(S(d_{i;\omega}, g(\theta(\omega)), X - [\theta(\omega)]_X), z) \times P(d_{i;\omega} \mid \omega)], \end{aligned}$$

where by $P(d_{i;\omega} \mid \omega)$ is meant the probability that the target generalizer produces the function $d_{i;\omega}$ when taught with $\theta(\omega)$, given the information of the meta training set ω. Note that $d_{i;\omega}$ is defined (only) for questions outside $\theta(\omega)$. This means that the quantity $\sum_i [\delta(S(d_{i;\omega}, g(\theta(\omega)), X - [\theta(\omega)]_X), z) \times P(d_{i;\omega} \mid \omega)]$ is identical to the quantity $\sum_{\{f_1,\ldots,f_{n-m}\}} \{P(f_1, \ldots, f_{n-m} \mid \theta) \times \delta([\sum_{i=1}^{n-m} \delta(f_i, h(x_i))], z)\}$ from (a slightly rewritten version of) equation (3.1), under the substitution of $P(f_1, \ldots, f_{n-m} \mid \theta)$ for $P(d_{i;\omega} \mid \omega)$, and $h(x)$ for $g(\theta(\omega))$. The conclusions are therefore the same: if $P(d_{i;\omega} \mid \omega)$ does not vary with i (corresponding to $P(f_1, \ldots, f_{n-m} \mid \theta)$ not varying with f, the case investigated in section 2.2), then $P(E \mid g, \omega)$ is equal to $C_z^{(n-m)} \times (r-1)^{(n-m-z)}/r^{(n-m)}$. ∎

Appendix D. The ramifications of the various "sleights of hand" in the PAC formalism

This appendix discusses how the various sleights of hand going into the PAC formalism make it an inappropriate framework for addressing machine learning.

The powerful PAC formalism [18–21, 41] has been promoted by many researchers as a framework for addressing generalization. This is despite the many peculiar aspects (from the point of view of generalization) of PAC. In fact, those very features of PAC that serve as its strength as a formalism actually make it ill-suited for the task of addressing generalization. PAC's strength is that it manages to reach conclusions while making very few assumptions. In fact, not only are the theorems of PAC completely independent of the sampling distribution over X, they are also independent of the training set, and they are close to independent of the generalizer used and of the distribution of target functions. All of this would lead one to suspect that PAC can have little direct bearing on real-world issues concerning generalization, especially for those issues discussed in this paper. Further support for such a suspicion comes from PAC's allowing questions to run over the training set, and from its concentrating on the game-theory type issue of whether a strategy achieves success polynomially or exponentially.

In a misleadingly entitled article [20], Blumer et al. have presented striking evidence in support of the view that PAC has little direct relevance to issues of generalization. They show that one version of PAC implies that guessing according to lower complexity measure is a "good" thing (as defined by PAC), *independent of the particular complexity measure used.* This result serves as a reductio ad absurdum of PAC, as far as inductive inference is concerned. For example, this result shows that PAC counsels that choosing between theories according to the alphabetical listing of the creators of the theories is a "good" way to generalize. (Here the complexity measure of a theory is the alphabetical listing of the theory's creator). More prosaically, if one is trying to guess a function from $\{0,1\}^n \rightarrow \{0,1\}$, basing the guess on a training set of m samples of that function, PAC counsels simply to guess 0's for all questions not in the training set.[32] Alternatively, guessing all 1's for questions outside the training set is a "good" strategy. In fact, almost *any* strategy that completely ignores any patterns in the training set is a "good" strategy in the PAC sense.

PAC is a very powerful formalism, with many interesting features. However, one must be extremely careful in trying to apply it beyond its domain of definition, to issues of real-world machine learning.[33,34] Very few empirical studies of real-world machine learning meet the assumptions that go into PAC. Moreover, very few such real-world studies are directly interested in the issues that concern PAC (the concerns of PAC being things like polynomial vs. exponential convergence). A more detailed analysis of this inappropriateness of PAC can be found in [41].

Appendix E: The meaning of $P(f)$

This appendix is a semi-philosophical discussion of what $P(f)$ "really means."

First, note that given *any* well-defined event, I can always slap down a probability distribution on that event. This fact alone means that $P(f)$ is "real" and "exists." In fact, strictly speaking nothing more needs to be said to justify the use of $P(f)$. Nonetheless, there are some other facts that might

[32] This comes about by choosing the complexity measure of a function from $\{0,1\}^n \rightarrow \{0,1\}$ to be the binary decoding of the string of 2^n 0's and 1's defining that function. In Blumer's notation, for such a measure guessing 0's is an Occam algorithm with α equal to 0.

[33] "Real world" meaning problems in which we have no direct control over the target function distribution (that distribution being set by the physical universe). This is meant to contrast with "toy world" problems in which we directly construct the target function distribution.

[34] For example, one of the defining features of PAC is its assumption that both the training set and the testing set were made according to the same sampling distribution $\pi(x)$ over the input space. PAC's strength is that it is "distribution-free"; in other words, it manages to reach conclusions without being told the precise distribution $\pi(x)$. In the real world, without access to the algorithm used to choose them, it is impossible to determine that a particular finite training set and a particular finite testing set were created via the same sampling distribution. On the other hand, if we *do* have access to the algorithm used to choose the data sets, then we know the sampling distribution, in which case there is no reason to handcuff ourselves with a distribution-free formalism like PAC.

make the reader feel more comfortable with the use of $P(f)$. Second, note that $P(\theta \mid f)$ is certainly a "meaningful" quantity. Yet formally speaking, since it is defined as $P(\theta, f)/P(f)$, for $P(\theta \mid f)$ to have "meaning" so must $P(f)$. Third, note that *there is no way* to deal with quantities like off-training set error without distinguishing target functions from hypothesis functions, and in particular there is no way to deal with such quantities in a probabilistic framework without a $P(f)$. Third, note that conventional Bayesian analysis implicitly uses a $P(f)$ (*not* a $P(h)$). Indeed, the whole game in Bayesian analysis is to make "informationless" arguments that uniquely fix $P(f)$. Therefore, implicitly or otherwise, anyone comfortable with conventional Bayesian analysis is already comfortable with $P(f)$.

These arguments notwithstanding, the very fact that Bayesians use $P(f)$ serves as a warning. The reason is that, to a strict Bayesian, "probability" means "*personal* degree of belief." However, it is doubtful that one really wants to interpret $P(f)$ that way, because such an interpretation suggests that the researcher has complete control over $P(f)$. This is an uncomfortable thing to claim even in conventional Bayesian analysis; it is even more so in the context of this paper, which is an analysis of the (common-sense) notion that one's personal biases, which go into making hypothesis functions, do *not* in general equal the "true priors" of the universe, $P(f)$.

At this point it is worth noting that nothing in Cox's derivation of the laws of probability as the unique calculus of inductive logic requires that probability be interpreted as something so subjective as "degree of belief." All that is required is that a probability be interpreted as a means of reasoning in the absence of complete information. And there are other ways of interpreting such a "means of reasoning" that do not imply one can set probability as one wishes with complete impunity (up to the constraints set by the rules of probability).

This paper is not meant to be a treatise on the philosophical issues associated with the question of what probability "really means" (probability of target functions being a special kind of probability). After all, this very issue has bothered some of the greatest minds in mathematics for two centuries. (See in particular [35] and some of the articles in [42] for a discussion of this controversy from the "degree of belief" point of view.) If the reader is uncomfortable with leaving the "true meaning" of $P(f)$ less than completely resolved, then (s)he should simply view this paper as an investigation of the following two-person game:

Person A generates target functions according to some pre-fixed and unalterable distribution $P(f)$, which person A determined before the game started.

A fixed mechanism exists that translates such a target function f into a training set θ. This mechanism may or may not be known, in part or in whole, to person B.

Person B can see θ. On the basis of θ (and only θ), person B guesses a hypothesis function h. The means by which that guessing is done are completely under person B's control.

The goal of person B is to perform the guessing in such a way as to minimize some (pre-determined) error function, $\mathrm{Er}(f, h, \theta)$.

Nobody is allowed to cheat.

Acknowledgments

I would like to thank the members of the Complex Systems group at Los Alamos and especially M. Stein for fruitful discussion. This work was done under the auspices of the Department of Energy.

References

[1] D. Rumelhart and J. McClelland, *Explorations in the Microstructure of Cognition*, volumes I and II (Cambridge, MIT Press, 1986).

[2] J. Holland, *Adaptation in Natural and Artificial Systems* (Ann Arbor, University of Michigan Press, 1975).

[3] J. Rissanen, "Stochastic Complexity and Modeling," *The Annals of Statistics*, **14** (1986) 1080–1100.

[4] J. Rissanen, "A Universal Prior for Integers and Estimation by Minimum Description Length," *The Annals of Statistics*, **11** (1983) 416–431.

[5] D. Wolpert, "The Relationship between Occam's Razor and Convergent Guess," *Complex Systems*, **4** (1990) 319–368.

[6] C. Stanfill and D. Waltz, "Toward Memory-based Reasoning," *Communications of the ACM*, **29** 1213–1228.

[7] T. Poggio and the MIT AI Lab staff, "MIT Progress in Understanding Images," in *Proceedings of the Image Understanding Workshop*, edited by L. Bauman (McLean, VA, 1988).

[8] V. Morozov, *Methods for Solving Incorrectly Posed Problems* (New York, Springer-Verlag, 1984).

[9] D. Wolpert, "A Benchmark for How Well Neural Nets Generalize," *Biological Cybernetics*, **61** (1989) 303–313.

[10] D. Wolpert, "Constructing a Generalizer Superior to NETtalk via a Mathematical Theory of Generalization," *Neural Networks*, **3** (1990) 445–452.

[11] D. Wolpert, "A Mathematical Theory of Generalization: Part I," *Complex Systems*, **4** (1990) 151–200.

[12] J. Farmer and J. Sidorowich, "Exploiting Chaos to Predict the Future and Reduce Noise," Los Alamos report LA-UR-88-901 (1988).

[13] S. Omohundro, "Efficient Algorithms with Neural Network Behavior," Report UIUCSCS-R-87-1331 of the University of Illinois at Urbana-Champaign Computer Science Department (1987).

[14] J. Quinlan, "Induction of Decision Trees," *Machine Learning*, **1** (1986) 81–106.

[15] T. Dietterich, "Machine Learning," *Annual Review of Computer Science*, **4** (1990) 255–306.

[16] R. Duda and P. Hart, *Pattern Classification and Scene Analysis* (New York, Wiley, 1973).

[17] N. Tishby, E. Levin, and S. Solla, "Consistent Inference of Probabilities in Layered Networks: Predictions and Generalization," pages 403–409 in volume II of *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C. (1989).

[18] E. Baum and D. Haussler, "What Size Neural Net Gives Valid Generalization?" *Neural Computation*, **1** (1989) 151–160.

[19] L. Valiant, "A Theory of the Learnable," *Communications of the ACM*, **27** (1984) 1134–1142.

[20] A. Blumer, et al., "Occam's Razor," *Information Processing Letters*, **24** (1987) 377–380.

[21] A. Blumer, et al., "Learnability and Vapnik-Chervonenkis Dimension," *Journal of the ACM*, **36** (1989) 929–965.

[22] H. Sompolinsky and N. Tishby, "Learning from Examples in Large Neural Networks," *Physical Review Letters*, **65** (1990) 1683–1686.

[23] J. Pearl, "On the Connection between the Complexity and Credibility of Inferred Models," *International Journal of General Systems*, **4** (1978) 255–264.

[24] H. Reichenbach, *Experience and Prediction* (Chicago, The University of Chicago Press, 1938).

[25] H. Akaike, "Information Theory and an Extension of the Maximum Likelihood Principle," in *International Symposium on Information Theory*, edited by B. A. Petrov and F. Csaki, editors (Budapest, Akademiai Kiado, 1973).

[26] M. Stone, "An Asymptotic Equivalence of Choice of Model by Cross-validation and Akaike's Criterion," *Journal of the Royal Statistical Society B*, **39** (1977) 44–47.

[27] M. Stone, "Asymptotics for and against Cross-validation," *Biometrika*, **64** (1977) 29–35.

[28] Ker-Chau Li, "From Stein's Unbiased Risk Estimates to the Method of Generalized Cross-validation," *The Annals of Statistics*, **13** (1985) 1352–1377.

[29] M. Stone, "Cross-validatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society B*, **36** (1974) 111–120.

[30] B. Efron, "Computers and the Theory of Statistics: Thinking the Unthinkable," *SIAM Review*, **21** (1979) 460–480.

[31] S. Gustafson, G. Little, and D. Simon, "Neural Network for Interpolation and Extrapolation," Report number 1294-40, University of Dayton, Research Institute, Dayton, OH (1990).

[32] D. Wolpert, "Stacked Generalization," *Neural Networks*, in press.

[33] Xiru Zhang, private communication concerning stacked generalization and protein folding (1991).

[34] D. Wolpert, "A Mathematical Theory of Generalization: Part II," *Complex Systems*, **4** (1990) 201–249. Cross-validation is a special case of the technique of "self-guessing" discussed here.

[35] I. Good, "Kinds of Probability," *Science*, **129** (1959) 443–447.

[36] P. Cheeseman, "In Defense of Probability," pages 1002–1009 in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (1985).

[37] C. Smith and G. Erickson, "From Rationality and Consistency to Bayesian Probability," *Maximum Entropy and Bayesian Methods*, edited by J. Skilling (Norwell, MA, Kluwer Academic Publishers, 1989).

[38] D. Wolpert, "Improving the Performance of Generalizers by Time-series-like Pre-processing of the Training Set," Los Alamos Laboratory Report LA-UR-91-350 (1991).

[39] He Xiangdong and Zhu Zhaoxuan, "Nonlinear Time Series Modeling by Self-Organizing Methods," Report from the Department of Mechanics, Peking University, Beijing, PRC (1990).

[40] G. Schulz, et al., "Comparison of Predicted and Experimentally Determined Secondary Structure of Adenyl Kinase," *Nature*, **250** (1974) 140–142.

[41] D. Wolpert, "The Limitations of the PAC Framework for Addressing Real-World Generalization," in preparation.

[42] *Maximum Entropy and Bayesian Methods* (Norwell, MA, Kluwer Academic Publishers, 1988, 1989, 1990, 1991). See also references therein.

[43] J. Hertz, et al., *Introduction to the Theory of Neural Computation* (Reading, MA, Addison-Wesley, 1991).

[44] D. Schwartz, et al., "Exhaustive Learning," *Neural Computation*, **2** (1990) 374–385.