iii) 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 don't know $P(f \mid \theta)$ a priori - it's determined by the physical universe. As was mentioned in section III, 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 doesn't 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 telling 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 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, i.e., 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 up 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 $\{x_1, y_1, x_2, y_2, ..., x_m, y_m; q\}$ for all $m > 0$ (and < 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 is to be viewed as a "meta" target function, i.e., 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 is to be viewed as a meta-hypothesis function, i.e., 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, i.e., 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're 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 $P(d \mid g, \omega) = P(d \mid \omega)$.

iv) Calculating errors via the meta event space.

The meta error function $Er_{U'}$ 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 aren't interested in using an error function $Er_{U'}(d, g, \omega_{X'}) \propto \Sigma_{\{x' \notin \omega\}} [1 - \delta(d(x'), g(x')]$, the direct analogue of the base error function $Er_U(f, h, \theta_X)$. Rather, since what ultimately interests us is the same kind of generalization error involved in base generalizing, $Er_{U'}$ is given in terms of the base error function. This is done by the following formula: $Er_{U'}(d, g, \omega_{X'}) \equiv Er_U(d(\theta(\omega)), g(\theta(\omega)), [\theta(\omega)]_X)$, where d(.) is being viewed as an element of F while g(.) is being viewed as an element of H, and $\theta(\omega)$ is defined as the base training set which generated ω (see appendix C). Expanding, we can write $Er_{U'}(d, g, \omega_{X'}) = \Sigma_{\{x \notin [\theta(\omega)]_X\}} [1 - \delta(d(\theta(\omega); x), g(\theta(\omega); x)] / (n - m)$. From now on, unless noted otherwise the subscript will be dropped from error functions; the arguments will determine if we mean Er_U or $Er_{U'}$.

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

$$(5.1)\quad P(E \mid g, \omega) = \frac{\Sigma_{\{d \in D\}} P(d, g, \omega) \times \delta(S(d(\theta(\omega)), g(\theta(\omega)), X - [\theta(\omega)]_X), z)}{\Sigma_{\{d \in D\}} P(d, g, \omega)}$$

$$= \Sigma_{\{d \in D\}} P(d \mid \omega) \times \delta[S(d(\theta(\omega)), g(\theta(\omega)), X - [\theta(\omega)]_X), z],$$

where S(., ., .) is the same function S as in section III, z is defined in terms of E, n, and m just as in section II, 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 $P(E \mid g, w) = C_z^{(n-m)} \times (r - 1)^{(n-m-z)} / r^{(n-m)}$, the same value calculated in section II 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.

v) Generalizing how to generalize.

We can now evaluate the U'-space analogue of the U-space probability P(E | s, θ). We are interested in P(E | s, ω), 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 III, is to analyze P(E | s, ω) 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 | ω), 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 III, we write $P(E, s, \omega) = \Sigma_{\{d,g\}} P(d, g, \omega) \times \delta(S(\omega, g, \omega_{X'}), s) \times \delta(Er(d, g, \omega_{X'}), E) = \Sigma_{\{g\}} P(E, g, \omega) \times \delta(S(\omega, g, \omega_{X'}), s)$. Similarly, $P(s, \omega) = \Sigma_{\{g\}} P(g, \omega) \times \delta(S(\omega, g, \omega_{X'}), s)$. This allows us to write down the analogue of (3.2):

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

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

$$(5.3)\quad P(E \mid s, \omega) = \Sigma_{\{d,g\}} P(d \mid \omega) \times P(g \mid \omega) \times M_{E,s,\omega}(d, g),$$

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 $\Sigma_{\{E\}} P(E \mid s, \omega) = 1$. If we insist that only those target generalizers d which 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 III. 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 or not 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.

vi) Discussion.

One could argue about whether and how (5.3) can have ramifications for our physical universe. In particular, one can argue about whether or not 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 to simply say that instead of viewing the problem of inductive inference as your being given a target input-output function which is sampled, simply view the problem as your being given a target generalizer which is sampled. After all, recall that the probability of target input-output functions really means our degree of belief that we are likely to encounter a sampling of such a function. (The frequentist interpretation of probability, in which we'd 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 means 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: $P(d \mid \omega)$ is simply whatever distribution gives the experimentally observed[30] function $P(E \mid s, \omega)$ for our hypothesis generalizer substrate $P(g \mid \omega)$.

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, etc. (see the end of section III), all have meta-generalizing versions. For example over-training corresponds to reducing cross-validation error too much, so 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 ω 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 which 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 doesn't 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 which 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 or not 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 [3,4] to hypothesis generalizers rather than hypothesis functions; pick the generalizer which 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 that hypothesis function which 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, i.e., 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 amongst generalizers. In meta-meta-generalization, one generalizes amongst meta-generalizers. An example of such meta-meta-generalization is to decide between using cross-validation and some other scheme for choosing amongst generalizers by partitioning the meta training set and then calculating whether cross-validation or the other scheme results in lower generalization error.

VI. 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 which 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, etc.

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. They 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 uses only 2/3 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* that 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 can not be ruled out on 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 <u>not</u> chosen at random, 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 which 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 which are not "representative" of their target function, the utility of regularizers,

etc. That inner product can also be used to demonstrate some very counter-intuitive phenomena, e.g., 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, etc. 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 which is "representative" of the target function, etc.

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 over-training, 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. Ex-

tending the analysis of section IV 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 which 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, etc.

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

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.

FOOTNOTES

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.

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 feedforward neural nets (as in [17] and any attempts to apply studies like [18] to the real world). Another slight 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 of the training set; allowing questions to also run over the training set is, at best, obfuscatory.) See appendix D for a discussion of such sleights of hand in the PAC formalism.

3. It should be noted that this limitation of scope is shared by essentially all other theoretical machine-learning research in the literature which does not directly make assumptions about he real universe (e.g., PAC, the statistical mechanical formalism, the analysis of various linear models, etc.). Essentially none of this previous research even *suggests* novel machine-learning algorithms (as is done in this paper in section V), never mind goes on to empirically investigate the real-world utility of such an algorithm. In point of fact, as a general rule any of this previous machine learning research which at first glance appears to have real-world ramifications will, on closer inspection, be found to have none (e.g. [18], cf. [41]).

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 <u>never</u> what we're directly interested in (since training sets are always finite), and therefore arguments concerning such behavior can be extremely misleading. (This is es-

pecially true when those arguments are mae without any concern for bounding the error which comes from applying their results to the finite case.)

6. It should be noted that this requirement is more of a tautology than an “assumption”, given that we’re doing supervised machine learning. (Indeed, this requirement is implicitly made in every single theoretical treatment of machine-learning of which I aware.) Consider changing the target function *while keeping the training data the same*. Since the generalizer only sees the training data (by definition), such a change in the target function provides no change in the information at our disposal telling 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, given that $P(f, h, \theta)$ is zero unless f and θ are mutually consistent.) On the other hand, either the hypothesis function h chosen by us is also determined (at least in part) by θ, or it’s random. In either case, specifying the hypothesis function we guess in addition to specifying θ doesn’t help the universe determine what target function generated θ. This is an intuitive justification of the result that $P(f \mid h, \theta) = P(f \mid \theta)$.

8. Although in the real world we usually have direct access both to the hypothesis function (after all, we constructed it) and to the training set, we don’t 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 *which 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 defining U says that we assume there exists a fixed, unknown target function f which is sampled to create θ. Nonetheless, it is often the case that the researcher has in mind precisely this scenario where 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.

9. $P(\theta \mid f) = P(f, \theta) / P(f) = P(f, \theta) / [\Sigma_{\{w \subseteq f\}} P(f, w)]$, 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' which are consistent with f then implies that $P(\theta \mid f) = \{1 / [\Sigma_{\{\omega \subseteq f\}} 1]\} \equiv k$. (k can be calculated by summing over all allowed training set cardinalities the number of training sets of that cardinality which can be chosen from f. For example, if all training set cardinalities are allowed in U but training sets aren't allowed to contain repeats, then $k^{-1} = \Sigma_{i=1}^{n}\ [n! / i!]$.) 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's also possible to avoid the assumption implicit in an error function concerning how questions are chosen. To do this we must calculate "errors" conditioned on a particular question, i.e., we replace an investigation of the probability of errors of the form $\Sigma_{\{x \notin \theta_X\}} [1 - \delta(f(x), h(x)] / (n - m)$ 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, one must reduce X to the set of m + 1 elements $\{\theta_X \cup q\}$ and then use the original error measure

$\Sigma_{\{x \notin \theta_X\}}$ [1 – δ(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.

11. Note that conventional Bayesian analysis, which doesn't distinguish H from F, can not serve as such an over-arching framework.

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.

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 isn't an integer.

14. One could define "random" differently from how it's defined here. For example, considered as a function of f, with h and θ fixed, P(f, h, θ) is a vector in $\mathbf{R}^{(r^{(n-m)})}$. The only constraint on this vector is that the sum of its components equals Σ_f P(f, h, θ), which is some constant between 0 and 1. This means that, a priori, P(f, h, θ) can live anywhere in a certain space U which 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 $< P(E \mid h, \theta) >_{\{T\}}$. For simplicity, in this paper no such alternative definition of "random" will be used.

16. It's important to realize that we still have no a priori basis to believe P(f | θ) depends on f. To <u>prove</u> that P(f | θ) 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.

17. Apparently humanity has learned to recognize the local maxima in this non-uniform P(f | θ) -

humans refer to the f at those maxima as being “regular” or “parsimonious”. (See [5].)

18. 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.

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

20. See [5] for a discussion of how the existence (or lack thereof) of this kind of correspondence affects whether or not application of Occam’s razor results in improved generalization.

21. This next subsection will <u>not</u> 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 II(ii). Rather it’s this knowledge together with the knowledge that $P(h \mid \theta)$ corresponds to $P(f \mid \theta)$ which (the next subsection shows) can affect the probable generalization error.

22. 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 only being a common substrate over which other generalizers are defined by their s value. What is important is how $P(h \mid \theta)$ is used in the mathematics.

23. 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’s worth not-

ing 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.

24. 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.

25. 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 which has 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 amongst hypothesis functions indirectly rather than directly, by having the direct choice be amongst generalizers. Second, they do their choosing by means of partitioning the training set. See [32].

26. One interesting feature of cross-validation is that the $\{g_i\}$, the set of generalizers amongst which one is choosing, can not 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 which makes the guess t in response to the question q. (The parallel with the discussion in section II is immediate.) In other words, by itself the criterion of zero cross-validation error is under-restrictive in the sense that it can't 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 which has zero cross-validation error for

that training set. In other words, even in concert with other generalization criteria, one can't use cross-validation to uniquely fix how one should generalize. See [34] for details.

27. One might want to have some restrictions on the set of generalizers. (For example, in this paper we 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.

28. 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 $P_V(f \in F, g \in G, \theta) \equiv P_U(f \in F, g(\theta), \theta)$, where $g(\theta)$ is a particular hypothesis function. However there is no a priori reason to require that this equality holds, 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.)

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

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

31. Note that since repeats are being allowed, if training sets were unordered then a uniform sampling distribution over X would <u>not</u> result in $P(f, h, \theta_1) = P(f, h, \theta_2)\ \forall\ \theta_1$ and θ_2 of the same cardinality and both $\subset f$. For example, if training sets were unordered, then if θ_1 contained 3 elements with three distinct X components whereas θ_2 contained 3 elements two of which shared the same

X component, then a uniform sampling distribution over X gives $P(f, h, \theta_1) / P(f, h, \theta_2) = 3! / 3 = 2$.

32. A sampling assumption concerns $P(\theta \mid f)$. For example the uniform sampling assumption introduced in section II 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're 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 I implicitly first fixes h; *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, i.e., 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.

33. 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, with such a measure guessing 0's is an Occam algorithm with α equal to 0.

34. "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 where we directly construct the target function distribution.

35. For example, one of the defining features of PAC is its assuming that both the training set and 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", i.e., it manages to reach conclusions without being told the precise distribution $\pi(x)$. Now 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's no reason to handcuff ourselves with a distribution-free formalism like PAC.

APPENDIX A

The ramifications of requiring that P(h | f, θ) 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 $P(h \mid f, \theta) = P(h \mid f', \theta)\ \forall\ h, \theta, f$ and f' such that $f \supset \theta$ and $f' \supset \theta$.

First note that this requirement follows from the requirement that $P(h \mid f, \theta) = P(h \mid \theta)\ \forall\ h, \theta$, and $f \supset \theta$. Therefore to prove the equivalence of these requirements, we must prove the converse: $\{P(h \mid f, \theta) = P(h \mid f', \theta)\ \forall\ h, \theta, f$ and f' such that $f \supset \theta$ and $f' \supset \theta\} \Rightarrow \{P(h \mid f, \theta) = P(h \mid \theta)\ \forall\ h, \theta$, 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 that k_f is constant and equals 1. Expanding both the conditional probability in the numerator and the one in the denominator, we get

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

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

$$\frac{P(h, f, \theta)}{P(h, f', \theta)} = \frac{\Sigma_{\{h\}} P(h, f, \theta)}{\Sigma_{\{h\}} P(h, f', \theta)} \quad \forall h, \theta, \text{ f and f' such that } f \supset \theta \text{ and } f' \supset \theta.$$

Plugging this into (A.1), we get

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

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

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

A similar proof, without the " $\supset \theta$" conditions, holds when noise is allowed so that $P(h \mid f, \theta)$ si 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

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

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

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

Note that in both (A.3) and (A.4) even when there is no noise we don't 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 $\{1, 2, ..., n\}$. $P(f, h, \theta) = 0$ 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, i.e., 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.[31,32] Let the cardinality of the training set be m, as usual. Use an error function independent of θ: $Er(f, h) = \Sigma_{\{x \in X\}} \{1 - \delta(f(x), h(x))\} / n \equiv \Sigma_{i=1}^{n} \{1 - \delta(f_i, h_i)\} / n$.

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

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

where by $\{\theta \subset f;\, m\}$ 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 III(i)), and by {f; Er(f, h) = E} is meant the set of all target functions which, which have error E with hypothesis function h.

We are assuming that P(f, h, θ) 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} | Er(f, h) = E, h, θ has m elements) becomes

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

where by P(f, h, _) is meant P(f, h, θ) for any m-element θ in agreement with f.

Since repeats are allowed and training sets are ordered, $\Sigma_{\{\theta \subset f;\ m\}}$ {1} = n^m for any f. Now examine a function f such that Er(f, h) = E. To calculate $\Sigma_{\{\theta \subset f;\ m\}}$ {δ(S(θ, h, θ_X), s))}, relabel the elements of X so that h disagrees with f on the elements 1, 2, ..., nE, and h agrees with f on the remaining elements nE + 1, ..., 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, ..., n}. This tells us that $\Sigma_{\{\theta \subset f;\ m\}}$ {δ(S((θ, h, θ_X), s))} $= C^m_s \times [n - (nE + 1) + 1]^s \times [nE]^{(m-s)} = C^m_s \times [n]^m \times [1 - E]^s \times [E]^{(m-s)}$.

These values for $\Sigma_{\{\theta \subset f;\ m\}}$ {1} and $\Sigma_{\{\theta \subset f;\ m\}}$ {δ(S((θ, h, θ_X), s))} are both independent of f. Therefore P({s of the elements of θ agree with h} | Er(f, h) = E, h, θ has m elements) = $C^m_s \times [1 - E]^s \times [E]^{(m-s)}$. This is exactly the value of P(E | 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 | 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 | B, C) = P(Er(f, h) = E | h, θ has m elements) is uniform (i.e., has the same value for all values E). However P(Er(f, h) = E | h, θ has m elements) equals

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

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

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

As before, $\Sigma_{\{\theta \subset f;\, m\}} \{1\} = n^m$, independent of f. Therefore we only need to calculate $\Sigma_{\{f\}}$ $\{\delta[S(f, h, X), n - nE]$ to see if $P(Er(f, h) = E \mid h, \theta$ 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 $C^n_{n\ E} \times (r - 1)^{(nE)}$. 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 which, together with the benchmark distribution $P(f, h, \theta)$, 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's 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 which implies $P(\theta \mid h)$ is independent of h - see footnote 32.) A perhaps more natural distribution to calculate is $P(E \mid m, s)$. However this distribution can be very nasty indeed. Consider the case where, 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 of the training set, for every training set. Although for this problem Er(f, h) 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 V 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 which are produced via a cross-validation-type (or similar) 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 an element of a set product of a question component and a training set component: $x'_i \equiv \{x'_i\}_q \times \{x'_i\}_{ts}$. $\{x'_i\}_q$ is a single base input-space value, and $\{x'_i\}_{ts}$ is a base training set. We require that P(d, g, ω) is zero unless 1) the number of $X \times Y$ pairs in $\{x'_i\}_{ts}$ is independent of i, 2) the unordered set of $X \times Y$ pairs given by $\theta(\omega) \equiv \{x'_i\}_{ts} \times (\{x'_i\}_q, y_i)$ is independent of i, 3) the cardinality of θ(ω) equals the cardinality of ω, m'. These three restrictions formally express the fact that we restrict our attention to those meta-training sets which 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, ω) is zero unless 4) θ(ω) meets that sampling assumption over X for base training sets discussed in the previous sections. (Though note that, formally speaking, $\theta(\omega)$ isn't a base training set since it isn't ordered.)

For an appropriate sampling assumption, requirement four 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. 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 of the 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 last 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; blank, 1, 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 we might wish to impose a particular sampling assumption over the allowed region of X' (and thereby over the allowed ω). Similarly, in some circumstances we might wish to disallow meta-training sets ω for which $\theta(\omega)$ has too low a cardinality (see footnote 24). 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 $P(E \mid h, \theta)$ does under the assumption of a uniform distribution over target functions. The proof is as follows:

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 $x \in [\theta(\omega)]_X$, $d_{ij}(\theta(\omega); x)$ doesn't 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're 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

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

$$= \Sigma_i\ [\ \delta(S(d_{i;\omega}, g(\theta(\omega)), X - [\theta(\omega)]_X), z) \times \Sigma_j\ P(d_{ij} \mid \omega)\ ]$$

$$= \Sigma_i\ [\ \delta(S(d_{i;\omega}, g(\theta(\omega)), X - [\theta(\omega)]_X), z) \times P(d_{i;\omega} \mid \omega)\ ],$$

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 of $\theta(\omega)$. This means that the quantity $\Sigma_i\ [\ \delta(S(d_{i;\omega}, g(\theta(\omega)), X - [\theta(\omega)]_X), z) \times P(d_{i;\omega} \mid \omega)\ ]$ is identical to the quantity $\Sigma_{\{f_1, \ldots, f_{n-m}\}}\{P(f_1, \ldots, f_{n-m} \mid \theta)\ \delta([\Sigma_{i=1}^{n-m}\ \delta(f_i, h(x_i))], z)\}$ 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 $P(f_1, ..., f_{n-m} \mid \theta)$ not varying with f, the case investigated in section II(ii)), then $P(E \mid g, \omega)$ is equal to $C_z^{(n-m)} \times (r - 1)^{(n-m-z)} / r^{(n-m)}$. QED.

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 PAC 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 which serve as its strength as a formalism turn out to make it ill-suited to 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), <u>independent of the particular complexity measure used</u>. 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 one 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 one to simply guess 0's for all questions not in the training set.[33] Or alternatively, guessing all 1's for questions outside the training set is a "good" strategy. In fact almost *any* strategy which 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.[34,35] Very few of the empirical studies of real-world machine learning meet the assumptions going into PAC. Moreover, very few such real-world studies are directly interested in the issues which 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 <u>any</u> 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 which might make the reader feel more comfortable with the use of P(f). First, 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). Second, 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 try to make "informationless" arguments uniquely fixing 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's 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's required is that 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" which don't imply that 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 perhaps (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 which translates such a target function f into a training set θ. This mechanism either may or may not be known, in part or in whole, to person B.

Person B can see θ. On the basis of 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, Er(f, h, θ).

Nobody is allowed to cheat.

REFERENCES

[1] Rumelhart, D, and McClelland, J. (1986). Explorations in the microstructure of cognition, volumes I and II. MIT Press, Cambridge, MA.

[2] Holland, J. (1975). Adaptation in natural and artificial systems. University of Michigan Press.

[3] Rissanen, J. (1986). Stochastic complexity and modeling. *The Annals of Statistics*, **14**, 1080-1100.

[4] Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, **11**, 416-431.

[5] Wolpert, D. (1990). The relationship between Occam's razor and convergent guess. *Complex Systems*, **4**, 319-368.

[6] Stanfill, C., and Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, **29**, 1213-1228.

[7] Poggio, T., and staff, MIT AI Lab (1988). MIT progress in understanding images. In L. Bauman

(Ed.), *Proceedings of the image understanding workshop*. McLean, VA.

[8] Morozov, V. (1984). Methods for solving incorrectly posed problems. Springer-Verlag.

[9] Wolpert, D. (1989). A benchmark for how well neural nets generalize. *Biological Cybernetics*, **61**, 303-313.

[10] Wolpert, D. (1990). Constructing a generalizer superior to NETtalk via a mathematical theory of generalization. *Neural Networks*, **3**, 445-452.

[11] Wolpert, D. (1990). A mathematical theory of generalization: part I. *Complex Systems*, **4**, 151-200.

[12] Farmer, J., and Sidorowich, J. (1988). Exploiting chaos to predict the future and reduce noise, Los Alamos report LA-UR-88-901

[13] Omohundro, S. (1987). Efficient algorithms with neural network behavior. Report UIUCSCS-R-87-1331 of the University of Illinois at Urbana-Champaign Computer Science Department.

[14] Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, **1**, 81-106.

[15] Dietterich, T. (1990). Machine learning. *Annual review of computer science*, **4**, 255-306.

[16] Duda, R., Hart, P. (1973). *Pattern classification and scene analysis*. Wiley, NY, NY.

[17] Tishby, N., Levin, E., and Solla, S. (1989). Consistent inference of probabilities in layered networks: predictions and generalization. *Proceedings of the international Joint Conference on Neural Networks*, Washington, D.C., pp. II 403-409, IEEE.

[18] Baum, E., and Haussler, D. (1989). What size neural net gives valid generalization? *Neural Computation*, **1**, 151-160.

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

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

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

[22] Sompolinsky, H., and Tishby, N. (1990). Learning from examples in large neural networks. *Physical Review Letters*, **65**, 1683-1686.

[23] Pearl, J. (1978). On the connection between the complexity and credibility of inferred models. *International Journal of General Systems*, **4**, 255-264.

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

[25] Akaike, H. (1973). *Information theory and an extension of the maximum likelihood principle*. In Petrov, B.A. and Csaki, F. (Ed.'s), International Symposium on information theory, Akademiai Kiado, Budapest, Hungary.

[26] Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *J. Roy. Statistic. Soc. B*, **39**, 44-47.

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

[28] Ker-Chau Li (1985). From Stein's unbiased risk estimates to the method of generalized cross-validation, *The Annals of Statistics*, **13**, 1352-1377.

[29] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions, *J. Roy. Statistic. Soc. B*, **36**, 111-120.

[30] Efron, B. (1979). Computers and the theory of statistics: thinking the unthinkable, *SIAM REVIEW*, **21**, 460-480.

[31] Gustafson, S., Little, G., and Simon, D. (1990). Neural network for interpolation and extrapolation. *Report number 1294-40, the University of Dayton, Research Institute, Dayton, Ohio*.

[32] Wolpert, D. (1990). Stacked generalization. *Neural Networks*, in press.

[33] Xiru Zhang. Private communication concerning stacked generalization and protein folding.

[34] Wolpert, D. (1990). A mathematical theory of generalization: part II. *Complex Systems*, **4**, 201-249. Cross-validation is a special case of the technique of "self-guessing" discussed here.

[35] Good, I. (1959). Kinds of probability. *Science*, **129**, 443-447.

[36] Cheeseman, P. (1985). In defense of probability. In Proceedings of the ninth international joint conference on artificial intelligence, 1002-1009.

[37] Smith, C., and Erickson, G. (1989). From rationality and consistency to Bayesian probability. In Skilling, J. (Ed.), *Maximum entropy and Bayesian methods*, Kluwer Academic Publishers.

[38] Wolpert, D. (1991). Improving the performance of generalizers by time-series-like pre-processing of the training set. Los Alamos Laboratory Report LA-UR-91-350.

[39] He Xiangdong and Zhu Zhaoxuan (1990). Nonlinear time series modeling by self-organizing methods. *Report from the Department of mechanics, Peking University, Beijing, PRC*. No report number.

[40] Schulz, G., et al. (1974). Comparison of predicted and experimentally determined secondary structure of adenyl kinase, *Nature*, **250**, 140-142.

[41] Wolpert. (1992). The limitations of the PAC framework for addressing real-world generalization, in preparation.

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

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

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