The Bayesian and Computational Learning Theories

David H. Wolpert
NASA Ames Research Center
MS 269-1, Moffett Field, CA 94035, USA
(dhw@ptolemy.arc.nasa.gov)

Keywords: Supervised Learning, Bayesian Learning, Computational Learning Theory, No-Free-Lunch Theorems

Article Definition: The foundations of two of the most popular mathematical approaches to the problem of supervised inductive learning are examined.

October 3, 2000

Abstract

In all forms of reasoning that do not proceed by strict logical induction, some kind of statistical algorithm must be employed. One of the major types of such reasoning is 'supervised learning', in which one is provided a training set of input-output pairs, and must guess what the entire input-output function is. Outside of conventional sampling theory statistics, there are two primary mathematical approaches to supervised learning: Bayesian Learning Theory and Computational Learning Theory. This article examines the foundations of those two mathematical approaches, especially in light of the 'no-free-lunch' theorems that limit what *a priori* formal assurances one can have concerning a learning algorithm without making assumptions concerning the real world.

1 How well can we learn: The mathematics of inductive learning

Inductive learning is the process of coming to statistical conclusions based on past experiences. As compared to deduction, with induction one is never perfectly sure of one's conclusion, instead arriving at a (hopefully highly probable) guess. Inductive learning is performed by the human brain continually; almost all of a brain's conclusions, from the "simplest" ones involved in sensor-motor decisions, to the most "sophisticated" existential ones concerning how one should live one's life, are based at least in part on inductive learning. Even science, arguably the acme of human thought, is ultimately inductive in nature, with the "past experiences" its conclusions are based on being previous experimental data, and with its "conclusions" being theories that are always open to revision.

A lot of work has been directed at implementing inductive learning algorithmically, in computers. "Adaptive computation", involving neural networks, fuzzy logic, and computational statistics, can be viewed as a set of attempts to do this. The topic of algorithmic induction also looms large in other fields like artificial intelligence and genetic algorithms. Recently this work has fostered renewed research on the mathematical underpinnings of inductive learning. A thorough understanding of that mathematics would not only result in improvements in our applied computational learning systems. It would also provide us insight into the entire scientific method, as well as human cognition. A more profound line of research cannot be imagined.

This chapter surveys "Bayesian learning theory" and "computational learning theory". These are the two primary mathematical approaches that have been applied to supervised learning, a particularly important branch of inductive learning. For reasons of space, the form of supervised learning considered in this chapter is extremely simplified, designed to highlight the distinctions between these two learning theories rather than present either approach in its full form.

The only mathematical framework that can encapsulate both learning theories is the over-arching "Extended Bayesian Framework" (Wolpert, 1997). For current (simplified) purposes, it can be synopsized as follows. Say we have a finite **input space** X and a finite **output space** Y, and a set of m input-output pairs, $d = \{d_X(i), d_Y(i)\}$. Refer to

d as a **training set**, and presume it was created by repeated noise-free sampling of an $X \rightarrow Y$ **target function**, f. More formally, assume that the **likelihood** governing the generation of d from f is $P(d \mid f) = \prod_{i=1}^{m} \pi(d_X(i))\delta(d_Y(i), f(d_X(i)))$, where $\delta(.,.)$ is the Kronecker delta function that equals 1 if its arguments are equal, 0 otherwise, and π is known as the **sampling distribution**. $P(f)$ is known as the **prior distribution** over targets, and $P(f \mid d)$ is known as the **posterior**.

Let h be the $X \rightarrow Y$ function our learning algorithm produces in response to d. So as far as learning accuracy is concerned, that learning algorithm is specified *in toto* by $P(h \mid d)$. Since our learning algorithm only sees d, not f (although it might make assumptions concerning f), $P(h \mid d, f) = P(h \mid d)$. Note that $P(h) = \sum_{d,f} P(h \mid d)P(d \mid f)P(f)$, and in general need not equal the prior $P(f)$ evaluated for $f = h$.

Take s to be the number of $d_X(i)$ such that $d_Y(i) = h(d_X(i))$, i.e., the learning algorithm's average accuracy on the training set. Finally, let C be the average (according to π) across all $x \in X$ lying outside of the training set of whether h and f agree on x. Call C the **off-training set error**. It is the measure of how well our learning algorithm generalizes from the training set. An alternative error function, indicted by c, isthe "IID" error function. It is the same average, just not restricted to $x \notin d_X$, so that a learning algorithm gets some credit simply for memorizing what it's already seen.

Extensions of these definitions to allow for other kinds of error functions, noise in the target, uncertain sampling distributions, different likelihoods, infinite input and output spaces, etc., are all straightforward though laborious. See (Wolpert, 1997). The next section presents some contextual theorems for our comparison of the Bayesian and computational learning "theories" of supervised learning. In the following two section those two theories are presented.

2 No free lunch: a formalization of inductive bias

We start with the following theorem (Wolpert, 1995), which says what our expected generalization error is after training on some particular training set:

Theorem 1 The conditional expectation value $E(C \mid d)$ can be written as a (non-Euclidean) inner product between the distributions $P(h \mid d)$ and $P(f \mid d)$: $E(C \mid d) = \sum_{h,f} Er(h, f, d)P(h \mid d)P(f \mid d)$.

(Similar results hold for $E(C \mid m)$, etc.)

Theorem 1 says that how well a learning algorithm $P(h \mid d)$ performs is determined by how "aligned" it is with the actual posterior, $P(f \mid d)$. This theorem allows one to ask questions like "for what set of posteriors is algorithm G_1 better than algorithm G_2?" It also means that, unless one can somehow prove (!), from first principles, that $P(f \mid d)$ has a certain form, one cannot prove that a particular $P(h \mid d)$ will be aligned with $P(f \mid d)$ and, therefore, one cannot prove anything concerning how well that learning algorithm generalizes.

There are a number of way to formalize this impossibility of establishing the superiority of some particular learning algorithm with a proof that is first-principles and assumption-free, and in particular is not implicitly predicated on a particular posterior. One of them is in the following set of **no-free-lunch** theorems (Wolpert, 1996):

Theorem 2: Let "$E_i(.)$" indicate an expectation value evaluated using learning algorithm i. Then for any two learning algorithms $P_1(h \mid d)$ and $P_2(h \mid d)$, independent of the sampling distribution,

i) Uniformly averaged over all f, $E_1(C \mid f, m) - E_2(C \mid f, m) = 0$;

ii) Uniformly averaged over all f, $E_1(C \mid f, d) - E_2(C \mid f, d) = 0$, for any training set d;

iii) Uniformly averaged over all $P(f)$, $E_1(C \mid m) - E_2(C \mid m) = 0$;

iv) Uniformly averaged over all $P(f)$, $E_1(C \mid d) - E_2(C \mid d) = 0$, for any training set d.

According to these results, by any of the measures $E(C \mid d)$, $E(C \mid m)$, $E(C \mid f, d)$, or $E(C \mid f, m)$, all algorithms are equivalent, on average. The uniform averaging that goes into these results should be viewed as a calculational tool for comparing algorithms, rather than as an assumption concerning the real world. In particular, the proper way to interpret 2.i is that, appropriately weighted, there are "just as many" targets for which algorithm 1 has better $E(C \mid f, m)$ as there are for which the reverse is true. Accordingly, unless one can establish *a priori*, before seeing any of the data d, that the f that generated d is one of the ones for which one's favorite algorithm performs better than other algorithms, one has *no* assurances that that algorithm performs any better than the algorithm of purely random guessing.

This does not mean that one's algorithm must perform the same as random guessing in the real world. Rather it means that, formally, one cannot establish superiority to random guessing without making some assumptions. Note in particular that you cannot use your prior experience — or even the billion years or so of "prior experiences" of your genome, reflected in the design of your brain — to circumvent this problem, since all that prior experience is, formally, just an extension to the training set d.

As an important example of the foregoing, consider assessing the validity of a hypothesis by using experimental data that was not available when the hypothesis was created. In the form of "falsifiability" this concept is one of the primary tools commonly employed in the scientific method. It can be viewed as a crude version of a procedure that is common in applied supervised learning: Choose between the two hypothesis functions h_A and h_B, made by running two generalizers A and B on a training set d_1, by examing their accuracies on a distinct "held-out" training set d_2 that was generated from the same target that generated d_1.

Such a procedure for choosing between hypotheses seems almost unimpeachable. Certainly its crude implementation in the scientific method has resulted in astonishing success. Yet it cannot be justified without making assumptions about the real world. To state this more formally, take any two learning algorithms A and B, and consider two new algorithms based on them, S and T. S uses an extension of the choosing procedure outlined above, know as **cross-validation**: given a training set d, S breaks d into two disjoint portions, d_1 and d_2; trains A and B on d_1 alone; sees which resultant hypothesis is more accurate on d_2; and then trains the associated learning algorithm on all of d and uses the associated hypothesis. In contrast, T uses *anti*-cross-validation: It is identical to S except that it chooses the learning algorithm whose associated hypothesis' accuracy on d_2 was *worst*. However by the no-free-lunch theorems, we know that T must outperform S as readily as vice-versa, regardless of A and B. It is only when a certain (subtle) relationship holds between $P(f)$ and the A and B one is considering that S can be preferable to T (cf. Theorem 1). When that relationship does not hold, T will outperform S.

This result means in particular that the scientific method must fail as readily as it succeeds, absent some *a priori* relation between the learning algorithms it uses (i.e., scientists) and the actual truth. Unfortunately though, next to nothing is known formally about that required relation. In this, the whole of science —not to mention human cognition— is based on a procedure whose assumptions not only are formally unjustified, but also have not even been formally stated.

3 Bayesian learning theory

Intuitively, the Bayesian approach to supervised learning can be viewed as an attempt to circumvent the no-free-lunch theorems by explicitly making an assumption for the posterior. Usually, to do this it first restricts attention to situations in which the likelihood is known (in the context of this chapter, that means it presumes that one knows there is no noise). It then makes an assumption the prior, $P(f)$. Next Bayes' theorem is invoked to combine the prior with the likelihood to give us our desired posterior: $P(f \mid d) \propto P(d \mid f)P(f)$, where the proportionality constant is independent of f. (Besides those concerning the prior, there are other kinds of assumptions which, when combined with the likelihood, fix the posterior (Wolpert, 1993). However such assumptions have not yet been investigated in any detail.)

Given such a posterior, one has uniquely specified the value of $E(C \mid d)$ that accompanies any particular learning algorithm $P(h \mid d)$ (cf. Theorem 1). In particular, one can solve for the $P(h \mid d)$ that minimizes $E(C \mid d)$, known as the **Bayes-optimal** learning algorithm. This algorithm is given by the following theorem, which is actually a bit more general than we need:

Theorem 3: Let $C(f, h, d) = \sum_{x \in X} \pi'(x) G[h(x), f(x)]$ for some real-valued function $G(.,.)$ and some real-valued $\pi'(.)$ that is nowhere-negative and either may or may not equal the distribution $\pi(.)$ arising in $P(d \mid f)$. Then the Bayes-optimal $P(h \mid d)$ always guesses the same function h^* for the same d:

$h^* = \{x \in X \longrightarrow \arg\min_{y \in Y} \Omega(x, y)\}$, where

$\Omega(x, y) \equiv \sum_f G(f(x), y) P(f \mid d)$.

(There are no restrictions on whether the function $\pi'(.)$ may vary with d, as it does in OTS error.)

Intuitively, Theorem 3 says that for any x, one should choose the $y \in Y$ that minimizes the average distance from y to $f(x)$, where the average is over all $f(.)$, according to the distribution $P(f \mid d)$, and "distance" is measured by $G(.,.)$. Note that this result holds regardless of the form of $P(f)$, and regardless of what (if any) noise process is present; all such considerations are taken care of automatically, in the $P(f \mid d)$ term. Note also that h^* might be an f with zero-valued posterior; in the Bayesian framework, h does not really constitute a "guess for the f which generated the data."

This is all there is to the Bayesian framework, as far as foundational issues are concerned (Berger, 1985), (Loredo, 1990), (Buntine and Weigend, 1991), (Wolpert, 1995). Everything else one reads concerning the framework involves either philosophical or calculational issues. The philosophical issues usually revolve around what $P(f)$ "means" (Wolpert, 1993). In particular, some hard-core Bayesians do not view the $P(f)$ they use to derive their learning algorithm as an assumption for the actual $P(f)$, one which may or not correspond to reality. Rather in general they interpret the probability of an event as one's "personal degree of belief" in that event, and therefore in particular interpret $P(f)$ that way. According to this view, probability theory is simply a calculus for forcing consistency in one's use of probability to manipulate one's subjective beliefs. Accordingly, no matter how absurd an Bayesian's prior, under this interpretation practitioners of non-Bayesian approaches to supervised learning are *by definition* always going to perform worse than that Bayesian (since the Bayesian "fixes" $P(f)$ and therefore $P(f \mid d)$ and guesses accordingly in an optimal manner — cf. Theorem 1).

Some of the calculational issues in the Bayesian framework involve calculating $P(f \mid d)$. However even when one knows $P(f \mid d)$, it is still often extremely difficult to evaluate the associated Bayes-optimal algorithm. Accordingly, people often settle for approximations to the Bayes-optimal algorithm, and/or incorporate into their algorithm estimates whose justification is intuitive rather than mathematical. See the discussion of empirical Bayes and ML-II in Section 9 of (Wolpert, 1995).

4 Computational learning theory

The Computational Learning Framework takes a number of forms, the primary ones being the statistical physics, PAC and VC (uniform convergence) approaches (Baum and Haussler, 1989), (Vapnik, 1982), (Wolpert, 1995). All three can be cast as bounds concerning a probability distribution that involves IID error, and that is conditioned on f (in contrast to the Bayesian framework, in which f is not fixed). In addition, in their most common forms they all have m rather than d fixed in their distribution of interest (again, in contrast to the Bayesian framework). This last point means that they do not address the question of what the likely outcome is for the training set at hand. Rather, they address the question of what the outcome likely would be if one had different training sets than the actual d. Such varying of quantities that are in fact fixed and known has been criticized by Bayesian practitioners on formal grounds, as violating any possible self-consistent principles for induction. (See (Wolpert, 1993) and the discussion of the "Honesty Principles" in (Wolpert, 1995) for an overview of the conflict between the two learning theories).

As a pedagogical example, this section focuses on (a pared-down version of) the VC framework. Start with the following simple result, which concerns the **confidence interval** relating c and s, for the case where $H\tilde{}$, the h-space support of a learning algorithm's $P(h)$, consists of a single h (Wolpert, 1995):

Theorem 4: Assume that there is an h' such that $P(h \mid d) = \delta(h - h')$ for all d. Then

$$P(c > s + \varepsilon \mid f, m) < 2e^{-m\varepsilon^2}.$$

(Recall that s is the empirical misclassification rate.) Note that this bound is independent of f, and therefore of the prior $P(f)$.

If $H\tilde{}$ instead consists of more than one h, the bound in Theorem 4 still applies if one multiplies the right-hand-side by $|H\tilde{}|$, the number of functions in $H\tilde{}$. The major insight behind the **uniform convergence** framework was how how to derive tighter bounds still by characterizing $P(h \mid d)$ in terms of its VC dimension (Baum and Haussler, 1989), (Vapnik, 1982), (Wolpert, 1995). (Care should be taken to distinguish between this use of the VC dimension and its use in other contexts, as a characterization of $P(f)$.) For $Y = \{0, 1\}$ and our error function, the VC dimension is given by the smallest m such that for any d_X of size m, all of whose elements are distinct, there is a d_Y for which no h in $H\tilde{}$ goes through d. (The VC dimension is this smallest number minus one.)

Common to all such extensions of Theorem 4 is a rough equivalence (as far as the likely values of c are concerned) between (i) lowering s; (ii) lowering the expressive power of $P(h \mid d)$ (i.e., shrinking its VC dimension, or shrinking $|H\tilde{}|$); and (iii) raising m. Important as these extensions of Theorem 4 are though, to understand the foundational issues underpinning the uniform convergence framework, it makes sense to restrict attention to the scenario in which there is a single h in $H\tilde{}$.

In general, Since you can measure s and want to know c (rather than the other way around), a bound on something like $P(c > k \mid s, m)$, perhaps with $k \equiv s + \varepsilon$, would provide us some useful information concerning generalization error. With such a bound, we could say that since we observe m and s to be such-and-such, with high probability c is lower than function(such-and-such). However since both f and (for our learning algorithm) h are fixed in the probability distribution in Theorem 4, c is also fixed there, for IID error. (This differs from the Bayesian framework, which has c's value is only probabilistically determined.) In fact, in Theorem 4 what is varying is d_X (or more generally, when there is noise, d). So Theorem 4 does *not* directly give us the probability that c lies in a certain region, given the training set at hand. Rather it gives the probability of a d_X (generated via experiments other than ours) such that the difference between the fixed c and (the function of d_X) s lie in a certain region.

It might seem that Theorem 4 can be modified to provide us a bound of the type we seek though. After all, Theorem 4 can be written as a bound on the "inverse" of $P(c > k \mid s, m)$, $P(s < \kappa \mid c, m)$, where $\kappa \equiv c - \varepsilon$. How does $P(s \mid c, m)$ relate to what we wish to know, $P(c \mid s, m)$? The answer is given by Bayes' theorem: $P(c \mid s, m) = P(s \mid c, m)\ P(c \mid m)\ /\ P(s \mid m)$.

Unfortunately, this result has the usual problem associated with Bayesian results; it is prior-dependent. Does it somehow turn out that that prior has little effect? Alas, no; depending on $P(c)$, $P(c > s + \varepsilon \mid s, m)$ can differ markedly from the bound on $P(s < c - \varepsilon \mid m, a, c)$ given in Theorem 4. Even if *given a truth* c, the probability of an s that differs substantially from the truth is small, it does not follow that *given an s*, the probability of a truth that differs substantially from that s is small.

To illustrate this point, say we have two random variables, A and B, which can both take on the values "low" and "high". Say that the joint probability distribution is $P(A = \text{high}, B = \text{high}) = 100$, $P(A = \text{high}, B = \text{low}) = 2$, $P(A = \text{low}, B = \text{low}) = 1$, and $P(A = \text{low}, B = \text{low}) = 1$. Then the probability that A and B differ is quite small (3/114); we have a tight confidence interval relating them, just as in Theorem 4. Nonetheless, $P(A = \text{high} \mid B = \text{low})$ is 2/3; despite the tight confidence interval, if we observe $B = \text{low}$, we cannot infer that A is as well. Replace "A" with "c", and "B" with "s", and we see that results like Theorem 4's do not imply that *having observed a low s*, one can conclude that one has a low c.

A more concrete example of this effect in the context of supervised learning is the following result, established in (Wolpert, 1995):

Theorem 5: Let $\pi(x)$ be flat over all x and $P(f)$ flat over all f. For IID error, the noise-free IID likelihood considered in this paper, and the learning algorithm of Theorem 4,

$$P(c \mid s, m) \;=\; [\tbinom{m}{sm}\, c^{sm}\, (1-c)^{m-sm}] \;\;\times\;\; [\tbinom{n}{nc}\, (r-1)^{nc}]\,.$$

Theorem 5 can be viewed as a sort of compromise between the likelihood-driven "something for nothing" results of the VC framework, and the no-free-lunch theorems. The first term in the product has no c-dependence. The second and third terms together reach a peak when $c = s$; they "push" the true misclassification rate towards the empirical misclassification rate, and would disappear if we were using off-training-set error. These two terms are closely related to the likelihood-driven VC bounds. However, the last two terms, taken together, form a function of c whose mean is $1/r$. They reflect the fact that all f's are being allowed with equal prior probability, and are closely related to the no-free-lunch theorems (despite the fact that iid error is being used). In this sense, our result for $P(c \mid s, m)$ is nothing other than a product of a no-free-lunch term with a VC-type term.

In response to such formal admonitions, one is tempted to make the following intuitive reply: "Consider the sequence where: a sample point is drawn from f; some pre-fixed hypothesis h' that has no *a priori* bias relative to f correctly predicts that point; another sample point is drawn; h' correctly predicts that point as well, etc. Based on a set of such points, you guess that h' will correctly predict the next sample point. And lo and behold, it does (s is small). In other words, the generalizer {always guess h'} has excellent cross-validation error. In this situation, wouldn't you believe that it is unlikely for h' and f to disagree on future sample points, regardless of the no-free-lunch theorems?

To disentangle the implicit assumptions behind this argument, consider it again in the the case where h' is some extremely complex function that was formed by a random process. Now the claim in the intuitive argument is that h' was fixed independent of any determination of f, d, or anything else, and is not biased in any way towards f. Then, so goes the claim, f was sampled to generate d, and it just so happened that f and h' agree on d. According to the intuitive argument presented above, we should conclude in such a case that h' and f would agree on points not yet sampled. Yet in such a situation our first suspicion might instead be that the claims that were made are wrong, that cheating has taken place and that h' is actually based on prior knowledge concerning f. After all, how else could the "essentially random" h' agree with f—h' was supposedly fixed without any information concerning d, and therefore without any coupling to f.

If, however, we are assured that no cheating is going on, then "intuition" might very well just shrug its shoulders and say that the agreements between f and h must be simple coincidence. *They have to be* since, by hypothesis, there is nothing that could possibly connect h and f. So intuition need not proclaim that the agreements on the data set mean that f and h' will agree on future samples. Moreover, if cheating did occur, then to formulate the problem correctly, then we have to know about the *a priori* connection between f and h in order to properly analyze the situation. This results in a different (prior-dependent) distribution that the one investigated in the uniform convergence framework.

5 References

Baum, E., and Haussler, D. (1989). What Size Net Gives Valid Generalization? *Neural Comp.* **1**

Berger, J. (1985). Statistical Decision Theory and Bayesian Analysis.

New York: Springer-Verlag, 1985.

Loredo, T. (1990) From Laplace to Supernova 1987a: Bayesian Inference in Astrophysics. In P. Fougere (Ed.), *Maximum Entropy and Bayesian Methods.* Kluwer.

Buntine, W., and Weigend, A. (1991). Bayesian Back-Propagation. *Complex Systems* **5**.

Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data.* Springer-Verlag.

Wolpert, D.H. (1993). Reconciling Bayesian and non-Bayesian analysis. In G. Heidbreder (Ed.), *Maximum Entropy and Bayesian Methods 1993.* Kluwer.

Wolpert, D.H. (1995). The Relationship between PAC, the Statistical Physics Framework, the Bayesian Framework, and the VC Framework. In D. H. Wolpert (Ed.), *The mathematics of generalization* (pp. 117–214). Addison-Wesley.

Wolpert, D.H. (1996). The Lack of a Priori Distinctions between Learning Algorithms. *Neural Computation*, **7**.

Wolpert, D.H. (1997). On Bias plus Variance. *Neural Computation*, **9**.

6 Glossary

Algorithm: An unambiguous step-by-step procedure, especially one that is carried out on a computer.

Bayes' Theorem: The definition that for any two events A and B, the conditional probability $P(A \mid B) \equiv P(A,B)/P(B)$, and therefore $P(B \mid A) = \frac{P(A|B)P(B)}{\sum_A P(A|B)P(B)}$.

Deductive Reasoning: The process of coming to a unique conclusion about the state of the world by logically deriving that conclusion from provided information. Compare to inductive reasoning.

Expectation Value: The expectation value of a numerical random variable A is its average according to the associated probability distribution, $\sum_A AP(A)$.

IID: Independent, Identically Distributed. The process of generating data by repeatedly sampling the same underlying probability distribution, where each sample is generated independently of the others.

Inductive Reasoning: An algorithm for making a guess as to the state of the world based on information that is insufficient to logically fix that state. Compare to deductive reasoning.

Learning Algorithm: An algorithm for making a probabilistic guess for some quantity based on data concerning that quantity, especially such an algorithm for performing supervised learning. Examples include neural networks, fuzzy logic, and statistical regression.

Scientific Method: The process by which the scientific community comes to agreement concerning the legitimacy of a particular theory, especially how data related to that theory is used to this end.

Supervised Learning: The general problem of how to infer an entire input-output function given only a finite data set formed by sampling that function.

Training Set: A set of input-output pairs formed by sampling an input-output function.