– Neural Computation, Vol 12, No. 3, 1999 –

No Free Lunch For Noise Prediction

Malik Magdon-Ismail
Caltech 136-93
Pasadena, CA 91125
magdon@cco.caltech.edu

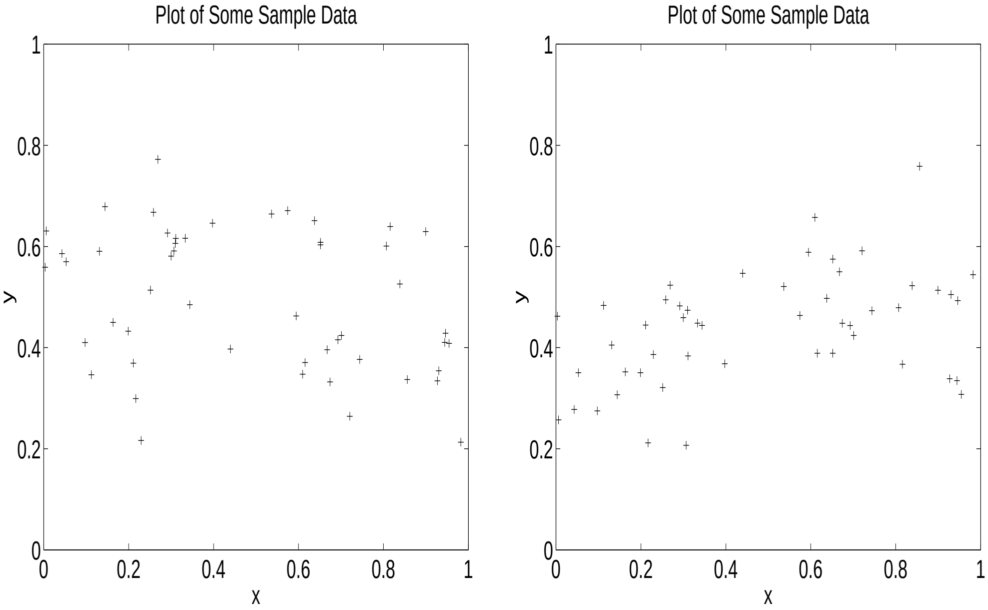
November 20, 1999

Abstract

No Free Lunch theorems have shown that learning algorithms cannot be universally good. We show that No Free Lunch exists for noise prediction as well. We show that when the noise is additive and the prior over target functions is "uniform", a prior on the noise distribution cannot be updated, in the Bayesian sense, from any finite data set. We emphasize the importance of a prior over the target function in order to justify superior performance for learning systems.

Keywords: No Free Lunch, Noise Prediction, Bayesian Prior.

Which data set has more noise?



(a) (b)

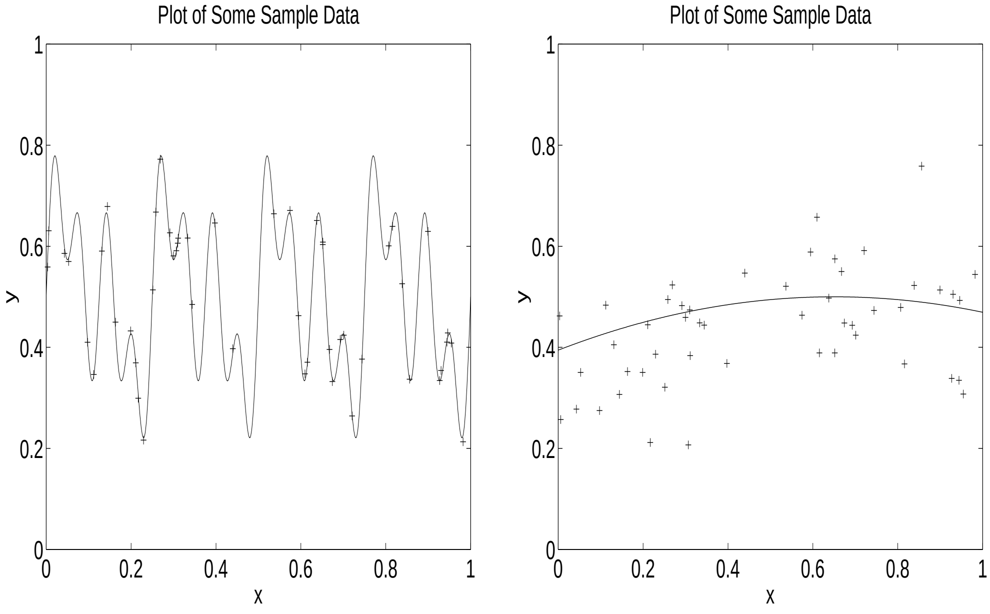
Two sample noisy data sets created by taking two functions and adding noise to them. The x*-values are the same in both cases.*

Figure 1: Two sample noisy data sets.

1 Introduction

Already established in the learning theory literature are a set of “No Free Lunch” (NFL) theorems for various types of learning systems, (Wolpert, 1996b), (Wolpert, 1996a), (Zhu, 1996), (Cataltepe, Abu-Mostafa, & Magdon-Ismail, 1998) and for optimization methods (Wolpert & Macready, 1997). The essential content of these theorems is that no learning algorithm can be universally good. A learning algorithm that performs exceptionally well in certain situations will perform comparably poorly in other situations. Thus, in particular, random algorithms that always yield average performance, cross validation, bootstrapping and anti-cross validation are all, in some sense, on an equal footing. Stopping early at a higher training error than the minimum achievable gains nothing if the hypotheses yielding that training error are chosen with equal probability. A search technique that performs well on one cost function will perform poorly on another cost function. These existing results demonstrate that for certain problems, given a certain criterion of performance, it is not possible to consistently do well in the most general setting. Thus, one needs to make some assumptions about the underlying structure of the problem to be able to claim any kind of superior performance.

Here we look at the problem of inferring the noise distribution from a data set. The existing NFL results look at problem of learning a dependency (not necessarily deterministic) from a finite data set and making statements about the out of sample performance. The problem we treat here is to deduce properties of the noise distribution (for example, the noise variance) from a finite data set that consists of input–output pairs. The outputs have a deterministic dependence on the inputs (the target function) and a noisy component. One approaches the problem with some prior belief about the noise distribution, and one hopes that the data will enable a fine tuning of that prior belief to the extent that more precise statements can be made about the properties of the noise. An accurate estimate of the noise distribution is useful because the noise sets a fundamental limit on the performance of a learning system (Cortes, Jackel, & Chiang, 1994). Noise may play other roles as well: for example, in financial markets, the noise level is itself a tradable quantity.



(a) (b)

The same noisy data sets that were shown in figure 1, and the functions that were used to produce them. In (b), the noise variance is roughly six times larger.

Figure 2: Noisy data and the functions that generated them.

Our criterion for selection of a noise distribution will be to maximize its posterior probability. We will show that having a uniform prior on the possible realizations of the target function leads to no update for a well behaved prior on the noise distributions (in the Bayesian sense) from any finite data set. Thus the posterior over noise distributions will equal the prior and we may as well have not even looked at the data. One can compare this to the analogous NFL result of Wolpert: if the prior over target functions is uniform then the out of sample error is constant and thus one need not have even looked at the data before picking a hypothesis function. Let's consider the question posed in figure 1. Which data set has more noise? Figure 2 shows the same data, but this time with the function that created the data. Now, which data set has more noise? The task is considerably easier given this extra information. We will show that some extra information of this form is essential.

This paper is organized as follows. We start with an example in section 2 where we attempt to predict a noise variance when learning using linear models. In this section we show that one method for predicting the noise variance using a bootstrap technique does not work in the naive form presented. As a result, one might therefore spend a lot of effort trying to find a more sophisticated technique that might work. That this would be fruitless (for this specific case and the more general case) is the purpose of the remainder of the paper. In section 3, we set up the problem of noise prediction in general and section 4 presents the main results, beginning with the simple case of boolean functions, and continuing to the more general case. We will show that information about the noise distribution cannot be obtained from a finite data set if the target distribution is assumed uniform. We conclude with section 5 where we also discuss the implications of the NFL theorems.

2 An Example Using Linear models

In this section we illustrate the conclusions of this paper for the case of linear learning models on the input space $\mathbf{R}^d$. It is not our goal to be strictly rigorous, but rather to illustrate the point. Let the hypothesis functions be of the form

$$g_{\mathbf{w}}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + \mathbf{w}_0 \tag{1}$$

where $\mathbf{w}, \mathbf{w}_0$ are called the weights. We are given a data set $\mathcal{D} = \{\mathbf{x}_\alpha, y_\alpha = f(\mathbf{x}_\alpha) + n_\alpha\}_{\alpha=1}^{N}$. $f(\cdot)$ is some deterministic function (we place no other restrictions on the function) and n_α is noise, drawn *i.i.d.* with zero mean and variance σ^2. We use mean squared deviation as the measure of error. Define the training error as

$$\mathcal{E}_{tr} = \frac{1}{N}\sum_{\alpha}(\mathbf{w}\cdot\mathbf{x}_\alpha + w_0 - y_\alpha)^2 \tag{2}$$

Define $\mathbf{w}^*, \mathbf{w}_0^*$ as those weights that minimize $\mathcal{E}_{tr}$. It can then be shown that at this minimum, the expected training error is given by (theorem 6.1 in the appendix)[1]

$$\langle {\mathcal{E}_{tr}}^* \rangle_{\mathcal{D},\mathbf{n}} = E_0 + \sigma^2 - \frac{\sigma^2(d+1)+B}{N} + O\left(\frac{1}{N^{\frac{3}{2}}}\right) \tag{3}$$

we use $\langle\cdot\rangle$ to denote expectations and the expectation here is over the possible data sets (the input space and the noise, where we have made explicit the expectation over the noise)[2]. E_0 is the *bias*, where the *bias* is the expected

[1] We use standard order notation

$$f(N) = o(g(N)) \Rightarrow f(N)/g(N) \stackrel{N\to\infty}{\longrightarrow} 0$$

$$f(N) = O(g(N)) \Rightarrow f(N) \leq Cg(N) \text{ for some } C > 0$$

[2] In the language of the NFL theorems (Wolpert, 1996b), $\langle {\mathcal{E}_{tr}}^* \rangle_{\mathcal{D},\mathbf{n}}$ is $E(C|f,m)$ where the on-training-set likelihood and the quadratic loss func-

training error of the best hypothesis function in our learning model. B is a constant given by theorem 6.1 in the appendix. For the purposes of this section, the exact expression for B is not essential. It suffices to know that B is related to the variance of a sample statistic around its population value.

The goal here is to estimate the noise variance, σ^2. The training error used as an estimate of the noise variance is asymptotically biased. For large N, it over predicts the noise variance by the *bias*, E_0. Since we know the form of $\langle {\mathcal{E}_{tr}}^* \rangle$, perhaps we can do better. We can sample N_1 data points from the N data points and estimate the training error. We use bootstrapping (Shao & Tu, 1996) to estimate $\langle {\mathcal{E}_{tr}}^*(N_1) \rangle$, and by varying N_1, we can fit the resulting dependence of $\langle {\mathcal{E}_{tr}}^*(N_1) \rangle$ on $1/N_1$ using (3). Let the slope of this fit be $\hat{A}$. From (3), we see that $\hat{A}$ is an estimate for $\sigma^2(d+1)+B$, therefore, $\hat{A}/(d+1)$ used as an estimate of σ^2 is off by $B/(d+1)$. Perhaps we can estimate B by the bootstrap as well? Given a bootstrapped estimate for B, we can combine it with $\hat{A}$ to obtain an estimate for σ^2. A more careful analysis presented in the appendix shows that this too leads to failure.

Thus, to get information on σ^2, perhaps a more subtle approach such as looking at terms with higher order in $1/N$ would work. Perhaps some method other than simply using the training error would yield a better result. Instead of using linear models, why not use a more complex model for which E_0 is zero? Then at least asymptotically, the training error will approach σ^2 (provided that the model is not too complex). However, for any finite N, we run the risk of over fitting the data, perhaps even obtaining an estimate $\sigma^2 = 0$. Perhaps there is a systematic way of choosing the model complexity as a function of N to get an optimal noise variance estimate.

tion are used

The purpose of the next few sections is to show that such an exhaustive search will necessarily be fruitless unless we make some statement about f – in this case it would suffice to tell us "how good our model is at estimating f" (i.e., E_0). Rather than explore every other potential method for estimating σ^2 in this linear scenario, we will set up a more general framework for the estimation of noise distributions and show that unless some assumptions are made about the target function, all noise models are just as probable, given the data. The intuition is that the data points are consistent with *any* noise level if we do not restrict the target function. We pursue these issues next.

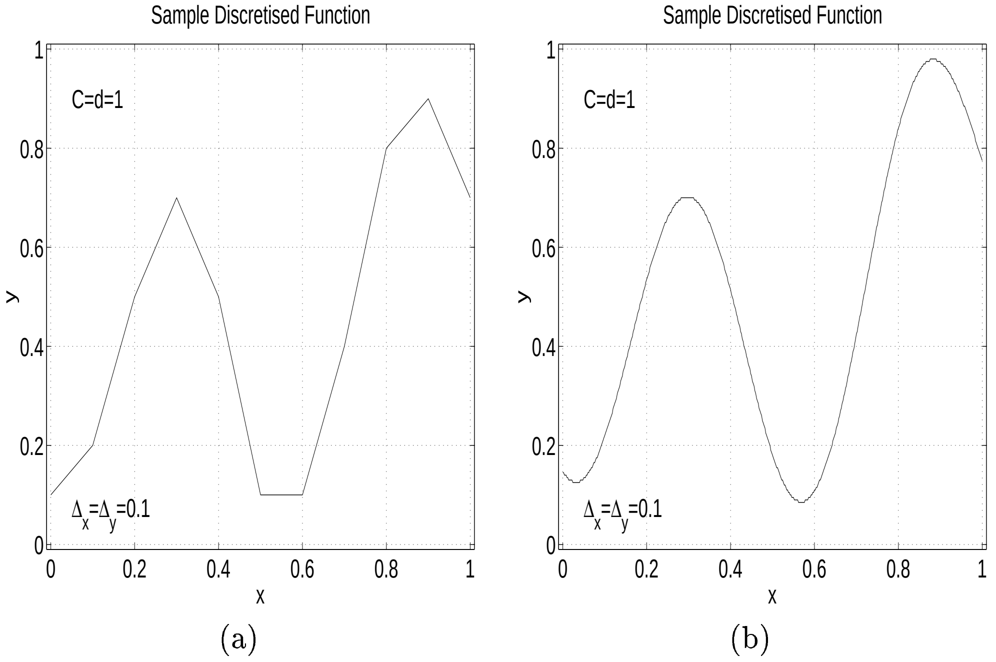
3 Problem Set Up

A finite data set $\mathcal{D} = \{\mathbf{x}_\alpha, y_\alpha\}_{\alpha=1}^{l}$ is given, where $\mathbf{x}_\alpha \in \mathbf{R}^d$, $y_\alpha = f(x_\alpha) + n_\alpha$, $f : \mathbf{R}^d \to \mathbf{R}$ and the data set consists of l data points[3]. We will concentrate on the case where the noise distribution is fixed. The results can be modified to accommodate noise distributions that vary with $\mathbf{x}$. It will always be understood that α indexes points in the data set.

We will concentrate on compact spaces, and assume that the input space is the unit hypercube in d dimensions. Let the output space be the bounded interval $[-C, C]$. Discretize the input and output spaces as follows. Let the input variable, $\mathbf{x}$, take on values in the set

$$\mathcal{X} = \{0, \Delta_x, 2\Delta_x, \ldots, 1 - \Delta_x, 1\}^d \tag{4}$$

[3] We use l rather than the customary N to avoid confusions with N_x, N_y to be defined shortly



The discretized function space has only a finite number of functions. An arbitrary precision can be obtained by making the grid finer and finer.

Figure 3: The discretized function space.

so there are $N_{\mathbf{x}} = 1 + \frac{1}{\Delta_x}$ possible values for each input variable. Let the output variable, y, take a value in the set

$$\mathcal{Y} = \{-C, -C + \Delta_y, \ldots, 0, \Delta_y, 2\Delta_y, \ldots, C - \Delta_y, C\} \qquad (5)$$

There are $N_y = \frac{2C}{\Delta_y} + 1$ possible y values. A function (f) will be a mapping, $f : \mathcal{X} \rightarrow \mathcal{Y}$. There are $\Gamma = N_y^{N_{\mathbf{x}}^d}$ different functions. By making Δ_x, Δ_y as small as we choose, we can achieve an arbitrary precision (for example see figure 3). In all applications, only a finite number of functions are available (computers are finite precision machines). Due to this finiteness, we can define a probability distribution on the function space – a prior over the function space. One natural prior is to take this distribution to be uniform. Therefore to every function we assign a probability of $\frac{1}{\Gamma}$.

We assume that C can be made as large as we wish. Further, without loss of generality, we can scale the outputs by $\frac{1}{\Delta_y}$, therefore we can assume that $y \in \{-\tilde{C}, -\tilde{C}+1, -\tilde{C}+2, \ldots, \tilde{C}-1, \tilde{C}\}$, $N_y = 2\tilde{C}+1$ and $\tilde{C} = C/\Delta_y$. With this representation, the noise $n \in \{0, \pm 1, \pm 2, \ldots\}$ has a distribution given by a vector $\mathbf{P}$ where $P_i = Pr[n = i]$ and $\sum_i P_i = 1$. Assume a prior probability measure on the possible noise distributions $g(\mathbf{P})$. We are interested in the Bayesian posterior on these noise distributions

$$g(\mathbf{P}|\mathcal{D}) = \frac{Pr[\mathcal{D}|\mathbf{P}]g(\mathbf{P})}{Pr[\mathcal{D}]} \tag{6}$$

where $Pr[\mathcal{D}] = \sum_{\mathbf{P}} Pr[\mathcal{D}|\mathbf{P}]g(\mathbf{P})$. We can calculate $Pr[\mathcal{D}|\mathbf{P}]$ as follows.

$$Pr[\mathcal{D}|\mathbf{P}, f] = \prod_{\alpha=1}^{l} Pr[n_\alpha = y_\alpha - f(\mathbf{x}_\alpha)] \tag{7}$$

Note that the RHS implicitly depends on $\mathbf{P}$ through the probabilities for the noise realizations n_α. Multiplying by $P[f]$ and summing over f, we have

$$\begin{aligned} Pr[\mathcal{D}|\mathbf{P}] &= \frac{1}{\Gamma} \sum_{f(\mathbf{x}_1)} \sum_{f(\mathbf{x}_2)} \cdots \sum_{f(\mathbf{x}_{N_\mathbf{x}})} \prod_{\alpha=1}^{l} Pr[n_\alpha|\mathbf{f}, \mathcal{D}] \\ &\overset{(a)}{=} \frac{1}{N_y^l} \prod_{\alpha=1}^{l} \sum_{f(x_\alpha)=-C}^{C} Pr[n_\alpha|\mathbf{f}, \mathcal{D}] \end{aligned} \tag{8}$$

where we have used the notation $Pr[n_\alpha|\mathbf{f}, \mathcal{D}]$ to mean $Pr[n_\alpha = y_\alpha - f(\mathbf{x}_\alpha)]$.

(a) follows when we sum over all points not in the data set, and we have used the fact that $Pr[f] = 1/\Gamma$ - uniform prior.

4 No Free Lunch for Noise Prediction

4.1 Boolean Functions

Notation:

$$\begin{aligned} \mathbf{x} &\in \{0,1\}^d \\ f : \mathbf{x} &\to \{0,1\} \\ (1-p) \in [0,1] &= \textit{probability of flip} \end{aligned}$$

There are $\Gamma = 2^{2^d}$ possible functions.

Theorem 4.1 *Let $g(p)$ be the prior distribution for the noise parameter, p. If the prior distribution on the possible functions is uniform then $g(p|\mathcal{D}) = g(p)$.*

Proof.

Let m_i be the number of times f_i agrees with $\mathcal{D}$ where we use i to index functions. Then $Pr[\mathcal{D}|p, f_i] = p^{m_i}(1-p)^{l-m_i}$, so

$$Pr[\mathcal{D}, f_i|p] = Pr[\mathcal{D}|p, f_i]Pr[f_i|p] = \frac{1}{\Gamma}p^{m_i}(1-p)^{l-m_i} \tag{9}$$

Let $\rho(m_i)$ be the number of functions agreeing exactly m_i times with the data set $\mathcal{D}$. $\rho(m_i) = \binom{l}{m_i} 2^{2^d-l} = \binom{l}{m_i} 2^{-l}\Gamma$. Summing (9) over f_i, we get

$$Pr[\mathcal{D}|p] = \sum_{f_i} Pr[\mathcal{D}, f_i|p]$$

$$
\begin{aligned}
&= \frac{1}{\Gamma}\sum_{m_i=0}^{l} p^{m_i}(1-p)^{l-m_i}\rho(m_i) \\
&= \frac{1}{2^l}\sum_{m_i=0}^{l} p^{m_i}(1-p)^{l-m_i}\binom{l}{m_i} \\
&= 2^{-l}
\end{aligned}
$$

independent of p. Therefore,

$$
g(p|\mathcal{D}) = \frac{Pr(\mathcal{D}|p)g(p)}{Pr(\mathcal{D})} = g(p)
$$

■

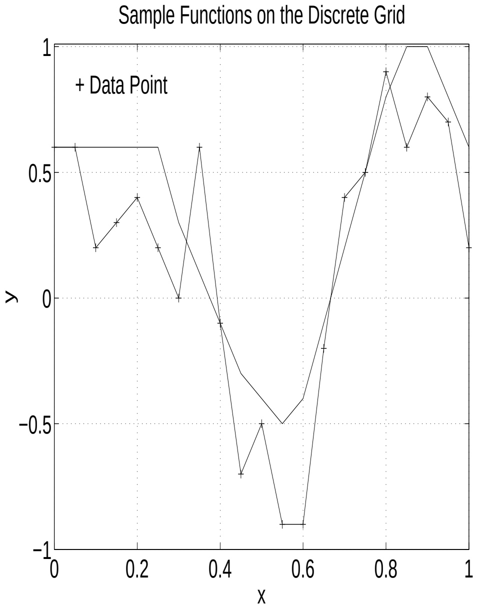
4.2 Extension to More General Functions

We would like to extend the previous analysis to the case where the output is not binary and the input is the discretized version of $[0,1]^d$. Figure 4 illustrates the conclusions of this section on a particular discretized grid. This section is somewhat technical. Its essential content is contained in theorem 4.8 which formalizes the intuition that if every target function value is equally likely, then, each noise realization for every data point is equally likely. Hence nothing can be deduced about the relative likelihoods of different noise realizations.

4.2.1 Cyclic Noise

When the noise is additive modulo N_y then $\mathbf{P} = [P_0, P_1, \ldots, P_{N_y-1}]$ fully specifies the noise distribution.

example: The binary case with probability of flip $1-p$ we have $N_y = 2$, $P_0 = p,\ P_1 = 1-p$



Suppose that the two functions shown are both equally likely (probability 1/2) to be the function that created the data. Then, the two hypotheses noise=0 and noise>0, are still equally likely to be true if they were apriori *equally likely to be true.*

Figure 4: Illustration of NNFL.

Lemma 4.2 $\sum_{f(x_\alpha)} Pr[n_\alpha|\mathbf{f},\mathcal{D}] = 1$ *for all* $\mathbf{x}_\alpha \in \mathcal{D}$.

PROOF:

$$\sum_{f(x_\alpha)=-C}^{C} Pr[n_\alpha|\mathbf{f},\mathcal{D}] = \sum_{n_\alpha=y_\alpha+C}^{y_\alpha-C} Pr[n_\alpha|\mathbf{f},\mathcal{D}] = \sum_{n_\alpha=0}^{N_y-1} P_{n_\alpha} = 1$$

■

Combining Lemma 4.2 with (8) we have the following theorem.

Theorem 4.3 *Let $g(\mathbf{P})$ be the prior distribution for the cyclic noise. If the prior distribution for the functions is uniform then $g(\mathbf{P}|\mathcal{D}) = g(\mathbf{P})$*

PROOF:

By Lemma 4.2 and (8), $Pr[\mathcal{D}|\mathbf{P}] = \frac{1}{N_y^l}$. Therefore,

$$g(\mathbf{P}|\mathcal{D}) = \frac{Pr[\mathcal{D}|\mathbf{P}]g(\mathbf{P})}{Pr[\mathcal{D}]} = g(\mathbf{P})$$

$\blacksquare$

Note that the boolean case is a special case of the cyclic noise case. In the language of information theory (Cover & Thomas, 1991), the noise forms a symmetric channel between the function value and the output value. A uniform distribution for the function values induces a uniform distribution on the output values, independent of the details of the symmetric noisy channel. Hence, observing the output conveys no information about the noisy channel. This is essentially the content of theorem 4.3.

4.2.2 Thresholded Additive Noise

Noise is added to the target and then the resultant value is thresholded so,

$$y_i = \begin{cases} \min\{C, f(\mathbf{x}_i) + n_i\}, & f(\mathbf{x}_i) + n_i \geq 0 \\ \max\{-C, f(\mathbf{x}_i) + n_i\}, & f(\mathbf{x}_i) + n_i < 0 \end{cases}$$

$\mathbf{P} = [P_0, P_{\pm 1}, P_{\pm 2}, \ldots]$. Unlike in the previous case, edge effects make $Pr[\mathcal{D}|\mathbf{P}]$ dependent on $\mathbf{P}$. These effects can be made as small as we please by allowing C to be as large as we please. The intuition is that the data set is finite (y_i is bounded) and so, because the noise distribution must decay to zero, if C is large enough, the probability that $f(x_\alpha)$ is close to the edge and y_α so far away becomes negligible. This is formalized in the next lemma.

Lemma 4.4 *Given $\epsilon > 0, \exists C^*$ such that for all $C \geq C^*$,*

$$\frac{1}{N_y^l}(1-\epsilon) \leq Pr[\mathcal{D}|\mathbf{P}] \leq \frac{1}{N_y^l} \tag{10}$$

($N_y = 2C + 1$ and intuitively speaking, $\epsilon \overset{C\to\infty}{\longrightarrow} 0$).

PROOF:

Because $\{P_i\}$ is summable, there exists η such that $\sum_{|i|>\eta} P_i < \frac{\epsilon}{l}$. Let $y_{max} = \max\{y_i\}$ and $y_{min} = \min\{y_i\}$. Choose $C^* > \max\{|y_{max} + \eta|, |y_{min} - \eta|\}$ and let $C \geq C^*$. We then have,

$$\begin{aligned} \sum_{f(x_\alpha)=-C}^{C} Pr[n_\alpha|\mathbf{f}, \mathcal{D}] &= \sum_{n_\alpha = y_\alpha - C}^{y_\alpha + C} Pr[n_\alpha|\mathbf{f}, \mathcal{D}] \\ &= 1 - \sum_{n_\alpha < y_\alpha - C} Pr[n_\alpha|\mathbf{f}, \mathcal{D}] - \sum_{n_\alpha > y_\alpha + C} Pr[n_\alpha|\mathbf{f}, \mathcal{D}] \\ &\geq 1 - \frac{\epsilon}{l} \end{aligned}$$

Where the last inequality follows because by construction $y_{max} - C < -\eta$ and $y_{min} + C > \eta$. Using (8) we have

$$\frac{1}{N_y^l} \geq Pr[\mathcal{D}|\mathbf{P}] \geq \frac{1}{N_y^l}(1 - \frac{\epsilon}{l})^l$$

The lemma now follows by using the inequality $(1-x)^n \geq 1 - nx$ for $0 \leq x \leq 1$. ■

Thus we see that by choosing C, the bound on our function f to be large enough, the noise distribution has almost no effect on the probability of obtaining a particular data set. All data sets become equally probable, in the limit of large C. This in turn tells us that the data should not be telling us any more about the noise distribution than we already know. The prior distribution for the noise is unaffected given *any finite* data set. To formalize this intuition, we need to impose a technical condition on the prior over the noise distribution space.

Definition 4.5 *Let* $0 < \epsilon \leq 1$. *Define* $\eta_{\mathbf{P}}(\epsilon)$ *by*

$$\eta_{\mathbf{P}}(\epsilon) = \min \left\{ \eta : \sum_{|i|>\eta} P_i < \epsilon \right\}$$

For all $0 < \epsilon \leq 1$, $\eta_{\mathbf{P}}(\epsilon) < \infty$, by the summability of $\mathbf{P}$.

Definition 4.6 *Let* $0 < \epsilon \leq 1$. *Define* $\bar{\eta}(\epsilon)$ *by*

$$\bar{\eta}(\epsilon) = \sup_{\mathbf{P}} \{\eta_{\mathbf{P}}(\epsilon)\}$$

where the sup *is taken over* $\mathbf{P} \in$ *support*$\{g(\mathbf{P})\}$.

Definition 4.7 *We say that a prior on the space in which* $\mathbf{P}$ *is defined,* $g(\mathbf{P})$, *is totally bounded if*

$$\bar{\eta}(\epsilon) < \infty$$

for all $\epsilon > 0$.

Essentially this means that the support set for the prior on the noise distributions is not "too large." For example, a prior over the noise distribution space that has support only over a finite number of points (i.e., that is non zero only for a finite number of noise distributions) is totally bounded. We are now in a position to make precise the intuition mentioned above.

Theorem 4.8 (No Noisy Free Lunch (NNFL)) *Let the prior over noise distributions,* $g(\mathbf{P})$, *be totally bounded. For all* $0 < \epsilon < 1$, $\exists C^* > 0$ *such that* *for all* $C \geq C^*$*,*

$$(1-\epsilon)g(\mathbf{P}) \leq g(\mathbf{P}|\mathcal{D}) \leq \frac{g(\mathbf{P})}{1-\epsilon} \tag{11}$$

Therefore,

$$\lim_{C \to \infty} g(\mathbf{P}|\mathcal{D}) = g(\mathbf{P})$$

because $\epsilon \to 0$ *can be attained by letting* $C \to \infty$*.*

Proof:

Following lemma 4.4, now let $C^* > \max\{|y_{max} + \bar{\eta}(\epsilon)|, |y_{min} - \bar{\eta}(\epsilon)|\}$ and let $C \geq C^*$. Then Lemma 4.2 applies uniformly to every $\mathbf{P} \in \mathit{support}\{g(\mathbf{P})\}$. Therefore (10) holds for every $\mathbf{P} \in \mathit{support}\{g(\mathbf{P})\}$.

$$\begin{aligned} g(\mathbf{P}|\mathcal{D}) &= \frac{Pr[\mathcal{D}|\mathbf{P}]g(\mathbf{P})}{Pr[\mathcal{D}]} \\ &= \frac{Pr[\mathcal{D}|\mathbf{P}]g(\mathbf{P})}{\sum\limits_{\mathbf{P}} Pr[\mathcal{D}|\mathbf{P}]g(\mathbf{P})} \end{aligned} \tag{12}$$

Using (10), an upper bound can be obtained by substituting $Pr[\mathcal{D}|\mathbf{P}]$ with $1/N_y^l$ in the numerator, and $(1-\epsilon)/N_y^l$ in the denominator. To get a lower bound, we do the opposite. Then, using the fact that $\sum_{\mathbf{P}} g(\mathbf{P}) = 1$, we get (11). ■

Therefore, having no bound on f and allowing all f's to be equally likely does not allow anything new to be deduced about the noise distribution given any finite data set. It should also be clear how to extend these results to the case where the noise distribution has $\mathbf{x}$ dependence – one looks at each point independently.

The non-thresholded additive noise model is asymptotically obtained by letting C become arbitrarily large. Further the results apply to arbitrary $\Delta_{\mathbf{x}}$, Δ_y, so by allowing $\Delta_{\mathbf{x}}$, $\Delta_y \to 0$, these results can be applied to the non-discretized model in this asymptotic sense.

5 Discussion

The posterior $g(\mathbf{P}|\mathcal{D})$ is given by

$$g(\mathbf{P}|\mathcal{D}) \propto g(\mathbf{P}) \sum_f Pr[\mathcal{D}|\mathbf{P}, f]P[f] \tag{13}$$

where $P[f]$ is the prior over target functions. We have demonstrated that when $P[f]$ is "uniform", it is not possible to update a prior on the noise distribution from a finite data set, provided that a certain technical condition is satisfied. A non trivial $P[f]$ is needed for $g(\mathbf{P}|\mathcal{D})$ to be different from $g(\mathbf{P})$. Exactly how $P[f]$ will factor into $g(\mathbf{P}|\mathcal{D})$ is given by (13).

The result is more useful than it appears at first sight, for one might argue that no one ever has a uniform prior. A restatement of the result is: Given two noise distributions, $\mathbf{P}_1$ and $\mathbf{P}_2$, there are just as many (non-uniform) priors that favor $\mathbf{P}_1$ over $\mathbf{P}_2$ (when one weights by the amount by which $\mathbf{P}_1$ is favored), as vice versa.

This result fits into a set of NFL theorems that might be considered a collection of negative results, offering no hope for learning theory. The situation is considerably more positive, however. One is never in a situation with a uniform $P[f]$. Further, it is rare that no information about $P[f]$ is known (excepting that it is non-uniform). Therefore, incorporating $P[f]$ into the noise prediction or learning algorithm is a must in order to claim superior performance. Distribution independent bounds such as the VC bounds (Vapnik, 1995) say that with high probability, the training error is close to the test error given some fixed target function f. But, VC results do not guarantee that the training error will be low with high probability, therefore they are of little practical use (for a deeper discussion of the subtleties underlying the connection between NFL and VC theory, see (Wolpert, 1996b), (Wolpert, 1995))[4]. How could one guarantee that a training error close to zero is achievable? One has to use the prior information available on the target function. Along a similar vein, results that exemplify the superiority of algorithm A over algorithm B on a certain bench mark problem are also of little use, for the reverse will be true on some other "benchmark" problem, and we never know which "benchmark" problem we will run into. Similarly, looking at a data set alone, one cannot produce any reliable estimate for the noise variance. Perhaps we "feel" as though we ought to be able to (for, how often can it be heard: "this data set is clearly noisy"), but only because we already have a smoothness prior built into our neural network. Why? Because the types of functions we tend to encounter in practice are smooth.

NFL focuses our attention on the importance of the prior information available. One should not attempt to find learning systems that are universally good. Instead, one should study the possibilities that various priors present. Unfortunately, identifying and justifying the prior for the problem

[4]

VC results make statements about $Pr[training\ error|test\ error]$ by making the statement that the probability of a low training error given a high test error is extremely unlikely. In practice, what we would really like to make statements about is the $Pr[test\ error|training\ error]$ for which we really need to make some statement about the *apriori* probablity of a certain test error, i.e., we need to make some statement about a prior over target functions.

at hand is already a daunting task, let alone incorporating it into the learning problem. One way would be through the use of hints (?), (Sill, 1998), (Sill & Abu-Mostafa, 1997). A possible starting point would be to identify priors, $P[f]$, with learning systems that perform well with respect to them, for example (Barber & Williams, 1997), (Zhu & Rohwer, 1996). One might then be able to relate real problems to these priors and thus choose an appropriate learning system, or at least it might become possible to make precise the assumptions behind a belief that a certain learning system is appropriate for a certain problem. What would be a useful result? One of the form "Algorithm A (or Learning System A) performs better than average on problems drawn from the (useful) class B".

6 Appendix

6.1 Estimating Noise Variance using Linear Models

We have $\hat{A}$, an estimate of $\sigma^2(d+1)+B$ and we would like to estimate B by bootstrapping. Suppose we draw N_B data points and estimate B (using the bootstrap technique) by $\hat{B}$. It can be shown that (theorem 6.2)

$$\left\langle \hat{B} \right\rangle_{\mathbf{n}} = \tilde{B} + \sigma^2(d+1)(1-\frac{N_B}{N}) + \frac{\sigma^2}{N} \underbrace{tr\left\langle \Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{V} \right\rangle}_{O(d+1)} \tag{14}$$

where $\tilde{B}$ is an unbiased estimate of B and the expectation in the last term is with respect to the bootstrapping. The definitions of $\mathbf{V}$ and Σ are given in the next section and are unimportant for our present purpose. Thus, we now have two quantities, $\hat{A}$, an estimate of $B+\sigma^2(d+1)$ and $\hat{B}$, an estimate of $B+\sigma^2((d+1)(1-N_B/N)+tr\langle\Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{V}\rangle/N)$. Solving for σ^2, we have

$$\sigma^2 = \frac{\hat{A}-\hat{B}}{(d+1)\frac{N_B}{N}-\frac{tr\langle\Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{V}\rangle}{N}} \tag{15}$$

However, we are bootstrapping a quantity B, which is related to the deviation of a statistical quantity of the sample from its population value. This estimate is good only as long as N represents the population – i.e., $N_B/N \rightarrow 0$. In this case, $\hat{B}=B+\sigma^2(d+1)$ and thus we cannot combine this with $\hat{A}$ to solve for σ^2 (the expression in (15) becomes indeterminate). We are faced with a dilemma. For finite N_B/N we can get an estimate of σ^2 but this estimate is not very good. However, we have no way to make it better because by making the bootstrap more effective for B, the final result cannot be solved for σ^2 - technically speaking, we cannot in this way get a consistent estimate of σ^2, even as $N \rightarrow \infty$.

6.1.1 Proof of Theorems

We use the notation

$$\mathbf{X} = [\,\mathbf{x}_1 \quad \mathbf{x}_2 \quad \ldots \quad \mathbf{x}_N\,], \qquad \mathbf{y}=\mathbf{f}+\mathbf{n}$$

$$\Sigma \equiv \left\langle \mathbf{x}\mathbf{x}^T \right\rangle_{\mathbf{x}}, \qquad \mathbf{q} = \langle \mathbf{x} f(\mathbf{x})\rangle_{\mathbf{x}}$$

$\mathbf{X}$ is a $(d+1)\times N$ matrix where the $\mathbf{x}_i$ are the data points (we use the convention that the first component is set to 1 and the remaining d components are the coordinates of the data point). $\mathbf{f}$ is an $N\times 1$ vector of output values at the data points and $\mathbf{n}$ is a vector of noise realizations. If $\langle\mathbf{x}\rangle = 0$ then Σ is the covariance matrix for the input distribution. The law of large numbers gives us that $\mathbf{X}\mathbf{X}^T/N \underset{N\rightarrow\infty}{\longrightarrow} \Sigma$ and $\mathbf{X}\mathbf{f}/N \underset{N\rightarrow\infty}{\longrightarrow} \mathbf{q}$. where we assume that the conditions for this to happen are satisfied. We thus define $\mathbf{V}$ and $\mathbf{a}$ by

$$\frac{\mathbf{X}\mathbf{X}^T}{N} = \Sigma + \frac{\mathbf{V}(\mathbf{X})}{\sqrt{N}}, \qquad \frac{\mathbf{X}\mathbf{f}}{N} = \mathbf{q} + \frac{\mathbf{a}(\mathbf{X})}{\sqrt{N}} \tag{16}$$

Note that $\langle \mathbf{V} \rangle_{\mathbf{X}} = \langle \mathbf{a} \rangle_{\mathbf{X}} = 0$ and $\mathrm{Var}(\mathbf{V})$ and $\mathrm{Var}(\mathbf{a})$ are $O(1)$.

Theorem 6.1 *Let the learning model be the set of linear functions $\mathbf{w} \cdot \mathbf{x}$ and let the learning algorithm be minimization of the squared error. Then*

$$\langle {\mathcal{E}_{tr}}^* \rangle_{\mathcal{D},\mathbf{n}} = E_0 + \sigma^2 - \frac{\sigma^2(d+1) + B}{N} + O\left(\frac{1}{N^{\frac{3}{2}}}\right) \tag{17}$$

where E_0 and B are constants given by

$$E_0 = \left\langle f^2 \right\rangle - \mathbf{q}^T \Sigma^{-1} \mathbf{q} \tag{18}$$

$$B = \frac{\left\langle \mathbf{q}^T \Sigma^{-1} \mathbf{V} \Sigma^{-1} \mathbf{V} \Sigma^{-1} \mathbf{q} + \mathbf{a}^T \Sigma^{-1} \mathbf{a} - 2\mathbf{a}^T \Sigma^{-1} \mathbf{V} \Sigma^{-1} \mathbf{q} \right\rangle}{N} \tag{19}$$

Proof: The Least Squares estimate of $\mathbf{w}$ is given by

$$\hat{\mathbf{w}} = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{y} \tag{20}$$

from which we calculate the expected training error as

$$\begin{aligned}
\langle \mathcal{E}_{tr} \rangle &= \left\langle \frac{1}{N}(\mathbf{X}^T\hat{\mathbf{w}} - \mathbf{y})^2 \right\rangle = \left\langle \frac{1}{N}((\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X} - \mathbf{1})\mathbf{y})^2 \right\rangle \\
&= \frac{\left\langle \mathbf{f}^T\mathbf{f} \right\rangle - \left\langle f^T\mathbf{X}(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{f} \right\rangle + \left\langle \mathbf{n}^T\mathbf{n} \right\rangle - \left\langle \mathbf{n}^T\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{n} \right\rangle}{N} \\
&= \underbrace{\left\langle f^2 \right\rangle - \frac{\left\langle \mathbf{f}^T X^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{f} \right\rangle}{N}}_{T_1} + \sigma^2 - \frac{\sigma^2(d+1)}{N}
\end{aligned}$$

Using (16) and the identity $[1 + \lambda\mathbf{A}]^{-1} = 1 - \lambda\mathbf{A} + \lambda^2\mathbf{A}^2 + O(\lambda^3)$ we evaluate

T_1 as

$$\begin{aligned} T_1 &= \left\langle f^2 \right\rangle - \mathbf{q}^T\Sigma^{-1}\mathbf{q} \\ &\quad - \frac{\left\langle \mathbf{q}^T\Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{q} + \mathbf{a}^T\Sigma^{-1}\mathbf{a} - 2\mathbf{a}^T\Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{q} \right\rangle}{N} + O\left(\frac{1}{N^{\frac{3}{2}}}\right) \end{aligned}$$

The theorem now follows. ■

This result is similar to results obtained by (Amari, Murata, Müller, & Yang, 1997),(Moody, 1991). From theorem 6.1, we have that

$$B = \frac{1}{N}\left(\underbrace{\left\langle \mathbf{q}^T\Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{q} \right\rangle}_{T_1} + \underbrace{\left\langle \mathbf{a}^T\Sigma^{-1}\mathbf{a} \right\rangle}_{T_2} - 2\underbrace{\left\langle \mathbf{a}^T\Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{q} \right\rangle}_{T_3} \right) \tag{21}$$

We wish to estimate T_1, T_2, T_3. Suppose that we try to bootstrap them as follows. For T_1, we take $\hat{\mathbf{q}} = \mathbf{X}\mathbf{y}/N = \mathbf{X}(\mathbf{f}+\mathbf{n})/N$ and $\hat{\Sigma} = (\mathbf{X}\mathbf{X}^T/N)$. We estimate $\mathbf{V}$ by sampling N_B of the N data points and compute $\sqrt{N_B}((\mathbf{X}_B\mathbf{X}_B^T/N_B) - (\mathbf{X}\mathbf{X}^T/N))$. We then estimate T_1 by $\hat{T}_1$, a bootstrapped expectation of $\hat{\mathbf{q}}^T\hat{\Sigma^{-1}}\mathbf{V}\hat{\Sigma^{-1}}\mathbf{V}\hat{\Sigma^{-1}}\hat{\mathbf{q}}$. This requires N_B/N to be small. Doing all this, we find that

$$\hat{T}_1 = \frac{\mathbf{f}^T\mathbf{X}^T}{N}\left\langle \hat{\Sigma^{-1}}\mathbf{V}\hat{\Sigma^{-1}}\mathbf{V}\hat{\Sigma^{-1}} \right\rangle \frac{\mathbf{X}\mathbf{f}}{N} + \frac{\sigma^2}{N} tr\left\langle \hat{\Sigma^{-1}}\mathbf{V}\hat{\Sigma^{-1}}\mathbf{V} \right\rangle \tag{22}$$

where we have taken the expectation with respect to the noise and the remaining expectation is with respect to the bootstrapping. Notice that the first term is the unbiased estimator of T_1 – denote this term by $\tilde{T}_1$.

For $\hat{T}_2 = \left\langle \mathbf{a}^T\Sigma^{-1}\mathbf{a} \right\rangle$ we use $\mathbf{a} = \sqrt{N_B}(\mathbf{X}_B\mathbf{y}_B/N_B - \hat{\mathbf{q}})$. Partitioning the input matrix and noise vector into the part in the sampled N_B points and the remaining part, write $\mathbf{X} = [\mathbf{X}_B\ \tilde{\mathbf{X}}_B]$ and $\mathbf{n} = [\mathbf{n}_B^T\ \tilde{\mathbf{n}}_B^T]^T$. Noting that $\mathbf{n}_B$

and $\tilde{\mathbf{n}}_B$ are independent, we have for $\hat{T}_2$

$$\begin{aligned}
\hat{T}_2 &= \tilde{T}_2 + \frac{1}{N^2 N_B}(N_B\mathbf{X}\mathbf{n} - N\mathbf{X}_B\mathbf{n}_B)\Sigma^{-1}(N_B\mathbf{X}\mathbf{n} - N\mathbf{X}_B\mathbf{n}_B) \\
&= \tilde{T}_2 + \frac{N_B^2(N-N_B)^2}{N^2 N_B}\left[\frac{\tilde{\mathbf{X}}_B\tilde{\mathbf{n}}_B}{N-N_B} - \frac{\tilde{\mathbf{X}}_B\mathbf{n}_B}{N_B}\right]\Sigma^{-1}\left[\frac{\tilde{\mathbf{X}}_B\tilde{\mathbf{n}}_B}{N-N_B} - \frac{\tilde{\mathbf{X}}_B\mathbf{n}_B}{N_B}\right] \\
&= \tilde{T}_2 + \sigma^2(d+1)\left(1-\frac{N_B}{N}\right) + o\left(\frac{1}{N}\right) \qquad (23)
\end{aligned}$$

where we have taken the expectation with respect to the noise and $\tilde{T}_2$ is an unbiased estimate of T_2.

Performing the same kind of analysis for $\hat{T}_3$ we find that $\hat{T}_3 = \tilde{T}_3$, an unbiased estimate of T_3. Now we estimate $\hat{B}$ by $\hat{T}_1 + \hat{T}_2 - 2\hat{T}_3$ which is the unbiased estimate of B plus some correction terms. Therefore we have proved the following theorem.

Theorem 6.2 *For $N \to \infty$, $N_B/N \to 0$, the bootstrapped estimate of B, which we call $\hat{B}$ has an expectation given by*

$$\left\langle \hat{B}\right\rangle_{\mathbf{n}} = \tilde{B} + \frac{\sigma^2(d+1)}{N}\left(1-\frac{N_B}{N}\right) + \frac{\sigma^2}{N}tr\left\langle \Sigma^{-1}\mathbf{V}\Sigma^{-1}\mathbf{V}\right\rangle_{\mathbf{X}} \qquad (24)$$

where $\tilde{B}$ is an unbiased estimate of B and the remaining expectation is with respect to the bootstrapping.

■

7 Acknowledgments

We would like to thank Yaser Abu-Mostafa and the members of the Caltech Learning Systems Group for helpful comments. In addition, two anonymous referees provided useful comments.

References

Amari, S., Murata, N., Müller, K., & Yang, H. (1997). Asymptotic statistical theory of overtraining and cross validation. *IEEE Transactions on Neural Networks*, *8*(5), 985-996.

Barber, D., & Williams, C. K. I. (1997). Gaussian processes for bayesian classification via hybrid monte carlo. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems (nips)* (Vol. 9, pp. 340–346). Morgan Kaufmann.

Cataltepe, Z., Abu-Mostafa, Y. S., & Magdon-Ismail, M. (1998). No free lunch for early stopping. *To appear in Neural Computation.*

Cortes, C., Jackel, L. D., & Chiang, W.-P. (1994). Limits on learning machine accuracy imposed by data quality. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Neural information processing systems (nips)* (Vol. 7, p. 239-246).

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory.* Wiley.

Moody, J. E. (1991). The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in neural information processing systems (nips)* (Vol. 4, pp. 847–854).

Shao, J., & Tu, D. (1996). *The jackknife and the bootstrap.* New York: Springer-Verlag.

Sill, J. (1998). Monotonic networks. In *Advances in neural information processing systems (nips)* (Vol. 10).

Sill, J., & Abu-Mostafa, Y. S. (1997). Monotonicity hints. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems (nips)* (Vol. 9, pp. 634–640). Morgan Kaufmann.

Vapnik, V. N. (1995). *The nature of statistical learning theory.* Springer–Verlag.

Wolpert, D. H. (1996a). The existence of a priori distinctions between learning algorithms. *Neural Computation*, *8*(7), 1391–1420.

Wolpert, D. H. (1996b). The lack of a priori distinctions between learning algorithms. *Neural Computation*, *8*(7), 1341–1390.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*, 67–82.

Wolpert, e., D. H. (1995). *The mathematics of generalization.* New York: Addison Wesley.

Zhu, H. (1996). No free lunch for cross validation. *Neural Computation*, *8*(7), 1421–1426.

Zhu, H., & Rohwer, R. (1996). Bayesian regression filters and the issue of priors. *Neural Computation Applications*, *4*, 130–142.