On Learning Functions from Noise-Free and Noisy Samples via Occam's Razor

Balas K. Natarajan
Computer Systems Laboratory
HPL-94-112
November, 1994

learning theory, noise, non-linear filters

An Occam approximation is an algorithm that takes as input a set of samples of a function and a tolerance ε, and produces as output a compact representation of a function that is within ε of the given samples. We show that the existence of an Occam approximation is sufficient to guarantee the probably approximate learnability of classes of functions on the reals even in the presence of arbitrarily large but random additive noise. One consequence of our results is a general technique for the design and analysis of non-linear filters in digital signal processing.

Internal Accession Date Only

1. Introduction

We begin with an overview of our main result. Suppose we are allowed to randomly sample a function f on the reals, with the sample values of f being corrupted by additive random noise ν of strength b. Let g be a sparse but approximate interpolant of the random samples of f, sparse in the sense that it can be compactly represented, and approximate in the sense that the interpolation error in g with respect to the samples is at most b. Intuitively, we would expect that the noise and the interpolation error add in their contribution to the error in g with respect to f, i.e., $\|g - f\| \sim 2b$. However, our results show that the noise and the interpolation error tend to cancel, i.e., $\|g - f\| \sim 0$, with the extent of the cancellation depending on the sparseness of g and how often f is sampled.

We consider the paradigm of probably approximately correct learning of Valiant, as reviewed in Natarajan (1991), Anthony and Biggs (1992). Broadly speaking, the paradigm requires an algorithm to identify an approximation to an unknown *target* set or function, given random samples of the set or function. Of central interest in the paradigm is the relationship between the complexity of the algorithm, the a priori information available about the target set or function, and the goodness of the approximation. When learning sets, if the target set is known to belong to a specific *target* class, Blumer et al. (1991) establish that the above quantities are related via the *Vapnik-Chervonenkis* dimension of the target class, and if this dimension is finite, then learning is possible. They also show that even if the Vapnik-Chervonenkis dimension is infinite, learning may still be possible via the use of Occam's Razor. Specifically, if it is possible to compress collections of examples for the target set, then a learning algorithm exists. For instance, the class of sets composed of finitely many intervals on the reals satisfy this condition and hence can be learned. This tacitly implies that the class of Borel sets can be learned, since each Borel set can be approximated arbitrarily closely by the union of finitely many intervals. Kearns and Schapire (1990) extend the above results to the case of probabilistic concepts, and offer an Occam's Razor result in that setting. For the case of functions, Natarajan (1989) examines conditions under which learning is possible, and shows that it is sufficient if the "graph" dimension of the target class is finite. Haussler (1989) generalizes this result to a "robust" model of learning, and shows that it is sufficient if the "metric" dimension of the class is finite. Learning sets in the presence of noise was studied by Angluin and Laird (1988), Kearns and Li (1993), and Sloan (1988) amongst others. Kearns and Li (1988) show that the principle of Occam's Razor can be used to learn in the presence of a limited amount of noise.

We consider the learning of functions, both with and without random sampling noise. In the latter case, we mean that the function value obtained in the sampling process is corrupted by additive random noise. In this setting, we show that if it is

possible to construct sparse but approximate interpolants to collections of examples, then a learning algorithm exists. For instance, it is possible to construct optimally sparse but approximate piecewise linear interpolants to collections of examples of univariate functions. As a consequence, we find that the class of all univariate Baire functions, (see Feller (1957)), can be learned in terms of the piecewise linear functions, with the number of examples required to learn a particular Baire function depending on the number of pieces required to approximate it as a piecewise linear function. In the absence of noise, our results hold for all L_p metrics over the space of functions. In the presence of noise, the results are for the L_2 and L_∞ metrics. There are two points of significance in regard to these results: (1) The noise need not be of zero mean for the L_∞ metric. (2) For both metrics, the magnitude of the noise can be made arbitrarily large and learning is still possible, although at increased cost.

Our results are closely related to the work in signal processing, Oppenheim and Schafer (1974), Papoulis (1965), Jazwinski (1970), and Krishnan (1984), on the reconstruction and filtering of discretely sampled functions. However, much of that literature is on linear systems and filters, with the results expressed in terms of the Fourier decomposition, while relatively little is known about non-linear systems or filters. Our results allow a unified approach to both linear and non-linear systems and filters, and in that sense, offer a new and general technique for signal reconstruction and filtering. This is explored in Natarajan (1993a), (1994), where we analyze a broad class of filters that separate functions with respect to their encoding complexity – since random noise has high complexity in any deterministic encoding and is hard to compress, it can be separated from a function of low complexity.

Of related interest is the literature on sparse polynomial interpolation, Berlekamp (1970), Ben-Or and Tiwari (1988), Grigoriev et al. (1990), and Ar et al. (1992). These authors study the identification of an unknown polynomial that can be evaluated at selected points.

The results in this paper appeared in preliminary form in Natarajan (1993b). More recently, Bartlett et al. (1994) and Anthony and Bartlett (1994) pursue further the implications of approximate interpolation on the learnability of real-valued functions with and without noise.

2. Preliminaries

We consider functions $f : [0,1] \rightarrow [-K,K]$, where $[0,1]$ and $[-K,K]$ are intervals on the reals **R**. A *class* of functions F is a set of such functions, and is said to be of *envelope* K. In the interest of simplicity, we fix $K = 1$. A *complexity measure* l is a mapping from F to the natural numbers **N**. An *example* for a function f is a pair

$(x,f(x))$. Our discussion will involve metrics on several spaces. For the sake of concreteness, we will deal only with the L_p metrics, $p \in \mathbf{N}, p \geq 1$ and will define these exhaustively. For two functions f and g and probability distribution P on [0,1],

$$L_p(f, g, P) = \left(\int_x | f(x) - g(x) |^p dP \right)^{1/p}.$$

For function f and collection of examples $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$,

$$L_p(f, S) = \left(\frac{1}{m} \sum | f(x_i) - y_i |^p \right)^{1/p}.$$

Let l be a complexity measure on a class G, f a function not necessarily in G, L a metric, and P a distribution on [0,1]. For $\epsilon > 0$, $l_{\min}(f, \epsilon, P, L)$ is defined by

$$l_{\min}(f, \epsilon, P, L) = \min \{l(g) \mid g \in G, L(f, g, P) \leq \epsilon\}.$$

If $\{l(g) \mid g \in G, L(f, g, P) \leq \epsilon\}$ is empty, then $l_{\min}(f, \epsilon, P, L) = \infty$.

Analogously, for a collection of examples S and metric L,

$$l_{\min}(S, \epsilon, L) = \min \{l(g) \mid g \in G, L(g, S) \leq \epsilon\}.$$

If $\{l(g) \mid g \in G, L(g, S) \leq \epsilon\}$ is empty, then $l_{\min}(S, \epsilon, L) = \infty$.

If f and g are of envelope 1, it is easy to show that

$$\left(\frac{L_1(f, g, P)}{2} \right)^2 \leq L_p(f, g, P) \leq \left(L_1(f, g, P) \right)^{1/p}. \tag{1}$$

A learning algorithm A for *target class* F has at its disposal a subroutine EXAMPLE, that at each call returns an example for an unknown *target function* $f \in F$. The example is chosen at random according to an arbitrary and unknown probability distribution P on [0,1]. After seeing some number of such examples, the learning algorithm identifies a function g in the *hypothesis class* G such that g is a good approximation to f. Formally, we have the following. Algorithm A is a *learning algorithm* for F in terms of G, with respect to metric L, if (a) A takes as input ϵ and δ. (b) A may call EXAMPLE. (c) For all probability distributions P and all functions f in F, A identifies a function $g \in G$, such that with probability at least $(1 - \delta)$, $L(f, g, P) \leq \epsilon$.

The *sample complexity* of a learning algorithm for F in terms of G is the number of examples sought by the algorithm as a function of the parameters of interest. In noise-free learning, these parameters are ϵ, δ and $l_{\min}(f)$, where f is the target function. In the presence of random sampling noise, the properties of the noise distribution would be additional parameters. Since we permit our learning algorithms to be probabilistic, we will consider the *expected sample complexity* of a learning algorithm, which is the expectation of the number of examples sought by the algorithm as a function of the parameters of interest.

In light of the relationship between the various L_p metrics established in (1), we focus on the L_1 metric and unless we explicitly state otherwise, use the term learning algorithm to signify a learning algorithm with respect to the L_1 metric.

We denote the probability of an event A occurring by $\mathbf{Pr}\{A\}$, and the expected value of a random variable x by $\mathbf{E}\{x\}$. Let $r:\mathbf{N}\rightarrow\mathbf{N}$; $r(m)$ is said to $o(m)$ if $\lim_{m\rightarrow\infty} r(m)/m = 0$.

With respect to a class of functions F of envelope K, $\epsilon > 0$ and $m \in \mathbf{N}$, and metric L: The size of the minimum ϵ-cover of a set $X = \{x_1, x_2, ..., x_m\}$ is the cardinality of the smallest set of functions G from X to $\mathbf{R}$ such that for each $f \in F$, there exists $g \in G$ satisfying $L(f,g,P_X) \leq \epsilon$, where P_X is the distribution placing equal mass $1/m$ on each of the x_i in X. The *covering number* $N(F, \epsilon, m, L)$ is the maximum of the size of the minimum ϵ-cover over all $\{x_1, x_2, ..., x_m\}$.

The following convergence theorem will be of considerable use to us.

Theorem 1: Pollard (1984) pp. 25-27. For a class of functions F with envelope K, the probability that the observed mean over m random samples of any function in F will deviate from its true mean by more than ϵ is, when $m \geq 8/\epsilon^2$, at most

$$8N(F, \epsilon/8, m, L_1)e^{-(m/128)(\epsilon/K)^2}.$$

3. Occam Approximations

An *approximation* algorithm Q for hypothesis class G and metric L is an algorithm that takes as input a collection S of examples and a tolerance $\epsilon > 0$, and identifies in its output a function $g \in G$ such that $L(g, S) \leq \epsilon$, if such exists.

Fix $m, t \in \mathbf{N}$ and $\epsilon > 0$. Let $\hat{G}$ be the class of all functions identified in the output of Q when its input ranges over all collections S of m examples such that $l_{\min}(S, \epsilon, L) \leq t$. Q is said to be an *Occam approximation*, if there exists polynomial $q(a,b):\mathbf{N}\times\mathbf{R}\rightarrow\mathbf{N}$, and $r:\mathbf{N}\rightarrow\mathbf{N}$ such that $r(m)$ is $o(m)$, and for all m, t, ϵ, and $\zeta > 0$, $\log(N(\hat{G}, \zeta, m, L)) \leq q(t, 1/\zeta)r(m)$. We say that (q, r) is the

characteristic of Q. The above notion of an *Occam approximation* is a generalization to functions of the more established notion of an Occam algorithm for concepts, Blumer et al. (1991), and its extension to probabilistic concepts in Kearns and Schapire (1990).

Example 1 Let G be the class of univariate polynomials. As complexity measure on G, we choose the degree of the polynomial. Consider the following approximation algorithm Q with respect to the L_2 metric. Given is a collection of samples $S = \{(x_1, y_1), (x_2, y_2)..., (x_m, y_m)\}$, and tolerance ϵ. We shall fit the data with a polynomial of the form $a_0 + a_1 x + \cdots a_i x^i + \cdots a_d x^d$. For $d = 0, 1, 2, \cdots$ construct a linear system $Xa = y$, where X is the matrix of m rows and d columns, with row vectors of the form $(1, x_i, x_i^2 \cdots, x_i^d)$, y is the column vector $(y_1, y_2, ..., y_m)$, and a is the column vector $(a_0, a_1, ..., a_d)$. Find the smallest value of d for which the least squares solution of the linear system has residue at most ϵ in the L_2 norm. The solution vector a for this value of d determines the polynomial g. This computation can be carried out efficiently, Golub and Van Loan (1983).

Claim 1: Algorithm Q is Occam.

Proof: Fix the number of samples m, the degree d of the polynomial, and the tolerance ϵ. For collection of samples S, $l_{\min}(S, \epsilon, L_2)$ is the degree of the lowest-degree polynomial that fits the samples in S with error at most ϵ in the L_2 metric. By construction, Q outputs only polynomials of degree d on such inputs. Thus, $\hat{G}$ is a subset of the class of polynomials of degree d. By Claim 2,

$$\log(N(\hat{G}, \zeta, m, L_2) \leq (d+1)\log(m^2/\zeta),$$

which implies that q is Occam. □

Claim 2: If F is the class of polynomials of degree d,

$$N(F, \zeta, m, L_2) \leq \left(\frac{(d+1)^{d+1}}{\zeta} \right)^{d+1}.$$

Proof: Let $f(x)$ be a polynomial of degree d, and let $x_1, x_2, \ldots,$ be $d+1$ uniformly spaced points in $[0,1]$, such that $x_1 = 0$ and $x_{d+1} = 1$. Let $y_i = f(x_i)$. We can write f in Lagrange form as

$$f(x) = \sum_{i=1}^{d+1} \prod_{i \neq j} \frac{x - x_i}{x_j - x_i} y_i .$$

Since $|x_j - x_i| \geq 1/(d+1)$, it follows from the above equation that

$$\left| \frac{\partial f(x)}{\partial y_i} \right| \leq (d+1)^d ,$$

for $x \in [0,1]$. Let $\hat{y}_i$ denote y_i rounded off to the nearest multiple of $\zeta/(d+1)^{d+1}$. It is clear that the polynomial

$$\hat{f}(x) = \sum_{i=1}^{d+1} \prod_{i \neq j} \frac{x - x_i}{x_j - x_i} \hat{y}_i .$$

is within ζ of $f(x)$ everywhere on the interval $[0,1]$. We have therefore shown that for every polynomial f of degree d, there exists a polynomial $\hat{f}$ of degree d within ζ of f in the L_2 metric, and such that $\hat{f}$ takes on values spaced $\zeta/(d+1)^{d+1}$ apart on the uniformly spaced points $x_1, x_2 \cdots x_{d+1}$. Since $\hat{f}$ can be constructed by choosing one of $\zeta/(d+1)^{d+1}$ values at each of the $d+1$ points, there are at most $((d+1)^{(d+1)}/\zeta)^{d+1}$ choices for $\hat{f}$. Hence the claim. □

Example 2: Another class with an Occam approximation algorithm with respect to the L_2 metric is the class of trigonometric interpolants with complexity measure the highest frequency, i.e., functions of the form

$$f(x) = \sum_{i=0}^{\infty} A_i \cos(2\pi i x) + \sum_{i=0}^{\infty} B_i \sin(2\pi i x) .$$

For the class of such functions, the least squares technique described in the context of the polynomials in Example 1 can be used to construct an Occam approximation.

4. Noise-Free Learning

In this section we show that the existence of an Occam approximation is sufficient to guarantee the efficient learnability of a class of functions in the absence of noise. The theorem is stated for Occam approximations with respect to the L_1 metric, and subsequently extended to other metrics.

Theorem 2: Let F be the target class and G the hypothesis class with complexity measure l, both of envelope 1. Let Q be an Occam approximation for G, with respect to metric L_1, with characterstic (q,r). Then, there exists a learning algorithm for F with expected sample complexity polynomial in $1/\epsilon$, $1/\delta$, and $l_{\min}(f,\epsilon/8,P,L_1)$.

Proof: We claim that Algorithm A_1 below is a learning algorithm for F. Let f be the target function. In words, the algorithm makes increasingly larger guesses for $l_{\min}(f)$, and based on its guesses constructs approximations for f, halting if and

Algorithm A_1
input ϵ, δ
begin
$t = 2$;
repeat forever
let m be the least integer satisfying

$$m \geq \left(\frac{9216}{\epsilon^2} q(t, 32/\epsilon) \right) r(m);$$

make m calls of EXAMPLE to get collection S;

let $g = Q(S, \epsilon/4)$;

let $m_1 = 16t^2/(\epsilon^2\delta)$;

make m_1 calls of EXAMPLE to get collection S_1;
if $L_1(g, S_1) \leq (3/4)\epsilon$ **then** output g and halt;
else $t = t+1$;
end
end

when a constructed approximation appears to be good.

We first show that the probability of the algorithm halting with a function g as output such that $L_1(f, g, P) \geq \epsilon$ is at most than δ. At each iteration, by Chebyshev's inequality,

$$\mathbf{Pr}\left\{ | L_1(g,S_1) - L_1(g, f, P) | \geq \epsilon/4 \right\} \leq \frac{16}{\epsilon^2 m_1}.$$

Since $m_1 = 16t^2/(\epsilon^2\delta)$,

$$\mathbf{Pr}\left\{ | L_1(g,S_1) - L_1(g, f, P) | \geq \epsilon/4 \right\} \leq \frac{\delta}{t^2}.$$

If the algorithm halts, then $L_1(g,S_1) \leq (3/4)\epsilon$, and hence

$$\mathbf{Pr}\left\{ L_1(g, f, P) > \epsilon \right\} \leq \frac{\delta}{t^2}.$$

Hence, the combined probability that the algorithm halts at any iteration with

$L_1(g, f, P) > \epsilon$ is at most $\sum_{t=2}^{\infty} \frac{\delta}{t^2} < \delta$.

Let $t_0 = l_{\min}(f, \epsilon/8, P, L_1)$. We now show that when $t \geq t_0$, the algorithm halts with probability at least 1/4. Let $h \in G$ be such that $L_1(h, f, P) \leq \epsilon/8$ and $l(h) = t_0$. Fix $t \geq t_0$. By Chebyshev's inequality,

$$\mathbf{Pr}\left\{ |L_1(h, S) - L_1(h, f, P)| \geq \epsilon/8 \right\} \leq \frac{64}{\epsilon^2 m}.$$

Hence, if $m \geq 256/\epsilon^2$ with probability at least $(1 - 1/4) = 3/4$,

$$|L_1(h, S) - L_1(h, f, P)| \leq \epsilon/8.$$

Since $L_1(h, f, P) \leq \epsilon/8$, it follows that with probability 3/4, $L_1(h, S) \leq \epsilon/4$. In other words, with probability 3/4, $l_{\min}(S, \epsilon/4, L_1) \leq l(h) = t_0 \leq t$.

At each iteration of A_1, let $\hat{G}$ be the class of all functions that Q could output if $l_{\min}(S, \epsilon/4, L_1) \leq t$ holds. Since Q is Occam, $\log(N(\hat{G}, \epsilon/32, m)) \leq q(t, 32/\epsilon)r(m)$. Let H be the class of functions $H = \{h \,|\, h(x) = |f(x) - g(x)|, g \in \hat{G}\}$, and let h_1 and h_2 be any two functions in H. Now,

$$|h_1(x) - h_2(x)| = \Big| |f(x) - g_1(x)| - |f(x) - g_2(x)| \Big| \leq |g_1(x) - g_2(x)|.$$

Hence,

$$N(H, \epsilon/32, m, L_1) \leq N(\hat{G}, \epsilon/32, m, L_1) \leq q(t, 32/\epsilon)r(m).$$

If

$$m \geq \left(\frac{9216}{\epsilon^2} q(t, 32/\epsilon) \right) r(m),$$

m satisfies

$$8e^{q(t, 32/\epsilon)r(m)} e^{-(m/128)(\epsilon/4)^2} \leq 1/4,$$

and then by Theorem 1, with probability at least $(1 - 1/4) = 3/4$, the observed mean of each function $h \in H$ will be within $\epsilon/4$ of its true mean. That is, with probability at least 3/4, for each $g \in \hat{G}$,

$$\Big| L_1(g, S) - L_1(f, g, P) \Big| \leq \epsilon/4.$$

If g is the function that is returned by $Q(S, \epsilon/4)$, then $L_1(g, S) \leq \epsilon/4$. Therefore, if $l_{\min}(S, \epsilon/4, L_1) \leq t$, with probability at least 3/4, $L_1(f, g, P) \leq \epsilon/2$. Since we showed earlier that with probability at least 3/4, $l_{\min}(S, \epsilon/4, L) \leq t$, we get that with probability at least 1/2, the function g returned by Q will be such that $L_1(f, g, P) \leq \epsilon/2$.

We now estimate the probability that the algorithm halts when Q returns a function g such that $L_1(f, g, P) \leq \epsilon/2$. Once again by Chebyshev's inequality,

$$\mathbf{Pr}\left\{ \mid L_1(g,S_1) - L_1(g, f, P) \mid \geq \epsilon/4 \right\} \leq \frac{16}{\epsilon^2 m_1}.$$

Given that $L_1(f, g, P) \leq \epsilon/2$, and that $m_1 = 16t^2/(\epsilon^2\delta)$, we have

$$\mathbf{Pr}\left\{ L_1(g,S_1) \geq 3/4\epsilon \right\} \leq \frac{\delta}{t^2} \leq 1/2.$$

Hence, we have $\mathbf{Pr}\{L_1(g,S_1) < 3/4\epsilon\} \geq 1/2$. That is if the function g returned by Q is such that $L_1(f, g, P) \leq \epsilon/2$ then, A_1 will halt with probability at least 1/2.

We have therefore shown that when when $t \geq t_0$, the algorithm halts with probability at least 1/4. Hence, the probability that the algorithm does not halt within some $t > t_0$ iterations is at most $(3/4)^{t-t_0}$. Noting that the function $q(\,)$ is a polynomial in its arguments, it follows that the expected sample complexity of the algorithm is polynomial in $1/\epsilon$, $1/\delta$ and t_0. □

We now discuss the extension of Theorem 2 to other metrics. The one ingredient in the proof of Theorem 2 that does not directly generalize to the other L_p metrics is the reliance on Theorem 1, which is in terms of the L_1 metric. This obstacle can be overcome by using the relationship between the various L_p metrics as given by inequality (1), to note that $N(\hat{G},\zeta,m,L_1) \leq N(\hat{G},\zeta^2/4,m,L_p)$.

5. Learning in the Presence of Noise

We assume that the examples for the target function are corrupted by additive random noise in that EXAMPLE returns $(x, f(x)+\nu)$, where ν is a random variable. The noise variable ν is distributed in an arbitrary and unknown fashion.

We can show that the existence of an Occam approximation suffices to guarantee learnability in the presence of such noise, under some special conditions. Specifically we assume that the strength of the noise in the metric of interest is known a priori to the learning algorithm. Then, a learning algorithm would work as follows.

Obtain a sufficiently large number of examples, large enough that the observed strength of the noise is close to the true strength. Pass the observed samples through an Occam approximation, with tolerance equal to the noise strength. The function output by the Occam approximation is a candidate for a good approximation to the target function. Test this function against additional samples to check whether it is indeed close. Since these additional samples are also noisy, take sufficiently many samples to ensure that the observed noise strength in these samples is close to the true strength. The candidate function is good if it checks to be roughly the noise strength away from these samples.

The essence of the above procedure is that the error allowed of the Occam approximation is equal to the noise strength, and the noise and the error subtract rather than add. In order for us to prove that this indeed the case, we must restrict ourselves to linear metrics, metrics L such that in the limit as the sample size goes to infinity, the observed distance between a function g in the hypothesis class and the noisy target function is the sum of the strength of the noise, and the distance between g and the noise-free target function. Two such metrics are the L_∞ metric, and the square of the L_2 metric when the noise is of zero mean.

5.1 Noise of known L_2 measure

We assume that the noise is of bounded magnitude, zero mean, and of known L_2 measure. Specifically, (1) we are given $b \geq 0$ such the noise ν is a random variable in $[-b, +b]$; (2) the noise ν has zero mean; (3) we are given the variance c of the noise. It is easy to see that statistical access to the noise variable ν is sufficient to obtain accurate estimates of the variance c.

Theorem 3: Let F be the target class and G the hypothesis class with complexity measure l, where F and G are of envelope 1 and $b+1$ respectively. Let Q be an Occam approximation for G with respect to the L_2 metric, with characterstic (q,r). Then, there exists a learning algorithm for F with expected sample complexity polynomial in $1/\epsilon$, $1/\delta$, $l_{\min}(f,\epsilon/8,P,L_2^2)$ and the noise bound b.

Proof: Let f be the target function. Algorithm A_2 below is a learning algorithm for F in the L_2^2 metric, i.e., on input ϵ and δ, with probability at least $1-\delta$, the algorithm will identify a function g such that $L_2^2(f,g,P) \leq \epsilon$.

We first show that the probability of the algorithm halting with a function g as output such that $L_2^2(f, g, P) > \epsilon$ is less than δ. Now,

$$L_2^2(g,S_1) = \frac{1}{m_1}\sum(g-(f+\nu))^2 = \frac{1}{m_1}\sum(g-f)^2 + \frac{2}{m_1}\sum\nu(g-f) + \frac{1}{m_1}\sum\nu^2 ,$$

Algorithm A_2
input ϵ, δ, noise bound b, noise variance c;
begin
$t = 2$;
repeat forever
let m be the least integer satisfying

$$m \geq \frac{73288}{\epsilon^2} q\left(t, \left(\frac{256(b+2)}{\epsilon}\right)^2\right) r(m) + \frac{4096(b+1)^4}{\epsilon^2};$$

make m calls of EXAMPLE to get collection S;

let $g = Q(S, \sqrt{c+\epsilon/4})$;

let $m_1 = \frac{256(b+1)^4 t^2}{\delta\epsilon^2}$;

make m_1 calls of EXAMPLE to get collection S_1;
if $L_2^2(g, S_1) \leq c + (3/4)\epsilon$ **then** output g and halt;
else $t = t+1$;
end
end

where the summation is over the samples in S_1. Since $\mathbf{E}\{\nu(g-f)\} = 0$ and $\mathbf{E}\{\nu^2\} = c$, and $\mathbf{E}\{(g-f)^2\} = L_2^2(g,f,P)$, it follows that $\mathbf{E}\{L_2^2(g,S_1)\} = L_2^2(g,f,P) + c$.

Noting that $(g - (f+\nu))^2 \leq (2(b+1))^2$, we can invoke Chebyshev's inequality to write,

$$\mathbf{Pr}\left\{\left|L_2^2(g,S_1) - L_2^2(g,f,P) - c\right| \geq \epsilon/4\right\} \leq \frac{256(b+1)^4}{m_1\epsilon^2}. \tag{2}$$

If $L_2^2(g,f,P) > \epsilon$, (2) implies that

$$\mathbf{Pr}\{L_2^2(g,S_1) < c + 3/4\epsilon\} \leq \frac{256(b+1)^4}{m_1\epsilon^2}.$$

If

$$m_1 = \frac{256(b+1)^4 t^2}{\delta\epsilon^2},$$

then,

$$\mathbf{Pr}\{L_2^2(g,S_1) < c+3/4\epsilon\} \leq \frac{\delta}{t^2}\,. \tag{3}$$

Summing over all $t \geq 2$, we get that the probability that the algorithm halts with $L_2^2(g,f,P) > \epsilon$ is at most δ.

Let $h \in G$ be such that $L_2^2(h,f,P) \leq \epsilon/8$ and $l_{\min}(f,\epsilon/8,P,L_2^2) = l(h) = t_0$. As in (2) and (3), with Chebyshev's inequality we can show that

$$\mathbf{Pr}\{L_2^2(g,S) > c+1/4\epsilon\} \leq \frac{1024(b+1)^4}{m\epsilon^2}\,.$$

If $m \geq 4096(b+1)^4/\epsilon^2$, then $\mathbf{Pr}\{L_2^2(g,S) > c+\epsilon/4\} \leq 1/4$ and as a consequence

$$\mathbf{Pr}\{l_{\min}(S,c+\epsilon/4,L_2^2) \leq l(h) \leq t_0\} \geq 3/4\,. \tag{4}$$

Let $\hat{G}$ be the class of functions that Q could output on inputs S and $\sqrt{c+\epsilon/4}$, when S satisfies $l_{\min}(S,c+\epsilon/4,L_2^2) \leq t_0$. Let H be the class of functions $\{(f-g)^2 \mid g \in \hat{G}\}$. Then, for every pair h_1 and h_2 in H, there exists g_1 and g_2 in $\hat{G}$ such that

$$\begin{aligned} |h_1 - h_2| &= |(f-g_1)^2 - (f-g_2)^2|\,. \\ &= |f^2+g_1^2 - 2fg_1 - f^2 - g_2^2 + 2fg_2|\,. \\ &= |g_1^2 - g_2^2 - 2f(g_1-g_2)|\,. \\ &= |(g_1-g_2)(g_1+g_2-2f)|\,. \end{aligned}$$

Since g_1 and g_2 have envelope $b+1$ and f has envelope 1, we can write

$$|(g_1-g_2)(g_1+g_2-2f)| \leq 2(b+2)\,|(g_1-g_2)|\,.$$

Hence,

$$|h_1-h_2| \leq 2(b+2)\,|(g_1-g_2)|\,,$$

and $L_1(h_1,h_2,P) \leq 2(b+2)L_1(g_1,g_2,P)$. Invoking (1), we get

$$L_1(h_1,h_2,P) \leq 2(b+2)L_1(g_1,g_2,P) \leq 4(b+2)\sqrt{L_2(g_1,g_2,P)}\,.$$

Combining the above with the assumption that Q is Occam, it follows that

$$N(H, \epsilon/64, m, L_1) \leq N\left(\hat{G}, \left(\frac{\epsilon}{(256(b+2)}\right)^2, m, L_2\right) \leq q\left(t_0, \left(\frac{256(b+2)}{\epsilon}\right)^2\right) r(m) .$$

Invoking Theorem 1, we have that if $t \geq t_0$, $l_{\min}(S, c+\epsilon/4, L_2^2) \leq t_0$, and

$$m \geq \frac{73288}{\epsilon^2} q\left(t, \left(\frac{256(b+2)}{\epsilon}\right)^2\right) r(m) ,$$

with probability at least 3/4 the observed mean over the m samples of S of each $h \in H$ will be within $\epsilon/8$ of its true mean. That is,

$$\mathbf{Pr}\left\{\left|\frac{1}{m}\sum(g-f)^2 - L_2^2(g,f,P)\right| \leq \epsilon/8\right\} \geq 3/4 .$$

Noting that above inequality is conditional on (4), we can remove the conditionality by combining it with (4) to get

$$\mathbf{Pr}\left\{\left|\frac{1}{m}\sum(g-f)^2 - L_2^2(g,f,P)\right| \leq \epsilon/8\right\} \geq 1/2 , \tag{5}$$

when

$$m \geq \frac{73288}{\epsilon^2} q\left(t, \left(\frac{256(b+2)}{\epsilon}\right)^2\right) r(m) + \frac{4096(b+1)^4}{\epsilon^2} .$$

Now,

$$L_2^2(g,S) = \frac{1}{m}\sum(g-f)^2 + \frac{2}{m}\sum v(g-f) + \frac{1}{m}\sum v^2 , \tag{6}$$

where the summation is over the m samples of S.

Once again by Chebyshev's inequality, we can write,

$$\mathbf{Pr}\left\{\left|\frac{2}{m}\sum v(g-f) + \frac{1}{m}\sum v^2 - c\right| \geq \epsilon/8\right\} \leq \frac{256(b+2)^4}{\epsilon^2 m} .$$

If

$$m \geq \frac{512(b+2)^4}{\epsilon^2} ,$$

then

$$\mathbf{Pr}\left\{\left|\frac{2}{m}\sum\nu(g\text{-}f)+\frac{1}{m}\sum\nu^2-c\right|\geq\epsilon/8\right\}\leq 1/2\,. \tag{7}$$

Combining (5), (6) and (7), we have

$$\mathbf{Pr}\left\{\left|L_2^2(g,S)-L_2^2(g,f,P)-c\right|\leq\epsilon/4\right\}\geq 1/4\,, \tag{8}$$

when

$$m\geq\frac{73288}{\epsilon^2}q\left(t,\left(\frac{256(b+2)}{\epsilon}\right)^2\right)r(m)+\frac{4096(b+1)^4}{\epsilon^2}\,.$$

If g is the function output by $Q(S,\sqrt{c+\epsilon/4})$, then $L_2^2(g,S)\leq c+\epsilon/4$ and (8) can be written as

$$\mathbf{Pr}\{L_2^2(g,f,P)\leq\epsilon/2\}\geq 1/4\,. \tag{9}$$

If $L_2^2(g,f,P)\leq\epsilon/2$, then by (2) we get

$$\mathbf{Pr}\{L_2^2(g,S_1)\leq 3/4\epsilon\}\geq 1-\frac{256(b+1)^4}{m_1\epsilon^2}\,.$$

If m_1 is chosen as in the algorithm, then we can rewrite the above as

$$\mathbf{Pr}\{L_2^2(g,S_1)\leq 3/4\epsilon\}\geq 1-\delta/t^2\geq 3/4. \tag{10}$$

Combining (9 and (10), we get that when $t\geq t_0$, with probability at lease 3/16, the function g output by $Q(S,\sqrt{c+\epsilon/4})$ will be such that $L_2^2(g,S_1)\leq 3/4\epsilon$ and the algorithm will halt. Hence, the probability that the algorithm does not halt within some $t>t_0$ iterations is at most $(13/16)^{t-t_0}$. Noting that the function $q(\,)$ is a polynomial in its arguments, it follows that the expected sample complexity of the algorithm is polynomial in $1/\epsilon$, $1/\delta$, t_0 and b. □

Our assumption that the noise lie in a bounded range $[-b,+b]$ excludes natural distributions over infinite spans such as the normal distribution. However, we can include such distributions by a slight modification to our source of examples. Specifically, assume that we are given a value of b such that the second moment of the noise variable outside the range $[-b,+b]$ is small compared to ϵ. Noting that

the target function f has range [-1,1] by assumption, we can screen the output of EXAMPLE, rejecting the examples with values outside [-(b+1),+(b+1)], and otherwise passing them onto the learning algorithm. Thus, the noise distribution effectively seen by the learning algorithm is of bounded range $[-b, +b]$. We leave it to the reader to calculate the necessary adjustments to the sampling rates of the algorithm.

5.2 Noise of known L_∞ measure

We assume that we know the L_∞ measure of the noise in that we are given

(1) $b \geq 0$ such that the noise ν is a random variable in $[-b, +b]$, not necessarily of zero mean. Also, the symmetry is not essential; it suffices if $b_1 \leq b_2$ are given, with $\nu \in [b_1, b_2]$.

(2) A function $\gamma(\epsilon)$ such that $\mathbf{Pr}\{\nu \in [b - \epsilon, b]\} \geq \gamma(\epsilon)$, and $\mathbf{Pr}\{\nu \in [-b, -b + \epsilon]\} \geq \gamma(\epsilon)$.

It is easy to see that statistical access to the noise variable ν is sufficient to obtain an accurate estimate of γ. We now revisit our definition of an Occam approximation to define the notion of a strongly Occam approximation.

For a function f, let band(f, ϵ) denote the set of points within ϵ of f, i.e., band$(f, \epsilon) = \{(x, y) \mid |y - f(x)| \leq \epsilon\}$. For a class F, band(F, ϵ) is the class of all band sets for the functions in F, i.e., band$(F, \epsilon) = \{\text{band}(f, \epsilon) \mid f \in F\}$. A class of sets F is said to shatter a set S if the set $\{f \bigcap S \mid f \in F\}$ is the power set of S. $\mathbf{D}_{VC}(F)$ denotes the Vapnik-Chervonenkis dimension of F, and is the cardinality of the largest set shattered by F. The Vapnik-Chervonenkis dimension is a combinatorial measure that is useful in establishing the learnability of sets, Blumer et al. (1991) or Natarajan (1991).

Let Q be an approximation algorithm with respect to the L_∞ metric. Fix $m, t \in \mathbf{N}$ and $\epsilon > 0$. Let $\hat{G}$ be the class of all functions identified in the output of Q when its input ranges over all collections S of m examples such that $l_{\min}(S, \epsilon, L) \leq t$. Q is said to be *strongly Occam*, if there exists function $q(a,b)$ that is polynomial in a and $1/b$, and $r : \mathbf{N} \to \mathbf{N}$ such that $r(m)\log(m)$ is $o(m)$, such that for all m, t, ϵ, and $\zeta > 0$, $\mathbf{D}_{VC}(\text{band}(\hat{G}, \zeta)) \leq q(t, 1/\zeta)r(m)$.

Example 2: Let F be the Baire functions, and G the class of piecewise linear functions with complexity measure the number of pieces.

Consider the following approximation algorithm Q with respect to the L_∞ metric.

Given is a collection of samples $S = \{(x_1, y_1), (x_2, y_2)..., (x_m, y_m)\}$, and tolerance ϵ. Using the linear time algorithm of Imai and Iri (1986), Suri (1988), construct a piecewise linear function g such that $|g(x_i) - y_i| \leq \epsilon$ for each of the samples in S, and g consists of the fewest number of pieces over all such functions.

Claim 3: Algorithm Q is strongly Occam.

Proof: Fix the number of samples m, tolerance ϵ and complexity bound t. For set of examples S, $l_{\min}(S, \epsilon, L_2)$ is the fewest number of pieces in any piecewise linear function g such that $L_\infty(g,S) \leq \epsilon$. By construction, Q outputs such a function. Thus $\hat{G}$ is a subset of the class of all piecewise linear functions of t pieces. By Claim 4, for all ζ, $\mathbf{D}_{VC}(\text{band}(\hat{G}, \zeta)) \leq 7t$, and the claim follows. □

Claim 4: If F is the class of piecewise linear functions of at most t pieces, for all ζ, $\mathbf{D}_{VC}(\text{band}(F, \zeta)) \leq 7t$.

Proof: Assume that we are given a set S of more than $7t$ points that is shattered by band(F, ζ). Let $S = \{(x_1, y_1), (x_2, y_2)..., (x_m, y_m)\}$, where the x_i are in increasing order. We shall construct a subset S_1 of S that is not induced by any set in band(F, ζ), thereby contradicting the assumption that band(F, ζ) shatters S. Start with $S_1 = \{(x_1, y_1), (x_7, y_7)\}$. Now some three of (x_i, y_i) for $i = 2,3,..,6$ must lie on the same side of the line joining (x_1, y_1) and (x_7, y_7). Call these points a, b, and c, in order of their x coordinate. If b is within the quadrilateral formed by (x_1, y_1), (x_7, y_7), a, and c, add a and c to S_1, else add b to S_1. Repeat this procedure with the rest of the points in S. The resulting set S_1 is such that no function g of fewer than $m/7$ pieces is such that band(g, ζ) picks out S_1. A contradiction and hence the claim. □

We leave it to the reader to show that the following classes also possess efficient strongly Occam approximation algorithms:

(1) The polynomials with complexity measure the highest degree, (can construct a strongly Occam approximation via linear programming).

(2) The trigonometric interpolants with complexity measure the highest frequency, (can construct a strongly Occam approximation via linear programming).

(3) The piecewise constant functions with complexity measure the number of pieces. (can construct a greedy approximation algorithm that is strongly Occam.)

Claim 5 shows that every strongly Occam approximation is an Occam approximation

as well, confirming that the strong notion is indeed stronger. In order to prove the claim, we need the following definition and lemma.

Let F be a class of functions from a set X to a set Y. We say F *shatters* $S \subseteq X$ if there exist two functions f, $g \in F$ such that (1) For all $x \in S$, $f(x) \neq g(x)$. (2) For all $S_1 \subseteq S$, there exists $h \in F$ such that h agrees with f on S_1 and with g on $S - S_1$. i.e., for all $x \in S_1$, $h(x) = f(x)$, and for all $x \in S - S_1$, $h(x) = g(x)$.

Lemma 1: Natarajan (1991), Haussler and Long (1990). Let X and Y be two finite sets and let F be a set of total functions from X to Y. If d is the cardinality of the largest set shattered by F, then,
$2^d \leq |F| \leq |X|^d |Y|^{2d}$.

Claim 5: If Q is a strongly Occam approximation for a class F, then it is also an Occam approximation with respect to the L_∞ metric.

Proof: Let Q be a strongly Occam approximation for a class G of envelope 1. Fix $m, t \in \mathbf{N}$,$\epsilon > 0$ and $\zeta > 0$. Let $\hat{G}$ be the class of all functions identified in the output of Q when its input ranges over all collections S of m examples such that $l_{\min}(S, \epsilon, L_\infty) \leq t$. Let $X = \{x_1, x_2 ..., x_m\}$, be a set of m points and C the minimum ζ-cover of $\hat{G}$ on X for L_∞. For each function g in C, construct the function

$$\hat{g}(x) = \lfloor g(x)/\epsilon \rfloor ,$$

and let $\hat{C}$ be the class of such functions. Since C is a minimum ζ cover, all the functions in $\hat{C}$ are distinct, and there is a one to one correspondence between C and $\hat{C}$. Let $n = \lceil 1/\zeta \rceil$. Now $\hat{C}$ is a class of functions from X to $Y = \{-n, -..., 0, 1, 2, ..., n\}$. By Lemma 1, there exists $X_1 \subseteq X$ of d points that is shattered by $\hat{C}$, where

$$d \geq \frac{\log |\hat{C}|}{2\log(3m/\zeta)}.$$

For each point $x \in X_1$, let $A(x)$ be the subset of $\hat{C}$ that agree on x, and $D(x)$ be the subset of $\hat{C}$ that disagree. Let $A_1(x)$ be the functions in C that correspond to $A(x)$ and similarly $D_1(x)$ for $D(x)$. It is clear that we can find y such that for all $g \in A_1(x)$, $|y - g(x)| \leq \zeta$ and for all $g \in D_1(x)$, $|y - g(x)| > \zeta$. It follows that the set $\{(x,y) | x \in X_1\}$ is shattered by band(C, ζ), implying that,

$$\log(N(\hat{G}, \zeta, m, L_\infty)) \leq 2\log(3m/\zeta)\mathbf{D}_{\mathrm{VC}}(band(C, \zeta)) \leq 2\log(3m/\zeta)\mathbf{D}_{\mathrm{VC}}(band(\hat{G}, \zeta)) .$$

It follows that Q is an Occam approximation as well. □

Theorem 4: Let F be the target class and G the hypothesis class with complexity measure l, where F and G are of envelope 1 and $b+1$ respectively. Let Q be a strongly Occam approximation for G, with characterstic (q,r). Then, there exists a learning algorithm for F with expected sample complexity polynomial in $1/\epsilon$, $1/\delta$, b and $l_{\min}(f,\epsilon/4,P_u,L_1)$, where P_u is the uniform distribution on $[0,1]$.

Proof: Let f be the target function, and let (q,r) be the characteristic of the Occam approximation Q. We claim that Algorithm A_3 is a learning algorithm for F with respect to the L_1 metric.

Algorithm A_3
input ϵ, δ, b;
begin
  $t = 2$;
  $\eta = \dfrac{\gamma(\epsilon/4)\epsilon}{2(b+2)}$;
  repeat forever
    let m be the least integer satisfying
    $m \geq \dfrac{16}{\eta} q(t, 1/(b+\epsilon/4))r(m)\log(m)$;
    make m calls of EXAMPLE to get collection S;
    let $g = Q(S, b+\epsilon/4)$.
    let $m_1 = \dfrac{16t^2}{\eta^2\delta}$;
    make m_1 calls of EXAMPLE to get collection S_1;
    if no more than a fraction $(3/4)\eta$ of S_1
      is outside band$(g,b+\epsilon/4)$**then** output g and halt;
    else $t = t+1$;
  end
end

For a particular function g, let $\mu(g,\beta)$ be the probability that a call of EXAMPLE will result in an example outside band$(g,b+\beta)$. We now estimate $\mu(g,\epsilon/4)$ when g is such that $L_1(f,g,P) > \epsilon$. For such g, since F is of envelope 1 and G is of envelope $b+1$, $|f-g| \leq b+2$, and

$$\int_{|f-g|>\epsilon/2} dP > \epsilon/(2(b+2)).$$

It follows that

$$\mu(g,\epsilon/4) = \mathbf{Pr}\left\{f(x)+\nu \notin \text{band}(g,b+\epsilon/4)\right\}$$

$$> \mathbf{Pr}\left\{f(x) - g(x) > \epsilon/2 \text{ and } v \in [b, b-\epsilon/4]\right\}$$

$$+ \mathbf{Pr}\left\{g(x) - f(x) > \epsilon/2 \text{ and } v \in [-b, -b+\epsilon/4]\right\}.$$

$$\geq \min\left(\mathbf{Pr}\left\{v \in [b-\epsilon/4, b]\right\}, \mathbf{Pr}\left\{v \in [-b, -b+\epsilon/4]\right\}\right) \times \mathbf{Pr}\left\{|f-g| > \epsilon/2\right\}$$

$$\geq \gamma(\epsilon/4)\epsilon/(2(b+2)) \geq \eta .$$

Let $\mu_1(g, \beta)$ denote the fraction of S_1 that is outside band$(g, b+\beta)$. By Chebyshev's inequality,

$$\mathbf{Pr}\left\{|\mu_1(g, \epsilon/4) - \mu(g, \epsilon/4)| > 1/4\eta\right\} \leq \frac{16}{\eta^2 m_1} .$$

Since $m_1 = 16t^2/(\eta^2\delta)$, and the algorithm halts only when $\mu_1(g, \epsilon/4) \leq (3/4)\eta$, when the algorithm halts,

$$\mathbf{Pr}\left\{\mu(g, \epsilon/4) > \eta\right\} \leq \frac{\delta}{t^2} .$$

Summing over all $t \geq 2$, we get that the probability that the algorithm halts with $\mu(g, \epsilon/4) > \eta$ is at most δ. It follows that the probability that the algorithm halts with $L_1(g,f,P) > \epsilon$ is at most δ. Let P_u be the uniform distribution on $[0,1]$. We now show that when $t \geq t_0 = l_{\min}(f, \epsilon/4, P_u, L_\infty)$ the algorithm halts with probability at least 1/4.

Let $t \geq t_0$ at a particular iteration, and let $\hat{G}$ be the class of all functions that Q could output during that iteration. Since $v \in [-b, +b]$, it is clear that $l_{\min}(S, b+\epsilon/4, L_\infty) \leq t_0 \leq t$. Since Q is strongly Occam, $\mathbf{D}_{VC}(\text{band}(\hat{G}, b+\epsilon/4)) \leq q(t, 1/(b+\epsilon/4))r(m)$. By Theorem 4.3 of Natarajan (1991), for $m \geq 8/\eta$, the probability that m random samples will all fall in band$(g, b+\epsilon/4)$ for any $g \in \hat{G}$ such that $\mu(g, \epsilon/4) > \eta/2$, is at most

$$2\sum_{i=0}^{d} \binom{2m}{i} 2^{-\frac{\eta m}{4}} .$$

Hence, if m is chosen so that the above probability is less than 1/2, then with

probability at least 1/2, $Q(S, b+\epsilon/4)$ will return a function g such that $\mu(g, \epsilon/4) \leq \eta/2$. Indeed

$$m \geq \frac{16}{\eta} q(t, 1/(b+\epsilon/4)) r(m) \log(m).$$

suffices. Since Q is strongly Occam, $r(m)\log(m)$ is $o(m)$, and such m exists. We now estimate the probability the algorithm halts when Q returns a function g satisfying $\mu(g, \epsilon/4) \leq \eta/2$. Once again by Chebyshev's inequality,

$$\mathbf{Pr}\left\{ \mid \mu_1(g, \epsilon/4) - \mu(g, \epsilon/4) \mid > \eta/4 \right\} \leq \frac{16}{\eta^2 m_1}.$$

Given that $\mu(g, \epsilon/4) \leq \eta/2$, and that $m_1 = 16t^2/(\eta^2\delta)$, we have

$$\mathbf{Pr}\left\{ \mu_1(g, \epsilon/4) > 3/4\eta \right\} \leq \frac{\delta}{t^2} \leq 1/2.$$

Hence, we have $\mathbf{Pr}\{\mu_1(g, \epsilon/4) \leq 3/4\eta\} \geq 1/2$. That is if the function g returned by Q is such that $\mu(g, \epsilon/4) \leq \eta/2$ then, A_1 will halt with probability at least 1/2. We have therefore shown that when when $t \geq t_0$, the algorithm halts with probability at least 1/4. Hence, the probability that the algorithm does not halt within some $t > t_0$ iterations is at most $(3/4)^{t-t_0}$, which goes to zero with increasing t. □

5.3 Application to Filtering

An important problem in signal processing is that of filtering random noise from a discretely sampled signal, Oppenheim and Schafer (1974). The classical approach to this problem involves manipulating the spectrum of the noisy signal to eliminate noise. This works well when the noise-free signal has compact spectral support, but is not effective otherwise. However, the noise-free signal may have compact support in some other representation, where the filtering may be carried out effectively.

When we examine algorithm A_2, we see that the sampling rate m varies roughly as $q(\,)r(m)/\epsilon^2$, or $\epsilon^2 \sim q(\,)r(m)/m$. In a sense, $q(\,)r(m)$ is the "support" of the noise-free target function in the hypothesis class G. While spectral filters choose G to be the trigonometric interpolants, we are free to choose any representation, aiming to minimize $q(\,)r(m)$. Furthermore, the Occam approximation Q need not manipulate its input samples in a linear fashion, as is the case with spectral filters. In this sense, our results offer the first general technique for the construction of non-linear filters.

In the practical situation, our results can be interpreted thus: pass the samples of the noisy signal through a data compression algorithm, allowing the algorithm an approximation error equal to the noise strength. The decompressed samples compose the filtered signal, and are closer to the noise-free signal than the noisy-signal. Implementations of this are pursued in Natarajan (1993a), (1994).

6. Conclusion

We showed that the principle of Occam's Razor is useful in the context of probably approximate learning functions on the reals, even in the presence of arbitrarily large additive random noise. The latter has important consequences in signal processing, in that it offers the first general technique for the design and construction of non-linear filters.

7. Acknowledgements

Thanks to A. Lempel and J. Ruppert for discussions and comments.

8. References

Angluin, D., and Laird, P., (1988). Learning from noisy examples, Machine Learning, Vol.2, No. 4, pp. 319-342.

Anthony, M. & Biggs, N. (1992). Computational Learning Theory: An Introduction, Cambridge University Press.

Anthony, M. & Bartlett, P., (1994). Function learning from interpolation, University of London, Neurocolt Tech. Rep. NC-TR-94-013.

Bartlett, P.L., Long, P. M., and Williamson, R.C., (1994) Proc. Seventh ACM Symposium on Comp. Learning Theory, pp.299-310.

Ar, S., Lipton, R., Rubenfeld, R., and Sudan M., (1992). Reconstructing algebraic functions from mixed data, Proc. 33rd IEEE Foundations of Comp. Science, pp.503-511.

Ben-Or, M. and Tiwari, P., (1988). A deterministic algorithm for sparse multivariate polynomial interpolation, Proc. 20th ACM Symp. on Theory of Computing, pp.394-398.

Berlekamp, E., and Welch, L., (1970). Error correction of algebraic block codes, U.S. Patent No. 4,633,470.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M., (1991). Learnability and the Vapnik-Chervonenkis dimension, JACM, Vol.36, No.4, pp.929-965.

Feller, W., (1957). Intro. to Prob. Theory and its Applications, Vol II, John Wiley, New York.

Golub, G.H., and Van Loan, C.F., (1983). *Matrix Computations*, Johns Hopkins Press, Baltimore, MD.

Grigoriev, D., Karpinski, M., and Singer, M.F., (1990). Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields, SIAM J. on Computing, pp.1059-1063.
Haussler, D., (1989). Generalizing the PAC model for neural net and other learning applications, Proc. 30th IEEE Foundations of Computer Science, pp. 40-45.
Haussler, D., and Long, P., (1990). A generalization of Sauer's Lemma, Tech. Report UCSC-CRL-90-15. University of California, Santa Cruz.
Imai, H., and Iri, M., (1986). An optimal algorithm for approximating a piecewise linear function. J. of Information Processing, Vol. 9, No. 3, pp. 159-162.
Jazwinski, A.H., (1970). Stochastic Processes and Filtering Theory, Academic Press, New York.
Kearns M., and Li, M., (1993). Learning in the presence of malicious errors, SIAM J. on Computing, 22:807-837.
Kearns M., and Schapire, R. E., (1990). Efficient distribution-free learning of probabilistic concepts, Proc. IEEE Foundations of Computer Science, pp. 382-391.
Krishnan, V., (1984). Non-Linear Filtering and Smoothing, John Wiley, New York.
Natarajan, B.K., (1989). On learning sets and functions, Machine Learning, Vol.4, No.1, pp.67-97.
Natarajan, B.K., (1991). Machine Learning: A Theoretical Approach, Morgan Kaufmann, San Mateo, CA.
Natarajan, B.K., (1993a). Filtering random noise via data compression, Proc. IEEE Data Compression Conference, pp.60-69..
Natarajan, B.K., (1993b). Occam's Razor for functions, Proc. Sixth ACM Symposium on Comp. Learning Theory, pp.370-376.
Natarajan, B.K., (1994). Sharper bounds on Occam Filters and application to digital video, Proc. IEEE Data Compression Conference.
Oppenheim, A.V., and Schafer, R., (1974). Digital Signal Processing, Prentice Hall, Englewood Cliffs, N.J.
Papoulis, A. (1965). Probability, Random Variables and Stochastic Processes. McGraw Hill, New York.
Pollard, D., (1984). Convergence of Stochastic Processes, Springer Verlag, New York.
Sloan, R., (1988). Types of noise in data for concept learning, Proc. 1988 Workshop on Computational Learning Theory, pp.91-96.
Suri, S., (1988). On some link distance problems in a simple polygon. IEEE Trans. on Robotics and Automation, Vol.6, No.1, pp.108-113.