数理解析研究所講究録
第 731 巻 1990 年 49-60

Occam Algorithms for Learning from Noisy Examples

Yasubumi Sakakibara
榊原 康文

International Institute for Advanced Study of Social Information Science (IIAS-SIS)
FUJITSU LIMITED
140, Miyamoto, Numazu, Shizuoka 410-03, Japan
E-mail : yasu%iias.fujitsu.co.jp@uunet.uu.net

Abstract

In the distribution-independent model of concept learning from examples introduced by Valiant [Val84], it has been shown that the existence of an *Occam algorithm* for a class of concepts implies the computationally feasible (polynomial) learnability of that class [BEHW87a, BEHW87b]. An Occam algorithm is a polynomial-time algorithm that produces, for any sequence of examples, a nearly minimum hypothesis consistent with the examples. These works, however, depend strongly on the assumption of perfect, noise-less examples. This assumption is generally unrealistic and in many situations of the real world there is always some chance that a noisy example is given to the learning algorithm. In this paper we present a practical extension to Occam algorithms in the Valiant learnability model: Occam algorithms that can tolerate the *classification noise*, a noise process introduced in [AL88] (classifying the example is subject to independent random mistakes with some small probability), and it is shown that the *noise-tolerant Occam algorithm* is a powerful algorithmic tool to establish computationally feasible learning that copes with the classification noise; the existence of a noise-tolerant Occam algorithm for a class of concepts is a sufficient condition for the polynomial learnability of that class in the presence of noise.

1 Introduction

Inductive learning (concept learning) from examples is viewed as a heuristic search through a space of hypotheses. Inductive learning algorithms are often faced with a common problem, which is that of how to search a large space efficiently to find a consistent hypothesis that describes the given examples. *Occam's razor* is an old scientific heuristic, with a sound basis in human behavior [Sal85, BP89]. This heuristic claims that "entities should not be multiplied unnecessarily", which usually means that when offered a choice among explanations, all other things being equal, the simplest one is to be preferred. This principle can be interpreted in the area of machine learning to mean a "inductive bias" [Hau86, Utg86] that among hypotheses consistent with a given sample of examples, the learning algorithm should choose the simplest hypothesis. Usually "simplest" means "minimum size" of the representation for the hypothesis.

However this is not always practical, because for many domains it is a very hard problem (NP-hard) to find a consistent hypothesis of minimum size. In the distribution-independent model of concept learning introduced by Valiant [Val84], in order to obtain computationally feasible (polynomial) learning, so-called *Occam algorithms* [BEHW87b, BEHW87a] have been proposed that weaken this criterion of minimality. An Occam algorithm is a polynomial-time algorithm that produces its consistent hypothesis of size polynomially larger than minimum and sublinearly on the size of the given sample. [BEHW87b] has shown that the existence of an Occam algorithm for a class of concepts implies polynomial learnability for that class. Thus Occam algorithms guarantee that it suffices to produce simpl*er* hypotheses rather than simpl*est* ones for feasible learning. Several interesting concept classes have been shown to be polynomially learnable by using Occam algorithms while it is NP-hard to find a consistent hypothesis of minimum size in those classes.

Many works making progress in the Valiant learnability model including Occam algorithms depend strongly on the assumption of perfect, noise-less examples. However, this assumption is generally unrealistic and in many situations of the real world, we are not always so fortune, our observations will often be afflicted by noise and hence there is always some chance that a noisy example is given to the learning algorithm. Few works have suggested any way to make their learning algorithms noise tolerant and two formal models of noise have been studied so far in the Valiant learnability model [AL88, KL88].

The main contributions of this paper are a practical extension to Occam algorithms in the Valiant learnability model: Occam algorithms that can tolerate the classification noise,

a noise process introduced in [AL88] (classifying the example is subject to independent random mistakes with some small probability), and evidence that the noise-tolerant Occam algorithm is in fact an algorithmic tool to establish computationally feasible learning that copes with the classification noise; the existence of a noise-tolerant Occam algorithm for a class of concepts is a sufficient condition for the polynomial learnability of that class in the presence of noise. Further we demonstrate an example of learning geometric concepts to exhibit how the noise-tolerant Occam algorithm can be used to establish the polynomial learnability of many concept classes in the presence of classification noise.

2 Polynomial Learnability

We first give a brief outline of Valiant's learnability model [Val84] and the notion of polynomial learnability [BEHW87a, BEHW87b]. A *concept* is defined by a subset of some instance space (domain) X. A *sample* of a concept is a sequence of examples, each of which is an instance of the concept, called *positive example*, labeled $+$ or a non-instance of the concept, called *negative example*, labeled $-$. Samples are assumed to be created from independently, random examples, chosen according to some fixed but unknown probability distribution P on X. The *size* of a sample is the number of examples in it. Let H be a class of concepts defined on X. We define a *learning algorithm* for H as an algorithm that takes as input a sample of a target concept in H and produces as output a *hypothesis* that is itself a concept in H. A hypothesis is *consistent* with the given sample if it includes all positive examples and no negative examples in the sample. A consistent hypothesis may still disagree with the target concept by failing to include unobserved instances of the target concept or including unobserved non-instances of the target concept. The *error* of a hypothesis is the probability that the hypothesis will disagree with a random example of the target concept selected according to the distribution P. A successful learning algorithm is one that with high probability with respect to P finds a hypothesis whose error is small.

Two performance measures are applied to learning algorithms in this setting.

1. The *convergence rate* of the learning algorithm is measured in terms of the sample size that is required for the algorithm to produce, with high probability, a hypothesis that has a small error.

2. The *computational efficiency* of the learning algorithm is measured in terms of the computation time required to produce a hypothesis from a sample of a given size.

With respect to these two measures the notion of *polynomial learnability* is going to be defined. Before giving that, we must assume some representation for the hypotheses produced by a learning algorithm and a *complexity measure* for the concepts with respect to the representation because the sample size needed for a successful learning algorithm usually depends on the complexity of the target concept. Let Γ be a (not necessarily finite) alphabet used to describe representations for hypotheses. We fix some *encoding* (function) from Γ^* to 2^X so that a set of strings $R \subseteq \Gamma^*$ represent a class of concepts and let the *size* of a concept be the number of characters needed to represent it in the encoding (that is, the length of the representation).

> A class of concepts H is *polynomially learnable* (with respect to *a fixed encoding*) if there exists a learning algorithm for H and a function $m(\epsilon, \delta, n)$, polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, and n, such that
>
> 1. for any target concept $C \in H$ of size at most n and any distribution P on X, given a sample of size $m(\epsilon, \delta, n)$ of C, the algorithm produces a hypothesis in H with error at most ϵ with probability at least $1 - \delta$, and
> 2. the algorithm produces its hypothesis in time polynomial in the length of the given sample.

The algorithm that has the property 2 is called a *polynomial hypothesis finder* for H. Notice that the sample size is not only polynomially bounded in the inverses of ϵ and δ but also allowed to grow polynomially in the size of the target concept.

3 Occam's Razor and Occam Algorithms

Length-based Occam algorithm First we consider the case where the alphabet Γ is finite. This is typically the case when concepts are defined over discrete domains (e.g., automata, Boolean formulas, etc.). Assume some fixed encoding to represent a hypothesis in Γ.

Given a sample of a target concept in H, the fundamental strategy that a learning algorithm takes is producing its hypothesis consistent with the sample. When a hypothesis class H is infinite, there may be infinitely many consistent hypothesis and a polynomial learning algorithm for H cannot in general afford to choose its consistent hypothesis arbitrarily, it needs some "inductive bias" [Hau86, Utg86]. Occam's Razor would suggest that a learning algorithm should choose its consistent hypothesis among those that have

"minimum" size. In fact, given a sample of size $\frac{n}{\epsilon}\ln(\frac{|\Gamma|}{\delta})$ for a target concept $C \in H$ of size at most n, a learning algorithm for H that produces its hypothesis of minimum size consistent with the sample could produce a hypothesis with error at most ϵ with probability at least $1-\delta$. This satisfies the property 1 of polynomial learnability for H.

However, this is not always practical. For example, finding a minimum state deterministic finite automaton consistent with positive examples and negative examples of a regular language is a NP-hard problem. Thus for many domains finding a hypothesis of minimum size cannot be a polynomial hypothesis finder. In order to obtain polynomial algorithms, Occam algorithms are proposed that weaken this criterion of minimality [BEHW87b].

> An *(l-based) Occam algorithm* for H with a polynomial $p(x)$ and a constant α, $0 \leq \alpha < 1$, (with respect to a fixed encoding) is a learning algorithm that
>
> 1. produces a consistent hypothesis of size at most $p(n)m^{\alpha}$ when given a sample of size m of any target concept in H of size at most n, and
> 2. runs in time polynomial in the length of the sample.

Thus an Occam algorithm allows the size of the hypothesis produced by a learning algorithm to be polynomially larger than minimum and sublinearly on the size of the given sample. By taking $m \geq \max[\frac{2}{\epsilon}\ln(\frac{1}{\delta}), (\frac{2p(n)\ln(|\Gamma|)}{\epsilon})^{\frac{1}{1-\alpha}}]$, the existence of an Occam algorithm for H implies polynomial learnability for H [BEHW87b].

Dimension-based Occam algorithm When Γ is infinite, the existence of a l-based Occam algorithm is not sufficient to guarantee polynomial learnability. The proof of sufficiency and the sample size needed for a l-based Occam algorithm depends critically on the finiteness of Γ. Consequently the proof fails when Γ is infinite. Such representations typically occur when concepts are defined over continuous domains (for example, several geometric concepts such as axis-parallel rectangles in Euclidean space). [BEHW87a] defines a more general type of Occam algorithm, which uses a combinatorial parameter called the *Vapnik-Chervonenkis dimension (VC dimension)* to measure the complexity of the class of hypotheses produced by the learning algorithm. The larger the VC dimension of the class of hypotheses, the greater the expressibility, and hence the complexity, of that hypothesis class. Rather than measuring simplicity by the length of the representations produced by the Occam algorithm, this definition uses the notion of VC dimension to measure the simplicity of the class of hypotheses produced by the Occam algorithm.

The following definition of an Occam algorithm [BEHW87a] allows the learning algorithm for H to produce hypotheses from a class of VC dimension $p(n)m^\alpha$ in H for a target concept of size at most n in H, where m is the size of the given sample, $p(x)$ is a polynomial and $0 \leq \alpha < 1$.

An *(d-based) Occam algorithm* for H is a learning algorithm that

1. produces a consistent hypothesis such that the class of hypotheses produced has the VC dimension at most $p(n)m^\alpha$ when given a sample of size m of any target concept in H of size at most n, and
2. runs in time polynomial in the length of the sample.

By taking $m \geq \max[\frac{4}{\epsilon}\log(\frac{2}{\delta}), (\frac{8p(n)}{\epsilon}\log(\frac{13}{\epsilon}))^{\frac{1}{1-\alpha}}]$, the existence of an Occam algorithm for H implies polynomial learnability for H [BEHW87a]. The proof of sufficiency of a d-based Occam algorithm relies on the fact that a learning algorithm is successful if and only if the VC dimension of the class of hypotheses produced by the learning algorithm is finite.

4 Noise-Tolerant Occam Algorithms

Classification noise model Many works making progress in the Valiant learnability model or on machine learning from examples depend strongly on the assumption of perfect, noise-less examples. However, this assumption is generally unrealistic and in many situations of the real world, we are not always so fortune, our observations will often be afflicted by noise and hence there is always some chance that a noisy example is given to the learning algorithm. Few works have suggested any way to make their learning algorithms noise tolerant and two formal models of noise have been studied so far in the Valiant learnability model. One is the *malicious error model* initiated in [Val85] and investigated in [KL88]: independently for each example, the example is replaced, with some small probability, by an arbitrary example classified perhaps incorrectly. The goal of this model is to capture the worst possible case of noise process by the adversary. The other is the *classification noise model* introduced in [AL88]: independently for each example, the label of the example is reversed with some small probability. The goal of this model is to study the question of how to compensate for randomly introduced errors, or "noise, in classifying the example data. In this paper, we consider the classification noise model to study the effect on the polynomial learnability.

In the classification noise model, an example is selected from the instance space X according to the relevant distribution P without error, but the process of determining and reporting whether the example is positive or negative is subject to independent random mistakes with some unknown probability η. The precise definition of it is that independently for each example, after it has been selected and classified but before presentation to the learning algorithm, the label of the example is reversed with probability η. It is assumed that the rate of noise η is less than $\frac{1}{2}$.

In [AL88], the following argument is discussed: in the presence of noise, we should assume that there is some information about the noise rate η available to the learning algorithm, namely an upper bound η_b such that $\eta \leq \eta_b < \frac{1}{2}$, and we should also permit the size of samples to depend on the upper bound η_b and just as the sample size for polynomial learnability is permitted in the absence of noise to be polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$, we should permit the polynomial to have $\frac{1}{1-2\eta_b}$ as one of its arguments. Thus the statement "*there exists ... a function* $m(\epsilon, \delta, n)$, *polynomial in* $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, *and* n" in the definition of polynomial learnability will be replaced with "*there exists ... a function* $m(\epsilon, \delta, n, \eta_b)$, *polynomial in* $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, *and* $\frac{1}{1-2\eta_b}$" in the presence of noise.

Length-based noise-tolerant Occam algorithm When the sample contains noise, the fundamental strategy of finding a hypothesis consistent with the given sample may fail because there is no guarantee that such consistent hypotheses will exist. For a finite concept class, [AL88] has proposed the simple strategy of finding a hypothesis that minimizes the number of disagreements with the given sample; a learning algorithm for a finite class H of N concepts that produces its hypothesis minimizing the number of disagreements could produce a hypothesis with error at most ϵ with probability at least $1 - \delta$ when given a sample of size $\frac{2}{\epsilon^2(1-2\eta_b)^2} \ln(\frac{2N}{\delta})$. To establish polynomial learnability for an infinite concept class, however, this strategy could not work because any hypothesis that minimizes the number of disagreements may have the exponentially larger size. The following Occam algorithm solves this problem by taking the strategy of finding a hypothesis whose rate of disagreements is less than some fixed value calculated from ϵ and η_b instead of finding one of minimum rate so that with high probability hypotheses of at most polynomially larger size can be found among the hypotheses that have the rate of disagreements less than the value.

A *(l-based) noise-tolerant Occam algorithm* for H (with respect to a fixed encoding) is a learning algorithm that

1. produces a hypothesis of size at most $p(n)m^\alpha$ such that

$$\frac{\text{the number of disagreements}}{m} \leq \eta_b + \frac{\epsilon(1-2\eta_b)}{2},$$

when given a sample of size m of any target concept in H of size at ~~most~~ n, and

2. runs in time polynomial in the length of the sample.

Thus a noise-tolerant Occam algorithm is identical to an (usual) Occam algorithm, except that rather than finding a consistent hypothesis, the algorithm finds a hypothesis consistent with at least $(1-(\eta_b+\frac{\epsilon(1-2\eta_b)}{2}))m$ of the examples. The similar notions can be found in [BP89, KL88].

Theorem 1 *Given independent examples of any concept in H of size at most n afflicted by classification noise of rate η, a (l-based) noise-tolerant Occam algorithm produces a hypothesis with error at most ϵ with probability at least $1-\delta$ using sample size polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, and $\frac{1}{1-2\eta_b}$. The sample size required is*

$$m \geq \max\left[\frac{4}{\epsilon^2(1-2\eta_b)^2}\ln\left(\frac{2}{\delta}\right), \left(\frac{p(n)\ln(|\Gamma|)}{\epsilon^2(1-2\eta_b)^2}\right)^{\frac{1}{1-\alpha}}\right].$$

Sketch of proof. Let $s = \frac{\epsilon(1-2\eta_b)}{2}$. The first lower bound on m implies that the probability that the target concept has more than $(\eta_b+s)m$ disagreements with the sample is less than $\frac{\delta}{2}$ by the Hoeffding's inequality lemma in [AL88]. The second lower bound on m implies that $|\Gamma|^{p(n)m^\alpha} \leq e^{-2s^2(-m/2)}$. Since the number of hypotheses of size at most $p(n)m^\alpha$ is $|\Gamma|^{p(n)m^\alpha}$ and the probability that a hypothesis with error greater than ϵ has at most $(\eta_b+s)m$ disagreements is e^{-2s^2m} by the Hoeffding's inequality lemma, the probability of producing a hypothesis with error greater than ϵ is less than $|\Gamma|^{p(n)m^\alpha}e^{-2s^2m} \leq e^{-2s^2m/2}$, which is less than $\frac{\delta}{2}$ by the first lower bound on m. □

Dimension-based noise-tolerant Occam algorithm

A *(d-based) noise-tolerant Occam algorithm* for H is a learning algorithm that

1. produces a hypothesis such that

$$\frac{\text{the number of disagreements}}{m} \leq \eta_b + \frac{\epsilon(1-2\eta_b)}{4},$$

and the class of hypotheses produced has the VC dimension at most $p(n)m^\alpha$ when given a sample of size m of any target concept in H of size at most n, and

2. runs in time polynomial in the length of the sample.

Theorem 2 *Suppose that $\eta \leq \eta_b \leq \frac{\epsilon}{4}$. Given independent examples of any concept in H of size at most n afflicted by classification noise of rate η, a (d-based) noise-tolerant Occam algorithm produces a hypothesis with error at most ϵ with probability at least $1-\delta$ using sample size polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, and $\frac{1}{1-2\eta_b}$. The sample size required is*

$$m \geq \max\left[\frac{8}{\epsilon^2(1-2\eta_b)^2}\ln\left(\frac{2}{\delta}\right), \frac{128}{\epsilon^3}\ln\left(\frac{16}{\delta}\right), \left(\frac{256p(n)}{\epsilon^3}\ln\left(\frac{256}{\epsilon^3}\right)\right)^{\frac{1}{1-\alpha}}\right].$$

Sketch of proof. Let $s = \frac{\epsilon(1-2\eta_b)}{4}$. We use the following fact from [BEHW87a]: given a sample of size m of a target concept in H of VC dimension d, the probability that a hypothesis consistent with at least $(1-\gamma)\epsilon m$ of the examples has error greater than ϵ is at most $8(\frac{2em}{d})^d e^{-\gamma^2\epsilon m/4}$ and when $m = \max[\frac{8}{\gamma^2\epsilon}\ln(\frac{8}{\delta}), \frac{16d}{\gamma^2\epsilon}\ln(\frac{16}{\gamma^2\epsilon})]$, it is less than δ.

The first lower bound on m implies that the probability that the target concept has more than $(\eta_b + s)m$ disagreements with the sample is less than $\frac{\delta}{2}$. Therefore with probability at least $1-\frac{\delta}{2}$, the number of examples on which the target concept and a hypothesis that has at most $(\eta_b + s)m$ disagreements disagree is less than $2(\eta_b + s)m$. By letting $\gamma = \frac{1}{2} + (1-\frac{2}{\epsilon})\eta_b$, the probability of producing a hypothesis with error greater than ϵ is less than $8(\frac{2em}{p(n)m^\alpha})^{p(n)m^\alpha}e^{-\gamma^2\epsilon m/4}$ and by the second and third lower bound on m and $\eta_b \leq \frac{\epsilon}{4}$, it is less than $\frac{\delta}{2}$. □

Application: learning finite unions of geometric concepts We now demonstrate an example of learning geometric concepts to exhibit how the noise-tolerant Occam algorithm can be used to establish the polynomial learnability of many concept classes in the presence of classification noise. The target concept class H_* is the set of all finite unions

of axis-parallel rectangles in Euclidean k-dimensional space E^k, that is, $H_* = \bigcup_{s\geq 1} H_s$, where H_s is the class of s-fold unions of axis-parallel rectangles. Each concept C of H_s can be represented as the form of $C = C_1 \cup C_2 \cup \cdots \cup C_s$ (each tuple C_i $(1 \leq i \leq s)$ is an axis-parallel rectangle). Let H denote the class of axis-parallel rectangles in E^k. Let the *size* of a concept C in H_* be the smallest s such that $C = C_1 \cup C_2 \cup \cdots \cup C_s$, where $C_i \in H$ $(1 \leq i \leq s)$. H has the finite VC dimension $2k$, but H_* has the infinite one. H_s has the finite VC dimension $4ks\log(3s)$ [BEHW87a]. Thus the problem is formulated as follows: Given independent examples of any union of at most s axis-parallel rectangles in H afflicted by classification noise of rate η, find a hypothesis in H_* with error at most ϵ with probability at least $1-\delta$ using a sample of size polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, s, and $\frac{1}{1-2\eta_b}$ in time polynomial in the length of the sample.

While there exists a polynomial hypothesis finder for the class H of axis-parallel rectangles, it is NP-hard to find a union of s or fewer axis-parallel rectangles that contains all positive examples and no negative example in the given sample of any union of at most s axis-parallel rectangles, because this problem can be formulated as a "set cover problem". Hence given a sample of any union of at most s axis-parallel rectangles afflicted by classification noise, it is also NP-hard to find a union of s or fewer axis-parallel rectangles such that the rate of disagreements with the sample is less than $\eta_b + \frac{\epsilon(1-2\eta_b)}{4}$. The key techniques used in the following are a simple approximation algorithm for the set cover problem and a (d-based) noise-tolerant Occam algorithm.

First we assume that the classification noise model is restricted to that the label of the negative example is only reversed with probability η and no errors will be made in reporting the positive example. Then the problem becomes finding a union of axis-parallel rectangles that contains more than $(1-\eta_b-\frac{\epsilon(1-2\eta_b)}{4})m - m_{neg}$ positive examples and no negative example, where m is the size of the given sample and m_{neg} is the number of the negative examples. This problem can be formulated as a "partial cover problem". By employing a greedy approximation algorithm of [KL88] for the partial cover problem, a (d-based) noise-tolerant Occam algorithm can be given to establish the polynomial learnability for H_* in the presence of classification noise.

Given a sample S of a target concept $C = C_1 \cup C_2 \cup \cdots \cup C_s$ of s-fold union of axis-parallel rectangles, Algorithm A takes the following procedure:

1. Let $\Pi_H(S)$ denotes the set of all ways the instances in S can be labeled with $+$ and $-$ so as to be consistent with at least one axis-parallel rectangle in H. Find the largest set in $\Pi_H(S)$ that contains only positive

examples;

2. Use the polynomial hypothesis finder for H to produce an axis-parallel rectangle that includes only these instances in S;
3. By deleting these instances from S and iterating this procedure, obtain a union of axis-parallel rectangles that contains more than $(1 - \eta_b - \frac{\epsilon(1-2\eta_b)}{4})m - m_{neg}$ positive examples and no negative example in S.

By [KL88], the algorithm A produces a hypothesis of a union of at most $s(c\log(m) + 3)$ axis-parallel rectangles (c is a constant). The VC dimension of the hypothesis class of $s(c\log m + 3)$-fold unions of axis-parallel rectangles is $O(s\log(m)(\log(s) + \log\log(m)))$ [BEHW87a]. Hence A is a (d-based) noise-tolerant Occam algorithm and thus by Theorem 2, H_* is polynomially learnable in the presence of restricted classification noise.

5 Concluding Remarks

We have introduced noise-tolerant Occam algorithms in the Valiant learnability model and shown that the existence of a noise-tolerant Occam algorithm for a class of concepts is a sufficient condition for the polynomial learnability of that class in the presence of noise. An application of noise-tolerant Occam algorithms has been demonstrated in the example of learning finite unions of axis-parallel rectangles in the presence of noise. The validity of the learning algorithm A in the example, however, depends on the assumption of "restricted" classification noise. Can we find a noise-tolerant Occam algorithm that learns finite unions of axis-parallel rectangles (or some other geometric concepts) in the presence of classification noise (not restricted one)?

[BP89] has proved that not only are Occam algorithms a sufficient condition for polynomial learnability, but they are in fact a necessary condition for many natural concept classes. It would be interesting to investigate whether this result can also be hold in the presence of noise; the existence of a noise-tolerant Occam algorithm is also a necessary condition for polynomial learnability in the presence of classification noise, which will turn out to prove the correctness of our definition of noise-tolerant Occam algorithms.

References

[AL88] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988.

[BEHW87a] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. Technical Report UCSC-CRL-87-20, Department of Computer and Information Sciences, University of California, Santa Cruz, 1987. To appear in *Journal of the ACM.*

[BEHW87b] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[BP89] R. Board and L. Pitt. On the necessity of Occam algorithms. Technical Report UIUCDCS-R-89-1544, Department of Computer Sciences, University of Illinois at Urbana-Champaign, 1989.

[Hau86] D. Haussler. Quantifying the inductive bias in concept learning. In *Proceedings of AAAI-86*, pages 485–489, 1986.

[KL88] M. Kearns and M. Li. Learning in the presence of malicious errors. In *Proceedings of 20th Annual ACM Symposium on Theory of Computing*, pages 267–279. ACM, 1988.

[Sal85] S. Salzberg. Heuristics for inductive learning. In *Proceedings of 9th International Joint Conference on Artificial Intelligence*, pages 603–609. Morgan Kaufmann, 1985.

[Utg86] P. E. Utgoff. *Machine Learning of Inductive Bias.* Kluwer Academic, 1986.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

[Val85] L. G. Valiant. Learning disjunctions of conjunctions. In *Proceedings of 9th International Joint Conference on Artificial Intelligence*, pages 560–566. Morgan Kaufmann, 1985.