Analyses of Instance-Based Learning Algorithms

Marc K. Albert
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717 U.S.A.
albert@ics.uci.edu

David W. Aha
The Turing Institute
36 North Hanover Street
Glasgow G1 2AD Scotland
aha@turing.ac.uk

Abstract

This paper presents PAC-learning analyses for instance-based learning algorithms for both symbolic and numeric-prediction tasks. The algorithms analyzed employ a variant of the k-nearest neighbor pattern classifier. The main results of these analyses are that the IB1 instance-based learning algorithm can learn, using a polynomial number of instances, a wide range of symbolic concepts and numeric functions. In addition, we show that a bound on the degree of difficulty of predicting symbolic values may be obtained by considering the size of the boundary of the target concept, and a bound on the degree of difficulty in predicting numeric values may be obtained by considering the maximum absolute value of the slope between instances in the instance space. Moreover, the number of training instances required by IB1 is polynomial in these parameters. The implications of these results for the practical application of instance-based learning algorithms are discussed.

1 Introduction and Style of Analysis

Several instance-based learning (IBL) algorithms based on variants of the k-nearest neighbor function (k-NN) have performed well in challenging learning tasks (Bradshaw, 1987; Stanfill & Waltz, 1986; Kibler & Aha, 1987; Salzberg, 1988; Aha & Kibler, 1989; Aha, 1989; Moore, 1990; Tan & Schlimmer, 1990; Waltz, 1990; Cost & Salzberg, 1990). However, these investigations contained only *empirical* evaluations. This paper generalizes our previous mathematical analyses, which restricted the values of k and the instance space's dimensionality (Kibler, Aha, & Albert, 1989; Aha, Kibler, & Albert, 1991).

Early mathematical analyses of k-NN investigated its asymptotic capabilities by comparing it against the Bayes decision rule, a strategy that is given all of the instances' joint probability densities and minimizes the probability of classification error. Cover and Hart (1967) showed that 1-NN's error rate was less than twice that of the Bayes rule and, thus, less than twice the error rate of *any* decision algorithm. Cover (1968) showed that, as k increases, k-NN's error rate quickly converges to the optimal Bayes rate. However, these analyses assume an infinite number of training instances, which are rarely available in practice. They did not determine how many instances k-NN requires to ensure that it will yield acceptably good classification decisions.

Therefore, our analysis employs Valiant's (1984) PAC-learning (probably, approximately correct) model for investigating learnability issues, which states that a class of concepts is *polynomially learnable* from examples if at most a polynomial number[1] of instances is required to generate, with a certain level of confidence, a relatively accurate approximation of the target concept. This definition of learnability is more relaxed than those used in earlier studies, which required that the algorithms have perfect accuracy with 100% confidence. It is formalized as follows.

Definition 1.1 *A class of concepts $\mathcal{C}$ is* polynomially learnable *iff there exists a polynomial p and a learning algorithm A such that, for any $0 < \epsilon, \delta < 1$, if at least $p(\frac{1}{\epsilon}, \frac{1}{\delta})$ positive and negative instances of a concept $C \in \mathcal{C}$ are drawn according to an arbitrary fixed distribution, then, with confidence at least $(1 - \delta)$, A will generate an approximation $C' \in \mathcal{C}$ whose probability of error is less than ϵ. Moreover, A will halt in time bounded by a polynomial in the number of instances.*

This definition states that the approximation generated may be imperfect, but must have its inaccuracy bounded by ϵ. Also, A is not required to always generate sufficiently accurate concept descriptions (i.e., within ϵ), but must do so only with probability at least $(1-\delta)$. The polynomial time bound is automatic for IB1 since the amount of time it takes to generate a prediction is polynomial in the number of training instances. Thus, we will not mention time bounds further in this article.

Valiant's model trades off accuracy for generality; it applies to any fixed distribution. Learnability is indicative only of the concept in $\mathcal{C}$ that is most difficult to learn and the probability distribution that results in the slowest possible learning rate. Thus, the number

[1] That is, polynomial with respect to the parameters for confidence and accuracy.

of instances required to PAC-learn concepts is a loose upper bound that can be tightened by restricting the set of allowable probability distributions. Therefore, many researchers adapt this model to their own needs by adding restrictions on probability distributions (Li & Vitanyi, 1989; Kearns, Li, Pitt, & Valiant, 1987). Many investigations also analyze a specific set of learning algorithms (Littlestone, 1988; Valiant, 1985; Haussler, 1987; Rivest, 1987; Angluin & Laird, 1988). Our analyses do both. Only the capabilities of a specific k-NN-based learning algorithm (i.e. IB1) are investigated, rather than the learnability of concept classes in general.[2] Also, predictor attributes are assumed to be numeric-valued and noise-free. The analyses in Section 3.1 address symbolic learning tasks and those in Section 3.2 address the learnability of *numeric functions*. Since few researchers have investigated learnability for numeric functions, this is a relatively novel contribution to computational learning theory.

2 Coverage Lemmas

This section presents two lemmas that are used to prove the theorems in Section 3, which concern the PAC-learning capabilities of IB1. These lemmas establish how large a set of training instances S is required to give, with high confidence, a "good coverage" of an instance space. That is, they ensure that, for all instances x in the space, except for those in regions of low probability, there is an instance (or set of k instances) in S that is sufficiently similar to x (i.e., their similarity is above a threshold). This information will be used in the learnability theorems to bound IB1's amount of prediction error.

The first lemma applies to the case when $k = 1$ (where only the *single* most similar training instance is used to predict target values) while the second lemma allows for $k \geq 1$. First, we need to define the notion of a $\langle \alpha, \gamma \rangle$-net.

Definition 2.1 *Let $X \subseteq \Re^d$ have an arbitrary but fixed probability distribution. Then $S \subseteq X$ is an $\langle \alpha, \gamma \rangle$-net for X if, for all x in X, except for a set with probability less than γ, there exists an $s \in S$ such that distance$(s, x) < \alpha$.*

The following proof shows that a sufficiently large random sample from a bounded subset of $\Re^d$ will probably be an $\langle \alpha, \gamma \rangle$-net.

[2]IB1 performed well in previous empirical evaluations (Aha, Kibler, & Albert, 1991). An attribute-value representation was used for instances, where all attributes are used for predictions (i.e., *predictors*) except the *target* attribute, whose value is to be predicted. The k-NN algorithm predicts a given instance's target attribute value from those of its k most similar previously processed instances. IB1 uses a majority vote to predict symbolic values and a similarity-weighted prediction for numeric values, where similarity is defined as the inverse of Euclidean distance (or some monotonicly increasing function of that).

Lemma 2.1 *Let X be a bounded subset of $\Re^d$. Then there exists a polynomial p such that for any $0 < \alpha, \gamma, \delta < 1$, a random sample S containing $N \geq p(\frac{1}{\alpha}, \frac{1}{\gamma}, \frac{1}{\delta})$ instances from X, drawn according to any fixed probability distribution, will form an $\langle \alpha, \gamma \rangle$-net with probability at least $(1 - \delta)$.*

Proof 2.1 Without loss of generality we may assume that X is $[0-1]^d$ (i.e., the unit hypercube in $\Re^d$). This lemma is proven by partitioning $[0-1]^d$ into m^d disjoint hyper-squares, each with diagonal of length less than α. Thus, pairs of instances lying in a single hyper-square are less than α apart.

The desired value for m is found using the Pythagorean Theorem, which shows that $m = \lceil \frac{\sqrt{d}}{\alpha} \rceil$. (We assume, without loss of generality, that $\lceil \sqrt{d}/\alpha \rceil > \sqrt{d}/\alpha$.) Let $S_{<\alpha}$ be the subset of hyper-squares with probability greater than or equal to γ/m^d. Let $S_{\geq\alpha}$ be the set of remaining hyper-squares, which will therefore have summed probability less than $\frac{\gamma}{m^d} m^d = \gamma$. The probability that arbitrary instance $i \in [0-1]^d$ will not lie in some selected hyper-square in $S_{<\alpha}$ is at most $1 - \gamma/m^d$. The probability that none of the N sample instances will lie in a selected hyper-square in $S_{<\alpha}$ is at most $(1 - \gamma/m^d)^N$. The probability that any hyper-square in $S_{<\alpha}$ is excluded by all N sample instances is at most $m^d(1 - \gamma/m^d)^N$. Since $m^d(1 - \gamma/m^d)^N < m^d e^{-N\gamma/m^d}$, we can force this probability to be small by setting $m^d e^{-N\gamma/m^d} \leq \delta$. N can then be solved for to yield

$$N \geq \frac{\lceil \frac{\sqrt{d}}{\alpha} \rceil^d}{\gamma} \ln \frac{\lceil \frac{\sqrt{d}}{\alpha} \rceil^d}{\delta}.$$

Consequently, with probability at least $(1 - \delta)$, each hyper-square in $S_{<\alpha}$ contains some sample instance of S. Also, the total probability of all the sub-squares in $S_{\geq\alpha}$ is less than γ. Since each instance of $[0-1]^d$ is in some hyper-square of $S_{<\alpha}$, except for a set of probability less than γ, then, with confidence at least $(1 - \delta)$, an arbitrary instance of $[0-1]^d$ is within α of some instance of S (except for a set of small probability). ■

The next lemma extends Lemma 2.1 to $k \geq 1$. The following generalization of an $\langle \alpha, \gamma \rangle$-net will be used.

Definition 2.2 *Let $X \subseteq \Re^d$ have an arbitrary but fixed probability distribution. $S \subseteq X$ is a k-$\langle \alpha, \gamma \rangle$-net for X if, for all $x \in X$, except for a set with probability less than γ, there exists at least k instances $s \in S$ such that distance$(s, x) < \alpha$.*

Lemma 2.2 *Let X be a bounded subset of $\Re^d$. Then there exists a polynomial p such that for any $0 < \alpha, \gamma, \delta < 1$, a random sample S containing $N \geq p(\frac{1}{\alpha}, \frac{1}{\gamma}, \frac{1}{\delta})$ instances from X, drawn according to any fixed probability distribution, will form a k-$\langle \alpha, \gamma \rangle$-net with probability at least $(1 - \delta)$.*

Proof 2.2 This proof ensures that, with high confidence, at least k of the N sample instances lies in each hyper-square of sufficient probability.

The process of drawing instances described in Lemma 2.1 needs to be repeated k times for this lemma. Since the probability distribution is fixed, the set $S_{\geq\alpha}$ is the same for each repetition. This yields the following inequality for assuring that k training instances are in each hyper-square of $S_{<\alpha}$:

$$N \geq \frac{k\lceil\frac{\sqrt{d}}{\alpha}\rceil^d}{\gamma} \ln \frac{\lceil\frac{\sqrt{d}}{\alpha}\rceil^d}{\delta'}$$

if the desired level of confidence that a single repetition will produce an $\langle\alpha,\gamma\rangle$-net is $(1-\delta')$. We will get a k-$\langle\alpha,\gamma\rangle$-net if *each* of the k repetitions produces an $\langle\alpha,\gamma\rangle$-net. A lower bound on the probability of this occurring is $(1-\delta')^k$. Thus, if we are required to produce a k-$\langle\alpha,\gamma\rangle$-net with confidence $(1-\delta)$, then we should set $(1-\delta')^k = (1-\delta)$. This yields $\delta' = 1 - \sqrt[k]{1-\delta}$. Substituting this expression for δ' above yields

$$N \geq \frac{k\lceil\frac{\sqrt{d}}{\alpha}\rceil^d}{\gamma} \ln \frac{\lceil\frac{\sqrt{d}}{\alpha}\rceil^d}{1-\sqrt[k]{1-\delta}}.$$

However, since $1 - \sqrt[k]{1-\delta} \geq \delta^k$ for small values of δ, this completes the proof. ■

Thus we are guaranteed that, by picking enough random samples, we will probably get a good coverage of any instance space.

3 Convergence Theorems

This section shows that IB1 can PAC-learn a large class of concepts and numeric functions with a polynomial bound on its number of required training instances. Nonetheless, IB1 cannot learn some target concepts, including those whose predictor attributes are *logically inadequate* (e.g., the concept of even numbers given positive and negative instances whose only attribute is their integer value).

Section 3.1 describes two theorems concerning IB1's ability to predict symbolic values. The second theorem makes a statistical assumption on the distributions of training sets, which requires a small extension of Valiant's model. Both theorems make *geometric* assumptions to constrain the target concept. Rosenblatt (1962) demonstrated that if a concept is an arbitrary hyper-half-plane (i.e., the set of instances on one side of a hyper-plane), then the perceptron learning algorithm is guaranteed to converge. The proofs analyzing IB1's learning abilities use a more general geometric assumption – that a target concept's boundary is a finite union of closed hyper-curves of finite length.[3]

Section 3.2 presents a theorem concerning IB1's ability to predict numeric values. The proof shows that IB1 can learn the class of continuous, real-valued numeric functions with bounded slope in polynomial time. Since the PAC-learning model has rarely been used to address the learning of numeric-valued functions, a substantially different definition of polynomial learnability is used in Section 3.2, although it preserves the spirit of Valiant's original model.

[3]This class has *infinite* VC dimension (Vapnik & Chervonenkis, 1971). Blumer, Ehrenfeucht, Haussler, and Warmuth (1986) proved that a concept class $\mathcal{C}$ is learnable with respect to the class of all probability distributions iff $\mathcal{C}$ has a *finite* VC dimension. We can show that IB1 can learn a class with an infinite VC dimension because our theorem restricts the class of allowable probability distributions.

3.1 Convergence Theorems: Predicting Symbolic Values

This section details convergence theorems for the IB1 algorithm. The following definition for polynomial learnability will be used. It modifies Valiant's (1984) model by constraining the class of allowable probability distributions $\mathcal{P}$.

Definition 3.1 *A class of concepts $\mathcal{C}$ is* polynomially learnable *with respect to a class of probability distributions $\mathcal{P}$ iff there exists a polynomial p and an algorithm A such that, for any $0 < \epsilon, \delta < 1$, if at least $p(\frac{1}{\epsilon}, \frac{1}{\delta})$ positive and negative instances of $C \in \mathcal{C}$ are drawn according to any fixed probability distribution $P \in \mathcal{P}$, then, with confidence at least $(1-\delta)$, A will generate an approximation $C' \in \mathcal{C}$ that differs from C on a set of instances with probability less than ϵ.*

By definition, the instances predicted by IB1 to belong to a concept C are those for which at least $\lceil\frac{k}{2}\rceil$ of the set of k nearest training instances are in C. However, the proofs in this section also apply if a similarity-weighted vote among the k nearest training instances is used instead.

Theorem 3.1 describes the relationship, for a particular class of concepts, between a target concept C and the concept description approximation C' converged on by IB1 for $d, k \geq 1$. Theorem 3.2 then shows that, by restricting the class of allowable probability distributions $\mathcal{P}$ to those representable by bounded probability density functions, Theorem 3.1 can be used to prove polynomial learnability for this class of concepts.

Figure 1 on the following page illustrates a few more definitions needed for the analysis.

Definition 3.2 *For any $\alpha > 0$, the* α-core *of a set C is the set of instances of C that are at least a distance α from any instance not in C.*

Definition 3.3 *The* α-neighborhood *of C is the set of instances that are within α of some instance of C.*

Definition 3.4 *A set of instances C' is an* $\langle\alpha,\gamma\rangle$-approximation *of C if, ignoring some set $S_{\geq\alpha}$ with probability less than γ, it contains the α-core of C and is contained in the α-neighborhood of C.*

The following theorem describes, in a geometric sense, how accurately IB1's derived concept description approximates the target concept. In particular, the IB1 algorithm converges (with probability at least $(1-\delta)$) to a concept that is an $\langle\alpha,\gamma\rangle$-approximation of the target concept.

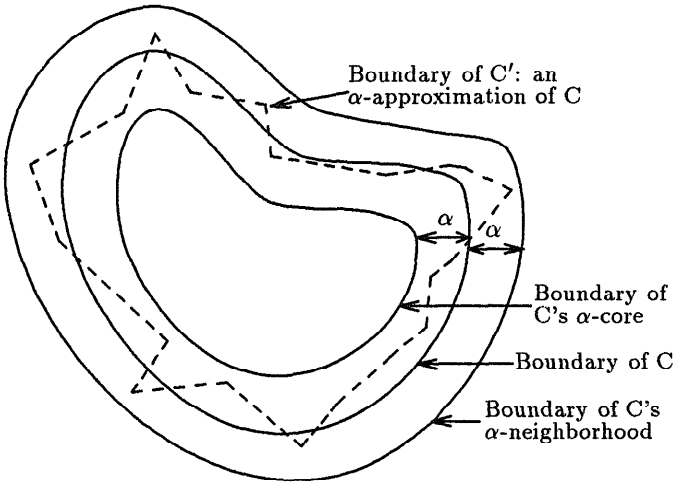

Figure 1: Exemplifying some terms used for analyzing learnability. This instance space has two numeric-valued attributes.

Theorem 3.1 *Let C be any region bounded by a closed curve in a bounded subset of $\Re^d$. Given $0 < \alpha, \delta, \gamma < 1$, then the IB1 algorithm with $k \geq 1$ converges with a polynomial number of training instances to C', where*

$$(\alpha\text{-}core(C) - S_{\geq\alpha}) \subseteq (C' - S_{\geq\alpha}) \subseteq (\alpha\text{-}neighborhood(C) - S_{\geq\alpha})$$

with probability $\geq (1-\delta)$, where $S_{\geq\alpha}$ is a set with probability less than γ.

Proof 3.1 We may assume, without loss of generality, that the bounded subset of $\Re^d$ is $[0-1]^d$. Let $0 < \alpha, \delta, \gamma < 1$. Lemma 2.2 states that, if

$$N \geq \frac{k\lceil\frac{\sqrt{d}}{\alpha}\rceil^d}{\gamma} \ln \frac{\lceil\frac{\sqrt{d}}{\alpha}\rceil^d}{1 - \sqrt[k]{1-\delta}},$$

then any N randomly-selected training instances will form a k-$\langle\alpha,\gamma\rangle$-net (with probability at least $(1-\delta)$). Let $S_{\geq\alpha}$ be the set of instances in $[0-1]^d$ that are not within α of k of the N training instances.

Two inclusions need to be proven. The first must show that, excluding the instances of $S_{\geq\alpha}$, the α-core of C is contained in C' (and thus in $C' - S_{\geq\alpha}$). Let p be an arbitrary instance in the α-core of C not in $S_{\geq\alpha}$ and let K be its set of k nearest (i.e., most similar) training instances. Since the distance between each $s \in K$ and p is less than α and p is in the α-core, then each s is also in C. Thus K correctly predicts that p is a member of C. Equivalently, this shows that p is a member of C'. Consequently, $(\alpha\text{-core}(C) - S_{\geq\alpha}) \subseteq (C' - S_{\geq\alpha})$.

The second inclusion states that $C' - S_{\geq\alpha}$ is contained in the α-neighborhood of C. This can be proven by showing that, if p is outside the α-neighborhood of C, then p is outside of $C' - S_{\geq\alpha}$. Let p be an arbitrary instance outside the α-neighborhood of C and let K be its set of k most similar neighbors. If p is not in $S_{\geq\alpha}$, then each $s \in K$ is within α of p, so each s is outside of C. In this case, K correctly predicts that p is not a member of C. Since no instance outside the α-neighborhood of C, excluding instances in $S_{\geq\alpha}$, is predicted by C' to be a member of C, then $(C' - S_{\geq\alpha}) \subseteq (\alpha\text{-neighborhood}(C) - S_{\geq\alpha})$. ■

Notice that Theorem 3.1 does not specify what the probability is of the set on which C' and C differ. Rather, it only shows where and how prediction errors could occur in terms of the geometry of C within the instance space. Some constraints on both the length of the boundary of C and the probability of regions of a given area are needed to bound this probability of error.

Theorem 3.2 adds these constraints. For reasons of simplicity, it arbitrarily delegates half of the allowed prediction error ϵ to each way that errors can arise (i.e., (1) when $p \in \alpha$-neighborhood of C and $p \notin \alpha$-core of C or (2) when $p \in S_{\geq\alpha}$). The proof shows that, for a large class of probability distributions, IB1 will, with high probability, converge to an approximately correct definition of the target concept for a large class of concepts in a bounded subset of $\Re^d$ with $d \geq 1$.

Theorem 3.2 *Let $\mathcal{C}$ be the class of all concepts in a bounded subset of $\Re^d$ that consist of a finite set of regions bounded by closed hyper-curves of total hyper-length less than L. Let $\mathcal{P}$ be the class of probability distributions representable by probability density functions bounded from above by B. Then $\mathcal{C}$ is polynomially learnable from examples with respect to $\mathcal{P}$ using IB1.*

Proof 3.2 Again we may assume that the bounded region is $[0-1]^d$. In Theorem 3.1, if the length of the boundary of C is less than L, then the total area between the α-core and the α-neighborhood of C is less than $2L\alpha$. Then $2LB\alpha$ is an upper bound on the probability of that area. Therefore, the total error made by C' in Theorem 3.1 is less than $2LB\alpha + \gamma$. If we fix $\gamma = 2LB\alpha = \frac{\epsilon}{2}$, then $\alpha = \frac{\epsilon}{4LB}$, and the theorem follows by substituting these expressions for γ and α into the inequality derived in Lemma 2.2. This yields

$$N \geq \frac{2k\lceil\frac{4LB\sqrt{d}}{\epsilon}\rceil^d}{\epsilon} \ln \frac{\lceil\frac{4LB\sqrt{d}}{\epsilon}\rceil^d}{1 - \sqrt[k]{1-\delta}}$$ ■

This proof has several practical implications. First, the number of instances required by IB1 to learn this class of concepts is also polynomial in L and B, which suggests that IB1 will perform best when the target concept's boundary size is minimized. Second, C' could be any subset of the α-neighborhood of C when the α-core is empty, which could occur when C's shape is extremely thin and α is chosen to be too large. The IBL approximation of C could be poor in this case. Third, IB1 cannot distinguish a target concept from anything containing its α-core and contained in its α-neighborhood; small perturbations in the shape of a target concept are not captured by IB1. Fourth, except for a set of size less than γ, the set of false positives is contained in the "outer ribbon" (the α-neighborhood of C excluding C) and the set of false negatives is contained in the "inner ribbon." Fifth, as the number of

predictor attributes increases, the expected number of instances required to learn concepts will increase exponentially. Space transformations that reduce this dimensionality or reduce L will significantly increase the efficiency of IBL algorithms. Finally, no assumptions about the convexity of the target concept, its number of disjuncts, nor their relative positions were made.

3.2 Convergence Theorems: Predicting Numeric Values

This section defines PAC-learnability for predicting numeric values and proves that IB1 can PAC-learn the class of continuous functions with bounded slope.

Definition 3.5 *The* error of a real-valued function f' *in predicting a real-valued function f, for an instance x, is* $|f(x) - f'(x)|$.

Definition 3.6 *Let f be a real-valued target function. Let B_f be the least upper bound of the absolute value of the slope between any two instances on the curve of f. If B_f is finite, then we say that f has* bounded slope. *If $\mathcal{C}$ is a class of functions in which each $f \in \mathcal{C}$ has bounded slope, and $B_f < B$ for some number B, then we say that $\mathcal{C}$ has bounded slope (bounded by B).*

Continuously differentiable functions on $[0, 1]$ have bounded slope. As a counterexample, the function $\sin(\frac{1}{x})$ does not have bounded slope.

Definition 3.7 *Let $\mathcal{C}$ be a class of functions for which each $f \in \mathcal{C}$ is a function from the unit hypercube in $\Re^d$ to $\Re$. $\mathcal{C}$ is* polynomially learnable *if there exists an algorithm A and a polynomial p such that, for any $0 < \gamma, \alpha, \delta < 1$, given an $f \in \mathcal{C}$, if $p(\frac{1}{\gamma}, \frac{1}{\alpha}, \frac{1}{\delta})$ or more examples are chosen according to any fixed probability distribution on $[0 - 1]^d$, then, with confidence at least $(1 - \delta)$, A will output an approximation of f with error less than α everywhere except on a set of instances with probability less than γ.*

In this definition γ is a bound on the size of the set on which "significant" prediction errors can occur.

By definition, IB1 computes a new instance's similarity to all instances in a concept description and predicts that the target value is the similarity-weighted average derived from the k most similar instances. Thus, it generates a piecewise linear approximation to the target function.

The next theorem demonstrates that continuous, real-valued functions with bounded slope in the unit hypercube in $\Re^d$ are polynomially learnable by IB1. Note that the class of functions with slope bounded by B includes the large class of differentiable functions with derivative bounded by B.

Theorem 3.3 *Let $\mathcal{C}$ be the class of continuous, real-valued functions on the unit hypercube in $\Re^d$ with slope bounded by B. Then $\mathcal{C}$ is polynomially learnable by IB1 with $k \geq 1$.*

Proof 3.3 Let f be a continuous function on $[0 - 1]^d$. Let $0 < \alpha, \gamma, \delta < 1$. The bound B ensures that f will not vary much on a small interval.

Let $\alpha' = \frac{\alpha}{B}$. Draw N training instances in accordance with Lemma 2.2, with α replaced by α'. Let f' be the approximation that IB1 generates for f, and let x be an arbitrary instance in an interval of $S_{<\alpha'}$. The point is to ensure that the error of f' at x is small (i.e., less than α). That is, it must be shown that $|f(x) - f'(x)| < \alpha$.

Let K be the set of x's k most similar training instances. Since $f'(x)$ is a weighted-average of the target values of each $x' \in K$, it suffices to show that $|f(x) - f(x')| < \alpha$.

Because N is sufficiently large, the k most similar neighbors of x must all be within α' of x. Also, we know that B is an upper bound on the slope between x and x'. Thus, since

$$|f(x) - f(x')| = |slope(x, x')| \times \text{distance}(x, x'),$$

then

$$|f(x) - f(x')| < B \times \alpha' = \alpha.$$

Therefore, IB1 will yield a prediction for x's target value that is within α of x's actual target value if at least N training instances are provided, where

$$N \geq \frac{k\lceil \frac{B\sqrt{d}}{\alpha} \rceil^d}{\gamma} \ln \frac{\lceil \frac{B\sqrt{d}}{\alpha} \rceil^d}{1 - \sqrt[k]{1 - \delta}}.$$

Thus, given any $f \in \mathcal{C}$, if at least that many training instances are provided, IB1 will, with confidence at least $(1 - \delta)$, yield an approximation f' with error less than α for all instances except those in a set of probability less than γ. ■

Thus, the number of required training instances is also polynomial in B.

Many functions have bounded slope. For example, any piecewise linear curve and any function with a continuous derivative defined on a closed and bounded region in $\Re^d$ has a bounded slope. Therefore, IB1 can accurately learn a large class of numeric functions using a polynomial number of training instances. However, it may not be able to PAC-learn numeric functions whose maximum absolute slope is unbounded. For example, $\sin(\frac{1}{x})$. As x approaches 0, the derivative of this function is unbounded.

4 Conclusion

This paper detailed PAC-learning analyses of IB1, a simple instance-based learning algorithm that performed well on a variety of supervised learning tasks (Aha, Kibler, & Albert, 1991). The analyses show that IB1 can PAC-learn large classes of symbolic concepts and numeric functions. These analyses help to explain IB1's capabilities and complement our earlier empirical studies. However, we did not address their average-case behavior, an important topic of future research. Analyses for more elaborate IBL algorithms, such as those that tolerate noisy instances, tolerate irrelevant

attributes, or process symbolic-valued attributes, would also improve our understanding of these practical learning algorithms' capabilities and limitations.

Acknowledgments

Thanks to Dennis Kibler who initiated this line of research and was a collaborator on the initial analyses (published elsewhere). Thanks also to Dennis Volper and our reviewers for comments and suggestions, and to Caroline Ehrlich for her assistance on preparing the final draft.

References

Aha, D. W. (1989). Incremental, instance-based learning of independent and graded concept descriptions. In *Proceedings of the Sixth International Workshop on Machine Learning* (pp. 387–391). Ithaca, NY: Morgan Kaufmann.

Aha, D. W., & Kibler, D. (1989). Noise-tolerant instance-based learning algorithms. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 794–799). Detroit, MI: Morgan Kaufmann.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning, 6*, 37–66.

Angluin, D., & Laird, P. (1988). Learning from noisy examples. *Machine Learning, 2*, 343–370.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. (1986). Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension. In *Proceedings of the Eighteenth Annual Association for Computing Machinery Symposium on Theory of Computing* (pp. 273–282). Berkeley, CA: Association for Computing Machinery.

Bradshaw, G. (1987). Learning about speech sounds: The NEXUS project. In *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 1–11). Irvine, CA: Morgan Kaufmann.

Cost, S., & Salzberg, S. (1990). *A weighted nearest neighbor algorithm for learning with symbolic features* (Technical Report JHU-90/11). Baltimore, MD: The Johns Hopkins University, Department of Computer Science.

Cover, T. M. (1968). Estimation by the nearest neighbor rule. *IEEE Transactions on Information Theory, 14*, 50–55.

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory, 13*, 21–27.

Haussler, D. (1987). Bias, version spaces and Valiant's learning framework. In *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 324–336). Irvine, CA: Morgan Kaufmann.

Kearns, M., Li, M., Pitt, L., & Valiant, L. G. (1987). On the learnability of Boolean formulae. In *Proceedings of the Nineteenth Annual Symposium on the Theory of Computer Science* (pp. 285–295). New York, NY: Association for Computing Machinery.

Kibler, D., & Aha, D. W. (1987). Learning representative exemplars of concepts: An initial case study. In *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 24–30). Irvine, CA: Morgan Kaufmann.

Kibler, D., Aha, D. W., & Albert, M. (1989). Instance-based prediction of real-valued attributes. *Computational Intelligence, 5*, 51–57.

Li, M., & Vitanyi, P. M. B. (1989). *A theory of learning simple concepts under simple distributions and average case complexity for universal distribution (preliminary version)* (Technical Report CT-89-07). Amsterdam, Holland: University of Amsterdam, Centrum voor Wiskunde en Informatica.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning, 2*, 285–318.

Moore, A. W. (1990). Acquisition of dynamic control knowledge for a robotic manipulator. In *Proceedings of the Seventh International Conference on Machine Learning* (pp. 244–252). Austin, TX: Morgan Kaufmann.

Rivest, R. (1987). Learning decision lists. *Machine Learning, 2*, 1–20.

Rosenblatt, F. (1962). *Principles of neurodynamics.* New York, NY: Spartan.

Salzberg, S. (1988). *Exemplar-based learning: Theory and implementation* (Technical Report TR-10-88). Cambridge, MA: Harvard University, Center for Research in Computing Technology.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM, 29*, 1213–1228.

Tan, M., & Schlimmer, J. C. (1990). Two case studies in cost-sensitive concept acquisition. In *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 854–860). Boston, MA: American Association for Artificial Intelligence Press.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM, 27*, 1134–1142.

Valiant, L. G. (1985). Learning disjunctions of conjunctions. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 560–566). Los Angeles, CA: Morgan Kaufmann.

Vapnik, V. N., & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications, 16*, 264–280.

Waltz, D. (1990). Massively parallel AI. In *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 1117–1122). Boston, MA: American Association for Artificial Intelligence Press.