Machine Learning, 21, 269–304 (1995)

Sample Compression, Learnability, and the Vapnik-Chervonenkis Dimension

SALLY FLOYD* floyd@ee.lbl.gov
M/S 46A-1123, Lawrence Berkeley Laboratory, One Cyclotron Road, Berkeley, CA 94720.

MANFRED WARMUTH**
Department of Computer Science, University of California, Santa Cruz, CA 95064.

Editor: Leonard Pitt

Abstract. Within the framework of pac-learning, we explore the learnability of concepts from samples using the paradigm of sample compression schemes. A sample compression scheme of size k for a concept class $C \subseteq 2^X$ consists of a compression function and a reconstruction function. The compression function receives a finite sample set consistent with some concept in C and chooses a subset of k examples as the compression set. The reconstruction function forms a hypothesis on X from a compression set of k examples. For any sample set of a concept in C the compression set produced by the compression function must lead to a hypothesis consistent with the whole original sample set when it is fed to the reconstruction function. We demonstrate that the existence of a sample compression scheme of fixed-size for a class C is sufficient to ensure that the class C is pac-learnable.

Previous work has shown that a class is pac-learnable if and only if the Vapnik-Chervonenkis (VC) dimension of the class is finite. In the second half of this paper we explore the relationship between sample compression schemes and the VC dimension. We define *maximum* and *maximal* classes of VC dimension d. For every maximum class of VC dimension d, there is a sample compression scheme of size d, and for sufficiently-large maximum classes there is no sample compression scheme of size less than d. We discuss briefly classes of VC dimension d that are maximal but not maximum. It is an open question whether every class of VC dimension d has a sample compression scheme of size $O(d)$.

Keywords: Sample compression, Vapnik-Chervonenkis dimension, pac-learning

1. Introduction

In this paper we discuss the use of sample compression schemes within computational learning theory, and explore the relationships between pac-learning, sample compression, and the Vapnik-Chervonenkis dimension (a combinatorial parameter measuring the difficulty of learning of a concept class).

There are many examples of learning algorithms that use *sample compression*; that is, that select a subset of examples from a sample set, and use those examples to represent a hypothesis. A common example is the algorithm for learning axis-parallel rectangles in the plane. From any sample of positive and negative points in the plane, where the positive points are those contained in the target rectangle, and the negative points are

* S. Floyd was supported in part by the Director, Office of Energy Research, Scientific Computing Staff, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.
** M. Warmuth was supported by ONR grants N00014-K-86-K-0454 and NO0014-91-J-1162 and NSF grant IRI-9123692.

those not contained in the target rectangle, the learning algorithm needs only to save the top, bottom, leftmost, and rightmost positive examples. The hypothesis represented by these examples is the smallest axis-parallel rectangle containing these four points.

From Littlestone and Warmuth (1986), a sample compression scheme of size k for a concept class consists of a compression function and a reconstruction function. Given a finite set of examples, the compression function selects a compression set of at most k of the examples. The reconstruction function uses this compression set to construct a hypothesis for the concept to be learned. For a sample compression scheme, the reconstructed hypothesis must be guaranteed to predict the correct label for all of the examples in the original sample set.

For a concept class with a sample compression scheme of size k, any sample set can be represented by a subset of size k. This quantifies the extent to which the compression scheme can *generalize* from an arbitrary sample set. Previous work from Blumer, Ehrenfeucht, Haussler, and Warmuth (1989) has shown that a class is pac-learnable if and only if the Vapnik-Chervonenkis dimension is finite. In this work we follow an alternate approach, and show that the existence of an appropriate sample compression scheme is sufficient to ensure learnability. For certain concept classes we show that there always exists a sample compression scheme of size equal to the VC dimension of the class. Recently Freund (1995) and Helmbold and Warmuth (1995) have shown that there is always an extended version of a sample compression scheme that is essentially of size $O(d \log m)$ for any concept class of VC dimension d and sample set size m. (See the last section for a discussion of these results).

The VC dimension has also been used in computational geometry by Haussler and Welzl (1987) to characterize the complexity of processing range queries. However recently alternate methods have been developed for the same problem using random sampling and divide and conquer. Some of the proofs by Clarkson (1992) for these alternate methods are similar to the proofs presented here of sample complexity bounds for sample compression schemes in the pac-learning model.

Our use of a sample compression scheme is also somewhat different from Rissanen's Minimum Description Length Principle (MDLP) (1986). Borrowing from Quinlan and Rivest (1989), the minimum description length principle states that from a given set of hypotheses, the one that can predict the future with the most accuracy is the one that minimizes the combined coding length of the hypothesis and of the data that is incorrectly predicted by the hypothesis. In contrast, we measure the size of our compression set not by the number of bits used to encode the examples, but simply by the number of examples in the compression set. In addition, we use the approach of pac-learning to quantify the sample size needed to predict the future with an acceptable degree of accuracy.

Nevertheless, the idea behind MDLP, applied to a learning algorithm that is restricted to save only examples from the sample set, would be to minimize the sum of the number of examples used to express the hypothesis and the number of examples incorrectly predicted by the hypothesis. Thus, MDLP would imply that a learning algorithm restricted to saving a subset of the sample set should save the smallest number of examples that it can, while still correctly predicting all of the examples in the sample set. That is the approach followed in this paper.

This paper considers several general procedures for constructing sample compression schemes. We show that the existence of an appropriate mistake-bounded on-line learning algorithm is sufficient to construct a sample compression scheme of size equal to the mistake bound. With this result, known computationally-efficient mistake-bounded learning algorithms can be used to construct computationally-efficient sample compression schemes for classes such as k-literal monotone disjunctions and monomials. Also, since the Halving algorithm from Angluin (1988) and Littlestone (1988) has a mistake bound of $\log|C|$ for any finite concept class C, this implies that there is a sample compression scheme of size at most $\log|C|$.

For infinite concept classes a different approach is needed to construct sample compression schemes. Following Blumer et al. (1989), we use the Vapnik-Chervonenkis dimension for this purpose. Using definitions from Welzl (1987), we consider *maximum* and *maximal* concept classes of VC dimension d. Maximal concept classes are classes where no concept can be added without increasing the VC dimension of the class. Maximum classes are in some sense the largest concept classes. We show that any maximum concept class of VC dimension d has a sample compression scheme of size d. Further, this result is optimal; for any sufficiently large maximum class of VC dimension d, there can be no sample compression scheme of size less than d. From Littlestone and Warmuth (1986), it remains an open question whether there is a sample compression scheme of size $O(d)$ for every class of VC dimension d.

This paper explores the use of sample compression schemes in *batch* learning algorithms, where the learner is given a finite sample set, and constructs a hypothesis after examining all of the examples in the sample set. Sample compression schemes can also be used in *on-line* learning algorithms, where the learner receives examples one at a time and updates its hypothesis after each example. Such compression schemes in on-line learning algorithms are explored in more detail in Floyd's thesis (1989).

The paper is organized as follows. Section 2 reviews pac-learning and the VC dimension. In Section 3 we define the sample compression schemes of size at most k used in this paper. Section 4 gives sample compression schemes based on on-line mistake-bounded learning algorithms. As a special case we obtain sample compression schemes of size at most $\log|C|$ for any finite class C. In Section 5 we show that the existence of a sample compression scheme for a class C is sufficient to give a pac-learning algorithm for that class, and we improve slightly on the original sample complexity bounds from Littlestone and Warmuth for pac-learning algorithms based on sample compression schemes (1986).

In Section 6 we define maximal and maximum classes of VC dimension d, and give a sample compression scheme of size d that applies to any maximum class of VC dimension d. Combined with the results from previous sections, this improves somewhat on the previously-known sample complexity of batch learning algorithms for maximum classes. In Section 6.4 we show that this result is optimal; that is, for any sufficiently large maximum class of VC dimension d, there is no sample compression scheme of size less than d. Section 7 discusses sample compression schemes for maximal classes of VC dimension d. One of the major open problems is whether there is a sample compression scheme of size d for every maximal class of VC dimension d.

Definition. A-B is used to denote the difference of sets, so A-B is defined as {a ∈ A: a ∉ B}. We let $\ln x$ denote $log_e x$ and we let $\log x$ denote $log_2 x$.

A *domain* is any set X. A *concept* c on the domain X is any subset of X. For a concept $c \in C$ and element $x \in X$, $c(x)$ gives the classification of x in the class c. That is, $c(x) = 1$ if $x \in c$, and $c(x) = 0$ otherwise. The elements of $X \times \{0,1\}$ are called *examples. Positive examples* are examples labeled "1" and *negative examples* are labeled "0". The elements of X are sometimes called *unlabeled examples.* For any set $A \subset X$, we let $A^{\pm,c} \subset A \times \{0,1\}$ denote the set A of examples labeled as in the concept c. We let $A^{\pm}$ refer to an arbitrary set of labeled examples A.

A *concept class* C on X is any subset of 2^X. For $Y \subset X$, we define $C|Y$ as the *restriction* of the class C to the set Y. That is, $C|Y = \{c \cap Y : c \in C\}$. We say that the class C is finite if $|C|$ is finite; otherwise we say that the class C is infinite. Because C is finite, C can be considered as a class on a finite domain X. (If two elements in X have the same label for all concepts in the class C, then these two elements can be considered as a single element.)

A *sample set* is a set of examples; a *sample sequence* is a sequence of examples, possibly including duplicates. A sample set or sequence is *consistent* with a concept c if the labels of its examples agree with c.

2. Pac-learning and the VC dimension

In this section we review the model of probably approximately correct (pac) learning, and the connections between pac-learning and the Vapnik-Chervonenkis dimension. From Vapnik (1982) and Blumer et al. (1987, 1989), Theorem 1 states that finite classes are pac-learnable with a sample size that is linear in $\ln |C|$. For infinite classes, the Vapnik-Chervonenkis dimension was used to extend this result, showing that a class is pac-learnable from a fixed sample size only if the Vapnik-Chervonenkis dimension is finite. In this case, the sample size is linear in the Vapnik-Chervonenkis dimension.

First we review the model of pac-learning. Valiant (1984) introduced a pac-learning model of learning concepts from examples taken from an unknown distribution. In this model of learning, each example is drawn independently from a fixed but unknown distribution D on the domain X, and the examples are labeled consistently with some unknown target concept c in the class C.

Definition. The goal of the learning algorithm is to learn a good approximation of the target concept, with high probability. This is called "probably approximately correct" learning or *pac-learning.* A learning algorithm has as inputs an accuracy parameter ϵ, a confidence parameter δ, and an oracle that provides labeled examples of the target concept c, drawn according to a probability distribution D on X. The *sample size* of the algorithm is the number of labeled examples in the sample sequence drawn from the oracle. The learning algorithm returns a hypothesis h. The error of the hypothesis is the total probability, with respect to the distribution D, of the symmetric difference of c and h.

A concept class C is called *learnable* if there exists a pac-learning algorithm such that, for any ϵ and δ, there exists a fixed sample size such that, for any concept $c \in C$ and for any probability distribution on X, the learning algorithm produces a probably-approximately-correct hypothesis; a *probably-approximately-correct hypothesis* is one that has error at most ϵ with probability at least 1-δ. The *sample complexity* of the learning algorithm for C is the smallest required sample size, as a function of ϵ, δ, and parameters of the class C.

For a finite concept class C, Theorem 1 gives an upper bound on the sample complexity required for learning the class C. This upper bound is linear in $\ln|C|$.

THEOREM 1 *(Vapnik, 1982, and Blumer et al., 1989) Let C be any finite concept class. Then for sample size greater than $\frac{1}{\epsilon}\ln\frac{|C|}{\delta}$, any algorithm that chooses a hypothesis from C consistent with the examples is a learning algorithm for C.*

Definition. For infinite classes such as geometric concept classes on R^n, Theorem 1 cannot be used to obtain bounds on the sample complexity. For these classes, a parameter of the class called the Vapnik-Chervonenkis dimension is used to give upper and lower bounds on the sample complexity. For a concept class C on X, and for S $\subseteq$ X, if $C|S = 2^S$, then the set S is *shattered* by C. The *Vapnik-Chervonenkis dimension* (VC dimension) of the class C is the largest integer d such that some $S \subseteq X$ of size d is shattered by C. If arbitrarily large finite subsets of X are shattered by the class C, then the VC dimension of C is infinite. Note that a class C with one concept is of VC dimension 0. By convention, the empty class is of VC dimension -1.

If the class $C \subseteq 2^X$ has VC dimension d, then for all $Y \subset X$, the restriction $C|Y$ has VC dimension at most d.

Theorem 2 from Blumer et al. (1989), adapted from Vapnik and Chervonenkis (1971), gives an upper bound on the sample complexity of learning algorithms in terms of the VC dimension of the class.

THEOREM 2 *(Blumer et al., 1989) Let C be a well-behaved[1] concept class. If the VC dimension of C is $d < \infty$, then for $0 < \epsilon, \delta < 1$ and for sample size at least*

$$max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{8d}{\epsilon}\log\frac{13}{\epsilon}\right),$$

C is learnable by any algorithm that finds a concept c from C consistent with the sample sequence.

In Theorem 2, the VC dimension essentially replaces $\ln|C|$ from Theorem 1 as a measure of the size of the class C. Blumer et al. (1989) further show that a class is pac-learnable from a fixed-size sample if and only if the VC dimension of the class is finite. Shawe-Taylor, Anthony, and Biggs (1989) improve the sample size in Theorem 2 to

$$\frac{1}{\epsilon(1-\alpha)}\ln\frac{2}{\delta} + \frac{d}{\epsilon(1-\alpha)}\left(\ln\frac{1}{\epsilon} + 2(\ln 2 + \ln\frac{1}{\alpha})\right)$$

$$= \frac{1}{(1-\alpha)}\left(\frac{1}{\epsilon}\ln\frac{2}{\delta} + \frac{2d\ln 2}{\epsilon} + \frac{d}{\epsilon}\ln\frac{1}{\epsilon\alpha^2}\right)$$

for $0 < \alpha < 1$. The second line is to facilitate comparison with bounds derived later in the paper.

As in the theorems in this section, the theoretical results in this paper generally do not address computational concerns. We address the question of computationally-efficient learning algorithms separately in our examples. Computational issues regarding compression schemes are discussed further in Floyd's thesis (1989).

3. Sample compression schemes

In this section we define a sample compression scheme of size at most k for a concept class C, and we give several examples of such a sample compression scheme.

Definition. This paper uses the simplest version of the compression schemes introduced by Littlestone and Warmuth (1986); extended versions of these compression schemes save additional information. A *sample compression scheme of size at most* k for a concept class C on X consists of a compression function and a reconstruction function. The *compression function* f maps every finite sample set to a *compression set*, a subset of at most k labeled examples. The *reconstruction function* g maps every possible compression set to a hypothesis $h \subseteq X$. This hypothesis is not required to be in the class C. For any sample set $Y^{\pm,c}$ labeled consistently with some concept c in C, the hypothesis $g(f(Y^{\pm,c}))$ is required to be consistent with the original sample set $Y^{\pm,c}$. In this paper the *size* of a compression set refers to the number of examples in the compression set.

Example: rectangles. Consider the class of axis-parallel rectangles in R^2. Each concept corresponds to an axis-parallel rectangle; the points within the axis-parallel rectangle are labeled '1' (positive), and the points outside the rectangle are labeled '0' (negative). The compression function for the class of axis-parallel rectangles in R^2 takes the leftmost, rightmost, top, and bottom positive points from a set of examples; this compression function saves at most four points from any sample set. The reconstruction function has as a hypothesis the smallest axis-parallel rectangle consistent with these points. This hypothesis is guaranteed to be consistent with the original set of examples. This class is of VC dimension four. □

Example: intersection closed concept classes and nested differences thereof. For any concept class $C \in 2^X$ and any subset S of the domain X the *closure of* S *with respect to* C, denoted by $\mathrm{CLOS}(S)$, is the set $\bigcap\{c : c \in C \text{ and } S \subseteq c\}$. A concept class C is *intersection closed*[2] if whenever S is a finite subset of some concept in C then the *closure* of S, denoted as $\mathrm{CLOS}(S)$, is a concept of C. Clearly axis-parallel rectangles in R^n are intersection closed and there are many other examples given by Helmbold, Sloan, and Warmuth (1990, 1992) such as monomials, vector spaces in R^n and integer lattices.

For any set of examples labeled consistently with some concept in the intersection closed class C, consider the closure of the positive examples. This concept, the "smallest"

concept in C containing those positive examples, is clearly consistent with the whole sample. A *minimal spanning set* of the positive examples is any minimal subset of the positive examples whose closure is the same as the closure of all positive examples. Such a minimal spanning set can be used as a compression set for the sample. Helmbold et al. (1990) have proved that the size of such minimal spanning sets is always bounded by the VC dimension of the intersection closed concept class. Thus, any intersection closed class has a sample compression scheme whose size is at most the VC dimension of the class.

Surprisingly, using the methods of Helmbold et al. (1990) one can obtain a sample compression scheme for nested differences of concepts from an intersection-closed concept class. For example, $c_1 - (c_2 - (c_3 - (c_4 - c_5)))$ is a nested difference of depth five. The VC dimension of nested differences of depth p is at most p times the VC dimension of the original class. Again the compression sets for classes of nested differences of bounded depth have size at most as large as the VC dimension of the class. We will only sketch how the compression set is constructed for the more complicated case of nested differences. First a minimal spanning set of all positive examples is found and added to the compression set. Then all negative examples falling in the closure of the first minimal spanning set are considered. Again a minimal spanning set of all these examples is added to the compression set. Then all positive example falling in the closure of the last spanning set are considered and a minimal spanning is added to the compression set. This process is iterated until there are no more examples left. □

Example: intervals on the line. Space efficient learning algorithms for the class of at most n intervals on the line have been studied by Haussler (1988). One compression function for the class of at most n intervals on the line scans the points from left to right, saving the first positive example, and then the first subsequent negative example, and so on. At most $2n$ examples are saved. The reconstruction function has as a hypothesis the union of at most n intervals, where the leftmost two examples saved denote the boundaries of the first positive interval, and each succeeding pair of examples saved denote the boundaries of the next positive interval. For the sample set in Figure 1, this compression function saves the examples $\{\langle x_3, 1\rangle, \langle x_5, 0\rangle, \langle x_7, 1\rangle, \langle x_{11}, 0\rangle, \langle x_{14}, 1\rangle, \langle x_{16}, 0\rangle\}$ which represents the hypothesis $[x_3, x_5) \cup [x_7, x_{11}) \cup [x_{14}, x_{16})$. Note that the class of at most n intervals on the line is of VC dimension $2n$. □

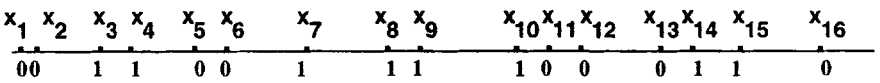

Figure 1. A sample set from the union of three intervals on the line.

Note that the sample compression scheme defined in this section differs from the traditional definition of data compression. Consider the compression function for axis-parallel rectangles that saves at most four positive examples from a sample set. From a

compression set of at most four examples, it is not possible to reconstruct the original set of examples. However, given any unlabeled point from the original set of examples, it is possible to reconstruct the label for that point.

4. Compression Schemes and Mistake-bounded Learning Algorithms

In this section, we discuss the relationship between sample compression schemes and mistake-bounded on-line learning algorithms. An on-line learning algorithm P proceeds in trials. In each trial t the algorithm is presented with an unlabeled example x_t over some domain X and produces a binary prediction. It then receives a label for the example and incurs a *mistake* if the label differs from the algorithm's prediction.

Definition. The *mistake bound* of algorithm P for a concept class $C \subseteq 2^X$ is the worst-case number of mistakes that P can make on any sequence of examples labeled consistently with some concept in C. We say that an on-line learning algorithm is *mistake-driven*[3] if its hypothesis is a function of the past *sequence* of examples in which the algorithm made a mistake. Thus mistake-driven algorithms "ignore" those examples on which the algorithm predicted correctly.

Any mistake-bounded learning algorithm P can be converted to a mistake-driven algorithm Q; after a mistake, let Algorithm Q take the hypothesis that algorithm P would have taken if the intervening non-mistake examples had not happened. It is easy to see that this conversion does not increase the worst-case mistake bound of the algorithm. We show that any class with a mistake-bounded learning algorithm that is also mistake-driven has a One-Pass Compression Scheme with the same size bound as the mistake bound.

Consider a class $C \subseteq 2^X$ with an on-line learning algorithm P that has a mistake bound k. That is, given any concept c from the class C, and any sequence of examples from X, the on-line learning algorithm P makes at most k mistakes in predicting the labels of the examples of c. We assume some default ordering on X, and in the algorithm below we are only concerned with mistake bounds where the order of examples in the arbitrary sample set is consistent with the default order on X. Further, assume that the algorithm P is mistake-driven. The One-Pass Compression Scheme below uses the on-line learning algorithm P to construct a sample compression scheme of size at most k for C.

The One-Pass Compression Scheme (for finite classes with mistake-driven and mistake-bounded learning algorithms).

- The compression function: The input is the labeled sample set $Y^{\pm,c} \subseteq X \times \{0,1\}$. The One-Pass Compression Scheme examines the examples from the sample set in the default order, saving all examples for which the mistake-bounded learning algorithm P made a mistake predicting the label.
- The reconstruction function: The input to the reconstruction function is a compression set $A^{\pm}$. For any element in the compression set, the reconstruction function

predicts the label for that element in the compression set. For any element x not in the compression set, the reconstruction function considers those elements from the compression set that precede x in the default order on X. The reconstruction function applies the learning algorithm P to those elements, in order. The reconstruction function then uses the current hypothesis of the learning algorithm P to predict the label for x.

Theorem 3 shows that for any class with a mistake-driven and mistake-bounded learning algorithm with bound k, the One-Pass Compression Scheme gives a sample compression scheme of size at most k.

THEOREM 3 *Let $C \subseteq 2^X$ have a mistake-driven mistake-bounded learning algorithm P with bound k, with respect to the default ordering on X. Then the One-Pass Compression Scheme is a sample compression scheme of size at most k for the class C.*

Proof: For any concept c from C and for any sample set, the mistake-bounded learning algorithm makes at most k mistakes. Therefore, the One-Pass Compression Scheme produces a compression set of size at most k.

We will reason that for any element x not in the compression set, the reconstruction function reproduces the same prediction on x as it did when the mistake-bounded learning algorithm was used to construct the compression set. For any element from the compression set that precedes x in the default order on X, the reconstruction function examines that element in the same order as did the mistake-bounded learning algorithm looking at the original sample set. Since the algorithm is mistake-driven we have that in both cases the algorithm predicts on x using the hypothesis represented by the sequence of mistakes preceding x. (Note that all the mistakes were saved in the compression set.) Since x was not added to the compression set the learning algorithm predicted correctly on x when the compression set was constructed and the reconstruction function produces the same correct prediction. ■

Example: k-literal disjunctions. Littlestone (1988) gives a computationally-efficient, mistake-driven and mistake-bounded learning algorithm, called Winnow1, for learning k-literal monotone disjunctions. This algorithm is designed to learn efficiently even in the presence of large numbers of irrelevant attributes. In particular, the results from Littlestone, along with Theorem 3, give a sample compression scheme of size at most $k(1+2\log n/k)$ for the class of monotone disjunctions with at most k literals. This class has a VC dimension, and therefore a lower bound on the optimal mistake bound, of at least $(k/8)(1+\log n/k)$. Using reductions, Littlestone (1988, 1989) shows that Winnow1 and its relatives lead to efficient learning algorithms with firm mistake bounds for a number of classes: non-monotone disjunctions with at most k literals, k-literal monomials, r of k threshold functions, l-term DNF where each monomial has at most k literals. Thus immediately we can obtain sample compression schemes for these classes. □

Example: monomials. We now discuss an elimination algorithm for learning monomials from Valiant (1984) and Angluin (1988); similar elimination algorithms exist for more

general conjunctive concepts. Let $C \subseteq 2^X$ be the class of monomials over n variables, for $X = \{0,1\}^n$ and for some default order on X. The initial hypothesis for the mistake-bounded learning algorithm is the monotone monomial containing all $2n$ literals. The first positive example eliminates n literals and for each further positive example whose label is predicted incorrectly by the learning algorithm, at least one literal is deleted from the hypothesis monomial; thus, this learning algorithm is mistake-driven, and by Theorem 3 we obtain a sample compression scheme of size at most $n+1$ using the One-Pass Compression Algorithm. After one pass, the most general (maximal) monomial consistent with the compression set is consistent with all of the examples in the sample set, so there is a reconstruction function that is independent of the ordering on X. The class of monomials has VC dimension at least n. □

4.1. Sample compression schemes for finite classes

Angluin (1988) and Littlestone (1988) have considered a simple on-line learning algorithm for learning an arbitrary finite concept class $C \subset 2^X$ with at most $\log |C|$ mistakes. This is called the Halving algorithm and works as follows: The algorithm keeps track of all concepts consistent with all past examples. From Mitchell (1977), this set of concepts is called the *version space*. For each new example, the algorithm predicts the value that agrees with the majority of the concepts in the version space. (In the case of a tie, the Halving algorithm can predict either label.) After each trial, the algorithm updates the version space.

Clearly the mistake bound of this algorithm comes from the fact that each time the algorithm makes a mistake the size of the version space is reduced by at least 50%. There is also a mistake-driven version of this algorithm: simply update the version space only when a mistake occurs. This mistake-driven algorithm together with Theorem 3 immediately leads to a sample compression scheme.

THEOREM 4 *Let $C \subseteq 2^X$ be any finite concept class. Then the One-Pass Compression Scheme using the mistake-driven Halving algorithm gives a sample compression scheme of size at most $\log |C|$ for the class C.*

Note that the hypotheses of the mistake-driven Halving algorithm, of the algorithm Winnow1 for learning k literal monotone disjunctions, and of the elimination algorithm for learning monomials, all have the property that they depend only on the *set* of examples on which the algorithm made a mistake in the past; their order does not matter. This immediately leads to a simpler construction of a sample compression scheme for mistake-driven algorithms that have this additional property. Now the compression function iteratively finds any example that is not yet in the compression set on which the algorithm makes a mistake and feeds this example to the algorithm as well as adding it to the compression set. The number of examples saved until the hypothesis of the algorithm is consistent with all remaining examples is clearly bounded by the mistake bound of the on-line algorithm. The reconstruction function simply feeds the compression set to the on-line algorithm (in any order) and uses the resulting hypothesis to predict on examples not in the compression set.

5. Batch learning algorithms using sample compression schemes

In this section, we show that any class with a sample compression scheme of size at most d has a corresponding batch learning algorithm, with the sample complexity of the batch learning algorithm given by Theorem 6. Coupled with results later in the paper, this improves slightly on known results of the sample complexity of maximum classes with finite VC dimension.

For a class $C \subseteq 2^X$ with a sample compression scheme of size at most k, the corresponding batch learning algorithm is a straightforward application of the sample compression scheme. The learner requests a sample sequence $Y^{\pm,c}$ of m examples labeled consistently with some concept in the class C. The learner then converts the sample sequence to a sample set, removing duplicates, and uses the compression function from the sample compression scheme to find a compression set for this sample set. The hypothesis of the batch learning algorithm is the hypothesis reconstructed from the compression set by the sample compression scheme's reconstruction function. Note that this hypothesis is guaranteed to be consistent with all of the examples in the original sample set. However, the sample compression scheme does not require this hypothesis to be consistent with some concept from the class C.

THEOREM 5 *(Littlestone and Warmuth, 1986) Let D be any probability distribution on a domain X, c be any concept on X, and g be any function mapping sets of at most d examples from X to hypotheses that are subsets of X. Then the probability that $m \geq d$ examples drawn independently at random according to D contain a subset of at most d examples that map via g to a hypothesis that is both consistent with all m examples and has error larger than ϵ is at most $\sum_{i=0}^{d} \binom{m}{i} (1-\epsilon)^{m-i}$.*

Proof: The proof is in the appendix. ■

Using the techniques of Littlestone et al. (1994), the above theorem could be used to obtain bounds on the expected error of a compression scheme. Here we develop sample complexity bounds for pac-learning.

LEMMA 1 *For $0 \leq \epsilon, \delta \leq 1$, if*

$$m \geq \frac{1}{(1-\beta)} \left(\frac{1}{\epsilon} \ln \frac{1}{\delta} + d + \frac{d}{\epsilon} ln \frac{1}{\beta \epsilon} \right)$$

for any $0 < \beta < 1$, then $\sum_{i=0}^{d} \binom{m}{i} (1-\epsilon)^{m-i} \leq \delta$.

Proof: Let

$$\frac{1}{(1-\beta)} \left(\frac{1}{\epsilon} \ln \frac{1}{\delta} + d + \frac{d}{\epsilon} ln \frac{1}{\beta \epsilon} \right) \leq m$$

for $0 < \beta < 1$, which is equivalent to

$$\frac{1}{\epsilon} \ln \frac{1}{\delta} + d + \frac{d}{\epsilon} (1 + \ln \frac{d}{\beta \epsilon} - 1 + \frac{\beta \epsilon}{d} m - \ln d) \leq m. \tag{1}$$

We use the fact from Shawe-Taylor et al. (1989) that

$$-\ln\alpha - 1 + \alpha m \geq \ln m \quad \text{for all} \quad \alpha > 0.$$

For $\alpha = \frac{\beta\epsilon}{d}$ we get

$$ln\frac{d}{\beta\epsilon} - 1 + \frac{\beta\epsilon}{d}m \geq \ln m.$$

By substituting $\ln m$ into the left hand side of equation (1) we get

$$\frac{1}{\epsilon}\ln\frac{1}{\delta} + d + \frac{d}{\epsilon}(1 + \ln m - \ln d) \leq m$$

$$\Leftrightarrow \quad \ln\frac{1}{\delta} + d(1 + \ln m - \ln d) \leq \epsilon(m - d)$$

$$\Leftrightarrow \quad \left(\frac{em}{d}\right)^d \leq e^{\epsilon(m-d)}\delta.$$

Since, from Blumer et al. (1989),

$$\Phi_d(m) \leq \left(\frac{em}{d}\right)^d, \quad \text{for all} \quad m \geq d \geq 1,$$

we have

$$\sum_{i=0}^{d}\binom{m}{i}(1-\epsilon)^{m-i} \leq \Phi_d(m)(1-\epsilon)^{m-d} \leq \left(\frac{em}{d}\right)^d e^{-\epsilon(m-d)} \leq \delta.$$

■

The above lemma leads to sample size bounds that grow linearly in the size of the compression scheme d. The original bounds of Littlestone and Warmuth (1986) had $d\log d$ dependence.

THEOREM 6 *Let $C \subseteq 2^X$ be any concept class with a sample compression scheme of size at most d. Then for $0 < \epsilon, \delta < 1$, the learning algorithm using this scheme learns C with sample size*

$$m \geq \frac{1}{(1-\beta)}\left(\frac{1}{\epsilon}\ln\frac{1}{\delta} + d + \frac{d}{\epsilon}ln\frac{1}{\beta\epsilon}\right)$$

for any $0 < \beta < 1$.

Proof: This follows from Theorem 5 and Lemma 1. ■

Theorem 6 can be used to give simple bounds on sample complexity for pac-learning algorithms based on sample compression schemes. This choice of β can be optimized as done by Cesa-Bianchi et al. (1993) leading to a marginal improvement. From Theorem 6, batch learning algorithms are computationally efficient for any class with a computationally efficient sample compression scheme.

Later in this paper we give a sample compression scheme of size d that applies to all maximum classes of VC dimension d and that produces hypotheses from the class. For maximum classes of VC dimension d Theorem 6 slightly improves the sample complexity of batch learning from the previously known results from Blumer et al. and Shawe-Taylor et al. given in Theorem 2.

Note that the upper bounds have the form $O(\frac{1}{\epsilon}(d \log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$ where d is the size of a compression scheme. For maximum classes, these bounds cannot be improved in that there exist concept classes of VC dimension d for which there are learning algorithms that produce consistent hypotheses from the same class that require sample size $\Omega(\frac{1}{\epsilon}(d \log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$. (This essentially follows from lower bounds on the size of ϵ-nets for concept classes of VC dimension d from Pach and Woeginger (1990) and Littlestone, Haussler, and Warmuth (1994).) Similarly, following Littlestone et al. (1994), one can show that there are concept classes of VC dimension d with a learning algorithm using a compression scheme of size d that requires the same sample size. Ehrenfeucht, Haussler, Kearns, and Valiant (1987) give general lower bounds of $\Omega(\frac{1}{\epsilon}(d + \log \frac{1}{\delta}))$ for learning any concept class of VC dimension d.

For a class with a mistake-bounded on-line learning algorithm with mistake bound k, Littlestone (1989) gives a conversion to a batch pac-learning algorithm with sample complexity at most

$$\frac{1}{\epsilon}(48 \ln \frac{2}{\delta} + 4k + 23 \ln(k+2) - 2)$$

$$= O(\frac{1}{\epsilon} \ln \frac{1}{\delta} + \frac{k}{\epsilon}).$$

This is an improvement over the sample complexity that would result from applying Theorem 6 to sample compression schemes based on mistake-bounded learning algorithms. Note that our One-Pass Compression Scheme based on mistake-bounded learning algorithms can be modified to give "unlabeled" compression sets, where the labels of the examples in the compression set do not have to be saved. It is an open problem whether these "unlabeled" compression schemes result in an improved sample complexity.

In practical algorithms one might want to compress the sample to a small compression set plus some additional information. This motivates the following extension of a sample compression scheme by Littlestone and Warmuth (1986).

Definition. An *extended sample compression scheme of size at most k using b bits* for a concept class C on X consists of a compression function and a reconstruction function. The *compression function* f maps every finite sample set to b bits plus a *compression set*, which is a subset of at most k labeled examples. The *reconstruction function* g maps every possible compression set and b bits to a hypothesis $h \subseteq X$. This hypothesis is not required to be in the class C. For any sample set $Y^{\pm,c}$ labeled consistently with some concept c in C, the hypothesis $g(f(Y^{\pm,c}))$ is required to be consistent with the original sample set $Y^{\pm,c}$.

Theorem 5 can be easily generalized as follows.

THEOREM 7 *(Littlestone and Warmuth, 1986) Let D be any probability distribution on a domain X, c be any concept on X, and g be any function mapping from sets of at most d examples from X plus b bits to hypotheses that are subsets of X. Then the probability that $m \geq d$ examples drawn independently at random according to D contain a subset of at most d examples that combined by some b bits map via g to a hypothesis that is both consistent with all m examples and has error larger than ϵ is at most* $2^b \sum_{i=0}^{d} \binom{m}{i} (1-\epsilon)^{m-i}$.

By generalizing Lemma 1 this leads to the following sample complexity bound.

THEOREM 8 *Let $C \subseteq 2^X$ be any concept class with an extended sample compression scheme of size at most d examples plus b bits. Then for $0 < \epsilon, \delta < 1$, the learning algorithm using this scheme learns C with sample size*

$$m \geq \frac{1}{(1-\beta)} \left(\frac{1}{\epsilon} \ln \frac{1}{\delta} + d + \frac{b}{\epsilon} \ln 2 + \frac{d}{\epsilon} \ln \frac{1}{\beta\epsilon} \right)$$

for any $0 < \beta < 1$.

6. Sample compression schemes for maximum classes

In this section we explore sample compression algorithms based on the VC dimension of a class. Theorem 4 shows that any finite class C has a sample compression scheme of size $\log |C|$. Theorems 1 and 2 suggest that analogs to Theorem 4 should hold when we replace $\log |C|$ with the VC dimension of C. Thus, in order to discuss sample compression schemes for infinite as well as finite classes, we consider sample compression schemes based on the VC dimension of the class.

We first define a maximum class of VC dimension d, and then show that any maximum class of VC dimension d has a sample compression scheme of size d. We further show that this result is optimal, in that for X sufficiently large, there is no sample compression scheme of size less than d for a maximum class $C \subseteq 2^X$ of VC dimension d.

Definition. We use the definitions from Welzl (1987) of maximum and maximal concept classes. A concept class is called *maximal* if adding any concept to the class increases the VC dimension of the class. Let $\Phi_d(m)$ be defined as $\sum_{i=0}^{d} \binom{m}{i}$ for $m \geq d$, and as 2^m for $m < d$. From Vapnik and Chervonenkis (1971) and Sauer (1972), for any class C of VC dimension d on a domain X of cardinality m, the cardinality of C is at most $\Phi_d(m)$.

A concept class C of VC dimension d on X is called *maximum* if, for every finite subset Y of X, $C|Y$ contains $\Phi_d(|Y|)$ concepts on Y. Thus a maximum class C restricted to a finite set Y is of maximum size, given the VC dimension of the class. From Welzl and Woeginger (1987), a concept class that is maximum on a finite domain X is also maximal on that set.

Class D	Class E
w x y z	w x y z
0 0 0 0	0 0 0 1
0 0 1 0	0 0 1 0
0 0 1 1	0 0 1 1
0 1 0 0	0 1 0 0
0 1 0 1	0 1 0 1
0 1 1 0	0 1 1 0
0 1 1 1	0 1 1 1
1 0 0 0	1 0 0 1
1 0 1 0	1 0 1 0
1 0 1 1	1 1 0 0
1 1 0 0	

Figure 2. Class D is maximum. Class E is maximal but not maximum.

Example: maximum and maximal classes. Figure 2 shows two classes, class D and class E, on the set $X = \{w, x, y, z\}$. For each table, each row represents one concept on X. Recall that a concept c in a class can be thought of either as a subset of positive examples from the set X, or as the characteristic function of that subset on X. For example, Figure 2 shows that the null concept, represented by "0000", is in class D but not in class E. Both classes are of VC dimension 2; for example, in both classes the set $\{y, z\}$ is shattered, because both classes contain all four possible concepts restricted to $\{y, z\}$. However, neither class contains a shattered set of size 3. That is, there is no subset of X of size 3 for which either class contains all 8 possible concepts restricted to that class. Class D and class E are both maximal of VC dimension 2, because adding an additional concept to either class would shatter a set of size 3. For example, adding the concept "0000" to the class E would shatter the set $\{x, y, x\}$, increasing the VC dimension to three.

Class D is maximum, because class D is of maximum size on all subsets: every subset of X of size 1 or 2 is shattered, class D contains $\Phi_2(3) = 7$ of the 8 possible concepts on every subset of size 3, and class D contains $\Phi_2(4) = 11$ concepts on X. Class E is not maximum of VC dimension two because it contains less than $\Phi_2(4) = 11$ concepts. Thus, class E in Figure 2 is maximal but not maximum. □

Natural examples of maximum classes that are discussed later in the paper include positive halfspaces, balls in R^n, and positive sets in the plane defined by polynomials of degree at most $n - 1$. More examples of maximal but not maximum classes can be found in Figure 6, and are discussed by Welzl and Woeginger (1987) and Floyd (1989). As Section 7 shows, most maximal classes are in fact not maximum. Nevertheless, we do not know any natural example of a maximal but not maximum class.

6.1. *The sample compression function for maximum classes*

Definition. For $x \in X$, we define $C - x$, the *x-restriction* of C, as $C|(X - \{x\})$, and we define $C^{\{x\}}$, the *x-reduction* of C, as the class $\{c \in C | x \notin c \text{ and } c \cup \{x\} \in C\}$. Both the x-restriction of C and the x-reduction of C have the domain X-$\{x\}$. For each concept c in $C^{\{x\}}$, the two concepts $c \cup \{\langle x, 0\rangle\}$ and $c \cup \{\langle x, 1\rangle\}$ are both in the class C.

As an illustration, consider the maximum class D in Figure 2. The class $C^{\{z\}}$ on $X - \{z\}$ contains four concepts. These concepts, represented as characteristic vectors on $\{w, x, y\}$, are 001, 010, 011, and 101.

Theorem 9 from Welzl shows that we can determine if a class C of VC dimension d is maximum simply from the cardinality of the class. The proof shows that if C is of maximum size on the entire domain X, then C must be of maximum size when restricted to any subset of X.

THEOREM 9 *(Welzl, 1987) A concept class C of VC dimension d on a finite domain X is maximum if and only if* $|C| = \Phi_d(|X|)$.

Proof: By definition, if C is maximum, then $|C| = \Phi_d(|X|)$. We show that if $|C| = \Phi_d(|X|)$, then for every $Y \subseteq X$, $|(C|Y)| = \Phi_d(|Y|)$.

Assume that $|C| = \Phi_d(m)$, for $|X| = m$. Let $x \in X$. By definition, for every concept c in $C^{\{x\}}$ the class C contains two concepts that are consistent with c on X-$\{x\}$; for every concept c in $C - x$ but not in $C^{\{x\}}$, the class C contains one concept that is consistent with c on X-$\{x\}$. Thus $|C| = |C - x| + |C^{\{x\}}|$. The class $C - x$ is of VC dimension at most d on X-$\{x\}$, so $|C - x| \leq \Phi_d(m - 1)$.

The class $C^{\{x\}}$ is of VC dimension at most $d - 1$ on X-$\{x\}$. If some set $Z \subseteq X$ of cardinality d was shattered by the class $C^{\{x\}}$, then the set $Z \cup \{x\}$ would be shattered by the class C, contradicting the fact that C is of VC dimension d. Thus $|C^{\{x\}}| \leq \Phi_{d-1}(m - 1)$.

Because $\Phi_d(m) = \Phi_d(m - 1) + \Phi_{d-1}(m - 1)$, it follows that $|C - x| = \Phi_d(m - 1)$, and that $|C^{\{x\}}| = \Phi_{d-1}(m - 1)$. By induction, for any Y $\subseteq$ X, $|(C|Y)| = \Phi_d(|Y|)$. ■

Corollary 1 from Welzl is the key statement of the underlying structure of maximum classes. Corollary 1 shows that for a finite maximum class C of VC dimension d, the x-restriction $C - x$ and the x-reduction $C^{\{x\}}$ are both maximum classes on X-$\{x\}$. The result is based on the observation that the class $C^{\{x\}}$ is of VC dimension at most $d - 1$, and follows from the counting argument used in Theorem 9.

COROLLARY 1 *(Welzl, 1987) Let* $C \subseteq 2^X$ *be a maximum concept class of VC dimension* $d \geq 1$ *on the finite domain X. Then for any* $x \in X$, $C^{\{x\}}$ *is a maximum class of VC dimension* $d - 1$ *on X-*$\{x\}$. *If* $|X - \{x\}| \geq d$, *then* $C - x$ *is a maximum class of VC dimension d on X-*$\{x\}$.

Proof: Let X be of cardinality m. From the definition of maximum classes, $|C - x| = \Phi_d(m - 1)$. From Theorem 9, if $|X - \{x\}| \geq d$, then $C - x$ is a maximum class

on X-$\{x\}$ of VC dimension d. Similarly, because $|C| = |C - x| + |C^{\{x\}}|$, and $C^{\{x\}}$ is of VC dimension at most $d-1$, $C^{\{x\}}$ is a maximum class of VC dimension $d-1$ on X-$\{x\}$ of size $\Phi_{d-1}(m-1)$. ■

Definition. Let $C \subseteq 2^X$ be a maximum concept class of VC dimension d, and let A = $\{x_1,..., x_k\}$ for A $\subseteq$ X. Then C^A, the *A-reduction* of C, is defined by Welzl (1987) as the class $(((C^{\{x_1\}})^{\{x_2\}})...)^{\{x_k\}}$.

The class C^A consists of all concepts c on $X - A$ such that for any labeling of A, the concept c when extended by that labeling remains in the class C. Thus, for each concept in C^A, there are $2^{|A|}$ related concepts in C. From Welzl (1987), for any distinct x and y in X, $(C^{\{x\}})^{\{y\}} = (C^{\{y\}})^{\{x\}}$. Therefore for any $A \subseteq X$, the class C^A is well-defined.

COROLLARY 2 *(Welzl, 1987) Let $C \subseteq 2^X$ be a maximum concept class of VC dimension d on the finite domain X. Let A be any subset of X of cardinality d. Then the class C^A is of VC dimension 0, and thus consists of a single concept.*

Proof: This follows from repeated application of Corollary 1. The class C^A contains the single concept c on X-A such that, for any labeling $A^{\pm}$ of the elements of A, $c \cup A^{\pm}$ remains a concept in C. ■

Definition. For any maximum concept class $C \subseteq 2^X$ of VC dimension d on the finite domain X, and for any set A $\subseteq$ X of cardinality d, c_A, the *expansion* of A, is defined by Welzl (1987) as the unique concept in C^A, the A-reduction of C, on the domain X-A.

Example: at most two positive examples. As an example, consider the maximum class C of VC dimension two on X that consists of all concepts with at most two positive examples. Then, for $\{x_1, x_2\} \subseteq$ X, $c_{\{x_1,x_2\}}$ denotes the concept on $X - \{x_1, x_2\}$ where every example is a negative example. This is the only concept on $X - \{x_1, x_2\}$ that remains a concept in C if both x_1 and x_2 are positive examples. □

Example: intervals on the line. Let C_n be the class containing all unions of at most n positive intervals on the line. This class is maximum of VC dimension $2n$. This follows because for any finite set of m points on the line, for $m \geq 2n$, there are $\sum_{i=0}^{2n} \binom{m}{i}$ ways to label those m points consistent with at most n positive intervals on the line. For C_3, let A be the set of 6 points $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ shown below. Figure 3 shows the unique labeling of the rest of the line for the concept c_A. For any labeling of the points in A, the resulting labeling of the entire line corresponds to some concept in C_3. □

Definition. For any maximum concept class C of VC dimension d on the finite domain X, and for any $A \subseteq X$ of cardinality d, there is a corresponding concept c_A on the set X-A. For the labeled set $A^{\pm}$, let $c_{A^{\pm}}$, the *expansion* of $A^{\pm}$, denote the concept on X with X-A labeled as in the concept c_A, and with A labeled as in $A^{\pm}$. Thus for every labeled

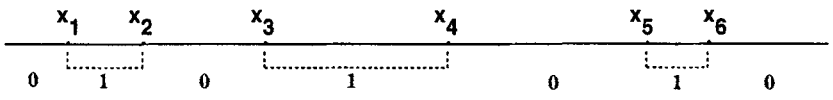

Figure 3. A compression set for the union of three intervals on the line.

set $A^{\pm}$ of cardinality d, for A $\subseteq$ X, there is a corresponding concept $c_{A^{\pm}}$ on X. We say that the set $A^{\pm}$ is a *compression set* for the concept $c_{A^{\pm}}$, and that the set $A^{\pm}$ *represents* the concept $c_{A^{\pm}}$. Thus every set of d labeled examples from the domain X represents a concept from the maximum class C. Speaking loosely, we say that a compression set $A^{\pm}$ *predicts labels* for the elements in X, according to the concept $c_{A^{\pm}}$.

Example: positive halfspaces. Let C_n be the class containing all positive half-spaces in R^n. That is, C_n contains all n-tuples $(y_1, \ldots, y_n)$ such that $y_n > a_{n-1}y_{n-1} + \ldots + a_1y_1 + a_0$, for $y_1, .., y_n \in X$. (Note that for positive half-spaces the coefficient before y_n is always $+1$.) Let the finite sample set $X \subset R^n$ contain at most n examples on any hyperplane. Then from Floyd (1989), the class C_n is maximum of VC dimension n on X. Let $A \subset X$ be a set of n examples in R^n. The concept c_A on $X - A$ represents the positive halfspace defined by the unique hyperplane determined by the set A. For any labeling $A^{\pm}$ of A, the concept $c_{A^{\pm}}$ represented by the compression set $A^{\pm}$ is consistent with some positive half-space in R^n. The compression set A essentially defines the boundary between the positive and negative examples. This gives a computationally-efficient compression scheme of size n for positive half-spaces in R^n.

This application of compression schemes can be generalized to any class of positive sets defined by an n-dimensional vector space of real functions on some set X. This generalization includes classes such as balls in R^{n-1} and positive sets in the plane defined by polynomials of degree at most $n - 1$. □

Lemma 2 shows that $c_{A^{\pm}}$ in the class $C|Y$ is identical to the restriction to the set Y of $c_{A^{\pm}}$ in the class C. This lemma is necessary to show that there is a unique definition of the concept $c_{A^{\pm}}$ for infinite classes.

LEMMA 2 *Let $C \subseteq 2^X$ be a maximum class of VC dimension d, for X finite. Let $A \subseteq Y \subseteq X$, for $|A| = d$. Then for any labeling $A^{\pm}$ of A, and for $x \in Y$, $c_{A^{\pm}}$ in the class C and $c_{A^{\pm}}$ in the class $C|Y$ assign the same label to the element x.*

Proof: If $x \in A$, then for both $c_{A^{\pm}}$ in the class C and $c_{A^{\pm}}$ in the class $C|Y$, x is labeled as in $A^{\pm}$. If $x \notin A$, then assume for purposes of contradiction that Lemma 2 is false. Without loss of generality, assume that $c_{A^{\pm}}$ in the class C contains $\langle x, 0\rangle$, and that $c_{A^{\pm}}$ in the class $C|Y$ contains $\langle x, 1\rangle$. Because $c_{A^{\pm}}$ in the class C contains $\langle x, 0\rangle$, then for every labeling of A, the class C contains a concept with that labeling, and with $\langle x, 0\rangle$. Because $c_{A^{\pm}}$ in the class $C|Y$ contains $\langle x, 1\rangle$, then for every labeling of A, the class $C|Y$ contains a concept with that labeling, and with $\langle x, 1\rangle$. Then the set $A \cup \{x\}$ is shattered in the class C, contradicting the fact that C is of VC dimension d. ■

Definition. The concept c_A for an infinite set X. For the infinite set X, for A⊆B⊆X, for |A|=d, and for B finite, we define the concept c_A for the class C as identical, when restricted to the finite set B-A, to the concept c_A in the finite class $C|B$. The concept c_A in the class C assigns a unique label to each element $x \in$X-A; if not then there would be two finite subsets B_1 and B_2 of X containing x, such that c_A in $C|B_1$ and c_A in $C|B_2$ assigned different labels to the element x, contradicting Lemma 2.

Example: two-dimensional positive halfspaces. Consider the class of positive half-spaces in R^2, given an infinite domain $X \subset R^2$ with at most two collinear points. This class is maximum of VC dimension 2. The sample set shown in Figure 4 can be represented by the compression set $A^{\pm} = \{\langle x_1, 1\rangle, \langle x_2, 0\rangle\}$. For any finite subset of X, the concept represented by $c_{A^{\pm}}$ is consistent with some positive halfspace on X. □

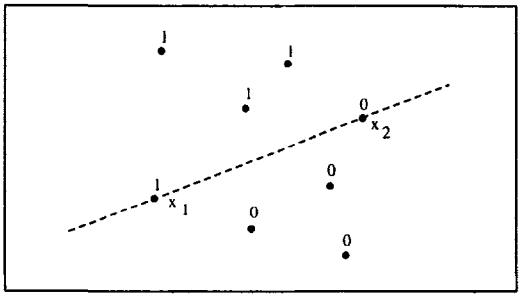

Figure 4. A compression set for positive halfspaces in the plane.

Theorem 10 is the main technical result used to show that every maximum class C of VC dimension d has a sample compression scheme of size d. In particular, Theorem 10 shows that for a maximum class C of VC dimension d, given a finite domain X, every concept in the class is represented by some labeled set $A^{\pm}$ of cardinality d. Theorem 10 is also stated without proof by Welzl (1987).

THEOREM 10 *Let $C \subseteq 2^X$ be a maximum concept class of VC dimension d on a finite domain X, for $|X| = m \geq d$. Then for each concept $c \in C$, there is a compression set $A^{\pm}$ of exactly d elements, for $A^{\pm} \subseteq X \times \{0,1\}$, such that $c = c_{A^{\pm}}$.*

Proof: The proof is by double induction on d and m. The first base case is for $m = d$ for any $d \geq 0$. In this case, we save the complete set $X^{\pm}$ of d elements.

The second base case is for $d = 0$, for any m. In this case there is a single concept in the concept class, and this concept is represented by the empty set.

Induction step: We prove that the theorem holds for d and m, for $d > 0$ and $m > d$. By the induction hypothesis, the theorem holds for all d' and m' such that $d' \leq d$, $m' \leq m$, and $d' + m' < d + m$. Let $X = \{x_1, x_2, ..., x_m\}$. There are two cases to consider.

Case 1: Let c be a concept in $C - x_m$ such that $c \cup \{\langle x_m, 0\rangle\}$ and $c \cup \{\langle x_m, 1\rangle\}$ are not both in C. Without loss of generality, assume that only $c \cup \{\langle x_m, 0\rangle\}$ is in C.

From Corollary 1, $C - x_m$ is maximum of VC dimension d. Thus by the induction hypothesis, each concept c in $C - x_m$ can be represented by a compression set $A^{\pm}$ of d labeled elements, for $A \subseteq X - x_m$, with c identical to $c_{A^{\pm}}$ in the class $C - x_m$. From Corollary 2, $A^{\pm}$ represents some concept $c_{A^{\pm}}$ on X in the class C. From Lemma 2, $c_{A^{\pm}}$ in the class $C - x_m$ agrees on $X - x_m$ with $c_{A^{\pm}}$ in the class C . If $c_{A^{\pm}}$ in the class C contains $\langle x_m, 1\rangle$, then $c \cup \{\langle x_m, 1\rangle\}$ is in C, violating the assumption for Case 1. Thus $c_{A^{\pm}}$ in the class C contains $\langle x_m, 0\rangle$, and case 1 is done.

Case 2: Let c be a concept in $C - x_m$ such that $c \cup \{\langle x_m, 0\rangle\}$ and $c \cup \{\langle x_m, 1\rangle\}$ are both in C. Thus $c \in C^{\{x_m\}}$. From Corollary 1, $C^{\{x_m\}}$ is a maximum class of VC dimension $d - 1$ on $X - x_m$. By the induction hypothesis, there is a compression set $B^{\pm}$ of $d - 1$ elements of $X - x_m$, such that c is identical to $c_{B^{\pm}}$ in the class $C^{\{x_m\}}$.

Let $c_1 = c \cup \{\langle x_m, 0\rangle\}$. Let $A^{\pm} = B^{\pm} \cup \{\langle x_m, 0\rangle\}$. From Corollary 2, the labeled set $A^{\pm}$ of cardinality d represents a unique concept $c_{A^{\pm}}$ in C.

We show that $c_{A^{\pm}}$ in the class C and $c_{B^{\pm}}$ in the class $C^{\{x_m\}}$ assign the same labels to all elements of $X - x_m$. Assume not, for purposes of contradiction. Then there is some element x_i of $(X - x_m) - B$ such that x_i is assigned one label l_i in $c_{A^{\pm}}$ in the class C, and another label $\overline{l_i}$ in $c_{B^{\pm}}$ in the class $C^{\{x_m\}}$. Because $c_{A^{\pm}}$ in the class C contains $\langle x_i, l_i\rangle$, then for each of the 2^d labelings of A, and for $\langle x_i, l_i\rangle$, there is a concept consistent with that labeling in C. Because $c_{B^{\pm}}$ in the class $C^{\{x_m\}}$ contains $\langle x_i, \overline{l_i}\rangle$, then for each of the 2^{d-1} labelings of B, and for $\langle x_i, \overline{l_i}\rangle$, there is a concept consistent with that labeling in $C^{\{x_m\}}$. For each concept in $C^{\{x_m\}}$, there is a concept in C with $\langle x_m, 0\rangle$, and another concept in C with $\langle x_m, 1\rangle$. Thus the $d+1$ elements in A $\cup\{x_i\}$ are shattered by the concept class C. This contradicts the fact that the class C is of VC dimension d. Thus the set $A^{\pm}$ is a compression set for the concept $c \cup\{\langle x_m, 0\rangle\}$, and case 2 is done. ■

Note that for a concept $c \in C$, there might be more than one compression set $A^{\pm}$ such that c is identical to $c_{A^{\pm}}$.

Theorem 11 extends Theorem 10 to give a compression scheme for any maximum class of VC dimension d. Let $C \subseteq 2^X$ be any maximum class of VC dimension d. The input to the sample compression scheme is any labeled sample set $Y^{\pm,c}$ of size at least d, for $Y \subseteq X$. The examples in $Y^{\pm,c}$ are assumed to be labeled consistently with some concept c in C.

The VC Compression Scheme (for maximum classes).

- The compression function: The input is a sample set $Y^{\pm,c}$ of cardinality at least d, labeled consistently with some concept c in C. The finite class $C|Y$ is maximum of VC dimension d. From Theorem 10, in the class $C|Y$ there is a compression set $A^{\pm}$ of exactly d elements, for $A^{\pm} \subseteq Y^{\pm,c}$, such that the concept $c|Y$ is represented by the compression set $A^{\pm}$. The compression function chooses some such set $A^{\pm}$ as the compression set.

- The reconstruction function: The input is a compression set $A^{\pm}$ of cardinality d. For an element $x \in A$, the reconstruction function predicts the label for x in the set $A^{\pm}$.

For $x \notin A$, let C_1 be the class C restricted to $A \cup \{x\}$. C_1 is a maximum class of VC dimension d on $A \cup \{x\}$. Let c_A in C_1 predict $\langle x, l \rangle$, for $l \in \{0, 1\}$. Then the reconstruction function predicts the label 'l' for x.

Note that in the VC Compression Scheme sample sets of size at least d are compressed to subsets of size equal to d.

THEOREM 11 *Let $C \subseteq 2^X$ be a maximum class of VC dimension d on the (possibly infinite) domain X. Then the VC Compression Scheme is a sample compression scheme of size d for C.*

Proof: Let the input to the sample compression scheme be a finite labeled sample set $Y^{\pm,c}$ of size at least d. The compression function saves some labeled set $A^{\pm}$ of cardinality d, for $A^{\pm} \subseteq Y^{\pm,c}$, such that $c_{A^{\pm}}$ in the class $C|Y$ is consistent with the sample set. The reconstruction function gives as a hypothesis the concept $c_{A^{\pm}}$ in the class C.

From Theorem 9, $C|Y$ is a maximum class of VC dimension d. Thus, by Theorem 10, there exists a subset $A^{\pm}$ of $Y^{\pm,c}$, for $|A^{\pm}| = d$, such that $c_{A^{\pm}}$ in the class $C|Y$ is consistent with the sample set. From Lemma 2, the concept $c_{A^{\pm}}$ in the class C is consistent with $c_{A^{\pm}}$ in the class $C|Y$ on the original sample set Y. Thus we have a sample compression scheme of size d for maximum classes of VC dimension d. ■

6.2. An algorithm for the compression function

This section gives a Greedy Compression Algorithm that implements the VC Compression Scheme for a maximum class C of VC dimension d on the (possibly infinite) domain X. The proof of Theorem 10 suggests an algorithm to find the compression set. The input for the compression algorithm is a finite sample set $Y^{\pm,c}$ of size at least d, labeled consistently with some concept c in C. The output is a labeled compression set $A^{\pm} \subseteq Y^{\pm,c}$ of cardinality d that represents some concept in C consistent with the sample set. The Greedy Compression Algorithm uses the *consistency oracle* defined below. The more efficient Group Compression Algorithm, described later in this section, requires the more powerful *group consistency oracle*, also defined in this section.

Definition. The *consistency problem* for a particular concept class C is defined in Blumer et al. (1989) as the problem of determining whether there is a concept in C consistent with a particular set of labeled examples on X. We define a *consistency oracle* as a procedure for deciding the consistency problem.

The Greedy Compression Algorithm (for the VC Compression Scheme).

- The compression algorithm: The input is the finite sample set

$$Y^{\pm,c} = \{\langle x_1, l_1 \rangle, .., \langle x_m, l_m \rangle\},$$

labeled consistently with some concept c in C. Initially the compression set $A^{\pm}$ is the empty set. The compression algorithm examines each element of the sample set in arbitrary order, deciding whether to add each element in turn to the compression set $A^{\pm}$. At step i, the algorithm decides whether to add the labeled element $\langle x_i, l_i \rangle$ to the partial compression set $A^{\pm}$, for $0 \leq |A| \leq d-1$.

The algorithm determines whether, for each possible labeling of the elements in $A \cup \{x_i\}$, there exists a concept in $C|Y$ consistent with that labeling along with the labeling of other elements given in the sample set. If so, then $\langle x_i, l_i \rangle$ is added to $A^{\pm}$. The compression algorithm terminates when $A^{\pm}$ is of cardinality d.

- The reconstruction algorithm: The input is a compression set $A^{\pm}$ of cardinality d, and the reconstruction function is asked to predict the label for some element $x \in X$. If $x \in A$, then the reconstruction algorithm predicts the label for x in $A^{\pm}$. If $x \notin A$, let C_1 be $C|(A \cup \{x\})$. If, for each of the 2^d possible labelings $A^{\pm}$ of A, there is a concept in C_1 consistent with $A^{\pm} \cup \langle x, 0 \rangle$, then $c_{A^{\pm},C_1}$ predicts label '0' for the element x. Otherwise, $c_{A^{\pm},C_1}$ predicts the label '1' for x.

Example: intervals on the line. Consider the Greedy Compression Algorithm applied to a finite sample set from the class C_3 of at most 3 intervals on the line, as in Figure 1. The examples in Figure 1 are labeled consistently with some concept c in C_3. Consider the examples one at a time, starting with the leftmost example. Let the initial compression set $A^{\pm}$ be the empty set. First consider the example "x_1". There is no concept in C_3 with $\langle x_1, 1 \rangle$, and with the other examples labeled as in Figure 1. Therefore the example "x_1" is not added to the current compression set. There is a concept in C_3 with $\langle x_2, 1 \rangle$, and with the other examples labeled as in Figure 1. Therefore, $\langle x_2, 0 \rangle$ is added to the current compression set $A^{\pm}$. Now consider the element "x_3". For every labeling of the elements $\{x_2, x_3\}$, is there a concept in C_3 consistent with that labeling, and with the labeling of the other points in the sample set? No, because there is no concept in C_3 with $\langle x_2, 1 \rangle$, $\langle x_3, 0 \rangle$, and with the given labeling of the other points. Therefore 'x_3' is not added to the current compression set. Proceeding in this fashion, the Greedy Compression Algorithm constructs the compression set A = $\{\langle x_2, 0 \rangle, \langle x_4, 1 \rangle, \langle x_6, 0 \rangle, \langle x_{10}, 1 \rangle, \langle x_{13}, 0 \rangle, \langle x_{15}, 1 \rangle\}$. The reconstruction function for this class is illustrated by Figure 3. □

Theorem 12 shows that the Greedy Compression Algorithm terminates with a correct compression set of size d.

THEOREM 12 *Let $C \subseteq 2^X$ be a maximum class of VC dimension d, and let $Y^{\pm,c}$ be a finite sample set labeled consistently with some concept $c \in C$, for $|Y^{\pm,c}| \geq d$. Then the Greedy Compression Algorithm after each step maintains the invariant that, for the partial compression set $A^{\pm}$, the labeled set $(Y - A)^{\pm,c}$ is consistent with some concept in C^A. Furthermore, the Greedy Compression Algorithm on $Y^{\pm,c}$ terminates with a compression set of cardinality d for the concept c.*

Proof: From the algorithm it follows immediately that at each step the invariant is maintained that the labeled set $(Y - A)^{\pm,c}$ is consistent with some concept in C^A.

Assume for purposes of contradiction that the Greedy Compression Algorithm ends with the compression set $A^{\pm}$, where $|A^{\pm}| = s < d$. Then the labeled sample set $Y^{\pm,c}$ is consistent with some concept on Y-A in C^A. From Corollary 1, C^A is a maximum class of VC dimension $d - s$ on Y-A. From Theorem 10, there is a compression set of cardinality $d - s$ from $(Y - A)^{\pm,c}$ for $c|(Y - A)$ in the class C^A. Let x_j be a member of some such compression set of cardinality $d - s$. Then $(C^A)^{\{x_j\}}$ is a maximum class of VC dimension $d - s - 1$ on (Y-A)-$\{x_j\}$ that contains a concept consistent with c. Let $A_1 \subseteq A$ denote the partial compression set held by the compression algorithm before the compression algorithm decides whether or not to add the element x_j. Then $(C^{A_1})^{\{x_j\}}$ contains a concept consistent with c. Therefore x_j would have been included in the partial compression set. This contradicts the fact that $x_j \notin A$. Therefore the compression algorithm can not terminate with a compression set of cardinality $s < d$. ■

The compression algorithm from the Greedy Compression Algorithm requires at most $(m-d)2^{d-1}+2^d-1$ calls to the consistency oracle for C. This upper bound holds because the d elements added to the compression set require at most $2^0 + 2^1 + ... + 2^{d-1} = 2^d - 1$ calls to the consistency oracle, and each other element requires at most 2^{d-1} calls to the consistency oracle. When the compression set A_i contains i examples, and the compression algorithm considers whether or not to add the next example $\langle x_j, l_j \rangle$ to the compression set, it is already known that the sample set on $Y - A_i - \{x_j\}$ is consistent with $\langle x_j, l_j \rangle$ and each possible labeling of the elements in A. Thus the compression algorithm has to make at most 2^i calls to the consistency oracle to decide whether or not to add x_j to the compression set.

To predict the label for x_i, the reconstruction function needs to make at most 2^d calls to the consistency oracle.

Note that if the consistency problem for C is not efficiently computable, then we are not likely to find computationally-efficient algorithms for the sample compression scheme. From Pitt and Valiant (1988) and Blumer et al. (1989), if the consistency problem for C is NP-hard and $\mathbf{RP} \neq \mathbf{NP}$,[4] then C is not polynomially learnable by an algorithm that produces hypotheses from C.

A more efficient algorithm for the VC Compression Scheme is based on the more powerful *group consistency oracle* defined below.

Definition. We define a *group consistency oracle* for the maximum class C of VC dimension d as a procedure that, given as input a set $Y^{\pm}$ of labeled examples and a set $A \subseteq Y$, can determine whether the examples in $Y^{\pm}$ are labeled consistently with some concept in the class C^A. We say that a family of classes $C_n \subseteq 2^{X_n}, n \geq 1$, has a *polynomial-time group consistency oracle* if there is a group consistency oracle that runs in time polynomial in the sample size and in the VC dimension of the class.

Example: intervals on the line. Let C be the maximum class C of VC dimension $2n$ of at most n intervals on the line. Given a set A, a poly-time group consistency oracle needs to determine if a particular labeled set $Y^{\pm}$ of points is consistent with some concept in C^A. The group consistency algorithm assigns that labeling $A^{\pm}$ to the points in A that gives the

maximum number of intervals for the set of points $(Y \cup A)^{\pm}$. The labeled set $(Y - A)^{\pm}$ is in C^A if and only if the maximum number of positive intervals for $(Y \cup A)^{\pm}$ is at most n. □

The Group Compression Algorithm (for the VC Compression Scheme).

- The compression algorithm: The elements of the sample $Y^{\pm,c}$ are examined one at a time, as in the earlier Greedy Compression Algorithm. Initially, the compression set A is empty. At step i, to determine whether to add $\langle x_i, l_i \rangle$ to the set $A^{\pm}$, the compression algorithm calls the group-consistency oracle to determine if for every labeling of A $\cup \{x_i\}$, there is a concept in C consistent with that labeling, and with the labeling of the other elements of $Y^{\pm,c}$. If so, $\langle x_i, l_i \rangle$ is added to the compression set $A^{\pm}$.
- The reconstruction algorithm: The input is a compression set $A^{\pm}$ of cardinality d, and the hypothesis is the concept $c_{A^{\pm}}$. The group consistency oracle is used to predict the label for an element $x \in X - A$.

The group compression algorithm requires at most m calls to the group consistency oracle, for a sample set of size m. The reconstruction algorithm can predict the label for x_i in $c_{A^{\pm}}$ with one call to the group consistency oracle. If the family of classes $C_n \subseteq 2_n^X, n \geq 1$, has a polynomial-time group consistency oracle, then the sample compression scheme for C_n can be implemented in time polynomial in the VC dimension and the sample size.

6.3. Infinite maximum classes are not necessarily maximal

This section discusses infinite maximum classes that are not maximal. For a maximum but not maximal class C of VC dimension d on an infinite domain X, and for some compression set A of size d, the concept $c_{A^{\pm}}$ is not necessarily a member of the class C.

Corollary 1 from Welzl showed that for a finite maximum class C, both the x-restriction $C - x$ and the x-reduction $C^{\{x\}}$ are maximum classes on $X - \{x\}$. In this section we show that for an infinite maximum class, the x-restriction $C - x$ is a maximum class on $X - \{x\}$, but the x-reduction $C^{\{x\}}$ is not necessarily a maximum class. Floyd (1989, p. 25) shows that Corollary 1 can be extended to any maximum and also maximal class on an infinite domain X.

Corollary 3 below shows that for an infinite maximum class, the x-restriction $C - x$ is also a maximum class. From Floyd (1989), the proof follows directly from the definition of a maximum class as of maximum size on every finite subset of X.

COROLLARY 3 *Let $C \subseteq 2^X$ be a maximum concept class of VC dimension d on the infinite domain X. Then for any $x \in X$, $C - x$ is a maximum class of VC dimension d on X-$\{x\}$.*

Proof: The proof follows directly from the definition of a maximum class. ■

Any maximum class C on a finite domain X is also a maximal class. However Welzl and Woeginger (1987) show that for an infinite domain X for any $d \geq 1$ there are concept classes of VC dimension d that are maximum but not maximal. This occurs because a maximum class C is defined only as being maximum, and therefore maximal, on finite subsets of X. A maximum class C on X is not required to be maximal on the infinite domain X.

Example: a maximum class that is not maximal. Consider the maximum class C of VC dimension 1 on an infinite domain X where each concept contains exactly one positive example. This class is not maximal, because the concept with no positive examples could be added to C without increasing the VC dimension of the class. However, the class C is maximum, because it is of maximum size on every finite subset of X. For this class, for $x \in$X, $C^{\{x\}}$ is the empty set. From the definition above, the concept $c_{\{x\}}$ is defined by its value on finite subsets of X. Thus $c_{\{x\}}$ is defined as the concept with all negative examples on X-$\{x\}$, even though $c_{\{x\}} \cup \langle x, 0 \rangle$ is not a concept in C. □

6.4. *A lower bound on the size of a sample compression scheme*

In this section we show that for a maximum class $C \subseteq 2^X$ of VC dimension d for X sufficiently large, there is no sample compression scheme of size less than d. This shows that the sample compression schemes of size at most d given above for maximum classes of VC dimension d are optimal, for X sufficiently large. We also show that for an arbitrary concept class $C \subseteq 2^X$ of VC dimension d (not necessarily maximum), there is no sample compression scheme of size at most $d/5$.

These results refers to a sample compression scheme as defined in Section 3, where a sample compression set consists of an unordered, labeled subset from the original sample set. Theorem 13 follows from a counting argument comparing the number of labeled compression sets of size at most $d-1$ with the number of different labelings of a sample of size m that are consistent with some concept in C.

THEOREM 13 *For any maximum concept class $C \subseteq 2^X$ of VC dimension $d > 0$, there is no sample compression scheme of size less than d for sample sets of size at least $d^2 2^{d-1}$.*

Proof: Let Y be any subset of X of cardinality $m \geq d^2 2^{d-1}$. The class $C|Y$ contains $\Phi_d(m)$ concepts. We show that there are less than $\Phi_d(m)$ labeled compression sets of size at most $d-1$ from Y. For each set of i elements in a compression set, for $0 \leq i \leq d-1$, those elements could be labeled in 2^i different ways. Therefore there are at most

$$\sum_{i=0}^{d-1} 2^i \binom{m}{i}$$

distinct labeled compression sets of size at most $d-1$ from Y.

We show that

$$\sum_{i=0}^{d-1} 2^i \binom{m}{i} < \sum_{i=0}^{d} \binom{m}{i} = \Phi_d(m)$$

$$\Leftrightarrow \quad \sum_{i=0}^{d-1} (2^i - 1) \binom{m}{i} < \binom{m}{d}.$$

It suffices to show that

$$d(2^{d-1} - 1) \binom{m}{d-1} < \binom{m}{d} = \binom{m}{d-1} \frac{m-d+1}{d}.$$

This is equivalent to showing that

$$d^2 2^{d-1} - d^2 + d - 1 < m.$$

This inequality holds because $m \geq d^2 2^{d-1}$. ■

Theorem 13 shows that for any infinite maximum class of VC dimension d, there is no sample compression scheme of size less than d. Theorem 13 is also likely to apply to most finite maximum classes of practical interest. For classes on finite domains with n attributes, each at least two-valued, the size of the domain X is at least 2^n. Typically the task is to learn polynomial-sized rules of some type, so classes of practical interest are likely to to have a VC dimension that is at most polynomial in n, giving a sample space that is exponential in the VC dimension.

Note that the argument in Theorem 13 does not necessarily apply for classes that are not maximum. For example, consider the class C of arbitrary halfspaces in the plane, on a set X with no three collinear points. From Blumer and Littlestone (1989), the class C has VC dimension three, but there exists a sample compression scheme of size two for this class. Class C is neither maximum nor maximal; for some sets of four points in the plane there are less than $\Phi_3(4) = 15$ ways to label those four points consistently with some arbitrary halfspace. Theorem 14 below gives a lower bound for the size of a sample compression scheme for an arbitrary concept class C of VC dimension d.

THEOREM 14 *For an arbitrary concept class C of VC dimension d, there is no sample compression scheme of size at most $d/5$ for sample sets of size at least d.*

Proof: Let Y be any set of d unlabeled examples. There are at most

$$\sum_{i=0}^{d/5} \binom{d}{i} 2^i \leq \Phi_{d/5}(d) 2^{d/5}$$

compression sets of size at most $d/5$ from Y. Since from Blumer at al. (1989)

$$\Phi_k(m) \leq \left(\frac{em}{k}\right)^k \quad \text{for all} \quad m \geq k \geq 1,$$

the number of compression sets is bounded above by

$$(10e)^{d/5} < 32^{d/5} = 2^d.$$

Thus if Y is shattered by the class C, then there are not enough compression sets for the 2^d labelings of Y. ■

7. Maximal classes

In this section we discuss randomly-generated maximal classes. Although we have no natural example of a maximal class that is not also maximum, we show that the vast majority of randomly-generated maximal classes are not maximum. It is an open question whether there is a sample compression scheme of size d for every maximal class of VC dimension d. In this section we describe a sample compression scheme that is a modification of the VC Compression Scheme and that applies for *some* classes that are maximal but not maximum.

Every class C of VC dimension d on a finite set X can be embedded in a maximal class of VC dimension d: simply keep adding concepts to the class C until there are no more concepts that can be added without increasing the VC dimension. From Welzl and Woeginger (1987), every maximal class of VC dimension 1 is also a maximum class, but for classes of VC dimension greater than 1, there are maximal classes that are not maximum. Figures 2 and 6 both show classes of VC dimension 2 that are maximal but not maximum.

7.1. Randomly-generated maximal classes

This section defines a *randomly-generated* maximal class of VC dimension d on a finite domain X. We show that for VC dimensions 2 and 3, a large number of randomly-generated maximal classes are not maximum. There are many natural examples of maximum classes, but in spite of the abundance of classes that are maximal but not maximum, we are not aware of a natural example from the literature of a class that is maximal but not maximum.

We define a randomly-generated maximal class by the following procedure for randomly generating such classes.

Procedure for generating a random maximal class of VC dimension d.

1. For a maximal class of VC dimension d on a set of m elements, there are 2^m possible concepts on these m elements. Each possible concept is classified as a member of the class C, not a member of C, or undecided. Initially, the status of each possible concept is undecided. At each step, the program independently and uniformly selects one of the undecided concepts c. Step 2 is repeated for each selected undecided concept.

2. If the undecided concept c can be added to the class C without increasing the VC dimension to $d+1$, then the concept c becomes a member of the class C. Otherwise, the concept c is not a member of the class C.

After the status of all 2^m possible concepts has been decided, the resulting class C is a maximal class of VC dimension d. No additional concepts can be added to the class without increasing the VC dimension of the class to $d+1$. Because the procedure for randomly generating a maximal class examines all 2^m possible concepts, the procedure can only be run for small values of m. Our program uses a pseudo-random number generator to select undecided concepts.

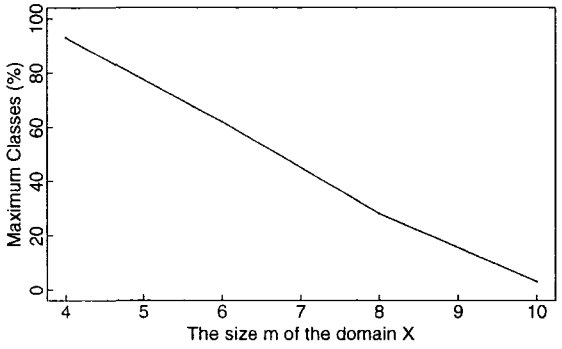

Figure 5. Randomly-generated maximal classes of VC dimension 2.

From Theorem 9, a program can determine if a given class $C \subseteq 2^X$ of VC dimension d is a maximum class simply by counting the number of concepts in the class. Figure 5 shows the percent of randomly-generated maximal classes of VC dimension 2 that are also maximum, from our experiments. The x-axis shows the size m of domain X; the y-axis shows the percent of the randomly-generated maximal classes that are maximum. For each value of $m \in \{4, 6, 8, 10\}$, our program created 100 randomly-generated maximal classes of VC dimension 2 on m elements. From Figure 5, as m increases, the percent of randomly-generated maximal classes that are also maximum decreases sharply. For maximal classes of VC dimension 3, none of the 100 randomly-generated classes on 6 or 8 elements were maximum. These results suggest that for m and d sufficiently large, few maximal classes of VC dimension d on m elements will be maximum.

7.2. Compression schemes for maximal classes: an open question

The VC Compression Scheme described in Section 6 applies to maximum classes of VC dimension d; it can not necessarily be applied to maximal but not maximum classes of VC dimension d. For example, Figure 6 shows a maximal class of VC dimension 2 for which the VC Compression Scheme does not apply. A key open question is whether there

is a compression scheme of size O(d) for every maximal class of VC dimension d. This section presents a modified version of the VC Compression Scheme, called the Subset Compression Scheme, that applies for *some* maximal classes of VC dimension d. It is an open question whether the Subset Compression Scheme gives a sample compression scheme of size d for *all* maximal classes of VC dimension d.

Class C

x_1	x_2	x_3	x_4
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0

Figure 6. A maximal but not maximum class C of VC dimension 2.

First, we show that the VC Compression Scheme given above for maximum classes does not apply to some maximal but not maximum classes. For example, class C in Figure 6 is maximal but not maximum, of VC dimension 2. For concept $c = 1100$ in C there is no compression set of size two using the VC Compression Scheme. That is, there is no A$\subseteq X$, for |A|=2, such that c restricted to $X - A$ is in the class C^A.

The Subset Compression Scheme defined below gives a sample compression scheme of size 2 for the class C in Figure 6. Note that no algorithm is given for the compression function in the Subset Compression Scheme, other than checking each possible set of size at most d to see if it suffices as a compression set.

The Subset Compression Scheme (for some maximal classes).

- The compression function: Let $C \subseteq 2^X$ be a maximal class of VC dimension d. The compression function is given as input the sample set $Y^{\pm,c}$, labeled consistently with some concept c in C, for $Y \subseteq X$. The compression function finds a subset $A^{\pm,c} \subseteq Y^{\pm,c}$, for $|A^{\pm}| = d$, that represents the concept c on Y using the reconstruction scheme below.

- The reconstruction function: The reconstruction function is given as input the compression set $A^{\pm}$, of cardinality d. For $x_i \notin A$, the label that the compression set $A^{\pm}$ predicts for x_i is determined by considering the class $C_i = C|(A \cup \{x_i\})$, of VC dimension at most d. The class $(C_i)^A$ is of VC dimension at most 0, and is either empty or contains exactly one concept. If $(C_i)^A$ is nonempty, then $(C_i)^A$ contains a single concept $\langle x_i, l_i \rangle$, for $l_i \in \{0, 1\}$. In this case, the compression set $A^{\pm}$ predicts

the label l_i for x_i. (This part of the reconstruction function is identical to that in the VC Compression Scheme.)

If $(C_i)^A$ is empty. then let the label predicted by $A^{\pm}$ for x_i depend on the labels of the elements in the compression set $A^{\pm}$. If there is only one possible label for x_i in concepts in the class C, given the labels of the elements in $A^{\pm}$, then that is the label predicted by the compression set $A^{\pm}$. Otherwise, arbitrarily let the compression set $A^{\pm}$ predict the label '0' for x_i. With this definition, each compression set $A^{\pm}$ predicts a unique label for each element x_i of X, and therefore a unique hypothesis on X. This hypothesis is not necessarily in the class C.

For a maximum class of VC dimension d, the Subset Compression Scheme is identical to the VC Compression Scheme. For a maximal class let $c_{A^{\pm}}$ denote the concept on X represented by the compression set $A^{\pm}$ using the Subset Compression Scheme. The Subset Compression Scheme is motivated by a combinatorial characterization of maximal classes of VC dimension d given by Floyd (1989) by "forbidden labels" on subsets of $d+1$ elements. It is an open question whether the subset compression scheme gives a sample compression scheme of size d for every maximal class of VC dimension d. It is easily verified that the subset compression scheme gives a sample compression scheme of size 2 for the maximal class C of VC dimension 2 in Figure 6.

It is an open question whether there exists a compression scheme of size $O(d)$ for every maximal class of VC dimension d, or for every class of VC dimension d. The structure of maximal classes of VC dimension d is discussed further in Floyd's thesis (1989).

8. Sample compression schemes and self-directed learning

Goldman and Sloan (1994) discussed the relationship between the VC dimension and *self-directed learning*, a variant of on-line learning where the learner gets to select the presentation order of instances from the finite domain X. The learning complexity of self-directed learning is defined by the number of examples for which the learner predicts the incorrect label.

Self-directed learning differs from sample compression in that the learner in self-directed learning is not restricted to a particular sample set. The learner gets to select the examples from the domain, one at a time.

Thus, for example, the self-directed learning complexity for monotone monomials is one, because the learner can first look at the example with all variables set to '0', and then at all examples with only one variable set to '1', and so on. The first positive example that the learner finds completely characterizes the monotone monomial. In contrast, in sample compression schemes the learner is restricted to saving a subset of the original sample set. Thus, even though each monotone monomial could be characterized by the positive example with the smallest number of variables set to '1', that positive example might not be included in the original sample set.

9. Conclusions and related work

This section summarizes some of the relationships between pac-learning, compression schemes, and the VC dimension. This section later discusses open questions concerning compression schemes for maximal classes, "unlabeled" compression schemes, and iterative compression algorithms.

From the work of Blumer et al. (1989) it has been known for some time that a concept class is pac-learnable from a fixed-size sample if and only if the VC dimension is finite. We summarize some of the relationships discussed in this paper between pac-learning and sample compression, between the VC dimension and sample compression, and, for finite classes, the relationships between sample compression and mistake-bounded learning algorithms.

First, following Littlestone and Warmuth (1986), sample compression implies pac-learning. In particular, Theorem 6 shows that for any class with a sample compression scheme of size at most k, there is a corresponding pac-learning algorithm with sample complexity that is linear in k.

Second, by using some results from Welzl (1987) we give a sample compression scheme of size d for any maximum class of VC dimension d. Further, for a maximum class of VC dimension d on a sufficiently large set X there is no sample compression scheme of size less than d. Together with Theorem 6, this result improves on the previously-known sample complexity for pac-learning for maximum classes of VC dimension d.

We have shown that mistake-bounded learning algorithms imply sample compression schemes of the same size. More precisely, we show in Theorem 3 that any finite class with a mistake-driven and mistake-bounded learning algorithm with bound k has a sample compression scheme of size at most k. In particular, this implies that any finite class C has a sample compression scheme of size at most $\log |C|$.

9.1. *Compression schemes of small size*

We believe that small compression sets are interesting in their own right since they represent a concept consistent with the whole sample set. The main open question of this paper is whether there is a sample compression scheme of size d, or of size $O(d)$, for every concept class of VC dimension d. We presented a compression scheme of size d for all maximum classes of VC dimension d and also gave a sample compression scheme of size d for *some* classes that are maximal but not maximum and have VC dimension d. We defined randomly-generated classes of VC dimension d, and showed that a large proportion of randomly-generated maximal classes of VC dimension d are not maximum.

There is a refinement of the above open problem that is even more intriguing. In the definition of compression schemes used in this paper the compression function maps every finite *set* of labeled examples to a *subset* of at most k *labeled* examples. From the combinatorial point of view the following restricted definition of a compression scheme might be the most interesting even though at this point we don't know how to obtain better sample complexity bounds by exploiting this restriction. In the restricted definition

we require that the compression function maps every finite set of labeled examples to a subset of k examples with their labels removed.

Most examples of sample compression schemes given in this paper can be modified to unlabeled sample compression schemes. For example, for the class of intervals (Section 3), since the compression set always consists of alternating positive and negative examples, starting with a positive example (when the points are ordered left to right), then it is not necessary to explicitly save the labels of the points in the compression set. The compression scheme given for intersection closed concept classes given in Section 3 only uses positive examples in the compression set and thus the labels are redundant. For any application of the One-Pass Compression Scheme in Section 4.1 the labels are also redundant since the on-line algorithm predicts wrongly on all examples in the compression set. However, we don't know of a modification of the VC Compression Scheme for maximum classes that saves only unlabeled examples. It is again an open problem whether there is an "unlabeled" compression scheme of size d for any concept class of VC dimension d. Note that the latter definition leaves no slack because for any maximum concept class C of VC dimension d and any finite set S of the domain, the number of concepts in $C|S$ equals exactly the number of subsets of at most d unlabeled examples from S. Thus, there would be exactly one compression set and exactly one hypothesis for every sample set of size at most d.

A reasonable question one might ask is whether there are any size bounds on the smallest compression scheme for an arbitrary concept class of VC dimension d. Freund (1995) and Helmbold and Warmuth (1995) have given a partial answer to this question, where a compression scheme is presented that compresses a sample of size m, labeled by a concept from an arbitrary concept class of VC dimension d, to a compression set of size $O(d \log m)$ plus $O(d \log m(\log \log m))$ bits. This compression scheme is based on recent results by Freund (1995) on boosting "weak" pac-learning algorithms. In this method $O(\log m)$ hypotheses are constructed by giving sets of $O(d)$ examples to the weak learning algorithm. Each individual hypothesis is guaranteed to have error at most half with probability at least half with respect to some judiciously chosen distribution. The majority of all hypotheses are guaranteed to be consistent with the whole sample. The compression set consists of all $O(d \log m)$ examples plus $O(\log \log m)$ bits per example to associate each example with the right call of the weak algorithm.

9.2. Generalized compression schemes

In the definitions discussed so far we required that the compression sets represent hypotheses that are consistent with the whole original example set. It would be natural to only require consistency with all but a fraction of γ of the original examples. If the goal is to find a hypothesis with error ϵ then a choice of $\gamma = \epsilon/2$ would be reasonable. In this paper the probability that a compression set of size d out of a sample of size m has error at least ϵ and is consistent with the whole sample can easily be bounded by $(1-\epsilon)^{m-d}$. If the hypotheses represented by the compression set only have to be γ-consistent, then Chernoff bounds must be used instead. The generalized notion of compression schemes

is likely to lead to improved sample complexity bounds. However we did not use this generalized notion in this paper, because we wanted to keep the proofs simple.

Sample compression schemes can also be used to learn classes of real-valued functions instead of concepts (which are binary functions). In this case compression sets must represent functions that have small loss on the whole original sample. Loss bounds for this generalization still have to be developed.

9.3. Iterative compression algorithms

This paper discussed briefly the use of sample compression schemes in constructing batch learning algorithms for pac-learning. Another application of sample compression schemes is for space-bounded iterative compression algorithms that save only a small number of examples at one time. Let $C \subseteq 2^X$ be a class with a sample compression scheme of size d. An iterative compression algorithm draws $d + 1$ examples, and saves only d of these examples, using the sample compression scheme. The iterative compression algorithm continues to draw a new example, to choose a compression set of size d from the $d + 1$ saved examples, and to discard the example that is not in the compression set. The compression set of size d represents the current hypothesis of the learning algorithm.

For a fairly simple example, one iterative compression algorithm for axis-parallel rectangles in R^2 (of VC dimension 4) saves the rightmost, leftmost, top, and bottom positive points seen so far; these points define the current hypothesis of the algorithm. When a new point is drawn whose label is predicted incorrectly by the current hypothesis, then the new point is saved and one of the old points might be discarded; the iterative compression algorithm always saves at most four points. Each time that the compression set is changed, the size of the hypothesized axis-parallel rectangle is increased.

As a more interesting application of the iterative compression algorithm, Floyd (1989) discusses classes defined by n-dimensional vector spaces of real functions on some domain X. Such classes include balls in R^{n-1}, positive halfspaces in R^n, and positive sets in the plane defined by polynomials of degree at most $n - 1$. With appropriate restrictions to the domain X given in Floyd's thesis (1989, p. 102), each of these classes is a maximum class of VC dimension n, and the iterative compression set for each class saves at most n examples at a time. This compression set of n examples saved by the iterative compression algorithm defines the boundary between the positive and the negative examples in the hypothesis. Note that the hypothesis represented by the current compression set is not necessarily consistent with all of the examples that have been seen so far. Nevertheless, for maximum classes the iterative compression algorithm is *acyclic*; that is, there is a partial order on the set of all possible compression sets, and each change of the compression set is to a compression set that is higher in the partial order. From Floyd (1989), there are many open questions concerning the use of iterative compression algorithms for pac-learning for maximum and maximal classes.

Appendix

Proof of Theorem 5

Proof: Let $Y^{\pm,c}$ be a sequence of m examples drawn independently at random according to the distribution D labeled by the concept c. Call any subset $A^{\pm}$ of at most d examples from $Y^{\pm,c}$ a *compression set* if $g(A^{\pm})$ is consistent with $Y^{\pm,c}$.

First we consider compression sets of size *exactly* d. Let $\mathcal{T}$ be the collection of d-element subsets of $\{1,..,m\}$. There are exactly $\binom{m}{d}$ such subsets. For any example x_i in the sample sequence, let $c(x_i)$ be the label for that example. For any $T = \{t_1,..,t_d\} \in \mathcal{T}$, let B_T contain all sample sequences $\langle\langle x_1, c(x_1)\rangle, \ldots, \langle x_m, c(x_m)\rangle\rangle$ such that the hypothesis $g(\{\langle x_{t_1}, c(x_{t_1})\rangle, \ldots, \langle x_{t_d}, c(x_{t_d})\rangle\})$ is consistent with the sample sequence $Y^{\pm,c} = \langle\langle x_1, c(x_1)\rangle, \ldots, \langle x_m, c(x_m)\rangle\rangle$. Let U_T contain all sample sequences $\langle x_1, \ldots, x_m\rangle$, where the hypothesis $g(\{\langle x_{t_1}, c(x_{t_1})\rangle, \ldots, \langle x_{t_d}, c(x_{t_d})\rangle\})$ has error greater than ϵ, with respect to the concept c. (Recall that the error of a hypothesis h is the probability, with respect to the distribution D, of the symmetric difference of c and h.) The probability that a sample sequence $Y^{\pm,c}$ of m examples is drawn, and the hypothesis represented by a sample compression set of d examples from $Y^{\pm,c}$ has error more than ϵ, is at most

$$\sum_{T \in \mathcal{T}} D^m(B_T \cap U_T).$$

For a particular T, what is an upper bound on the probability $D^m(B_T \cap U_T)$ of drawing m examples, such that $A^{\pm} = \{\langle x_{t_1}, c(x_{t_1})\rangle, \ldots, \langle x_{t_d}, c(x_{t_d})\rangle\}$ is a compression set of size exactly d for those m examples, and the hypothesis represented by $A^{\pm}$ has error greater than ϵ? Because the elements of $Y^{\pm,c}$ are drawn independently from the distribution D, for a fixed T we can assume that the d examples of the compression set $A^{\pm}$ are drawn first. Next the remaining $m - d$ elements of $Y^{\pm,c}$ are drawn. If $g(A^{\pm})$ has error greater than ϵ and is consistent with the remaining $m - d$ elements of $Y^{\pm,c}$ then the probability that a single example drawn from X is consistent with $g(A^{\pm})$ is less than $1 - \epsilon$. The probability that $m - d$ examples drawn from X are consistent with the hypothesis $g(A^{\pm})$ is less than $(1 - \epsilon)^{m-d}$. Thus

$$D^m(B_T \cap U_T) < (1 - \epsilon)^{m-d}.$$

Because $|\mathcal{T}| = \binom{m}{d}$,

$$\sum_{T \in \mathcal{T}} D^m(B_T \cap U_T) < \binom{m}{d}(1 - \epsilon)^{m-d}.$$

Now we consider compression sets of size at most d. What is the probability of drawing m examples, such that there is a compression set of size at most d for those m examples, and the hypothesis represented by the compression set has error greater than

ϵ? This probability is less than

$$\sum_{i=0}^{d} \binom{m}{i} (1-\epsilon)^{m-i}.$$

■

Acknowledgments

This work benefited from discussions with David Haussler, Dick Karp, Nick Littlestone, and Rob Schapire. This work developed from results in the unpublished manuscripts of Littlestone and Warmuth (1986) and Welzl (1987); we would like to acknowledge again these contributions from Nick Littlestone and Emo Welzl. We would also like to thank the anonymous referees for valuable suggestions, and to thank Lenny Pitt for helpful suggestions both in content and in style. In particular, the addition of the section about sample compression and mistake-bounded learning algorithms is due to a suggestion from Lenny.

Notes

1. This is a measure-theoretic condition given by Blumer et al. (1989). It is not likely to exclude any concept class considered in the context of machine learning applications.
2. If the concept class is finite then this definition is equivalent to requiring that the intersection of any pair of concepts in the class is also in the class.
3. Our definition of a *mistake-driven* algorithm is closely related to Haussler's (1988) definition of a *conservative* learning algorithm, where the current hypothesis is modified if and only if it is inconsistent with the current example.
4. **RP** is the class of problems solvable by randomized polynomial time algorithms, and **NP** is the class of problems solvable by nondeterministic polynomial time algorithms.

References

Angluin, D. (1988). Queries and concept learning. *Machine Learning*, Vol. 2 No. 4, 319-342, Apr. 1988.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. (1987). Occam's razor. *Information Processing Letters*, Vol. 24, 377-380.

Blumer, A., A. Ehrenfeucht, D. Haussler, & M. Warmuth. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, Vol. 36, No. 4, 929-965.

Blumer, A., & Littlestone, N. (1989). Learning faster than promised by the Vapnik-Chervonenkis dimension. *Discrete Applied Mathematics* 24, p.47-53.

Cesa-Bianchi, N., Freund, Y., Helmbold, D. P., Haussler, D., Schapire, R. E., & Warmuth, M. K. (1993). How to use expert advice. *Proceedings of the 25th ACM Symposium on the Theory of Computation*, 382-391.

Clarkson, K. L. (1992). Randomized geometric algorithms. In F. K. Hwang and D. Z. Hu (Eds.), *Euclidean Geometry and Computers*. World Scientific Publishing.

Ehrenfeucht, A., Haussler, D., Kearns, M., & Valiant, L. (1987). A general lower bound on the number of examples needed for learning. *Proceedings of the 1988 Workshop on Computational Learning Theory,* Morgan Kaufmann, 139-154.

Floyd, S. (1989). *On space-bounded learning and the Vapnik-Chervonenkis dimension.* PhD thesis, International Computer Science Institute Technical Report TR-89-061, Berkeley, California.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. To appear in *Information and Computation.*

Goldman, S., & Sloan, R. (1994). The power of self-directed learning. *Machine Learning,* Vol. 14 No. 3, 271-294.

Haussler, D. (1988). *Space efficient learning algorithms.* Technical Report UCSC-CRL-88-2, University of California Santa Cruz.

Haussler, D., Welzl, E. (1987). Epsilon-nets & simplex range queries. *Discrete and Computational Geometry 2,* 127-151.

Helmbold, D. P., & Warmuth, M. K. (1995). On weak learning. *Journal of Computer and System Sciences,* to appear.

Helmbold, D., Sloan, R., & Warmuth, M. (1990). Learning nested differences of intersection-closed concept classes. *Machine Learning* 5, 165–196, 1990.

Helmbold, D., Sloan, R., & Warmuth, M. (1992). Learning integer lattices. *Siam Journal on Computing,* Vol. 21 No. 2, 240–266.

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

Littlestone, N. (1989). *Mistake bounds and logarithmic linear-threshold learning algorithms.* PhD thesis, Technical Report UCSC-CRL-89-11, University of California Santa Cruz.

Littlestone, N., Haussler, D., & Warmuth, M. (1994). Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation,* Vol. 115 No. 2, 148-292.

Littlestone, N, & Warmuth, M. (1986). *Relating data compression and learnability.* Unpublished manuscript, University of California Santa Cruz.

Mitchell, T. (1977). Version spaces: a candidate elimination approach to rule learning. *Proceedings of the International Joint Committee for Artificial Intelligence 1977.* Cambridge, Mass., 305-310.

Pach, J., & Woeginger, G. (1990). Some new bounds for epsilon-nets. *Proceedings of the Sixth Annual Symposium on Computational Geometry,* Berkeley, California, 10-15.

Pitt, L., & Valiant, L. (1988). Computational limitations on learning from examples. *Journal of the Association for Computing Machinery,* Vol. 35 No. 4, 965-984.

Quinlan, J., & Rivest, R. (1989). Inferring decision trees using the minimum description length principle. *Information and Computation,* Vol. 80, 227-248.

Rissanen, J. (1986). Stochastic complexity and modeling. *Annals of Statistics,* Vol. 14 No. 3, 1080-1100.

Sauer, N. (1972). On the density of families of sets. *Journal of Combinatorial Theory (A)* 13, 145-147.

Schapire, R. (1990). The strength of weak learnability. *Machine Learning,* Vol. 5 No. 2, 197-227.

Shawe-Taylor, J., Anthony, M., & Biggs, N. (1989). *Bounding sample size with the Vapnik-Chervonenkis dimension.* Technical Report CSD-TR-618, University of London, Royal Halloway and New Bedford College.

Valiant, L.G. (1984). A theory of the learnable. *Communications of the Association for Computing Machinery,* Vol. 27, No. 11, 1134-42.

Vapnik, V.N. (1982). *Estimation of dependencies based on empirical data.* Springer Verlag, New York.

Vapnik, V.N. & Chervonenkis, A.Ya. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications,* Vol. 16, No. 2, 264-280.

Welzl, E. (1987). *Complete range spaces.* Unpublished notes.

Welzl, E., & Woeginger, G. (1987). *On Vapnik-Chervonenkis dimension one.* Unpublished manuscript, Institutes for Information Processing, Technical University of Graz and Austrian Computer Society, Austria.

Received April 5, 1993
Accepted June 2, 1995
Final Manuscript June 14, 1995