Sample compression, learnability, and the Vapnik-Chervonenkis dimension.

Sally Floyd*
Manfred Warmuth†

UCSC-CRL-93-13
March 30, 1993

Baskin Center for
Computer Engineering & Information Sciences
University of California, Santa Cruz
Santa Cruz, CA 95064 USA

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 d for a concept class $C \subseteq 2^X$ consists of a compression function and a reconstruction function. The compression function, given a finite sample set consistent with some concept in C, chooses a subset of k examples as the compression set. The reconstruction function, given a compression set of k examples, reconstructs a hypothesis on X. Given a compression set produced by the compression function from a sample of a concept in C, the reconstruction function must be able to reproduce a hypothesis consistent with that sample. We demonstrate that the existence of a fixed-size sample compression scheme for a class C is sufficient to ensure that the class C is learnable.

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, and we give a sample compression scheme of size d that applies to some maximal and nonmaximum classes. It is unknown whether there is a sample compression scheme of size d for every class of VC dimension d.

---

*Address: Lawrence Berkeley Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, floyd@ee.lbl.gov. This author 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.

†Address: Department of Computer Science, University of California, Santa Cruz, CA 95064, Thus author was supported by ONR grants N00014-K-86-K-0454 and NO0014-91-J-1162 and NSF grant IRI-9123692

1 Introduction

In this paper we discuss the use of sample compression schemes within computational learning theory. We define a sample compression scheme of size k for a concept class, consisting of a compression function and a reconstruction function; this formulation of a sample compression scheme was first introduced in [LW86]. Given a finite set of labeled examples, the compression function selects a compression set of at most k 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.

This research on sample compression schemes has several distinct motivations. One motivation is to demonstrate that the existence of an appropriate sample compression scheme is sufficient to ensure learnability. This approach provides an alternative to that of [BEHW89], which uses the Vapnik-Chervonenkis dimension to classify learnable geometric concepts.

A second motivation of this work is to explore the combinatorial properties of concept classes of VC dimension d. We give a sample compression scheme of size $\log |C|$ for any finite concept class $C \subseteq 2^X$. For infinite concept classes, we use the Vapnik-Chervonenkis dimension, and we define *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 [W87]. We give a sample compression scheme of size d for any maximum concept class of VC dimension d. Further, we show that for any sufficiently large maximum class of VC dimension d, there can be no sample compression scheme of size less than d. We give a sample compression scheme that applies for *some* concept classes that are maximal but not maximum. Recently a compression scheme of size $O(d \log m)$ for classes of VC dimension d was presented in [LSW93]; this result uses a more general definition of a sample compression scheme, and relies on the boosting schemes of weak learning algorithms [F90][S90]. It remains an open question [LW86] whether there is a sample compression scheme of size $O(d)$ for every class of VC dimension d.

A third motivation of this work is to explore the use of sample compression schemes of size at most d in batch learning algorithms, and in on-line learning algorithms that save at most d examples at each step. Batch learning algorithms are explored briefly in this paper; on-line learning algorithms are explored in more detail in [F89].

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 a One-Pass Halving Compression Scheme and a Multiple-Pass Halving Compression Scheme for any finite class $C \subseteq 2^X$. Both of these compression schemes are of size $\log |C|$. In Section 5 we define maximal and maximum classes of VC dimension d. For any maximum class $C \subseteq 2^X$ of VC dimension d, we give a sample compression scheme of size equal to d called the VC Compression Scheme; we discuss compression and reconstruction algorithms to implement this scheme. In Section 5.3 we prove that for any sufficiently large maximum class of VC dimension d, there can be no sample compression scheme of size less than d. In Section 6 we show that a sample compression scheme for a class C can be used as a basis for a learning algorithm for that class;[1] this result improves on the previously-known sample complexity of batch learning algorithms for such classes. Finally, Section 7 discusses sample compression schemes for maximal classes of VC dimension d.

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

[1]The original sample complexity bounds of [LW86] are slightly weaker; similar proofs for extended schemes appear in [LSW93].

A *domain* is any set X. 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*. If Y is a set of unlabeled examples, then Y' always denotes a corresponding set of (labeled) examples.

A *concept* c on the domain X is any subset from X. Concept c labels the element of c with "1" and the elements of $X - c$ with "0". A *concept class* 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: $C|Y = \{c \cap Y : c \in C\}$. We say that the class C is finite if $|C|$ is finite; otherwise we say that the class C is infinite. A *sample set* is a set of examples from $X \times \{0,1\}$; a *sample sequence* is a sequence of examples, possibly including duplicates. A sample set (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 we review the connection between pac-learning and the Vapnik-Chervonenkis dimension. In [V84], Valiant introduced a 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 P on the domain X, and the examples are labeled consistently with some unknown target concept c in the class C.

Definitions (pac-learning): 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 two inputs, the accuracy parameter ϵ and the confidence parameter δ, along with an oracle that provides labeled examples of the target concept c. The *sample size* of the algorithm is the number of labeled examples in the sample sequence drawn from the oracle. The learning algorithm returns the hypothesis h. The error of the hypothesis is the total probability, with respect to the distribution P, of the symmetric difference of c and h.

A concept class C is called *learnable* if there exists a 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 and δ. □

For a finite concept class $C \subseteq 2^X$, Theorem 2.1 gives an upper bound on the sample complexity required for learning the class C. This upper bound is linear in $\ln|C|$.

Theorem 2.1: *([V82], [BEHW87], [BEHW89]): Let $C \subseteq 2^X$ 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.*

Definitions (the Vapnik-Chervonenkis dimension): For infinite classes such as geometric concept classes on E^n, Theorem 2.1 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 ([VC71], [BEHW89], [EHKV87]). For a concept class C on X, and for S $\subseteq$ X, let $C|S$ denote the restriction of concept class C to the set S. 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. □

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 2.2 from Blumer, Ehrenfeucht, Haussler, and Warmuth in [BEHW89] gives an upper bound on the sample complexity of learning algorithms in terms of the VC dimension of the class. This result in [BEHW89] is adapted from Vapnik and Chervonenkis in [VC71].

Theorem 2.2 (BEHW89): *: Let C be a well-behaved[2] 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 2.2, the VC dimension essentially replaces $\ln|C|$ as a measure of the size of the class C. [SAB89] improves the sample size in Theorem 2.2 to

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

for $0 < \alpha < 1$. This is equivalent to

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

(to facilitate comparison with bounds derived later in the paper). In this paper we ignore computational concerns. Our focuses here are sample size bounds and the combinatorics of concept classes of VC dimension d. Computational issues regarding compression schemes are discussed in [F89] and [LSW93].

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.

Definitions (sample compression schemes) [LW86][3]: A *sample compression scheme of size at most k* for a concept class C on X consists of two functions, a compression function and a reconstruction function. The *compression function* f maps every finite sample set to a subset of at most k labeled examples from the sample set called a *compression set*. 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. The requirement for a sample compression scheme is that, for any sample set Y' consistent with some concept in C, the hypothesis $g(f(Y'))$ is consistent with the original sample set Y'. In this paper the *size* of a compression set is defined as simply the number of examples in the compression set, not the number of bits used to encode those examples. □

[2]This is a measure-theoretic condition given in [BEHW89]. It is not likely to exclude any concept class considered in the context of machine learning applications.

[3]This paper essentially uses the simplest version of the compression schemes introduced in [LW86]; various more sophisticated schemes are discussed in [LSW93].

Example (rectangles): Consider the class of axis-parallel rectangles in E^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 E^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. □

Rectangles are one example of an "intersection closed" concept class. The results of [HSW89] lead to compression schemes of size at most d for any intersection closed class of VC dimension d.

Example (intervals on the line): 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 are on the boundaries of the first positive interval, and each succeeding pair of examples saved are the boundaries of the next positive interval. For the sample set in Figure 3.1, this compression function saves the examples $\{< x_3, 1 >, < x_5, 0 >, < x_7, 1 >, < x_{11}, 0 >, < x_{14}, 1 >, < x_{16}, 0 >\}$ 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 $2n$. □

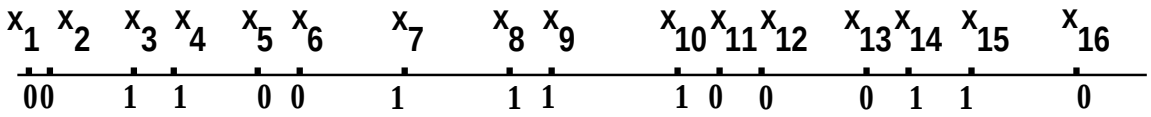

Figure 3.1: Union of three intervals on the line.

Although we have defined *labeled* compression schemes, where the compression function saves a labeled subset of the sample set, for the classes of rectangles and of intervals examined above, it would also be possible to define *unlabeled* compression schemes, where the compression function saves an unlabeled subset of the sample set. For example, for the class of intervals, if 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.

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 Sample compression schemes for finite classes

In this section we give two sample compression schemes of size $\log |C|$ for any finite concept class C. Compression schemes of size $\log |C|$ for finite classes have been proposed independently by Angluin.

The One-Pass Halving Compression Scheme gives a sample compression scheme of size $\lfloor \log |C| \rfloor$ for a finite class $C \subseteq 2^X$. 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.) Let X = $\{x_1, .., x_r\}$. We assume some arbitrary fixed order on the r elements of X.

The input to the sample compression algorithm is some labeled sample set Y', for $Y = \{x_{s_1}, .., x_{s_m}\}$ and for $Y \subseteq X$. The examples in Y are assumed to be labeled consistently with some concept in the class C.

The One-Pass Halving Compression Scheme (for finite classes).

- The compression function: The input to the compression function is the labeled sample set $Y' \subseteq X \times \{0, 1\}$. Let the current compression set A′ initially be the empty set, and let C_1 be the set of concepts consistent with A′. Initially, C_1 is set to C. The elements of Y' are examined one at a time, in order. Let the ith example from Y' be $< x_{s_i}, 0 >$, for example. If at least half of the concepts in C_1 contain $< x_{s_i}, 1 >$, then add $< x_{s_i}, 0 >$ to the current compression set, and remove all concepts that contain $< x_{s_i}, 1 >$ from C_1. If less than half of the concepts in C_1 contain $< x_{s_i}, 1 >$, then C_1 and A′ remain unchanged.

  The compression function can be viewed as an algorithm that examines the elements of Y' one at a time, considering the class C_1 of concepts consistent with the examples saved so far. For the next element x_{s_i}, the compression algorithm takes the label for x_{s_i} given by the majority of concepts in the class C_1, and predicts that label for x_{s_i}. If the compression algorithm predicts the correct label for x_{s_i}, then x_{s_i} is discarded. If the compression algorithm predicts the incorrect label for x_{s_i}, then the compression algorithm saves x_{s_i}, and updates C_1. Thus the compression algorithm saves an example when it makes a mistake predicting the label for that example.
- The reconstruction function: The reconstruction function is given as input the compression set A′ = $\{< x_{a_1}, l_{a_1} >, .., < x_{a_k}, l_{a_k} >\}$ containing k elements, for $l_i \in \{0, 1\}$. For x_i not in the compression set, let A'_i contain those elements from A' that precede x_i in the fixed order on the elements of X. Let C_i contain the concepts from C that are consistent with A'_i. Then the reconstruction function predicts the label for x_i that agrees with more than half of the concepts in C_i. If exactly half of the concepts in C_i contain one label for x_i, and half contain the other label, then the reconstruction function predicts '0' for x_i by default. For x_i in the compression set the reconstruction function predicts the label for x_i in the compression set. This will also be the label for x_i in less than half of the concepts in C_i.

Theorem 4.1: *Let $C \subseteq 2^X$ be any finite concept class. Then the One-Pass Halving Compression Scheme is a sample compression scheme of size $\lfloor \log |C| \rfloor$ for the class C.*

Proof: To show that the current compression set contains at most $\lfloor \log |C| \rfloor$ examples, it suffices to observe that during the compression function, each time an example is added to the current compression set, the size of C_1 is reduced by at least half. Thus if the current compression set reaches size $\lfloor \log |C| \rfloor$, there can be at most one concept in C consistent with the examples in the current compression set.

The current compression set predicts the correct label for all of the elements in the original sample set Y'. If, using the reconstruction function, some example in the target concept is not labeled consistently with more that half of the concepts in C_2, then either that example was not in the original sample set Y', or that example was saved in the current compression set. □

The One-Pass Halving Compression Scheme has a somewhat-involved reconstruction function that requires constructing several subclasses of the class C. By making more than one pass through the

sample set, we can give a multiple-pass halving compression scheme with a slightly more complicated compression function but a simplier reconstruction function. In this case, the compression set defines a class C_1 of all concepts consistent with the elements in the compression set. For each $x \in X$, the reconstruction function predicts the label for x given by the majority of concepts in C_1. Note that in the One-Pass Halving Compression Scheme it is not necessary to save the labels for the elements in the compression set. This property does not hold for the Multiple-Pass Halving Compression Scheme.

The Multiple-Pass Halving Compression Scheme (for finite classes).

- The compression function: The input to the compression function is a finite sample set $Y' \subseteq X \times \{0,1\}$. To start, let $C_1 = C$, and let the current compression set A$'$ be the empty set. Examine the elements of Y' one at a time, in any order. Let the ith example examined be $< x_i, 0 >$, for example. If at least half of the concepts in C_1 contain $< x_i, 1 >$, then add $< x_i, 0 >$ to the current compression set, and remove all concepts that contain $< x_i, 1 >$ from C_1. If less than half of the concepts in C_1 contain $< x_i, 1 >$, then C_1 and A$'$ remain unchanged.

  After one pass through the sample set Y', there still might be elements in Y' whose labels are not the same as the label predicted for that element by the majority of concepts in C_1. Additional passes through the sample set Y' might be required. In each pass, each element of Y' whose label is incorrectly-predicted by the current compression set is added to that compression set, and C_1 is modified accordingly. Once the compression set remains unchanged for one complete pass through the sample set, then no further passes are required.
- The reconstruction function: The reconstruction function is given as input the labeled compression set A'. Let C_2 contain all concepts consistent with the examples in the compression set A'. If x is in the current compression set, then the reconstruction function predicts the label for x in the compression set. If $x \in X$ is not in the current compression set, then the label predicted for x is the label for x in more than half of the concepts in C_2. If exactly half of the concepts in C_2 contain one label for x, and half contain the other label, then the compression set predicts '0' for x by default.

For both the One-Pass and the Multiple-Pass Halving Compression Scheme, the hypothesis predicted by the compression set is not necessarily a concept from the class C.

Theorem 4.2: *Let $C \subseteq 2^X$ be any finite concept class. Then the Multiple-Pass Halving Compression Scheme is a sample compression scheme of size $\lfloor \log |C| \rfloor$ for the class C.*

Proof: The proof is similar to the proof of Theorem 4.1. Because the compression set is of size at most $\lfloor \log |C| \rfloor$, the compression function requires at most $\lfloor \log |C| \rfloor$ passes through the sample set. □

Neither the one-pass halving compression scheme nor the multiple-pass halving compression algorithm is necessarily claimed to be an efficient algorithm. At this point, the discussion of these algorithms is of combinatorial interest, apart from questions of efficiency. Neither the one-pass nor the multiple-pass halving compression algorithm is applicable for an infinite class C. [LSW93] gives a compression scheme of size O$(d \log m)$ for any (possibly infinite) class of VC dimension d. As discussed later in this paper, it is an open question whether there are always compression schemes of size O(d) for arbitrary classes of VC dimension d.

5 Sample compression schemes for maximum classes

In this section we explore a sample compression algorithm based on the combinatorial structure of a class. Theorem 2.1 gives an upper bound on the sample complexity of a learning algorithm of a finite

class C that is linear in $\ln|C|$. The VC dimension was used to generalize this result to infinite classes; the more general result in Theorem 2.2 gives an upper bound on the sample complexity that is linear in the VC dimension of the class.

In the case of sample compression schemes, a finite class C has a sample compression scheme of size $\lfloor \log|C| \rfloor$. To discuss sample compression schemes for infinite as well as finite classes, we consider the combinatorial structure of the class based on the VC dimension. In this section we define a maximum class of VC dimension d. Section 5.1, gives a sample compression scheme of size d for any maximum class of VC dimension d; Section 5.2 gives an algorithm that implements the sample compression scheme for maximum classes. Section 5.3 shows 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.

Definitions (maximum and maximal classes): We use the definitions from [W87] 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 [VC71], [S72], the cardinality of C is at most $\Phi_d(m)$ for any class C of VC dimension d on a domain X of cardinality 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. Note that a concept class that is maximum on a finite domain X is also maximal on that set [WW87, pg. 53]. □

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 5.1: Class D is maximum. Class E is maximal but not maximum.

Figure 5.1, along with Figure 7.2 later in the paper, gives examples of classes that are maximal but not maximum. More examples can be found in [WW87] and [F89]. Recall that a concept c in a class C can be thought of either as a subset S of positive examples from the set X, or as the characteristic function of S on X. Each row in Figure 5.1 represents one concept on $\{w, x, y, z\}$.

5.1 The sample compression function for maximum classes

Definitions (the classes $\mathbf{C - x}, \mathbf{C^{\{x\}}}$): For $x \in$ X, define $C - x$ as $C|(X - \{x\})$, the restriction of the class C to the domain X-$\{x\}$. Define $C^{\{x\}}$ as the class $\{c \in C | x \notin c \text{ and } c \cup \{x\} \in C\}$; the class $C^{\{x\}}$ has the domain X-$\{x\}$. Thus each concept c in $C^{\{x\}}$ corresponds to the two concepts $c \cup \{< x, 0 >\}$ and $c \cup \{< x, 1 >\}$ in the class C. □

As an illustration, consider the maximum class D in Figure 5.1. 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.

We first give a theorem of Welzl on maximum concept classes.

Theorem 5.1 (W87, p. 9): *: 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, $|(C|Y)| = \Phi_d(|Y|)$. □

The following corollary from Welzl applies to maximum classes on a finite domain X. Corollary 5.3 extends one part of Corollary 5.2 to a maximum class on an infinite domain X. Corollary 5.2 is extended to any maximum and maximal class on an infinite domain X in [F89, p.25].

Corollary 5.2 (W87, p. 10): *: Let $C \subseteq 2^X$ be a maximum concept class of VC dimension $d \geq 1$ on the finite domain X. Then for $x \in$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 proof of Theorem 5.1, $|C - x| = \Phi_d(m-1)$. From the same theorem, if $|X - \{x\}| \geq d$, then $C - x$ is a maximum class on X-$\{x\}$ of VC dimension d. Similarly, because $|C^{\{x\}}| = \Phi_{d-1}(m-1)$, and $C^{\{x\}}$ is of VC dimension at most $d-1$, $C^{\{x\}}$ is a maximum class of VC dimension $d-1$ on X-$\{x\}$. □

Corollary 5.3: *Let $C \subseteq 2^X$ be a maximum concept class of VC dimension d on the infinite domain X. Then $C - x$ is a maximum class of VC dimension d on X-$\{x\}$.*

Proof: Because C is maximum of VC dimension d, for every finite subset Y of X, $C|Y$ contains $\Phi_d(|Y|)$ concepts on Y. For any finite subset Z of cardinality at least d of X-$\{x\}$, $C|Z$ is maximum of VC dimension d. From Corollary 5.2, if $Z - \{x\}$ is of cardinality at least d, then $(C|Z) - \{x\}$ is maximum of VC dimension d on Z-$\{x\}$. Because $(C|Z) - \{x\} = (C - \{x\})|Z$, $C - \{x\}$ is maximum of VC dimension d on every finite subset of X-$\{x\}$ of cardinality at least d. Therefore, $C - \{x\}$ is maximum of VC dimension d on X-$\{x\}$. □

Definitions (the class C^A) [W87]: Let $C \subseteq 2^X$ be a maximum concept class of VC dimension d. For A = $\{x_1, ..., x_k\}$, A $\subseteq$ X, C^A is defined as the class $(((C^{\{x_1\}})^{\{x_2\}})...)^{\{x_k\}}$. □

It is easy to see that for any distinct x,y in X, $(C^{\{x\}})^{\{y\}} = (C^{\{y\}})^{\{x\}}$ [W87, p. 8]. Therefore for any $A \subseteq X$, the class C^A is well-defined.

Corollary 5.4 (W87): *: 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 5.2. The class C^A contains the single concept c on X-A such that c remains a concept in C for any labeling of the elements of A. □

Definitions (the concept c_A) [W87]: 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, let c_A denote the unique concept in the class C^A 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 $\sum_{i=0}^{2n} \begin{pmatrix} m \\ i \end{pmatrix}$ 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 5.2 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. □

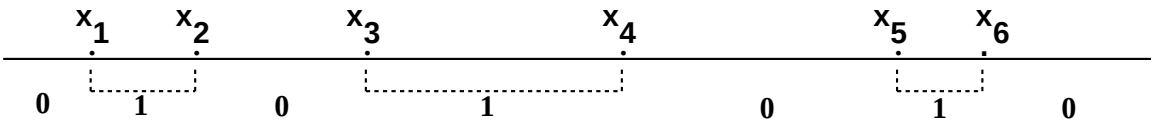

Figure 5.2: Union of three intervals on the line.

Definitions (the concepts $c_{A'}$, $c_{A',C|Y}$): 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 sample set A′, let $c_{A'}$ denote the concept with X-A labeled as in the concept c_A, and with A labeled as in A′. Thus for every labeled set A′ of cardinality d, for A $\subseteq$ X, there is a corresponding concept $c_{A'}$ on X. We say that the set A′ is a *compression set* for the concept $c_{A'}$, and that the set A′ *represents* the concept $c_{A'}$. 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′ *predicts labels* for the elements in X, according to the concept $c_{A'}$.

Let $c_{A',C|Y}$, for $A \subseteq Y \subseteq X$, denote the concept $c_{A'}$ in the maximum class $C|Y$. If not otherwise specified, $c_{A'}$ is assumed to be the concept $c_{A',C}$. □

Lemma 5.5 shows that $c_{A',C|Y}$ is the same as $c_{A',C}$ restricted to the set Y.

Lemma 5.5: *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′ of A, and for $x \in Y$, $c_{A'}$ and $c_{A',C|Y}$ assign the same label to the element x.*

Proof: If $x \in A$, then for both $c_{A'}$ and $c_{A',C|Y}$, x is labeled as in A′. If $x \notin A$, then assume for purposes of contradiction that Lemma 5.5 is false. Without loss of generality, assume that $c_{A'}$ contains $< x, 0 >$, and that $c_{A',C|Y}$ contains $< x, 1 >$. Because $c_{A'}$ contains $< x, 0 >$, for every labeling of A, the class C contains a concept with that labeling, and with $< x, 0 >$. Because $c_{A',C|Y}$ contains $< x, 1 >$, for every labeling of A, the class $C|Y$ contains a concept with that labeling, and with $< x, 1 >$. Then the set $A \cup \{x\}$ is shattered in the class C, contradicting the fact that C is of VC dimension d. □

Theorem 5.6 shows that for a maximum class C of VC dimension d on a finite domain X, every concept in C is represented by some labeled set A′ of cardinality d. Theorem 5.6 is also stated, although not with this proof, by Welzl in [W87, p. 27]. Theorem 5.7 shows that, using this approach, there is a sample compression scheme of size d for any maximum class C of VC dimension d on a (possibly infinite) domain X.

Theorem 5.6: *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′ of exactly d elements, for $A' \subseteq X \times \{0,1\}$, such that $c = c_{A'}$.*

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′ 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 $X = \{x_1, x_2, ..., x_m\}$, and let $Y \subset X$ for $Y = \{x_1, x_2, ..., x_{m-1}\}$. There are two cases to consider.

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

From Corollary 5.2, $C|Y$ is maximum of VC dimension d. Thus by the induction hypothesis, each concept c in $C|Y$ can be represented by a compression set A′ of d labeled elements, for $A \subseteq Y$, with $c = c_{A',C|Y}$. From Corollary 5.4, A′ represents some concept c_A (or $c_{A',C}$) on Y. From Lemma 5.5, $c = c_{A',C|Y}$ agrees with $c_{A',C}$ on Y. If $c_{A'}$ contains $< x_m, 1 >$, then $c \cup \{< x_m, 1 >\}$ is in C, violating the assumption for Case 1. Thus $c_{A'}$ contains $< x_m, 0 >$, and case 1 is done.

Case 2: Let c be a concept in $C|Y$ such that $c \cup \{< x_m, 0 >\}$ and $c \cup \{< x_m, 1 >\}$ are both in C. Thus $c \in C^{\{x_m\}}$. From Corollary 5.2, $C^{\{x_m\}}$ is a maximum class of VC dimension $d - 1$ on Y. By the induction hypothesis, there is a compression set B' of $d - 1$ elements of Y, such that $c = c_{B',C^{\{x\}}}$.

Let $c_1 = c \cup \{< x_m, 0 >\}$. Let A′ = B′ $\cup \{< x_m, 0 >\}$. From Corollary 5.4, the labeled set A′ of cardinality d represents a unique concept $c_{A'}$ in C.

Let $C_1 = C^{\{x_m\}}$.

We show that $c_{A',C}$ and $c = c_{B',C_1}$ assign the same labels to all elements of Y. Assume not, for purposes of contradiction. Then there is some element x_i of $Y - B$ such that x_i is assigned one label l_i in $c_{A',C}$, and another label $\overline{l_i}$ in c_{B',C_1} Because $c_{A',C}$ contains $< x_i, l_i >$, then for each of the 2^d labelings of A, and for $< x_i, l_i >$, there is a concept consistent with that labeling in C. Because c_{B',C_1} contains $< x_i, \overline{l_i} >$, then for each of the 2^{d-1} labelings of B, and for $< x_i, \overline{l_i} >$, there is a concept consistent with that labeling in $C_1 = C^{\{x_m\}}$. For each concept in $C^{\{x_m\}}$, there is a concept in C with $< x_m, 0 >$, and another concept in C with $< x_m, 1 >$. 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′ is a compression set for the concept $c \cup \{< x_m, 0 >\}$, and case 2 is done. □

Note that for a concept $c \in C$, there might be more than one compression set A′ such that $c = c_{A'}$.

Any maximum class C on a finite domain X is also a maximal class. However for an infinite domain X for any $d \geq 1$ there are concept classes of VC dimension d that are maximum but not maximal [WW87, p. 53]. 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. For a maximum class on an infinite domain X, we expand our definition of c_A where A⊆X, $|A| = d$. For a maximum concept class C of VC dimension d on an infinite domain X, it is not necessarily true that $C^{\{x\}}$ is maximum of VC dimension $d - 1$.[4]

Example (a maximum class that is not maximal): Consider the maximum class C of VC dimension 1 on an infinite domain X, where C contains all concepts with exactly one positive example. This class is not maximal, because the concept with no positive examples could be added to C without increasing

[4][F89] shows that every maximum class C of VC dimension d on an infinite domain X has a unique extension to a maximum and maximal class of VC dimension d on X. [F89] also shows that if C is both maximum and maximal of VC dimension d on the infinite domain X, then $C^{\{x\}}$ is maximum and maximal of VC dimension $d - 1$ on X-$\{x\}$.

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, and so $c_{\{x\}}$ does not represent a concept in C. For such a class, for A$\subseteq$X, $|$A$|$=d, we define c_A by its value on finite subsets of X.

Definitions (the concept c_A for infinite X): For the infinite set X, for A$\subseteq$B$\subseteq$X, for $|$A$|$=d, and for B finite, we define c_A on the elements in B-A as $c_{A,C|B}$. From Lemma 5.5, c_A assigns a unique label to each element $x \in$X-A. □

Thus, in the maximum class C above, $c_{\{x\}}$ is defined as the concept with all negative examples on X-$\{x\}$, even though $c_{\{x\}} \cup < x, 0 >$ is not a concept in C.

Theorem 5.7 extends Theorem 5.6 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' of size at least d, for $Y = \{x_1, .., x_m\} \subseteq X$. The examples in Y' are assumed to be labeled consistently with some concept in C.

The VC Compression Scheme (for maximum classes).

- The compression function: The compression function is given as input any sample set Y' of cardinality at least d of examples labeled consistently with some concept in the class C. Consider the finite class $C|Y$, which is maximum of VC dimension d. Let c be the concept on Y given by the sample set Y'. From Theorem 5.6, there is a compression set A$'$ of exactly d elements, for $A' \subseteq Y'$, such that the concept c in the class $C|Y$ is represented by the compression set A$'$. This set A$'$ is the compression set chosen by the sample compression function.
- The reconstruction function: The reconstruction function is given as input the compression set A$'$. For an element $x \in X$, the reconstruction function predicts the label for x in the set A'. If A' is of cardinality less than d, then the compression set arbitrarily predicts the label '0' for all $x \notin A$. Assume that A' is of cardinality d. 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 $A \cup \{x\}$. If c_A in C_1 contains $< x, 0 >$, then the reconstruction function predicts the label '0' for x; if c_A in C_1 contains $< x, 1 >$, then the reconstruction function predicts '1' 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 5.7: *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 exactly d for C.*

Proof: Let the input to the sample compression scheme be a finite labeled sample set Y' of size at least d. For c a concept on the finite set $Y \subseteq X$, the compression function saves the labeled set A$'$ of cardinality d, for $A' \subseteq Y'$, such that $c = c_{A',C|Y}$. The reconstruction function gives as a hypothesis the concept $c_{A'}$ on X from the class C.

From Theorem 5.1, $C|Y$ is a maximum class of VC dimension d. Thus, by Theorem 5.6, for the concept c on Y there exists a subset A$'$ of Y', for $|A'| = d$, such that $c = c_{A',C|Y}$. From Lemma 5.5, the concept $c_{A'}$ on X is consistent with the original sample set $c_{A',C|Y}$. Thus we have a sample compression scheme of size d for maximum classes of VC dimension d. □

5.2 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. Theorems 5.6 and 5.7 proved that there is a compression set for every finite labeled sample set. The proof of Theorem 5.6 suggests an algorithm to find the compression set. The input for the compression algorithm is a finite sample set Y' of size at least d, for $Y' \subseteq X \times \{0, 1\}$, labeled consistently with some concept c in C. Let

$Y' = \{< x_1, l_1 >, .., < x_m, l_m >\}$. The output of the compression algorithm is a labeled compression set $A' \subseteq Y'$ of cardinality d that represents some concept in C consistent with the labeled set Y'.

Definitions (the consistency oracle): The *consistency problem* for a particular concept class C is defined in [BEHW89] 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. □

From [BEHW89], if the consistency problem for C is NP-hard and **RP** $\neq$ **NP** then C is not polynomially learnable by an algorithm that produces hypotheses from C.

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

- The compression algorithm: The compression algorithm is given as input the finite sample set Y', labeled consistently with some concept in C. The compression algorithm examines each element of the set Y' in arbitrary order, deciding whether to add each element in turn to the compression set A'. Initially, A' is the empty set. At step i, the algorithm decides whether to add the labeled element $< x_i, l_i >$ to the partial compression set A', for $0 \leq |A'| \leq d-1$ and $< x_i, l_i > \in Y'$.

  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 of Y as in Y'. If so, then $< x_i, l_i >$ is added to A'. Each such decision requires at most $2^{|A|}$ calls to the consistency oracle. The compression algorithm terminates when A' is of cardinality d.
- The reconstruction algorithm: The reconstruction algorithm is given as input the compression set A' of cardinality d, and is asked to predict the label for some element $x_i \subseteq X$. If $x_i \in A$, then the reconstruction algorithm predicts the label for x_i in the compression set. If $x_i \notin A$, let C_1 be $C|(A \cup \{x_i\})$. If, for each of the 2^d possible labelings A' of A, there is a concept in C_1 consistent with A'$\cup < x_i, 0 >$, then c_{A',C_1} predicts label '0' for the element x_i. Otherwise, c_{A',C_1} predicts the label '1' for x_i. The label for x_i can be determined with at most 2^d calls to the consistency oracle.

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 3.1. The examples in Figure 3.1 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' be the empty set. First consider the example "x_1". There is no concept in C_3 with $< x_1, 1 >$, and with the other examples labeled as in Figure 3.1. Therefore the example "x_1" is not added to the current compression set. There is a concept in C_3 with $< x_2, 1 >$, and with the other examples labeled as in Figure 3.1. Therefore, $< x_2, 0 >$ is added to the current compression set A'. For every labeling of the point "x_2", there is a concept in C_3 consistent with that labeling, and with the labeling of the other points in the sample set. 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 $< x_2, 1 >$, $< x_3, 0 >$, 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 = $\{< x_2, 0 >, < x_4, 1 >, < x_6, 0 >, < x_{10}, 1 >, < x_{13}, 0 >, < x_{15}, 1 >\}$. The reconstruction function for this class is illustrated by Figure 5.2. □

Theorem 5.8 shows that the greedy compression algorithm terminates with a correct compression set.

Theorem 5.8: *Let $C \subseteq 2^X$ be a maximum class of VC dimension d, and let Y' be a finite sample set labeled consistently with some concept $c \in C$, for $|Y'| \geq d$. Then the Greedy Compression Algorithm*

after each step maintains the invariant that, for the partial compression set A′, the labeled set Y′-A′ is consistent with some concept in C^A. Further, the Greedy Compression Algorithm on Y′ 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: the labeled set Y′-A′ is consistent with some concept in C^A.

Assume for purposes of contradiction that the greedy compression algorithm ends with the compression set A′, where $|A'| = s < d$. Then the labeled sample set Y′ is consistent with some concept in C^A. From Corollary 5.2, C^A is a maximum class of VC dimension $d - s$ on Y-A. From Theorem 5.6, there is a compression set of cardinality $d - s$ from Y′-A′ for $c|(Y - A)$. 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 $s < d$. □

This 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. More efficient algorithms for the VC Compression Scheme are explored in [F89].

5.3 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, if the cardinality of the domain X is exponential in d, then there can be no sample compression scheme of size less than d. This refers to a sample compression scheme as defined in Section 3 , where a sample compression set consists of an (unordered) subset from the original sample set. We also show that for any concept class of VC dimension d there is no compression scheme that compresses sample sets of size at least d to subsets of size at most $d/5$.

Theorem 5.9: *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 $d2^{d-1}$.*

Proof: Let Y be any subset of X of cardinality $m \geq d2^{d-1}$. 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 $m \geq d^2 2^{d-1}$. □

Note that this argument does not necessarily apply for classes of VC dimension d that are not maximum. For example, the VC dimension of the class of arbitrary halfspaces in the plane is three, but there exists a sample compression scheme of size two for this class [BL89]. The class of arbitrary halfspaces in the plane 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 [F89].

Theorem 5.10: *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

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

[BEHW89], 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. □

6 Batch learning algorithms using sample compression schemes

Given a sample compression scheme for a class $C \subseteq 2^X$ then there is a learning algorithm that $uses$ this scheme as follows: It requests a sample sequence Y' of m examples from the oracle labeled consistently with some concept in the class C. It then converts the sample sequence to a sample set, removing duplicates, and uses the compression function to find a compression set for this sample set. The reconstruction function maps the compression set to a hypothesis on X which is the hypothesis of the learning algorithm. Note that this hypothesis is guaranteed to be consistent with all of the examples in the original sample set.

Littlestone and Warmuth [LW86] gave an upper bound on the sample size needed for a batch learning algorithm for the class C that uses a sample compression scheme of size at most d.

Theorem 6.1 (LW86): *Let P 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 P 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.

Lemma 6.2: *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

$$\frac{1}{\epsilon} \ln \frac{1}{\delta} + d + \frac{d}{\epsilon}(1 + \ln \frac{d}{\beta\epsilon} - 1 + \frac{\beta\epsilon}{d} m - \ln d) \leq m. \tag{6.1}$$

We use the fact from [SAB89] that

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

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 (6.1) we get

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

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

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

Since from [BEHW89]

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

we have

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

□

Theorem 6.3: *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 6.1 and Lemma 6.2. □

For maximum classes of VC dimension d, Theorem 6.3 slightly improves the sample complexity of batch learning from the previously known results from [BEHW89] and [SAB89] given in Theorem 2.2. Choosing $\beta = 1/2$ gives simple bounds. The bounds can be marginally improved by optimizing the choice of β as done in[5] [CBFH+93].

Note that the upper bounds have the form $O(\frac{1}{\epsilon}(d \log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$ where d is either the size of a compression scheme or the VC dimension for the concept class. 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 [PW90, HLW88].) Similarly one can show [HLW88] 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. There are also general lower bounds [EHKV87] of $\Omega(\frac{1}{\epsilon}(d + \log \frac{1}{\delta}))$ for learning any concept class of VC dimension d. It is an open problem whether there are particular compression schemes of size d for all (maximal) concept classes of VC dimension d with sample size $O(\frac{1}{\epsilon}(d + \log \frac{1}{\delta}))$.

7 Maximal classes

Every class C of VC dimension d can be embedded in a maximal class of VC dimension d: simply keep adding concepts to the class C until no more concepts can be added without increasing the VC dimension. Every maximal class of VC dimension 1 is also a maximum class [WW87], but for classes of VC dimension greater than 1, there are maximal classes that are not maximum. (Figures 5.1 and 7.2 show two different classes of VC dimension 2 that are maximal but not maximum.) In this section we discuss randomly-generated maximal classes, and we give a sample compression scheme that applies for *some* classes that are maximal but 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.

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 [F89]. 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.

[5] In [CBFH+93] a bound was optimized which had $2 \ln \frac{1}{1+\beta}$ in the denominator. Similar techniques can be used to optimize a bound with $1 - \beta$ in the denominator

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.

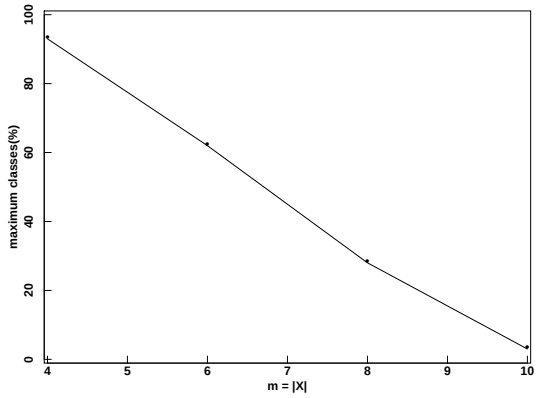

Figure 7.1: Randomly-generated maximal classes of VC dimension 2.

A program can determine whether or not a given class $C \subseteq 2^Y$ is a maximum class simply by counting the number of concepts in the class. (This is from Theorem 5.1.) Figure 7.1 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 the class Y; 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 7.1, 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 of VC dimension 3 on 6 or 8 elements were maximum. These results suggest that, for m and d sufficiently large, few of the randomly-generated maximal classes of VC dimension d on m elements will be maximum.

7.2 Compression schemes for maximal classes

The VC Compression Scheme described in Section 5 applies to maximum classes of VC dimension d; it can not necessarily be applied to maximal and nonmaximum classes of VC dimension d. For example, Figure 7.2 shows a maximal class of VC dimension 2 for which the VC Compression Scheme does not apply. 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 7.2: A maximal and nonmaximum class C of VC dimension 2.

Figure 7.2 gives a maximal and nonmaximum class $C \subseteq 2^X$ of VC dimension 2 for which the VC Compression Scheme does not apply. For example, 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 $c \in C^A$ on $X - A$. However, the subset compression scheme defined below does give a sample compression scheme of size 2 for the class C.

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', labeled consistently with some concept c in C, for $Y \subseteq X$. The compression function finds a subset $A' \subseteq Y'$, for $|A'| = d$, such that A' 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$'$, of cardinality d. The label that the compression set A$'$ predicts for an element $x_i \in X - A$ is determined by considering the class $C_i = C|(A \cup \{x_i\})$, which is 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 $< x_i, l_i >$, for $l_i \in \{0, 1\}$. In this case, the compression set A$'$ predicts the label l_i for x_i. (This reconstruction function is identical to that in the VC Compression Scheme, given the class C_i.)

  For the purpose of completeness, we define the label predicted for x_i when $(C_i)^A$ is empty. In this case, let the label predicted by A$'$ for x_i depend on the labels of the elements in the compression set A$'$. If there is only one possible label for x_i in concepts in the class C, given the labels of the elements in A$'$, then that is the label predicted by the compression set A$'$. Otherwise, arbitrarily let the compression set A$'$ predict the label '0' for x_i. With this definition, each compression set A$'$ 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 and the VC Compression Scheme are identical. For a maximal class let $c_{A'}$ denote the concept on X represented by the compression set A$'$ using the Subset Compression Scheme. The Subset Compression Scheme is motivated by a combinatorial characterization of maximal classes of VC dimension d by "forbidden labels" on subsets of $d + 1$ elements that is given in [F89]. 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; the subset

compression scheme has worked correctly for all of the maximal classes that we have examined. The following observation shows that the Subset Compression Scheme applies to *some* maximal classes that are not maximum.

Observation 7.1: *The subset compression scheme gives a compression scheme of size 2 for the maximal class C of VC dimension d in Figure 7.2.*

Proof: It is sufficient to show that for every possible set of 3 or 4 labeled examples consistent with some concept in C, the subset compression scheme gives a sample compression set of size 2. For example, the concept c=1100 is represented by the compression set A' ={$< x_1$, 1>, $< x_2$, 1>}. This follows because in $C|(A \cup \{x_3\})$, the compression set A′ predicts $< x_3, 0 >$, and in $C|(A \cup \{x_4\})$, the compression set A′ predicts $< x_4, 0 >$. The concept c=1100 is also represented by the compression set $\{< x_3, 0 >, < x_4, 0 >\}$. It is easily verified that the subset compression scheme gives a sample compression scheme of size 2 for the class C. □

To our knowledge, it is an open question whether there exists a compression scheme of size $O(d)$ for every maximal class of VC dimension d. The structure of maximal classes of VC dimension d is discussed further in [F89]. Because every class of VC dimension d can be embedded in a maximal class of VC dimension d, it follows that if there was a sample compression scheme of size d for every *maximal* class of VC dimension d, then there would be a sample compression scheme of size d for *every* class of VC dimension d.

8 Conclusions and related work

In this paper we described sample compression schemes within the context of pac-learning; we showed that for any finite concept class C there is a sample compression scheme of size $\log |C|$. For every maximum class of VC dimension d there is a sample compression scheme of size d; 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. We have given a greedy compression algorithm that implements the VC Compression Scheme for maximum classes of VC dimension d.

We have shown that for any class C with a sample compression scheme of size d, where each compression set contains exactly d examples, the sample compression scheme can be used as a pac-learning algorithm for that class, requiring at most

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

examples for $0 < \beta < 1$. Because we have given a suitable sample compression scheme of size d for maximum classes of VC dimension d, this result applies to all maximum classes of VC dimension d. This approach improves on the previously-known sample complexity for pac-learning for maximum classes of VC dimension d [BEHW89] [SAB89].

It is an open question whether there is a sample compression scheme of size d, or of size $O(d)$, for every maximal class of VC dimension d. We gave a sample compression scheme of size d that applies for at least *some* classes that are maximal but not maximum of VC dimension d. We defined randomly-generated classes of VC dimension d, and showed that for the parameters that we have investigated a large proportion of randomly-generated classes of VC dimension d are maximal but not maximum.

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 E^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, [F89] discusses classes defined by n-dimensional vector spaces of real functions on some domain X. Such classes include balls in E^{n-1}, positive halfspaces in E^n, and positive sets in the plane defined by polynomials of degree at most $n-1$. With appropriate restrictions to the domain X [F89, 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. For these classes the iterative compression algorithm is *acyclic*; 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. [F89] contains many open questions concerning the use of iterative compression algorithms for pac-learning for maximum and maximal classes.

Finally, there are other definitions of compression schemes that one might consider. In the definition used in this paper the compression function maps every finite set of labeled examples to a $subset$ of at most k labeled examples. (In the original paper [LW86] the compression function mapped every finite $sequence$ of labeled examples to a $subsequence$ of at most k labeled examples. The alternate definition is essentially the same.) From the combinatorial point of view the following definition of compression function might be the most interesting. The compression function maps every finite set of labeled examples to a subset of k examples with their labels removed.[6] It is again an open problem whether there is such a 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.

9 Acknowledgements

This work benefited from discussions with David Haussler, Dick Karp, Nick Littlestone, and Rob Schapire. This work developed from results in the unpublished manuscripts [LW86] and [W87]; we would like to acknowledge again these contributions from Nick Littlestone and Emo Welzl.

References

[CBFH+93] N. Cesa-Bianchi, Y. Freund, D.P. Helmbold, D. Haussler, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Proceedings of the 25th ACM Symposium on the Theory of Computation*, 1993. To appear.

[6] The compression schemes given in sections 3 and 4 can be modified so that the compression set consists of unlabeled examples. However, we don't know how modify the compression scheme for maximum classes.

[BEHW87] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M., "Occam's Razor", *Inf. Proc. Let.*, 24, 1987, pp. 377-380.

[BEHW89] Blumer, A., A. Ehrenfeucht, D. Haussler, and M. Warmuth, "Learnability and the Vapnik-Chervonenkis Dimension," *JACM*, 36(4), pp.929-965, October 1989.

[BL89] Blumer, A., and Littlestone, N., "Learning Faster than Promised by the Vapnik-Chervonenkis Dimension", Discrete Applied Mathematics 24, 1989, p.47-53.

[EHKV87] Ehrenfeucht, A., Haussler, D., Kearns, M., and Valiant, L., "A General Lower Bound on the Number of Examples Needed for Learning", *Proceedings of the 1988 Workshop on Computational Learning Theory*, Morgan Kaufmann, 1988, p. 139-154.

[HLW88] Haussler, D., Littlestone, N., and Warmuth, M. K., "Lower Bounds on PAC learning and the Size of Epsilon-Nets", unpublished notes.

[F89] Floyd, S., "On Space-bounded Learning and the Vapnik-Chervonenkis Dimension," International Computer Science Institute Technical Report TR-89-061, 1989.

[F90] Freund, Y., "Boosting a weak learning algorithm by majority", *Proceedings of the 1990 Workshop on Computational Learning Theory*, p. 202-231., August 1990.

[HSW89] Haussler, D., Sloan, R., and Warmuth, M., "Learning Nested Differences of Intersection Closed Concept Classes", *Proceedings of the 1989 Workshop on Computational Learning Theory*, Morgan Kaufmann, 1989, p.41-56.

[LSW93] Littlestone, N, Schapire, and Warmuth, M., "Hypothesis Schemas", in progress.

[LW86] Littlestone, N, and Warmuth, M., "Relating Data Compression and Learnability", unpublished manuscript, 1986.

[PW90] Pach, J., and Woeginger, G., "Soem New Bounds for Epsilon-Nets," *Proceedings of the Sixth Annual Symposium on Computational Geometry*, Berkeley, California, June 6-8, pp. 10-15, 1990.

[S72] Sauer, N., "On the Density of Families of Sets", Journal of Comb. Th. (A) 13, p. 145-147.

[S90] Schapire, R., "The strength of weak learnability", Machine Learning, 5(2):197-227, 1990.

[SAB89] Shawe-Taylor, J., Anthony, M., and Biggs, N., "Bounding Sample Size with the Vapnik-Chervonenkis Dimension", November 1989.

[V84] Valiant, L.G., "A theory of the learnable", *Comm. ACM*, 27(11), 1984, pp. 1134-42.

[V82] Vapnik, V.N., *Estimation of Dependencies based on Empirical Data*, Springer Verlag, New York, 1982.

[VC71] Vapnik, V.N. and Chervonenkis, A.Ya., "On the Uniform Convergence of Relative Frequencies of Events to their Probabilities", *Th. Prob. and its Appl.*, 16(2), 1971, pp. 264-280.

[W87] Welzl, E., Complete Range Spaces, unpublished notes,

[WW87] Welzl, E., and Woeginger, G., On Vapnik-Chervonenkis Dimension One, unpublished manuscript, 1987.

A Appendix

Proof of Theorem 6.1: Let Y' be a sequence of m examples drawn independently at random according to the distribution P labeled by the concept c. Call any subset A' of at most d examples from Y' a *compression set* if $g(A')$ is consistent with Y'.

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 samples sequences $\langle x_1, \cdots, x_m \rangle$, such that the hypothesis $g(\{\langle x_{t_1}, c(x_{t_1})\rangle, \cdots, \langle x_{t_d}, c(x_{t_d})\}\})$ is consistent with the sample sequence $Y' = \langle\langle x_1, c(x_1)\rangle, \cdots, \langle x_m, c(x_m)\rangle\rangle$. Let U_T contain all sample sequences $\langle x_1, \cdots, x_m \rangle$, where the hypothesis $g(\{\langle x_{t_1}, c(x_{t_1})\rangle, \cdots, \langle x_{t_d}, c(x_{t_d})\}\})$ 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 P, of the symmetric difference of c and h.) The probability that a sample sequence Y' of m examples is drawn, and the hypothesis represented by a sample compression set of d examples from Y' has error more that ϵ, is at most

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

For a particular T, what is an upper bound on the probability $P^m(B_T \cap U_T)$ of drawing m examples, such that $A' = \{\langle x_{t_1}, c(x_{t_1})\rangle, \cdots, \langle x_{t_d}, c(x_{t_d})\}\}$ is a compression set of size exactly d for those m examples, and the hypothesis represented by A' has error greater than ϵ? Because the elements of Y' are drawn independently from the distribution P, for a fixed T we can assume that the d examples of the compression set A' are drawn first. Next the remaining $m - d$ elements of Y' are drawn. If $g(A')$ has error greater than ϵ and is consistent with the remaining $m - d$ elements of Y' then the probability that a single example drawn from X is consistent with $g(A')$ is less than $1 - \epsilon$. The probability that $m - d$ examples drawn from X are consistent with the hypothesis $g(A')$ is less than $(1 - \epsilon)^{m-d}$. Thus

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

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

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

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

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

□