Journal of Machine Learning Research 1 (2007) 1-35 Submitted 9/05; Revised 11/06; Published

Unlabeled Compression Schemes for Maximum Classes*

Dima Kuzmin DIMA@CSE.UCSC.EDU
Manfred K. Warmuth MANFRED@CSE.UCSC.EDU
Computer Science Department
University of California, Santa Cruz

Editor: John Shawe-Taylor

Abstract

Maximum classes of domain size n and VC dimension d have $\binom{n}{\leq d}$ concepts, and this is an upper bound on the size of any such class. We give a compression scheme for any maximum class that represents each concept by a subset of up to d unlabeled domain points and has the property that for any sample of a concept in the class, the representative of exactly one of the concepts consistent with the sample is a subset of the domain of the sample. This allows us to compress any sample of a concept in the class to a subset of up to d unlabeled sample points such that this subset represents a concept consistent with the entire original sample. Unlike the previously known compression scheme for maximum classes (Floyd and Warmuth, 1995) which compresses to labeled subsets of the sample of size equal d, our new scheme is tight in the sense that the number of possible unlabeled compression sets of size at most d equals the number of concepts in the class.

Keywords: compression schemes, VC dimension, maximum classes, one-inclusion graph, combinatorics

1. Introduction

Consider the following type of protocol between a learner and a teacher. Both agree on a domain and a class of concepts (subsets of the domain). For instance, the domain could be the plane and a concept the subset defined by an axis-parallel rectangle (see Figure 1.1). The teacher gives a set of training examples (labeled domain points) to the learner. The labels of this set are consistent with a concept (rectangle) that is hidden from the learner. The learner's task is to predict the label of the hidden concept on a new test point.

Intuitively, if the training and test points are drawn from some fixed distribution, then the labels of the test point can be predicted accurately provided the number of training examples is large enough. The sample size should grow with the inverse of the desired accuracy and with the complexity or "dimension" of the concept class. The most basic notion of dimension in this context is the Vapnik-Chervonenkis dimension. This dimension is the size d of the maximum cardinality set such that all 2^d labeling patterns can be realized by a concept in the class. The VC dimension of axis-parallel rectangles is 4, since it is possible to label any set of 4 points in all possible ways as long as no point lies inside the

*. Supported by NSF grant CCR CCR 9821087. Some work on this paper was done while the authors were visiting National ICT Australia.

orthogonal hull of the other 3 points (where the orthogonal hull is defined as the smallest rectangle containing the points); also for any 5 points, at least one of the points lies inside the orthogonal hull containing the remaining 4 points and this disallows at least one of the 2^5 patterns.

This paper deals with an alternate notion of dimension used in machine learning that is related to compression (Littlestone and Warmuth, 1986). It stems from the observation that you can often select a subset of the training examples to represent a hypothesis consistent with all training examples. For instance in the case of rectangles, it is sufficient to keep only the uppermost, lowermost, leftmost and rightmost positive point. There are up to 4 points in the subsample (since some of the 4 extreme points might coincide.) The orthogonal hull of the subsample will always be consistent with the entire sample. More generally, a compression scheme is defined by two mappings: one mapping samples of concepts in the class to subsamples, and the other one mapping the resulting subsamples to hypotheses on the domain of the class. Compression schemes must have the property that the subsample always represents a hypothesis consistent with the entire original sample. However note that the reconstructed hypothesis doesn't have to lie in the original concept class. It only needs to be consistent with the original sample. The subsamples represent hypotheses and are called *representatives* in this paper. A compression scheme can be viewed as a set of representatives of hypotheses with the property that every sample of the class contains a representative of a consistent hypothesis. The size of the compression scheme is the size of the largest representative, and the minimum size of a compression scheme for a class serves as an alternate measure of complexity.

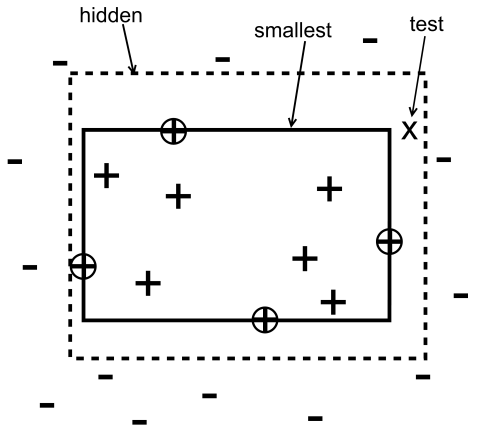

Figure 1.1: An example set consistent with some axis-parallel rectangle. Also shown is the smallest axis-parallel rectangle containing the subsample of circled points. This rectangle is consistent with all examples and the set of circled points represents a consistent concept. The hidden rectangle generating the data is dashed. "x" is the next test point.

Note that in the case of rectangles we need to keep at most 4 points and 4 also is the Vapnik-Chervonenkis dimension of that class. One of the most tantalizing conjectures in learning theory is the following (Floyd and Warmuth, 1995; Warmuth, 2003): *For any concept class of VC dimension d, there is a compression scheme of size at most d.*

The size of the compression scheme also replaces the VC dimension in the PAC sample size bounds (Littlestone and Warmuth, 1986; Floyd and Warmuth, 1995; Langford, 2003). However, in the case of compression schemes, the proofs of these bounds are much simpler. There are many practical algorithms based on compression schemes (e.g. Marchand and Shawe-Taylor (2002, 2003)). Also any algorithm with a mistake bound M leads to a compression scheme of size M (Floyd and Warmuth, 1995).

Let's consider some more illustrative examples of compression schemes. Unions of up to k intervals on the real line form a concept class of VC dimension $2k$. We can compress a sample from this class to the following set of points: the leftmost "+" point in the sample, the leftmost "−" point to the right of the last selected point, the leftmost "+" further to the right of the last selected point, and so forth; stop when there are no more points whose

label is opposite to the last selected point. It is easy to see that at most $2k$ points are kept when the original sample is consistent with a union of k intervals. Also the labels of the entire original sample can be reconstructed from this subsample. Note that in this case the labels of the subsample are always alternating starting with a "+". Thus these labels are redundant and the above scheme can be interpreted as compressing to unlabeled subsamples of size at most the VC dimension $2k$.

Support Vector Machines also lead to a simple labeled compression scheme for halfspaces (sets of the form $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{w} \cdot \boldsymbol{x} \geq b\}$), because only the set of support vectors is needed to reconstruct the hyperplane consistent with the original sample. Of course, the number of support vectors can be quite big. However, it suffices to keep any set of *essential* support vectors and these sets have size $n+1$, where n is the dimension of the feature space (von Luxburg et al., 2004). Not surprisingly, $n+1$ is also the VC dimension of arbitrary halfspaces of dimension n. However, the labels of a set of essential support vectors are not redundant and this provides an example of a labeled compression scheme for halfspaces. There also exists a compression scheme for the same class that compresses to at most $n+1$ *unlabeled points* (Ben-David and Litman, 1998). However this scheme is not constructive.

The compression scheme conjecture is easily proven for intersection-closed concept classes (Helmbold et al., 1992), which include the class of axis-parallel rectangles as a special case. More importantly, the conjecture was shown to be true for any *maximum class*. A finite class of domain size n and of VC dimension d is maximum if its size is equal to the upper bound $\binom{n}{\leq d}$. An infinite class of VC dimension d is maximum if all restrictions to a finite subset of the domain of size at least d are maximum classes of dimension d.

Of the example concept classes discussed so far, the class of up to k intervals on the real line is maximum. The class of halfspaces in $\mathbb{R}^n$ is not maximum, but it is in fact a union of two classes of VC dimension n which are "almost maximum": positive halfspaces and negative halfspaces (Floyd, 1989). *Positive* halfspaces are those that contain the "point" $(\infty, 0, \ldots, 0)$ and negative halfspaces are those that contain $(-\infty, 0, \ldots, 0)$. Both classes of halfspaces are almost maximum in the sense that the restriction to any set of points in general position always produces a maximum class. Finally, the class of axis-parallel rectangles is not maximum since for any five points at least two labelings are not realizable.

In (Floyd and Warmuth, 1995) it was shown that for all maximum classes there always exist compression schemes that compress to *exactly d labeled examples*. In this paper, we give an alternate compression scheme for finite maximum classes. Even though we do not resolve the conjecture for arbitrary classes, we have uncovered a great deal of new combinatorics. Our new

	x_1	x_2	$\mathbf{x_3}$	$\mathbf{x_4}$	$r(c)$
c_1	0	0	0	0	$\emptyset$
$\mathbf{c_2}$	**0**	**0**	**1**	**0**	$\{\mathbf{x_3}\}$
c_3	0	0	1	1	$\{x_4\}$
c_4	0	1	0	0	$\{x_2\}$
c_5	0	1	0	1	$\{x_3, x_4\}$
c_6	0	1	1	0	$\{x_2, x_3\}$
c_7	0	1	1	1	$\{x_2, x_4\}$
c_8	1	0	0	0	$\{x_1\}$
c_9	1	0	1	0	$\{x_1, x_3\}$
c_{10}	1	0	1	1	$\{x_1, x_4\}$
c_{11}	1	1	0	0	$\{x_1, x_2\}$

Figure 1.2: ,Illustration of the unlabeled compression scheme for some maximum concept class. The representatives for each concept are indicated in the right column and also by underlining the corresponding positions in each row. Suppose the sample is $\mathbf{x_3} = \mathbf{1}$,$\mathbf{x_4} = \mathbf{0}$. The set of concepts consistent with that sample is $\{c_2, c_6, c_9\}$. The representative of exactly one of these concepts is entirely contained in the sample domain $\{\mathbf{x_3}, \mathbf{x_4}\}$. For our sample this representative is $\{\mathbf{x_3}\}$ which represents $\mathbf{c_2}$. So the compressed sample becomes $\{\mathbf{x_3}\}$.

scheme compresses any sample consistent with a concept to *at most d unlabeled points* from the sample. If m is the size of the sample, then there are $\binom{m}{\leq d}$ sets of points of size up to d. For maximum classes, the number of different labelings induced on any set of size m is also $\binom{m}{\leq d}$. Thus our new scheme is "tight". In the previous scheme, the number of possible representatives was much bigger than the number of concepts.
The new scheme also has many interesting combinatorial properties. Let us represent finite classes as a binary table (see Figure 1.2) where the rows are concepts and the columns are all the points in the domain. Our compression scheme represents concepts by subsets of size at most d and for any $k \leq d$, the concepts represented by subsets of size up to k will form a maximum class of VC dimension k. Our scheme compresses as follows. After receiving a set of examples we first restrict ourselves to concepts that are consistent with the sample. We will show that for our choice of representatives, there always will be exactly one of the consistent concepts whose representative is completely contained in the sample domain. Thus we simply compress to this representative and use the associated concept as the hypothesis (see Figure 1.2).

Our new unlabeled compression scheme is connected to a certain undirected graph called the *one-inclusion graph* that characterizes the concept class on a set of example points (Haussler et al., 1994): the vertices are the possible labelings of the example points and there is an edge between two concepts if they disagree on a single point. The edges are naturally labeled by the differing points (see Figure 2.1).

Any prediction algorithm can be used to *orient* the edges of the one-inclusion graphs as follows. Assume we are given a labeling of some m points $x_1, \ldots, x_m$ and an unlabeled test point x. If there is still an ambiguity as to how x should be labeled, then this corresponds to an x-labeled edge in one-inclusion graph for the set $\{x_1, \ldots, x_m, x\}$. This edge connects the two possible extensions of the labeling of $x_1, \ldots, x_m$ to the test point x. If the algorithm predicts b, then orient the edge toward the concept that labels x with bit b.

The vertices in the one-inclusion graph represent the possible labelings of $\{x_1, \ldots, x_m, x\}$ produced by the target concepts and if the prediction is averaged over all permutations of the $m+1$ points, then the probability of predicting wrong is $\frac{d}{m+1}$, where d is the out-degree of the target. Therefore the canonical optimal algorithm predicts with an orientation of the one-inclusion graphs that minimizes the maximum out-degree (Haussler et al., 1994; Li et al., 2002) and in (Haussler et al., 1994) it was shown that this outdegree is at most the VC dimension.

How is this all related to our new compression scheme for maximum classes? We show that for any edge labeled with x, exactly one of the two representatives of the incident concepts contains the point x. Thus by orienting the edges toward concept that does not have x, we immediately obtain an orientation of the one-inclusion graphs in which all vertices have maximum outdegree at most d (which is the best possible). Again such a d-orientation immediately leads to prediction algorithms with a worst case expected mistake bound of $\frac{d}{m+1}$, where m is the sample size (Haussler et al., 1994), and this bound is optimal[1] (Li et al., 2002).

The conjecture whether there always exists a compression scheme of size at most the VC

1. Predicting with a d-orientation of the one-inclusion graphs is also conjectured to lead to optimal algorithms in the PAC model of learning (Warmuth, 2004).

dimension remains open. For finite domains it clearly suffices to resolve the conjecture for maximal classes (i.e. classes where adding any concept would increase the VC dimension). We do not know of any natural example of a maximal concept class that is not maximum or closely related. However, it is easy to find small artificial maximal classes (see Figure 1.3). We believe that much of the new methodology developed in this paper for maximum classes will be useful in deciding the general conjecture in the positive and think that in this paper we made considerable progress toward this goal. In particular, we developed a refined recursive structure of finite concept classes and made the connection to orientations of the one-inclusion graphs. Also our scheme constructs a certain unique matching that is interesting in its own right.

Even though the unlabeled compression schemes for maximum classes are tight in some sense, they are not unique. There is a strikingly simple algorithm that always seems to produce a valid unlabeled compression scheme for maximum classes: construct the one-inclusion graph for the domain; iteratively remove a lowest degree vertex and represent a concept c by the set of data points incident to c when vertex c was removed from the graph (see Figure 3.2 for an example run). However, we have no proof of correctness of this algorithm and the resulting schemes do not have as much recursive structure as the ones produced by our recursive algorithm for which we have correctness proof. For the small example given in Figure 1.2, both algorithms can produce the same scheme.

	x_1	x_2	x_3	x_4
c_1	0	0	1	0
c_2	0	1	0	0
c_3	0	1	1	0
c_4	1	0	1	0
c_5	1	1	0	0
c_6	1	1	1	0
c_7	0	0	1	1
c_8	0	1	0	1
c_9	1	0	0	0
c_{10}	1	0	0	1

Figure 1.3: A maximal class of VCdim 2 with 10 concepts. Maximum concept classes of VCdim 2 have $\binom{4}{\leq 2} = 11$ concepts (see Figure 1.2).

Outline of the paper

Some basic definitions are provided in Section 2. We then define unlabeled compression schemes in Section 3 and characterize the properties of the representation mappings of such schemes and their relation to the one-inclusion graph. In this section we also discuss the simple Peeling Algorithm in more detail. This algorithm always seems to provide an unlabeled compression scheme even though we currently do not have a correctness proof for it. Section 4 discusses linear arrangements, which are special maximum concept classes, and discuss how to interpret unlabeled compression schemes for these classes. Next, in Section 5, we briefly summarize the old scheme for maximum classes from (Floyd and Warmuth, 1995) which compresses to labeled subsamples, whereas ours uses unlabeled ones. The core of the paper is in Section 6, where we give a recursive algorithm for constructing an unlabeled compression scheme with a detailed proof of correctness. Section 7 contains additional combinatorial lemmas about the structure of maximum classes and unlabeled compression schemes. In Section 8, we discuss how to possibly extend various compression schemes to maximal classes and how the ideas developed previously might carry over to this more general case. We conclude in Section 9 with a large number of combinatorial open problems that we have encountered in this research.

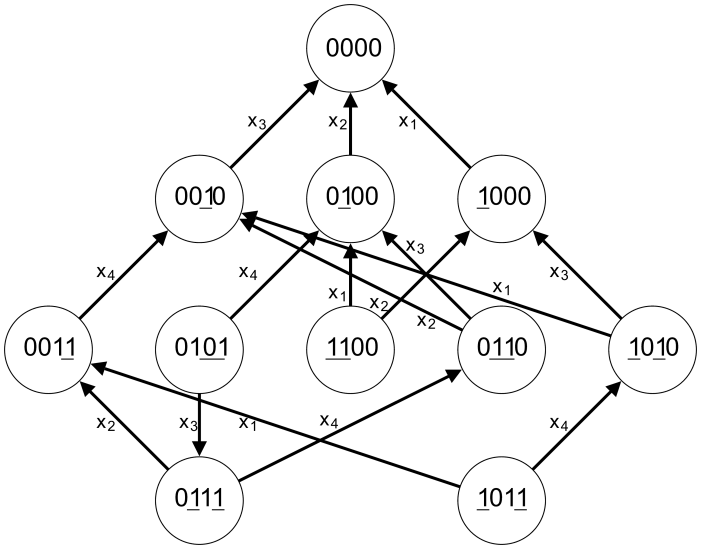

Figure 2.1: One-inclusion graph for the concept class from Figure 1.2. The concepts are the vertices of the graph and edges are labeled with the single differing dimension. Each concept c is given as a bit pattern and the set of underlined dimensions indicates its representative $r(c)$. Arrows show the d-orientation derived from the compression scheme.

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

$C - x_1$

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

C^{x_1}

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

$\mathrm{tail}_{x_1}(C)$

Figure 2.2: The reduction, restriction and the tail of the concept class from Figure 1.2 wrt dimension x_1.

2. Definitions

Let X be a domain, where we allow $X = \emptyset$. A concept c is a mapping from X to $\{0, 1\}$. We can also view a concept c as a characteristic function of a subset of $\mathrm{dom}(c)$, i.e for any domain point $x \in \mathrm{dom}(c)$, $c(x) = 1$ iff $x \in c$. A concept class C is a set of concepts with the same domain (denoted as $\mathrm{dom}(C)$). Such a class is represented by a binary table (see Figure 1.2), where the rows correspond to concepts and the columns to points in $\mathrm{dom}(C)$.

Alternatively, C can be represented as a subgraph of the Boolean hypercube of dimension $|\mathrm{dom}(C)|$. Each dimension corresponds to a particular domain point, the vertices are the concepts in C and two concepts are connected with an edge if they disagree on the label of a single point. This graph is called the *one-inclusion graph* of C (Haussler et al., 1994). Note that each edge is naturally labeled by the single dimension/point on which the incident concepts disagree (see Figure 2.1). The *set of incident dimensions* of a vertex c in a one-inclusion graph G is the set of dimensions labeling the edges incident to c. We denote this set as $I_G(c)$. Its size equals the degree of c in G.

We denote the *restriction* of a concept c onto $A \subseteq \mathrm{dom}(c)$ as $c|A$. This concept has domain A and labels that domain consistently with c. The restriction of an entire class is denoted as $C|A$. This restriction is produced by simply removing all columns not in A

from the table for C and collapsing identical rows.[2] Also the one-inclusion graph for the restriction $C|A$ is now a subgraph of the Boolean hypercube of dimension $|A|$ instead of the full dimension $|C|$. We use $c - x$ as shorthand for $c|(\mathrm{dom}(C) \setminus \{x\})$ and let $C - x$ denote $\{c - x | c \in C\}$ (removing column x from the table, see Figure 2.2). A *sample* of a concept c is any restriction $c|A$ for some $A \subseteq \mathrm{dom}(c)$.

The *reduction* C^x of a concept class C wrt a dimension $x \in \mathrm{dom}(C)$ consists of all those concepts in $C - x$ that have two possible extensions onto concepts in C. All such concepts correspond to an edge labeled with x in the one-inclusion graph (see Figure 2.2). In summary, the class C^x is a subset of $C - x$ and has the same domain $X - \{x\}$.

The *tail* of concept class C on dimension x consists of all concepts that do not have an edge labeled with x. Thus it corresponds to the subset of $C - x$ that has a unique extension onto the full domain. We denote the tail of C on dimension x as $\mathrm{tail}_x(C)$. The class C can therefore be partitioned as $0C^x \,\dot\cup\, 1C^x \,\dot\cup\, \mathrm{tail}_x(C)$, where $\dot\cup$ denotes the disjoint union and bC^x consists of all concepts in C^x extended with bit b in dimension x. Note that tails have the same domain as the original class, whereas the reduction and restriction are classes that have a reduced domain.

A finite set of dimensions $A \subseteq \mathrm{dom}(C)$ is *shattered* by a concept class C if for any possible labeling of A, the class C contains a concept consistent with that labeling (i.e. $\mathrm{size}(C|A) = 2^{|A|}$).[3] The *Vapnik-Chervonenkis dimension* of a concept class C is the size of a maximum subset that is shattered by that class (Vapnik, 1982). We denote this dimension as $\mathrm{VCdim}(C)$. Note that if $|C| = 1$, then $\mathrm{VCdim}(C) = 0$.[4]

In this paper we use the binomial coefficients $\binom{n}{d}$, for integers $n \geq 0$ and d, where $\binom{n}{d} = 0$ for $d > n$ or $d < 0$ and $\binom{0}{0} = 1$. We make use of the following identity which holds for $n > 0$: $\binom{n}{d} = \binom{n-1}{d} + \binom{n-1}{d-1}$. Let $\binom{n}{\leq d}$ be a shorthand for the binomial sums $\sum_{i=0}^{d} \binom{n}{i}$. Then we have a similar identity for the binomial sums when $n > 0$: $\binom{n}{\leq d} = \binom{n-1}{\leq d} + \binom{n-1}{\leq d-1}$.

From (Vapnik and Chervonenkis, 1971) and (Sauer, 1972) we know that for all concept classes with VC dimension d: $|C| \leq \binom{|\mathrm{dom}(C)|}{\leq d}$ (generally known as Sauer's lemma). A concept class C with $\mathrm{VCdim}(C) = d$ is called *maximum* (Welzl, 1987) if for all finite subsets Y of the domain $\mathrm{dom}(C)$, $\mathrm{size}(C|Y) = \binom{|Y|}{\leq d}$. If C is a maximum class with $d = \mathrm{VCdim}(C)$, then $\forall x \in \mathrm{dom}(C)$, the classes $C - x$ and C^x are also maximum classes and have VC dimensions d and $d - 1$, respectively (Welzl, 1987). From this it follows that for finite domains a concept class C is maximum iff $\mathrm{size}(C) = \binom{|\mathrm{dom}(C)|}{\leq d}$.

A concept class C is called *maximal* if adding any other concept to C will increase its VC dimension. Any maximum class on a finite domain is also maximal (Welzl, 1987). However, there exist finite maximal classes, which are not maximum (see Figure 1.3 for an example).

From now on we only consider finite classes. As our main result we construct an unlabeled compression scheme for any finite maximum class. Existance of an unlabeled scheme for infinite maximum classes follows from a compactness theorem given in (Ben-David and Litman, 1998). The proof of that theorem is, however, non-constructive.

2. We define $c|\emptyset = \emptyset$. Note that $C|\emptyset = \{\emptyset\}$ if $C \neq \emptyset$ and $\emptyset$ otherwise.
3. The notations $\mathrm{size}(A)$ and $|A|$ both denote the number of elements in set A.
4. $\mathrm{VCdim}(\{\emptyset\}) = 0$ and $\mathrm{VCdim}(\emptyset)$ is defined to be -1.

3. Unlabeled Compression Scheme

Our unlabeled compression scheme for maximum classes represents the concepts as *unlabeled* subsets of $\mathrm{dom}(C)$ of size at most d. For any $c \in C$ we call $r(c)$ its *representative*. Intuitively we want concepts to disagree on their representatives. We say that two different concepts *clash* wrt r if $c|\,(r(c) \cup r(c')) = c'|\,(r(c) \cup r(c'))$.

Main definition: A *representation mapping* r of a maximum concept class C must have the following two properties:

1. r is a bijection between C and (unlabeled) subsets of $\mathrm{dom}(C)$ of size at most $\mathrm{VCdim}(C)$ and
2. no two concepts in C clash wrt r.

The following lemma shows how the non-clashing requirement can be used to find a unique representative for each sample.

Lemma 3.1 *Let r be any bijection between a finite maximum concept class C of VC dimension d and subsets of $\mathrm{dom}(C)$ of size at most d. Then the following two statements are equivalent:*

1. *No two concepts clash wrt r.*
2. *For all samples s from C, there is exactly one concept $c \in C$ that is consistent with s and $r(c) \subseteq \mathrm{dom}(s)$.*

Based on this lemma it is easy to see that a representation mapping r for a maximum concept class C defines a compression scheme as follows. For any sample s of C we *compress* s to the unique representative $r(c)$ such that c is consistent with s and $r(c) \subseteq \mathrm{dom}(s)$. Reconstruction is even simpler, since r is bijective: if s is compressed to the set $r(c)$, then we reconstruct to the concept c. See Figure 1.2 for an example of how compression and reconstruction work.

Proof of Lemma 3.1

$2 \Rightarrow 1$: Proof by contrapositive. Assume $\neg 1$, i.e. there $\exists c, c' \in C$, $c \neq c'$ s.t. $c|r(c) \cup r(c') = c'|r(c) \cup r(c')$. Then let $s = c|r(c) \cup r(c')$. Clearly both c and c' are consistent with s and $r(c), r(c') \subseteq \mathrm{dom}(s)$. This negates 2.

$1 \Rightarrow 2$: At a high level, for any sample domain $\mathrm{dom}(s)$ there are as many representatives $r(c) \subseteq \mathrm{dom}(s)$ as there are different samples having that domain. The no clashing condition implies that all concepts with representatives in $\mathrm{dom}(s)$ are different from each other on $\mathrm{dom}(s)$, thus every sample has to get at least one representative.

For a more detailed proof assume $\neg 2$, i.e. there is a sample s for which there are either zero or (at least) two consistent concepts c for which $r(c) \subseteq \mathrm{dom}(s)$. If two concepts $c, c' \in C$ are consistent with s and $r(c), r(c') \subseteq \mathrm{dom}(s)$, then $c|r(c) \cup r(c') =$

$c'|r(c) \cup r(c')$ (which is $\neg 1$). If there is no concept c consistent with s for which $r(c) \subseteq \mathrm{dom}(s)$, then since

$$\mathrm{size}(C|\,\mathrm{dom}(s)) = \binom{|\,\mathrm{dom}(s)|}{\leq d} = |\{c : r(c) \subseteq \mathrm{dom}(s)\}| \ .$$

there must be another sample s' with $\mathrm{dom}(s') = \mathrm{dom}(s)$ for which there are two such concepts. So again $\neg 1$ is implied. □

Once we have a valid representation mapping for some maximum concept class C, we can easily derive a valid mapping for any restriction of the class $C|A$ by compressing every restricted concept. This is discussed in the following corollary.

Corollary 3.2 *For any maximum class C and $A \subseteq \mathrm{dom}(C)$, if r is a representation mapping for C then a representation mapping for $C|A$ can be constructed as follows. For any $c \in C|A$, let $r_A(c)$ be the representative of the unique concept $c' \in C$, such that $c'|A = c$ and $r(c') \subseteq A$.*

Proof The construction of the mapping for $C|A$ essentially tells us to treat the concept c as a sample from C and to compress it. Thus we can apply Lemma 3.1 to see that $r_A(c) \subseteq A$ is always uniquely defined. Now we need to show that r_A satisfies the conditions of the Main Definition. Since the representatives $r_A(c)$ are subsets of A, the non-clashing property for the representation mapping r_A for $C|A$ follows from the non-clashing condition for r for C. The bijection property follows from a counting argument like the one used in the proof of Lemma 3.1, since $\mathrm{size}(C|A) = \mathrm{size}(\{r(c) \text{ s.t. } r(c) \subseteq A\})$. ■

The following lemmas and corollaries will be stated only for the concept class C itself. However, in light of Corollary 3.2 they will also hold for any restrictions $C|A$.

We first show that a representation mapping r for a maximum classes can be used to derive a d-orientation for the one-inclusion graph of the class (i.e. an orientation of the edges such that the outdegree of every vertex is at most d). As discussed in the introduction such orientations lead to a prediction algorithm with a worst-case expected mistake bound of $\frac{d}{t}$ at trial t.

Lemma 3.3 *For any representation mapping r of a maximum concept class C and the one-inclusion graph of C, any edge $c \overset{x}{-} c'$ in the graph has the property that its associated dimension x lies in exactly one of the representatives $r(c)$ or $r(c')$.*

Proof Since c and c' differ only on dimension x and $c|r(c) \cup r(c') \neq c'|r(c) \cup r(c')$, x lies in at least one of $r(c), r(c')$. Next we will show that x lies in exactly one.

We say an edge *charges* its incident concept if the dimension of the edge lies in the representative of this concept. Every edge charges at least one of its incident concepts and each concept c can receive at most $|r(c)|$ charges. So the number of charges is lower bounded by the number of edges and upper bounded by the total size of all representations. We complete the proof of the lemma by showing that the number of edges equals the total size of all representatives. This means that no edge can charge both of its incident concepts

and each point labeling an edge must lie in exactly one of the representations of its incident concepts.

There are $|C^x|$ edges labeled with dimension x in the one-inclusion graph for C. Since there are n dimensions and C^x is always maximum and of dimension $d-1$, the total number of edges in the graph is $n\binom{n-1}{\leq d-1}$, where $n = |\mathrm{dom}(C)|$, $d = \mathrm{VCdim}(C)$. (This formula is also a special case of Lemma 7.4.) The total size of all representatives is the same number because:

$$\sum_{c\in C}|r(c)| = \sum_{i=0}^{d} i\binom{n}{i} = n\sum_{i=1}^{d}\binom{n-1}{i-1} = n\binom{n-1}{\leq d-1} \ .$$

■

The above lemma lets us orient the one-inclusion graphs for the class.

Corollary 3.4 *For any representation mapping of maximum class C and the one-inclusion graph of C, directing each edge away from the concept whose representative contains the dimension of the edge creates a d-orientation of the one-inclusion graph for the class.*

Proof The outdegree of every concept is equal to size of its representative, which is $\leq d$.■

The lemma also implies that the representatives of concepts are always subsets of the set of incident dimensions in the one-inclusion graphs.

Corollary 3.5 *Any representation mapping r of a maximum class C has the property that for any concept $c \in C$, its representative $r(c)$ is a subset of the dimensions incident to c in the one-inclusion graph for C.*

Proof From the counting argument in the proof of Lemma 3.3 we see that for every $x \in r(c)$ there must exist an edge leaving c in the graph labeled with x. ■

3.1 The Min-Peeling Algorithm

As discussed at the end of the introduction, there is a simple algorithm that always seems to construct a correct representation mapping for any maximum class. The algorithm iteratively removes any lowest degree vertex from the one-inclusion graph for the remaining class and sets the representative $r(c)$ to the set of dimensions of the edges incident to c when c was removed from the graph. The algorithm is formally stated in Figure 3.1. An illustration of several iterations of the algorithm is given in Figure 3.2.

Unfortunately, we do not have a proof that this algorithm always produces a correct unlabeled compression scheme for maximum classes. As one of the steps in the correctness proof we would need the following: By iteratively removing the lowest degree vertex from a maximum class, we never arrive at a subgraph whose lowest degree vertex has a degree larger than the VC dimension of the remaining class. A natural conjecture is that any class of VC dimension d has a vertex of degree at most d in its one-inclusion graph. However, an elegant counterexample to this conjecture was constructed in (Rubinstein et al., 2007a). Note that their counterexample is not a maximum class and thus it does not contradict the Min-Peeling algorithm. Maximum classes and classes obtained by peeling them appear to

Min-Peeling Algorithm
Input: a finite maximum concept class C.
Output: a representation mapping r for C
Let G be the one-inclusion graph for C

While $C \neq \emptyset$

1. Choose a minimum-degree vertex c among those in C
2. $r(c) :=$ set of dimension incident to c in the graph
3. Remove c from G and C

Figure 3.1: The **Min-Peeling** Algorithm for constructing an unlabeled compression scheme for maximum classes.

have a special structure that always ensures existence of a degree $\leq d$ vertex, where d is the VC dimension of the remaining class. However a complete proof of this statement needs to be found.

The representation mappings produced by the Min-Peeling Algorithm have less structure than the representation mappings produced by the Tail Matching Algorithm discussed in Section 6. In particular, they do not necessarily satisfy the condition of Lemma 7.1, which states the subset of concepts corresponding to representations of size up to k forms a maximum class of VC dimension k. For the maximum class of Figure 1.2, both algorithms can produce the same representation mapping.

Also note how the Min-Peeling Algorithm immediately leads to a d-orientation of the one-inclusion graph: as we peel away a vertex, its edges are naturally oriented away from the vertex before the edges are removed. Since each vertex has degree at most d when it is peeled away, the outdegree of each vertex will be at most d. Moreover, the resulting orientation of the one-inclusion graph is acyclic because all edges are oriented from a vertex toward a vertex that is removed later. In other words, the list of vertices produced by the Min-Peeling Algorithm is a topological ordering of the oriented one-inclusion graph (see Figure 3.3). As we shall see later, the Tail Matching Algorithm also produces a topological order of the graph. By Corollary 3.4, every representation mapping induces a d-orientation of the one-inclusion graph. However we found examples where a valid representation mapping for a maximum class induces a cyclic orientation (not shown).

4. Linear Arrangements

In this section we visualize many of our basic notations and unlabeled compression schemes for simple linear arrangements, which are special maximum classes. An unlabeled compression scheme for linear arrangements is also described in (Ben-David and Litman, 1998).

A *linear arrangement* is a collection of oriented hyperplanes in $\mathbb{R}^d$. The cells of the arrangement are the concepts and the planes the dimensions of the concept class. The

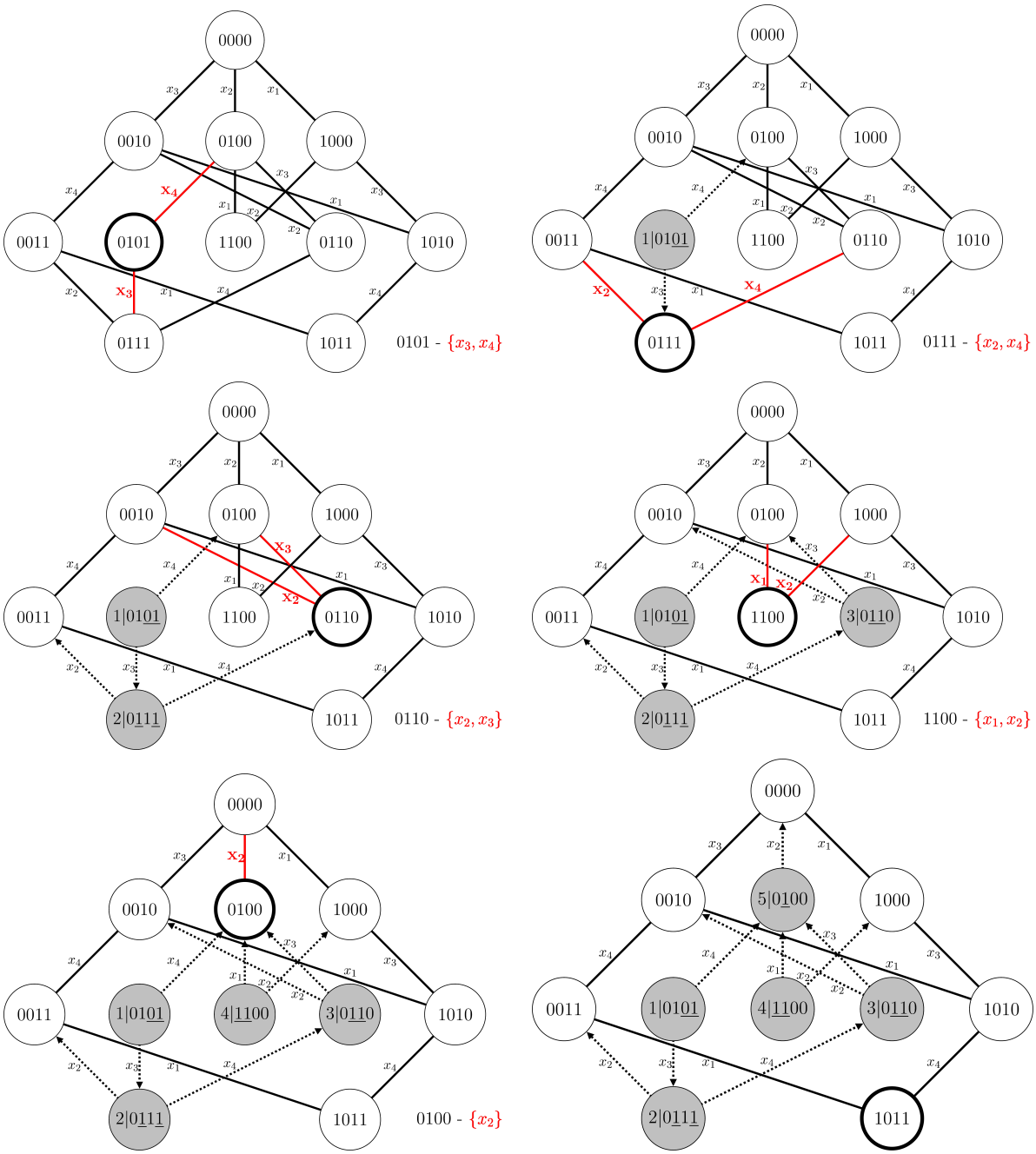

Figure 3.2: Illustration of a run of the **Min-Peeling** algorithm. Each figure corresponds to an iteration of the algorithm: the bold circle indicates the chosen concept to be removed and the grey circles are the concepts that were removed previously. By completing the run, one can produce the compression scheme given in Figure 1.2.

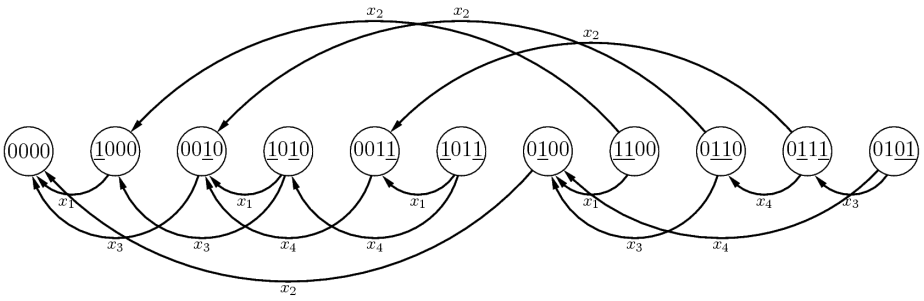

Figure 3.3: Topological order and d-orientation produced by a run of Min-Peeling algorithm for some maximum class.

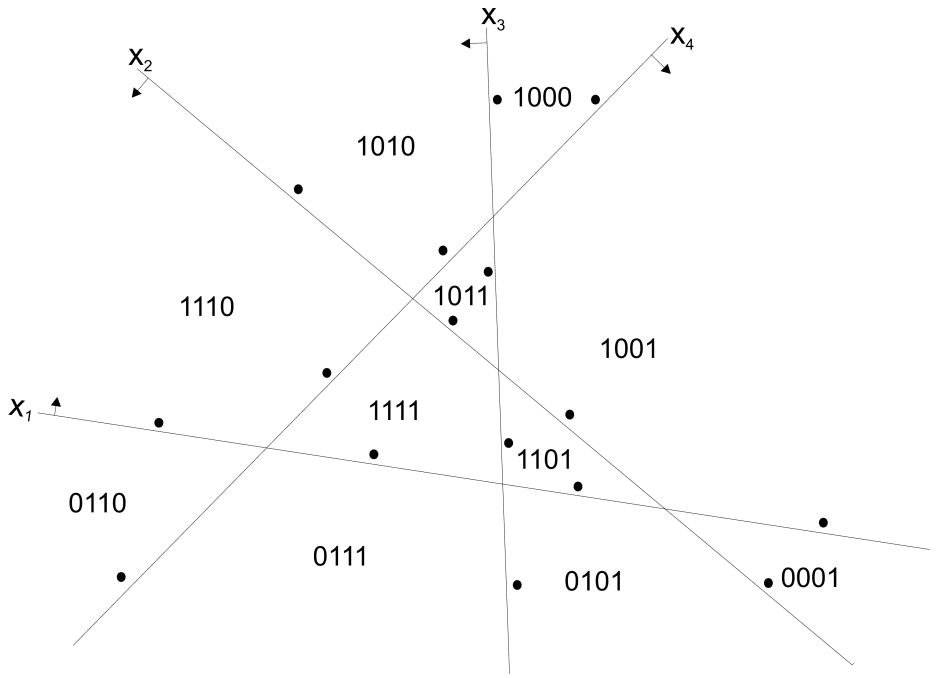

Figure 4.1: An example linear arrangement. The cells of the arrangement represent the concepts and the planes the dimensions of the class. All cells on the upper side of a hyperplane (indicated by an arrow) are labeled one in the corresponding dimension and the cells on the lower side are labeled zero. Up to d hyperplanes bordering a cell are marked and the set of dimensions of these marked planes forms the representative of the cell in the unlabeled compression scheme.

orientations of the planes are indicated by arrows (See Figure 4.1). All cells lying above the plane corresponding to dimension x label x with one, and the cells below label x with zero. If the n planes are in *general position* (any d hyperplanes have a unique point in

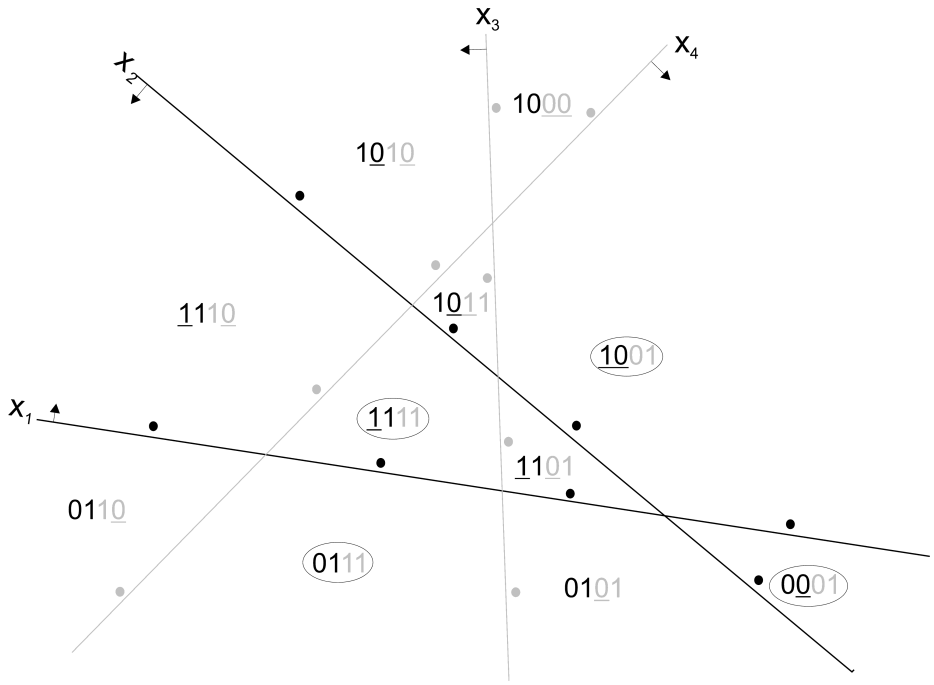

Figure 4.2: Restriction $C - \{x_3, x_4\}$ of the linear arrangements concept class of Figure 4.1. The planes x_3 and x_4 are grayed. Several cells of the full arrangement are combined into bigger cells of the restricted arrangement. One of the subcells forming each big cell (circled) has the property that all dimensions in its representative are still available (none of the corresponding hyperplanes were grayed). The representative of this subcell represents the big cell. For example the subcells $1110, 1111, 1101$ form the larger cell for sample $(x_1, 1), (x_2, 1)$ and only 1111 (circled) is represented by a subset of non-grayed hyperplanes: $r(1111) = \{x_1\}$ and the sample is compressed to $\{x_1\}$.

common and any $d + 1$ hyperplanes have no point in common) then the arrangement is called *simple*. Such arrangements are maximum classes because their VC dimension is $\min\{n, d\}$ and they have $\binom{n}{\leq d}$ cells (Edelsbrunner, 1987). However not all finite maximum classes are representable as linear arrangements (Floyd, 1989).

The vertices of the one-inclusion graph are the cells of the arrangement and edges connect neighboring cells. A restriction $C - x$ corresponds to removing the x plane from the arrangement. Pairs of cells that border this plane are now combined to larger cells (an example of a double restriction is given in Figure 4.2). Samples are combined cells produced by removing some hyperplanes. The reduction C^x is the arrangement in the space of dimension $d - 1$ induced by the projection of the remaining $n - 1$ planes onto the x plane. The subclasses $1C^x$ and $0C^x$ correspond to the cells directly above or below the x plane, respectively. All cells not bordering the x plane form the subclass $\text{tail}_x(C)$.

By Corollary 3.5, the representative of a concept in an unlabeled compression scheme is always a subset of the incident dimensions (here the bordering hyperplane) in the one-inclusion graph. So we can indicate the representatives of each cell by marking the inside borders with the corresponding neighbouring cells (See Figure 4.1). Each cell marks at most d bordering hyperplanes and no cell marks the same set of hyperplanes. The no-clashing condition of our Main Definition means that any two cells are on opposite sides of at least one hyperplane marked by one of the cells. Also by Lemma 3.3, any boundary shared between any two cells is marked on exactly one side.

We now visualize how we compress a sample, i.e. we restate the process described in Figure 1.2 for the special case of linear arrangements. Recall that a sample s corresponds to a combined cell produced by removing the hyperplanes in $\mathrm{dom}(c) \setminus \mathrm{dom}(s)$ where each of the original cells corresponds to a concept consistent with the sample. One (and only one) of the original cells in the combined cell that corresponds to the sample is marking only hyperplanes from the surviving set $\mathrm{dom}(s)$ (circled in Figure 4.2). We compress the sample to that set of marked dimensions and reconstruct based on the represented original cell. Note that if the selected original cell marks any plane, then it must always be at the boundary of the combined cell, since cells in the middle do not border any of the remaining hyperplanes.

It is interesting to observe how our algorithms construct representation mappings for linear arrangements.

Conjecture 4.1. *Sweeping the arrangement with a hyperplane that is not parallel to any plane in the arrangement produces a compression scheme as follows: as soon as a cell is completely swept, it marks the planes of all bordering currently live cells. The resulting sequential assignments of representatives to concepts corresponds to a run of the Min-Peeling Algorithm.*

In particular we conjecture that sweeping as prescribed, iteratively completes minimum degree cells.

The recursive Tail Matching Algorithm of Section 6 chooses some plane x and first finds a compression scheme for the projection C^x of the linear arrangement onto the this plane. Each projected cell from C^x corresponds to two cells, one from $0C^x$ and one from $1C^x$. The algorithm uses the scheme for C^x for all concepts in $0C^x$, i.e. all cells bordering the x plane from below. The sibling cells in $1C^x$ right above the plane receive the same marks but also mark the x plane. The recursive algorithm uses exactly d marks for all vertices in $\mathrm{tail}_x(C)$ (the cells not bordering the x plane). However, this assignment cannot be easily visualized.

Note that one of the planes has the property that the markings produced by the recursive algorithm all lie on one side of the plane. We initially conjectured that there always exist representation schemes that place the marks on the same side for all planes. However we found small counterexamples to this conjecture (not shown).

Simple linear arrangements are known to have the following property: the shortest path between any two cells is always equal to the Hamming distance between the cells (Edelsbrunner, 1987). Surprisingly, we were able to show in Lemma 7.3 that all maximum classes have this property.

5. Comparison with Old Scheme

In the unlabeled compression schemes introduced in this paper each subset of up to d domain points represents a unique concept in the class and every sample of a concept contains exactly one subset that represents a concept consistent with the sample. Before we show that there always exist unlabeled compression schemes, we present the old compression scheme for maximum classes from (Floyd and Warmuth, 1995) in a concise way that brings out the difference between both schemes.

In the old scheme every set of *exactly* d *labeled* points represents a concept. Let u denote such a set of d labeled points. By the properties of maximum classes, the reduction $C^{\mathrm{dom}(u)}$ is a maximum class of VC dimension 0, i.e just a single concept on the domain $\mathrm{dom}(C) \setminus \mathrm{dom}(u)$.[5] Augmenting this concept with any of the 2^d labelings of $\mathrm{dom}(u)$, leads to a concept in C on the full domain. Let c_u denote the concept in C represented by the labeled set u in this way.

Note that there are $2^d\binom{n}{d}$ labeled subsets of size d when the domain size is n, and the number of concepts in the maximum class C is $\binom{n}{\leq d}$. This means that some concepts have multiple representatives in the old scheme. In Figure 5.1 we give both compression schemes for the maximum class used in the previous figures.

We first reason that every concept in C is represented by some labeled subset u of the domain of size d. Since the one-inclusion graph for C is connected (see Gurvits (1997) or Lemma 7.3 of this paper), any concept c has an edge along some dimension x. Therefore, $c - x$ lies in C^x. Inductively we can find a labeled set v of size $d-1$ that represents $c - x$ in C^x. Now let $u = v \cup \{(x, c(x))\}$. Clearly, $c_u = c$ (since $C^{\mathrm{dom}(u)} = (C^x)^{\mathrm{dom}(v)}$).

We still need to show that for every sample s of C with at least d points, there is at least one labeled subset u of size d that represents a concept consistent with the entire sample. Since the restriction $C|\,\mathrm{dom}(s)$ is a maximum class of VC dimension d, it follows from the previous paragraph that there is a labeled subset u representing the concept s of $C|\,\mathrm{dom}(s)$. However, u also represents a concept c in C. It suffices to show that u represents the same concept on $\mathrm{dom}(s)$ wrt both classes C and $C|\,\mathrm{dom}(s)$.

Assume the concept for C labels some point x in $\mathrm{dom}(s) \setminus \mathrm{dom}(u)$ with 0 and the concept for $C|\,\mathrm{dom}(s)$ labels this point with 1. Then from the construction of the representations for C it follows that there are 2^d concepts in C that label x with 0 and $\mathrm{dom}(u)$ in all possible ways. Similarly there are 2^d concepts in $C|\,\mathrm{dom}(s)$ labeling x with 1. The latter concepts extend to concepts in C and therefore the $d+1$ points $\mathrm{dom}(u) \cup \{x\}$ are shattered by class C, which is a contradiction.

6. Tail Matching Algorithm for Constructing an Unlabeled Compression Scheme

The unlabeled compression scheme for any maximum class can be found by the recursive algorithm given in Figure 6.1. There are two "copies" of C^x in the original class, one in which the concepts in C^x are extended in the x dimension with label 0 and one with extension $(x, 1)$. This algorithm first finds a representation mapping r for C^x to subsets of

5. Here $C^{\mathrm{dom}(u)}$ is just the consecutive reduction on all d dimensions in $\mathrm{dom}(u)$. The result of this operation does not depend on the order of the reductions (Welzl, 1987).

x_1	x_2	x_3	x_4	Unlab.	Labeled Representatives
0	0	0	0	$\emptyset$	$\{(x_1,0),(x_2,0)\},\{(x_1,0),(x_3,0)\},\{(x_2,0),(x_3,0)\}$
0	0	1	0	$\{x_3\}$	$\{(x_1,0),(x_3,1)\},\{(x_1,0),(x_4,0)\},\{(x_2,0),(x_3,1)\},\{(x_2,0),(x_4,0)\}$
0	0	1	1	$\{x_4\}$	$\{(x_1,0),(x_4,1)\},\{(x_2,0),(x_4,1)\}$
0	1	0	0	$\{x_2\}$	$\{(x_1,0),(x_2,1)\},\{(x_2,1),(x_3,0)\},\{(x_3,0),(x_4,0)\}$
1	0	0	0	$\{x_1\}$	$\{(x_1,1),(x_2,0)\},\{(x_1,1),(x_3,0)\}$
1	0	1	0	$\{x_1,x_3\}$	$\{(x_1,0),(x_3,1)\},\{(x_1,1),(x_4,0)\}$
1	0	1	1	$\{x_1,x_4\}$	$\{(x_1,1),(x_4,1)\}$
1	1	0	0	$\{x_1,x_2\}$	$\{(x_1,1),(x_2,1)\}$
0	1	0	1	$\{x_3,x_4\}$	$\{(x_3,0),(x_4,1)\}$
0	1	1	0	$\{x_2,x_3\}$	$\{(x_2,1),(x_3,1)\},\{(x_2,1),(x_4,0)\},\{(x_3,1),(x_4,0)\}$
0	1	1	1	$\{x_2,x_4\}$	$\{(x_2,1),(x_4,1)\},\{(x_3,1),(x_4,1)\}$

Figure 5.1: The new unlabeled compression scheme and the old labeled compression scheme for a maximum class.

	Unlabeled	Labeled
Compression	Consider all concepts consistent with the sample and choose the concept whose representative lies completely in the domain of the sample. Compress to that representative	Compress to any set of d labeled points u in the sample such that the single concept $C^{\mathrm{dom}(u)}$ is consistent with the sample
Reconstruction	Predict with the represented concept	Predict with the single concept in $C^{\mathrm{dom}(u)}$ that is extended with the examples from u
# of representatives:	$\binom{n}{\leq d}$	$2^d\binom{n}{d}$

Figure 5.2: Comparison of the two compression schemes.

size up to $d-1$ of $\mathrm{dom}(C) \setminus x$. It then uses this mapping for the $(x,0)$ extension and adds x to all the representatives in the other extension. Finally, the algorithm completes r by finding the representatives for $\mathrm{tail}_x(C)$ via the subroutine given in Figure 6.2.

For correctness, it suffices to show that the constructed mapping satisfies both conditions of our Main Definition. We begin with some additional definitions and a sequence of lemmas.

For $a \in \{0,1\}$ and $c \in C - x$, ac denotes a concept formed from c by extending it with (x,a). It is usually clear from the context what the missing x dimension is. Similarly, aC^x denotes the concept class formed by extending all the concepts in C^x with (x,a). Each dimension $x \in \mathrm{dom}(C)$ can be used to split class C into three disjoint sets: $C = 0C^x \,\dot\cup\, 1C^x \,\dot\cup\, \mathrm{tail}_x(C)$.

A *forbidden labeling* (Floyd and Warmuth, 1995) of a class C of VC dimension d is a labeled set f of $d+1$ points in $\mathrm{dom}(C)$ that is not consistent with any concept in C. We first note that for a maximum class of VC dimension d there is exactly one forbidden

labeling f for each set of $d+1$ dimensions in $\mathrm{dom}(C)$. This is because the restriction $C|\,\mathrm{dom}(f)$ is maximum with dimension d and its size is thus $2^{d+1}-1$. Also if $C=\emptyset$, then $\mathrm{VCdim}(C)=-1$ and the empty set is the only forbidden labeling.

Our Tail Matching Algorithm assigns all concepts in $\mathrm{tail}_x(C)$ a forbidden labeling of the class C^x of size d. Since $c|r(c)$ is now a forbidden labeling for C^x, clashes between the $\mathrm{tail}_x(C)$ and C^x are avoided. If n is the domain size of C, then the number of such forbidden labelings is $\binom{n-1}{d}$. The class $\mathrm{tail}_x(C)$ contains the same number of concepts, since $C-x=C^x\,\dot\cup\,(\mathrm{tail}_x(C)-x)$ and C^x and $C-x$ are maximum classes:

$$|\mathrm{tail}_x(C)|=|C-x|-|C^x|=\binom{n-1}{\leq d}-\binom{n-1}{\leq d-1}=\binom{n-1}{d}. \tag{1}$$

We next show that every tail concept contains some forbidden labeling of C^x and each such forbidden labeling occurs in at least one tail concept. Since any finite maximum class is maximal, adding any concept increases the VC dimension. Adding any concept in $\mathrm{tail}_x(C)-x$ to C^x increases the dimension of C^x to d. Therefore all concepts in $\mathrm{tail}_x(C)$ contain at least one forbidden labeling of C^x. Furthermore, since $C-x$ shatters all sets of size d and $C-x=C^x\,\dot\cup\,(\mathrm{tail}_x(C)-x)$, all forbidden labels of C^x appear in the tail.

We will now show that the Tail Subroutine actually constructs a *matching* between the forbidden labelings of size d for C^x and the tail concepts that contain them. This matching is unique (Theorem 6.5 below) and using these matched forbidden labelings as representatives avoids clashes between tail concepts.

We begin by establishing a recursive structure for the tail (see Figure 6.3 for an example).

Lemma 6.1 *Let C be a maximum class and $x\neq y$ be two dimensions in* $\mathrm{dom}(C)$*. If we denote* $\mathrm{tail}_x(C^y)$ *as* $\{c_i : i\in I\}$ *and* $\mathrm{tail}_x(C-y)$ *as* $\{c_j : j\in J\}$ *(where $I\cap J=\emptyset$),*[6] *then there exist bit values $\{a_i : i\in I\},\{a_j : j\in J\}$ for the y dimension such that* $\mathrm{tail}_x(C)=\{a_ic_i : i\in I\}\,\dot\cup\,\{a_jc_j : j\in J\}$.

Proof First note that the sizes add up as they should (see equation (1) for the tail size calculation):

$$|\mathrm{tail}_x(C)|=\binom{n-1}{d}=\binom{n-2}{d-1}+\binom{n-2}{d}=|\mathrm{tail}_x(C^y)|+|\mathrm{tail}_x(C-y)|.$$

Next we will show that any concept in $\mathrm{tail}_x(C^y)$ and $\mathrm{tail}_x(C-y)$ can be mapped to a concept in $\mathrm{tail}_x(C)$ by extending it with a suitable y bit. We also have to account for the possibility that there can be some concepts $c\in\mathrm{tail}_x(C^y)\cap\mathrm{tail}_x(C-y)$. Concepts in the intersection will need to be mapped back to two different concepts of $\mathrm{tail}_x(C)$.

Consider some concept $c\in\mathrm{tail}_x(C^y)$. Since $c\in C^y$, both extensions $0c$ and $1c$ exist in C. (Note that the first bit is the y position.) If at least one of the extensions lies in $\mathrm{tail}_x(C)$, then we can choose one of the extensions and map c to it. Assume that neither $0c$ and $1c$ lie in $\mathrm{tail}_x(C)$. This means that these concepts both have x edges to some concepts

6. Note that while $C^y\subseteq C-y$, this does not imply that $\mathrm{tail}_x(C^y)\subseteq\mathrm{tail}_x(C-y)$, as the deletion of the concepts $(C-y)\setminus C^y$ from $C-y$ can remove x edges as well, and thus introduce new tail concepts. See Figure 6.3 for an example.

Tail Matching Algorithm
Input: a maximum concept class C
Output: a representation mapping r for C

1. If $\text{VCdim}(C) = 0$ (i.e. C contains only one concept c), then $r(c) := \emptyset$. Otherwise, pick any $x \in \text{dom}(C)$ and recursively find a representation mapping $\tilde{r}$ for C^x.
2. Expand $\tilde{r}$ to $0C^x \cup 1C^x$:

$$\forall c \in C^x : r(c \cup \{x = 0\}) := \tilde{r}(c) \text{ and } r(c \cup \{x = 1\}) := \tilde{r}(c) \cup x$$

3. Extend r to $\text{tail}_x(C)$ via the subroutine of Figure 6.2.

Figure 6.1: The recursive algorithm for constructing an unlabeled compression scheme for maximum classes.

$0c', 1c'$, respectively. But then $c' \in C^y$ and therefore (c, c') forms an x edge in C^y. Thus $c \notin \text{tail}_x(C^y)$, which is a contradiction.

Now consider a concept $c \in \text{tail}_x(C - y)$. It might have one or two y extensions in C. Assume $0c$ was an extension outside of the $\text{tail}_x(C)$. Then this extension has an x edge to some $0c'$ and therefore (c, c') forms an x edge in $C - y$. It follows that all extensions of c will be in the tail.

Finally we need to avoid mapping back to the same concept in $\text{tail}_x(C)$. This can only happen for concepts in $c \in \text{tail}_x(C^y) \cap \text{tail}_x(C - y)$. In this case $0c, 1c \in C$, and by the previous paragraph, both lie in $\text{tail}_x(C)$. So we can arbitrarily map $c \in \text{tail}_x(C^y)$ to $0c$ and $c \in \text{tail}_x(C - y)$ to $1c$. ■

The next lemma shows that the order of the restriction and reduction operations is interchangeable (see Figure 6.4 for an illustration).

Lemma 6.2 *For any maximum class C and two dimensions $x \neq y$ in $\text{dom}(C)$, $C^x - y = (C - y)^x$.*

Proof We first show that $C^x - y \subseteq (C - y)^x$. Take any $c \in C^x - y$. By the definition of restriction, there exists a bit a_y such that $a_y c \in C^x$. Since concepts in C^x always have two extensions in C, it follows that $0a_y c, 1a_y c \in C$. By first restricting these two concepts in y and then reducing in x we have $0c, 1c \in C - y$ and $c \in (C - y)^x$, respectively.

Both $(C-y)^x$ and $C^x - y$ are maximum classes with the same domain size $|\text{dom}(C)| - 2$ and the same VC dimension. Therefore both have the same size, and since $C^x - y \subseteq (C-y)^x$, they are in fact equal. ■

Corollary 6.3 *Any forbidden labeling of $(C - y)^x$ is also a forbidden labeling of C^x.*

Tail Subroutine
Input: a maximum concept class C, $x \in \mathrm{dom}(C)$
Output: an assignment of representatives to $\mathrm{tail}_x(C)$

1. If $\mathrm{VCdim}(C) = 0$ (i.e. $C = \{c\} = \mathrm{tail}_x(C)$), then $r(c) := \emptyset$.
   If $\mathrm{VCdim}(C) = |\mathrm{dom}(C)|$, then $\mathrm{tail}_x(C) = \emptyset$ and $r := \emptyset$.
   Otherwise, pick some $y \in \mathrm{dom}(C)$, $y \neq x$ and recursively find representatives for $\mathrm{tail}_x(C^y)$ and $\mathrm{tail}_x(C - y)$.
2. $\forall c \in \mathrm{tail}_x(C^y) \setminus \mathrm{tail}_x(C - y)$, find $c' \in \mathrm{tail}_x(C)$, s.t. $c' - y = c$, $r(c') := r(c) \cup \{y\}$.
3. $\forall c \in \mathrm{tail}_x(C-y) \setminus \mathrm{tail}_x(C^y)$, find $c' \in \mathrm{tail}_x(C)$, s.t. $c' - y = c$, $r(c') := r(c)$.
4. $\forall c \in \mathrm{tail}_x(C^y) \cap \mathrm{tail}_x(C-y)$, consider the concepts $0c, 1c \in \mathrm{tail}_x(C)$. Let r_1 be the representative for c from $\mathrm{tail}_x(C^y)$ and r_2 be the one from $\mathrm{tail}_x(C - y)$. Suppose, wlog, that $0c|r_1 \cup \{y\}$ is a sample not consistent with any concept in C^x. Then $r(0c) := r_1 \cup \{y\}$, $r(1c) := r_2$.

Figure 6.2: The **Tail Subroutine** for finding tail representatives

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

$C - x_2$

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

C^{x_2}

x_1	x_2	x_3	x_4	
0	1	0	1	$\mathrm{tail}_{x_1}(C - x_2)$
0	1	1	0	$\mathrm{tail}_{x_1}(C^{x_2})$
0	1	1	1	$\mathrm{tail}_{x_1}(C^{x_2})$

$\mathrm{tail}_{x_1}(C)$

Figure 6.3: Illustration of Lemma 6.1. $\mathrm{tail}_{x_1}(C)$ can be composed from $\mathrm{tail}_{x_1}(C^{x_2})$ and $\mathrm{tail}_{x_1}(C - x_2)$; class C is from Figure 1.2, tails in classes are separated by horizontal lines and the last column for $\mathrm{tail}_{x_1}(C)$ indicates whether the concept comes from $\mathrm{tail}_{x_1}(C^{x_2})$ or $\mathrm{tail}_{x_1}(C - x_2)$

Proof By the previous lemma, the forbidden labelings of $(C - y)^x$ and $C^x - y$ are the same. The corollary now follows from the fact that the forbidden labelings of $C^x - y$ are exactly all forbidden labeling of C^x that do not contain y. ■

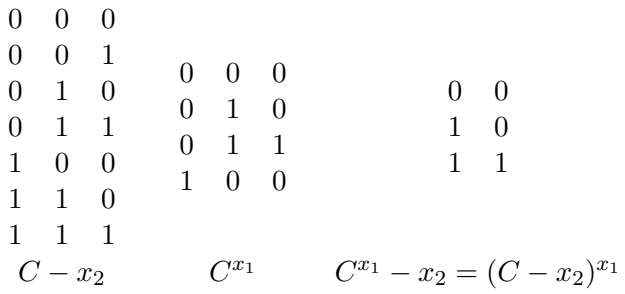

Figure 6.4: Illustration of the statement of Lemma 6.2 - $C^{x_1} - x_2 = (C - x_2)^{x_1}$; class C is given in Figure 1.2

Lemma 6.4 *If we have a forbidden labeling for C^{xy} of size $d-1$, then there exists a bit value for the y dimension such that extending this forbidden labeling with this bit results in a forbidden labeling of size d for C^x.*

Proof We will establish a bijection between forbidden labelings of C^x of size d that contain y and forbidden labelings of size $d-1$ for C^{xy}. Since C^x is a maximum class of VC dimension $d-1$, it has $\binom{n-1}{d}$ forbidden labelings of size d, one for every set of d dimensions from $\mathrm{dom}(C) \setminus x$. Exactly $\binom{n-2}{d-1}$ of these forbidden labelings contain y and this is also the total number of forbidden labelings of size $d-1$ for C^{xy}.

We map the forbidden labelings of size d for C^x that contain y to labelings of size $d-1$ by discarding the y dimension. Assume that a labeling constructed this way is not forbidden in C^{xy}. Then by extending the concept that contains this labeling with both $(y,0)$ and $(y,1)$ back to C^x, we will hit the original forbidden set, thus forming a contradiction.

It follows that every forbidden set is mapped to a different forbidden labeling and by the counting argument above we see that all forbidden sets are covered. Thus the mapping is a bijection and the inverse of this mapping proves the lemma. ■

Theorem 6.5 *Let C be any maximum class C of VC dimension d and domain size n. For any $x \in \mathrm{dom}(C)$ we can construct a bipartite graph between the $\binom{n-1}{d}$ concepts in $\mathrm{tail}_x(C)$ and the $\binom{n-1}{d}$ forbidden labelings of size d for C^x with an edge between a concept and a forbidden labeling if this labeling is contained in the concept. All such graphs have a unique matching.*

Proof The proof will proceed by induction on $n = |\mathrm{dom}(C)|$ and d. To be strictly formal, we can say that we induct on (n,d) pair (the set of pairs having the obvious restriction $n \geq d$), where the set of pairs is ordered in the obvious alphabetic manner. The minimal element of this order is $(0,0)$.

Note that in the Tail Matching Algorithm 6.1 we actually stop when $n = d$, in which case we have a complete hypercube with no tail and the matching is empty. Also, for $d = 0$ there is a single concept which is always in the tail and gets matched to the empty set.

Inductive hypothesis: For any maximum class $\widetilde{C}$, such that $(|\mathrm{dom}(\widetilde{C}|, \mathrm{VCdim}(\widetilde{C})) < (n, d)$ the statement of the theorem holds.

Inductive step. Let $x, y \in \mathrm{dom}(C)$ and $x \neq y$. By Lemma 6.1, we can compose $\mathrm{tail}_x(C)$ from $\mathrm{tail}_x(C^y)$ and $\mathrm{tail}_x(C - y)$. Since $\mathrm{VCdim}(C^x) = d - 1$ and $|\mathrm{dom}(C - x)| = n - 1$,[7] then $(n-1, d), (n-1, d-1) < (n, d)$ and we can use the inductive hypothesis for these classes and assume that the desired matchings already exist for $\mathrm{tail}_x(C^y)$ and $\mathrm{tail}_x(C - y)$.

Now we need to combine these matchings to form a matching for $\mathrm{tail}_x(C)$. See Figure 6.2 for a description of this process. Concepts in $\mathrm{tail}_x(C - y)$ are matched to forbidden labelings of $(C - y)^x$ of size d. By Lemma 6.3, any forbidden labeling of $(C - y)^x$ is also a forbidden labeling of C^x. Thus this part of the matching transfers to the appropriate part of $\mathrm{tail}_x(C)$ without alterations. On the other hand, $\mathrm{tail}_x(C^y)$ is matched to labelings of size $d - 1$. We can make them labelings of size d by adding some value for the y coordinate. Some care must be taken here. Lemma 6.4 tells us that one of the two extensions will in fact have a forbidden labeling of size d (that includes the y coordinate). In the case where just one of two possible extensions of a concept in $\mathrm{tail}_x(C^y)$ is in the $\mathrm{tail}_x(C)$, there are no problems: the single concept will be the concept of Lemma 6.4, since the other concept lies in C^x and thus does not contain any forbidden labelings. There is also the possibility that both extensions are in $\mathrm{tail}_x(C)$. From the proof of Lemma 6.1 we see that this only happens to the concepts that are in $\mathrm{tail}_x(C^y) \cap \mathrm{tail}_x(C - y)$. Then, by Lemma 6.4, we can figure out which extension corresponds to the forbidden labeling involving y and use that for the $\mathrm{tail}_x(C^y)$ matching. The other extension will correspond to the $\mathrm{tail}_x(C - y)$ matching. Essentially, where before Lemma 6.1 told us to map the intersection $\mathrm{tail}_x(C^y) \cap \mathrm{tail}_x(C - y)$ back to $\mathrm{tail}_x(C)$ by assigning a bit arbitrarily, we now choose a bit in a specific way.

So far we have shown that the matching exists. We still need to verify its uniqueness. From any matching for $\mathrm{tail}_x(C)$ we will show how to construct matchings for $\mathrm{tail}_x(C - y)$ and $\mathrm{tail}_x(C^y)$ with the property that two different matchings for $\mathrm{tail}_x(C)$ will disagree with the constructed matchings for either $\mathrm{tail}_x(C - y)$ or $\mathrm{tail}_x(C^y)$. Now, uniqueness follows by induction.

Consider any concept c in $\mathrm{tail}_x(C)$, such that $c - y \in \mathrm{tail}_x(C^y) \setminus \mathrm{tail}_x(C - y)$. Then $c - y$ lies in $C - y$, but not in $\mathrm{tail}_x(C - y)$. Therefore $c - y$ must belong to either $0(C - y)^x$ or $1(C - y)^x$, which means that this concept cannot contain a forbidden set for $(C - y)^x$. We claim that any forbidden set of c for C^x must contain y. Otherwise such a set would be forbidden for $C^x - y$, which by Lemma 6.2 equals $(C - y)^x$. By a similar argument, concepts $c \in \mathrm{tail}_x(C)$, such that $c - y \in \mathrm{tail}_x(C - y) \setminus \mathrm{tail}_x(C^y)$ have to be matched to forbidden sets that do not contain y (since a forbidden set of size d for C^x that contains y, becomes a forbidden set of size $d - 1$ for C^{xy} just by removing y, and condition $c - y \notin \mathrm{tail}_x(C^y)$ implies that c does not contain any such forbidden sets).

From these two facts it follows that if a concept in $\mathrm{tail}_x(C)$ is matched to a forbidden set containing y, then $c - y \in tail_x(C^y)$, and if it is matched to a set not containing y, then $c - y \in tail_x(C - y)$. We conclude that a matching for $\mathrm{tail}_x(C)$ splits into a matching for $\mathrm{tail}_x(C - y)$ and a matching for $\mathrm{tail}_x(C^y)$. This implies that if there are two matchings for all of $\mathrm{tail}_x(C)$, then there are either two matchings for $\mathrm{tail}_x(C - y)$ or two matchings for $\mathrm{tail}_x(C^y)$. ■

7. $\mathrm{VCdim}(C - x) = d$, unless $n = d$, in which case it would obviously drop by one as well.

Theorem 6.6 *The Tail Matching Algorithm of Figure 6.1 returns a representation mapping that satisfies both conditions of the Main Definition.*

Proof Proof by induction on $d = \text{VCdim}(C)$. The base case is $d = 0$: this class has only one concept which is represented by the empty set.

The algorithm recurses on C^x and $\text{VCdim}(C^x) = d - 1$. Thus we can assume that it has a correct representation mapping for C^x that uses sets of size at most $d - 1$ for the representatives.

Bijection condition: The representation mapping for C is composed of a bijection between $1C^x$ and all sets of size $\leq d$ containing x, a bijection between $0C^x$ and all sets of size $< d$ that do not contain x, and finally a bijection between $\text{tail}_x(C)$ sets of size equal d that do not contain x.

No clashes condition: By the inductive assumption there cannot be any clashes internally within each of the subclasses $0C^x$ and $1C^x$, respectively. Clashes between $0C^x$ and $1C^x$ cannot occur because such concepts are always differentiated on the x bit and x belongs to all representatives of $1C^x$. By Theorem 6.5, we know that concepts in the tail are assigned to representatives that define a forbidden labeling for C^x. Therefore, clashes between $\text{tail}_x(C)$ and $0C^x$, $1C^x$ are avoided. Finally, we need to argue that there cannot be any clashes internally within the tail. By Theorem 6.5, the matching between concepts in $\text{tail}_x(C)$ and forbidden labeling of C^x is unique. So if this matching resulted in a clash, i.e. $c_1|r_1 \cup r_2 = c_2|r_1 \cup r_2$, then both c_1 and c_2 would contain the forbidden labelings specified by representative r_1 and r_2. By swapping the assignment of forbidden labels between c_1 and c_2 (i.e c_1 is assigned to $c_1|r_2$ and c_2 to $c_2|r_1$) we would create a new valid matching, thus contradicting the uniqueness of the matching. ■

Note that by Corollary 3.4, the unlabeled compression scheme produced by our recursive algorithm induces a d-orientation of the one-inclusion graph: orient each edge away from the concept that contains the dimension of the edge in its representative. As was the case for the orientation produced by the Min-Peeling Algorithm, the resulting orientation is acyclic. As a matter of fact a topological order can be constructed by ordering the concepts of C as follows: $0C^x, 1C^x, \text{tail}_x(C)$. The concept within $\text{tail}_x(C)$ can be ordered arbitrarily and the concepts within $0C^x$ and $1C^x$ are ordered recursively based on the topological order for C^x.

7. Miscellaneous Lemmas

We conclude with some miscellaneous lemmas that highlight the combinatorics underlying the unlabeled compression schemes for maximum classes. The first one shows that the representatives constructed by our Tail Matching Algorithm induce a nesting of maximum classes. This is a special property, because there are cases where the simpler Min-Peeling Algorithm produces a representation mapping that does not have this property (not shown).

Lemma 7.1 *Let C be a maximum concept class with VC dimension d and let r be a representation mapping for C produced by the Tail Matching Algorithm. For $0 \leq k \leq d$, let $C_k = \{c \in C \text{ s. t. } |r(c)| \leq k\}$. Then C_k is a maximum concept class of VC dimension k.*

Proof Proof by induction on d. Base case $d = 0$: the class has only one concept and the lemma clearly holds.

The lemma trivially holds for $k = 0$ or $k = d$. Otherwise let $x \in \mathrm{dom}(C)$ be the first dimension used in the recursion of the Tail Matching Algorithm and assume by induction that the lemma holds for C^x. Consider which concepts in C belong to C_k. Clearly none of the concepts in $\mathrm{tail}_x(C)$ lie in C_k because their representatives are of size $d > k$. From the recursion of algorithm it follows that $C_k = 0C_k^x \cup 1C_{k-1}^x$, that is, it consists of all concepts in $0C^x$ with representatives of size $\leq k$ in the mapping for C^x, plus all the concepts in $1C^x$ with representatives of size $\leq k-1$ in the mapping for C^x. By the inductive assumption, C_k^x and C_{k-1}^x are maximum classes with VC dimension k and $k-1$, respectively. Furthermore, the definition of C_k implies that $C_{k-1}^x \subset C_k^x$.

Since $|C_k| = |0C_k^x| + |1C_{k-1}^x| = \binom{n-1}{\leq k} + \binom{n-1}{\leq k-1} = \binom{n}{\leq k}$, the class C_k has the right size and $\mathrm{VCdim}(C_k) \geq k$. We still need to show that C_k does not shatter any set of size $k+1$. Consider any such set that does not contain x. This set would have to be shattered by $C_k - x = C_k^x \cup C_{k-1}^x = C_k^x$, which is impossible. Now consider any set A of size $k+1$ that does contain x. All the 1 values for the x coordinate happen in the $1C_{k-1}^x$ part of C_k. Thus $A \setminus x$ must be shattered by C_{k-1}^x whose VC dimension is again one too low. ■

We actually proved that the C_k produced by the representation mapping of the Tail Matching Algorithm always satisfy the recurrence $C_k = 0C_k^x \cup 1C_{k-1}^x$. On the other hand there are nestings of maximum concept classes $C_0 \subset C_1 \subset \ldots \subset C_d = C$, where C_k has VC dimension k but the recurrence does not hold (not shown).

Open Problem 7.1. *We do not know whether for any nesting $C_0 \subset C_1 \subset \ldots \subset C_d = C$ of maximum classes, where C_k has VC dimension k, there always exists a representation mapping that induces this nesting.*

We now consider the connectivity of the one-inclusion graphs of maximum classes. It was known previously that they are connected (Gurvits, 1997). We show in Lemma 7.3 that the length of the shortest path between any two concepts in these graphs is always the Hamming distance between the concepts. This property was previously known for the one-inclusion graphs of linear arrangements, which are special maximum classes. The following technical lemma is necessary to prove the property for arbitrary maximum classes.

We use $I_C(c)$ to denote the set of dimensions incident to c in the one-inclusion graph for C and let $E(C)$ denote the set of all edges of the graph.

Lemma 7.2 *For any maximum class C and $x \in \mathrm{dom}(C)$, restricting wrt x does not change the sets of incident dimensions of concepts in $\mathrm{tail}_x(C)$, i.e. $\forall c \in \mathrm{tail}_x(C)$, $I_C(c) = I_{C-x}(c-x)$.*

Proof Let (c, c') be any edge leaving a concept $c \in \mathrm{tail}_x(C)$. By the definition of $\mathrm{tail}_x(C)$, this edge cannot be an x edge, and therefore c and c' agree on x and $(c-x, c'-x)$ is an edge in $C - x$. It follows that $I_C(c) \subseteq I_{C-x}(c-x)$ when $c \in \mathrm{tail}_x(C)$.

If $I_C(c)$ is a strict subset of $I_{C-x}(c-x)$ for some $c \in \mathrm{tail}_x(C)$, then the number of edges incident to $\mathrm{tail}_x(C) - x = (C-x) \setminus C^x$ in $C-x$ is larger than the number of edges incident to $\mathrm{tail}_x(C)$ in C. The first number is a difference between the sizes of edge sets of the two maximum classes $C - x$ and C^x. Recall that if C is maximum on domain of size n and

has VC dimension d, then its edge set $E(C)$ has size $n\binom{n-1}{\leq d-1}$ (see proof of Lemma 3.3 or Lemma 7.4). Thus the first number is

$$\begin{aligned}|E(C-x)| - |E(C^x)| &= (n-1)\binom{n-2}{\leq d-1} - (n-1)\binom{n-2}{\leq d-2} \\ &= (n-1)\binom{n-2}{d-1} = d\binom{n-1}{d}.\end{aligned}$$

Furthermore, the second number is the number of edges in C minus the number of intra edges in $0C^x$ and $1C^x$, respectively, minus the number of cross edges between $0C^x$ and $1C^x$:

$$\begin{aligned}|E(C)| - 2|E(C^x)| - |C^x| &= n\binom{n-1}{\leq d-1} - 2(n-1)\binom{n-2}{\leq d-2} - \binom{n-1}{\leq d-1} \\ &= (n-1)\left(\binom{n-1}{\leq d-1} - 2\binom{n-2}{\leq d-2}\right) \\ &= (n-1)\left(\binom{n-2}{\leq d-1} - \binom{n-2}{\leq d-2}\right) \\ &= (n-1)\binom{n-2}{d-1} = d\binom{n-1}{d}\end{aligned}$$

Thus the two numbers are the same and we have a contradiction. ■

Lemma 7.3 *In the one-inclusion graph for a maximum concept class C, the length of the shortest path between any two concepts is equal to their Hamming distance.*

Proof The proof will proceed by induction on $|\mathrm{dom}(C)|$. The lemma trivially holds when $|\mathrm{dom}(C)| = 0$ (i.e. $C = \emptyset$). Let c_1, c_2 be any two concepts in a maximum class C of domain size $n > 0$ and let $x \in \mathrm{dom}(C)$. Since $C - x$ is a maximum concept class with a reduced domain size, there is a shortest path P between $c_1 - x$ and $c_2 - x$ in $C - x$ of length equal their Hamming distance. The class $C - x$ is partitioned into C^x and $\mathrm{tail}_x(C) - x$. If $\hat{c}_1$ is the first concept of P in C^x and $\hat{c}_2$ the last, then by induction on the maximum class C^x (also of reduced domain size), there is a shortest path between $\hat{c}_1$ and $\hat{c}_2$ that only uses concepts of C^x. Thus we can assume that P begins and ends with a segment in $\mathrm{tail}_x(C) - x$ and has a segment of C^x concepts in the middle, where some of the three segments may be empty.

We partition $\mathrm{tail}_x(C)$ into $\mathrm{tail}_{x=0}(C)$ and $\mathrm{tail}_{x=1}(C)$. There are no edges between these two sets because they would have to be x edges. There are also no edges between the restrictions $\mathrm{tail}_{x=0}(C) - x$ and $\mathrm{tail}_{x=1}(C) - x$ of the two sets, because by Lemma 7.2 these edges would also exist between the original sets $\mathrm{tail}_{x=0}(C)$ and $\mathrm{tail}_{x=1}(C)$. It follows that any segment of P from $\mathrm{tail}_x(C) - x$ must be from the same part of the tail. Also if the initial segment and final segment of P are both non-empty and from different parts of the tail, then the middle C^x segment cannot be empty.

We can now construct a shortest path P' between c_1 and c_2 from the path P. If $c_1(x) = c_2(x)$ then we extend the concepts in P with $x = c_1(x)$ to obtain a path P' between c_1 and c_2 in C of the same length. Note that from the above discussion, all concepts in the beginning and ending tail segments of P come from the part of the $\mathrm{tail}_x(C)$ that label

x with $c_1(x) = c_2(x)$. Also for the middle segment of P we have the freedom to use label $c_1(x)$.

If $c_1(x) \neq c_2(x)$, then P must contain a concept $\tilde{c}_1$ in C^x, because if all concepts in P lied in $\text{tail}_x(C) - x$ then this would imply an edge between a concept in $\text{tail}_{x=0}(C) - x$ and a concept in $\text{tail}_{x=1}(C) - x$. We now construct a new path P' in C of length $|P| + 1$ which is one more than the Hamming distance $|P|$ between $c_1 - x$ and $c_2 - x$: extend the concepts up to $\tilde{c}_1$ in P with label $c_1(x)$ on x; then cross to the sibling concept $\tilde{c}_2$ which disagrees with $\tilde{c}_1$ only on its x dimension; finally extend the concepts in path P from $\tilde{c}_2$ onward with label $c_2(x)$ on x. ■

We already know that the number of vertices and edges in the one-inclusion graph of a maximum class of domain size n and VC dimension d is $\binom{n}{\leq d}$ and $n\binom{n-1}{\leq d-1}$, respectively. Since vertices and edges are hypercubes of dimension 0 and 1, respectively, these bounds are special cases of the below lemma and corollary, where we bound the number of hypercubes of dimension r, for $0 \leq r \leq d$.

Lemma 7.4 *Let C be any class of domain size n and VC dimension d. Then the number of hypercubes of dimension $0 \leq r \leq d$ which are subgraphs of the one-inclusion graph for C is at most $\binom{n}{r}\binom{n-r}{\leq d-r}$.*

Proof Pick any subset $A \subseteq \text{dom}(C)$ of size r. Recall that C^A consists of all concepts in $C|(\text{dom}(C) - A)$ with the property that all $2^{|A|}$ extensions to the original domain $\text{dom}(C)$ are in C. Thus any concept in the reduced class C^A defines a hypercube of dimension $|A|$ which is a subgraph of the original one-inclusion graph for C. Also from the definition of C^A it follows that all hypercubes that are subgraphs using the dimension set A correspond to a concept in C^A. Note that two hypercubes from the same C^A have no common concepts (vertices), but hypercubes from different restriction sets of the same size may overlap on their vertex set but they are never identical.

From the above discussion it follows that the total number of hypercubes of dimension r is the total size of all C^A, where A has size r. Since the reductions C^A are classes of domain size $n - r$ and VC dimenions at most $d - r$, the inequalies of the lemma follow from Sauer's lemma. ■

Corollary 7.5 *For maximum classes of domain size n and VC dimension d, all $d + 1$ inequalities of the previous lemma are tight. Also for any class C of domain size n and VC dimension d, if one of the inequalities is tight, then C is maximum and they are all tight.*

Proof For maximum classes we have that for any set A of size $0 \leq r \leq d$, the reduction C^A of C is also a maximum class on domain size $n - r$ and VC dimension $d - r$ (Welzl, 1987), (Floyd and Warmuth, 1995). Therefore for maximum classes all inequalities are tight.

Observe that since $|C| = |C^x| + |C - x|$, it follows that if C^x and $C - x$ are maximum, then $|C| = \binom{n-1}{\leq d-1} + \binom{n-1}{\leq d} = \binom{n}{\leq d}$ and C is maximum as well.

If the inequality of the previous lemma is tight for some size r, then for all sets of this size, C^A is a maximum class of VC dimension $n - r$. We will show by the usual double

induction on n and d, that in this case C is maximum. Essentially, for C^x the inequality for size $r-1$ is tight, since for all A containing x, $C^A = (C^x)^{(A \setminus x)}$. Furthermore, for $C - x$ the inequality for size r is tight, since for all A not containing x, $(C-x)^A \supseteq C^A - x$ and $C^A - x$ is maximum because C^A is maximum. ■

If the following lemma could be proven for any concept class produced by peeling minimum degree vertices off a maximum class, then this would be sufficient to prove the non-clashing condition for the representation map produced by the Min-Peeling Algorithm. However the current proof only holds for maximum classes, which is the base case.

Lemma 7.6 *In a maximum class C the labeling of the set of incident dimensions of any concept c uniquely identifies the concept, i.e.:*

$$\forall c' \in C \,:\, c' \neq c \;\Leftrightarrow\; c|I_C(c) \neq c'|I_C(c). \tag{2}$$

Proof We employ an induction on $|\mathrm{dom}(C)|$. The base case is $|\mathrm{dom}(C)| = \mathrm{VCdim}(C)$. In this case, C is a complete hypercube. Note that if $I_C(c) = \mathrm{dom}(C)$, then $c|I_C(c) = c|\,\mathrm{dom}(C) = c$ and equation (2) follows from the uniqueness of each concept. In the hypercube all concepts have this property.

For the general case, if $I_C(c) \neq \mathrm{dom}(C)$ pick $x \notin I_C(c)$ for which $c \in \mathrm{tail}_x(C)$. We have to show that $\forall c' \neq c$, $c|I_C(c) \neq c'|I_C(c)$. Consider the maximum concept class $C - x$ and its concept $c - x$. Because of the reduced domain, we know by induction that

$$\forall c'' \in C - x \,:\, c'' \neq c - x \;\Leftrightarrow\; c - x|I_{C-x}(c-x) \neq c''|I_{C-x}(c-x).$$

Since $c'' = c' - x$, for some $c' \in C$, we can let quantification run over $c' \in C$ and the above is equivalent to

$$\forall c' \in C \,:\, c' - x \neq c - x \;\Leftrightarrow\; c - x|I_{C-x}(c-x) \neq c' - x|I_{C-x}(c-x).$$

Also since $c \in \mathrm{tail}_x(C)$ does not have an x edge, $\forall c' \in C \,:\, c' - x \neq c - x$ is equivalent to $\forall c' \in C \,:\, c' \neq c$ and by Lemma 7.2, $I_{C-x}(c-x) = I_C(c)$. This gives us the equivalent statement: $\forall c' \in C \,:\, c' \neq c \;\Leftrightarrow\; c - x|I_C(c) \neq c' - x|I_C(c)$. Finally, since $x \notin I_C(c)$, $c - x|I_C(c) = c|I_C(c)$ and $c' - x|I_C(c) = c'|I_C(c)$, giving us equation (2). ■

Conjecture 7.1. *The above lemma holds for all classes produced by iteratively peeling minimum degree vertices off a maximum class.*

8. Discussion of Possible Compression Schemes for Maximal Classes

This section discusses the possibility of proving the compression scheme conjecture in the general case. Any finite or infinite concept class is called *maximal* if no concept can be added without increasing the VC dimension. Any concept class can be embedded into a maximal class by adding as many concepts to the class as possible until no new concept can be added. Figure 1.3 presents an example of a maximal class. All finite maximum classes

x_1	x_2	x_3	x_4	r
0	0	0	0	$\emptyset$
0	0	1	0	$\{x_3\}$
0	1	0	0	$\{x_2\}$
1	0	0	0	$\{x_1\}$
0	1	1	0	$\{x_2, x_3\}$
1	0	1	0	$\{x_1, x_3\}$
1	1	0	0	$\{x_1, x_2\}$
0	1	1	1	$\{x_1, x_4\}$
1	0	1	1	$\{x_2, x_4\}$
1	1	0	1	$\{x_3, x_4\}$
1	1	1	1	$\{x_4\}$

Table 8.1: A maximal class that does not have a compression scheme with a representation mapping from sets of domain points of size at most $\mathrm{VCdim}(C) = 2$ to concepts. However if we are allow mappings to concepts in and outside of the class, then the no-clashing condiction can still be satisfied and a valid scheme exists. In the given solution, $\{x_4\}$ represents 1111, which is not a concept in the class.

x_1	x_2	x_3	x_4	r
0	0	1	1	$\{x_1, x_2\}, \{x_3, x_4\}$
0	1	0	0	$\{x_3\}$
0	1	0	1	$\{x_2\}$
0	1	1	0	$\{x_1\}$
1	0	0	0	$\{x_2, x_3\}$
1	0	0	1	$\{x_1, x_3\}$
1	0	1	0	$\{x_1, x_2\}$
1	1	0	0	$\{x_1, x_4\}$
1	1	0	1	$\{x_2, x_4\}$
1	1	1	0	$\{x_3, x_4\}$

Table 8.2: A maximal class that does have an unlabeled scheme where concepts in the class have multiple representatives. The no-clashing holds for any representatives of different concepts, and for all samples there is exactly one consistent concept with a representative in the sample domain.

are also maximal, but there are infinite maximum classes which are not maximal (Floyd and Warmuth, 1995). For the rest of this section maximal means: finite, maximal and not maximum.

A natural idea for constructing a compression scheme for any class is to embed that class into some other class, for which a compression scheme is known. However adding any concepts to a maximal class increases its VC dimension, so we would want an embedding that does not increase the VC dimension too much. Whether and how this can be done is an intriguing open problem of its own.

The old labeled compression scheme for maximum classes cannot be extended to maximal classes due to the nature of its mapping between representatives and represented concepts. The old scheme compresses to a labeled set u of size $\mathrm{VCdim}(C) = d$ and u represents the single concept in the d-fold reduction $C^{\mathrm{dom}(u)}$ extended with the d examples of u (See Section 5). In the case of maximal classes, many d-fold reductions will be empty. Thus it is unclear, which concepts the corresponding set of d labeled examples should represent.

Essentially the old scheme relied on the fact that maximum classes are unions of hypercubes of dimension VCdim(C). Maximal classes do not have this property. Their one-inclusion graphs can be disconnected. In particular, there are maximal classes whose one-inclusion graph has several isolated vertices, i.e. vertices with no incident edges (not shown). Nevertheless, it may be possible to somehow cover maximal classes with hypercubes of dimension d or sligthly larger.

Now we consider the possibility of generalizing our new unlabeled compression scheme from finite maximum classes to finite maximal ones. Recall that our scheme has the property that for any sample from the class, there is *exactly one* representative contained within domain of the sample whose concept is consistent with the sample. Of particular importance in achieving this property was the no-clashing condition for the representatives of concepts. The following lemma describes the effect of having non-clashing representatives for arbitrary concept classes.

Lemma 8.1 *Let r be any injection between a finite concept class C of VC dimension d and subsets of* $\mathrm{dom}(C)$ *of size at most d. Then the following two statements are equivalent:*

1. *No two concepts clash wrt r.*
2. *For all samples s from C, there is at most one concept $c \in C$ that is consistent with s and $r(c) \subseteq \mathrm{dom}(s)$.*

Proof If there are two concepts c and c' that are consistent with s and $r(c), r(c') \subseteq \mathrm{dom}(s)$, then the concepts clash because $c|r(c) \cup r(c') = c'|r(c) \cup r(c')$. Conversely, if two concepts c and c' clash, then the sample $c|r(c) \cup r(c')$ is consistent with at least two concepts c and c' that satisfy $r(c), r(c') \subseteq \mathrm{dom}(s)$. ■

Intuitively, the no-clashing condition causes the representatives to be spread out as much as possible in an attempt to cover all the samples of concepts in the class. Lemma 8.1 says that there is *at most one* representative whose concept is consistent with the sample. However we also need the *at least one* condition, which assures that every sample can be compressed. For maximum classes, the latter condition was assured by a counting argument: the number of concepts in $C|\,\mathrm{dom}(s)$ and the number of subsets of size up to d in $\mathrm{dom}(s)$ is the same; also all such subsets must represent some concept and the no-clashing condition assured that these concepts disagreed on $\mathrm{dom}(s)$. It follows that for each sample, there is always at least one representative in its domain that represents a consistent concept.

There are maximal concept classes of VC dimension d that shatter any set of size d (see Table 8.2). There are $|C|$ representatives in total and this is less than the total number of subsets of size up to d (since C is maximal but not maximum). Therefore, for some domain of size d, there are 2^d concepts but less than that many representatives over the domain.

Of course it makes sense to use all subsets of size up to d by assigning some concepts more than one representative. Note that two representatives of the same concept always clash. However we still must avoid clashes between representatives of different concepts. A compression scheme is valid if for any sample domain, the number of concepts on the domain equals the number of concepts that have at least one representative inside the domain. There exists such a scheme with multiple representatives for the maximal class of Table 8.2.

Unfortunately, Table 8.1 presents a maximal concept class that does not have an unlabeled compression scheme of size equal the VC dimension with multiple representatives of concepts in the class. We checked that for any assignment of multiple representatives to concepts in this class, there is always some sample that cannot be compressed, i.e. there is no representative of a consistent concept in the sample domain. On the other hand, it is very easy to produce an unlabeled compression scheme for this class if we let some subsets of size at most d represent hypotheses outside of the class. Note that in the example of Table 8.1, the no-clashing condition is satisfied not only for all concept pairs, but also for the additional hypothesis and any concept in the class.

For the class of Table 8.1, there also is a compression scheme that maps labeled sets of size equal d to just concepts in the class. We do not know whether such schemes exist for arbitrary finite concept classes. However, there is an infinite maximum (but not maximal) concept class of VC dimension one with the following property: there is no scheme when labeled points must represent concepts in the class, but there is a scheme when labeled points can represent hypotheses outside of the class (Eaton, 2005). Also, there is no unlabeled compression scheme for positive halfspaces in $\mathbb{R}^2$ in which the compression sets (of size at most two) represent positive halfspaces (Neylon, 2006a).

As has become apparent, there are many variations of compression schemes. We now give a unified notation for all these variations. A compression scheme for a concept class C is essentially defined by a mapping f from representatives to hypotheses which are arbitrary subsets of $\mathrm{dom}(C)$. Note that the direction of this mapping is opposite to the represenatation mapping r used for maximum classes.

To define the mapping f, we first choose a set R of representatives which are sets of labeled and/or unlabeled sample points in $\mathrm{dom}(C)$. The mapping f maps R to hypotheses, which are abitrary subsets of $\mathrm{dom}(C)$. Note that f is not required to be injective, allowing for multiple representatives of the same hypothesis. The size of the scheme is the maximum size of any set in R. The mapping f produces a compression scheme as follows:

- **Compression.** For any sample s of a concept in C, consider the set of all representatives $r \in R$ that lie in s and compress to any such r for which the hypothesis $f(r)$ is consistent with the sample s.
- **Reconstruction.** Use hypothesis $f(r)$ to reconstruct the labels of the original sample s.
- **Validity.** Mapping f gives a valid compression scheme, if every sample of a concept in C can be compressed as above.

The no-clashing condition seems to be useful to construct compression schemes for maximal classes as well. However, as we shall see, many techniques for assuring this condition for maximum classes are not applicable in the more general case. Recall that our Tail Matching Algorithm used the forbidden sets of C^x to represent the concept in $tail_x(C)$. This immediately prevented clashes between tail concepts and the rest of the class. However in maximal classes the number of tail concept can be larger than the number of forbidden sets of C^x. For example, in the maximal class of Figure 1.3 all tails have size 4, but there are only $3 = \binom{4-1}{2}$ forbidden sets for C^{x_i}.

Of course, we can consider other splits of C into two parts and use forbidden sets of one part as representatives for the other. A natural idea is to split C into a concept class C' of VC dimension one lower than C and a *shell* $C \setminus C'$ of size at most $\binom{n}{d}$, which is the number of forbidden sets when the domain size is n. For maximum classes, such splits always exist (see Lemma 7.1), but we do not know this for maximal classes. However, we found a particular maximal class (not shown) with a split of the above form for which there are two concepts in the shell that contain only one forbidden set and this set is the same. Thus, we have a single forbidden set available for representing two concepts, and therefore using forbidden sets as representatives does not seem to work for maximal classes.

The Min-Peeling Algorithm provided a simple way of constructing a scheme for maximum classes. For maximal classes, this algorithm fails on simple examples. Part of the problem seems to be that maximal classes have too few edges. Thus a potential idea is to add "virtual" edges to maximal classes, so that a richer set of representative is obtained. We hope to add edges so that the representatives produced by the Min-Peeling Algorithm satisfy the no-clashing condition. Our tests along these lines were inconclusive.

We conclude this section by discussing which lemmas of the previous sections still hold for maximal classes. We have already discussed in this section how the no-clashing condition partially carries over to maximal classes. For a maximum class C, both $C - x$ and C^x are again maximum. This recursion lies at the core of many of the techniques. Maximal classes can still be split as $C = 0C^x \dot\cup 1C^x \dot\cup \mathrm{tail}_x(C)$, but now C^x and $C - x$ are not necessarily maximal and they do not have specific sizes. Our Tail Matching Algorithm relied on a further decomposition of the class $\mathrm{tail}_x(C)$ given in Lemma 6.1: for any concept in $\mathrm{tail}_x(C - y)$ or $\mathrm{tail}_x(C^y)$, we can extend these concepts with a y bit to concepts in $\mathrm{tail}_x(C)$. These extensions also exist for maximal classes, but we cannot get all concepts in $\mathrm{tail}_x(C)$ this way. Finally, for maximal classes, the equality of Lemma 7.2 becomes a subset relationship, i.e. $I_C(c) \subseteq I_{C-x}(c - x)$.

9. Conclusions and Combinatorial Open Problems

The main open problem of whether there always exist ompression schemes of size at most the VC dimension still remains open. (For a general discussion of the allowable schemes see Section 8.) In this paper we gave two algorithms for constructing an unlabeled compression scheme for maximum classes. These schemes have many interesting combinatorial properties. We gave a correctness proof for the recursive Tail Matching Algorithm, however the correctness of the simpler Min-Peeling Algorithm still remains to be shown. We already gave a number of conjectures in connection with the latter algorithm. Does sweeping a linear arrangement always correspond to a run of the Min-Peeling Algorithm (Conjecture 4)? Does Lemma 7.6 hold for partially peeled maximum classes (Conjecture 7)? Does any one-inclusion graph of VC dimension d that results from peeling a maximum class always have a vertex of degree at most d? It is already known, that general classes can have minimum degree larger than d (Rubinstein et al., 2007a).

In our empirical tests (not shown) we actually always found at least $d + 1$ vertices of degree at most d in maximum and peeled classes (instead of just one vertex). We were able to prove (not shown) that maximum classes of VC dimension d have at least one vertex of

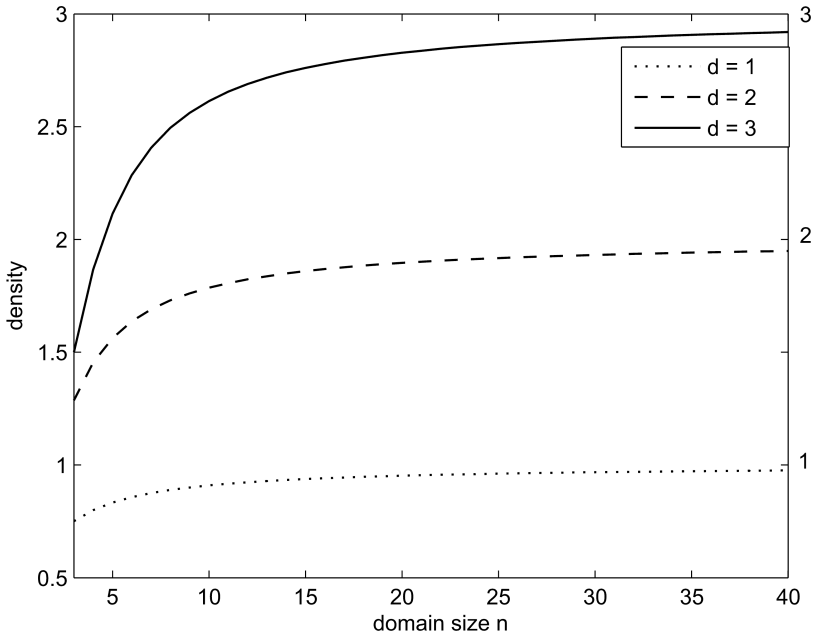

Figure 9.1: Density curves D_d^n (as a function of n) of the one-inclusion graphs of maximum classes with VC dimension $d = 1, 2, 3$.

degree d. However we have not been able to prove that maximum classes have at least $d+1$ such vertices.

Notice that in connection with the last conjecture, the obvious counting arguments are off by a factor of two: since the sum of vertex degrees is equal to twice the number of edges, the density of any graph of minimum degree d has to be bigger than at least $\frac{d}{2}$ and not d. Thus there must be something particular about maximum classes and their peelings that forces them to have a low degree vertex. This is likely related to the fact that maximum classes are unions of dimension d hypercubes.

The *density* of a graph is the ratio of the number of edges to the number of vertices. It is already known that one-inclusion graphs of VC dimension d can have density at most d (Haussler et al., 1994). For a maximum class of domain size n and VC dimension d this density can be expressed as:

$$D_d^n = \frac{n\binom{n-1}{\leq d-1}}{\binom{n}{\leq d}} = \frac{n\sum_{i=0}^{d-1}\frac{i+1}{n}\binom{n}{i+1}}{\binom{n}{\leq d}} = \frac{\sum_{i=1}^{d} i\binom{n}{i}}{\binom{n}{\leq d}} \leq \frac{d\binom{n}{\leq d}}{\binom{n}{\leq d}} = d.$$

Figure 9.1 plots the density curves D_d^n as a function of the domain size n for various values of the VC dimension d. These curves always start at $d/2$, which is the density of the complete hypercube and $\lim_{n\to\infty} D_d^n = d$.

We previously conjectured that maximum classes are the densest. This was recently proven in (Rubinstein et al., 2007b). Specifically, they show that any one-inclusion graph of domain size n and VC dimension d has density at most D_d^n. This is an improvement on the previously known density bound.

Our constructions for unlabeled compression schemes only apply to *finite* maximum classes whereas the original labeled compression scheme for maximum classes is applicable for infinite maximum classes as well. The existence of unlabeled compression schemes for infinite maximum classes of size equal to the VC dimension does follow from the compactness property of compression schemes as shown in (Ben-David and Litman, 1998). That theorem is, however, a non-constructive existence result and thus not completely satisfactory.

One of the most important natural infinite classes is the class of positive halfspaces (halfspaces containing $(\infty, 0, \ldots, 0)$). There are labeled compression schemes for this class that reconstruct only with halfspaces (e.g. compressing to a set of essential support vectors von Luxburg et al. (2004)). An unlabeled compression scheme is also known to exist (Ben-David and Litman, 1998) (via a nonconstructive proof) but it would be interesting to find a simple constructive unlabeled compression scheme for this class. Recall that the VC dimension of positive halfspaces in $\mathbb{R}^n$ is n. For the case of $n = 1, 2$, it is easy to find unlabeled compression schemes (not shown). However for $n = 2$, it is necessary that the sets of size at most two represent hypotheses which are not halfspaces (Neylon, 2006b).[8]

Open Problem 9.1. *Find a constructive unlabeled compression scheme of size n for the class of positive halfspaces in $\mathbb{R}^n$.*

One of the simplest ways to obtain compression schemes for arbitrary classes would be to embed them into maximum classes and then use one of the existing algorithms.

Open Problem 9.2. *For any concept class C, does there always exist a maximum class of VC dimension at most a constant times larger than* $\mathrm{VCdim}(C)$ *that contains C as a subset?*

Acknowledgments

We thank Sally Floyd for her personal encouragement and brilliant insights, Sanjoy Dasgupta for the discussions leading to Lemma 7.3, and Tyler Neylon for the discussion of unlabeled compression schemes for positive halfspaces.

References

Shai Ben-David and Ami Litman. Combinatorial variability of Vapnik-Chervonenkis classes with applications to sample compression schemes. *Discrete Applied Mathematics*, 86:3 – 25, 1998.

8. The construction in (Neylon, 2006b) requires a set of points not in general position, but this requirement can be removed (Neylon, 2006a). Moreover, it is possible to restrict the class of positive halfspaces to an everywhere dense subset of $\mathbb{R}^n$ with the property that all finite subsets of this set are in general position (Neylon, 2006a). This restriction is a natural infinite maximum class with no unlabeled compression scheme that reconstructs with halfspaces.

Frederik Eaton. Private communication, 2005.

Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, New York, 1987. ISBN 038713722X.

S. Floyd. *Space-bounded learning and the Vapnik-Chervonenkis Dimension (Ph.D)*. PhD thesis, U.C. Berkeley, December 1989. ICSI Tech Report TR-89-061.

Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

Leonid Gurvits. Linear algebraic proofs of VC-dimension based inequalities. In Shai Ben-David, editor, *EuroCOLT '97, Jerusalem, Israel, March 1997*, pages 238–250. Springer Verlag, March 1997.

D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0,1\}$ functions on randomly drawn points. *Inform. Comput.*, 115(2):248–292, 1994.

D. Helmbold, R. Sloan, and M. K. Warmuth. Learning integer lattices. *SIAM J. Comput.*, 21(2):240–266, 1992.

John Langford. Tutorial on practical prediction theory for classification. *ICML*, 2003.

Y. Li, P. M. Long, and A. Srinivasan. The one-inclusion graph algorithm is near optimal for the prediction model of learning. *Transaction on Information Theory*, 47(3):1257–1261, 2002.

N. Littlestone and M. K. Warmuth. Relating data compression and learnability. Unpublished manuscript, obtainable at http://www.cse.ucsc.edu/~manfred/pubs/T1.pdf, June 10 1986.

Mario Marchand and John Shawe-Taylor. The Set Covering Machine. *Journal of Machine Learning Research*, 3:723–746, 2002.

Mario Marchand and John Shawe-Taylor. The Decision List Machine. In *Advances in Neural Information Processing Systems 15*, pages 921–928. MIT-Press, Cambridge, MA, USA, 2003.

Tyler Neylon. Private communication, 2006a.

Tyler Neylon. *Sparse Solutions to Linear Prediction Problems*. PhD thesis, New York University, Courant Institute of Mathematical Sciences, May 2006b.

Benjamin I. P. Rubinstein, Peter Bartlett, and J. Hyam Rubinstein. Shifting: One-inclusion mistake bounds and sample compression. Technical Report UCB/EECS-2007-86, EECS Department, University of California, Berkeley, Jun 2007a. URL `http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-86.html`.

Benjamin I.P. Rubinstein, Peter L. Bartlett, and J. Hyam Rubinstein. Shifting, one-inclusion mistake bounds and tight multiclass expected risk bounds. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1193–1200. MIT Press, Cambridge, MA, 2007b.

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

V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.

V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probab. and its Applications*, 16(2):264–280, 1971.

Ulrike von Luxburg, Olivier Bousquet, and Bernard Schölkopf. A compression approach to support vector model selection. *Journal of Machine Learning Research*, 5:293–323, April 2004.

M. K. Warmuth. Compressing to VC dimension many points. In *Proceedings of the 16th Annual Conference on Learning Theory (COLT 03)*, Washington D.C., USA, August 2003. Springer. Open problem.

M. K. Warmuth. The optimal PAC algorithm. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT 04)*, Banff, Canada, July 2004. Springer. Open problem.

E. Welzl. Complete range spaces. Unpublished notes, 1987.