Unlabeled Compression Schemes for Maximum Classes*,**

Dima Kuzmin and Manfred K. Warmuth

Computer Science Department
University of California, Santa Cruz
{dima,manfred}@cse.ucsc.edu

Abstract. We give a compression scheme for any maximum class of VC dimension d that compresses any sample consistent with a concept in the class to at most d unlabeled points from the domain of the sample.

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 interior of an axis-parallel rectangle (see Fig. 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.

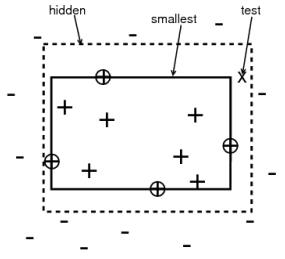

Fig. 1: An example set consistent with some axis-parallel rectangle. Also shown is the smallest axis-parallel rectangle containing the compression set (circled points). This rectangle is consistent with all examples. The hidden rectangle generating the data is dashed. "x" is the next test point.

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. So with axis-parallel rectangles, it is possible to label any set of 4 points in all possible ways as longs as no subset of 3 lies on a line. However for any 5 points, at least one of the points lies inside the smallest rectangle of the remaining 4 points and this

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

disallows at least one of the 2^5 patterns.

The idea of *sample compression* in learning ([LW86]) 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 compression set (since some of the 4 extreme points might coincide.) Assume the compression set represents the smallest rectangle that contains it. This rectangle will always be consistent with the entire sample.

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 ([FW95, War03]): *For any concept class of VC dimension d, there is a compression scheme that keeps at most VC dimension many examples.*

The maximum size of the compression sets also replaces the VC dimension in the PAC sample size bounds ([LW86, FW95, Lan03]). In the case of compression schemes, the proofs of these bounds are strikingly simple. There are many practical algorithms that use compression scheme techniques (e.g. [MST02, SMJST03]). In particular Support Vector Machines can be interpreted as a compression scheme where the essential support vectors are the compression set ([vLBS04]). Also any algorithm with mistake bound M leads to a compression scheme of size M [FW95].

The above conjecture was proven for *maximum classes* over a finite domain, which are classes where the number of concepts coincides with a certain upper bound. In [FW95] it was shown that for such classes there always exist compression schemes that compress to **exactly** d **labeled examples**. In this paper, we give an alternate compression scheme for maximum classes. Even though we do not resolve the conjecture for arbitrary classes, we uncovered a great deal of new combinatorics. Our new 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 labeling 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 all possible compression sets was much bigger than the number of concepts.

Our new scheme reveals a lot of interesting combinatorial structure. Let us represent finite classes as a binary table (see Fig. 2) where the rows are concepts and the columns are the all points in the domain. Our compression scheme represents concepts by subsets of size at most d. For any size $k \leq d$, the concepts represented by subsets of size up to k will be 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 Fig. 2).

Concept classes can also be represented by certain undirected graphs called the *one-inclusion graphs* (see, for instance, [HLW94]): The vertices are the possible labelings of the example points and edges are between concepts that disagree on a single point. Note that each edge is naturally labeled by the single point on which the incident concepts disagree (see Fig. 4). Each 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 edge (with label x) in the one-inclusion graph for $x_1, \ldots, x_m, x$. 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 towards the concept that labels x with bit b.

The vertices in the one-inclusion graph correspond to the possible target concepts and if the prediction is averaged over a random permutation 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 [HLW94, LLS02] and in [HLW94] it was shown that this outdegree is at most the VC dimension d.

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 x. Thus by orienting the edges towards concept that does not have x, we immediately obtain an orientation of the one-inclusion graph with maximum outdegree d (which is the best possible).

	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	1	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	$\{x_1, x_2\}$

Fig. 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 $\{\mathbf{x_3}\}$.

Again such a d-orientation immediately leads to prediction algorithms with expected error $\frac{d}{m+1}$, where m is the sample size [HLW94], and this bound is optimal[1] [LLS02].

[1] Predicting with a d-orientation of the one-inclusion graph is also conjectured to lead to optimal algorithms in the PAC model of learning [War04].

Regarding the general case: It suffices to show the conjecture for maximal classes (i.e. classes where adding any concept would increase the VC dimension). We don't know of any natural example of a maximal concept class that is not maximum even though it is easy to find small artificial cases (see Fig. 3). We believe that much of the new methodology developed in this paper for maximum classes will be useful in resolving the general conjecture in the positive and think that in this paper we made considerable progress towards this goal. In particular, we developed a refined recursive structure of concept classes and made the connection to orientations of the one-inclusion graph. Also our scheme constructs a certain unique matching that is interesting in its own right.

	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

Fig. 3: A maximal classes of VCdim 2 with 10 concepts. Maximum concept classes of VCdim 2 have $\binom{4}{\leq 2} = 11$ concepts (see Fig. 2).

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 produces a valid unlabeled compression scheme for maximum classes: Construct a one-inclusion graph for the whole class; iteratively remove a lowest degree vertex and represent this concept by its set of incident dimensions (see Fig. 4 for an example run). We have no proof of correctness of this algorithm and the resulting schemes do not have as much recursive structure as the one presented in this paper. For the small example given in Fig. 4 both algorithms can produce the same scheme.
Finally, we are reasonably confident that the conjecture holds in general because we did a brute-force search for compression schemes in maximal classes of domain size up to 6. In doing so we noticed that maximal classes have many more solutions than maximum ones.

2 Definitions

Let X be an instance domain (we allow $X = \emptyset$). A concept c is a mapping from X to $\{0, 1\}$. Or we can view c as a characteristic function of a subset of its domain X, denoted as $dom(c)$, where $c(x) = 1$ iff the instance $x \in dom(c)$ lies in c. A concept class C is a set of concepts over the same domain (denoted as $dom(C)$). Such a class is represented by a binary table (see Fig. 2), where the rows correspond to concepts and the columns to the instances.

We denote the *restriction* of a concept c onto $A \subseteq dom(c)$ 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] We use $C - x$ as shorthand for $C|(dom(C) \setminus \{x\})$ (removing column x from the table) and $C - A$ for $C|(dom(C) \setminus A)$ (see Fig. 5). A *sample* of a concept c is any restriction $c|A$ for some $A \subseteq dom(c)$.

[2] We define $c|\emptyset = \emptyset$. Note that $C|\emptyset = \{\emptyset\}$ if $C \neq \emptyset$ and $\emptyset|\emptyset = \emptyset$.

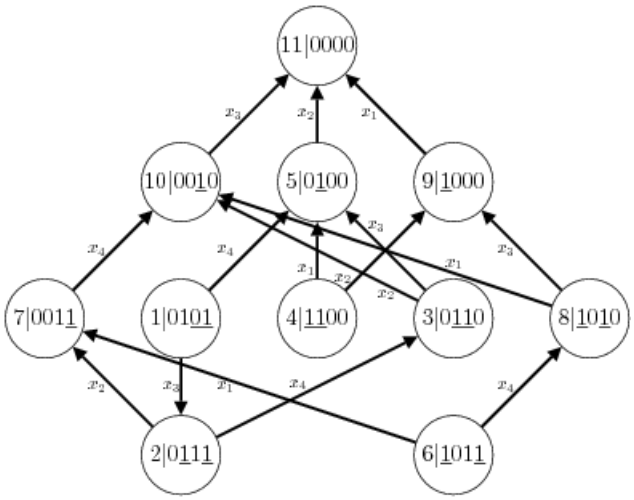

Fig. 4. One-inclusion graph for the concept class from Fig. 2: edges are labeled with the differing dimension. To construct an unlabeled compression scheme, iteratively remove a concept of minimum degree (numbers indicate order of removal). The underlined dimensions indicate the representative of each concept (the incident dimensions when the concept was removed). Arrows show the *d*-orientation derived from the scheme. In this case our recursive algorithm can produce the same 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
$tail_{x_1}(C)$

Fig. 5. the reduction, restriction and the tail of concept class from Fig. 2 wrt dimension x_1

The *reduction* C^x of a concept class C wrt a dimension $x \in dom(C)$ is a special subset of $C - x$ that also has domain $X - \{x\}$. It consists of all those concepts in $C - x$ that have two possible extensions onto concepts in C and thus correspond to an edge labeled with x in the one-inclusion graph (see Fig. 5).

The *tail* of concept class C on dimension x consists of all concepts that don't have an edge labeled with x. We denote this subset of C as $tail_x(C)$. Note that tails have the same domain as the original class.

A finite set of dimensions $A \subseteq dom(C)$, 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. $size(C|A) = 2^{|A|}$).[3] The *Vapnik-Chervonenkis dimension* of a concept class C is the size of a maximum subset that is shattered by that class ([Vap82]). We denote this size with $VCdim(C)$. Note that if $|C| = 1$, then $VCdim(C) = 0$.[4]

[3] $size(A)$ and $|A|$ denote the number of elements in set A

[4] $VCdim(\{\emptyset\}) = 0$ and $VCdim(\emptyset)$ is defined to be -1.

This paper makes some use of binomial coefficients $\binom{n}{d}$, for integers $n \geq 0$ and d.[5] We use 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 $\sum_{i=0}^{d} \binom{n}{i}$. Then we have a similar identity for the binomial sums ($n > 0$): $\binom{n}{\leq d} = \binom{n-1}{\leq d} + \binom{n-1}{\leq d-1}$.

From [VC71] and [Sau72] we know that for all concept classes with $VCdim(C) = d$: $|C| \leq \binom{|dom(C)|}{\leq d}$ (Sauer's lemma). A concept class C with $VCdim(C) = d$ is called *maximum* if $\forall Y \subseteq dom(C), |Y| < \infty : size(C|Y) = \binom{|Y|}{\leq d}$. For finite domains it is sufficient to check just the size of class itself. Additionally, if C is a maximum class with $d = VCdim(C)$, then $\forall x \in dom(C)$, $C - x$ and C^x are also maximum classes with VC dimensions d and $d-1$ respectively ([Wel87]).

A concept class C is called *maximal* if adding any other concept to C will increase its VC dimension. A maximum class on a finite domain is also maximal ([Wel87]). But there exist finite maximal classes, which are not maximum (see Fig. 3 for an example).

From now on we only consider finite classes.

3 Unlabeled Compression Scheme

Our unlabeled compression scheme for maximum classes "represents" the concepts as *unlabeled* subsets of $dom(C)$ 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 subsets of $dom(C)$ of size at most $VCdim(C)$ 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 1. *Let r be any bijection between a finite maximum concept class C of VC dimension d and subsets of $dom(C)$ 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 $r(c) \subseteq dom(s)$.*

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

[5] Boundary values: for $d > n$ or $d < 0$, $\binom{n}{d} = 0$; also $\binom{0}{0} = 1$.

consistent with s and $r(c) \subseteq dom(s)$. Reconstruction is even simpler, since r bijective. If s is compressed to the set $r(c)$, then we reconstruct to the concept c. See Fig. 2 for an example of how compression and reconstruction works.

Proof of Lemma 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 $r(c), r(c') \subseteq dom(s)$. This negates 2.

$1 \Rightarrow 2$: Assume $\neg 2$, i.e. there is a sample s for which there either zero or (at least) two consistent concepts c for which $r(c) \subseteq dom(s)$. If two concepts $c, c' \in C$ are consistent with s and $r(c), r(c') \subseteq dom(s)$, then $c|r(c) \cup r(c') = c'|r(c) \cup r(c')$ (which is $\neg 1$). If there is no concept consistent c with s for which $r(c) \subseteq dom(s)$, then since

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

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

□

We first show that a representation mapping r for a maximum classes can be used to derive a d-orientation of the one-inclusion graph of class (i.e. an orientation of the edges such that the outdegree of every vertex is $\leq d$).

Lemma 2. *For any representation mapping r of a maximum concept class C and any edge $c \xrightarrow{x} c'$, the dimension x is contained 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. The number of edges in C is $N\binom{N-1}{\leq d-1}$, where $N = |dom(C)|$, $d = VCdim(C)$.[6] However, the total size of all representatives is the same number because:

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

This means that no edge can charge both of its incident concepts. □

[6] Number of edges is the size of C^x times the domain size

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

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

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

3. Extend r to $tail_x(C)$ via the recursive process described in Fig. 7.

Fig. 6. the recursive algorithm for constructing an unlabeled compression scheme for maximum classes

Corollary 1. *For any representation mapping of a maximum class, 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 $\leq d$. □

4 Recursive Algorithm for Constructing a Compression Scheme

The unlabeled compression scheme for any maximum class can be found by the recursive algorithm given in Fig. 6. This algorithm first finds a representation mapping r for C^x (to subsets of size up to $d-1$ of $dom(C) - x$). It then uses this mapping for one copy of C^x in C and adds x to all the representatives in the other copy. Finally the algorithm completes r by finding the representatives for $tail_x(C)$ via yet another recursive procedure given in Fig. 7.

To prove the correctness of this algorithm (i.e. show that the constructed mapping satisfies both conditions of the main definition) we need some additional definitions and a sequence of lemmas.

Let aC^x, $a \in \{0, 1\}$ denote a concept class formed by extending all the concepts in C^x back to $dom(C)$ by setting the x dimension to a. Similarly, if $c \in C^x$ or $c \in C - x$, then ac denotes a concept formed from c by extending it with the x dimension set to a. It is usually clear from the context what the missing dimension is. Each dimension $x \in dom(C)$ can be used to split class C into three disjoint sets: $C = 0C^x \dot{\cup} 1C^x \dot{\cup} tail_x(C)$.

A *forbidden labeling* [FW95] for some class C is a sample s with $dom(s) \subseteq dom(C)$ 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 for each set A of $d+1$ dimensions. This is because $C|A$ is maximum with dimension

Recursive Tail Algorithm
Input: a maximum concept class C, $x \in dom(C)$
Output: an assignment of representatives to $tail_x(C)$

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

Fig. 7. the **Recursive Tail** Algorithm for finding tail representatives

d and its size is thus $2^{d+1} - 1$. Our recursive procedure for the tail assigns all concepts in $tail_x(C)$ a forbidden label of C^x (i.e. $c|r(c)$ is a forbidden labeling for C^x of size d). Then clashes between the $tail_x(C)$ and C^x are automatically prevented.

Note the number of such forbidden labelings is $\binom{n-1}{d}$ and we will now reason that $tail_x(C)$ is of the same size. $|dom(C)| = n$. Since $C - x = C^x \dot{\cup} tail_x(C) - x$ and C^x and $C - x$ are maximum classes, we have ($n = |dom(C)|$)

$$|tail_x(C)| = |C - x| - |C^x| = \binom{n-1}{\leq d} - \binom{n-1}{\leq d-1} = \binom{n-1}{d} \ .$$

We now reason that every tail concept contains some forbidden labeling of C^x (of size d) and each forbidden labeling occurs in some tail concept. Since any finite maximum class is maximal, adding any concept increases the VC dimension. Adding any concept in $tail_x(C)$ to C^x increases the dimension of C^x to d. Therefore all concepts in $tail_x(C)$ contain at least one forbidden labeling of size d for C^x. Furthermore, since $C - x$ shatters all sets of size d and $C - x = C^x \dot{\cup} tail_x(C) - x$ all forbidden labels of C^x appear in the tail. Our recursive procedure for the tail actually construct a *matching* between forbidden labelings of size d for C^x and tail concepts that contain them. It remains to be shown that such

1. the Recursive Tail Algorithm of Fig. 7 finds a matching and that
2. if the matched forbidden labelings are used as representatives, then there are no clashes between tail concepts.

The following sequence of lemmas culminating in Theorem 1 establishes Part 1. The theorem actually shows that the matching between concepts in the tail and forbidden labels of C^x is unique.

Lemma 3. *Let C be a maximum class and $x \neq y$ be two dimensions in $dom(C)$. Let the concepts of $tail_x(C^y)$ be indexed by i (i.e $tail_x(C^y) = \{c_i\}$) and let $tail_x(C - y) = \{c_j\}$. Then there exist bit values a_i, a_j for the y dimension such that $tail_x(C) = \{a_i c_i\} \cup \{a_j c_j\}$. (see Fig. 8 for an example).*

Proof. First note that the sizes add up as they should:

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

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

Consider some concept $c \in tail_x(C^y)$. 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 $tail_x(C)$, then we can choose one of the extensions and map c to it. Assume that neither $0c$ and $1c$ lie in $tail_x(C)$. This means that these concepts both have an x edge to some concepts $0c', 1c'$. But then $c' \in C^y$ and there is a x edge between c and c'. Thus $c \notin tail_x(C^y)$, which provides a contradiction.

Now consider a concept $c \in tail_x(C-y)$. It might have one or two extensions back onto the full domain. In either case, any of these extensions will be in $tail_x(C)$, because removing a y dimension will not hurt an existing x edge (e.g. suppose $0c$ was the extension and was not in the tail possessing an x edge to some $0c'$, then c, c' is an x edge in $C - y$).

Finally we need to avoid mapping back to the same concept. This can only happen for concepts in $tail_x(C^y) \cap tail_x(C-y)$. These concepts have two extensions back to C and from the previous paragraph it follows that both of these extensions are in $tail_x(C)$. So we can arbitrarily choose one of the extensions to be mapped back from $tail(C^y)$ and the other from $tail(C - y)$. □

Lemma 4. *$C^x - y = (C - y)^x$ (see Fig. 9 for an illustration)*

Proof. First, we show that $C^x - y \subset (C - y)^x$. Take any $c \in C^x - y$. By the definition of restriction there exists a_y such that $a_y c \in C^x$. Next, concepts in C^x have two extensions back onto C: $0a_y c, 1a_y c \in C$. From this we immediately have by definition restriction that $0c, 1c \in C - y$ and $c \in (C - y)^x$.

Both $(C-y)^x$ and $C^x - y$ are maximum classes with domain size $|dom(C)|-2$ and VC dimension $d - 1$, thus they have the same size. This plus the fact that $C^x - y \subset (C - y)^x$ means that they are in fact equal. □

Corollary 2. *Any forbidden labeling of $(C - y)^x$ is also a forbidden labeling of C^x.*

Proof. Forbidden labelings of $(C-y)^x$ do not include a label for y. Any forbidden labeling of C^x that does not include y is then a forbidden labeling of $C^x - y$. By Lemma 4, $(C - y)^x = C^x - y$ and thus these two classes have exactly the same forbidden labelings. □

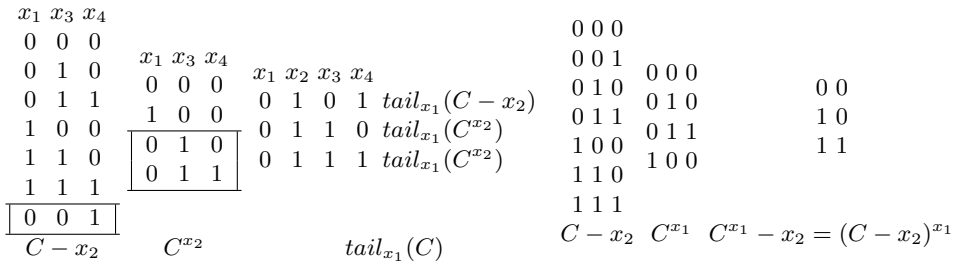

Fig. 8. illustration of Lemma 3. $tail_{x_1}(C)$ can be composed from $tail_{x_1}(C^{x_2})$ and $tail_{x_1}(C - x_2)$; class C is given in Fig. 2, tails in classes are boxed and the last column for $tail_{x_1}(C)$ indicates whether the concept comes from $tail_{x_1}(C^{x_2})$ or $tail_{x_1}(C - x_2)$

Fig. 9. illustration of the statement of Lemma 4 - $C^{x_1} - x_2 = (C - x_2)^{x_1}$; class C^{x_1} is given in the 2 last column

Lemma 5. *If we have a forbidden labeling for C^{xy} of size $d-1$, then there exists a value for the y dimension such that extending this forbidden labeling with this value gives us 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 involve 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 coordinates. If we look only at the labelings that involve y, there will be $\binom{n-2}{d-1}$ of them. This is also the total number of forbidden labelings of size $d-1$ for C^{xy}.

A mapping is constructed as follows: A forbidden labeling of size d for C^x that involves a y dimension becomes a forbidden labeling of size $d-1$ for C^{xy} by discarding the y dimension. Assume that the newly formed labeling is not forbidden in C^{xy}. Then by extending the concept that contains it with both one and zero back to C^x, we will hit the original forbidden set, thus forming a contradiction.

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 realizes the statement of the lemma. □

Theorem 1. *For any maximum class C of VC dimension d and any $x \in dom(C)$ it is possible to find a unique matching between the $\binom{n-1}{d}$ concepts in $tail_x(C)$ and the $\binom{n-1}{d}$ forbidden labelings of size d for C^x, such that every concept is matched to some forbidden labeling it contains and all forbidden labelings are matched.*

Proof. The proof will proceed by induction on d and $n = |dom(C)|$.

Base cases: If $n = d$, then we have a complete hypercube which has no tail and the matching is empty. If $d = 0$, then there is a single concept which is the tail and this concept is matched to the empty set.

Inductive hypothesis: For any maximum class C', such that $VCdim(C') < d$ or $|dom(C')| < n$, the statement of the theorem holds.

Inductive step. Let $x, y \in dom(C)$ and $x \neq y$. By Lemma 3, we can compose $tail_x(C)$ from $tail_x(C^y)$ and $tail_x(C - y)$. Since $VCdim(C^x) = d - 1$ and $|dom(C - x)| = n - 1$, we can use the inductive hypothesis for these classes and assume that the desired matchings already exist for $tail_x(C^y)$ and $tail_x(C - y)$.

Now we need to combine these matchings to form a matching for $tail_x(C)$. See Fig. 7 for a description of this process. Concepts in $tail_x(C - y)$ are matched to forbidden labelings of $(C - y)^x$ of size d. By Lemma 2, 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 $tail_x(C)$ without alterations. On the other hand, $tail_x(C^y)$ 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 5 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 $tail_x(C^y)$ is in the $tail_x(C)$, there are no problems (i.e. that concept will be the concept of Lemma 5, since the other concept is in C^x and thus does not contain any forbidden labelings). There still is the possibility that both extensions are in $tail_x(C)$. From the proof of Lemma 3 we see that this only happens to the concepts that are in $tail_x(C^y) \cap tail_x(C - y)$. Then, by Lemma 5 we can figure out which extension corresponds to the forbidden labeling involving y and use that for the $tail_x(C^y)$ matching. The other extension will correspond to the $tail_x(C - y)$ matching. Essentially, where before the Lemma 3 told us to map the intersection $tail_x(C^y) \cap tail_x(C - y)$ back to $tail_x(C)$ by assigning a bit arbitrarily, now we choose a bit in a specific way.

Now we know that a matching exists. Uniqueness of the matching can also be argued from inductive assumptions on uniqueness for $tail_x(C^y)$ and $tail_x(C - y)$. □

Theorem 2. *The Recursive Algorithm of Fig. 6 returns a representation mapping that satisfies both conditions of the Main Definition.*

Proof. Proof by induction on $d = VCdim(C)$. 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 $VCdim(C^x) = d-1$. 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: It is easily seen that the algorithm uses all possible sets that don't involve x and are of size $< d$ as representatives for $0C^x$. The concepts of $1C^x$ are represented by all sets of size $\leq d$ that contain x. Finally the concepts in $tail_x(C)$ are represented by sets of size equal d that don't contain x. This shows that all sets of size up to d represent some concept.

No clashes condition: By the inductive assumption there cannot be any clashes internally within each of the subclasses $0C^x$ and $1C^x$. 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 1, we know

that concepts in the tail are assigned to representatives that define a forbidden labeling for C^x, thus clashes between the tail and $0C^x$, $1C^x$ are prevented. Finally, we need to argue that there cannot be any clashes internally within the tail. By Theorem 1, the matching between concepts in $tail_x(C)$ and forbidden labeling of C^x is unique. So if this matching would result 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 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 create a new valid matching, thus contradicting the uniqueness. □

5 Miscellaneous Lemmas

We conclude with some miscellaneous lemmas. The first one shows that the representatives constructed by our algorithm induce a nesting of maximum classes. The algorithm that iteratively removes a lowest degree vertex (see introduction and Fig. 4) is not guaranteed to construct representatives with this property.

Lemma 6. *Let C be a maximum concept class with VC dimension d and let r be a representation mapping for C produced by the Recursive Algorithm. Let $C_k = \{c \in C \text{ s. t. } |r(c)| \leq k\}$. Then C_k is a maximum concept class with VC dimension k.*

Proof. Proof by induction on d. Base case $d = 0$, class has only one concept and the statement is trivial.

Let $x \in dom(C)$ be the first dimension along which the Recursive Algorithm works (i.e. it first recursively finds representatives for C^x). Then we can use the inductive assumption for C^x.

Let $0 < k < d$ (extreme values of k are trivial). Consider which concepts in C get representatives of size $\leq k$. They are all the concepts in $0C^x$ that got representatives of size $\leq k$ in the mapping for C^x plus all the concepts in $1C^x$ that got representatives of size $\leq k-1$ (as $1C^x$ representatives have size $+1$ compared to the $0C^x$ representatives). Thus, our class C_k is formed in the following manner - $C_k = 0C^x_k \cup 1C^x_{k-1}$. By inductive assumption C^x_k and C^x_{k-1} are maximum classes with VC dimension k and $k-1$. Furthermore, definition of C_k implies that $C^x_{k-1} \subset C^x_k$.

$|C_k| = |0C^x_k| + |1C^x_{k-1}| = \binom{n-1}{\leq k} + \binom{n-1}{\leq k-1} = \binom{n}{\leq k}$. Thus C_k has the right size and $VCdim(C_k) \geq k$. It remains to show that C_k does not shatter any set of size $k+1$. Consider all sets of dimensions of size $k+1$ that does not involve x. It would have to be shattered by $C_k - x = C^x_k \cup C^x_{k-1} = C^x_k$, which is impossible. Now consider sets of size $k+1$ that do involve x. All the 1 values for the x coordinate happen in the $1C^x_{k-1}$ part of C_k. Thus removing the x coordinate we see that C^x_{k-1} would have to shatter a set of size k, which is again impossible. □

It was known previously that the one-inclusion graph for maximum classes is connected ([Gur97]). We are able to extend that statement to a stronger one in

Lemma 8. Furthermore, this lemma is a known property of simple linear arrangements, which are restricted maximum classes (i.e. not all maximum classes can be represented as a simple linear arrangement [Flo89]). But first a necessary technical lemma is proven.[7]

Lemma 7. *For any maximum class C and $x \in dom(C)$, restricting wrt x does not change the incident dimension sets of concepts in $tail_x(C)$, i.e. $\forall c \in tail_x(C)$, $I_C(c) = I_{C-x}(c-x)$*

Lemma 8. *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. From Lemma 7 it follows that there are no edges between any concepts in $tail_{x=0}(C) - x$ and concepts in $tail_{x=1}(C) - x$.

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

Note that since there are no edges between concepts in $tail_{x=0}(C) - x$ and $tail_{x=1}(C) - x$, any segment of concepts in $tail_x(C) - x$ 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 can't be empty.

We can now construct a shortest path P' between c and c' from the path P. When $c(x) = c'(x)$ we can extend the concepts in P with $x = c(x)$ to obtain a path P' between c and c' in C of the same length. Note that from the above discussion all concepts of P from $tail_x(C) - x$ must be concepts that label x with bit $c(x)$.

If $c(x) \neq c'(x)$, let P be as above. We first claim that P must contain a concept $\tilde{c}$ in C^x, because if all concepts in P lied in $tail_x(C)$ then this would imply an edge between a concept in $tail_{x=0}(C) - x$ and a concept in $tail_{x=1}(C) - x$. We now construct a new path P' in C as follows: Extend the concepts up to $\tilde{c}$ in P with $x = c(x)$; then cross to the sibling concept $\tilde{c}'$ which disagrees with $\tilde{c}$ only on its x-dimension; finally extend the concepts in path P from $\tilde{c}$ onwards with $x = 1$. □

Acknowledgments: Thanks to Sally Floyd for personal encouragement and brilliant insights and to Sanjoy DasGupta for discussions leading to Lemma 8.

[7] Additional notation is as follows. $I_C(c)$ - is the set of incident dimensions, that is set of labels for all edges of c in C. $E(C)$ - set of all edges in a class.

Bibliography

[Flo89] 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.

[FW95] S. Floyd and M. K. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[Gur97] 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.

[HLW94] D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0,1\}$ functions on randomly drawn points. *Information and Computation*, 115(2):284–293, 1994.

[Lan03] John Langford. Tutorial on practical prediction theory for classification. *ICML*, 2003.

[LLS02] 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.

[LW86] N. Littlestone and M. K. Warmuth. Relating data compression and learnability. Unpublished manuscript, obtainable at http://www.cse.ucsc.edu/~manfred/pubs/lrnk-olivier.pdf, June 10 1986.

[MST02] Mario Marchand and John Shawe-Taylor. The Set Covering Machine. *Journal of Machine Learning Research*, 3:723–746, 2002.

[Sau72] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory (A)*, 13:145–147, 1972.

[SMJST03] Marina Sokolova, Mario Marchand, Nathalie Japkowicz, 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.

[Vap82] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.

[VC71] 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.

[vLBS04] Ulrike von Luxburg, Olivier Bousquet, and Bernard Schölkopf. A compression approach to support vector model selection. *Journal of Machine Learning Research*, 5(Apr):293–323, 2004.

[War03] 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.

[War04] 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.

[Wel87] E. Welzl. Complete range spaces. Unpublished notes, 1987.